

TUGAS APLIKASI KOMPUTER

BAB LaTeX dan Markdown



Muhamad Naufal Zaky

22305141012

Matematika B 2022

**PRODI MATEMATIKA
DEPARTEMEN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2023/2024**

DAFTAR ISI

1 Pekan 2: Pengenalan Software Euler Math Toolbox	2
2 Pekan 3-4: Penggunaan Software EMT untuk Aljabar	18
3 Pekan 5-6: Penggunaan Software EMT untuk Plot 2D	95
4 Pekan 7-8: Penggunaan Software EMT untuk Plot 3D	190
5 Pekan 9-10: Menggunakan EMT untuk Kalkulus	268
6 Pekan 11-12: Menggunakan EMT untuk Geometri	367
7 Pekan 13-14: Menggunakan EMT untuk Statistika	450

BAB 1

PEKAN 2: PENGENALAN SOFTWARE EULER MATH TOOLBOX

article
eumat

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 Tutorials 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 meng-klik 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: $\int x^3 dx = C + \frac{x^4}{4}$
- 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

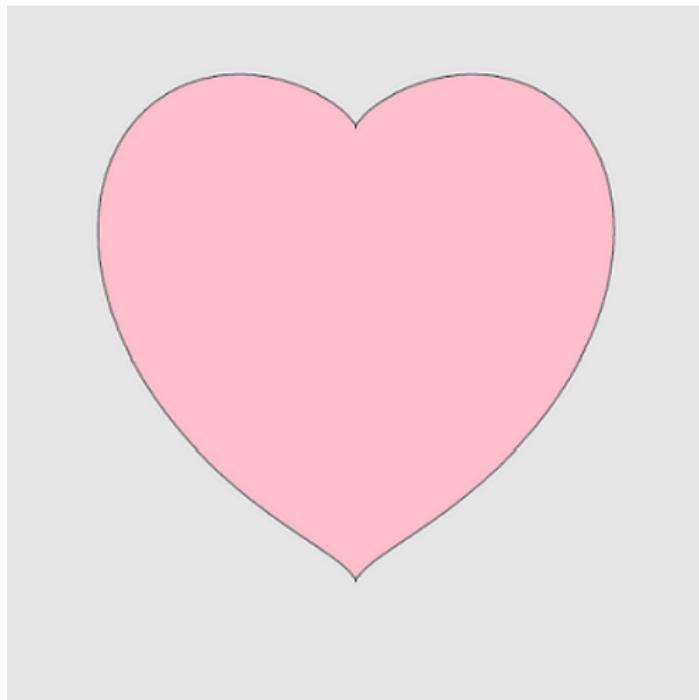
Sub-Judul

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

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

$$\int x^3 dx = C + \frac{x^4}{4}$$

<http://www.euler-math-toolbox.de>
See: <http://www.google.de> | Google



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=3, b=5, c=7
```

```
3  
5  
7
```

```
>s=(a+b+c)/2
```

```
7.5
```

```
>L=(s*(s-a)*(s-b)*(s-c))
```

42.1875

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.

```
>(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.

```
>&factor(x^6+x^2+1)
```

$$x^6 + x^2 + 1$$

```
>expr&=x^2*cos(x*y);  
>&diff(expr,x) // derivative at s
```

$$2x^2 \cos(xy) - x^2 y \sin(xy)$$

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

$$\begin{matrix} 1 & 2 & 1 \\ 3 & 4 & 2 \end{matrix}$$

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 mengatramilan 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 "deformat" atau "reset".

```
>longestformat; pi, deformat; 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 Variabel

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

```
>listvar
```

A	Type: Real Matrix (2x2)
r	1.25
s	7.5
a	4
b	15
c	2
L	42.1875
expr	$x^2 \cos(x \cdot y)$
B	Type: Real Matrix (2x1)
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,  
0.193585, 0.463387, 0.095153, 0.595017]
```

```
>normal(10)
```

```
[-0.495418, 1.6463, -0.390056, -1.98151, 3.44132, 0.308178,  
-0.733427, -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
```

$$(b + a)^2$$

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

$$b^2 + 2ab + a^2$$

$$(x + 2)(x + 3)$$

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

$$[x = \frac{-\sqrt{b^2 - 4ac} - b}{2a}, x = \frac{\sqrt{b^2 - 4ac} - b}{2a}]$$

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

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

$$a^2 - b^2$$

```
>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]

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

$$2^3 \cdot 5^3$$

```
>::: 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)!)
```

$$\frac{(m - 3)(m - 2)(m - 1)m}{24}$$

```
>::: 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) //menyederhanakan fungsi tri
```

$$\frac{\sin^4(x) + \cos^4(x)}{\cos^3(x)}$$

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

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

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

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

$$x^2 - x + 1$$

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
```

$$\frac{(b + a)^3 + 1}{b + a + 1}$$

$$b^2 + (2a - 1)b + a^2 - a + 1$$

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

$$\frac{3x^2 - (x^3 + 1)}{(x + 1)^2}$$

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

$$\begin{array}{r} 3 \quad \quad \quad 2 \\ 2x^3 - 3x^2 + 6x \\ \hline 6 \end{array}$$

Tampilan Matematika Simbolik dengan LaTeX

Anda dapat menampilkan hasil perhitungan simbolik secara lebih bagus menggunakan LaTeX. Untuk melakukannya, 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
```

$$(b + a)^2$$

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

$$b^2 + 2ab + a^2$$

$$(x + 2)(x + 3)$$

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

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

```
>$& (a^2-b^2)/(a+b), $&ratsimp(%)
```

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

$$a - b$$

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.

```
>k = 2
```

2

```
>z = k;  
>z = 1/2*(x^2 + y^2)
```

5.43480220054

```
>x = 1, y = 2
```

1
2

```
>f = y/x + x*y
```

4

```
>x = 1, y = pi, z = 2, sin(pi)
```

1
3.14159265359
2
0

```
>g = x^2*sin(y*z)
```

0

BAB 2

PEKAN 3-4: PENGGUNAAN SOFTWARE EMT UNTUK ALJABAR

article
eumat

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))))
```

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

Baris Perintah

Baris perintah Euler terdiri dari satu atau beberapa perintah Euler diikuti dengan titik koma ";" atau koma ",". Titik koma mencegah pencetakan hasilnya. 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 yang kosong. Baris perintah berikut mencetak dua hasilnya.

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

50.2654824574
100.530964915

Baris perintah dijalankan sesuai urutan yang ditekan pengguna kembali. Jadi, Anda mendapatkan nilai baru setiap kali Anda 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,
```

1.41666666667
1.41421568627
1.41421356237

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

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

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

Garis yang dimulai dengan %% tidak akan terlihat sama sekali.

81

Euler mendukung loop di baris perintah, asalkan cocok ke dalam satu baris atau multi-baris. Tentu saja, pembatasan ini tidak berlaku dalam program. Untuk informasi lebih lanjut lihat pendahuluan berikut.

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

```
1.5  
1.4166666667  
1.41421568627  
1.41421356237  
1.41421356237
```

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

```
>x := 1.5; // comments go here before the ...  
>repeat xnew:=(x+2/x)/2; until xnew~≈x; ...  
> x := xnew; ...  
>end; ...  
>x,
```

```
1.41421356237
```

Struktur bersyarat juga berfungsi.

```
>if E^pi>pi^E; then "Thought so!", endif;
```

```
Thought so!
```

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 tanda kurung atau tanda kurung buka dan tutup akan disorot. Juga, perhatikan baris status. Setelah tanda kurung buka dari fungsi sqrt(), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan kunci 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 pada 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 bersamaan 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 ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilainya, atau gunakan fungsi rad(x). Fungsi akar kuadrat disebut sqrt 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 jarak antar perintah diharapkan.

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma menekan keluaran 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

$$e^2 \cdot \left(\frac{1}{3 + 4 \log(0.6)} + \frac{1}{7} \right)$$

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

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

8.77908249441

Untuk menghitung ekspresi rumit seperti

$$\left(\frac{\frac{1}{7} + \frac{1}{8} + 2}{\frac{1}{3} + \frac{1}{2}} \right)^2 \pi$$

Anda harus memasukkannya dalam formulir baris.

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

23.2671801626

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

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

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

```
0.47619047619  
10/21
```

Perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi terbuat dari operator dan fungsi. Jika perlu, harus berisi tanda kurung untuk memaksakan urutan eksekusi yang benar. Jika ragu, memasang 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 meliputi

```
+ unary or operator plus  
- unary or operator minus  
*, /  
. the matrix product  
a^b power for positive a or integer b (a**b works too)  
n! the factorial operator
```

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  
conj,re,im,arg,conj,real,complex  
beta,betai,gamma,complexgamma,ellrf,ellf,ellrd,elle  
bitand,bitor,bitxor,bitnot
```

Beberapa perintah memiliki alias, misalnya 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 Real

Tipe data primer pada 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)
```

```
1.01010101010101010101010101010101010101010101010101010101010101*2^-2
```

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

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

Strings

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

```
>"A string can contain anything."
```

```
A string can contain anything.
```

String dapat digabungkan dengan `|` atau dengan `+`. Ini juga berfungsi dengan angka, yang dalam hal ini diubah menjadi string.

```
>"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 print juga mengubah angka menjadi string. Ini dapat memerlukan sejumlah digit dan sejumlah tempat (0 untuk keluaran padat), dan optimalnya satuan.

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

```
Golden Ratio : 1.61803
```

Ada string khusus `None` yang tidak dicetak. Itu dikembalikan oleh beberapa fungsi, ketika hasilnya tidak menjadi masalah. (Ini dikembalikan secara otomatis, jika fungsi tidak memiliki pernyataan `return`.)

```
>none
```

Untuk mengonversi string menjadi angka, cukup evaluasi saja. Ini juga berfungsi untuk ekspresi (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"\u03b1; = " + 45 + u"\u00b0;" // pdfLaTeX mungkin gagal menampilkan secara benar
```

```
= 45°
```

```
I
```

Di komentar, entitas yang sama seperti , dll. dapat digunakan. Ini mungkin merupakan alternatif cepat untuk Lateks. (Detail lebih lanjut di 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 angka 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 secara benar
```

```
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 "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 ini, kita menggunakan kondisi isprime(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]
```

Output Formats

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

```
>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 kita gunakan operator "longest" untuk menampilkan hasilnya 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" berfungsi 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 "longestformat" juga mengatur format skalar.

```
>longestformat; pi
```

```
3.141592653589793
```

Sebagai referensi, berikut adalah daftar format keluaran terpenting.

```
shortestformat shortformat longformat, longestformat  
format(length,digits) goodformat(length)  
fracformat(length)  
defformat
```

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

Namun 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 short operators yang hanya mencetak satu nilai. Operator "longest" akan mencetak semua digit nomor yang valid.

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

4.934802200544679

Ada juga short operator untuk mencetak hasil dalam format pecahan. Kami sudah 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 secara tepat. Kesalahannya bertambah sedikit, seperti yang Anda lihat pada 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 menyadarinya. 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+0.1-1
```

0

Expressions

String atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang dapat dievaluasi dengan EMT. Untuk ini, gunakan tanda kurung setelah ekspresi. Jika Anda ingin 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 memberikan 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" selain 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 mencatat bahwa koleksi panggilan (dibahas di tempat lain) dapat berisi ekspresi. Jadi kita bisa membuat contoh di atas 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

Berdasarkan 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
```

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

Bentuk ekspresi khusus memungkinkan variabel apa pun sebagai parameter yang tidak disebutkan namanya untuk mengevaluasi ekspresi, bukan 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 memerlukan ekspresi dalam "x".

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

Sebuah ekspresi tidak perlu bersifat simbolis. Hal ini diperlukan, jika ekspresi berisi fungsi, yang hanya diketahui di kernel numerik, bukan di Maxima.

Matematika Simbolik

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 sintaksis antara sintaksis asli Maxima dan sintaksis default ekspresi simbolik di EMT.

Matematika simbolik diintegrasikan 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 bilangan yang sangat besar.

```
>$&44!
```

```
26582715747884487680436258110146158903196385280000000000
```

Dengan cara ini, Anda dapat menghitung hasil yang besar dengan tepat. Mari kita menghitung

$$C(44, 10) = \frac{44!}{34! \cdot 10!}$$

```
> $& 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 padanya. Misalnya, coba klik dua kali pada "&binomial" di baris perintah sebelumnya. Ini membuka dokumentasi Maxima yang disediakan oleh penulis program tersebut.

Anda akan mengetahui bahwa cara berikut juga bisa dilakukan.

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

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

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

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

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

120

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

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

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

Untuk mencetak ekspresi simbolik dengan LaTeX, gunakan \$ di depan & (atau Anda dapat menghilangkan &) sebelum perintah. Jangan jalankan 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 memerlukan sintaksis kompleks dalam satu ekspresi, Anda dapat mengapit ekspresi tersebut di "...". Menggunakan lebih dari sekadar ekspresi sederhana bisa saja dilakukan, namun sangat tidak disarankan.

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

25

Untuk kelengkapannya, kami mencatat bahwa ekspresi simbolik dapat digunakan dalam program, namun perlu diapit dalam tanda kutip. Selain itu, akan jauh lebih efektif untuk memanggil Maxima pada waktu compile jika memungkinkan.

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

$$4 (x + 1)^3$$

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))}
```

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

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

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

2432902008176640000

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

$$\begin{matrix} 8 & 4 & 2 \\ 2 & 3 & 5 & 7 \end{matrix}$$

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

$$\begin{matrix} 18 & 8 & 4 & 2 \\ 2 & 3 & 5 & 7 & 11 & 13 & 17 & 19 \end{matrix}$$

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

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

$$g^2$$

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

$$\begin{matrix} 3 & x \\ x & E \end{matrix}$$

$$x^3 e^x$$

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

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

$$\begin{array}{c} 5 \\ 125 \text{ E} \end{array}$$

$$125 e^5$$

$$18551.64488782208$$

```
>fx(5)
```

$$18551.6448878$$

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

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

$$\begin{array}{c} 10 \\ 1000 \text{ E} \end{array} - \begin{array}{c} 5 \\ 125 \text{ E} \end{array}$$

$$2.20079141499189e+7$$

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

$$x \left(x^2 + 6x + 6\right) e^x$$

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

```
>tex(fx)
```

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

Ekspresi simbolik dapat dievaluasi seperti halnya ekspresi numerik.

```
>fx(0.5)
```

0.206090158838

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

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

$$\frac{\sqrt{e}}{8}$$

Penugasannya juga bisa bersifat simbolis.

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

$$(t + 1)^3 e^{t+1}$$

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

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

$$\left[x = \frac{-\sqrt{17} - 1}{2}, x = \frac{\sqrt{17} - 1}{2} \right]$$

Bandingkan dengan perintah numerik "solve" di Euler, yang memerlukan nilai awal, dan 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 membacakan tugas $x = \text{dst}$. Jika Anda tidak memerlukan hasil numerik untuk perhitungan lebih lanjut, Anda juga dapat membiarkan Maxima menemukan nilai numeriknya.

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

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

[-3.23607, 1.23607]

$$[x = -3.23606797749979, x = 1.23606797749979]$$

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

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

$$\frac{\sqrt{5} - 1}{2}$$

$$\frac{\sqrt{5} - 1}{2}$$

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]
```

2

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

```
> $& diff((x^3-1)/(x+1), x) //turunan bentuk pecahan
```

$$\frac{3x^2}{x+1} - \frac{x^3-1}{(x+1)^2}$$

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

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

```
> $& factor(%)
```

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

Fungsi

Dalam EMT, fungsi adalah program yang didefinisikan dengan perintah "function". Ini bisa berupa fungsi satu baris atau fungsi multibaris.

Fungsi satu baris dapat berupa numerik atau simbolik. Fungsi satu baris numerik didefinisikan oleh ":=".

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

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

```
>f(2)
```

4.472135955

Fungsi ini juga dapat digunakan untuk vektor, mengikuti bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi tersebut divektorkan.

```
>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 dapat diplot. Daripada ekspresi, kita hanya perlu memberikan nama fungsinya.

Berbeda dengan ekspresi simbolik atau numerik, nama fungsi harus diberikan dalam string.

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

0.786151377757

Secara default, jika Anda perlu menimpa fungsi bawaan, Anda harus menambahkan kata kunci "overwrite". Fungsi overwriting bawaan berbahaya dan dapat menyebabkan masalah pada fungsi lain yang bergantung pada fungsi tersebut.

Anda masih dapat memanggil fungsi bawaan sebagai "...", jika fungsi tersebut ada di inti Euler.

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

0.707106781187

Sebaiknya kita menghilangkan definisi ulang dari sin.

```
>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 akan menggunakan nilai default.

```
>f(4)
```

16

Menyetelnya akan menimpa nilai default.

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

80

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

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

16

Jika suatu variabel bukan parameter, maka harus bersifat global. Fungsi satu baris dapat melihat variabel global.

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

24

Namun parameter yang ditetapkan mengesampingkan nilai global.

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

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

20

Fungsi simbolik didefinisikan dengan "&=". Mereka didefinisikan di Euler dan Maxima, dan bekerja di kedua itu. 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
```

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

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

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: mengintegralkan
```

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

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

$$0.703467422498$$

Berikut ini juga berfungsi, karena Euler menggunakan ekspresi simbolik dalam fungsi g , jika tidak menemukan variabel simbolik g , dan jika terdapat fungsi simbolik g .

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

$$0.703467422498$$

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

$$(2x - 1)^n$$

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

$$(x + 2)^n$$

```
>${&P(x, 4)}, ${&expand(%)}
```

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

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

625

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

$$16x^4 - 31x^3 + 30x^2 + 4x + 9$$

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

$$16x^4 - 33x^3 + 18x^2 - 20x - 7$$

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

$$(x + 2)^3 (2x - 1)^4$$

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

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

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

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

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

$$x^3 - x$$

Dengan `&=` fungsinya bersifat simbolis, dan dapat digunakan dalam ekspresi simbolik lainnya.

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

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

Dengan `:=` fungsinya numerik. Contoh yang baik adalah integral tertentu

$$f(x) = \int_1^x t^t dt,$$

yang tidak dapat dievaluasi secara simbolis.

Jika kita mendefinisikan ulang fungsi tersebut dengan kata kunci "map" maka 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 fungsinya bisa 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, kita ingin menggunakan fungsi untuk vektor di satu tempat, dan untuk elemen individual di tempat lain. Hal ini dimungkinkan dengan parameter vektor.

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

$$y^2 - xy + y + x^2$$

Fungsi simbolik seperti ini dapat digunakan untuk variabel simbolik.

Namun fungsinya juga dapat digunakan untuk vektor numerik.

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

17

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

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

$$\text{diff(expr, } y, \ 2) \ + \ \text{diff(expr, } x, \ 2)$$

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

0

Namun 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)
```

$$10 \ (y^2 + x)^3 \ (9 y^2 + x + 2)$$

untuk meringkas

- &= defines symbolic functions,
- := defines numerical functions,
- &&= defines purely symbolic functions.

Menyelesaikan Expressions

Ekspresi dapat diselesaikan secara numerik dan simbolis.

Untuk menyelesaikan ekspresi sederhana dari satu variabel, kita dapat menggunakan fungsi solve(). Dibutuhkan nilai awal untuk memulai pencarian. Secara internal, solve() menggunakan metode secant.

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

1.41421356237

Ini juga berfungsi untuk ekspresi simbolik. 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)
```

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

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

$$\left[\left[x = -\frac{ce}{b(d-5) - ae}, y = \frac{c(d-5)}{b(d-5) - ae} \right] \right]$$

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

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

Sekarang kita mencari titik yang polinomialnya adalah 2. Dalam solve(), 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 simbolik dalam bentuk simbolik mengembalikan daftar solusi. Kami menggunakan pemecah simbolis solve() yang disediakan oleh Maxima.

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

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

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

```
>longest sol()
```

```
-0.6180339887498949 1.618033988749895
```

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

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

```
0
```

Penyelesaian sistem persamaan secara simbolis dapat dilakukan dengan vektor persamaan dan solver simbolis solve(). Jawabannya adalah daftar daftar persamaan.

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

```
[[x = -1, y = 3], [x = 1, y = 1], [x = 0, y = 2]]
```

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

$$a^x - x^a = 0, 1$$

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 berfungsi dengan ekspresi. Namun 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])
```

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

```
>$&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 pertidaksamaan
```

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

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

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

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

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

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

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

```
>&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 pembahasan rinci 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 ]
```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$$

Hasil kali matriks dilambangkan dengan titik.

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

$$\begin{matrix} 3 \\ 4 \end{matrix}$$

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

$$[3, 4]$$

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

$$\begin{matrix} -2 & 1 \\ 1.5 & -0.5 \end{matrix}$$

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

$$\begin{matrix} 11 \\ 25 \end{matrix}$$

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

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

Poin utama dari bahasa matriks adalah semua fungsi dan operator bekerja elemen demi 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 kolom)
```

0.333333	0.666667
0.75	1

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

-2
2.5

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

-2
2.5

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

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

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

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

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

$$\begin{matrix} 1 & 4 \\ 9 & 16 \end{matrix}$$

Ini bukan hasil kali matriks, melainkan perkalian elemen demi elemen. Hal yang sama juga berlaku untuk vektor.

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

$$\begin{matrix} 9 \\ 16 \end{matrix}$$

Jika salah satu operan adalah vektor atau skalar, maka operan tersebut diperluas secara alami.

```
>2*A
```

$$\begin{matrix} 2 & 4 \\ 6 & 8 \end{matrix}$$

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

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

$$\begin{matrix} 1 & 4 \\ 3 & 8 \end{matrix}$$

Jika ini adalah vektor baris, maka diterapkan ke semua kolom A.

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

$$\begin{matrix} 2 & 6 \\ 6 & 12 \end{matrix}$$

Kita dapat membayangkan perkalian ini seolah-olah vektor baris v telah diduplikasi untuk membentuk matriks yang berukuran sama dengan A.

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

```
1          2  
1          2
```

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

```
1          4  
3          8
```

Hal ini juga berlaku untuk dua vektor dimana yang satu adalah vektor baris dan yang lainnya adalah vektor kolom. Kita menghitung $i \cdot j$ untuk i, j dari 1 sampai 5. Caranya adalah dengan mengalikan 1:5 dengan transposenya. Bahasa matriks Euler secara otomatis menghasilkan tabel nilai.

```
>(1:5)*(1:5)' // hasil kali 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 produk matriks!

```
>(1:5).(1:5)' // hasil kali 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]
```

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.

Kita mendapatkan vektor 0 dan 1, dimana 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]
```

Dari vektor tersebut, "nonzeros" memilih elemen bukan nol.

Dalam hal ini, kita mendapatkan indeks semua elemen 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 yang sesuai dalam t.

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

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

Sebagai contoh, mari kita cari semua kuadrat bilangan 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 bilangan bulat. Ia menggunakan floating point presisi ganda secara internal. Namun, seringkali hal ini sangat berguna.

Kita dapat memeriksa primalitasnya. Mari kita cari tahu, berapa banyak persegi 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 mengatur 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 lain yang berguna adalah ekstrem, 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]
```

Namun 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 sebuah matriks, kita dapat menumpuk satu matriks di atas matriks lainnya. Jika keduanya tidak memiliki jumlah kolom yang sama, maka kolom yang lebih pendek akan diisi dengan 0.

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

1	2	3
1	2	3

Demikian pula, kita dapat melampirkan matriks ke matriks lain secara berdampingan, jika keduanya mempunyai 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 jumlah barisnya tidak sama, matriks yang lebih pendek diisi dengan 0.

Ada pengecualian untuk aturan ini. Bilangan real yang melekat pada suatu 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 ekspresi vektor untuk vektor kolom.

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

1	1
2	4
3	9

Untuk mendapatkan ukuran A, kita bisa menggunakan 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
```

Masih banyak fungsi lain yang menghasilkan matriks.

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

```
1 1  
1 1
```

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

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

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

Matriks bilangan acak juga dapat dihasilkan dengan acak (uniform distribution) atau normal (distribusi Gauß).

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

```
0.66566 0.831835  
0.977 0.544258
```

Berikut adalah fungsi lain yang berguna, yang merestrukturisasi elemen matriks menjadi matriks lain.

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

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

Dengan fungsi berikut, kita dapat menggunakan fungsi ini dan fungsi dup untuk menulis fungsi rep(), yang mengulangi 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, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3]  
[1, 1, 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]
```

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 di drop(v,i) mengacu pada indeks elemen di v, bukan nilai elemen. Jika Anda ingin menghapus elemen, Anda perlu mencari elemennya terlebih dahulu. Fungsi indexof(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 memasukkan indeks di luar rentang (seperti 0), indeks ganda, atau indeks yang tidak diurutkan.

```
>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 menghasilkan matriks diagonal.

Kita 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 kita atur 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 suatu matriks juga dapat diekstraksi dari matriks tersebut. Untuk mendemonstrasikannya, kami menyusun ulang vektor 1:9 menjadi matriks 3x3.

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

1	2	3
4	5	6
7	8	9

Sekarang kita dapat mengekstrak diagonalnya.

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

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

Misalnya, kita dapat membagi matriks dengan diagonalnya. Bahasa matriks menjaga agar vektor kolom diterapkan pada 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 juga berfungsi untuk input matriks dan vektor, jika hal ini masuk akal. 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 suatu fungsi (alternatifnya menggunakan ekspresi).

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

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

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

$$s = t^3 - t$$

```
>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 diperluas ke 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 dengan perkalian matriks. Hasil kali matriks dilambangkan dengan titik "." di EMT.

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

55

Secara default, vektor baris dicetak dalam format ringkas.

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

[1, 2, 3, 4]

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

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

5
25

Untuk mengubah urutan matriks kita menggunakan apostrof.

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

1
2
3
4

Jadi kita dapat menghitung matriks A dikalikan 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 dengan $v.v'$.

```
>v' . v
```

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

$v.v'$ menghitung norm v kuadrat untuk vektor baris v. Hasilnya adalah vektor 1x1, yang berfungsi seperti bilangan real.

```
>v.v'
```

30

Ada juga fungsi norm (bersama dengan banyak fungsi Aljabar Linier lainnya).

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

30

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ringkasan peraturannya.

- Suatu fungsi yang diterapkan pada vektor atau matriks diterapkan pada setiap elemen.
- Operator yang mengoperasikan dua matriks dengan ukuran yang sama diterapkan secara berpasangan pada elemen-elemen matriks.
- Jika kedua matriks mempunyai dimensi yang berbeda, keduanya diekspansi secara wajar sehingga mempunyai ukuran yang sama.

Misalnya, nilai skalar dikalikan vektor dengan mengalikan nilai setiap elemen vektor. Atau matriks dikalikan vektor (with *, not .) 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 vektor kolom memperluas keduanya dengan cara menduplikasi.

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

1	2	3
2	4	6
3	6	9

Perhatikan bahwa perkalian skalar menggunakan perkalian matriks, bukan *!

```
>v.v'
```

14

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

```
sum,prod computes the sum and products of the rows  
cumsum,cumprod does the same cumulatively  
computes the extremal values of each row  
extrema returns a vector with the extremal information  
diag(A,i) returns the i-th diagonal  
setdiag(A,i,v) sets the i-th diagonal  
id(n) the identity matrix  
det(A) the determinant  
charpoly(A) the characteristic polynomial  
eigenvalues(A) the eigenvalues
```

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

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

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

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

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

Untuk menggabungkan matriks dan vektor terdapat operator "|" Dan "_" .

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

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

Elemen-elemen 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 tersebut. Untuk matriks, ini mengembalikan baris ke-i yang lengkap dari matriks tersebut.

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

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

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

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

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

A short form for : menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

```
4
```

Matriks juga dapat 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 pada tabel, mari kita atur ulang ke format default, dan hitung tabel nilai sinus dan kosinus. Perhatikan bahwa sudut dinyatakan dalam radian secara default.

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

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

Sekarang kita 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, kita menghitung $t[j]^i$ untuk i dari 1 hingga n . Kita mendapatkan sebuah matriks, yang setiap barisnya merupakan tabel t^i untuk satu i . Yaitu, matriks memiliki elemen

$$a_{i,j} = t_j^i, \quad 1 \leq j \leq 101, \quad 1 \leq i \leq n$$

Fungsi yang tidak berfungsi untuk masukan vektor harus "vectorized". Hal ini dapat dicapai dengan kata kunci "map" dalam definisi fungsi. Kemudian fungsi tersebut akan dievaluasi untuk setiap elemen parameter vektor.

Integrasi numerik integral() hanya berfungsi untuk batas interval skalar. Jadi kita perlu membuat vektorisasi-nya.

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

Kata kunci "map" membuat vektorisasi fungsi tersebut. Fungsinya sekarang akan berfungsi untuk vektor bilangan.

```
>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 secara 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 dari indeks, Euler akan mengembalikan baris matriks yang sesuai.

Di sini kita menginginkan baris pertama dan kedua A.

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

1	2	3
4	5	6

Kita bahkan dapat menyusun ulang A menggunakan vektor indeks. Tepatnya, kita tidak mengubah A di sini, namun menghitung versi A yang disusun ulang.

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

7	8	9
4	5	6
1	2	3

Trik indeks juga berfungsi 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

Alternatifnya, 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 submatriks A ke beberapa nilai. Ini sebenarnya mengubah matriks A yang disimpan.

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

4	2	3
4	5	6
7	8	9

Kita juga dapat memberikan nilai pada baris A.

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

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

Kita bahkan dapat menetapkan sub-matriks jika ukurannya sesuai.

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

5	6	-1
7	8	6
7	8	9

Selain itu, beberapa shortcuts diperbolehkan.

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

0	0	-1
0	0	6
7	8	9

Warning : Indeks di luar batas mengembalikan matriks kosong, atau pesan kesalahan, bergantung pada pengaturan sistem. Standarnya adalah pesan kesalahan. Namun perlu diingat 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 Mengacak

Fungsi sort() mengurutkan vektor baris.

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

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

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

Mari kita mengacak sebuah vektor.

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

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

Indeks berisi urutan v.

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

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

Ini juga berfungsi untuk vektor string.

```
>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 double entrie agak acak.

```
>ind
```

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

Fungsi unique mengembalikan daftar elemen unique 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 juga berfungsi untuk vektor string.

```
>unique(s)
```

```
a
aa
d
e
```

Aljabar linier

EMT memiliki banyak sekali fungsi untuk menyelesaikan masalah sistem linier, sistem sparse, atau regresi. Untuk sistem linier $Ax=b$, Anda dapat menggunakan algoritma Gauss, matriks invers, atau linear fit. Operator $A\b$ menggunakan versi algoritma Gauss.

```
>A=[1,2;3,4]; b=[5;6]; A\b
```

```
-4
4.5
```

Contoh lain, kita membuat matriks berukuran 200x200 dan jumlah baris-barisnya. Kemudian kita selesaikan $Ax=b$ 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 mempunyai solusi, kecocokan linier meminimalkan norm kesalahan $Ax-b$.

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

1	2	3
4	5	6
7	8	9

Determinan matriks ini adalah 0.

```
>det (A)
```

0

Matriks Simbolik

Maxima memiliki matriks simbolik. Tentu saja Maxima dapat digunakan untuk permasalahan aljabar linier sederhana seperti itu. Kita dapat mendefinisikan matriks untuk Euler dan Maxima dengan &:=, lalu menggunakannya dalam ekspresi simbolik. Bentuk [...] yang 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 (%)
```

$$(a - 1)^2 (a + 2)$$

```
>$&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)
```

$$\left[x = \frac{3 - \sqrt{4ab + 1}}{2}, x = \frac{\sqrt{4ab + 1} + 3}{2} \right]$$

$$\left[x = \frac{3 - \sqrt{4ab + 1}}{2}, x = \frac{\sqrt{4ab + 1} + 3}{2} \right]$$

Nilai eigen juga dapat dihitung secara otomatis. Hasilnya adalah sebuah vektor dengan dua vektor nilai eigen dan multiplisitas.

```
> $&eigenvalues([a, 1; 1, a])
```

$$[[a - 1, a + 1], [1, 1]]$$

Untuk mengekstrak vektor eigen tertentu memerlukan pengindeksan yang teliti.

```
> $&eigenvectors([a, 1; 1, a]), &%[2] [1] [1]
```

$$[[[a - 1, a + 1], [1, 1]], [[[1, -1]], [[1, 1]]]]$$

$$[1, \quad - 1]$$

Matriks simbolik dapat dievaluasi dalam Euler secara numerik sama seperti ekspresi simbolik lainnya.

```
> A(a=4, b=5)
```

$$\begin{matrix} 1 & 4 \\ 5 & 2 \end{matrix}$$

Dalam ekspresi simbolik, gunakan "with".

```
> $&A with [a=4, b=5]
```

$$\begin{pmatrix} 1 & 4 \\ 5 & 2 \end{pmatrix}$$

Akses ke deretan matriks simbolik berfungsi sama seperti matriks numerik.

```
> $&A[1]
```

$$[1, a]$$

Ekspresi simbolis dapat berisi tugas. Dan itu mengubah matriks A.

```
> &A[1, 1]:=t+1; $&A
```

$$\begin{pmatrix} t + 1 & a \\ b & 2 \end{pmatrix}$$

Ada fungsi simbolik di 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
```

$$\left[\frac{1}{j+1}, \frac{1}{j+2}, \frac{1}{j+3} \right]$$

```
>B &:= [1, 2; 3, 4]; $B, $&invert(B)
```

$$\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

$$\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

Hasilnya dapat dievaluasi secara numerik dalam Euler. Untuk informasi lebih lanjut tentang Maxima, lihat pengenalan Maxima.

```
>$&invert(B)()
```

$$\begin{matrix} -2 & 1 \\ 1.5 & -0.5 \end{matrix}$$

Euler juga memiliki fungsi kuat xinv(), yang melakukan 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 menggunakanya di sini.

```
>longest B.xinv(B)
```

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

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)
```

$$\left[\left[\frac{15 - 3\sqrt{33}}{2}, \frac{3\sqrt{33} + 15}{2}, 0 \right], [1, 1, 1] \right]$$

Nilai Numerik dalam Ekspresi simbolik

Ekspresi simbolis hanyalah string yang berisi 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 simbolik, pendekatan pecahan untuk real akan digunakan.

```
>$&A
```

$$\begin{pmatrix} 1 & \frac{1146408}{364913} \\ 4 & 5 \end{pmatrix}$$

Untuk menghindari hal ini, ada fungsi "m xmset(variabel)".

```
>m xmset (A) ; $&A
```

$$\begin{pmatrix} 1 & 3.141592653589793 \\ 4 & 5 \end{pmatrix}$$

Maxima juga dapat menghitung dengan bilangan floating point, bahkan dengan bilangan mengambang besar dengan 32 digit. Namun evaluasinya jauh lebih lambat.

```
>$&bfloat(sqrt(2)), $&float(sqrt(2))
```

```
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 menggunakan "@var".

Perhatikan bahwa ini hanya diperlukan, jika variabel telah didefinisikan 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 sebesar 5.000 (katakanlah dalam dolar).

```
>K=5000
```

5000

Sekarang kami mengasumsikan tingkat bunga 3% per tahun. Mari kita tambahkan satu tarif sederhana dan hitung hasilnya.

```
>K*1.03
```

5150

Euler juga akan memahami sintaks berikut.

```
>K+K*3%
```

5150

Namun lebih mudah menggunakan faktor tersebut

```
>q=1+3%, K*q
```

1.03

5150

Selama 10 tahun, kita cukup mengalikan faktor-faktornya dan mendapatkan nilai akhir dengan tingkat bunga majemuk.

```
>K*q^10
```

6719.58189672

Untuk keperluan kita, kita dapat mengatur formatnya menjadi 2 digit setelah titik desimal.

```
>format(12,2); K*q^10
```

6719.58

Mari kita cetak yang dibulatkan menjadi 2 digit dalam satu kalimat lengkap.

```
>"Starting from " + K + "$ you get " + round(K*q^10,2) + "$."
```

Starting from 5000\$ you get 6719.58\$.

Bagaimana jika kita ingin mengetahui hasil antara dari tahun 1 sampai tahun ke 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menulis satu perulangan, tetapi cukup masuk

```
>K*q^(0:10)
```

Real 1 x 11 matrix

5000.00 5150.00 5304.50 5463.64 ...

Bagaimana keajaiban ini terjadi? 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 pada vektor elemen demi 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 sampai q^{10} . Ini dikalikan dengan K , dan kita mendapatkan vektor nilainya.

```
>VK=K*q^(0:10);
```

Tentu saja, cara realistik untuk menghitung suku bunga ini adalah dengan membulatkan ke sen terdekat setiap tahunnya. 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 mengulanginya selama bertahun-tahun. Euler memberikan banyak solusi untuk ini.

Cara termudah adalah fungsi iterate, yang mengulangi 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 mudah, 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 vektor tertentu, 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 vektor indeks. Ingatlah 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.

Menyelesaikan Persamaan

Sekarang kita mengambil fungsi yang lebih maju, yang menambahkan tingkat uang tertentu setiap tahunnya.

```
>function onepay (K) := K*q+R
```

Kita tidak perlu menentukan q atau R untuk definisi fungsi. Hanya jika kita menjalankan perintah, kita harus mendefinisikan 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 uangnya berkurang. Jelasnya, jika kita hanya mendapat bunga sebesar 150 pada tahun pertama, namun menghapus 200, kita kehilangan uang setiap tahunnya.

Bagaimana kita dapat menentukan berapa tahun uang tersebut akan bertahan? Kita harus menulis satu lingkaran untuk ini. Cara termudah adalah dengan melakukan iterasi cukup panjang.

```
>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
```

Alasannya adalah bukan nol (VKR<0) mengembalikan vektor indeks i, dengan VKR[i]<0, dan min menghitung indeks minimal.

Karena vektor selalu dimulai dengan indeks 1, maka jawabannya adalah 47 tahun.

Fungsi iterate() memiliki satu trik lagi. Ini dapat mengambil kondisi akhir sebagai argumen. Kemudian 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 mengetahui bahwa nilainya adalah 0 setelah 50 tahun. Berapa tingkat bunganya?

Ini adalah pertanyaan yang hanya bisa dijawab secara numerik. Di bawah ini, kita akan mendapatkan rumus yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada rumus yang mudah untuk menentukan tingkat suku bunga. Namun untuk saat ini, kami menargetkan solusi numerik.

Langkah pertama adalah mendefinisikan 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

$$x_{n+1} = x_n \cdot \left(1 + \frac{P}{100}\right) + R$$

Namun kami tidak lagi 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 kita ambil 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

Rutinitas penyelesaian menyelesaikan ekspresi=0 untuk variabel x. Jawabannya adalah 3,15% per tahun. Kami mengambil nilai awal 3% untuk algoritma. Fungsi solve() selalu membutuhkan nilai awal.

Kita dapat menggunakan fungsi yang sama untuk menyelesaikan 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 bilangan bulat.

Solusi Simbolis Masalah Suku Bunga

Kita dapat menggunakan bagian simbolis dari Euler untuk mempelajari masalahnya. Pertama kita mendefinisikan fungsi onepay() kita secara simbolis.

```
>function op(K) &= K*q+R; $&op(K)
```

$$R + q K$$

Sekarang kita dapat mengiterasinya.

```
>$&op(op(op(op(K)))) , $&expand(%)
```

$$q^3 R + q^2 R + q R + R + q^4 K$$

Kami melihat sebuah pola. Setelah n periode yang kita miliki

$$K_n = q^n K + R(1 + q + \dots + q^{n-1}) = q^n K + \frac{q^n - 1}{q - 1} R$$

Rumusnya adalah rumus jumlah geometri yang diketahui Maxima.

```
>&sum(q^k, k, 0, n-1); $& % = ev(%, simpsum)
```

$$\sum_{k=0}^{n-1} q^k = \frac{q^n - 1}{q - 1}$$

Ini agak rumit. Jumlahnya dievaluasi dengan tanda "simpsum" untuk menguranginya menjadi hasil bagi. Mari kita membuat fungsi untuk ini.

```
>function fs(K, R, P, n) &= (1+P/100)^n*K + ((1+P/100)^n-1)/(P/100)*R; $&fs(K, R, P, n)
```

$$\frac{100 \left(\left(\frac{P}{100} + 1\right)^n - 1\right) R}{P} + K \left(\frac{P}{100} + 1\right)^n$$

Fungsinya sama dengan fungsi f kita sebelumnya. Tapi ini lebih efektif.

```
>longest f(5000,-200,3,47), longest fs(5000,-200,3,47)
```

```
-19.82504734650985
-19.82504734652684
```

Sekarang kita dapat menggunakananya untuk menanyakan waktu n. Kapan modal kita habis? Perkiraan awal kami adalah 30 tahun.

```
>solve("fs(5000,-330,3,x)",30)
```

20.51

Jawaban ini mengatakan akan menjadi negatif setelah 21 tahun.

Kita juga dapat menggunakan sisi simbolis Euler untuk menghitung rumus pembayaran.

Asumsikan kita mendapatkan pinjaman sebesar K, dan membayar n pembayaran sebesar R (dimulai setelah tahun pertama) meninggalkan sisa hutang sebesar Kn (pada saat pembayaran terakhir). Rumusnya jelas

```
>equ &= fs(K,R,P,n)=Kn; $&equ
```

$$\frac{100 \left(\left(\frac{P}{100} + 1 \right)^n - 1 \right) R}{P} + K \left(\frac{P}{100} + 1 \right)^n = Kn$$

Biasanya rumus ini diberikan dalam bentuk

$$i = \frac{P}{100}$$

```
>equ &= (equ with P=100*i); $&equ
```

$$\frac{((i+1)^n - 1) R}{i} + (i+1)^n K = Kn$$

Kita dapat menyelesaikan nilai R secara simbolis.

```
>$&solve(equ,R)
```

$$\left[R = \frac{i Kn - i (i+1)^n K}{(i+1)^n - 1} \right]$$

Seperti yang Anda lihat dari rumusnya, fungsi ini mengembalikan kesalahan floating point untuk $i=0$. Euler tetap merencanakannya.

Tentu saja, kami memiliki batasan berikut.

```
>$&limit(R(5000,0,x,10),x,0)
```

$$\lim_{x \rightarrow 0} R(5000, 0, x, 10)$$

Yang jelas tanpa bunga kita harus membayar kembali 10 tarif 500.

Persamaan tersebut juga dapat diselesaikan untuk n. Akan terlihat lebih bagus jika kita menerapkan beberapa penyederhanaan padanya.

```
>fn &= solve(equ,n) | ratsimp; $&fn
```

$$n = \left\lceil \frac{\log \left(\frac{R+iKn}{R+iK} \right)}{\log(i+1)} \right\rceil$$

R.2 | Exercise Set

Tulis ekspresi yang setara tanpa negatif eksponen.

49.

$$\left(\frac{24a^{10}b^{-8}c^7}{12a^6b^{-3}c^5} \right)^{-5}$$

```
>$& ((24*a^10*b^(-8)*c^7)/(12*a^6*b^(-3)*c^5)) ^ (-5)
```

$$\frac{b^{25}}{32 a^{20} c^{10}}$$

50.

$$\left(\frac{125p^{12}q^{-14}r^{22}}{25p^8q^6r^{-15}} \right)^{-4}$$

```
>$& ((125*p^12*q^(-14)*r^22)/(25*p^8*q^6*r^(-15))) ^ (-4)
```

$$\frac{q^{80}}{625 p^{16} r^{148}}$$

90.

$$2^6 \cdot 2^{-3} \div 2^{10} \div 2^{-8}$$

```
>$& (2^6*2^(-3))/(2^10)/(2^(-8))
```

2

91.

$$\frac{4(8-6)^2 - 4 \cdot 3 + 2 \cdot 8}{3^1 + 19^0}$$

```
> $& 4*(8-6)^2-4*3+2*8/3^1+19^0
```

$$\frac{31}{3}$$

104.

$$(m^{x-b} \cdot n^{x+b})^x (m^b n^{-b})^x$$

```
> $& (m^(x-b)*n^(x+b))^x*(m^b*n^(-b))^x
```

$$\left(\frac{m^b}{n^b}\right)^x (m^{x-b} n^{x+b})^x$$

R.3 | Exercise Set

Lakukan operasi yang ditunjukkan

7.

$$(2x + 3y + z - 7) + (4x - 2y - z + 8) + (-3x + y - 2z - 4)$$

```
> $& (2*x+3*y+z-7)+(4*x-2*y-z+8)+(-3*x+y-2*z-4)
```

$$-2z + 2y + 3x - 3$$

9.

$$(3x^2 - 2x - x^3 + 2) - (5x^2 - 8x - x^3 + 4)$$

```
> $& (3*x^2-2*x-x^3+2)-(5*x^2-8*x-x^3+4)
```

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

```
> $& factor(%)
```

$$-2(x^2 - 3x + 1)$$

```
> $& solve(%, x)
```

$$\left[x = \frac{3 - \sqrt{5}}{2}, x = \frac{\sqrt{5} + 3}{2} \right]$$

11.

$$(x^4 - 3x^2 + 4x) - (3x^3 + x^2 - 5x + 3)$$

```
> $& (x^4 - 3*x^2 + 4*x) - (3*x^3 + x^2 - 5*x + 3)
```

$$x^4 - 3x^3 - 4x^2 + 9x - 3$$

```
> $& factor(%)
```

$$(x - 1) (x^3 - 2x^2 - 6x + 3)$$

```
> $& solve(%, x)
```

$$\left[x = \left(-\frac{\sqrt{3}i}{2} - \frac{1}{2} \right) \left(\frac{\sqrt{503}i}{6} + \frac{43}{54} \right)^{\frac{1}{3}} + \frac{22 \left(\frac{\sqrt{3}i}{2} - \frac{1}{2} \right)}{9 \left(\frac{\sqrt{503}i}{6} + \frac{43}{54} \right)^{\frac{1}{3}}} + \frac{2}{3}, x = \left(\frac{\sqrt{3}i}{2} - \frac{1}{2} \right) \left(\frac{\sqrt{503}i}{6} + \frac{43}{54} \right)^{\frac{1}{3}} + \frac{22 \left(-\frac{\sqrt{3}i}{2} - \frac{1}{2} \right)}{9 \left(\frac{\sqrt{503}i}{6} + \frac{43}{54} \right)^{\frac{1}{3}}} + \frac{2}{3} \right]$$

13.

$$(3a^2)(-7a^4)$$

```
> $& (3*a^2) * (-7*a^4)
```

$$-21a^6$$

21.

$$(x + 6)(x + 3)$$

```
> $& expand( (x+6) * (x+3) )
```

$$x^2 + 9x + 18$$

```
> $& solve(%, x)
```

$$[x = -3, x = -6]$$

R.4 | Exercise Set

Faktorkan soal berikut.

24.

$$y^2 + 12y + 27$$

```
> $& factor(y^2+12*y+27)
```

$$(y + 3) (y + 9)$$

```
> $& solve(%, y)
```

$$[y = -9, y = -3]$$

48.

$$m^2 - 4$$

```
> $& factor(m^2-4)
```

$$(m - 2) (m + 2)$$

```
> $& solve(%, m)
```

$$[m = -2, m = 2]$$

58.

$$y^2 - 6y + 9$$

```
> $&factor(y^2 - 6*y + 9)
```

$$(y - 3)^2$$

```
> $&solve(%, y)
```

$$[y = 3]$$

74.

$$250z^4 - 2z$$

```
> $&factor(250*z^4 - 2*z)
```

$$2z(5z - 1)(25z^2 + 5z + 1)$$

```
> $&solve(%, z)
```

$$\left[z = \frac{-\sqrt{3}i - 1}{10}, z = \frac{\sqrt{3}i - 1}{10}, z = \frac{1}{5}, z = 0 \right]$$

100.

$$x^2y^2 - 16xy + 64$$

```
> $&factor(x^2*y^2 - 16*x*y + 64)
```

$$(xy - 8)^2$$

```
> $&solve(%, x)
```

$$\left[x = \frac{8}{y} \right]$$

R.5 | Exercise Set

Selesaikan soal berikut.

31.

$$7(3x + 6) = 11 - (x + 2)$$

```
> $& ratsimp (7*(3*x+6)=11-(x+2))
```

$$21x + 42 = 9 - x$$

```
> $& solve (%)
```

$$\left[x = -\frac{3}{2} \right]$$

41.

$$x^2 + 100 = 200x$$

```
> $& ratsimp (x^2+100=200*x)
```

$$x^2 + 100 = 200x$$

```
> $& solve (%)
```

$$\left[x = 100 - 30\sqrt{11}, x = 30\sqrt{11} + 100 \right]$$

51.

$$14 = x(x - 5)$$

```
> $& expand (14=x*(x-5))
```

$$14 = x^2 - 5x$$

```
> $& solve (%)
```

$$[x = 7, x = -2]$$

61.

$$A = \frac{1}{2}bh, b$$

```
> $& solve (A=1/2*b*h, b)
```

$$\left[b = \frac{2A}{h} \right]$$

71.

$$Ax + By = C, A$$

```
> $& solve (A*x+B*y=C, A)
```

$$\left[A = \frac{C - yB}{x} \right]$$

R.6 | Exercise Set

Sederhanakan soal berikut.

10.

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

```
> $& (x^2+2*x-3) / (x^2-9), $&ratsimp (%)
```

$$\frac{x - 1}{x - 3}$$

$$\frac{x - 1}{x - 3}$$

12.

$$\frac{y^5 - 5y^4 + 4y^3}{y^3 - 6y^2 + 8y}$$

```
> $& (y^5-5*y^4+4*y^3) / (y^3-6*y^2+8*y), $&ratsimp (%)
```

$$\frac{y^3 - y^2}{y - 2}$$

$$\frac{y^3 - y^2}{y - 2}$$

14.

$$\frac{2x^2 - 20x + 50}{10x^2 - 30x - 100}$$

```
> $& (2*x^2-20*x+50) / (10*x^2-30*x-100), $&ratsimp(%)
```

$$\frac{x - 5}{5x + 10}$$

$$\frac{x - 5}{5x + 10}$$

18.

$$\frac{x^2 - y^2}{(x - y)^2} \cdot \frac{1}{x + y}$$

```
> $& ( (x^2 - y^2) / ((x - y)^2) * ((1) / (x + y))
```

$$\frac{x^2 - y^2}{(x - y)^2 (y + x)}$$

```
> $& expand(%)
```

$$\frac{x^2}{y^3 - xy^2 - x^2y + x^3} - \frac{y^2}{y^3 - xy^2 - x^2y + x^3}$$

```
> $&ratsimp(%)
```

$$-\frac{1}{y - x}$$

24.

$$\frac{a^2 - b^2}{x - y} \div \frac{a + b}{x - y}$$

```
> $& ( (a^2 - b^2) / (x - y) ) / ( (a + b) / (x - y) )
```

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

```
> $&expand(%)
```

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

```
> $&ratsimp(%)
```

$$a - b$$

2.3 | Exercise Set

Jika diketahui

$$f(x) = 3x + 1$$

$$g(x) = x^2 - 2x - 6$$

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

Tentukan masing-masing persamaan berikut.

1.

$$(f \circ g)(-1)$$

```
> function f(x) &= 3*x+1
```

$$3 \ x \ + \ 1$$

```
> function g(x) &= x^2-2*x-6
```

$$x^2 - 2 \ x - 6$$

```
> $&f(g(x))
```

$$3 (x^2 - 2x - 6) + 1$$

```
>${&expand(%)}
```

$$3x^2 - 6x - 17$$

```
>${&f(g(-1))}
```

$$-8$$

3.

$$(h \circ f)(1)$$

```
>function f(x) &=3*x+1
```

$$3x + 1$$

```
>function h(x) &=x^3
```

$$\begin{matrix} 3 \\ x \end{matrix}$$

```
>${&h(f(x))}
```

$$(3x + 1)^3$$

```
>${&expand(%)}
```

$$27x^3 + 27x^2 + 9x + 1$$

```
>${&h(f(1))}
```

$$64$$

5.

$$(g \circ f)(5)$$

```
>function f(x) &=3*x+1
```

$$3x + 1$$

```
>function g(x) &=x^2-2*x-6
```

$$x^2 - 2x - 6$$

```
>$&g(f(x))
```

$$(3x + 1)^2 - 2(3x + 1) - 6$$

```
>$&expand(%)
```

$$9x^2 - 7$$

```
>$&g(f(5))
```

$$218$$

Tentukan

$$(f \circ g)(x)$$

$$(g \circ f)(x)$$

Dari persamaan berikut.

17.

$$f(x) = x + 3, \quad g(x) = x - 3$$

```
>function f(x) &=x+3
```

$$x + 3$$

```
>function g(x) &=x-3
```

$$x - 3$$

```
>$&f(g(x))
```

$$x$$

```
>$&g(f(x))
```

$$x$$

19.

$$f(x) = x + 1, \quad g(x) = 3x^2 - 2x - 1$$

```
>function f(x) &=x+1
```

$$x + 1$$

```
>function g(x) &=3*x^2-2*x-1
```

$$3 x^2 - 2 x - 1$$

```
>$&f(g(x))
```

$$3 x^2 - 2 x$$

```
> $&g(f(x))
```

$$3(x+1)^2 - 2(x+1) - 1$$

```
> $&expand(%)
```

$$3x^2 + 4x$$

3.1 | Exercise Set

Sederhanakan persamaan berikut.

12.

$$(-6 - 5i) + (9 + 2i)$$

```
> $& (-6-5*i)+(9+2*i); $&ratsimp(%)
```

$$3 - 3i$$

14.

$$(7 - 2i) + (4 - 5i)$$

```
> $& (7-2*i)+(4-5*i); $&ratsimp(%)
```

$$11 - 7i$$

22.

$$(-3 - 4i) - (8 - i)$$

```
> $& (-3-4*i)-(8-i); $&ratsimp(%)
```

$$-3i - 11$$

26.

$$(8 - 3i) - (9 - i)$$

```
> $&(8-3*i)-(9-i); $ratsimp(%)
```

$$-2i - 1$$

30.

$$(10 - 4i) - (8 + 2i)$$

```
> $&(10-4*i)-(8+2*i); $ratsimp(%)
```

$$2 - 6i$$

3.4 | Exercise Set

Selesaikan soal berikut.

1.

$$\frac{1}{4} + \frac{1}{5} = \frac{1}{t}$$

```
> $&solve((1/4+1/5=1/t), t)
```

$$\left[t = \frac{20}{9} \right]$$

3.

$$\frac{x+2}{4} - \frac{x-1}{5} = 15$$

```
> $&solve(((x+2)/4-(x-1)/5), x)
```

$$[x = -14]$$

5.

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

```
>${&}solve((1/2+2/x=1/3+3/x),x)
```

$$[x = 6]$$

7.

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

```
>${&}solve((5/(3*x+2)=3/(2*x)),x)
```

$$[x = 6]$$

9.

$$x + \frac{6}{x} = 5$$

```
>${&}solve((x+6/x=5),x)
```

$$[x = 3, x = 2]$$

3.5 | Exercise Set

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

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

Selesaikan soal berikut.

24.

$$|x - 4| + 3 = 9$$

```
>${&} (abs(x-4)+3=9)
```

$$|x - 4| + 3 = 9$$

```
> $&fourier_elim(abs(x-4)+3=9, [x])
```

$$[x = 10] \vee [x = -2]$$

26.

$$|2x - 1| - 5 = -3$$

```
> $& (abs(2*x-1)-5=-3)
```

$$|2x - 1| - 5 = -3$$

```
> $&fourier_elim(abs(2*x-1)-5=-3, [x])
```

$$\left[x = \frac{3}{2}\right] \vee \left[x = -\frac{1}{2}\right]$$

28.

$$|5x + 4| + 2 = 5$$

```
> $& (abs(5*x+4)+2=5)
```

$$|5x + 4| + 2 = 5$$

```
> $&fourier_elim(abs(5*x+4)+2=5, [x])
```

$$\left[x = -\frac{1}{5}\right] \vee \left[x = -\frac{7}{5}\right]$$

30.

$$9 - |x - 2| = 7$$

```
> $& (9-abs(x-2)=7)
```

$$9 - |x - 2| = 7$$

```
> $&fourier_elim(9-abs(x-2)=7, [x])
```

$$[x = 4] \vee [x = 0]$$

32.

$$5 - |4x - 3| = 2$$

```
> $& (5-abs(4*x-3)=2)
```

$$5 - |4x - 3| = 2$$

```
> $&fourier_elim(5-abs(4*x-3)=2, [x])
```

$$\left[x = \frac{3}{2} \right] \vee [x = 0]$$

```
> reset();
```

BAB 3

PEKAN 5-6: PENGGUNAAN SOFTWARE EMT UNTUK PLOT 2D

article
eumat

Menggambar Grafik 2D dengan EMT

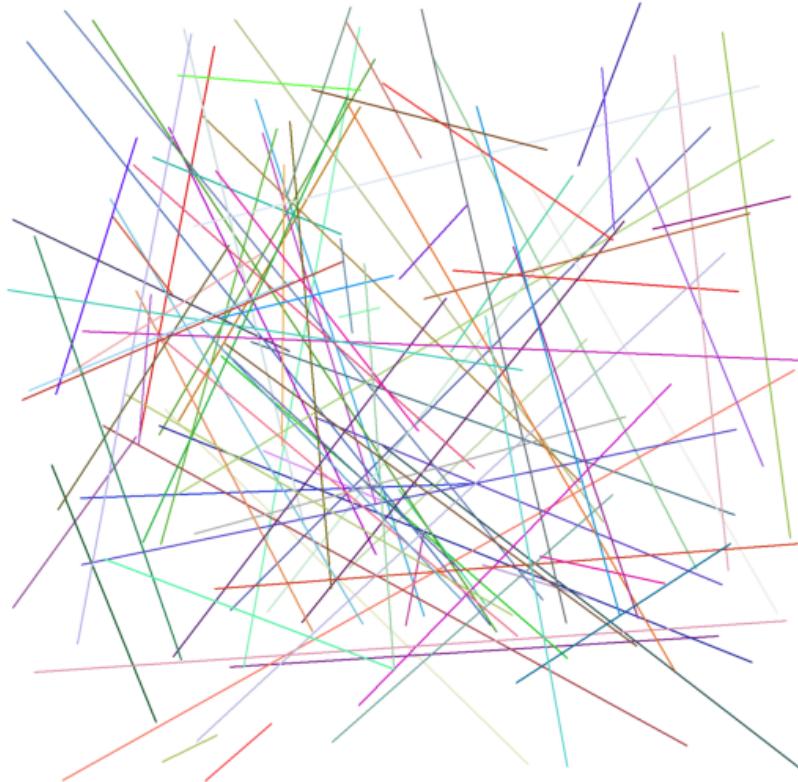
Notebook ini menjelaskan tentang cara menggambar berbagai kurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi plot2d() untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

Plot Dasar

Ada beberapa fungsi plot yang paling mendasar. Terdapat koordinat layar yang selalu memiliki rentang berkisar antara 0 hingga 1024 di setiap sumbu, tidak penting apakah layarnya berbentuk persegi atau tidak. Dan terdapat koordinat plot yang dapat diatur dengan setplot(). Pemetaan antar koordinat bergantung pada jendela plot saat ini. Misalnya, shrinkwindow() default memberikan ruang untuk label sumbu dan judul plot.

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

```
>clg; // bersihkan layar
>window(0,0,1024,1024); // gunakan seluruh jendela
>setplot(0,1,0,1); // atur koordinat plot
>hold on; // mulai mode overwrite
>n=100; X=random(n,2); Y=random(n,2); // mendapatkan poin random
>colors=rgb(random(n),random(n),random(n)); // mendapatkan warna random
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
>hold off; // mengakhiri mode overwrite
>insimg; // masukkan ke notebook
```



```
>reset;
```

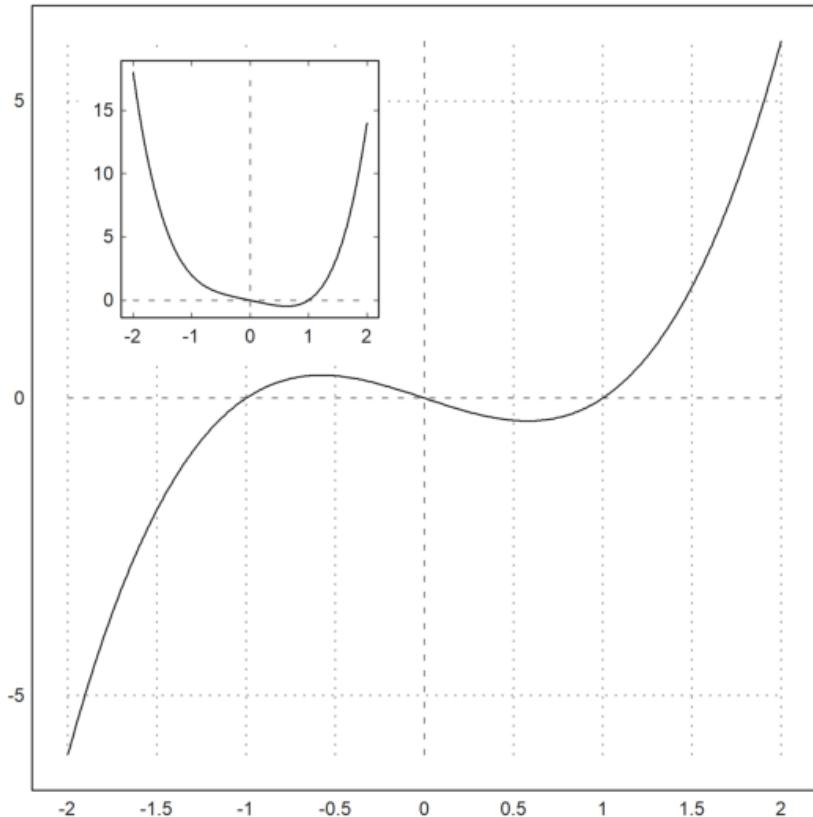
Grafik penting untuk ditahan, karena perintah plot() akan menghapus jendela plot.

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

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

Untuk contoh lain, kita menggambar plot sebagai sisipan di plot lain. Hal ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak memberikan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan beberapa margin untuk ini sesuai kebutuhan. Perhatikan bahwa kita menyimpan dan mengembalikan jendela secara penuh, dan menahan plot sekarang ini sementara kita memplot sisipan.

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



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

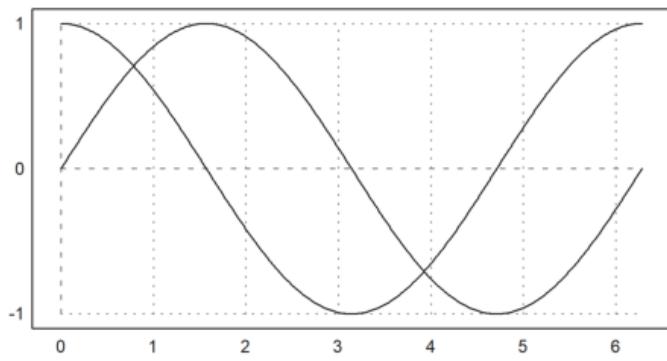
Plot dengan banyak gambar dicapai dengan cara yang sama. Ada fungsi utilitas figure() untuk ini.

Aspek Plot

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

Tapi Anda juga dapat mengubahnya untuk satu plot. Untuk ini, ukuran area plot sekarang ini diubah, dan jendela diatur sehingga label memiliki cukup ruang.

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



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

Fungsi reset() mengembalikan default plot termasuk rasio aspek.

Plot 2D di Euler

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

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

Euler dapat membuat plot 2D:

- ekspresi
- fungsi, variabel, atau kurva parameter,
- vektor dari nilai-xy,
- titik-titik pada bidang,
- kurva implisit dengan tingkatan atau tingkatan daerah,
- fungsi kompleks

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

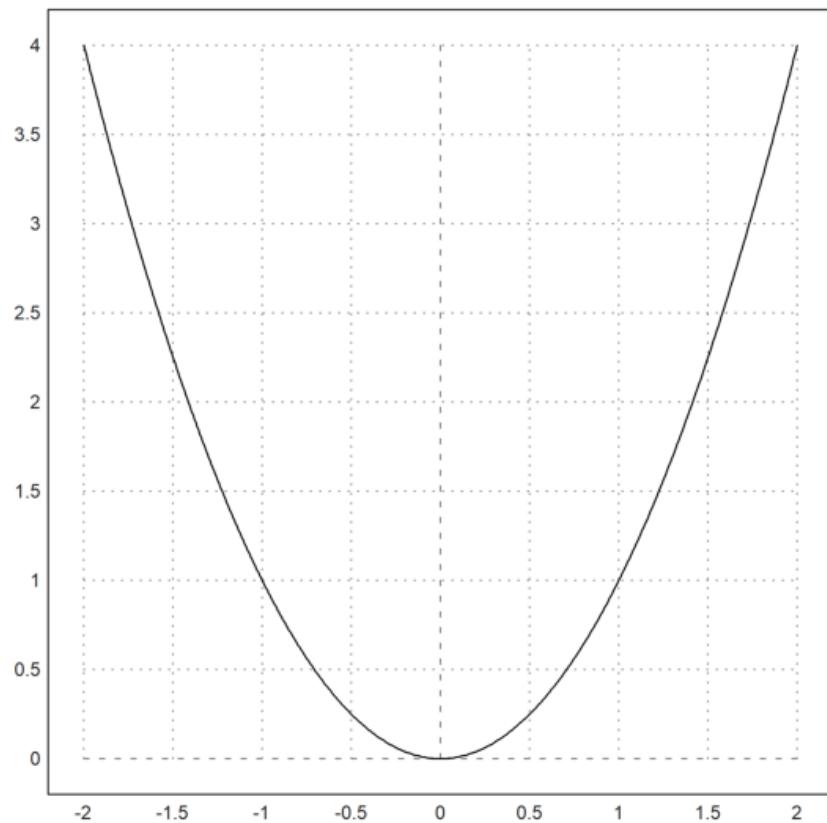
Plot Ekspresi atau Variabel

Ekspresi tunggal dalam "x" (misalnya "4*x^2") atau nama suatu fungsi (misalnya "f") menghasilkan grafik fungsi tersebut.

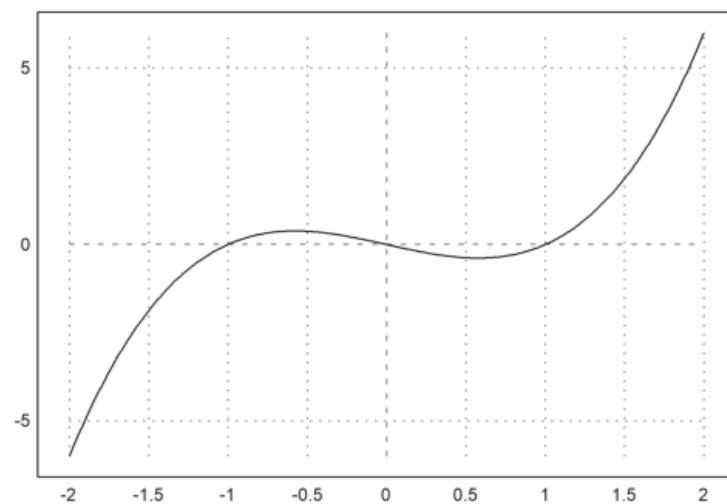
Berikut adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsinya.

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

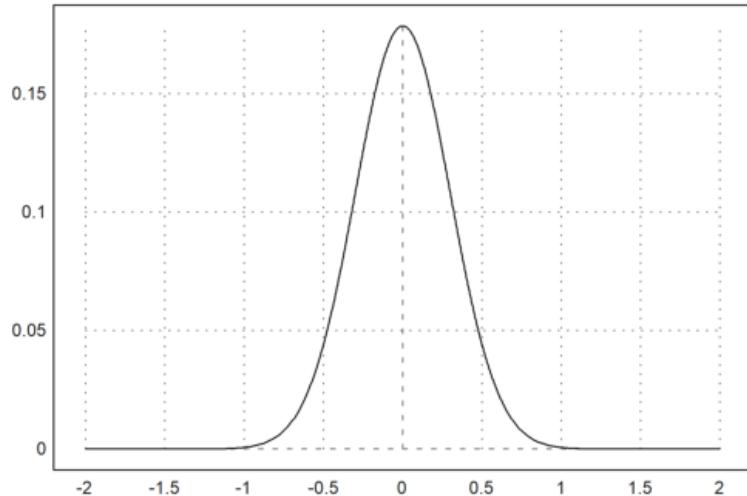
```
>plot2d("x^2":
```



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



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

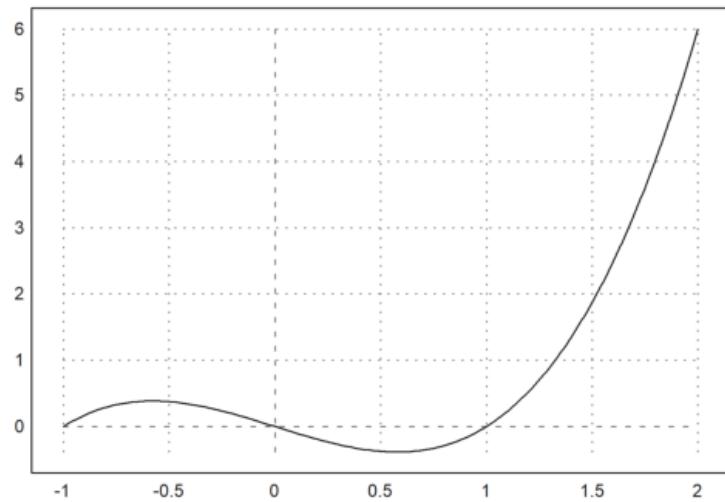


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

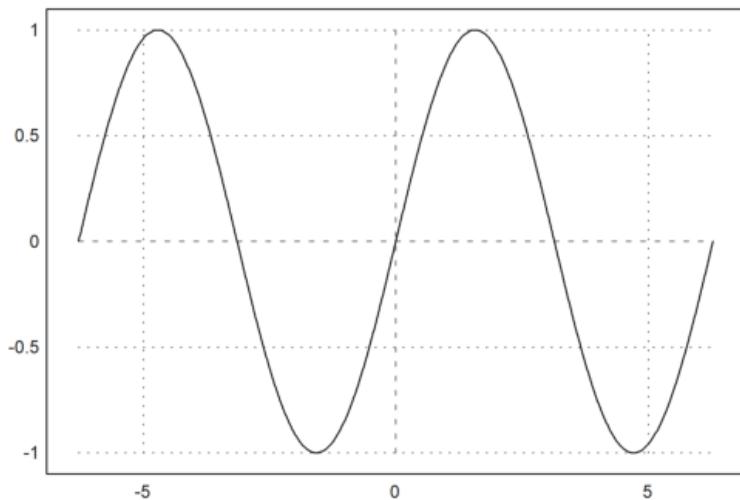
Rentang plot diatur dengan parameter yang ditetapkan sebagai berikut

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

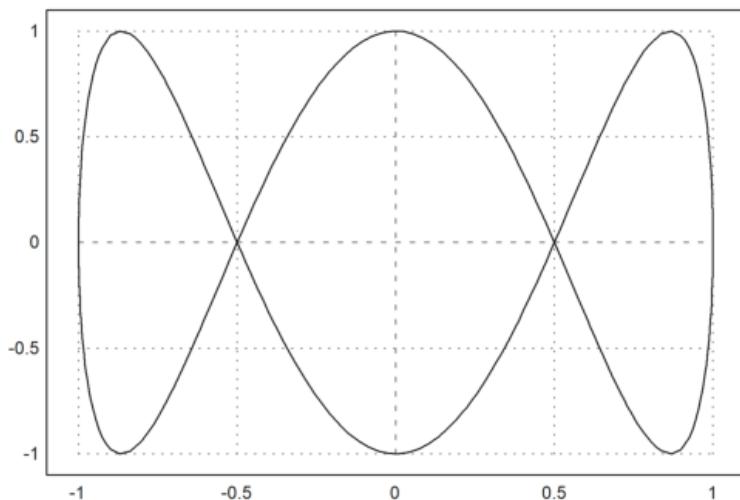
```
>plot2d("x^3-x", -1, 2):
```



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



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



Alternatif pengganti titik dua adalah perintah insimg(garis), yang menyisipkan plot yang menempati sejumlah baris teks tertentu.

Dalam pengaturan, plot dapat diatur agar muncul

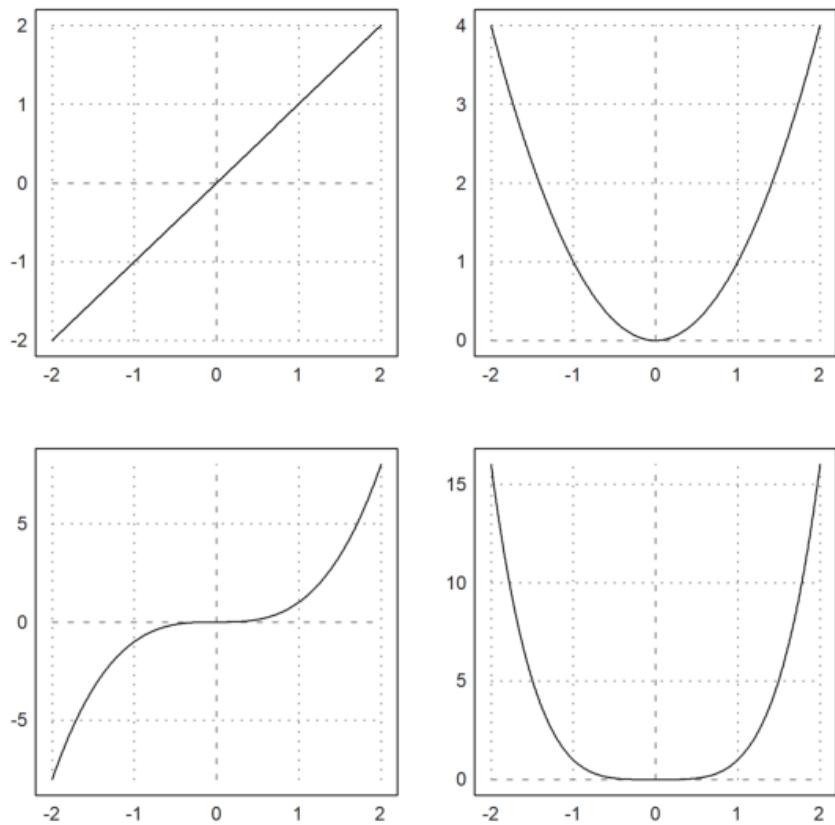
- di jendela terpisah yang dapat diubah ukurannya,
- di jendela buku catatan.

Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.

Dibeberapa kasus, tekan tombol tabulator untuk melihat plotnya, jika plot tersebut tersembunyi.

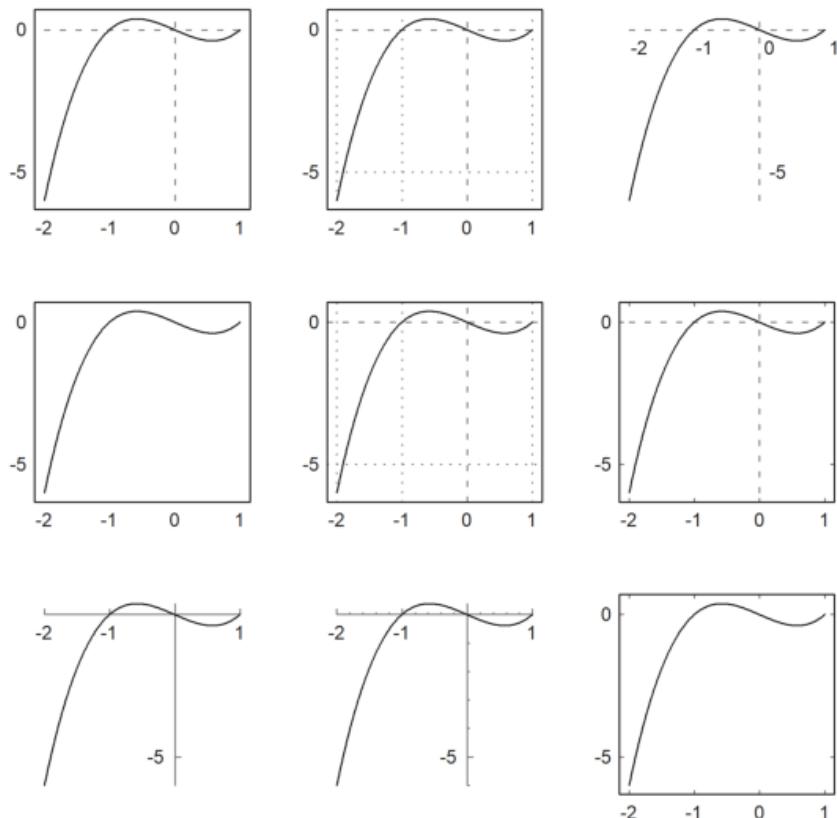
Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Contohnya, kita memplot x^1 hingga x^4 menjadi 4 bagian jendela. figure(0) mengatur ulang jendela default.

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



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

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

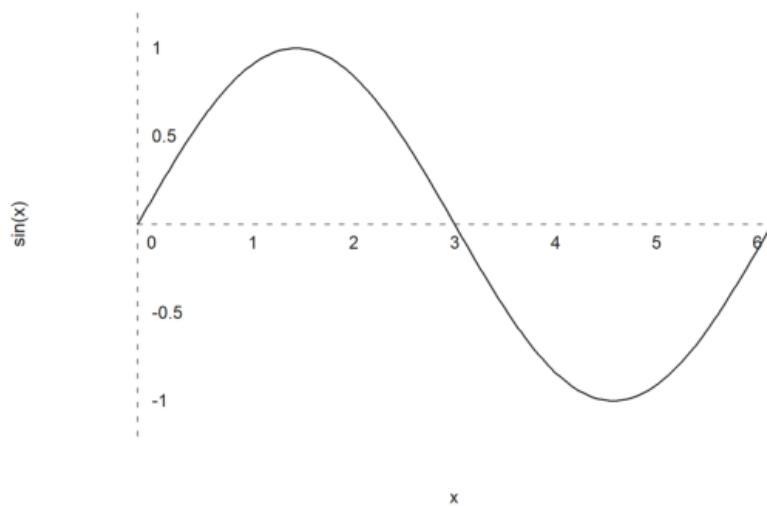


Jika argumen pada `plot2d()` adalah ekspresi yang diikuti oleh empat angka, angka-angka tersebut adalah rentang x dan y pada plot tersebut.

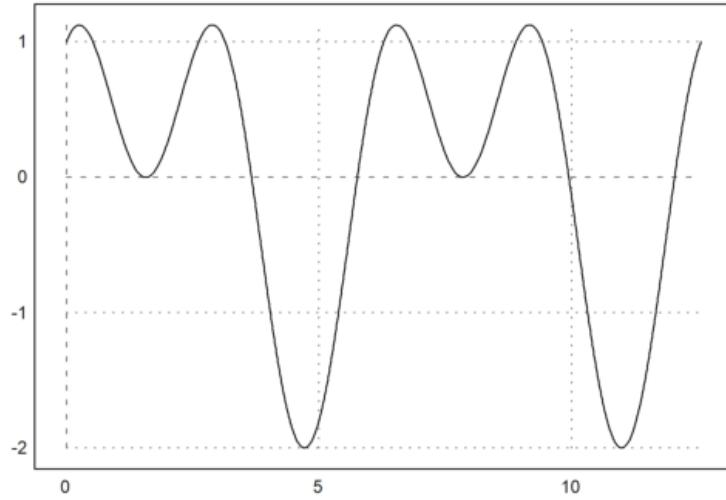
Alternatifnya, `a, b, c, d` dapat ditentukan sebagai parameter yang ditetapkan sebagai `a=...` dll.

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

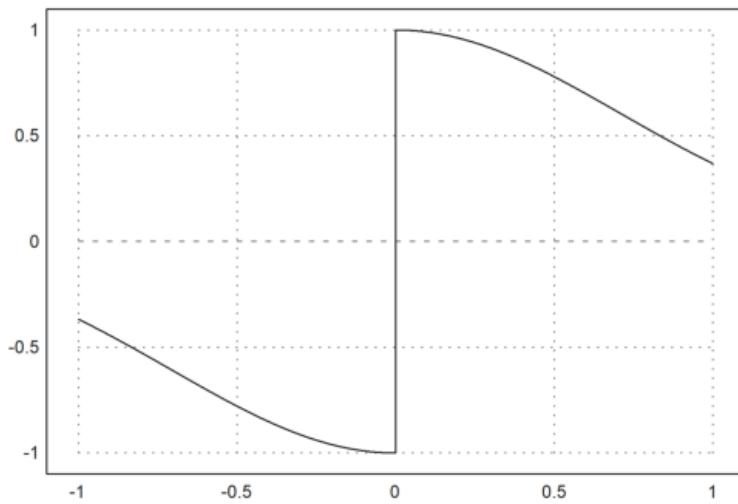


Gambar yang dihasilkan dengan memasukkan plot ke dalam jendela teks disimpan di direktori yang sama dengan notebook, secara default di subdirektori bernama "images". Subdirektori tersebut juga digunakan oleh ekspor HTML.

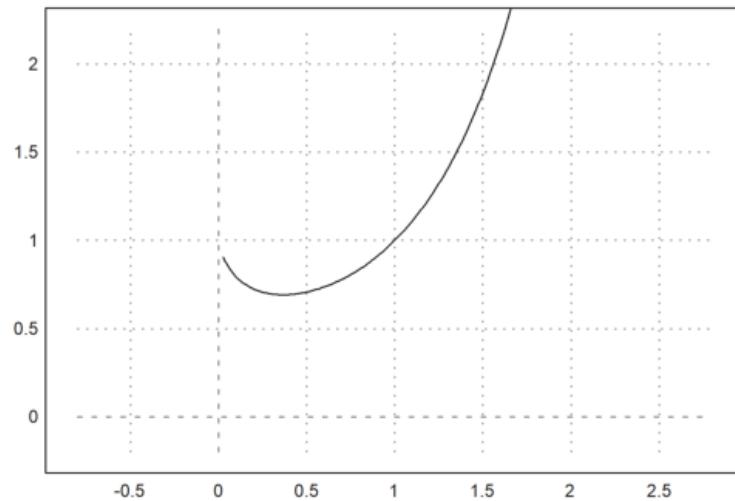
Anda bisa menandai gambar apa saja dan menyalinnya ke clipboard dengan Ctrl-C. Tentu Anda juga dapat mengekspor grafik saat ini dengan fungsi di menu File.

Fungsi atau ekspresi di plot2d dievaluasi secara adaptif. Agar lebih cepat, nonaktifkan plot adaptif dengan <adaptive dan tentukan jumlah subinterval dengan n=... Hal ini hanya diperlukan di kasus yang jarang terjadi.

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

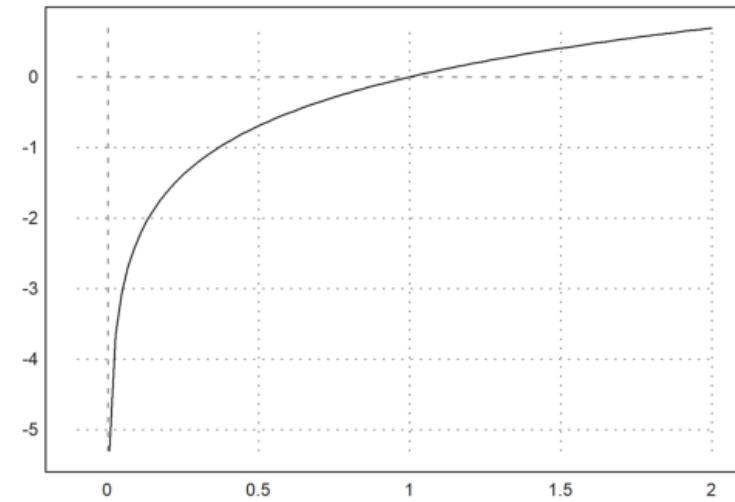


```
>plot2d("x^x", r=1.2, cx=1, cy=1):
```



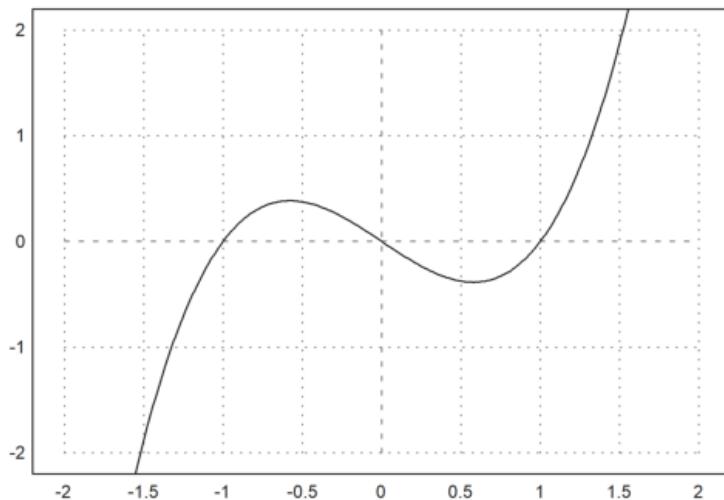
Perhatikan bahwa x^x tidak terdefinisi untuk $x <= 0$. Fungsi plot2d menangkap kesalahan ini, dan segera mulai membuat plot setelah fungsinya ditentukan. Ini berfungsi untuk semua fungsi yang mengembalikan NAN di luar jangkauan definisinya.

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

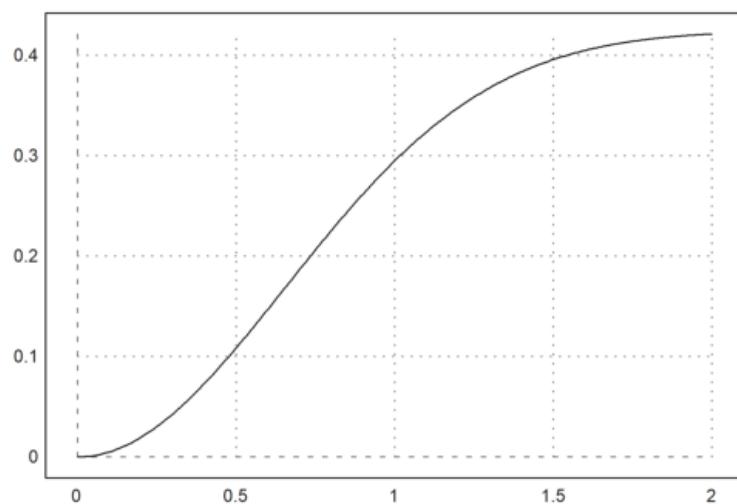


Parameter square=true (atau >square) memilih rentang y secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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

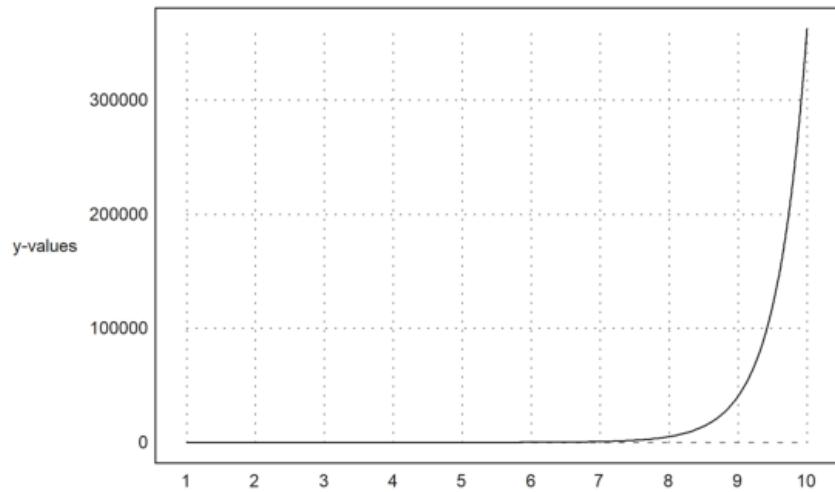


```
>plot2d(''integrate("sin(x)*exp(-x^2)", 0, x)'', 0, 2): // plot integral
```



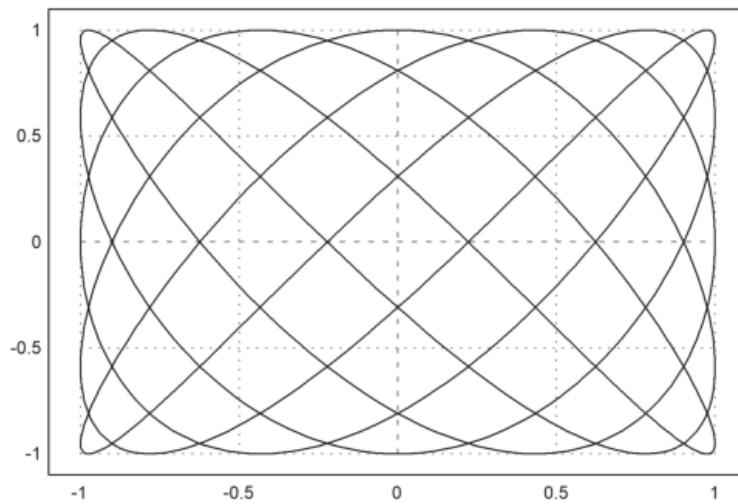
Jika Anda memerlukan lebih banyak ruang untuk label y, panggil shrinkwindow() dengan parameter yang lebih kecil, atau tetapkan nilai positif untuk "lebih kecil" di plot2d().

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

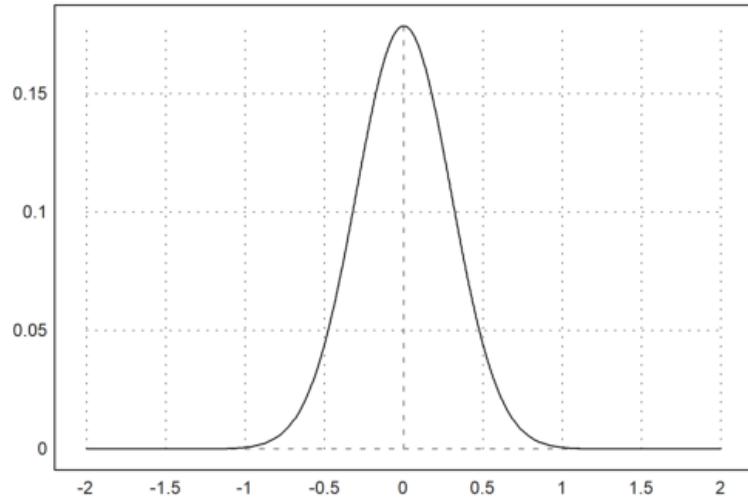


Ekspresi simbolik juga dapat digunakan karena disimpan sebagai ekspresi string sederhana.

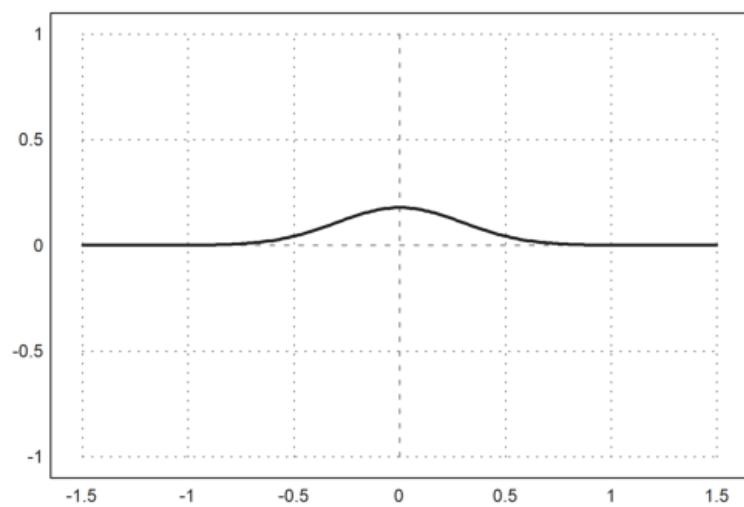
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



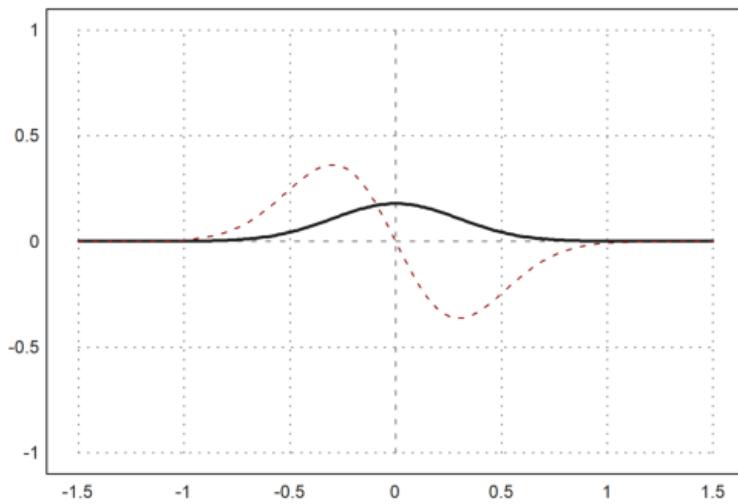
```
>a:=5.6; expr &= exp(-a*x^2)/a; // mendefinisikan ekspresi
>plot2d(expr,-2,2); // plot dari -2 sampai 2
```



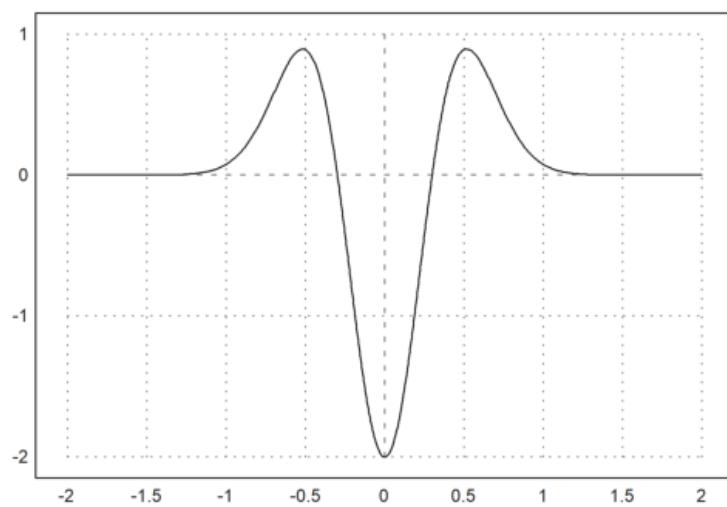
```
>plot2d(expr,r=1,thickness=2): // plot di persegi sekitar (0,0)
```



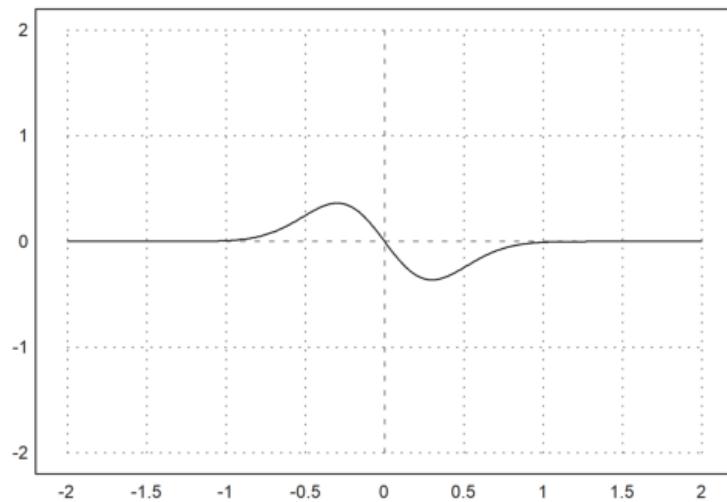
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // menambahkan plot yang lain
```



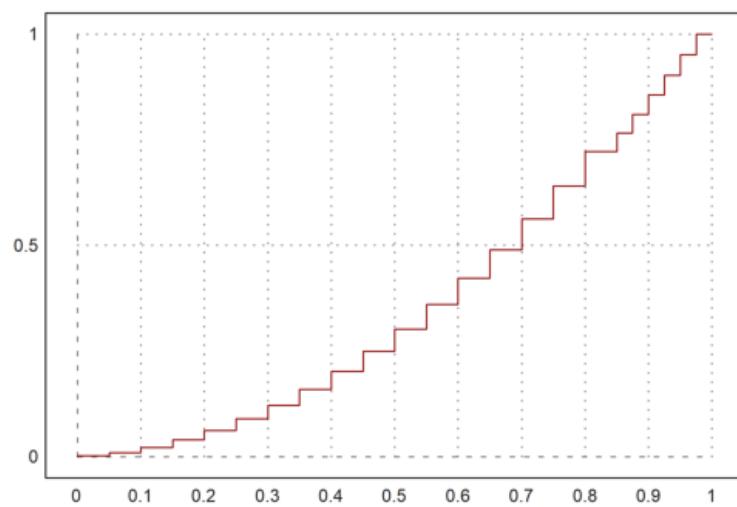
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot dalam persegi panjang
```



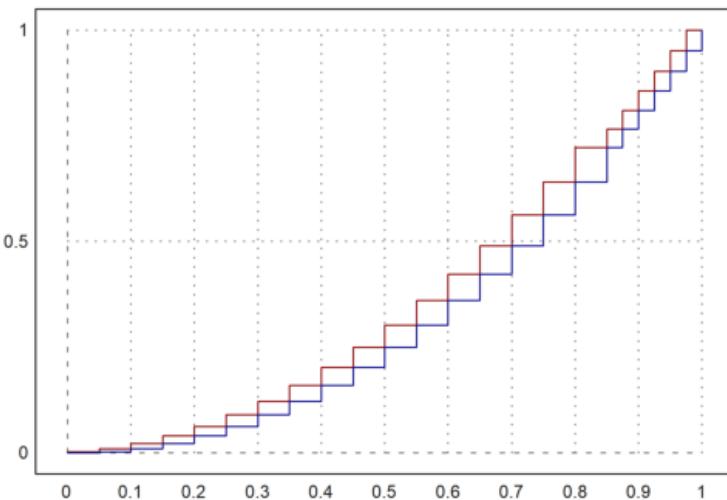
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // menjaga plot tetap persegi
```



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



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

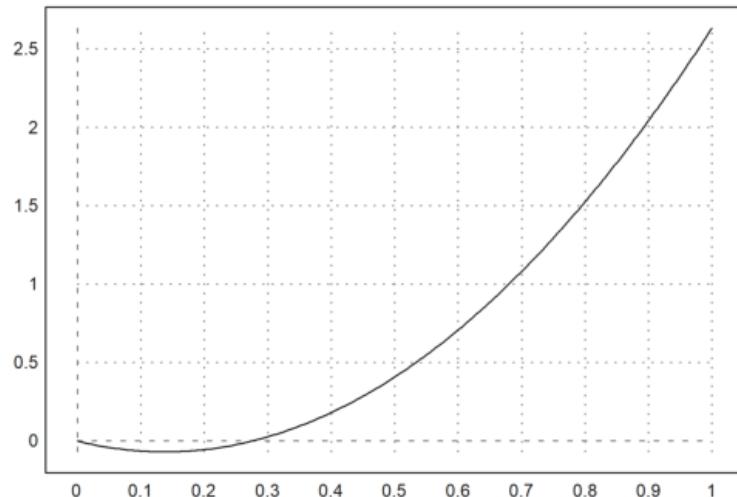


Fungsi dalam satu Parameter

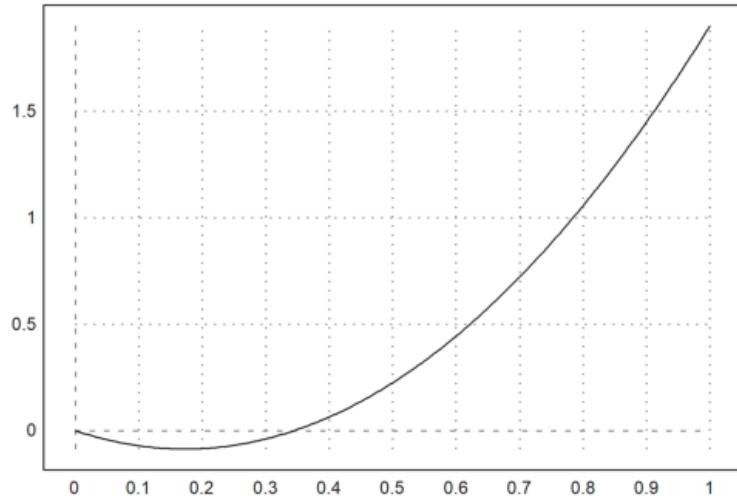
Fungsi plot yang paling penting untuk plot planar adalah `plot2d()`. Fungsi ini diimplementasikan dalam bahasa Euler di file "plot.e", yang dimuat di awal program.

Berikut beberapa contoh penggunaan suatu fungsi. Seperti biasa di EMT, fungsi yang berfungsi untuk fungsi atau ekspresi lain, Anda bisa meneruskan parameter tambahan (selain x) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan kumpulan panggilan.

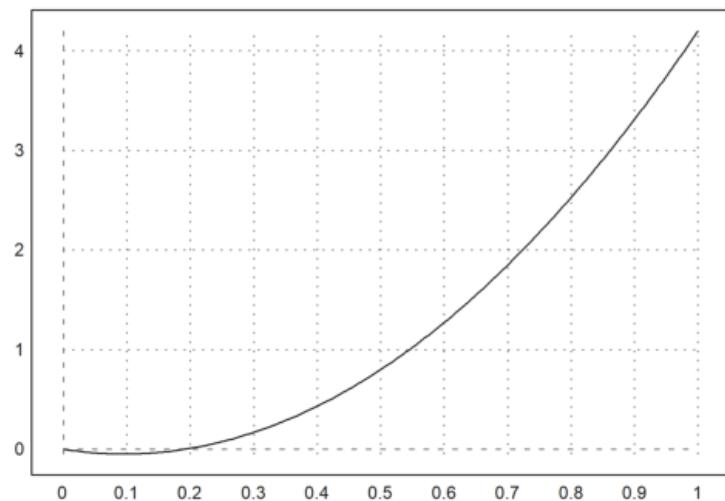
```
>function f(x,a) := x^2/a+a*x^2-x; // mendefinisikan fungsi
>a=0.3; plot2d("f",0,1;a); // plot dengan a=0.3
```



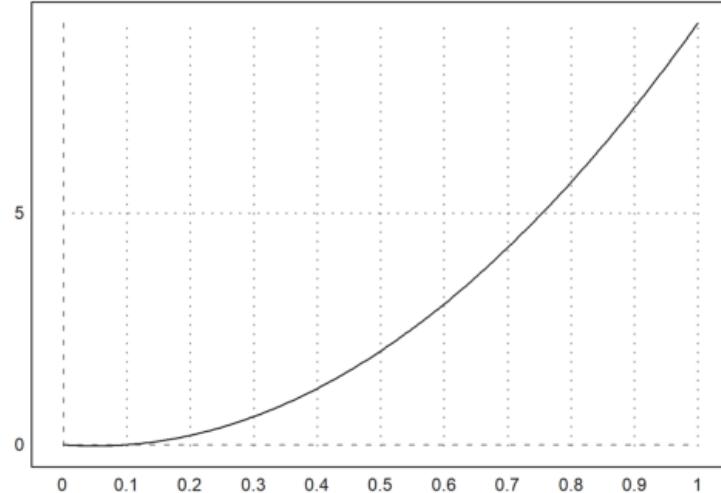
```
>plot2d("f",0,1;0.4); // plot dengan a=0.4
```



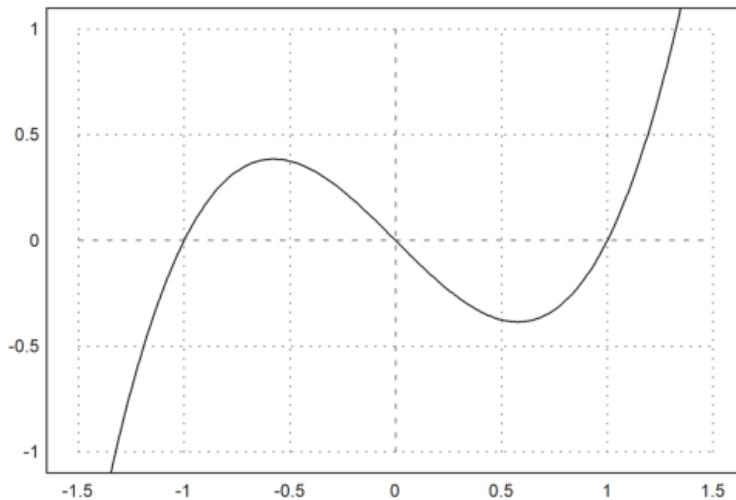
```
>plot2d({{"f",0.2}},0,1); // plot dengan a=0.2
```



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



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



Berikut ini ringkasan fungsi yang diterima

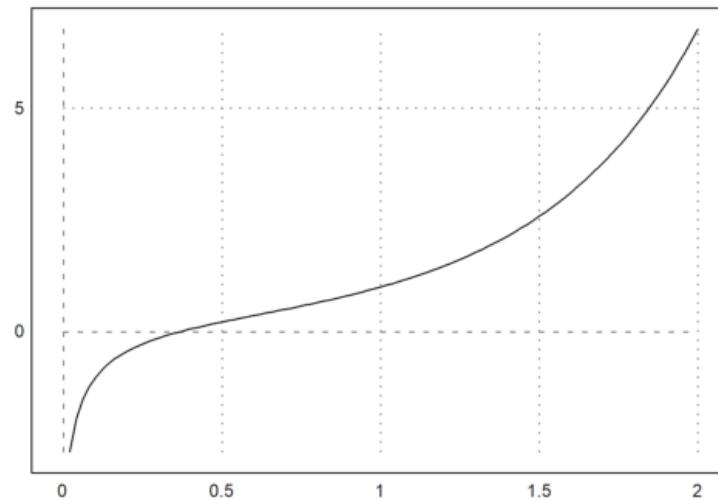
- ekspresi atau ekspresi simbolik di x
- fungsi atau fungsi simbolik dengan nama "f"
- fungsi simbolik hanya dengan nama f

Fungsi plot2d() juga menerima fungsi simbolik. Untuk fungsi simbolik, namanya saja yang berfungsi.

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

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

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

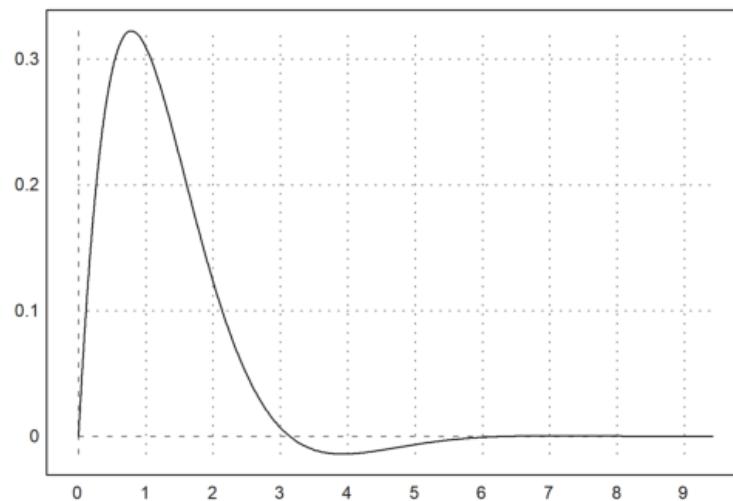


Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel cukup untuk memplotnya.

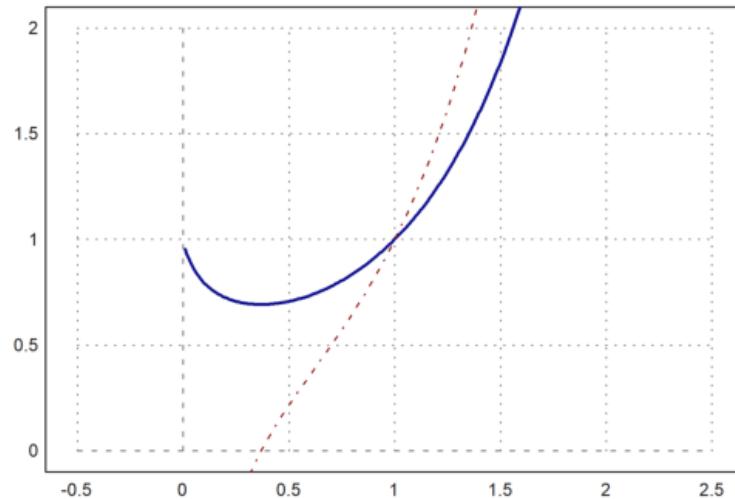
```
>expr &= sin(x)*exp(-x)
```

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

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



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



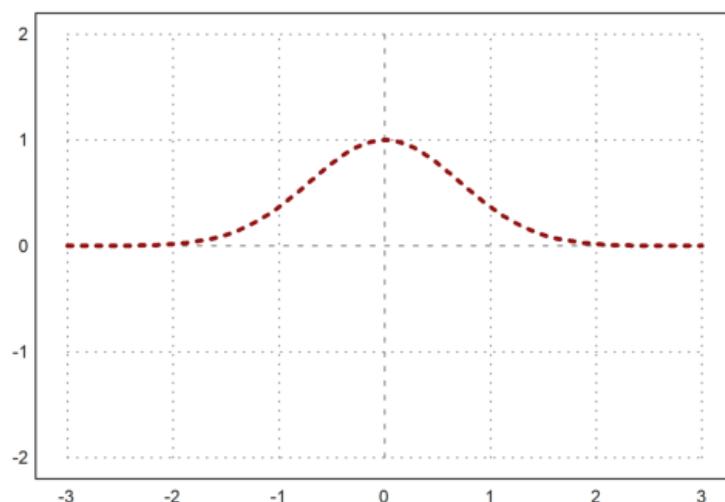
Untuk gaya garis ada berbagai pilihan.

- style="...". Pilih dari "-", "-.", ".-", ".","--"."
- color: Lihat di bawah untuk warna.
- thickness: Defaultnya adalah 1.

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

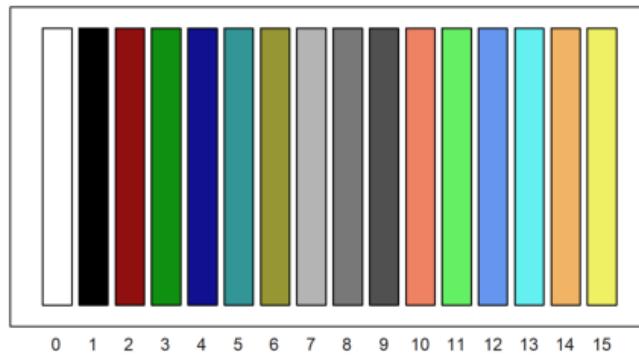
- 0.15: indeks warna default.
- color constants: white, black, red, green, blue, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, yellow
- rgb(red,green,blue): parameternya real di [0,1].

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



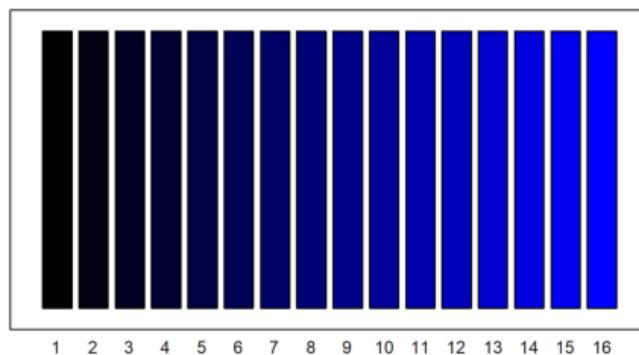
Berikut adalah tampilan warna EMT yang telah ditentukan sebelumnya.

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



Tapi Anda bisa menggunakan berbagai pilihan warna

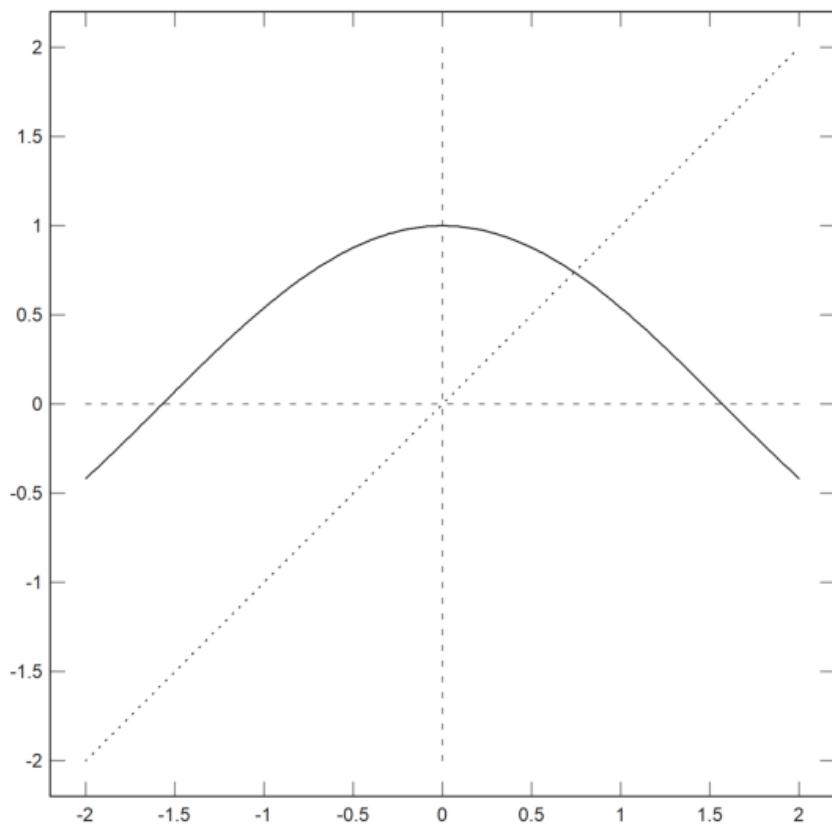
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



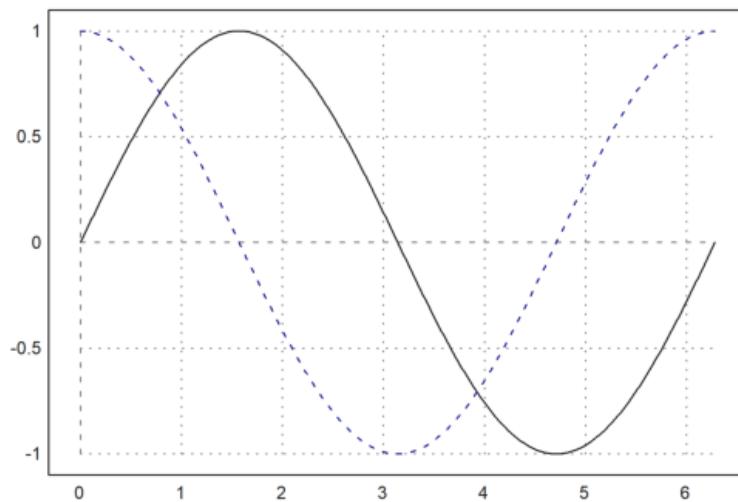
Menggambar Beberapa Kurva pada bidang yang sama

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

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

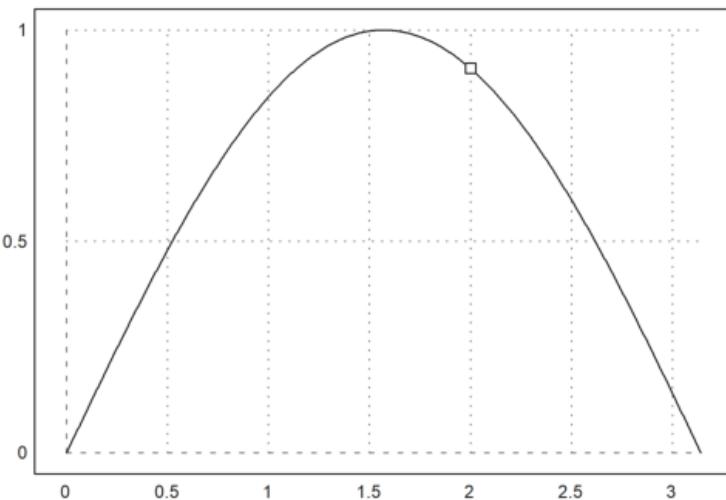


```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



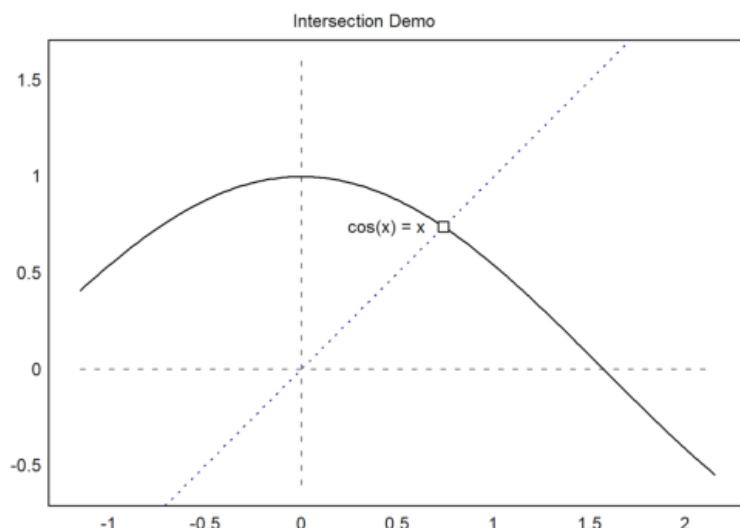
Salah satu kegunaan >add adalah untuk menambahkan titik pada kurva.

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



Kita menambahkan titik perpotongan dengan label (pada posisi "cl" untuk kiri tengah), dan masukkan hasilnya ke dalam notebook. Kita juga menambahkan judul pada 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 contoh berikut, kita memplot fungsi $\sin(x)/x$ dan ekspansi Taylor ke-8 dan ke-16. Kita menghitung perluasan tersebut menggunakan Maxima melalui ekspresi simbolik.

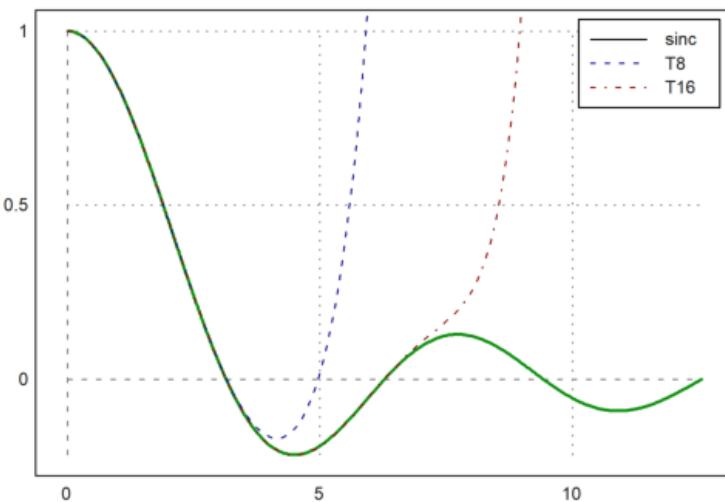
Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke `plot2d()`. Yang kedua dan ketiga memiliki kumpulan tanda `>add`, yang membuat plot menggunakan rentang sebelumnya.

Kita menambahkan kotak label yang menjelaskan fungsinya.

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

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

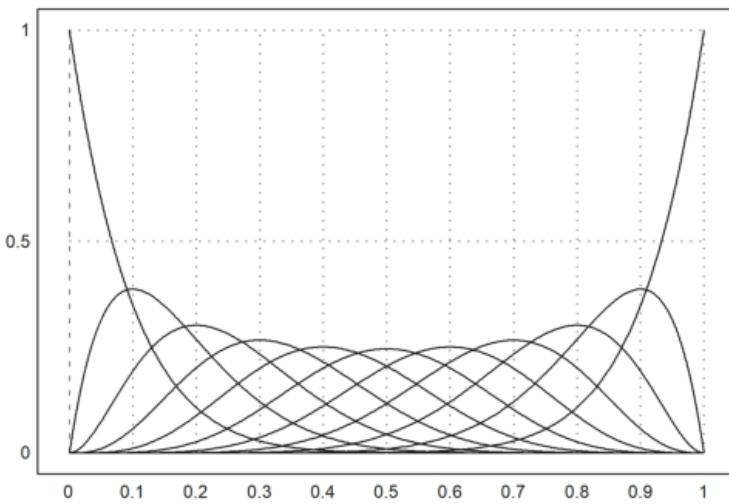
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
> plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
> plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
> labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
> colors=[black,blue,red]):
```



Pada contoh berikut, Kita menghasilkan Polinomial Bernstein.

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

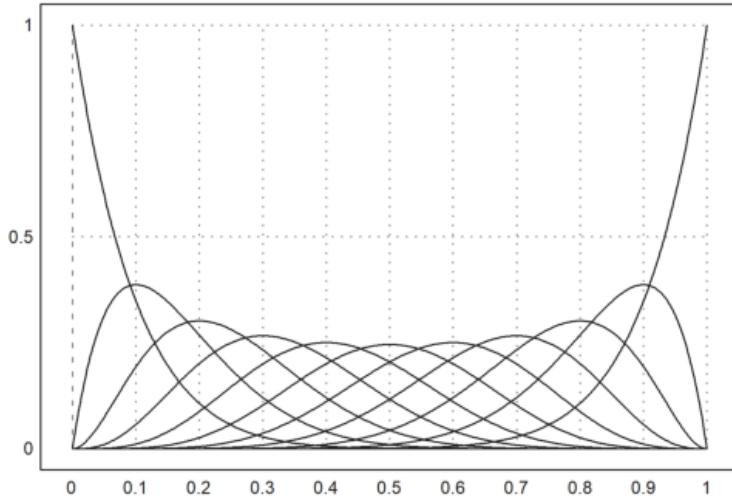
```
>plot2d("(1-x)^10",0,1); // fungsi pertama plot
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```



Cara kedua adalah dengan menggunakan pasangan matriks bernilai x dan matriks bernilai y yang berukuran sama.

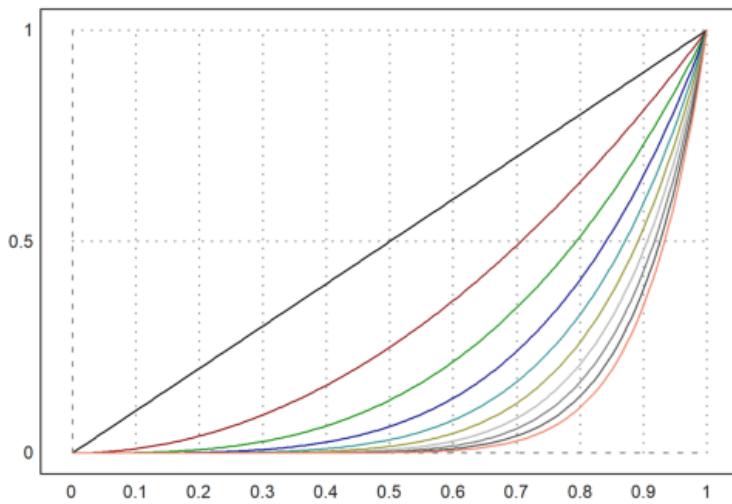
Kita menghasilkan nilai matriks dengan satu Polinomial Bernstein di setiap baris. Untuk itu, kita cukup menggunakan vektor kolom i. Lihat pendahuluan tentang bahasa matriks untuk mempelajari lebih detail.

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



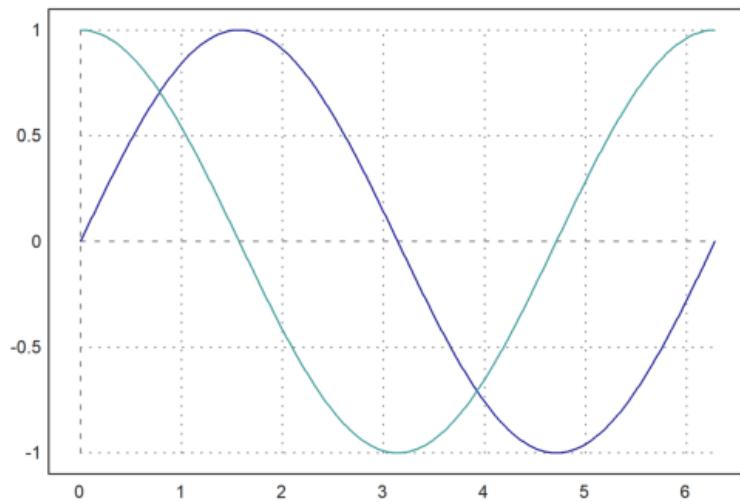
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

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

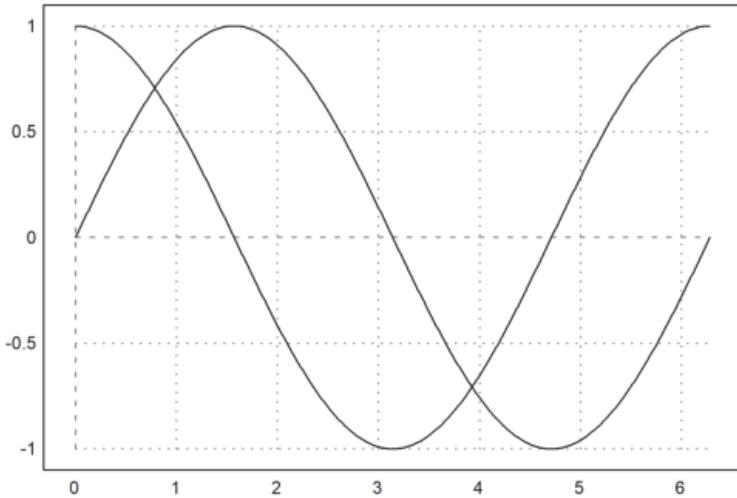


Metode lain adalah menggunakan vektor ekspresi (string). Anda dapat menggunakan susunan warna, susunan gaya, dan susunan ketebalan dengan panjang yang sama.

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



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



Kita bisa mendapatkan vektor seperti itu dari Maxima menggunakan makelist() dan mxm2str().

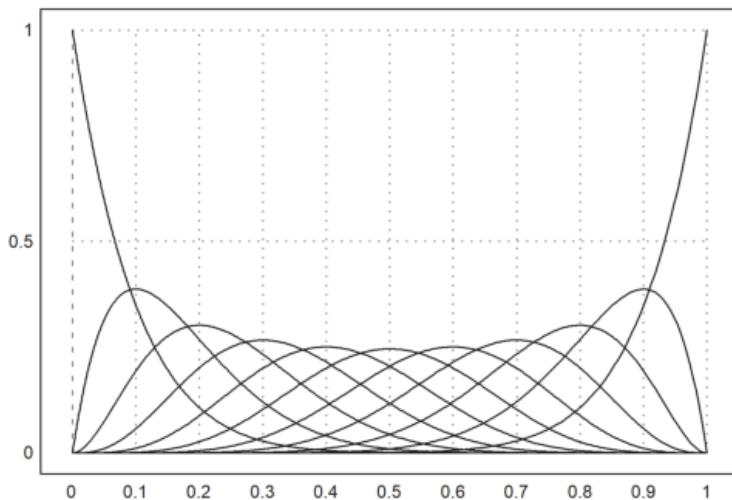
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // membuat list
```

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

```
>mxm2str(v) // mendapatkan vektor string dari vektor simbolik
```

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

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

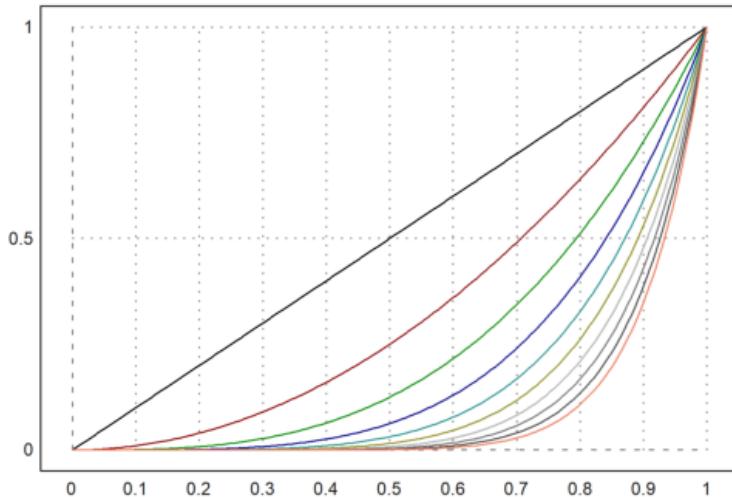


Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

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

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

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

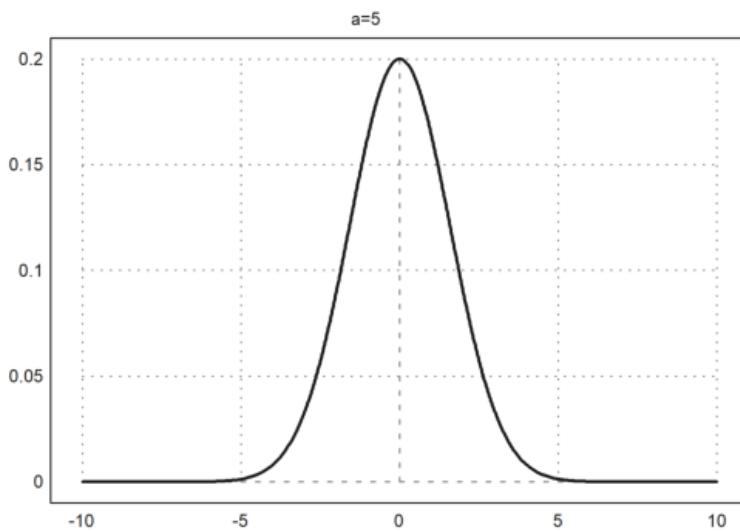


Ekspresi dan fungsi satu baris dapat melihat variabel global.

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

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

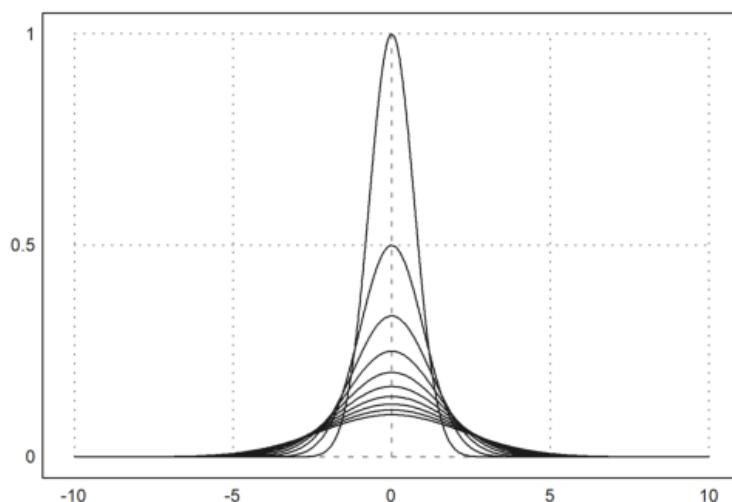
```
>function f(x,a) := 1/a*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5");
```



Alternatifnya, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. List khusus ini disebut kumpulan panggilan, dan ini adalah cara yang lebih disukai untuk meneruskan argumen ke suatu fungsi yang kemudian diteruskan sebagai argumen ke fungsi lain.

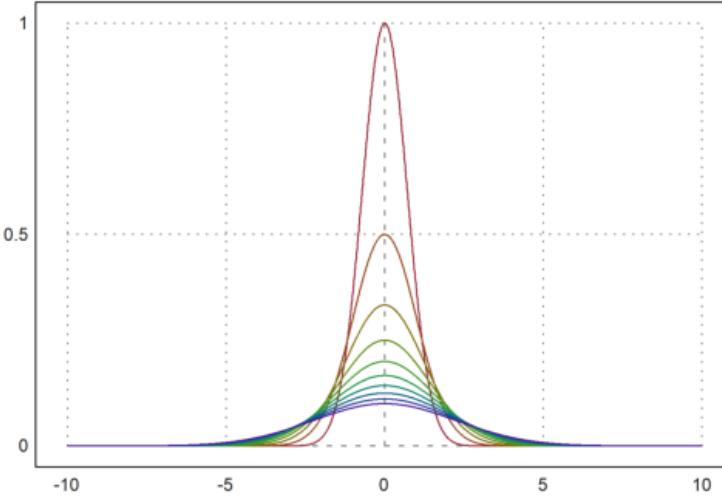
Pada contoh berikut, kita menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman loop).

```
>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 matriks $f(x,a)$ merupakan satu fungsi. Selain itu, kita dapat mengatur warna untuk setiap baris matriks. Klik dua kali pada fungsi getspectral() untuk penjelasannya.

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



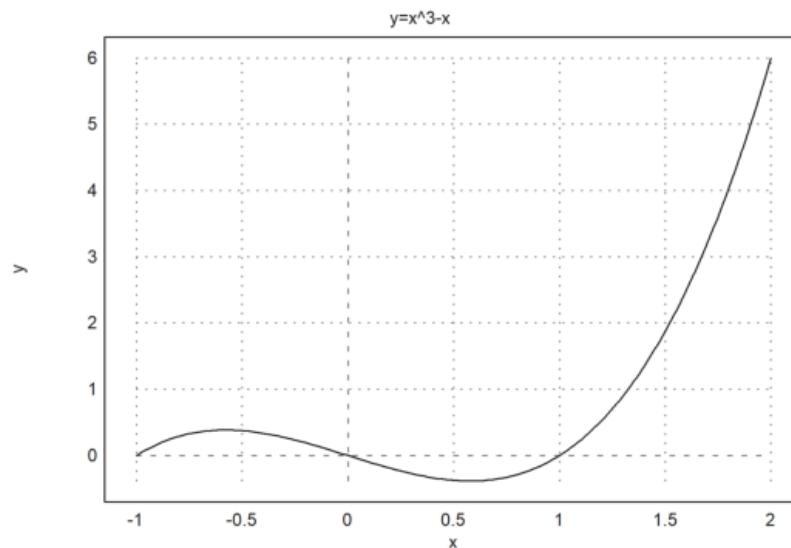
Label Teks

Dekorasi sederhana bisa menjadi

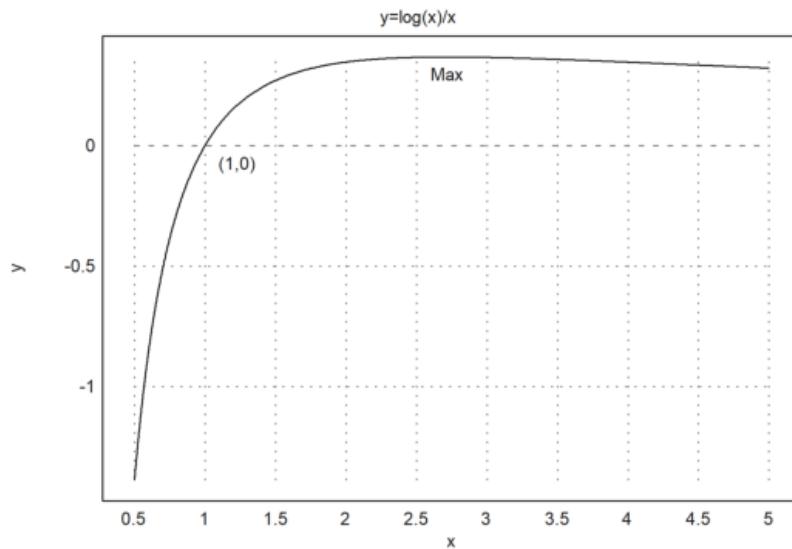
- a title with `title="..."`
- x- and y-labels with `xl="..."`, `yl="..."`
- another text label with `label("...",x,y)`

Perintah label akan memplot ke plot saat ini pada koordinat plot (x,y). Hal ini memerlukan argumen posisional.

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

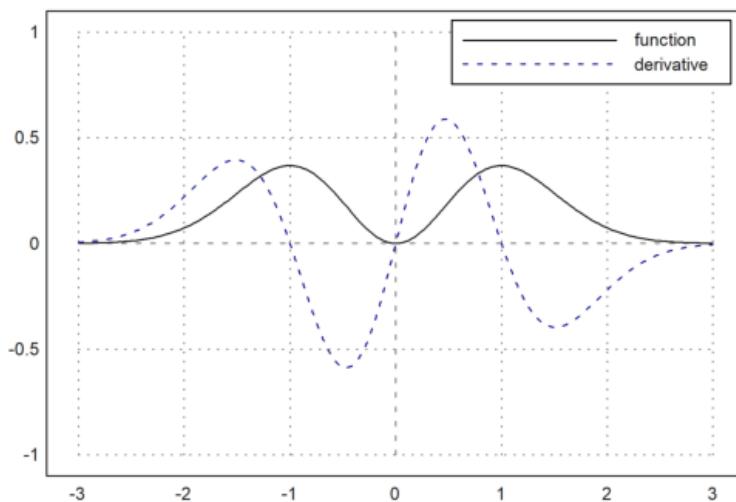


```
>expr := "log(x)/x"; ...
> plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
> label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



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

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

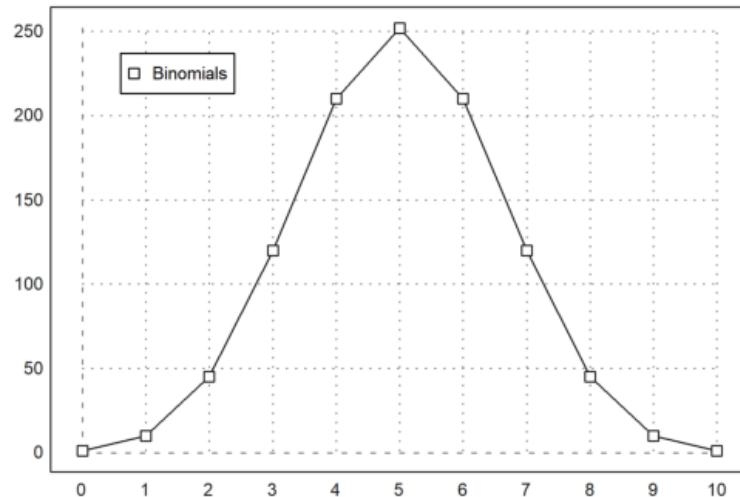


Kotak ini dimulai di kanan atas secara default, tetapi `>left` dimulai di kiri atas. Anda dapat memindahkannya ke tempat mana pun yang Anda suka. Posisi jangkar berada di pojok kanan atas kotak, dan angkanya merupakan pecahan dari ukuran jendela grafis. Lebarnya otomatis.

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

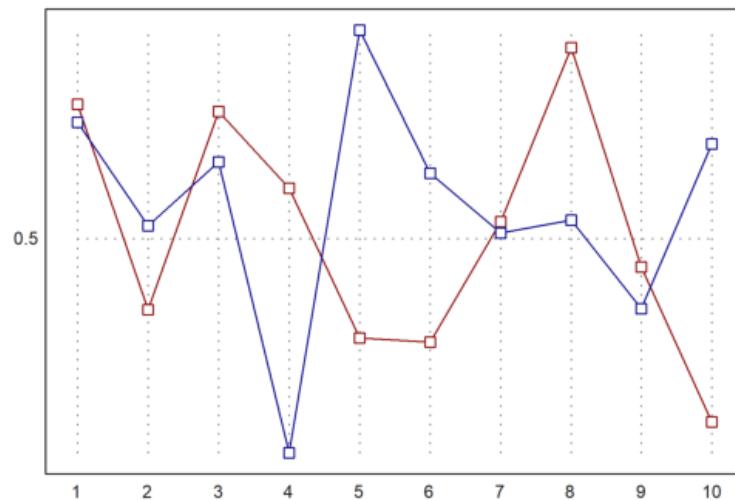
Pada contoh berikut, hanya ada satu fungsi. Jadi kita bisa menggunakan string sebagai pengganti vektor string. Kami mengatur warna teks menjadi hitam untuk contoh ini.

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



Gaya plot tersebut juga tersedia di statplot(). Seperti di plot2d() warna dapat diatur untuk setiap baris plot. Masih banyak lagi plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

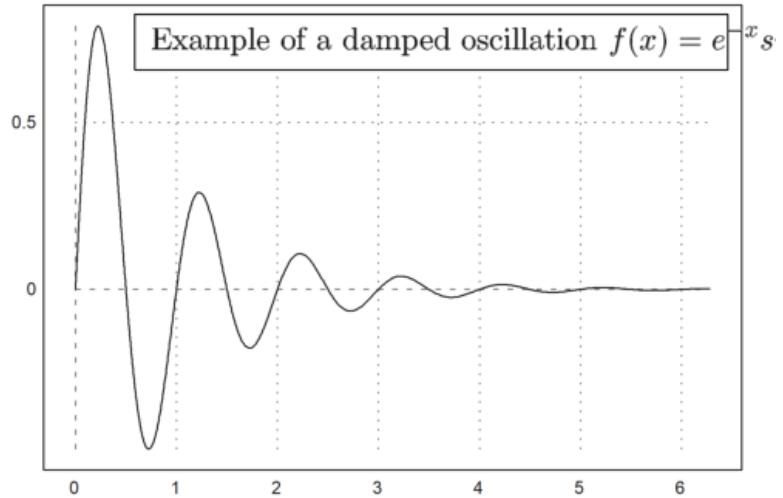
```
>statplot(1:10,random(2,10),color=[red,blue]):
```



Fitur serupa adalah fungsi textbox().

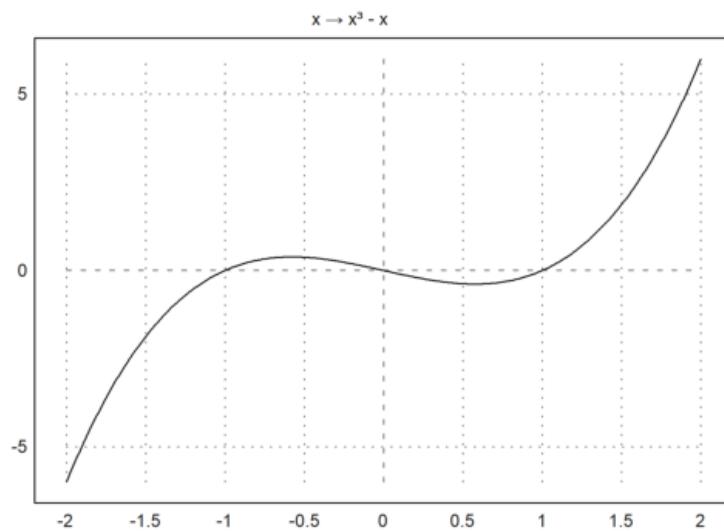
Lebarnya secara default adalah lebar maksimal baris teks. Tetapi itu bisa diatur oleh pengguna juga.

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



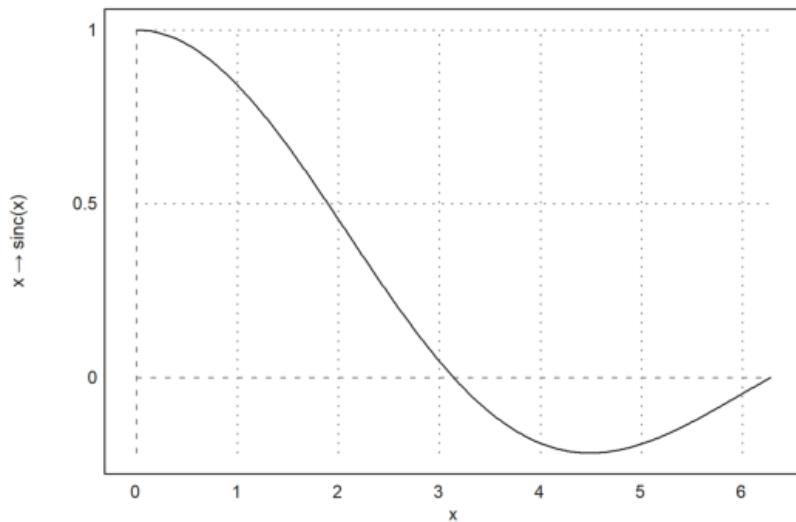
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

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



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

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



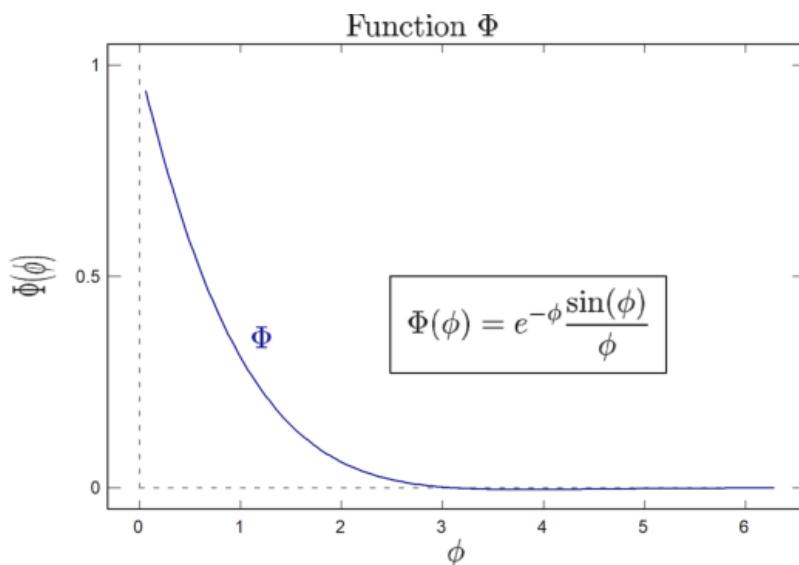
LaTeX

Anda juga dapat memplot rumus LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Langkah-langkah ke biner "lateks" dan "dvipng" harus berada di jalur sistem, atau Anda harus mengatur LaTeX di menu pengaturan.

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

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

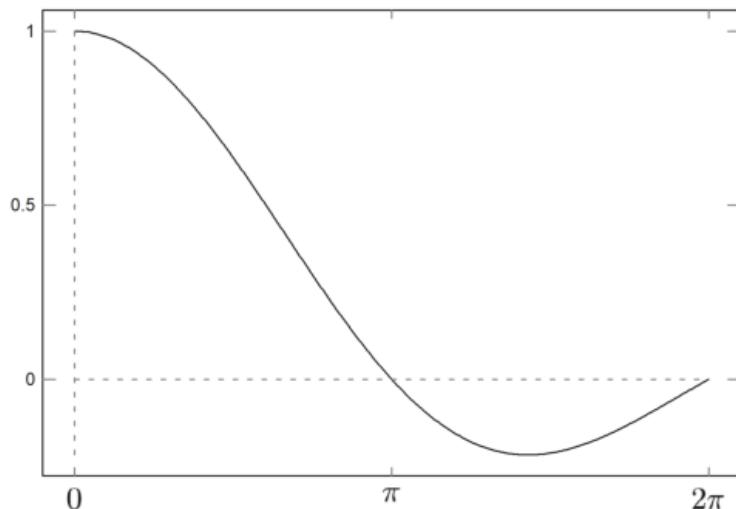
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
> title=latex("\text{Function } \Phi"), ...
> xl=latex("\phi"),yl=latex("\Phi(\phi)"); ...
>textbox( ...
> latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
>label(latex("\Phi",color=blue),1,0.4):
```



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

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

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



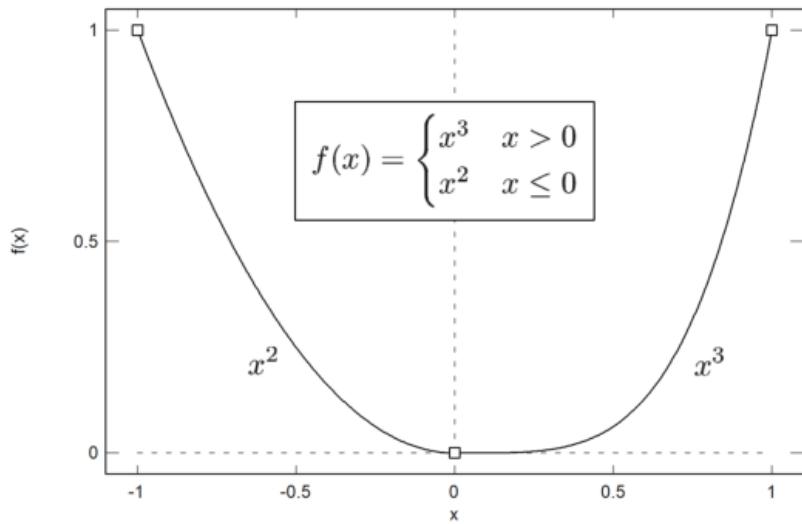
Tentu saja fungsinya juga bisa digunakan.

```
>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 lot, itu tidak perlu. Tapi untuk menunjukkan vektorisasi itu diperlukan, kita menambahkan beberapa poin penting ke plot di $x=-1$, $x=0$ dan $x=1$.

Pada plot berikut, kita juga memasukkan beberapa kode LaTeX. Kita menggunakananya untuk ua label dan kotak teks. Tentu saja, Anda hanya dapat menggunakananya aTeX jika Anda telah menginstal LaTeX dengan tepat.

```
>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 \leq 0 \end{cases}"), ...
>  x=0.7,y=0.2):
```



Interaksi pengguna

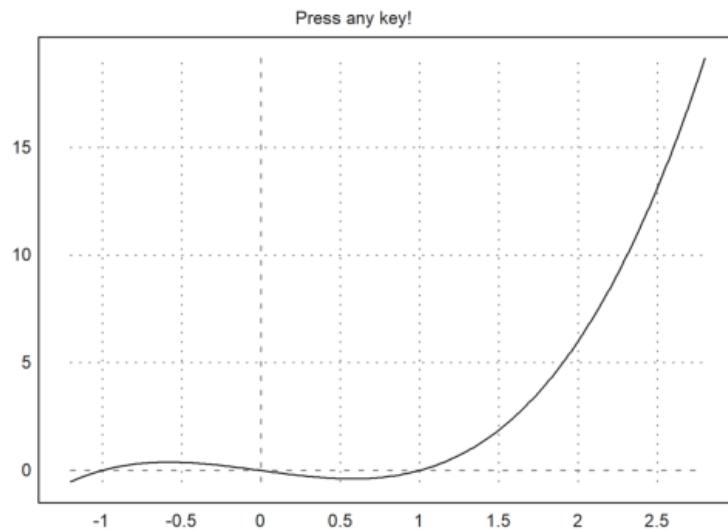
Saat memplot suatu fungsi atau ekspresi, parameter `>user` memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kurSOR atau mouse. Pengguna bisa

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

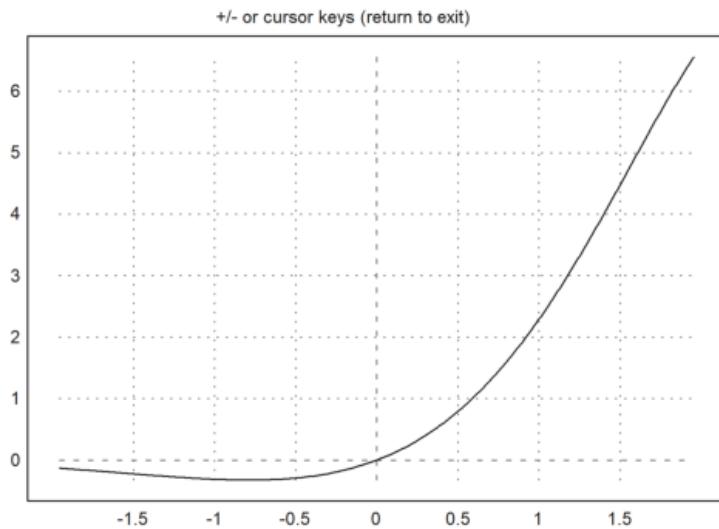
Tombol spasi akan mengatur ulang plot ke jendela plot aslinya.

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

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



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



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

Fungsi bawaan mousedrag() menunggu aktivitas mouse atau keyboard. Ini menerima laporan mouse ke bawah, gerakan mouse atau mouse ke atas, dan penekanan tombol. Fungsi dragpoints() memanfaatkan ini, dan memungkinkan pengguna menyeret titik mana pun dalam plot.

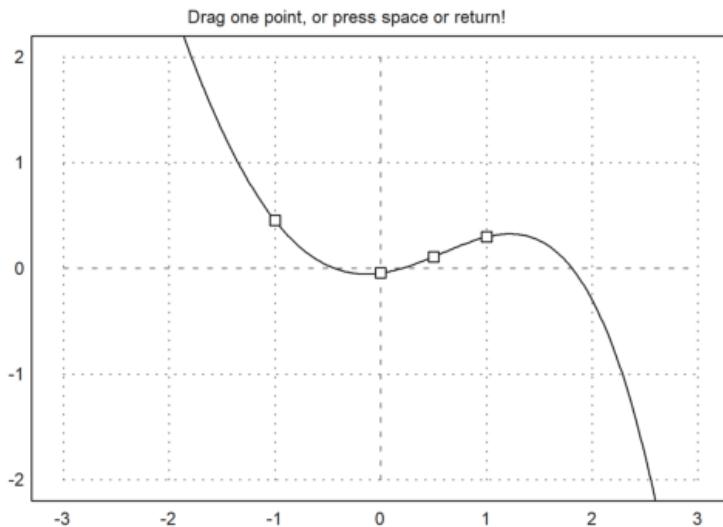
Kita membutuhkan fungsi plot terlebih dahulu. Misalnya, kita melakukan interpolasi pada 5 titik dengan polinomial. Fungsi tersebut harus diplot ke dalam area plot yang tetap.

```
>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 di plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, untuk mengakses nilainya secara global.

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

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



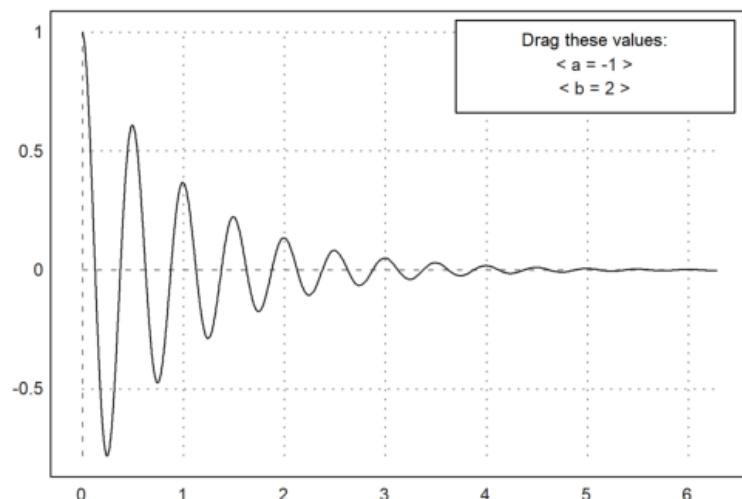
Ada juga fungsi yang memplot fungsi lain bergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

Pertama kita membutuhkan fungsi plot.

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

Kemudian kita memerlukan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional garis judul. Terdapat penggeser interaktif, yang dapat menetapkan nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

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

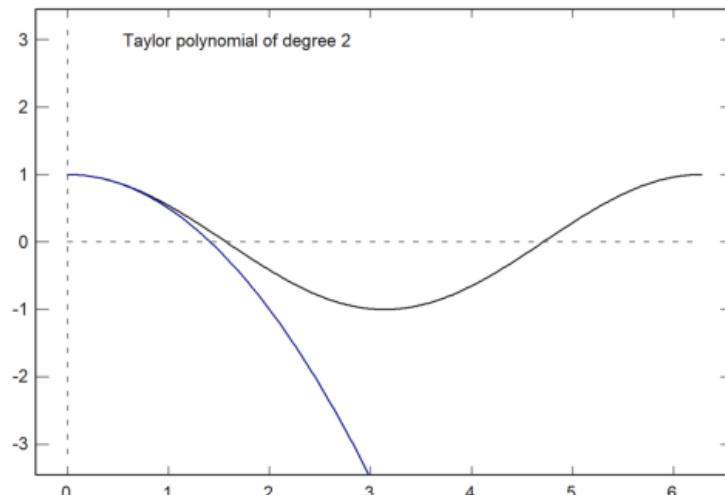


Ini dimungkinkan untuk membatasi nilai yang diseret menjadi bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor berderajat n ke fungsi kosinus.

```
>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 izinkan derajat n bervariasi dari 0 hingga 20 dalam 20 perhentian. Hasil dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk memasukkan plot ke dalam notebook.

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



Berikut ini adalah demonstrasi sederhana dari fungsinya. Pengguna dapat menggambar jendela plot, meninggalkan jejak titik.

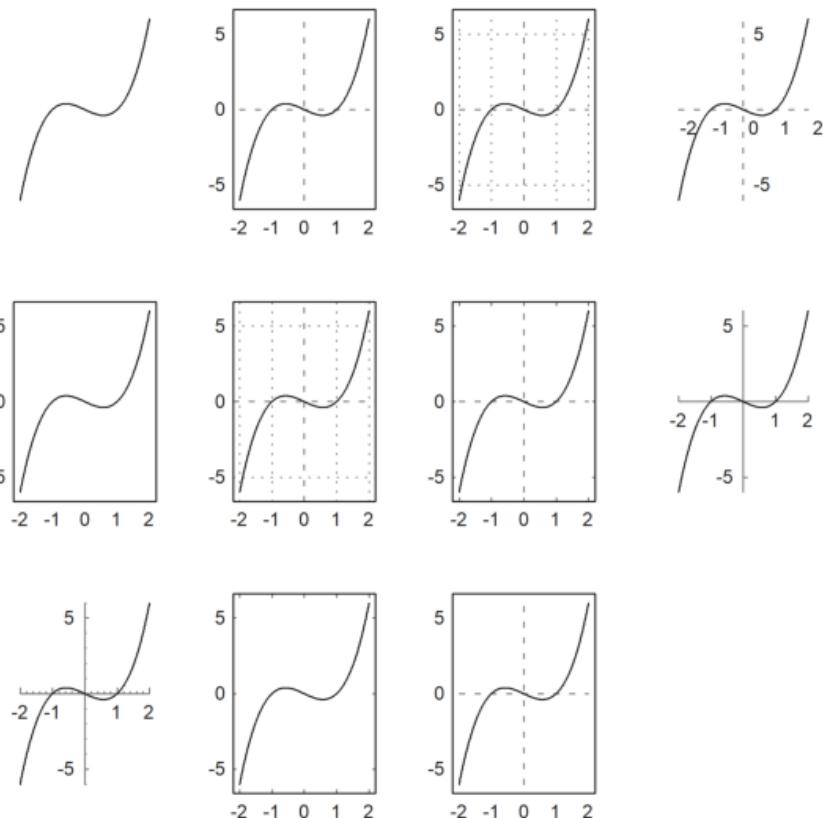
```
>function dragtest ...
    plot2d(None,r=1,title="Drag with the mouse, or press any key!");
    start=0;
repeat
    {flag,m,time}=mousedrag();
    if flag==0 then return; endif;
    if flag==2 then
        hold on; mark(m[1],m[2]); hold off;
    endif;
end
endfunction
```

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

Gaya Plot 2D

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

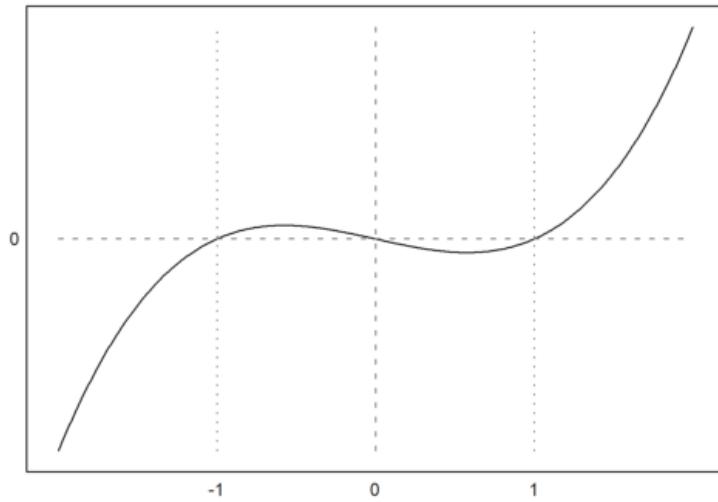
```
>aspect();  
>figure(3,4); ...  
> figure(1); plot2d("x^3-x",grid=0); ... // tidak ada grid, bingkai, atau sumbu  
> figure(2); plot2d("x^3-x",grid=1); ... // sumbu-xy  
> figure(3); plot2d("x^3-x",grid=2); ... // default ticks  
> figure(4); plot2d("x^3-x",grid=3); ... // sumbu-xy dengan label di dalamnya  
> figure(5); plot2d("x^3-x",grid=4); ... // tidak ada ticks, hanya label  
> figure(6); plot2d("x^3-x",grid=5); ... // default, tapi tidak ada margin  
> figure(7); plot2d("x^3-x",grid=6); ... // hanya sumbu  
> figure(8); plot2d("x^3-x",grid=7); ... // hanya sumbu, ticks pada sumbu  
> figure(9); plot2d("x^3-x",grid=8); ... // hanya sumbu, finer ticks pada sumbu  
> figure(10); plot2d("x^3-x",grid=9); ... // default, ticks kecil di dalam  
> figure(11); plot2d("x^3-x",grid=10); ...// tidak ada ticks, hanya sumbu  
> figure(0):
```



Parameter `<frame` menggantikan `frame`, dan `framecolor=blue` mengatur frame menjadi warna biru.

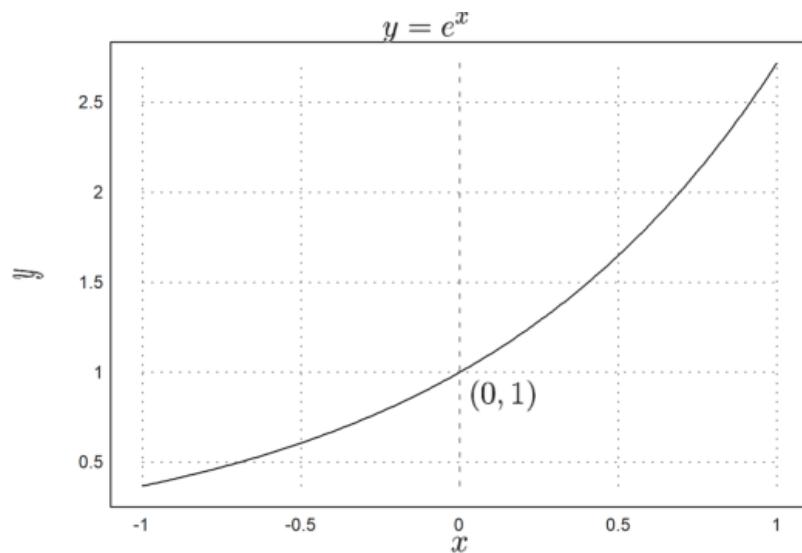
Jika Anda menginginkan tanda centang Anda sendiri, Anda dapat menggunakan `style=0`, dan menambahkan semuanya nanti.

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



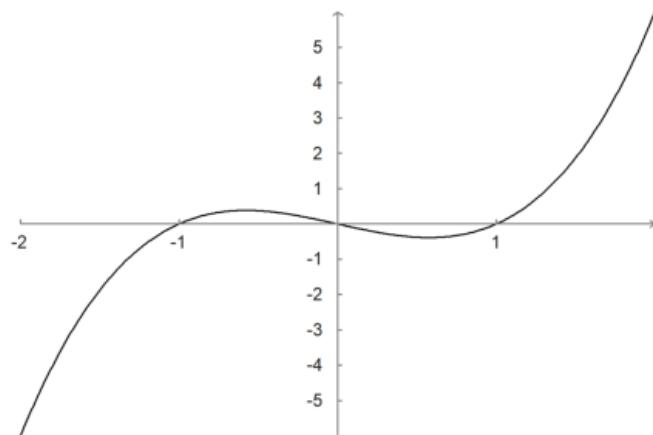
Untuk judul plot dan label sumbu, lihat contoh berikut.

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



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

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

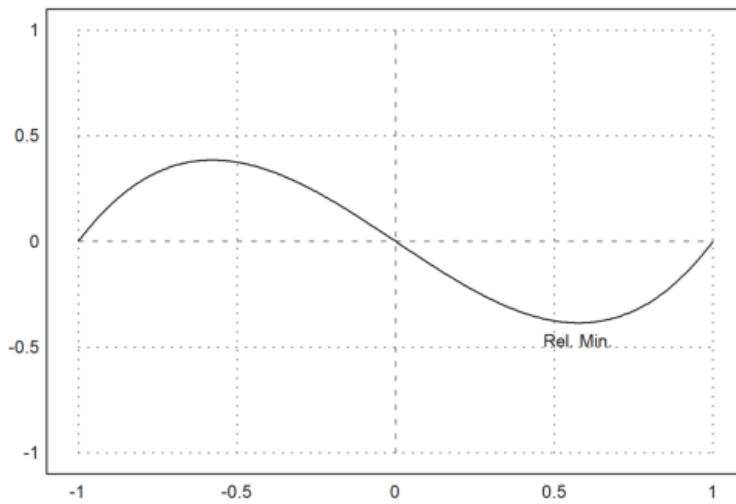


Teks pada plot dapat diatur dengan label(). Dalam contoh berikut, "lc" berarti bagian tengah bawah. Ini menetapkan posisi label relatif terhadap koordinat plot.

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

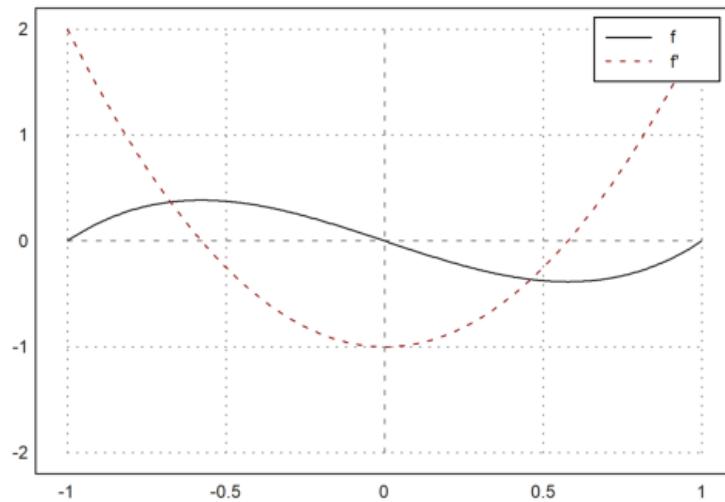
$$x^3 - x$$

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

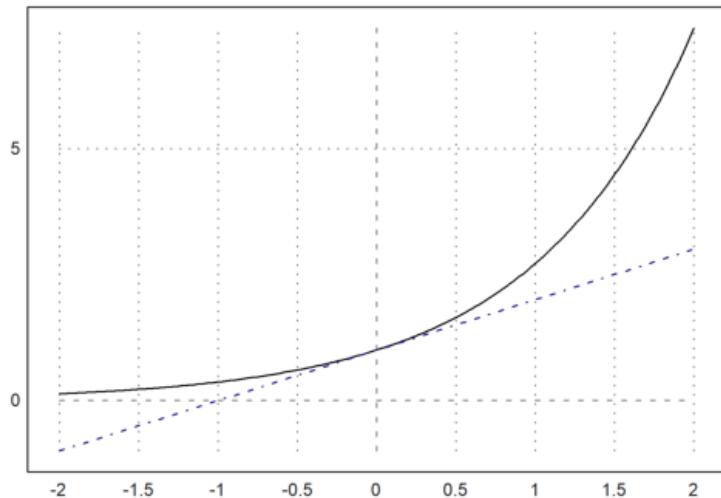


Ada juga kotak teks.

```
>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
```



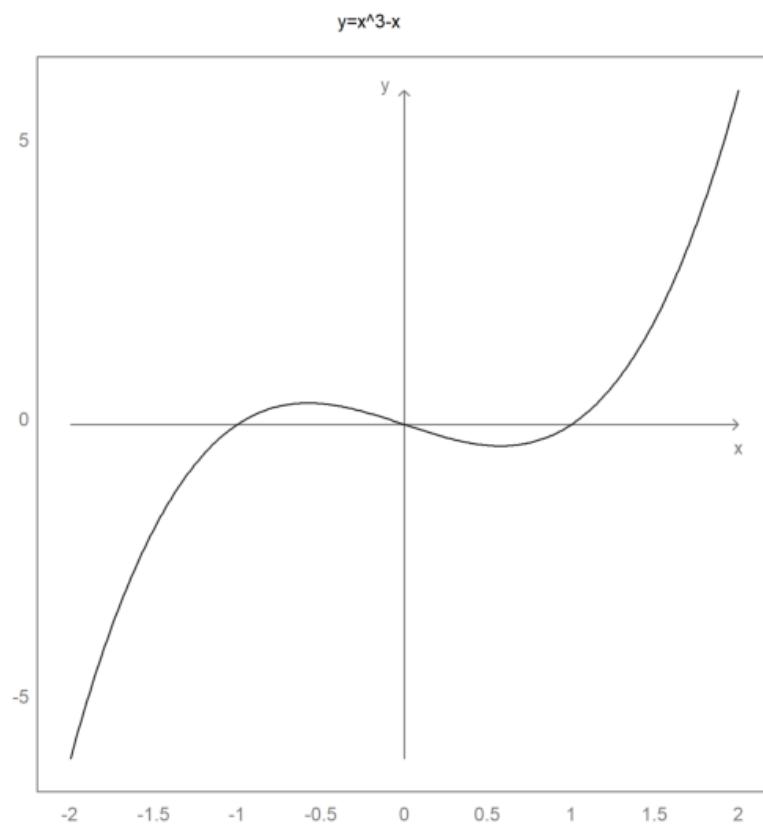
```
>plot2d(["exp(x)","1+x"],color=[black,blue],style=["-","-.-"]):
```



```

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

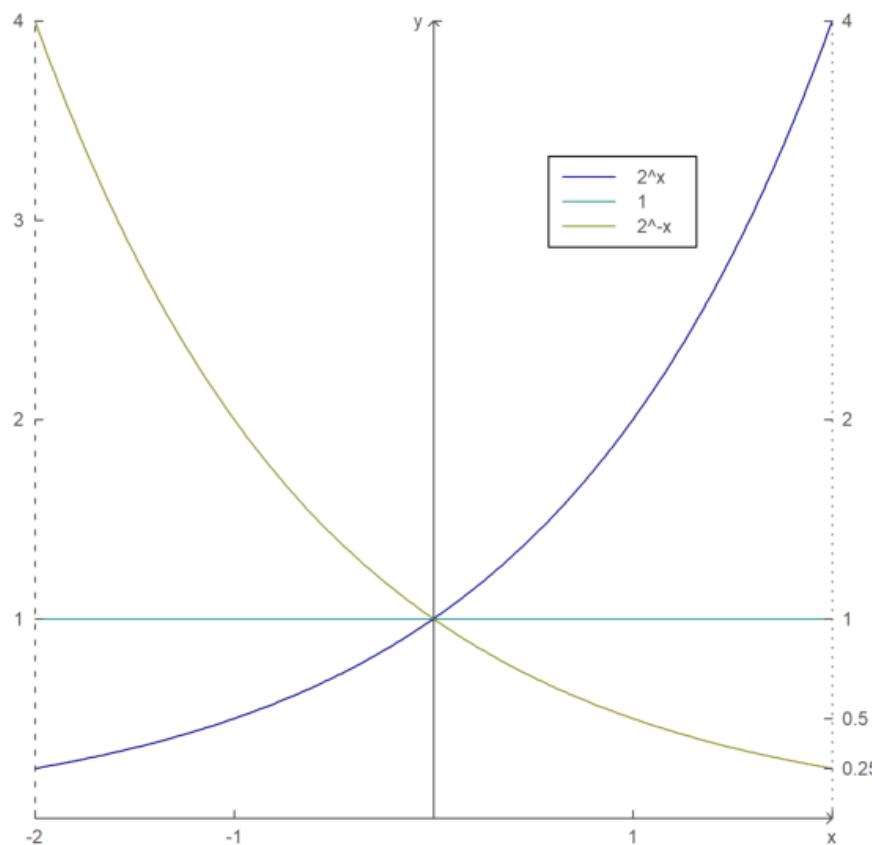
```



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

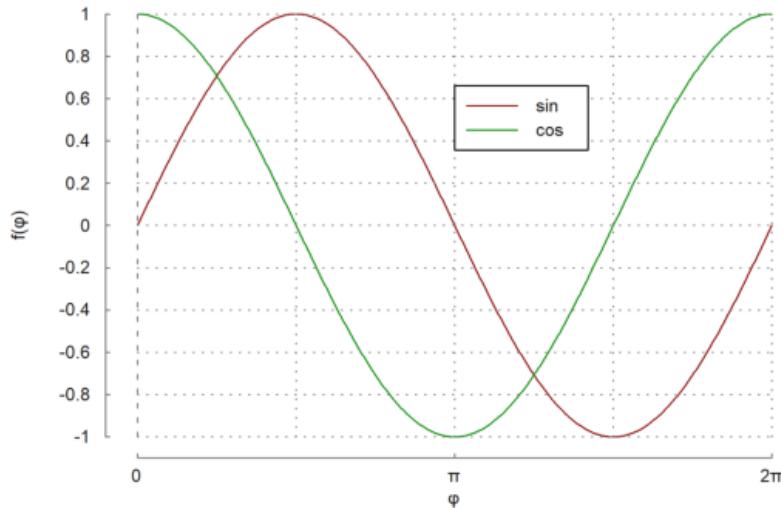
Perintah fullwindow() memperluas jendela plot karena kita tidak lagi memerlukan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk menyetel ulang ke default.

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



Berikut adalah contoh lain, di mana string Unicode digunakan dan sumbunya berada 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"\u03c0;","u"2\u03c0;"],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"\u03c6"); ylabel(u"f(\u03c6)"):
```



Memploting Data 2D

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

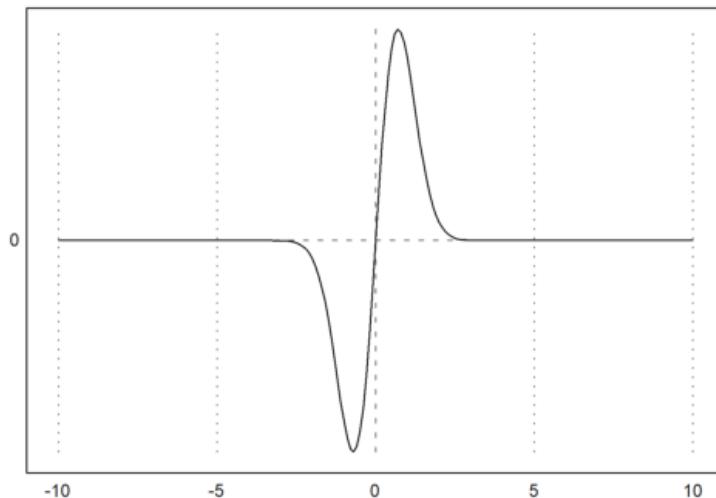
Ploting ekspresi hanyalah singkatan dari plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai x , dan satu atau beberapa baris nilai y . Dari rentang dan nilai x , fungsi plot2d akan menghitung data yang akan diplot, secara default dengan evaluasi fungsi yang adaptif. Untuk plot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

Tapi Anda bisa 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);
```



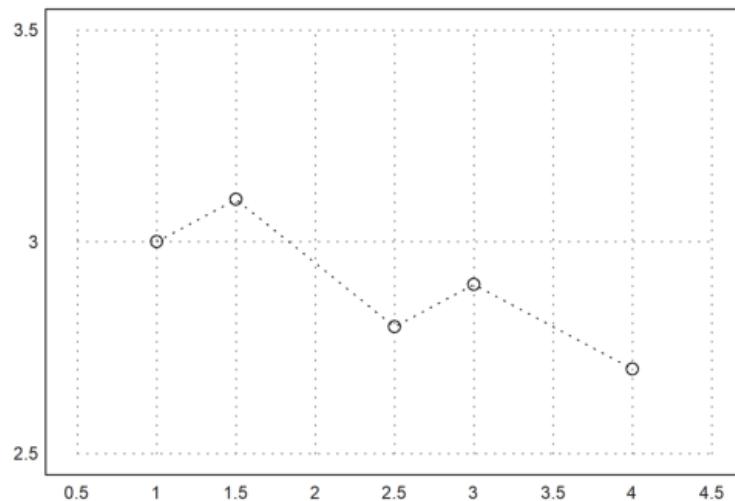
Data juga dapat diplot sebagai poin. Gunakan points=true untuk ini. Plotnya berfungsi seperti poligon, tetapi hanya menggambar sudutnya saja.

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

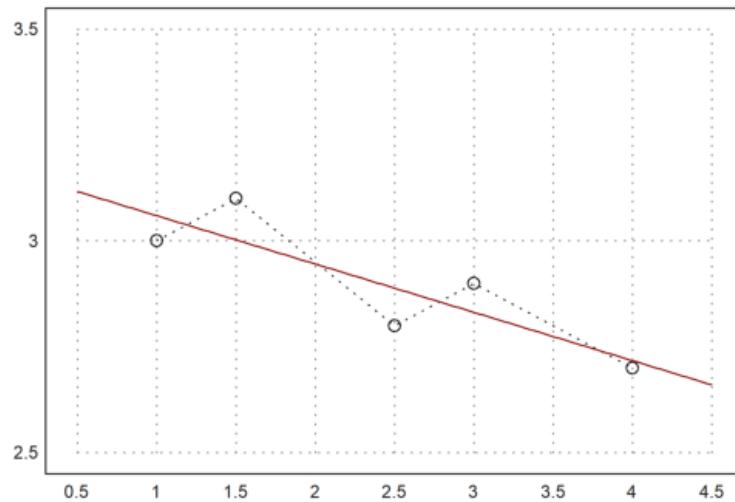
Untuk memplot kumpulan titik, gunakan >points. Jika warna merupakan vektor warna, maka setiap titik mendapat warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna diterapkan pada baris matriks.

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"); // menambahkan titik
```



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



Menggambar Daerah Yang Dibatasi Kurva

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

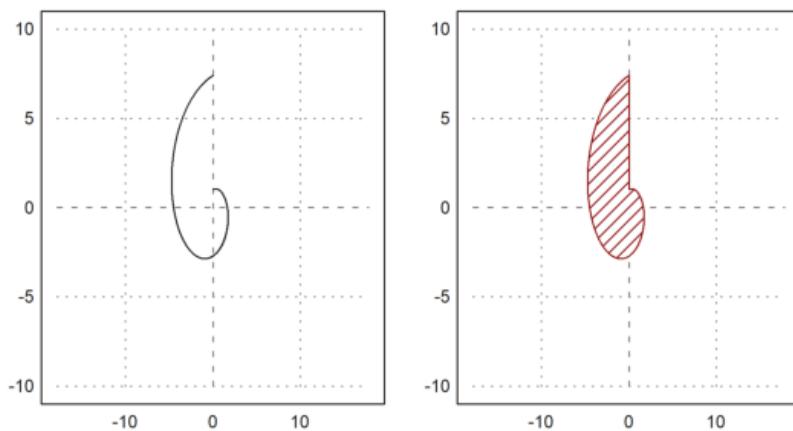
- filled=true fills the plot.

- style="...": Select from "", "/", "\\", "\\\".

- fillcolor: Lihat di atas untuk mengetahui warna yang tersedia.

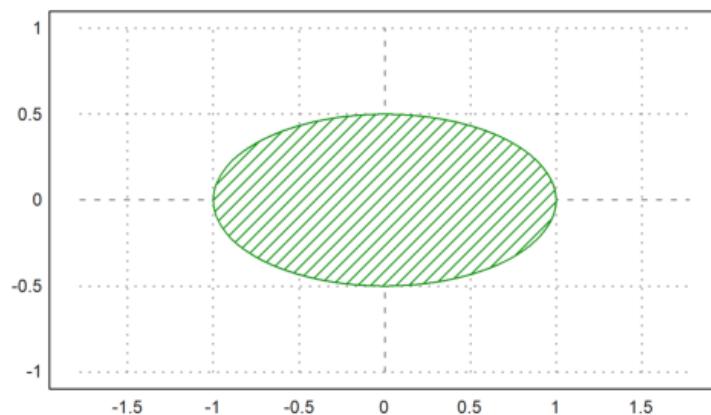
Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline Opsional mencegah menggambar melewati batas untuk semua gaya kecuali gaya default.

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

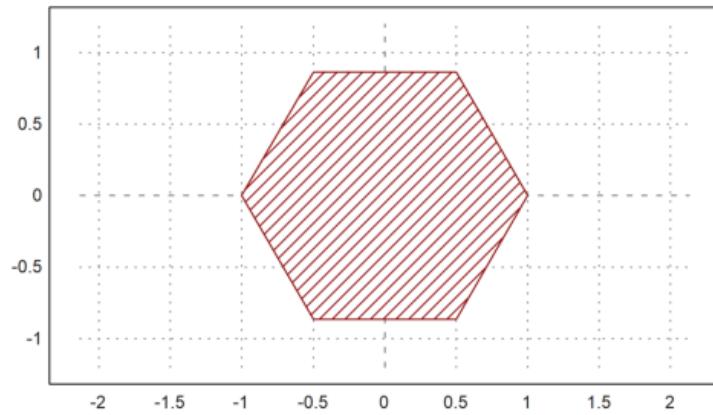


Dalam contoh berikut kita memplot elips yang terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian berbeda.

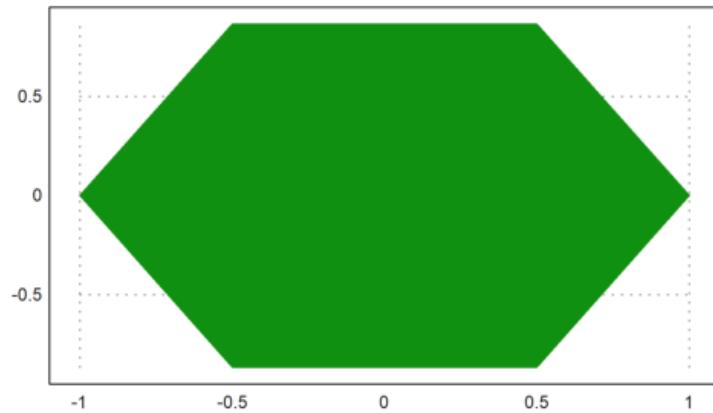
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```



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

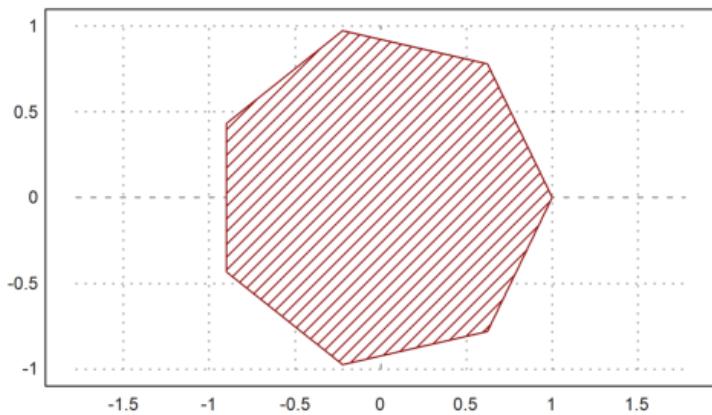


```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#" ):
```



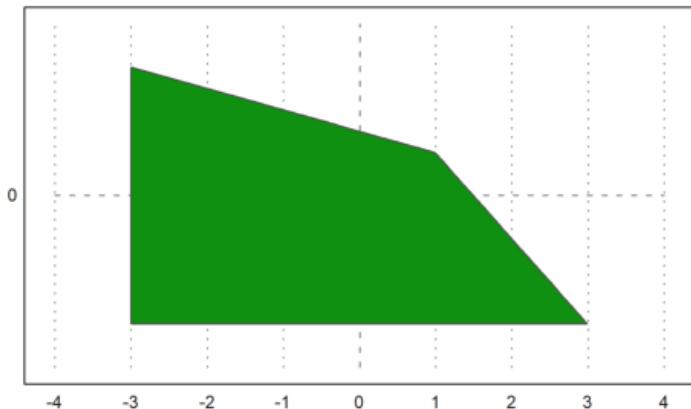
Contoh lainnya adalah septagon yang kita buat dengan 7 titik pada lingkaran satuan.

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



Berikut adalah himpunan nilai maksimal dari empat kondisi linier yang kurang dari atau sama dengan 3. Ini adalah $A[k].v \leq 3$ untuk semua baris A . Untuk mendapatkan sudut yang diinginkan, kita menggunakan n yang relatif besar.

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

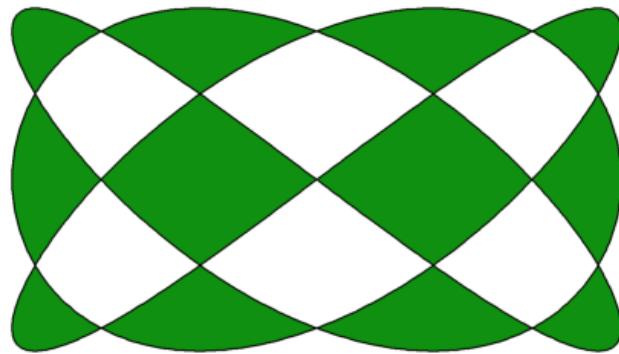


Poin utama dari bahasa matriks adalah memungkinkan pembuatan tabel fungsi dengan mudah.

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

Kami sekarang memiliki nilai vektor x dan y . `plot2d()` dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plotnya bisa diisi. Dalam hal ini hasil yang bagus diperoleh karena adanya aturan lilitan yang digunakan untuk pengisian.

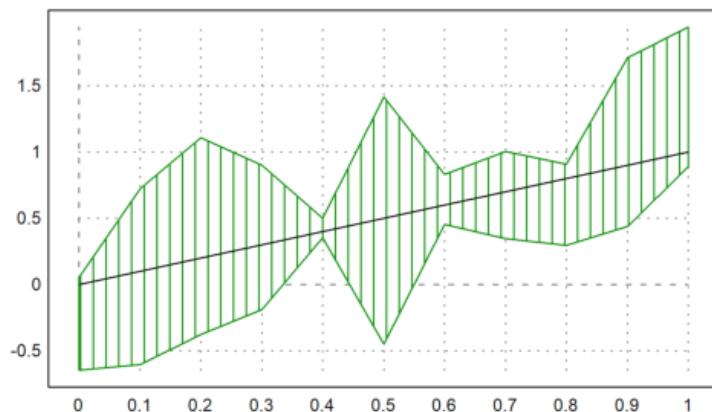
```
>plot2d(x,y,<grid,<frame,>filled):
```



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

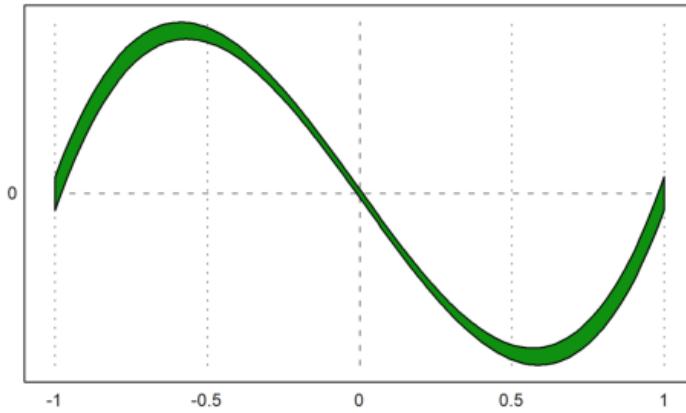
Hal ini dapat berguna untuk memplot kesalahan perhitungan. Namun hal ini juga dapat digunakan untuk memetakan kesalahan statistik.

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



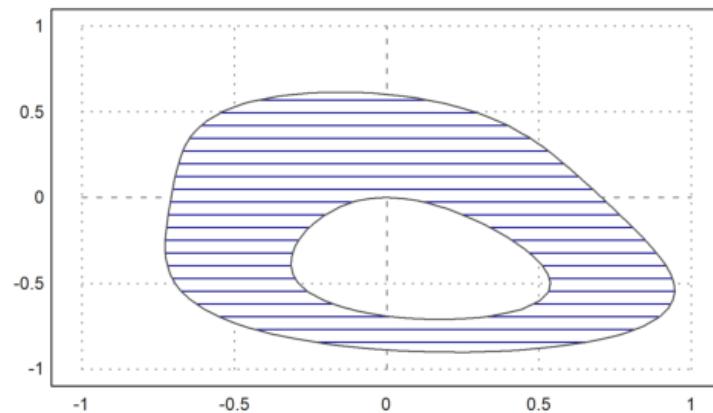
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi pada bidang. Gaya isiannya sama dengan gaya poligon.

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



Ini dimungkinkan untuk mengisi daerah nilai untuk fungsi tertentu. Untuk ini, tingkatan harus berupa matriks $2 \times n$. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

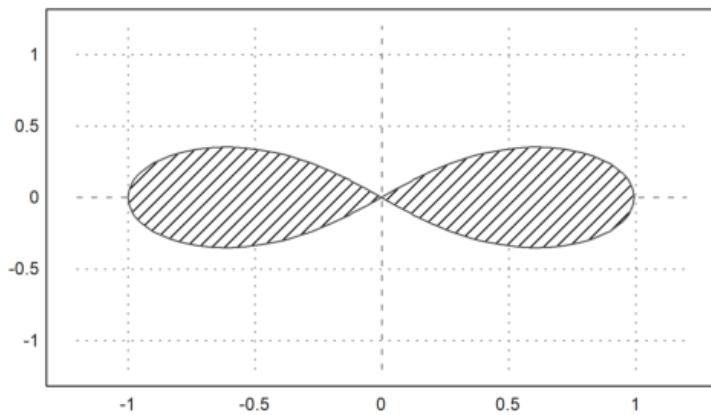
```
>expr := "2*x^2+x*y+3*y^4+y"; // mendefinisikan ekspresi f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue); // 0 <= f(x,y) <= 1
```



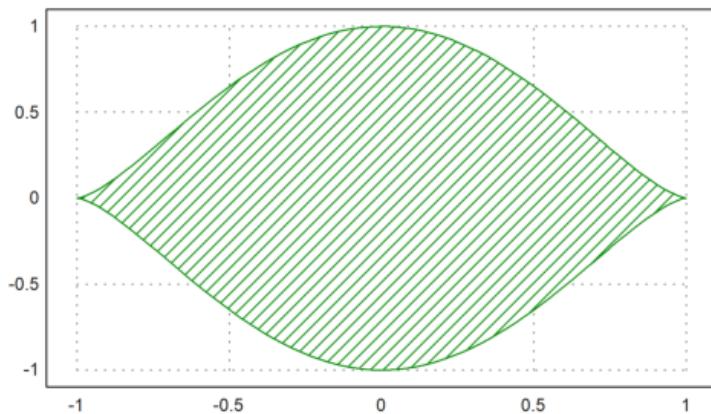
Kita juga dapat mengisi rentang nilai seperti

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

```
>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

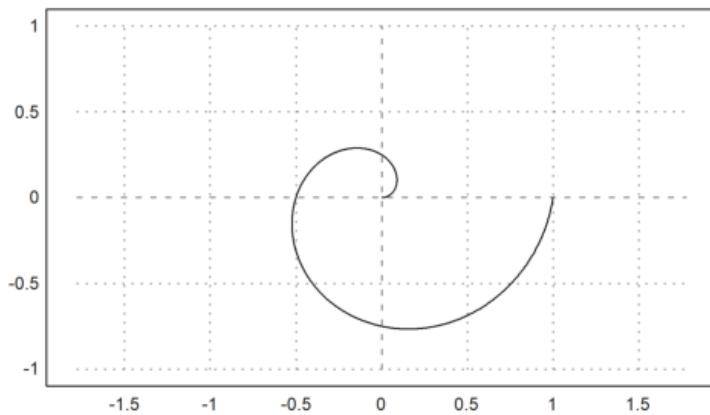
Nilai x tidak perlu diurutkan. (x,y) hanya menggambarkan sebuah kurva. Jika x diurutkan, kurva tersebut merupakan grafik suatu fungsi.

Dalam contoh berikut, kita memplot spiral

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

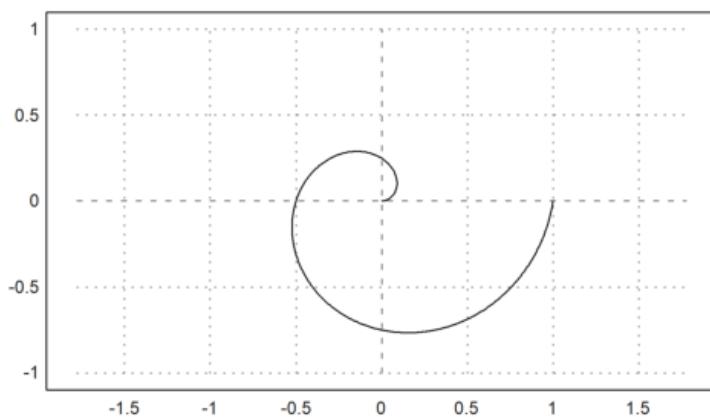
Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk lebih jelasnya).

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

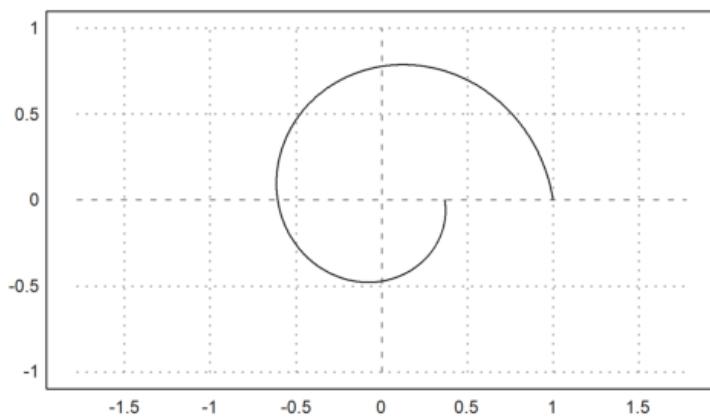


Sebagai alternatif, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

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



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



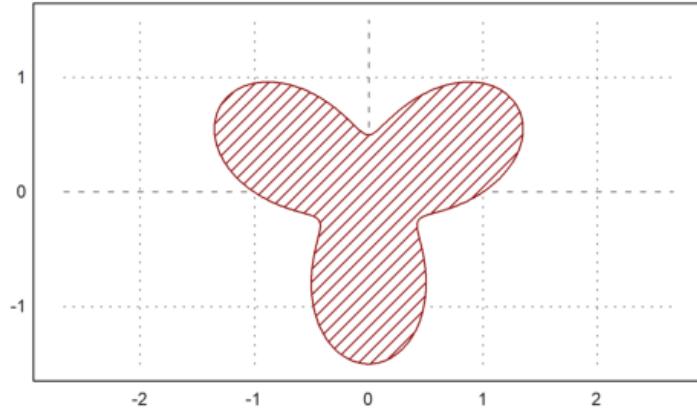
Pada contoh berikutnya, kita memplot kurvanya

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

dengan

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

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

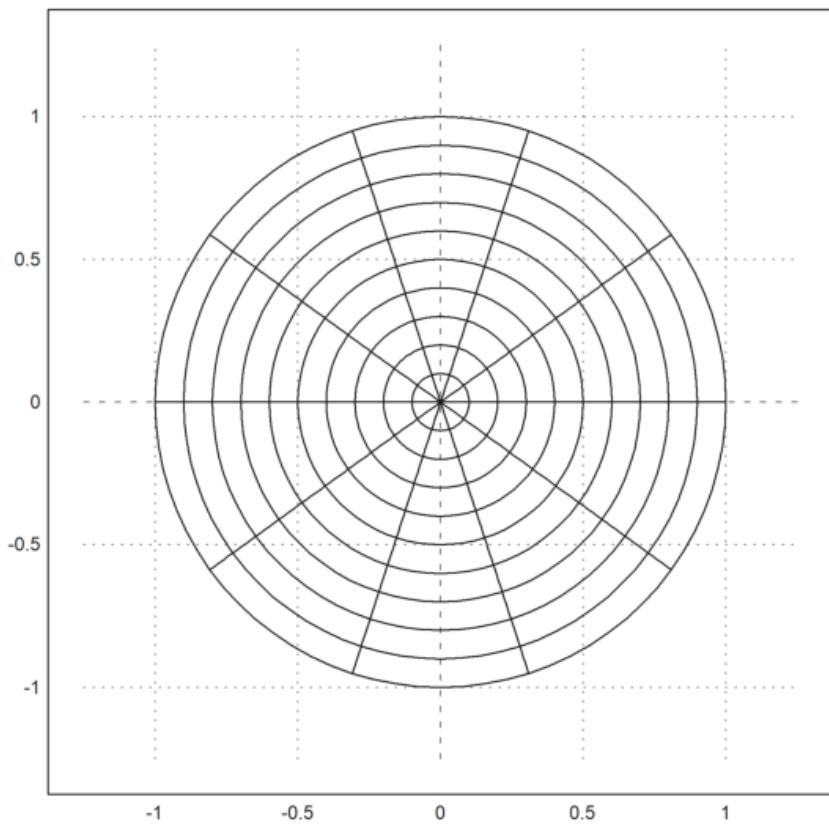


Menggambar Grafik Bilangan Kompleks

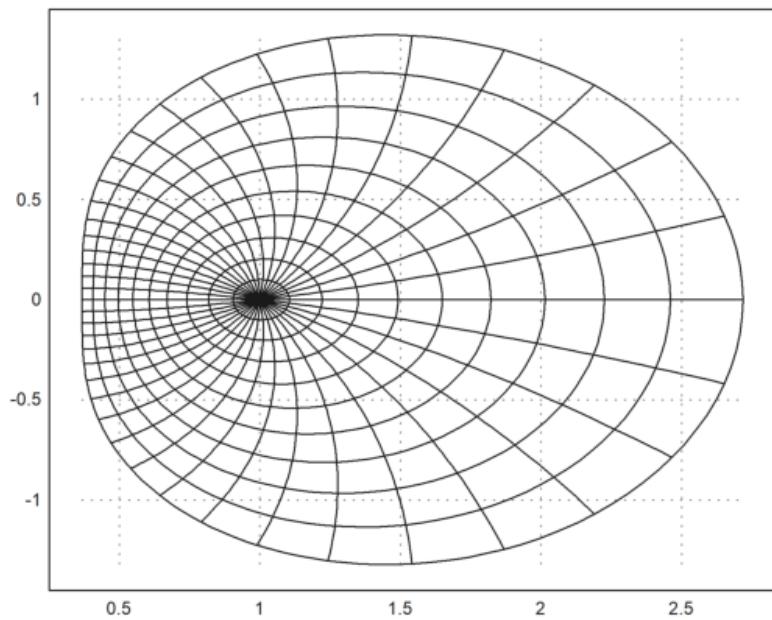
Serangkaian bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan dihubungkan. Jika sejumlah garis grid ditentukan (atau vektor garis grid 1×2) dalam argumen cgrid, hanya garis grid tersebut yang terlihat. Matriks bilangan kompleks secara otomatis akan diplot sebagai grid pada bidang kompleks.

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

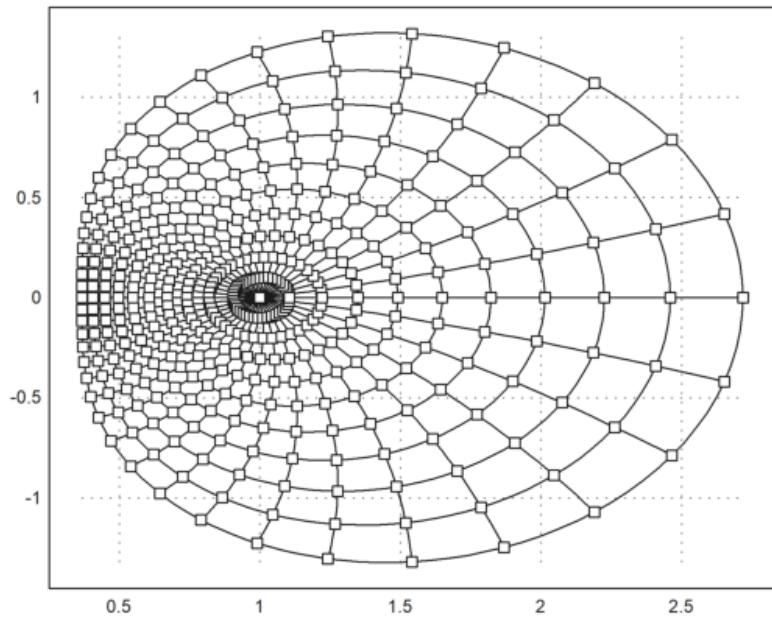
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```



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



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

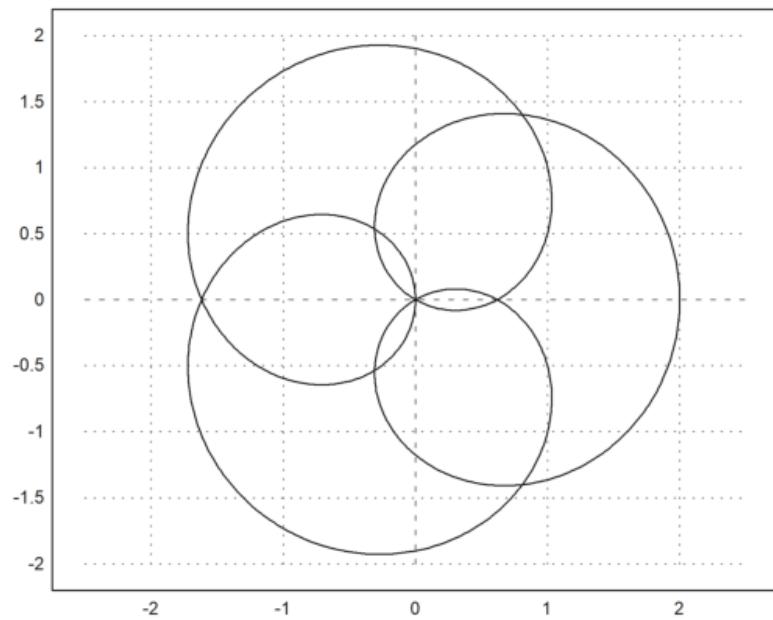


Vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian nyata dan bagian imajiner.

Dalam contoh, kita memplot lingkaran satuan dengan

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

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



Plot Statistik

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

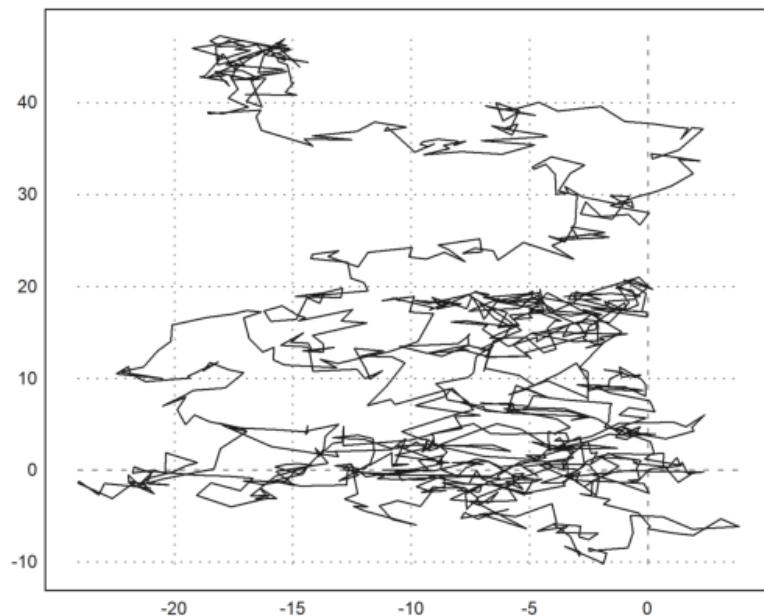
Jumlah kumulatif dari nilai terdistribusi normal 0-1 menghasilkan cara acak.

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

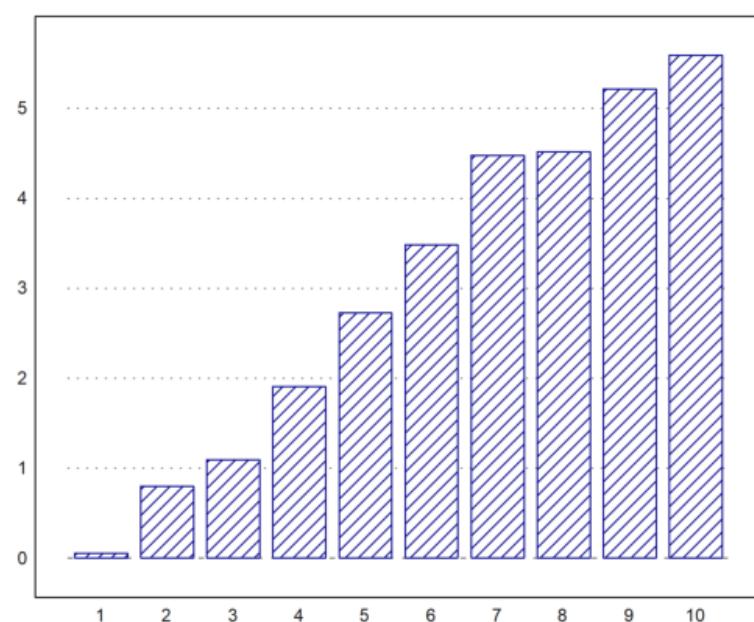


Penggunaan dua baris menunjukkan langkah-langkah dalam dua dimensi.

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



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

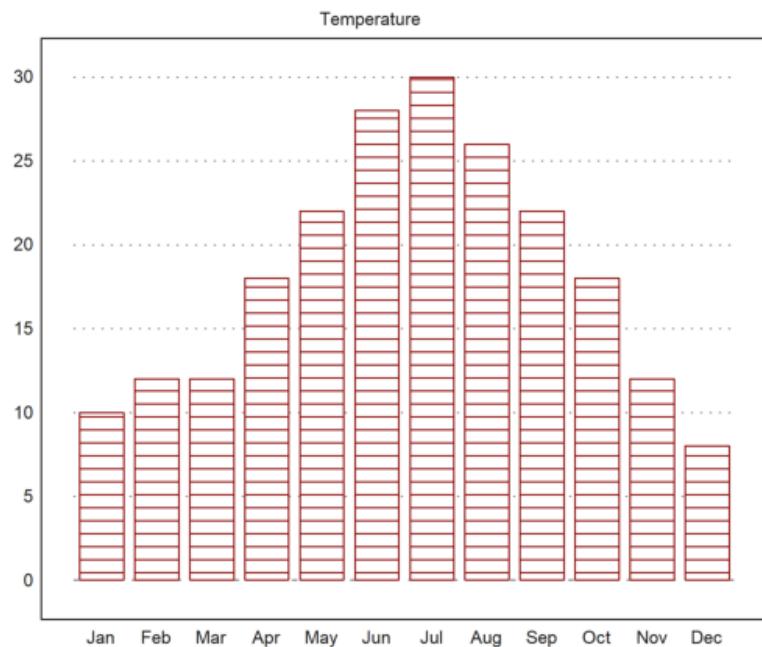


Itu juga dapat menampilkan string sebagai label.

```

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

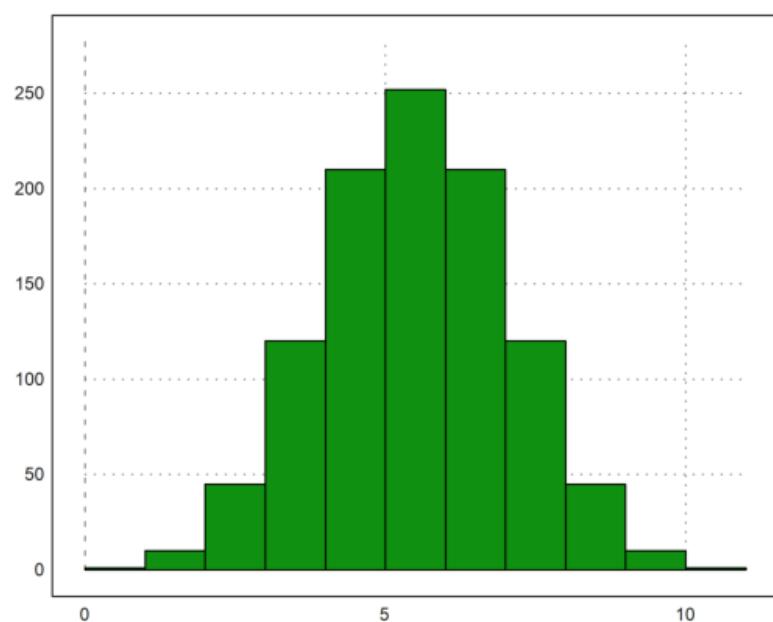
```



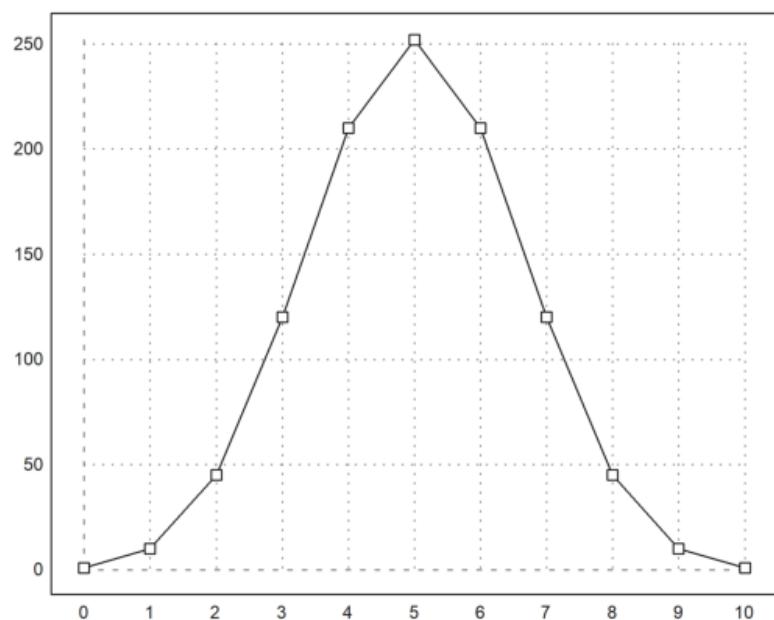
```

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

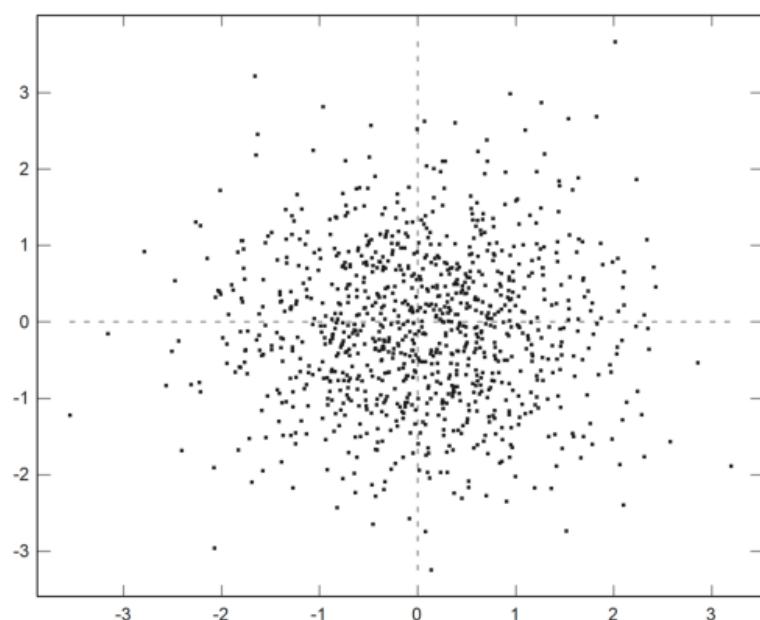
```



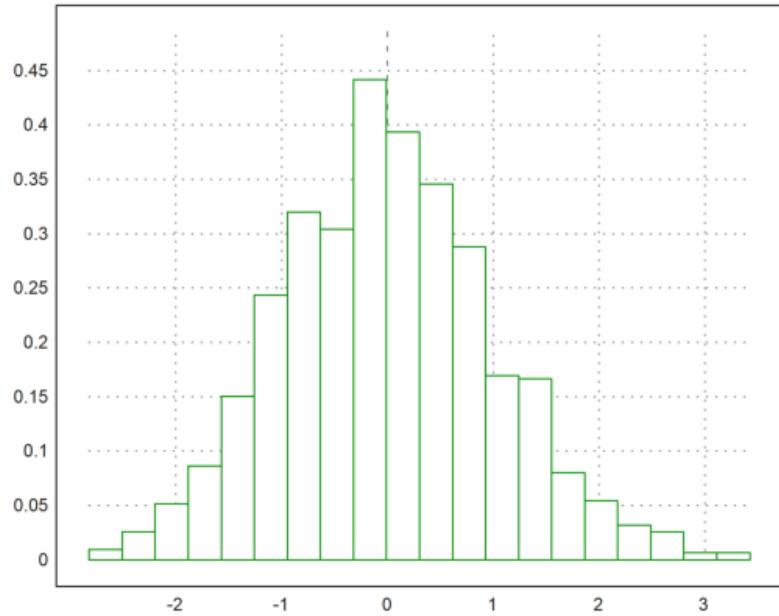
```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



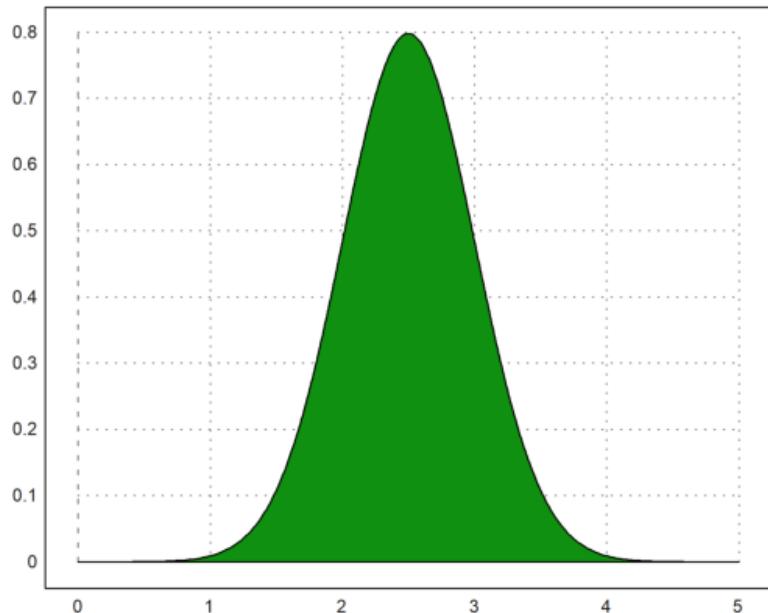
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=". . ."):
```



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

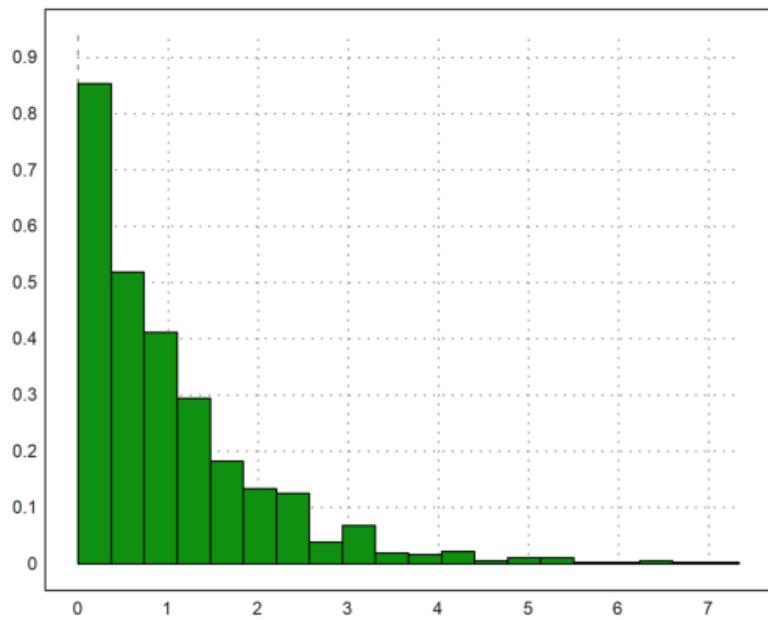


```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



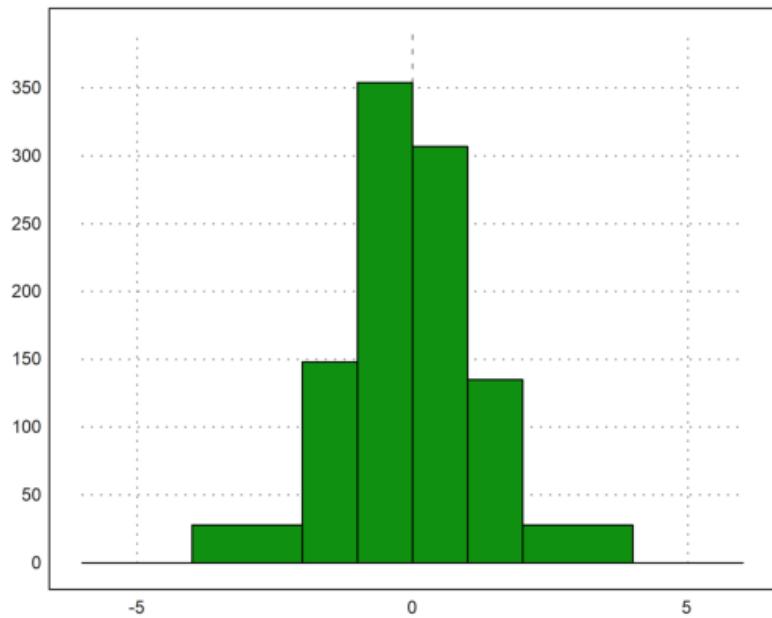
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

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



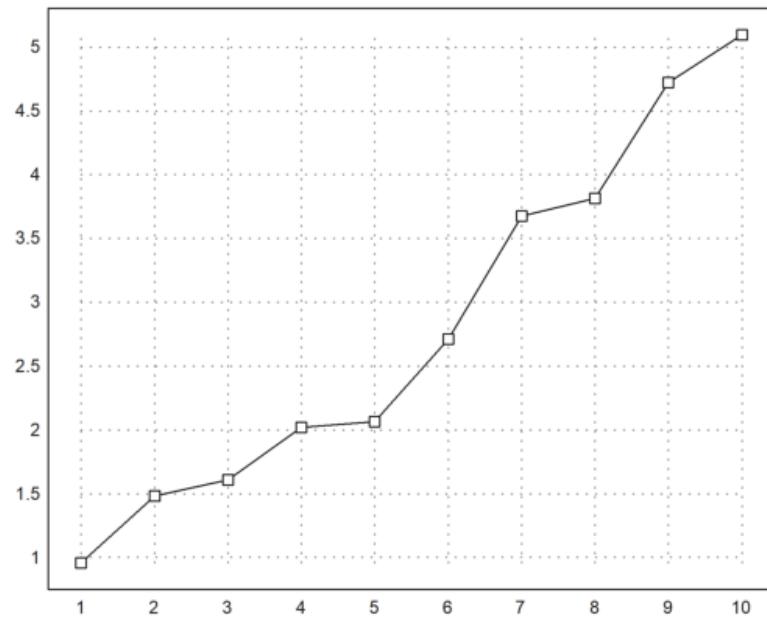
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom.

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

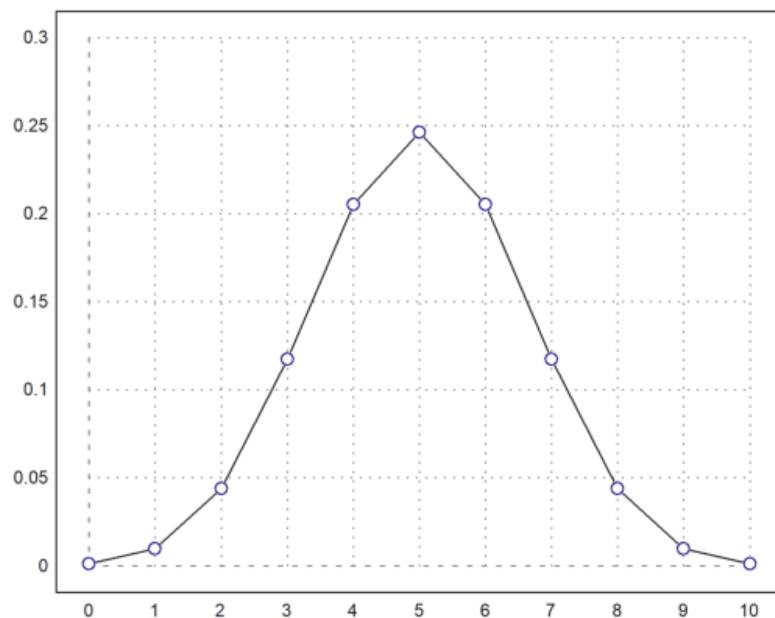


Fungsi statplot() mengatur gaya dengan string sederhana.

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



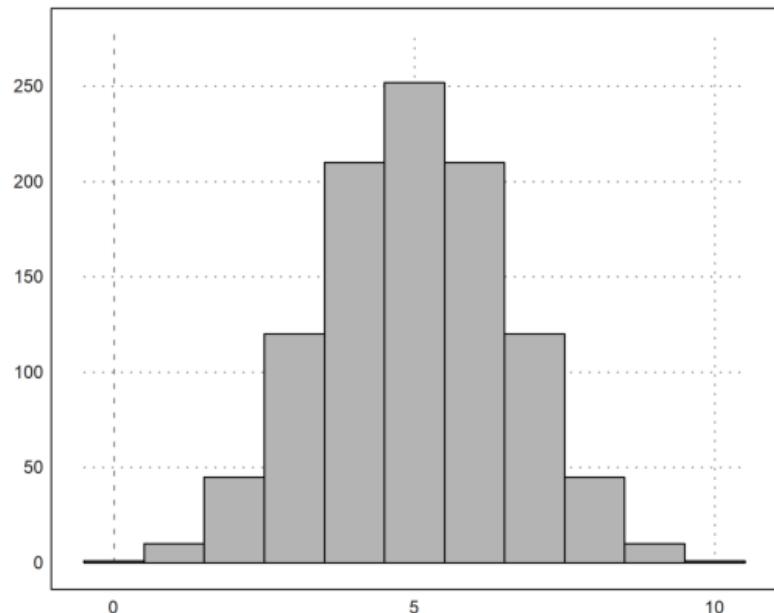
```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



Selain itu, data dapat diplot sebagai diagram batang. Dalam hal ini, x harus diurutkan dan satu elemen lebih panjang dari y. Diagram batangnya akan memanjang dari $x[i]$ hingga $x[i+1]$ dengan nilai $y[i]$. Jika x berukuran sama dengan y, maka x akan diperpanjang satu elemen dengan spasi terakhir.

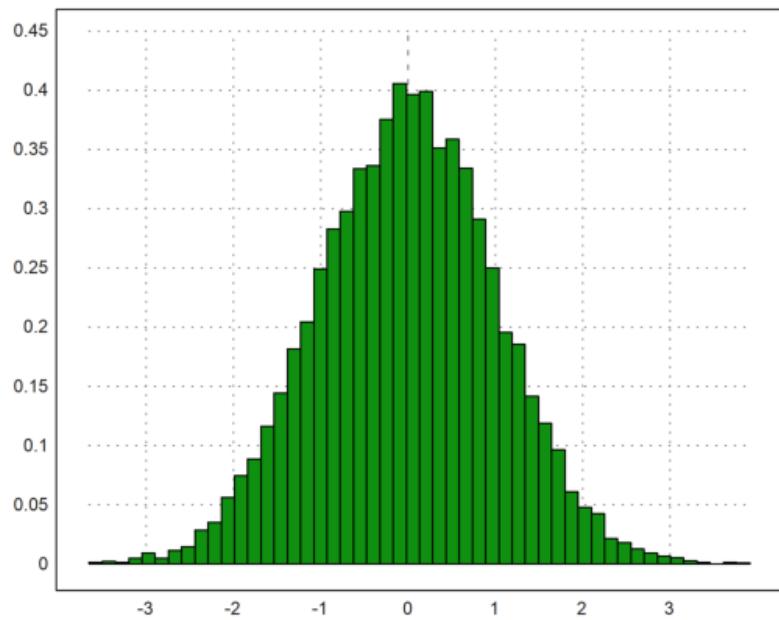
Gaya isian dapat digunakan seperti di atas.

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

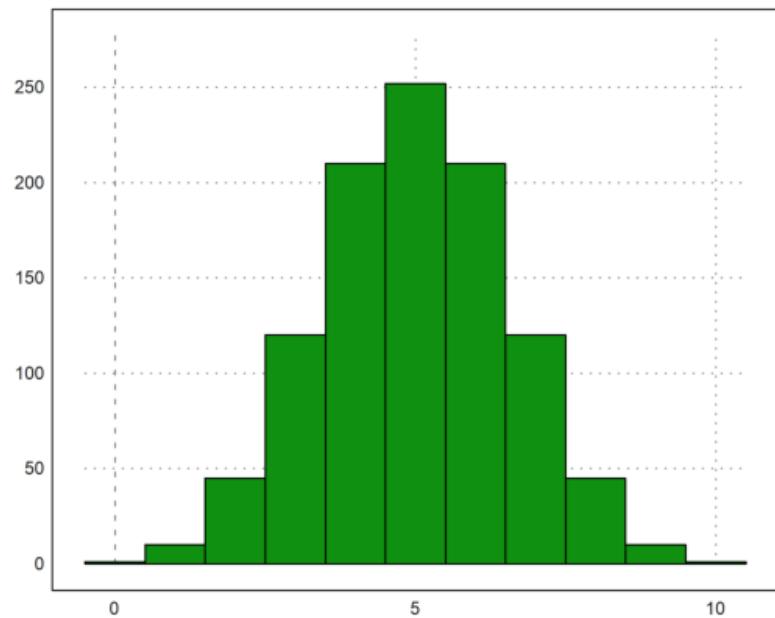


Data untuk plot batang (batang=1) dan histogram (histogram=1) dapat diberikan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribution (atau distribusi=n). Histogram nilai xv akan dihitung secara otomatis dengan >histogram. Jika >even ditentukan, nilai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

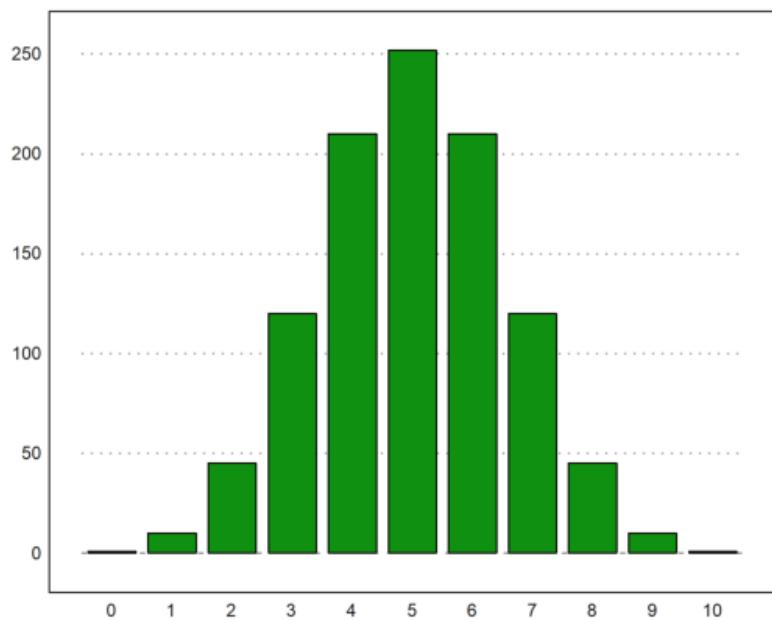
```
>plot2d(normal(10000),distribution=50):
```



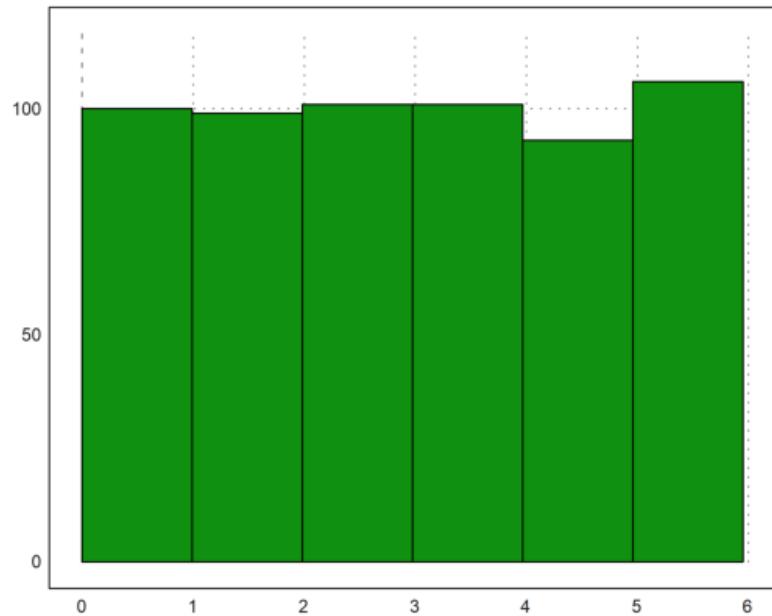
```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```



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

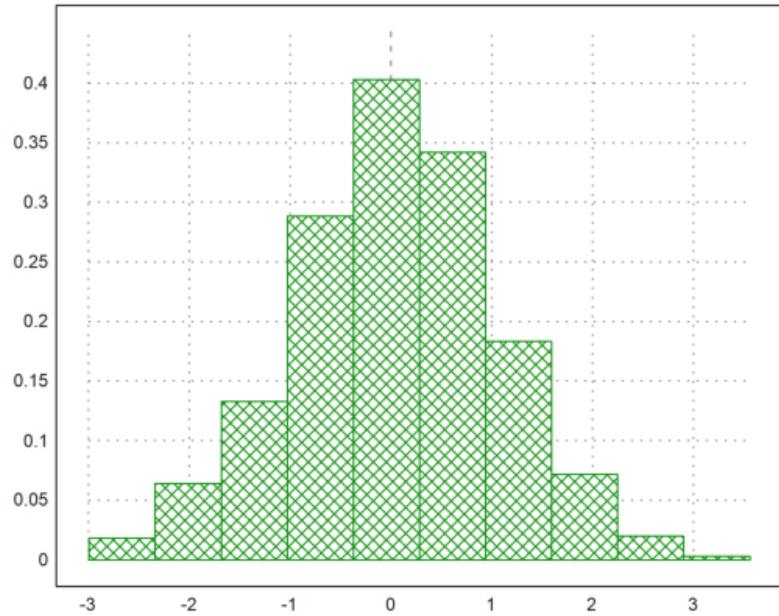


```
>plot2d(random(600)*6,histogram=6) :
```



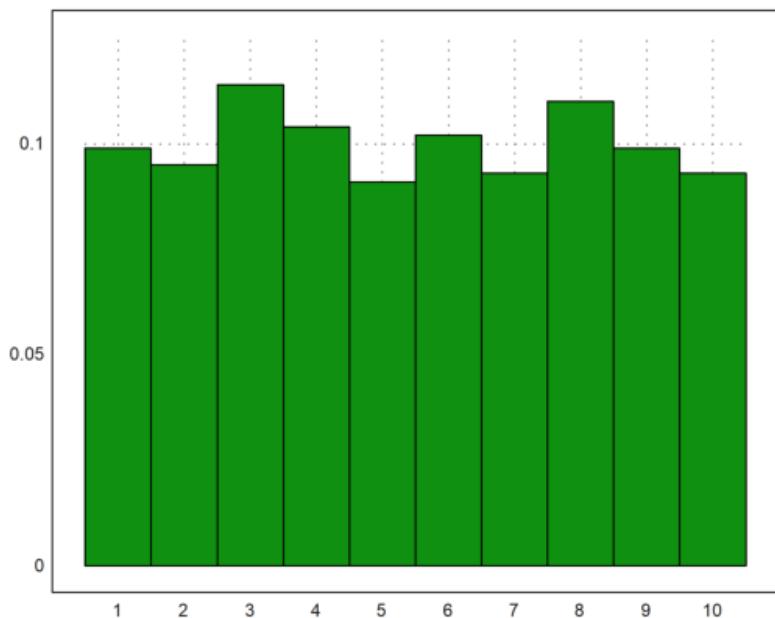
Untuk distribusi, terdapat parameter `distribution=n`, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

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



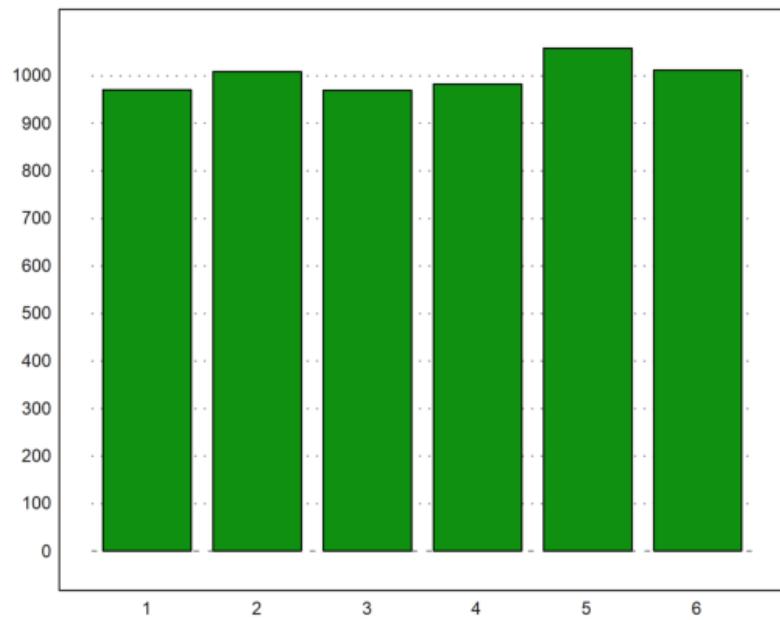
Dengan parameter even=true, ini akan menggunakan interval bilangan bulat.

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

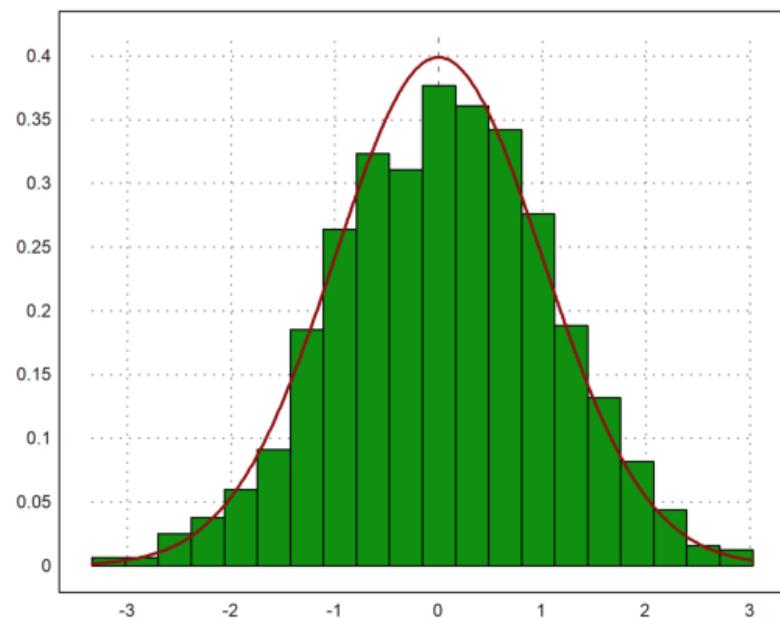


Perhatikan bahwa ada banyak plot statistik yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

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

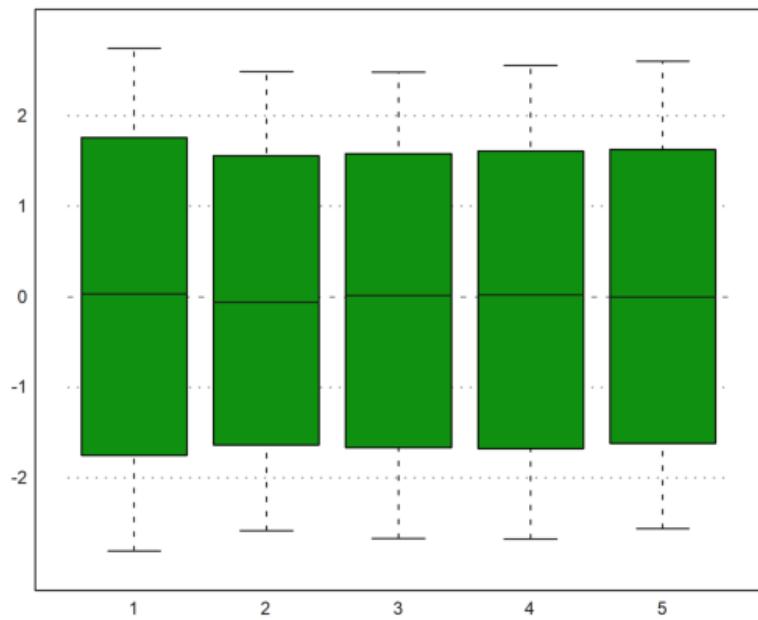


```
>plot2d(normal(1,1000),>distribution); ...
> plot2d("qnormal(x)",color=red,thickness=2,>add):
```



Ada juga banyak plot khusus untuk statistik. Plot kotak menunjukkan kuartil distribusi ini dan banyak outlier. Menurut definisinya, outlier dalam plot kotak adalah data yang melebihi 1,5 kali rentang 50% tengah plot.

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



Fungsi Implisit

Plot implisit menunjukkan penyelesaian garis level $f(x,y)=\text{level}$, dengan "level" dapat berupa nilai tunggal atau vektor nilai. Jika level = "auto", akan ada garis level nc, yang akan tersebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan >hue untuk menunjukkan nilai fungsi. Untuk fungsi implisit, xv harus berupa fungsi atau ekspresi parameter x dan y, atau alternatifnya, xv dapat berupa matriks nilai.

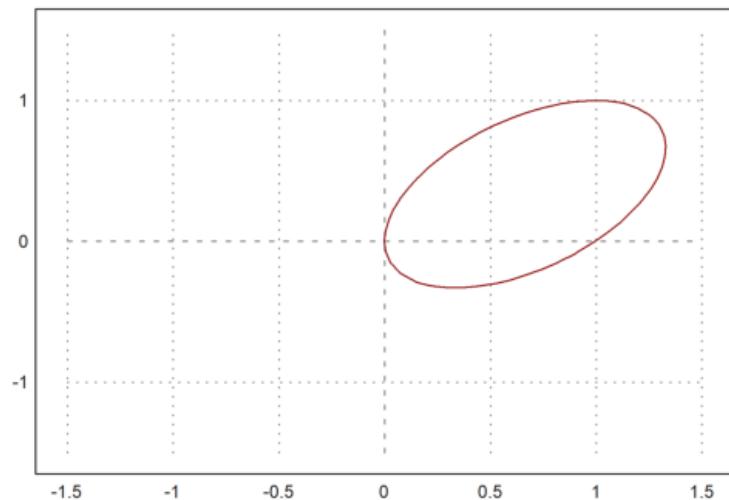
Euler dapat menandai garis level

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

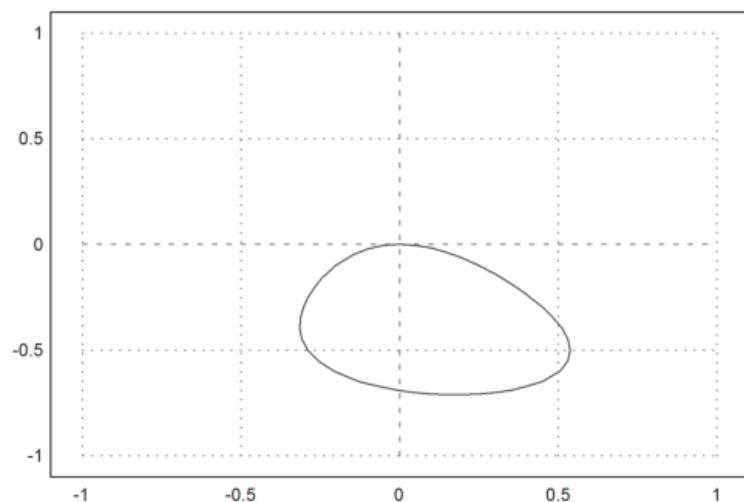
dari fungsi apa pun.

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

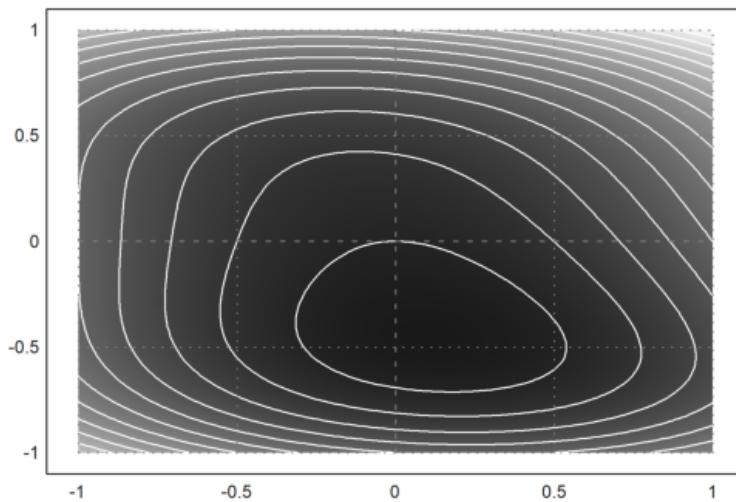
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x", r=1.5, level=0, contourcolor=red):
```



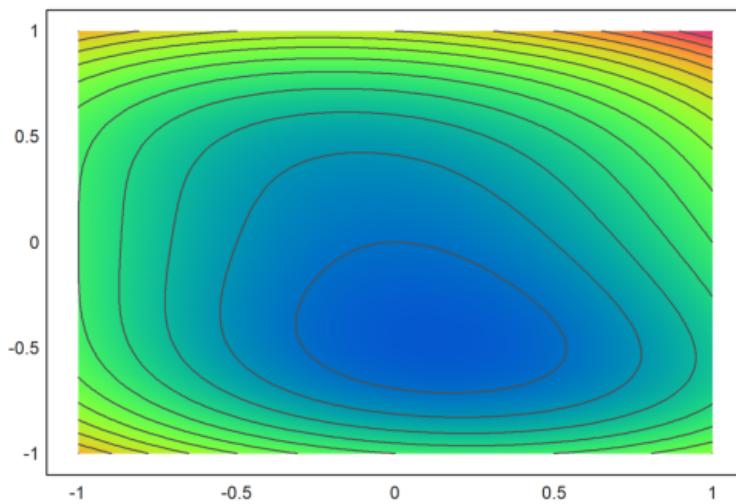
```
>expr := "2*x^2+x*y+3*y^4+y"; // definisi ekspresi f(x,y)
>plot2d(expr,level=0); // Solusi dari f(x,y)=0
```



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

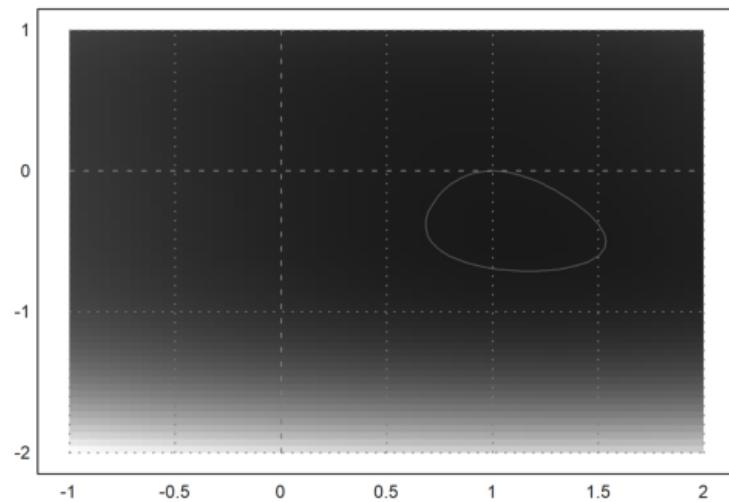


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // lebih bagus
```

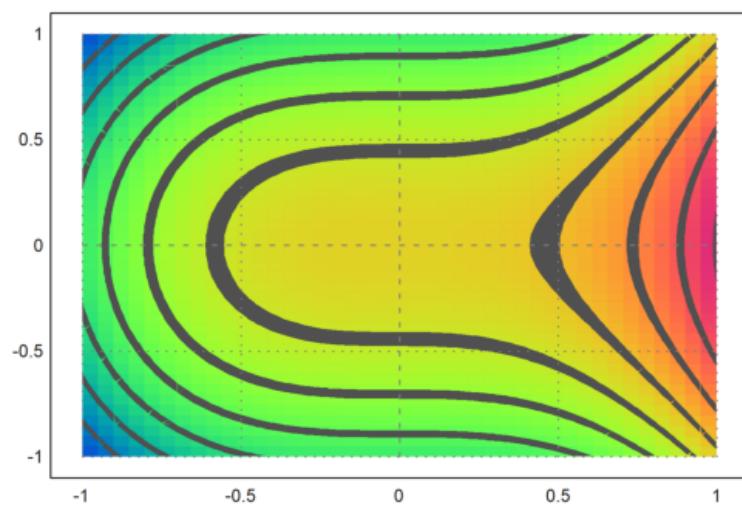


Ini juga berfungsi untuk plot data. Namun Anda harus menentukan rentang untuk label sumbu.

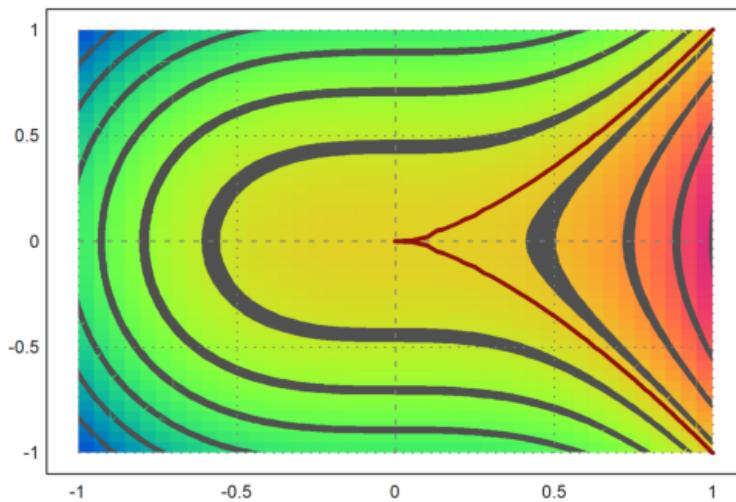
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



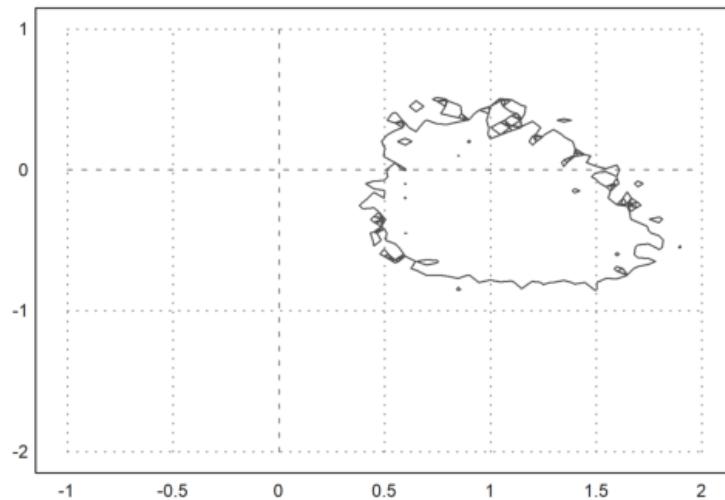
```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



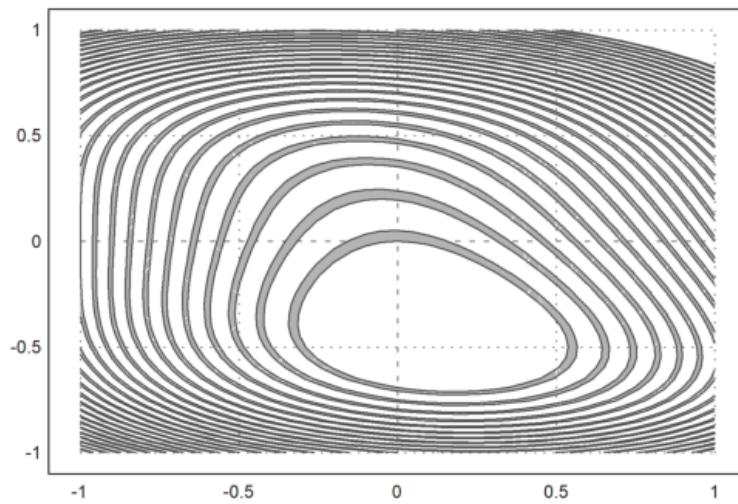
```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



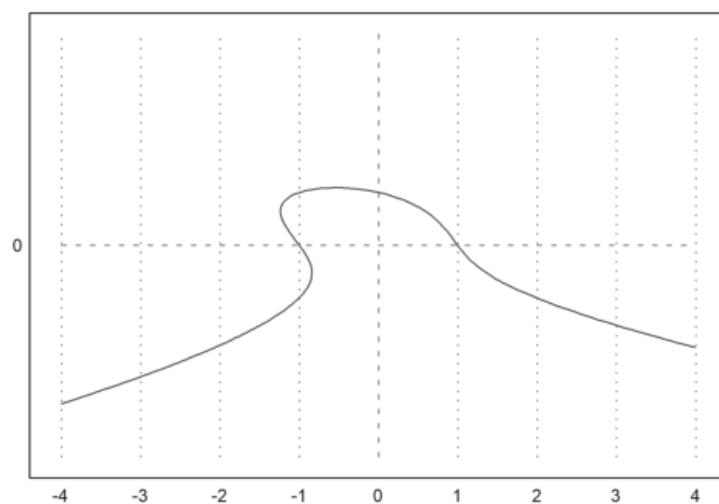
```
>z=z+normal(size(z))*0.2;  
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



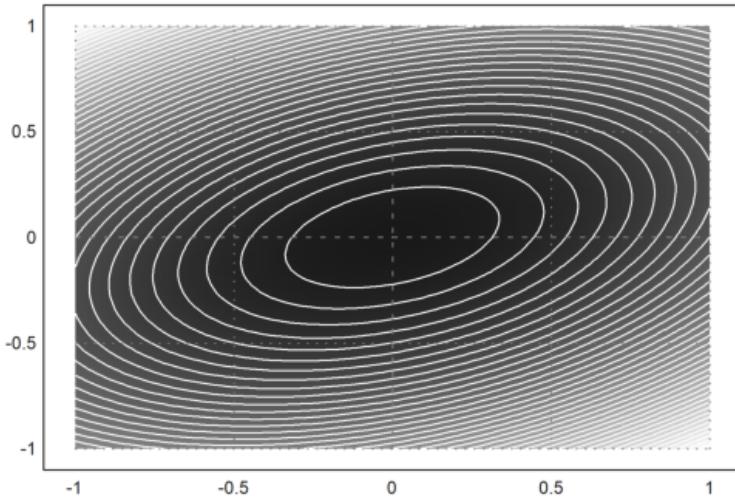
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



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



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



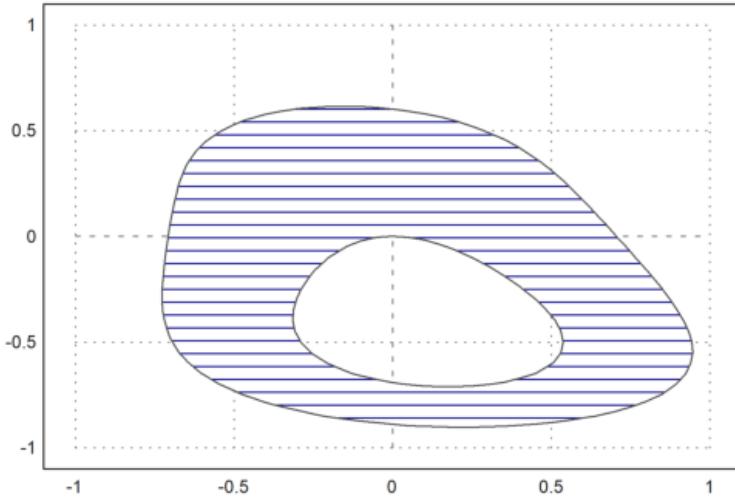
Itu dimungkinkan juga untuk mengisi set

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

dengan rentang level.

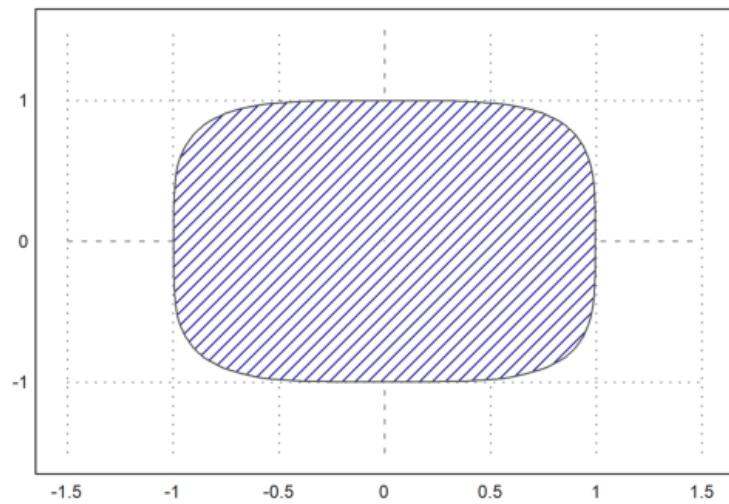
Dimungkinkan untuk mengisi daerah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

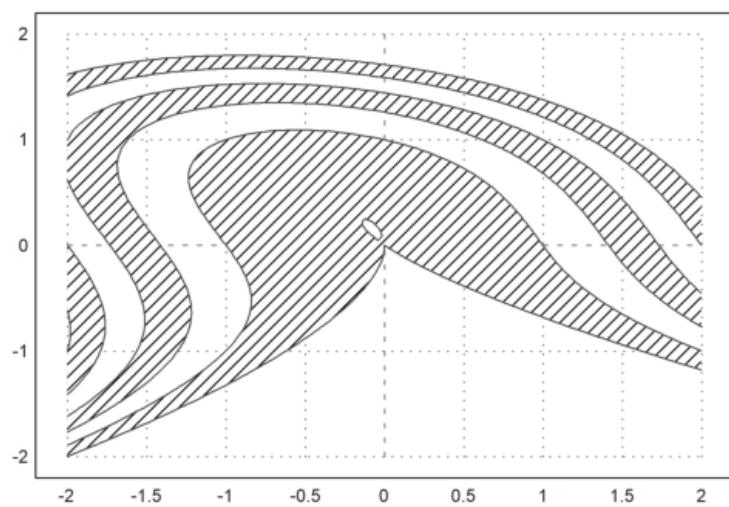


Plot implisit juga dapat menunjukkan rentang level. Maka level harus berupa matriks interval level 2xn, di mana baris pertama berisi awal dan baris kedua berisi akhir setiap interval. Alternatifnya, vektor baris sederhana dapat digunakan untuk level, dan parameter memperluas nilai level ke interval.

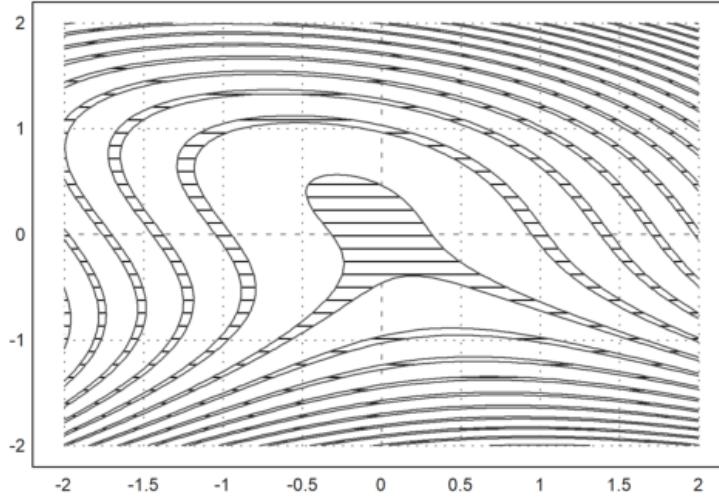
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



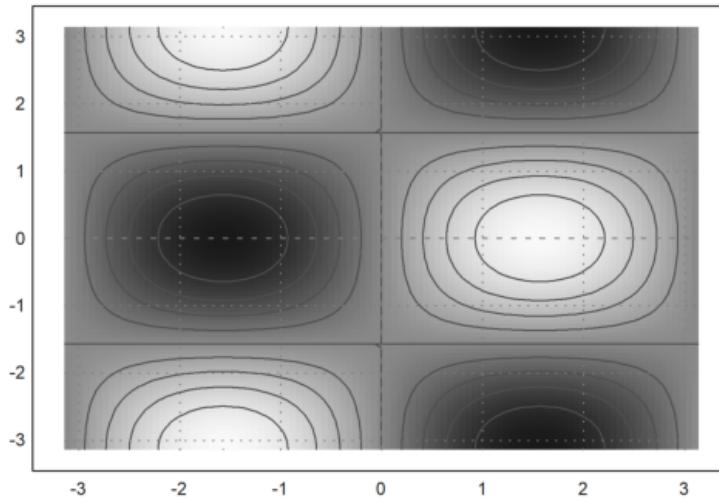
```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```



```
>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):
```

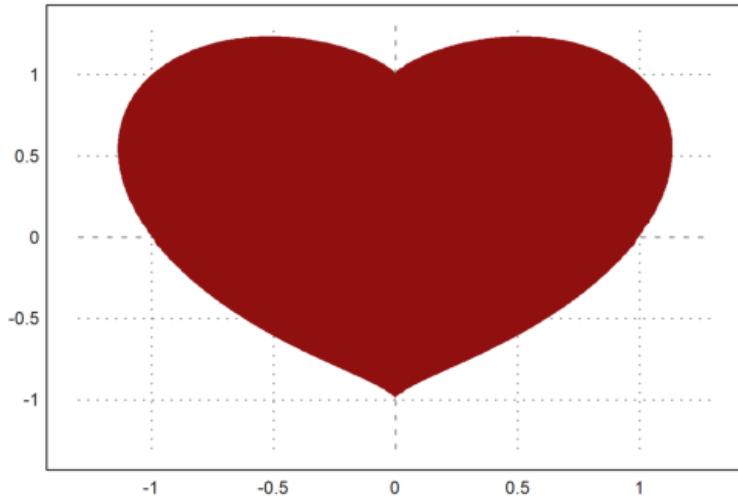


Itu dimungkinkan juga untuk menandai suatu daerah

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

Hal ini dilakukan dengan menambahkan level dengan dua baris.

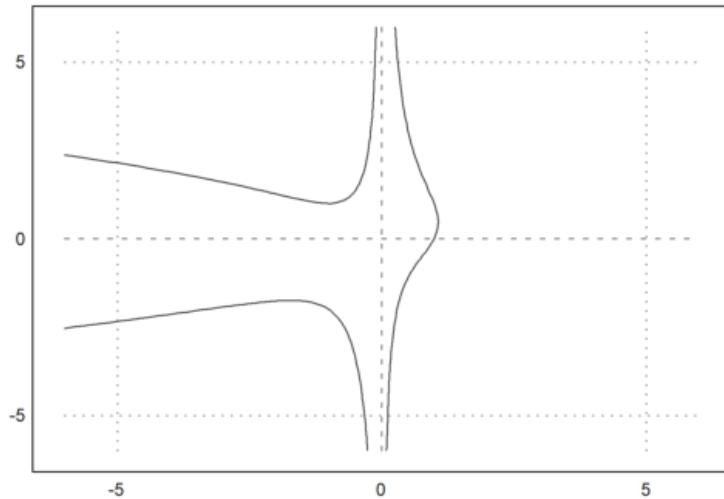
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3", r=1..3, ...
> style="#", color=red, <outline, ...
> level=[-2;0], n=100):
```



Itu dimungkinkan untuk menentukan level tertentu. Misalnya, kita dapat memplot solusi 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);
```

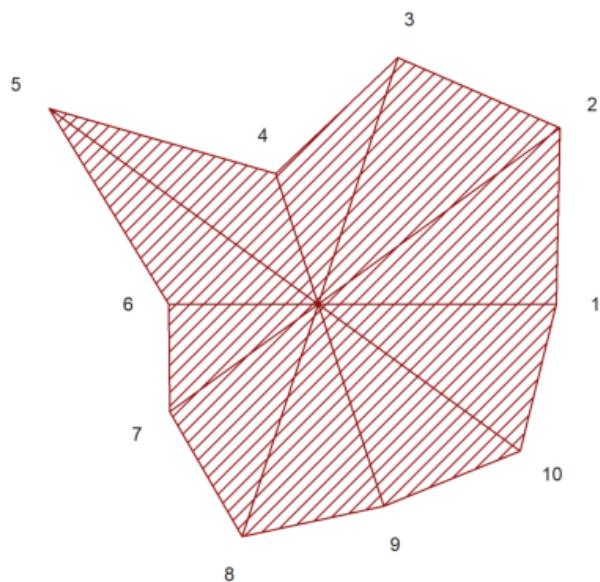
```

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 tanda ticks kotak atau sumbu di sini. Selain itu, kita menggunakan jendela penuh untuk plotnya. Kita memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Ini tidak perlu dilakukan jika Anda yakin plot Anda berhasil.

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



Terkadang, Anda mungkin ingin merencanakan sesuatu yang plot2d tidak bisa lakukan, tapi hampir berhasil. Dalam fungsi berikut, kita membuat plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

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

```

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

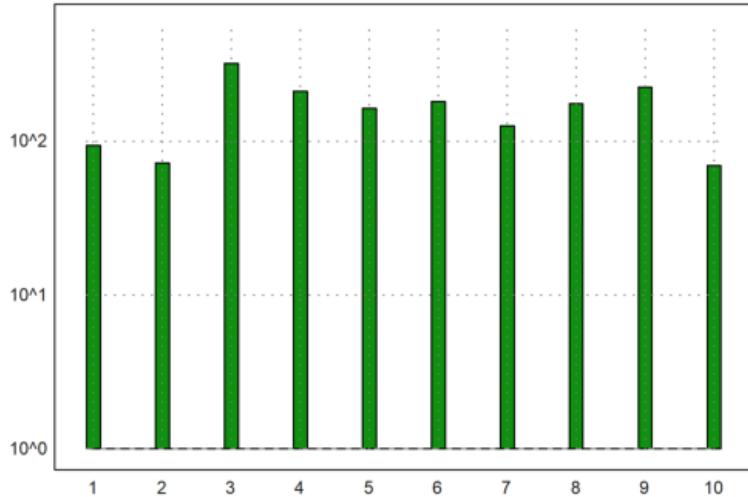
```

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

```

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

```



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

Fungsi wait(0) memaksa plot muncul di jendela grafis. Jika tidak, pengundian ulang akan dilakukan dalam interval waktu yang jarang.

```

>function animliss (n,m) ...

```

```

t=linspace(0,2pi,500);
f=0;
c=framecolor(0);
l=linewidth(2);
setplot(-1,1,-1,1);
repeat
  clg;
  plot(sin(n*t),cos(m*t+f));
  wait(0);

```

```
if testkey() then break; endif;
f=f+0.02;
end;
framecolor(c);
linewidth(l);
endfunction
```

Tekan tombol apa saja untuk menghentikan animasi ini.

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

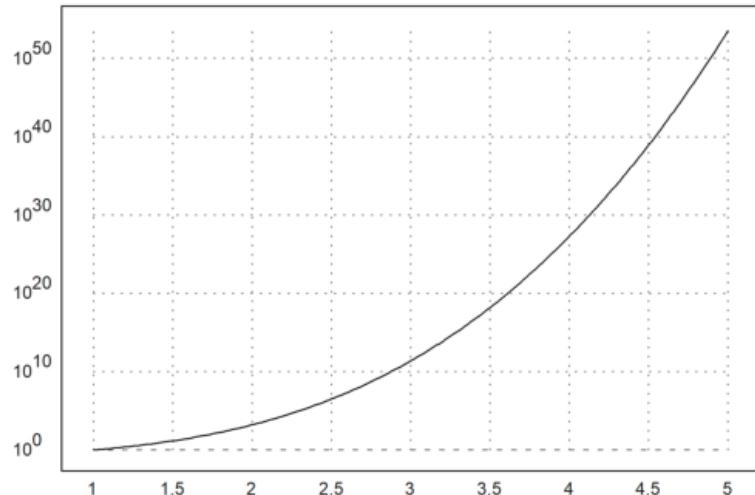
Plot Logaritmik

EMT menggunakan parameter "logplot" untuk skala logaritmik.

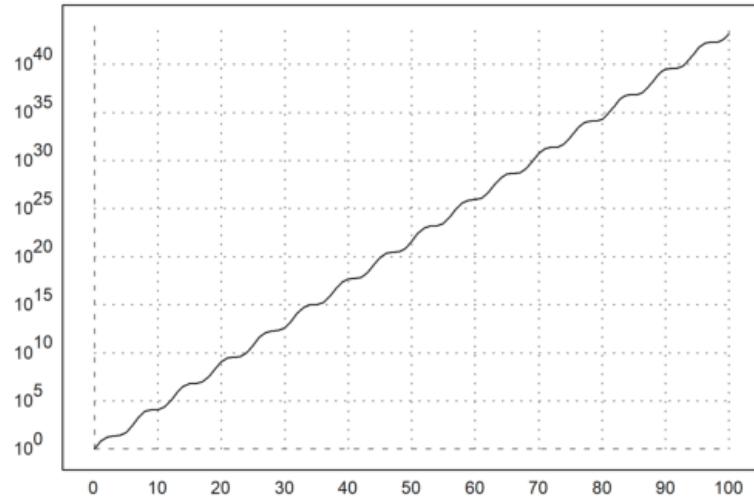
Plot logaritma dapat diplot menggunakan skala logaritma di y dengan logplot=1, atau menggunakan skala logaritma di x dan y dengan logplot=2, atau di x dengan logplot=3.

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

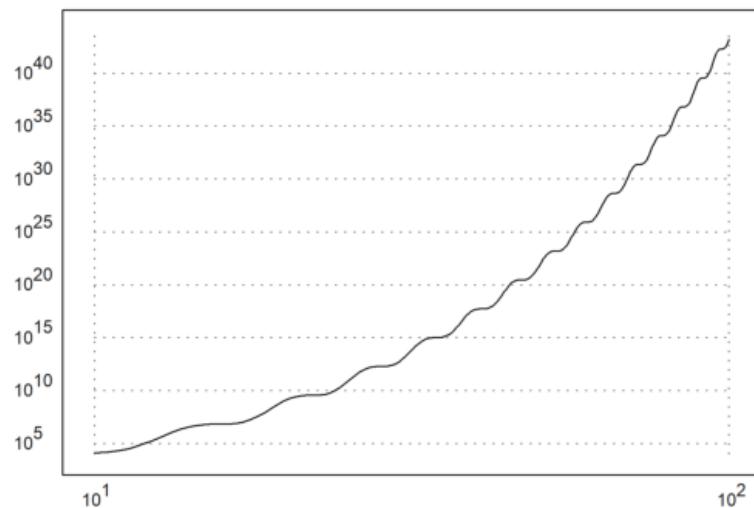
```
>plot2d("exp(x^3-x)*x^2", 1, 5, logplot=1):
```



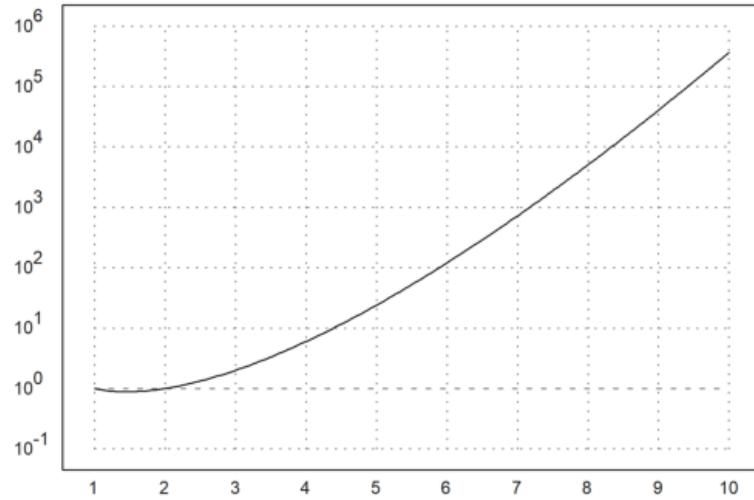
```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



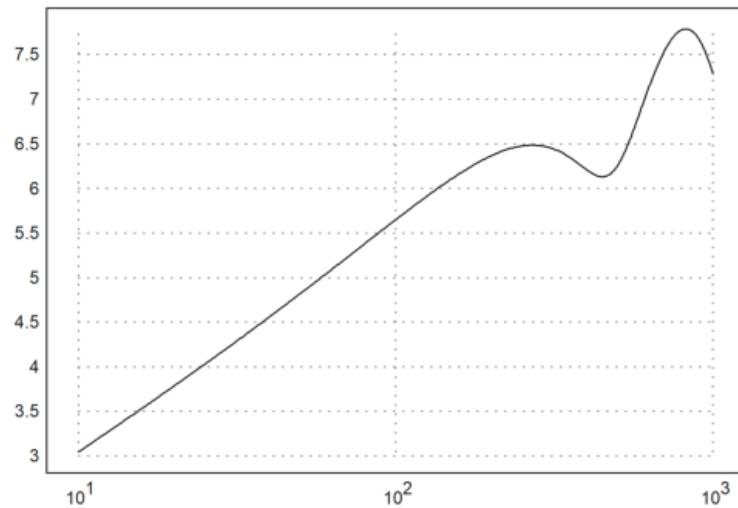
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```



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

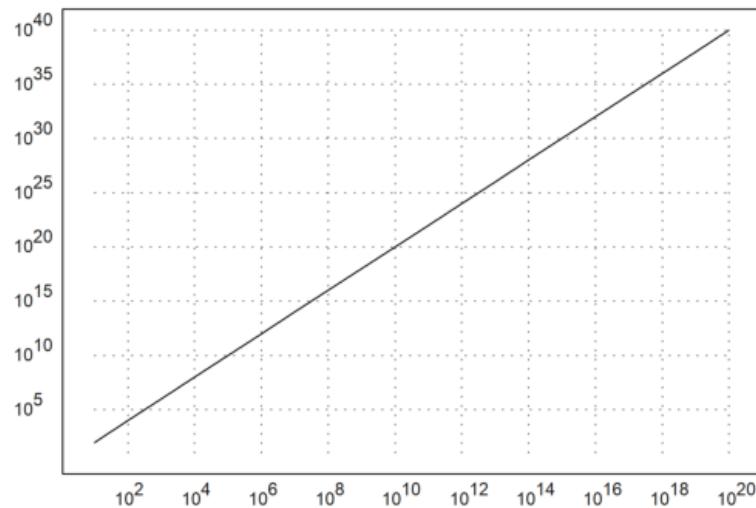


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Ini juga berfungsi dengan plot data.

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

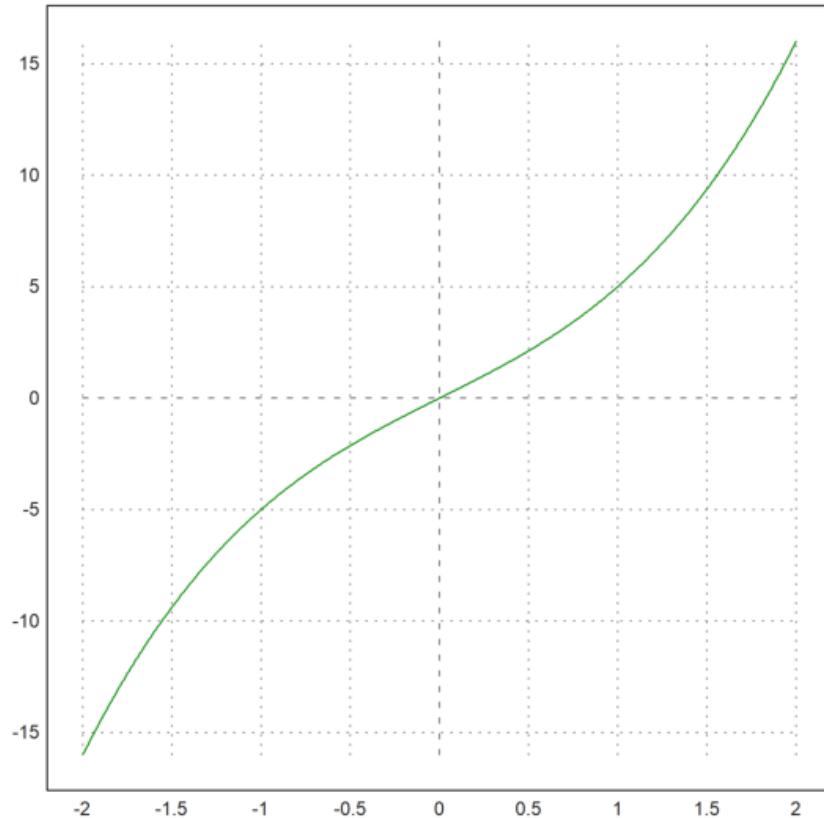


Penyelesaian Penggunaan plot2d

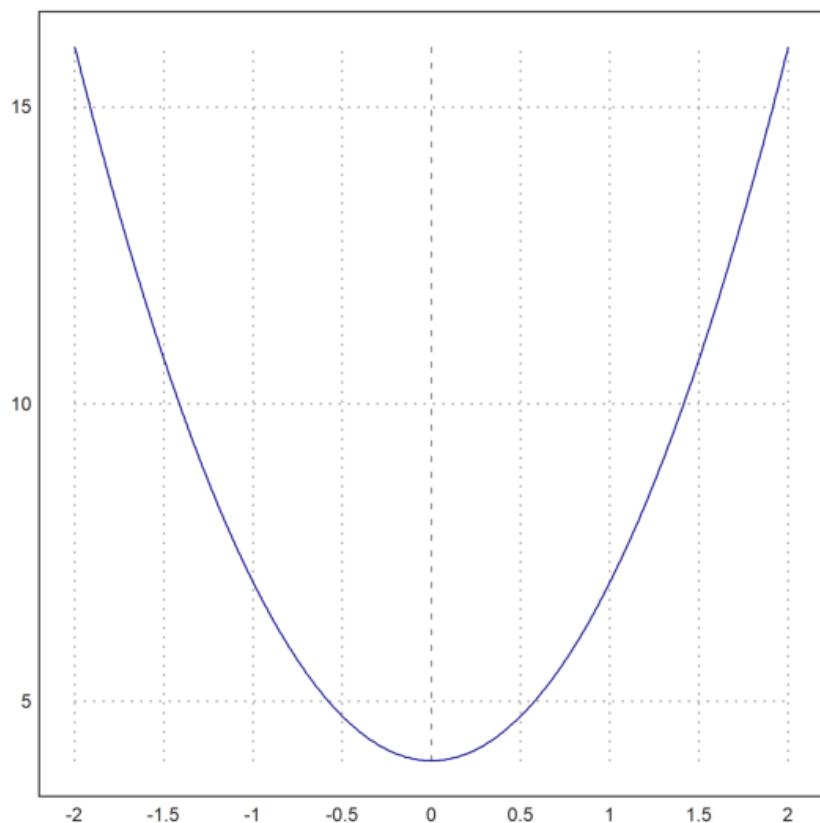
1. Gambarkan grafik fungsi dibawah ini beserta turunannya

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

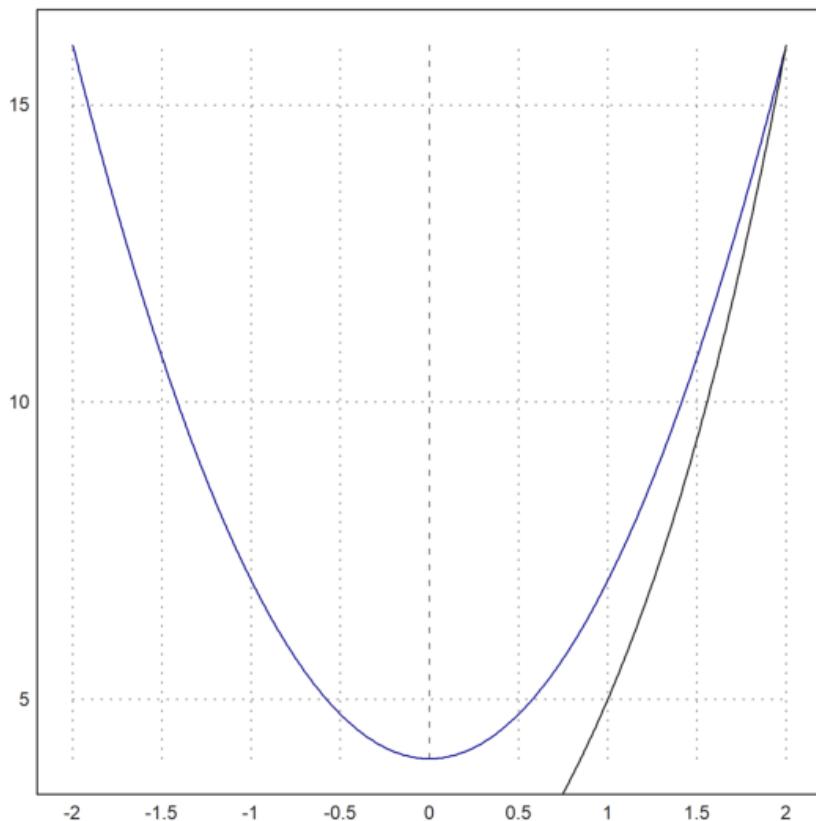
```
>reset();
>function f(x) &=x^3+4*x;
>plot2d(&f(x),color=green); // menggambar grafik fungsi f(x)
```



```
>plot2d(&diff(f(x),x),color=blue): // menggambar grafik fungsi f'(x)
```



```
>plot2d(&f(x),>add):
```

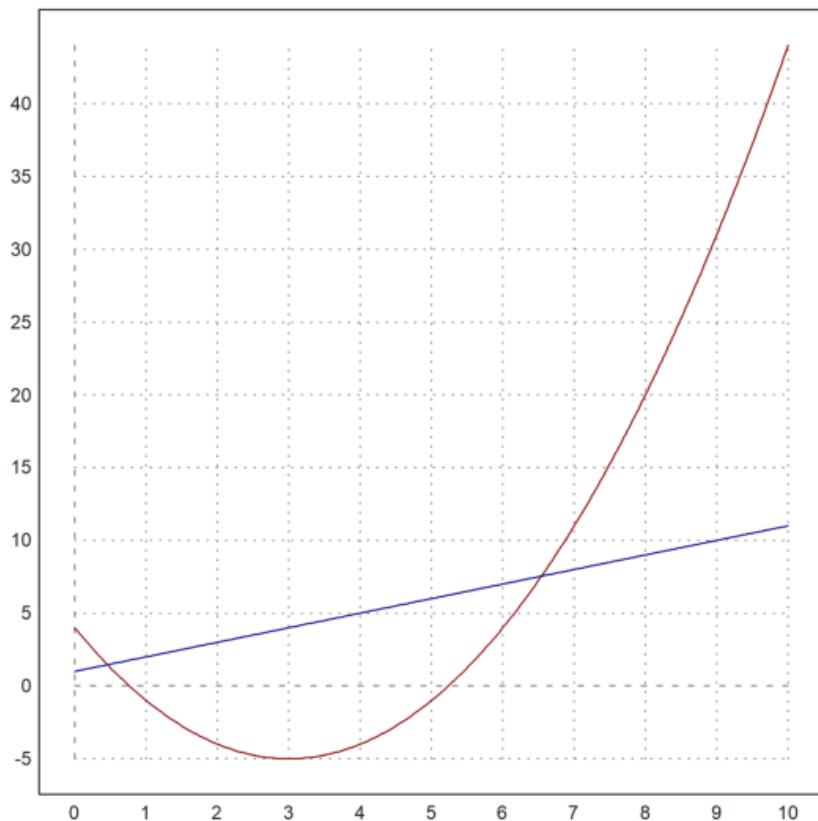


2. Gambarkan kedua fungsi berikut dan tentukan titik potong kedua grafik tersebut.

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

$$g(x) = x + 1$$

```
>reset();
>function f(x) &=x^2-6*x+4; // mendefinisikan fungsi f(x)
>function g(x) &=x+1; // mendefinisikan fungsi g(x)
>plot2d([&f(x),&g(x)],0,10,color=[red,blue]): // menggambar plot2d dari fungsi f(x) dan g(x)
```



Mencari titik potong dari kedua fungsi di atas.

```
>x1= solve("g(x)-f(x)", 0)
```

```
0.458618734851
```

```
>f(x1)
```

```
1.45861873485
```

```
>g(x1)
```

```
1.45861873485
```

Sehingga titik potong pertama di titik $(0.458, 1.458)$.

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

```
6.54138126515
```

```
>f(x2)
```

7.54138126515

```
>g(x2)
```

7.54138126515

Sehingga titik potong kedua berada di titik (6.541, 7.541)

Jadi, titik perpotongan fungsi $f(x)$ dan $g(x)$ tersebut adalah (0.458, 1.458) dan (6.541, 7.541).

3. Sebuah perusahaan memproduksi dua jenis produk, yaitu produk A dan produk B. Perusahaan tersebut ingin mengoptimalkan keuntungan dari penjualan kedua produk ini. Produk A memberikan keuntungan 2 ribu per unit, sedangkan produk B memberikan keuntungan 4 ribu per unit. Perusahaan memiliki dua jenis sumber daya yang terbatas, yaitu waktu produksi dan bahan baku. Waktu produksi maksimal adalah 4 jam per minggu, dan bahan baku yang tersedia adalah 12 kilogram per minggu. Tentukan berapa banyak produk A dan produk B yang harus diproduksi oleh perusahaan tersebut untuk memaksimalkan keuntungan mereka.

Penyelesaian:

- Variabel Keputusan

x : Jumlah unit produk A yang diproduksi per minggu.

y : Jumlah unit produk B yang diproduksi per minggu.

- Fungsi Tujuan

Memaksimalkan keuntungan per minggu: $f(x, y) = 2x + 4y$

- Kendala

$$x + y \leq 4$$

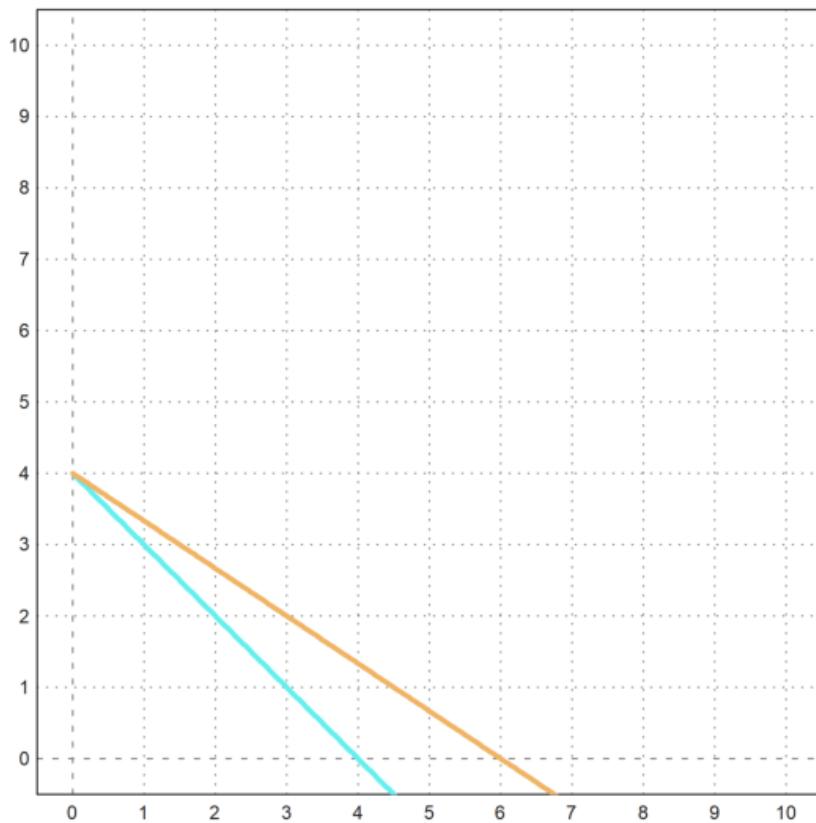
$$2x + 3y \leq 12$$

$$x \geq 0$$

$$y \geq 0$$

Selanjutnya akan digambar grafik dari kendala di atas

```
>reset();  
>plot2d(["(4-x)", "(12-2x)/3"] ...  
, 0, 10, 0, 10, color=13:16, thickness=3, grid=5, ):
```



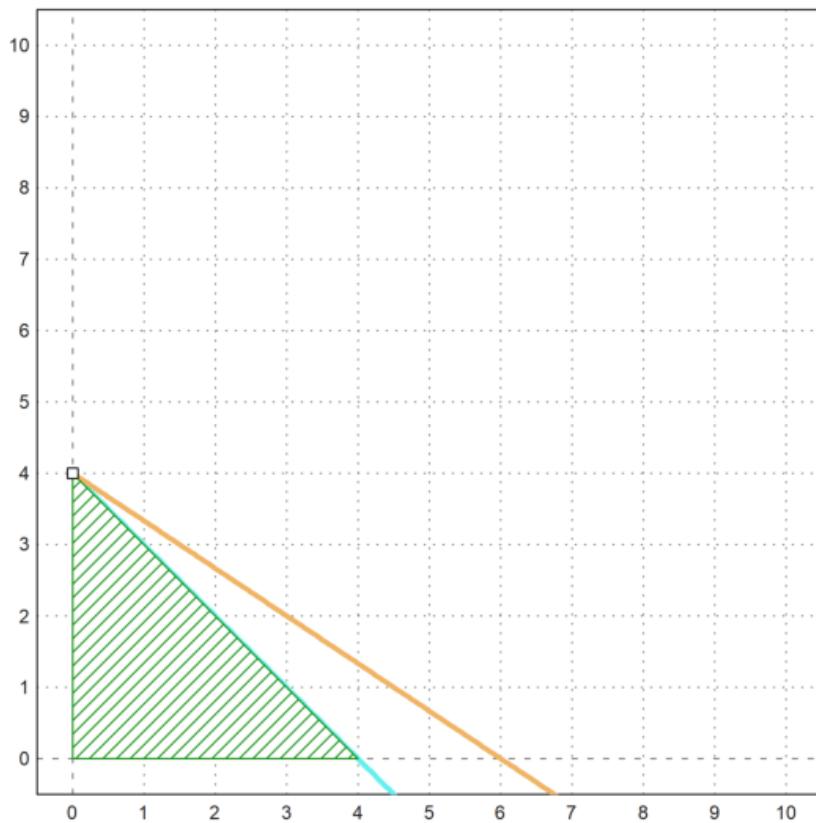
Kemudian kendala dibuat kedalam bentuk matriks koefisien

```
>A=[1,1;2,3]; b=[4;12];
>x=simplex(A,b,[2,4],>max); fraction x
```

```
0
4
```

Metode simpleks untuk maksimisasi atau minimisasi fungsi linier fungsi dengan kendala linier, dan variabel yang dibatasi atau tidak dibatasi variabel.

```
>xa=feasibleArea(A,b);
>plot2d(xa[1],xa[2],>filled,style="/" ,a=0,b=10,c=0,d=10,>add);
>plot2d(x[1],x[2],>add,>points):
```



Sehingga substitusikan saja titik (0,4) ke dalam fungsi tujuan maka ditemukan nilai optimal

$$2x + 4y$$

$$2 * 0 + 4 * 4$$

16

Jadi maksimal perusahaan memperoleh keuntungan adalah 16 ribu dan produk A diproduksi sebanyak 8 unit dan produk B diproduksi sebanyak 4 unit.

```
>reset();
```

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 of 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 for 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

PEKAN 7-8: PENGGUNAAN SOFTWARE EMT UNTUK PLOT 3D

article

eumat

Menggambar Plot 3D dengan EMT

Ini adalah pengantar tentang plot 3D dalam Euler. Kita memerlukan plot 3D untuk memvisualisasikan fungsi dua variabel.

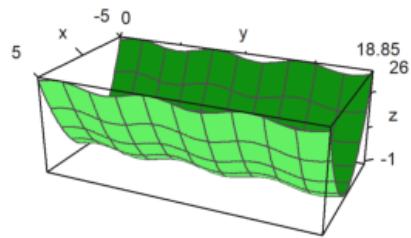
Euler menggambar fungsi-fungsi seperti itu menggunakan algoritma pengurutan untuk menyembunyikan bagian-bagian di latar belakang. Secara umum, Euler menggunakan proyeksi sentral. Defaultnya adalah dari kuadran x-y positif menuju titik pusat $x=y=z=0$, tetapi angle= 0° menghadap ke arah sumbu y. Sudut pandang dan tinggi dapat diubah.

Euler dapat membuat plot untuk

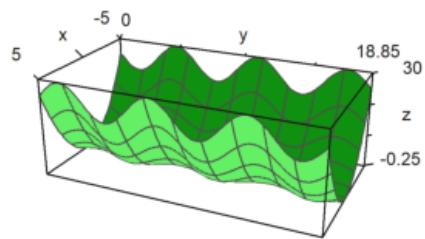
- permukaan dengan shading dan garis level atau rentang level,
- awan titik,
- kurva parametrik,
- permukaan implisit.

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):
```

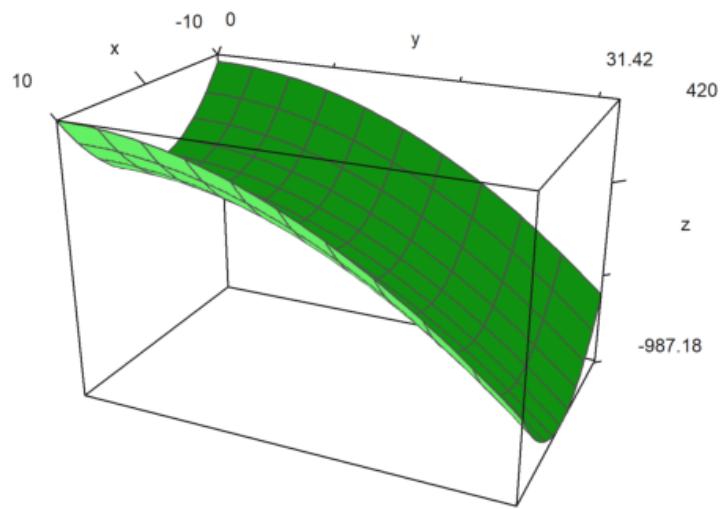


```
>plot3d("x^2+x*sin(y)", -5, 5, 0, 6*pi):
```

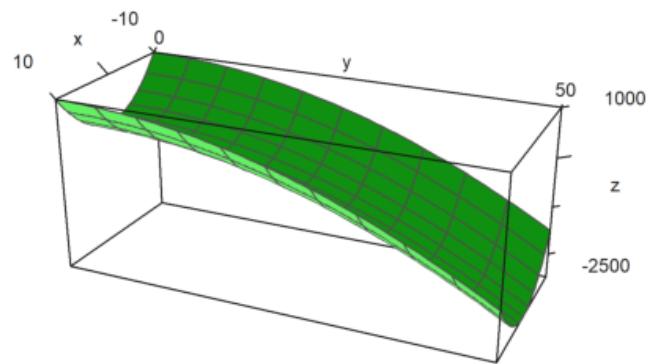


Silakan lakukan modifikasi agar gambar "talang bergelombang" tersebut tidak lurus melainkan melengkung/melingkar, baik melingkar secara mendatar maupun melingkar turun/naik (seperti papan peluncur pada kolam renang). Temukan rumusnya.

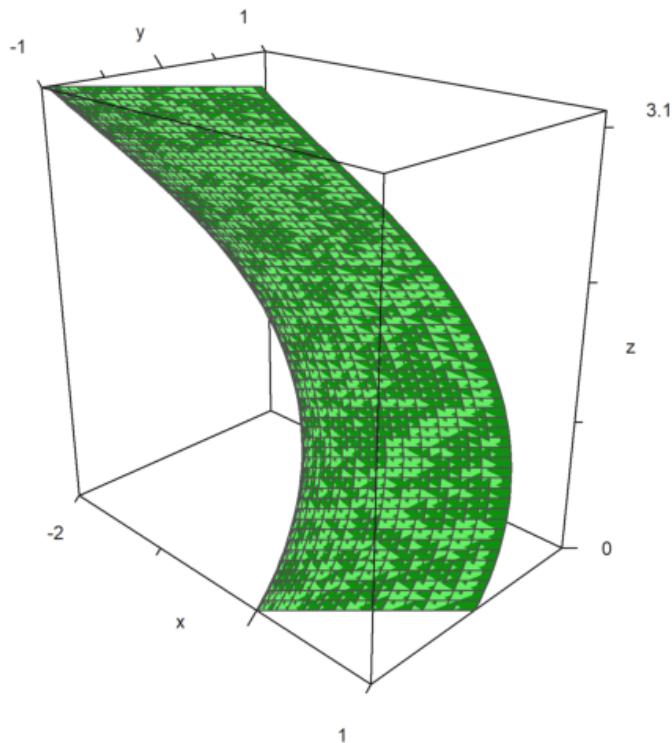
```
>aspect(); plot3d("4x^2+2*x-y^2", -10, 10, 0, 10*pi):
```



```
>aspect(); plot3d("10*x^2-y^2",-10,10,0,50):
```



```
>t=0*pi:0.05:1*pi; y=(-pi:0.1:0)'; ...
>plot3d(cos(t)+sin(y),sin(t)+sin(y),t,color=green,angle=45°):
```



Fungsi Dua Variabel

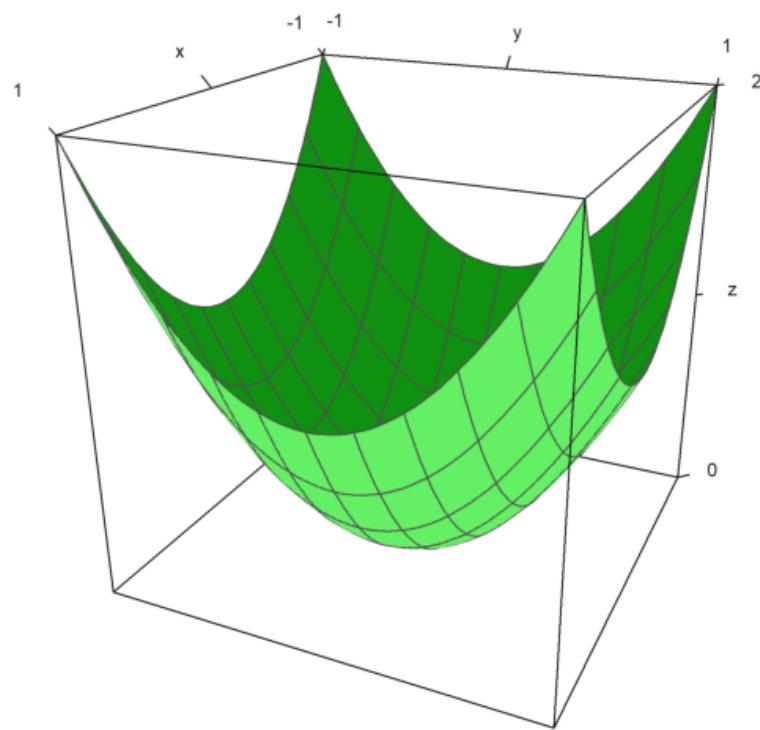
Untuk grafik dari sebuah fungsi, gunakan
 - ekspresi sederhana dalam x dan y ,
 - nama fungsi dua variabel,
 - atau matriks data.

Defaultnya adalah tampilan kisi kawat berisi dengan warna yang berbeda di kedua sisi. Perhatikan bahwa jumlah interval kisi default adalah 10, tetapi plot menggunakan jumlah persegi default 40×40 untuk membuat permukaan. Ini dapat diubah.

- $n=40, n=[40,40]$: jumlah garis kisi dalam setiap arah
- $grid=10, grid=[10,10]$: jumlah garis kisi dalam setiap arah.

Kami menggunakan nilai default $n=40$ dan $grid=10$.

```
>plot3d("x^2+y^2") :
```

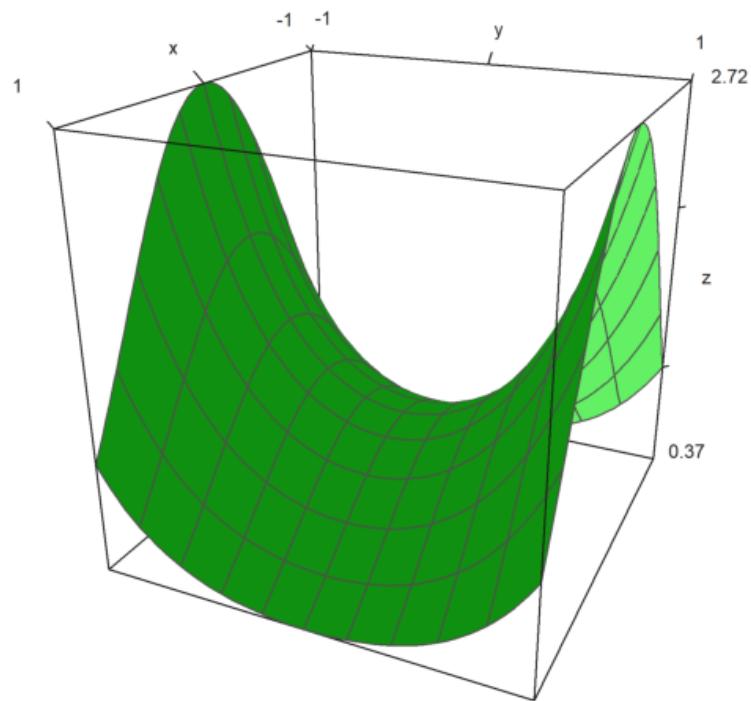


Interaksi pengguna dimungkinkan dengan parameter >user. Pengguna dapat menekan tombol-tombol berikut.

- kiri, kanan, atas, bawah: mengubah sudut pandang
- +, -: memperbesar atau memperkecil
- a: membuat gambar anaglyph (lihat di bawah)
- l: mengaktifkan atau menonaktifkan sumber cahaya (lihat di bawah)
- spasi: mengembalikan ke pengaturan awal
- enter: mengakhiri interaksi

```
>plot3d("exp(-x^2+y^2)",>user, ...
> title="Turn with the vector keys (press return to finish)":
```

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: persegi simetris sekitar (0,0).
- n: jumlah subinterval untuk plot.

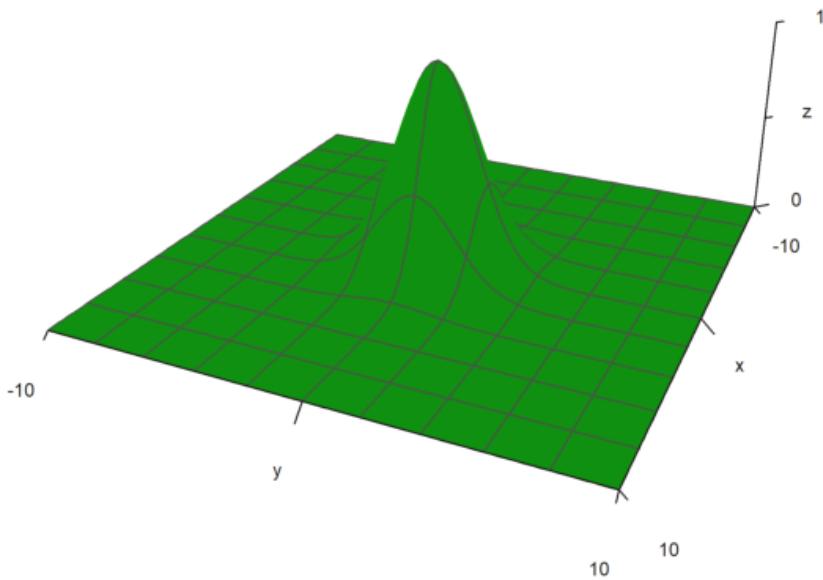
Ada beberapa parameter untuk mengubah skala fungsi atau tampilan grafik.

fscale: mengubah skala nilai-nilai fungsi (default adalah <fscale>).

skala: angka atau vektor 1x2 untuk mengubah skala pada 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):
```



Pandangan dapat diubah dengan berbagai cara yang berbeda.

- jarak: jarak pandang ke plot.
- zoom: nilai zoom.
- sudut: sudut terhadap sumbu y negatif dalam radian.
- tinggi: tinggi pandangan dalam radian.

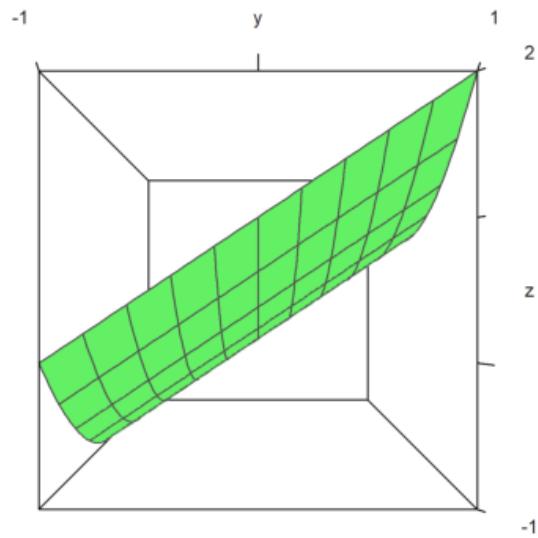
Nilai default dapat diperiksa atau diubah dengan fungsi `view()`. Fungsi ini mengembalikan parameter-parameter tersebut sesuai urutan di atas.

```
>view
```

```
[5, 2.6, 2, 0.4]
```

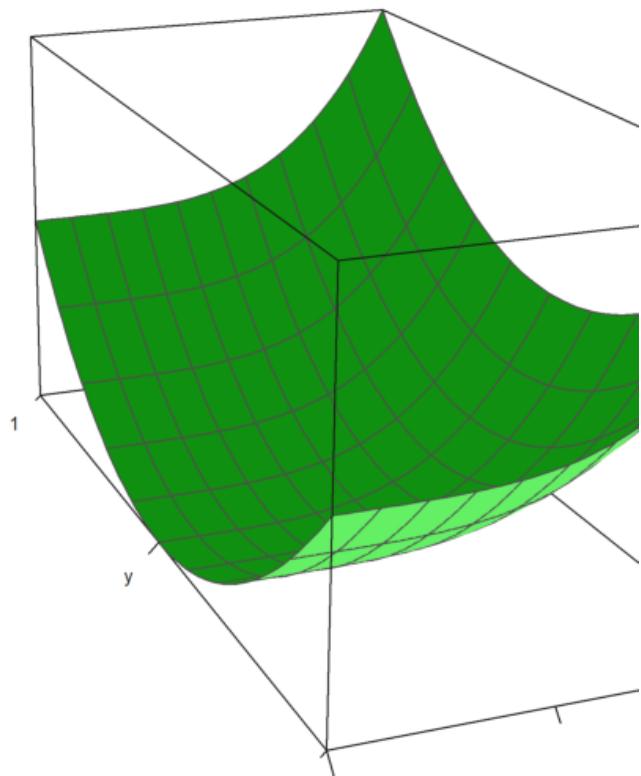
Jarak yang lebih dekat memerlukan zoom yang lebih sedikit. Efeknya lebih mirip dengan lensa sudut lebar. Pada contoh berikut, sudut=0 dan tinggi=0 dilihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^2+y", distance=3, zoom=1, angle=pi/2, height=0):
```



Plot selalu terlihat ke pusat kubus plot. Anda dapat memindahkan pusatnya dengan parameter pusat.

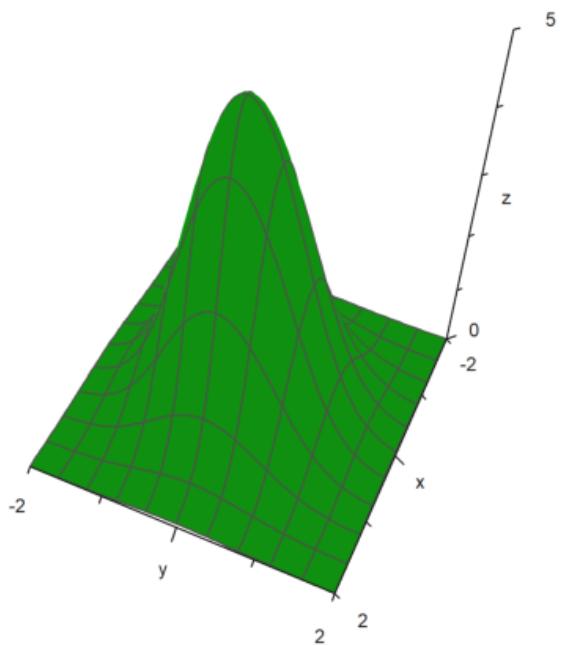
```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
>  center=[0.4,0,0],zoom=5):
```



Plot tersebut disesuaikan ukurannya agar masuk ke dalam kubus satuan untuk ditampilkan. Jadi tidak perlu mengubah jarak atau zoom tergantung pada ukuran plot. Namun, label-label mengacu pada ukuran sebenarnya.

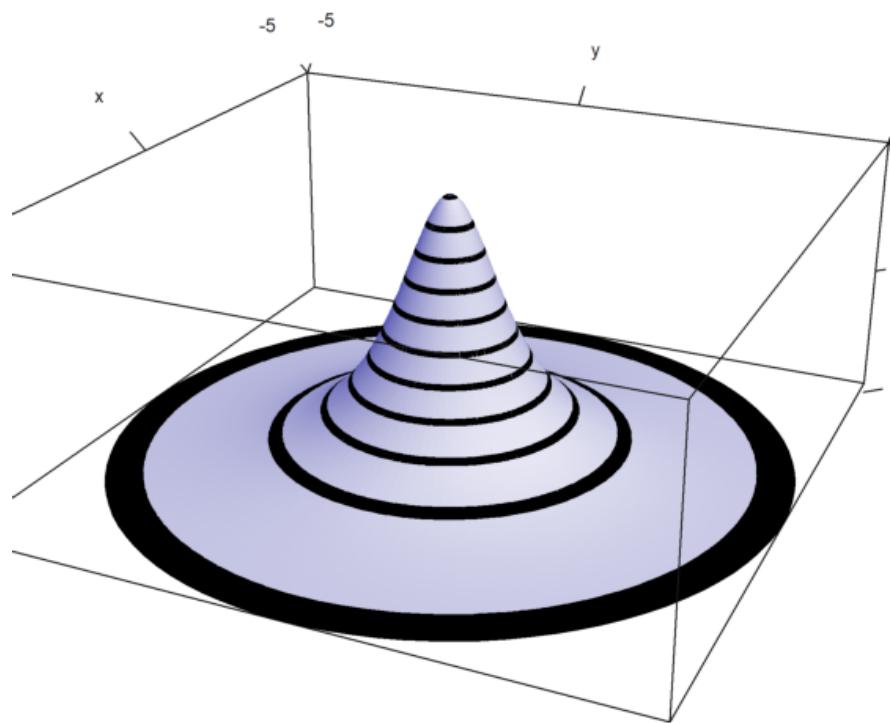
Jika Anda mematikan ini dengan `scale=false`, Anda perlu memperhatikan agar plot tetap muat ke dalam jendela plot dengan mengubah jarak tampilan atau zoom, dan memindahkan pusatnya.

```
>plot3d("5*exp(-x^2-y^2)", r=2, <fscale,<scale,distance=13,height=50°, ...
>  center=[0,0,-2], frame=3):
```

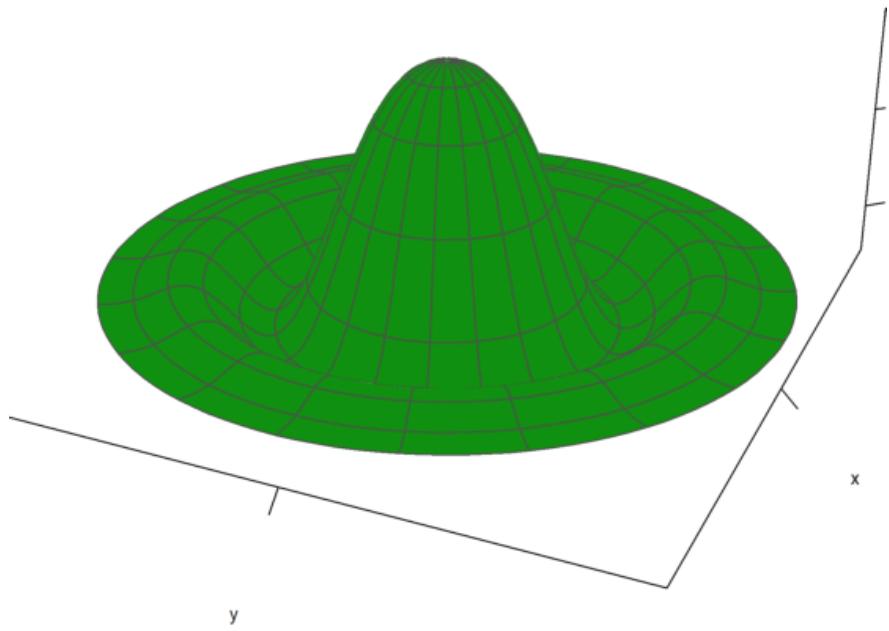


Grafik polar juga tersedia. Parameter `polar=true` menggambar grafik polar. Fungsi tetap harus menjadi fungsi dari `x` dan `y`. Parameter "fscale" mengubah skala fungsi dengan skala sendiri. Sebaliknya, fungsi tersebut akan diubah skala agar sesuai dalam sebuah kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=blue):
```



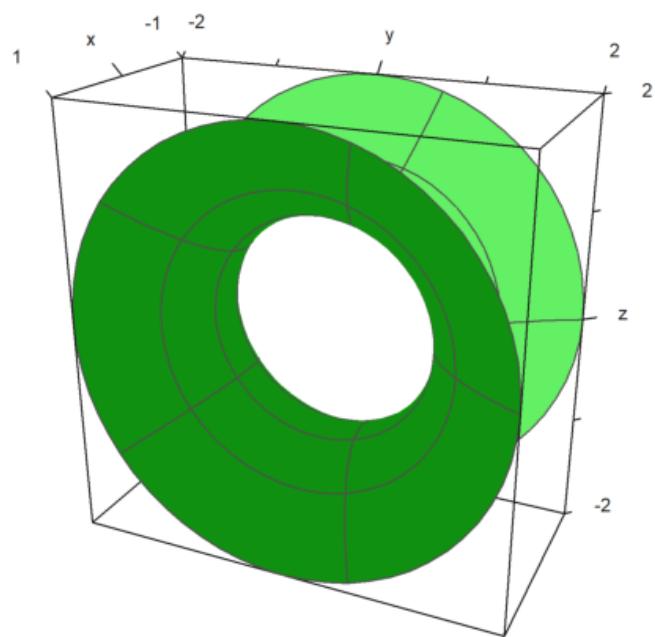
```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```



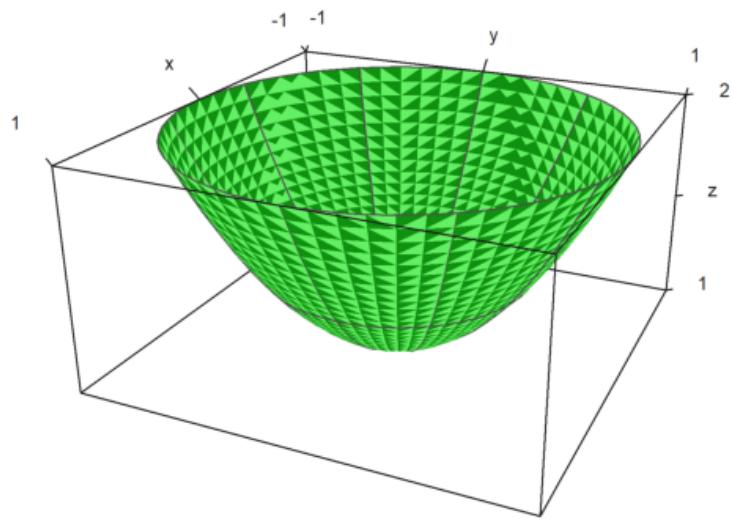
Parameter `rotate` memutar sebuah fungsi dalam sumbu x sekitar sumbu x.

- `rotate=1`: Menggunakan sumbu x
- `rotate=2`: Menggunakan sumbu z

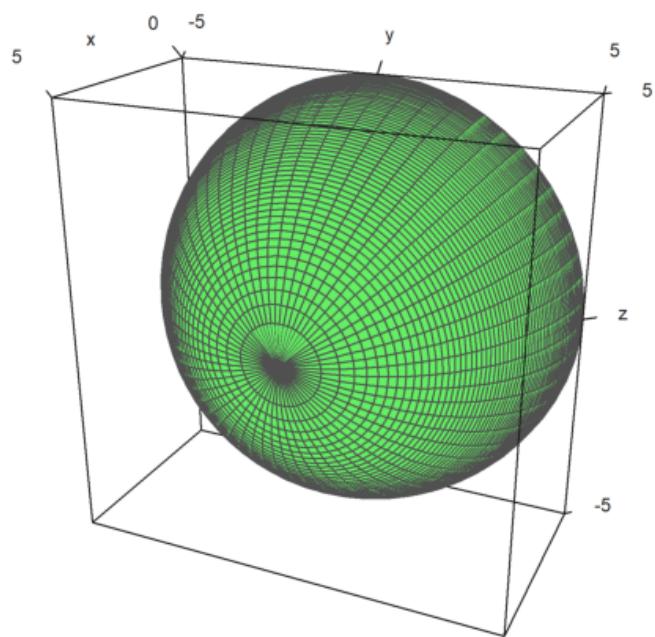
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



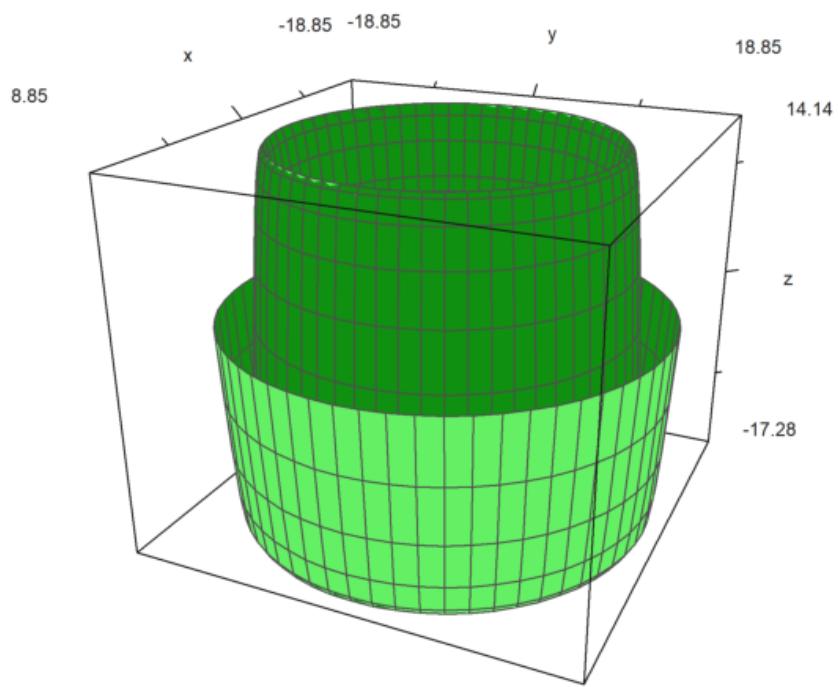
```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```



```
>plot3d("sqrt(25-x^2)", a=0, b=5, rotate=1) :
```

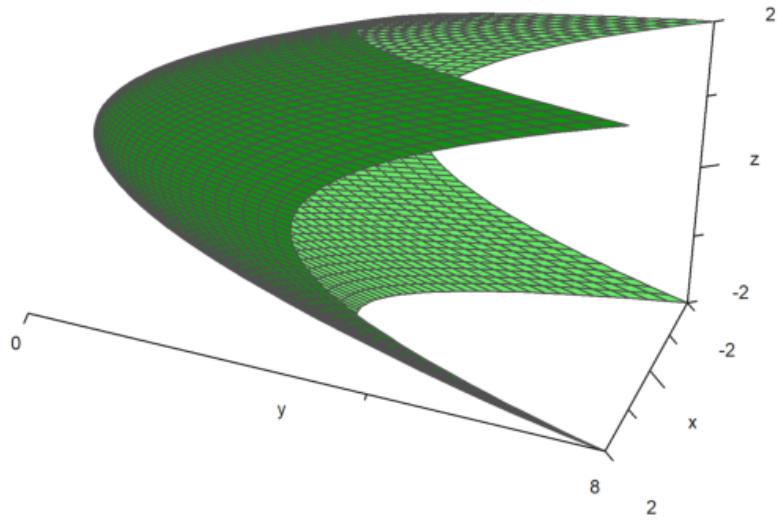


```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```



Ini adalah plot dengan tiga fungsi.

```
>plot3d("x", "x^2+y^2", "y", r=2, zoom=3.5, frame=3) :
```



Plot Kontur

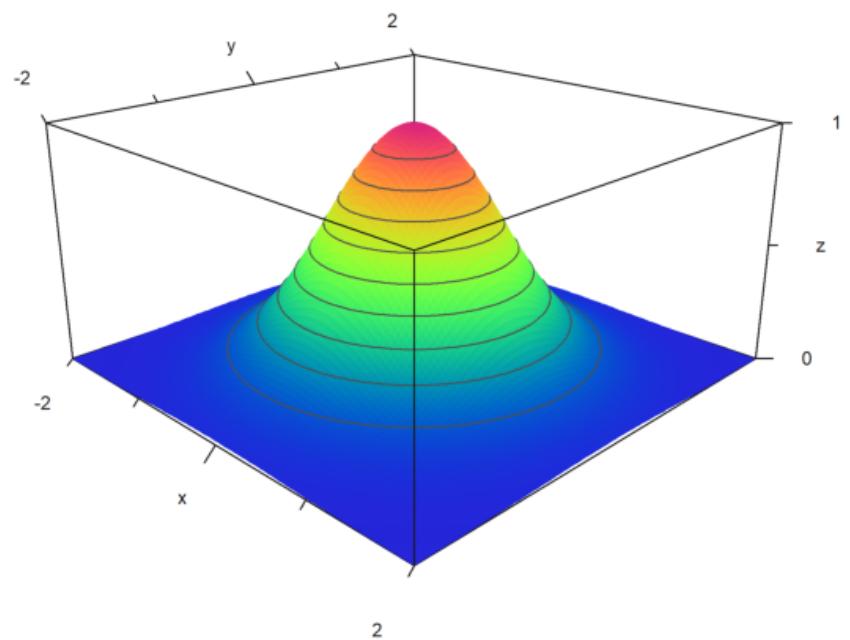
Untuk plot, Euler menambahkan garis-garis kisi. Sebagai alternatif, Anda dapat menggunakan garis level dan satu warna tampilan atau tampilan berwarna spektral. Euler dapat menggambar tinggi fungsi pada plot dengan shading. Pada semua plot 3D, Euler dapat menghasilkan gambar anaglyph merah/cyan.

- >hue: Mengaktifkan shading ringan sebagai ganti kawat.
- >contour: Menampilkan garis kontur otomatis pada plot.
- level=... (atau levels): Sebuah vektor nilai untuk garis kontur.

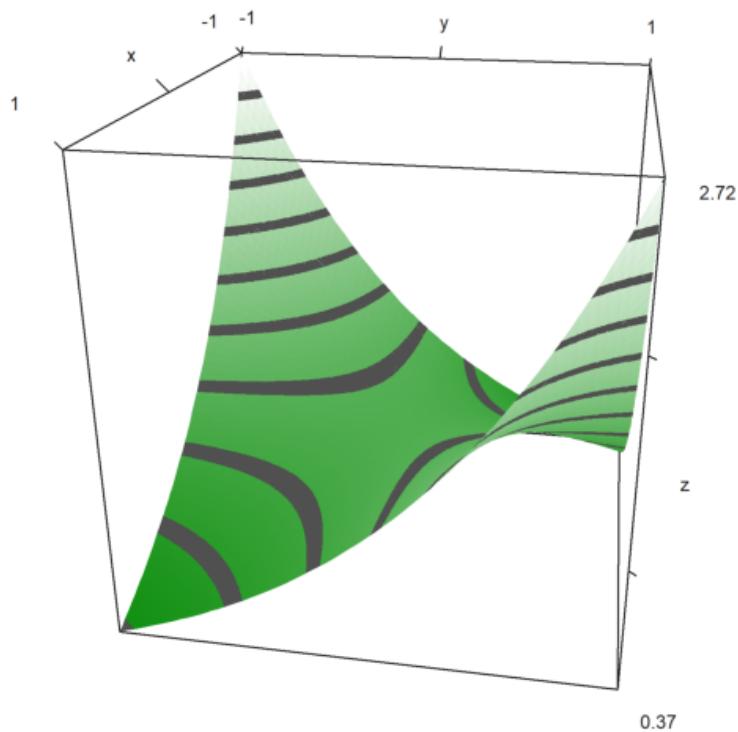
Nilai defaultnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat pada plot, level-level tersebut sebenarnya adalah rentang dari garis-garis level.

Gaya default dapat diubah. Untuk plot kontur berikutnya, kami menggunakan grid yang lebih halus dengan 100x100 titik, menyesuaikan skala fungsi dan plot, serta 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):
```

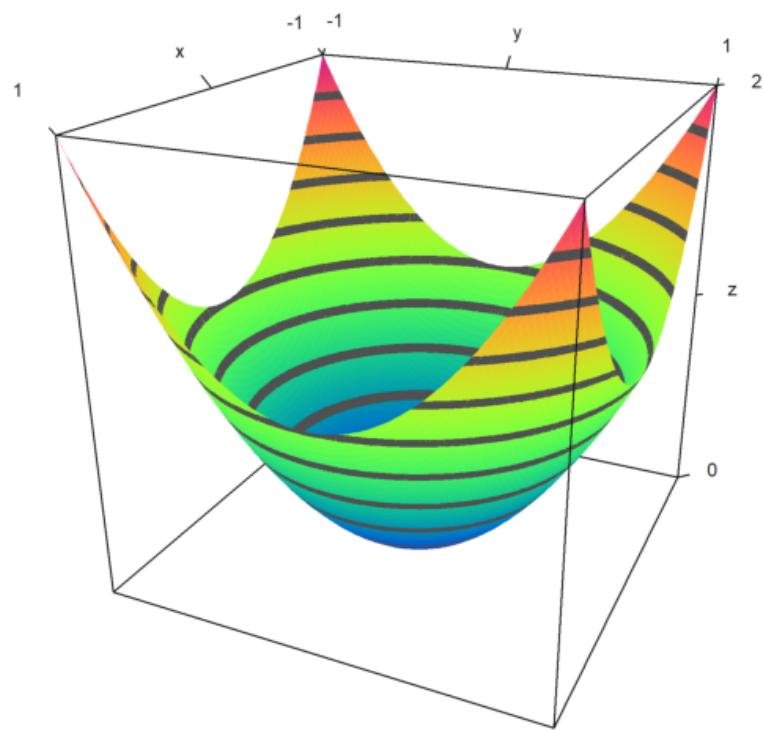


Pengaturan bawaan menggunakan warna abu-abu. Namun, berbagai warna dalam rentang spektral juga tersedia.

- >spektral: Menggunakan skema spektral bawaan
- warna=...: Menggunakan warna khusus atau skema spektral

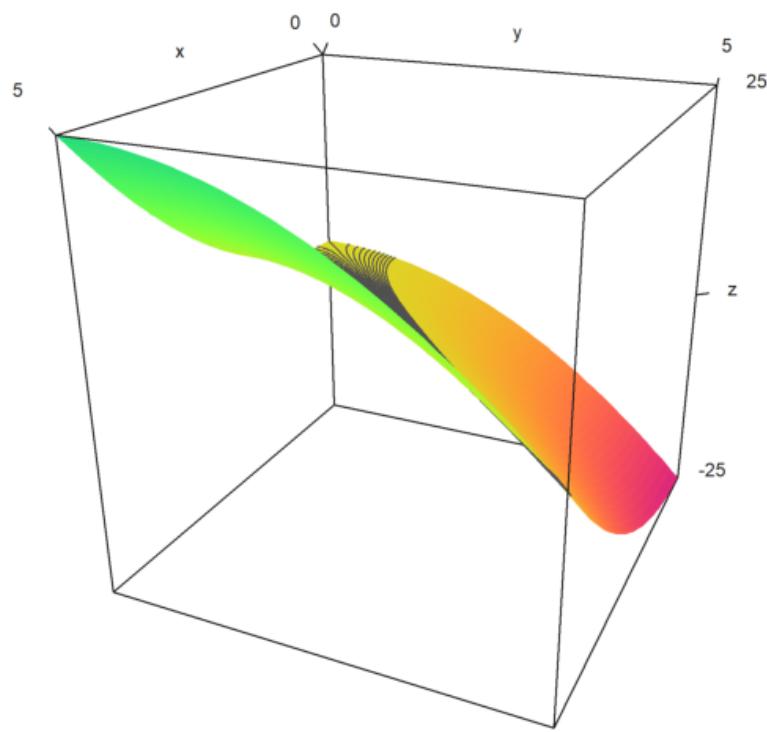
Untuk plot berikutnya, kami menggunakan skema spektral bawaan dan meningkatkan jumlah titik untuk mendapatkan tampilan yang sangat halus.

```
>plot3d("x^2+y^2", >spectral, >contour, n=100):
```



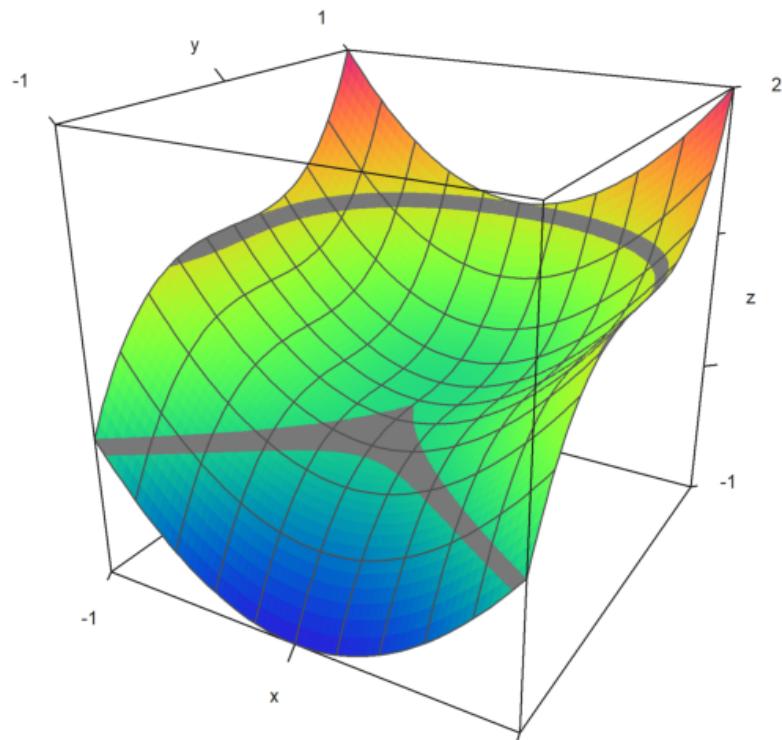
Daripada menggunakan garis level otomatis, kita juga dapat mengatur nilai-nilai garis level ini. Hal ini akan menghasilkan garis level yang tipis daripada rentang level.

```
>plot3d("x^2-y^2", 0, 5, 0, 5, level=-1:0.1:1, color=redgreen) :
```



Dalam plot berikut, kami menggunakan dua rentang tingkat yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas tingkat sebagai kolom.
Selain itu, kami melapisi grid 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):
```

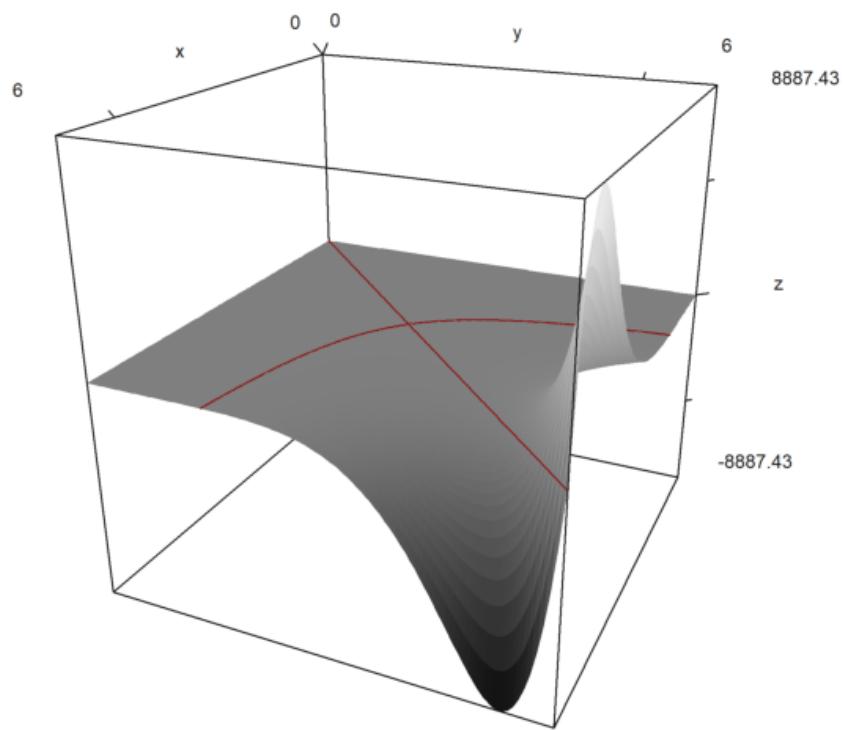


Dalam contoh berikut, kita menggambarkan himpunan, di mana

$$f(x, y) = x^y - y^x = 0$$

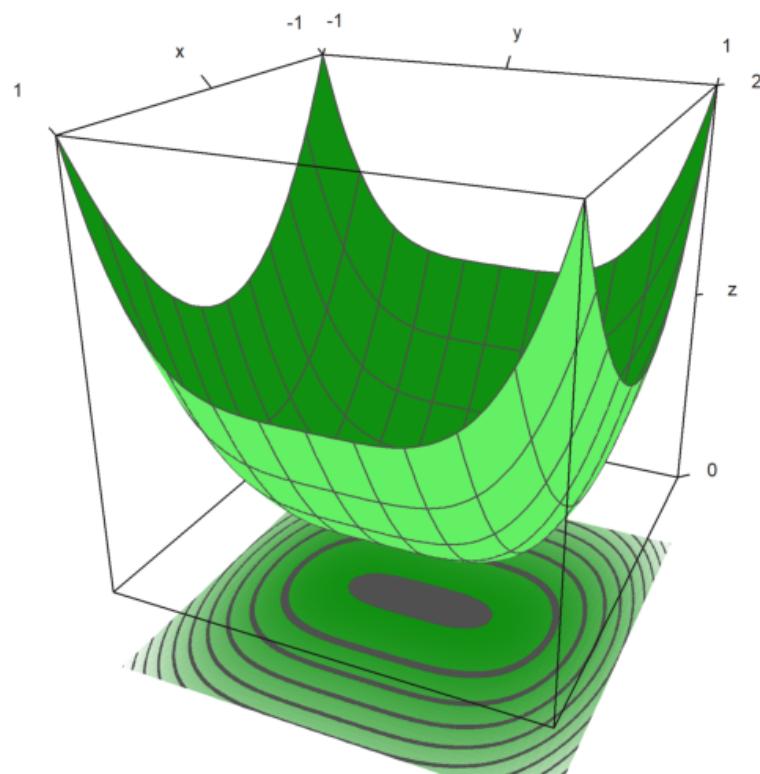
Kita menggunakan sebuah garis tipis tunggal untuk garis levelnya.

```
>plot3d("x^y-y^x", level=0, a=0, b=6, c=0, d=6, contourcolor=red, n=100) :
```



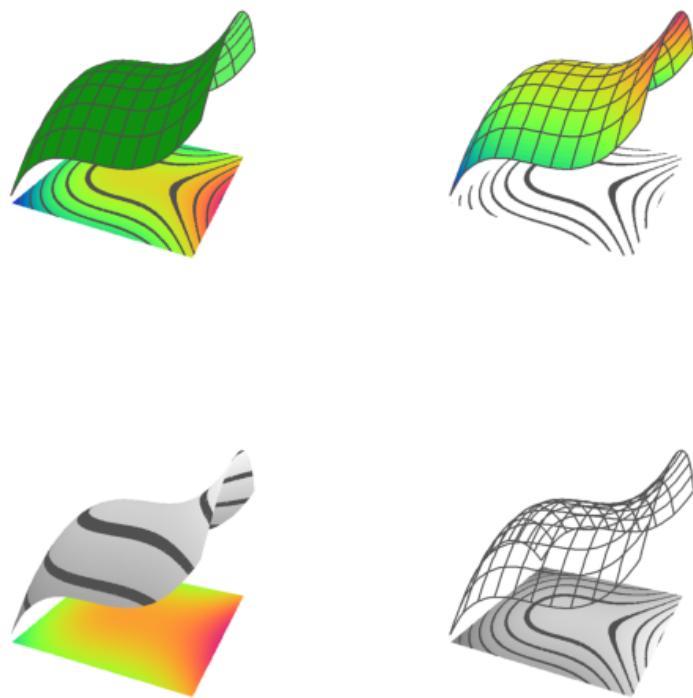
Mungkin untuk menampilkan sebuah bidang kontur di bawah plot. Warna dan jarak ke plot dapat diatur.

```
>plot3d("x^2+y^4", >cp, cpcolor=green, cpdelta=0.2):
```



Berikut beberapa gaya tambahan. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan grid.

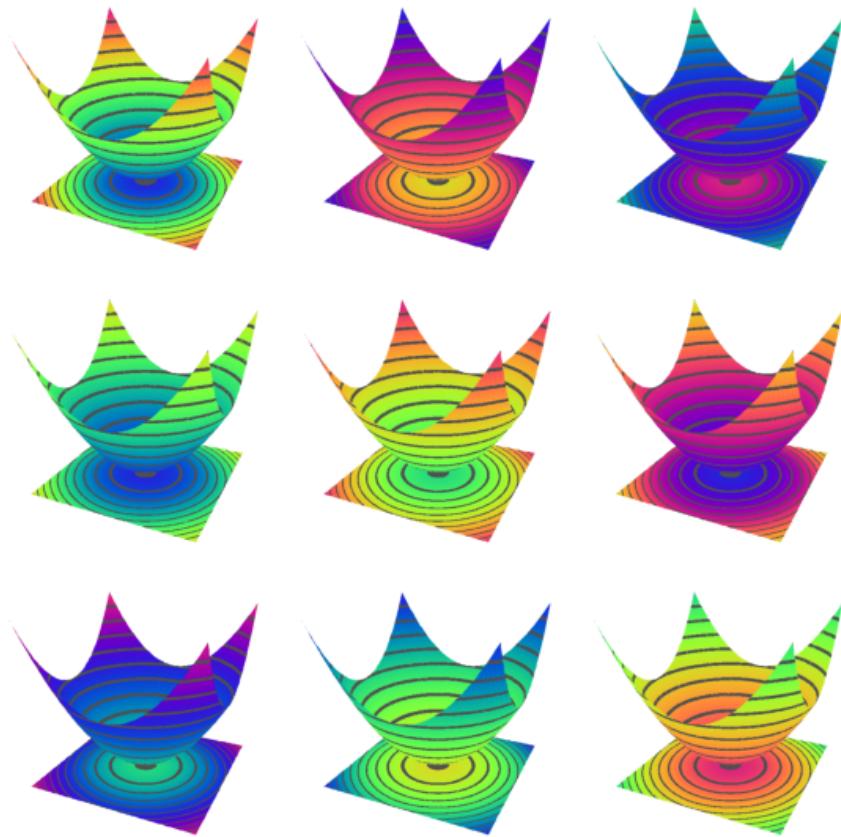
```
>figure(2,2); ...
>expr="y^3-x^2"; ...
>figure(1); ...
> plot3d(expr,<frame,>cp,cpcolor=spectral); ...
>figure(2); ...
> plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
>figure(3); ...
> plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
>figure(4); ...
> plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...
>figure(0):
```



Ada beberapa skema spektral lainnya, diberi nomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan warna=nilai, di mana nilai

- spektral: untuk rentang dari biru hingga merah
- putih: untuk rentang yang lebih lemah
- kuning biru, ungu hijau, biru kuning, hijau merah
- biru kuning, hijau putih, 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);
```



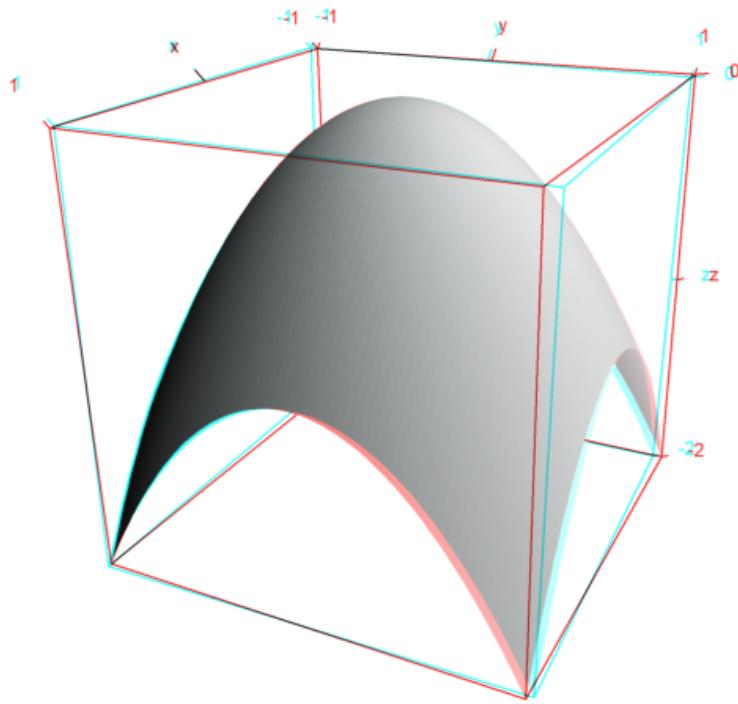
Sumber cahaya dapat diubah dengan tombol 1 dan tombol kurSOR selama interaksi pengguna. Ini juga dapat diatur dengan parameter.

- light: sebuah arah untuk cahaya
- amb: cahaya ambien antara 0 dan 1

Perhatikan bahwa program ini tidak membedakan antara sisi plot. Tidak ada bayangan. Untuk ini, Anda akan 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)":
```

Press l and cursor keys (return to exit)



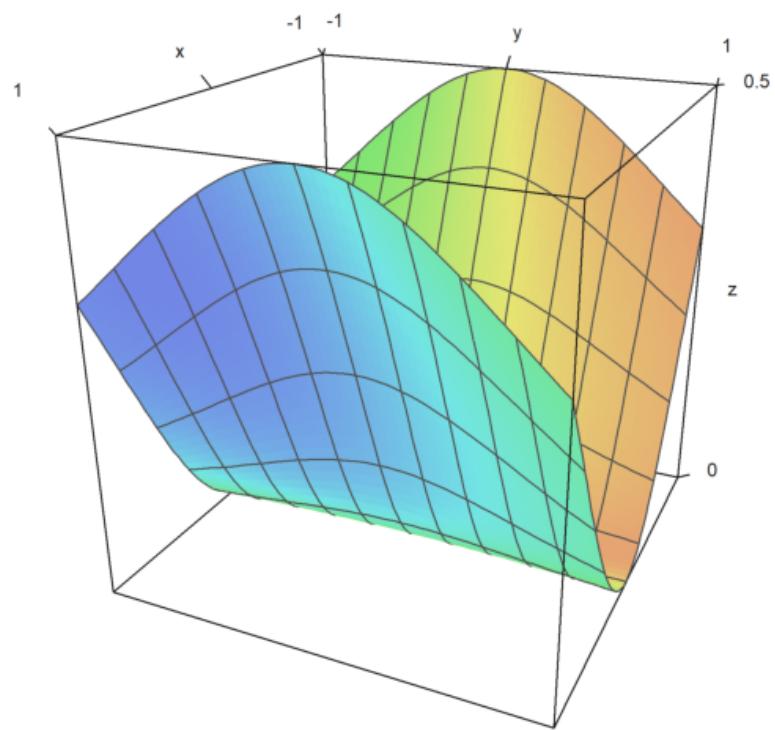
Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

```
>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):
```



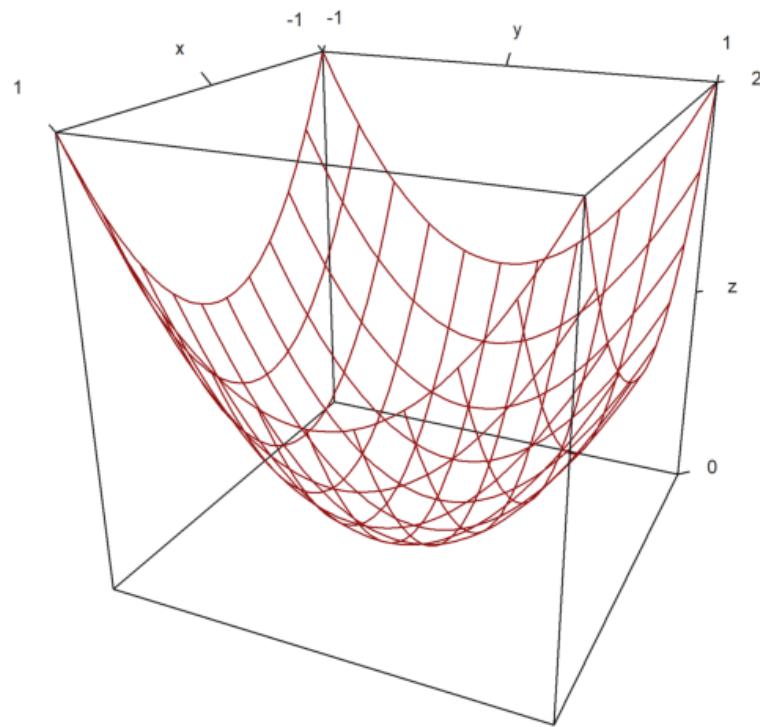
Warna 0 memberikan efek pelangi yang istimewa.

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



Permukaannya juga dapat transparan.

```
>plot3d("x^2+y^2", >transparent, grid=10, wirecolor=red) :
```



Plot Implisit

Terdapat juga plot implisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek-objek tersebut. Fitur-fitur dari plot3d mencakup plot implisit. Plot ini menampilkan himpunan nol dari sebuah fungsi dalam tiga variabel.

Solusi dari

$$f(x, y, z) = 0$$

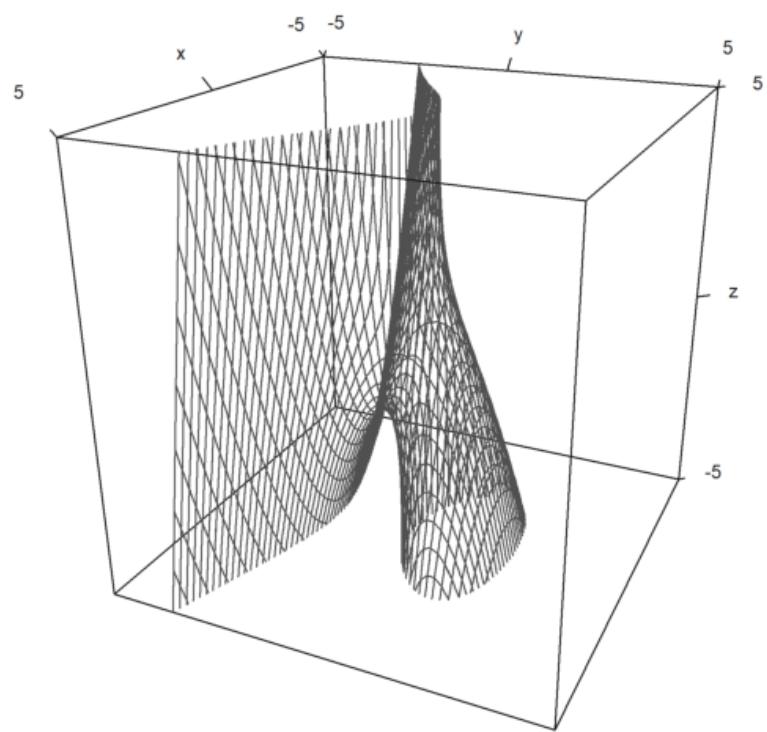
dapat divisualisasikan dalam potongan sejajar dengan bidang x-y, x-z, dan y-z.

- implicit=1: potongan sejajar dengan bidang y-z
- implicit=2: potongan sejajar dengan bidang x-z
- implicit=4: potongan sejajar dengan bidang x-y

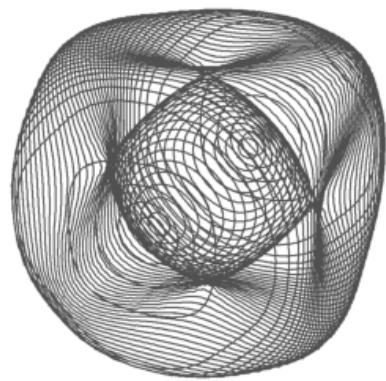
Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh ini, kami plot

$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

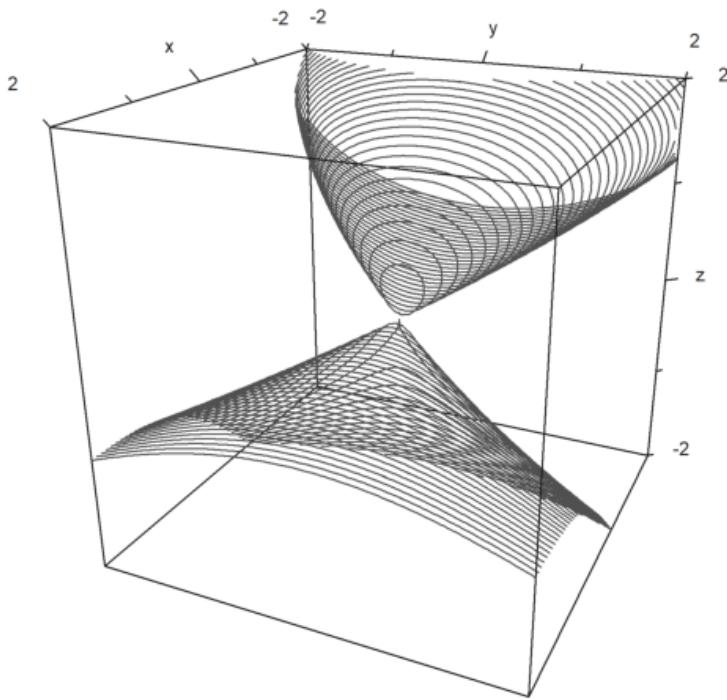
```
>plot3d("x^2+y^3+z*y-1", r=5, implicit=3):
```



```
>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-c^2)^2+(y^2-1)^2)
```



```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):
```



Plotting Data 3D

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu memberikan matriks nilai-nilai x, y, dan z, atau tiga fungsi atau ekspresi $fx(x, y)$, $fy(x, y)$, $fz(x, y)$.

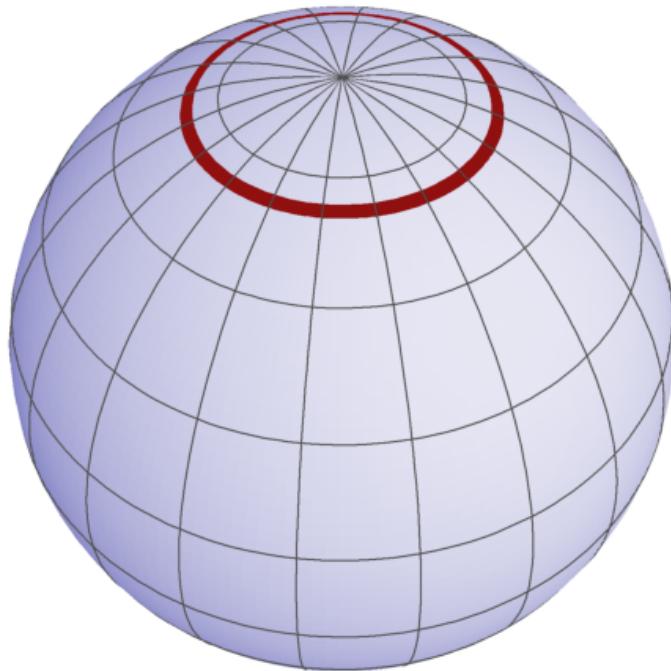
$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

Karena x, y, z adalah matriks, kami mengasumsikan bahwa (t, s) berjalan melalui grid persegi. Sebagai hasilnya, Anda dapat membuat gambar-gambar persegi panjang di ruang.

Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat dengan efektif.

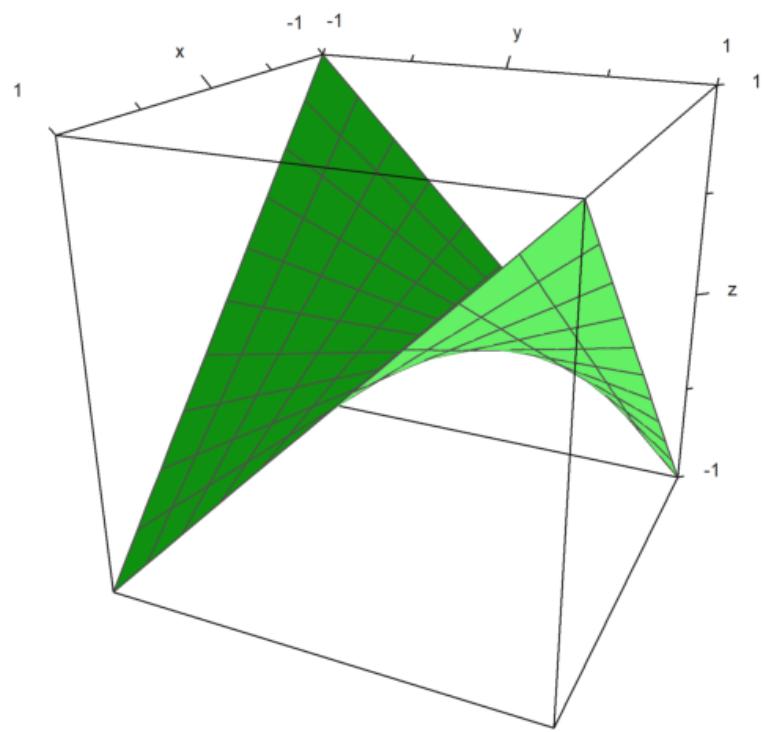
Pada contoh berikut, kami menggunakan vektor nilai-nilai t dan vektor kolom nilai-nilai s untuk memparametrisasi permukaan bola. Dalam gambaran, kita dapat menandai wilayah-wilayah, dalam hal ini wilayah kutub.

```
>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 contohnya, yaitu grafik suatu fungsi.

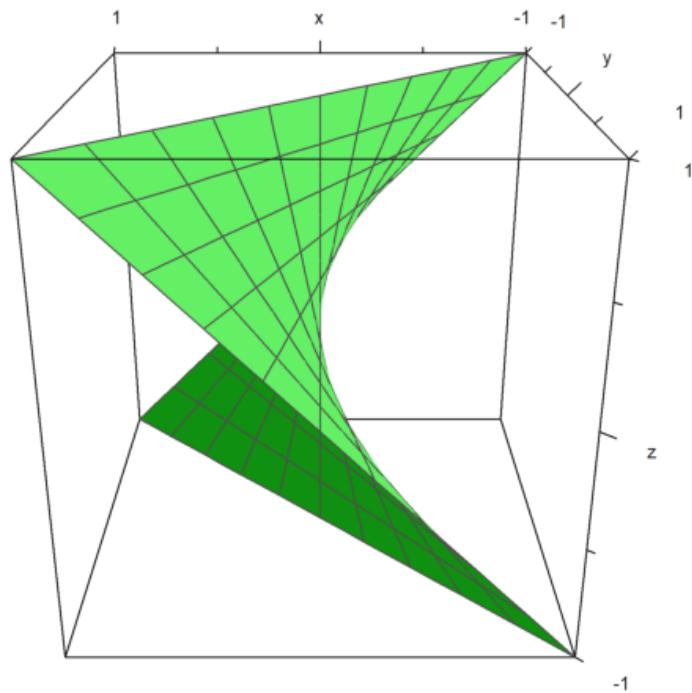
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10);
```



Namun, kita dapat membuat berbagai jenis permukaan. Berikut adalah permukaan yang sama sebagai fungsi:

$$x = yz$$

```
>plot3d(t*s, t, s, angle=180°, grid=10):
```



Dengan lebih banyak usaha, kita dapat menghasilkan banyak permukaan.

Pada contoh berikut, kita membuat tampilan yang berbayang dari bola yang distorsi. Koordinat biasa untuk bola tersebut adalah

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

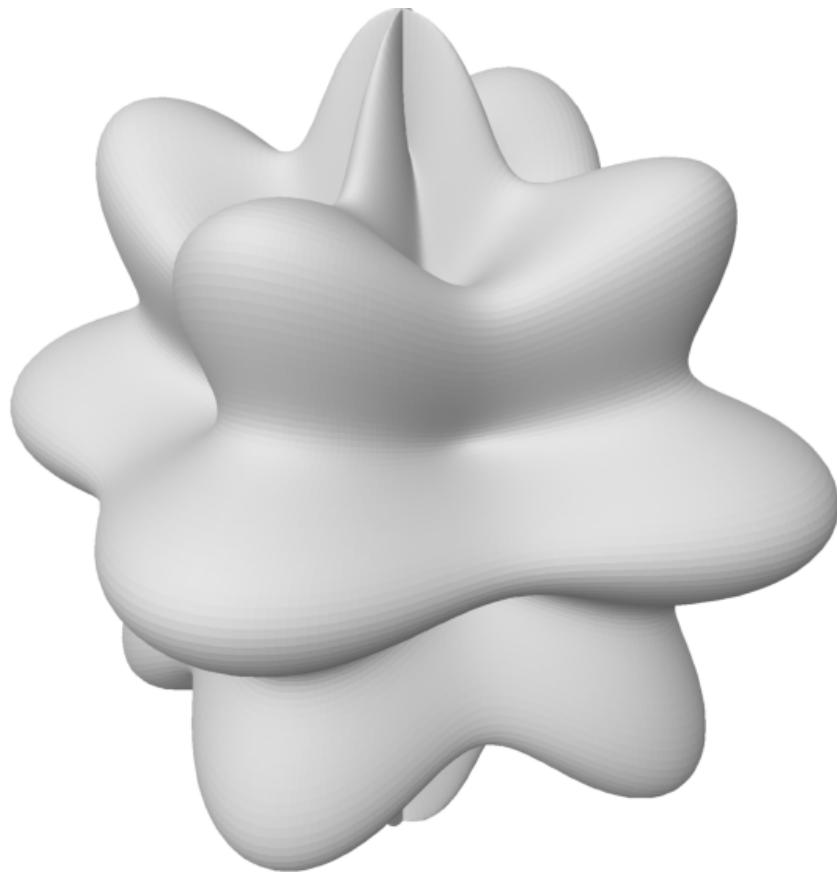
dengan

$$0 \leq t \leq 2\pi, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2}.$$

Kita merubahnya dengan faktor

$$d(t, s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

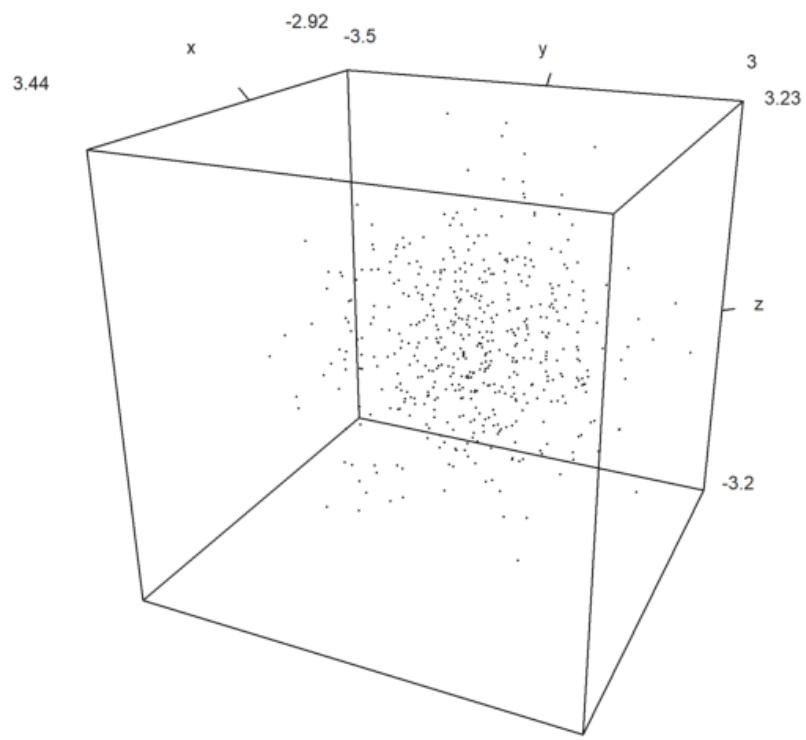
```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
> light=[1,0,1],frame=0,zoom=5):
```



Tentu saja, awan titik juga mungkin. Untuk menggambar data titik di dalam ruang, kita memerlukan tiga vektor untuk koordinat titik-titik tersebut.

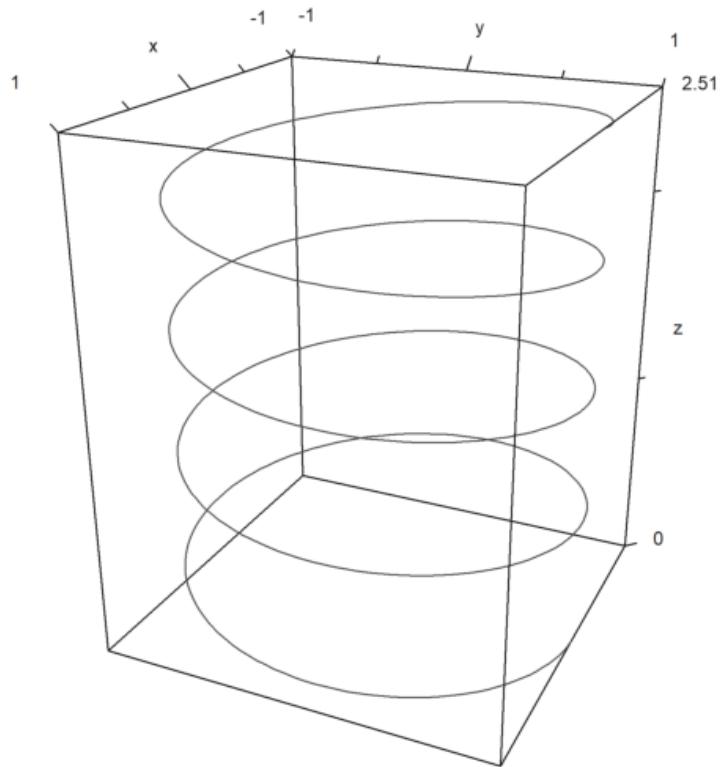
Gaya-gaya ini sama seperti dalam plot2d dengan points=true;

```
>n=500; ...  
> plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

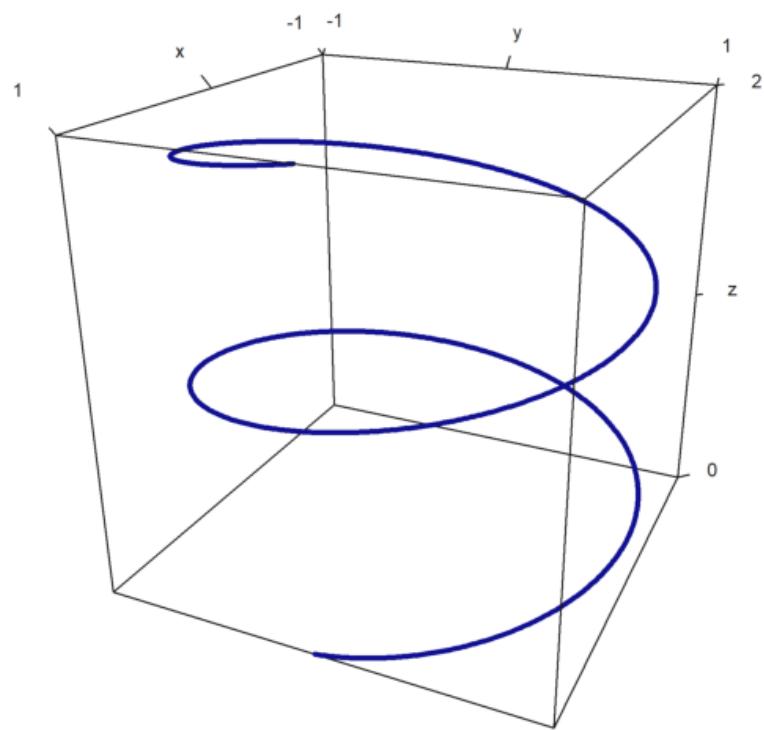


Ini juga memungkinkan untuk menggambar kurva dalam tiga dimensi (3D). Dalam kasus ini, lebih mudah untuk menghitung sebelumnya titik-titik dari kurva tersebut. Untuk kurva dalam bidang, kita menggunakan urutan koordinat dan parameter `wire=true`.

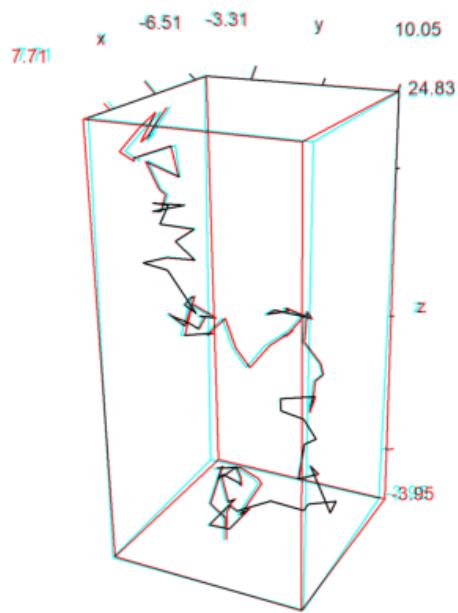
```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
>lineWidth=3, wirecolor=blue):
```

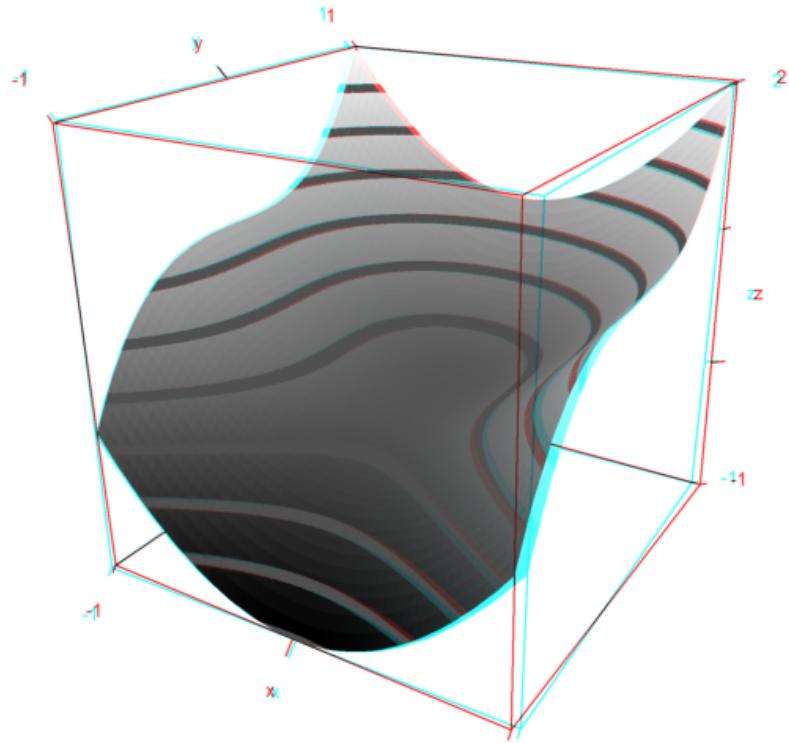


```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire) :
```



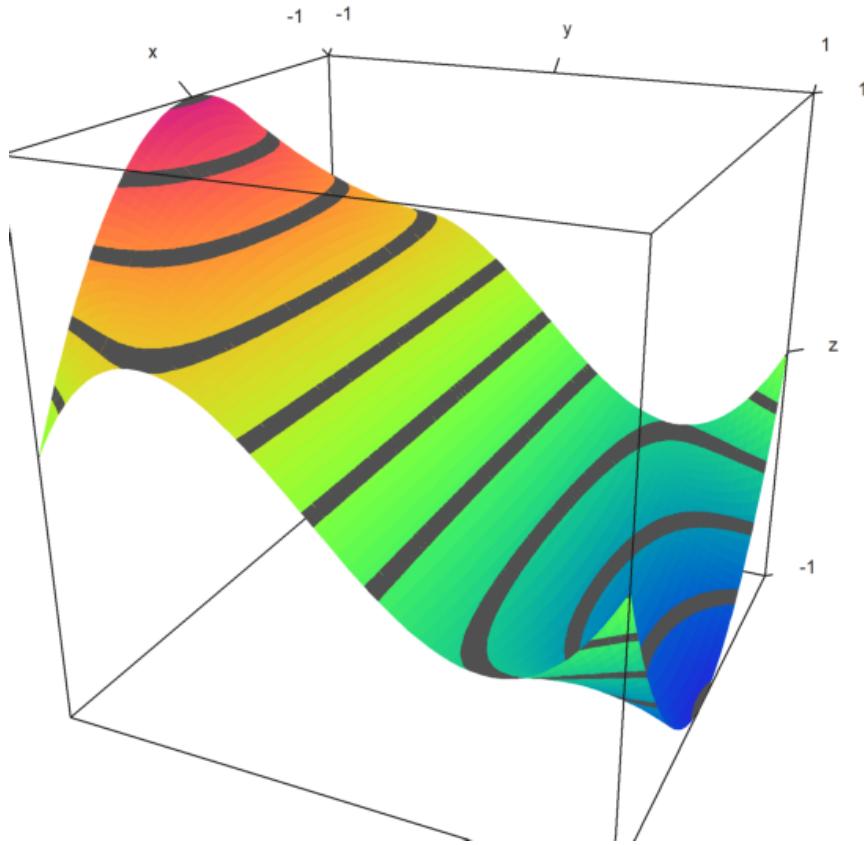
EMT juga dapat membuat plot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda memerlukan kacamata merah/biru.

```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```



Seringkali, sebuah skema warna spektral digunakan untuk plot. Ini menekankan tinggi dari fungsi tersebut.

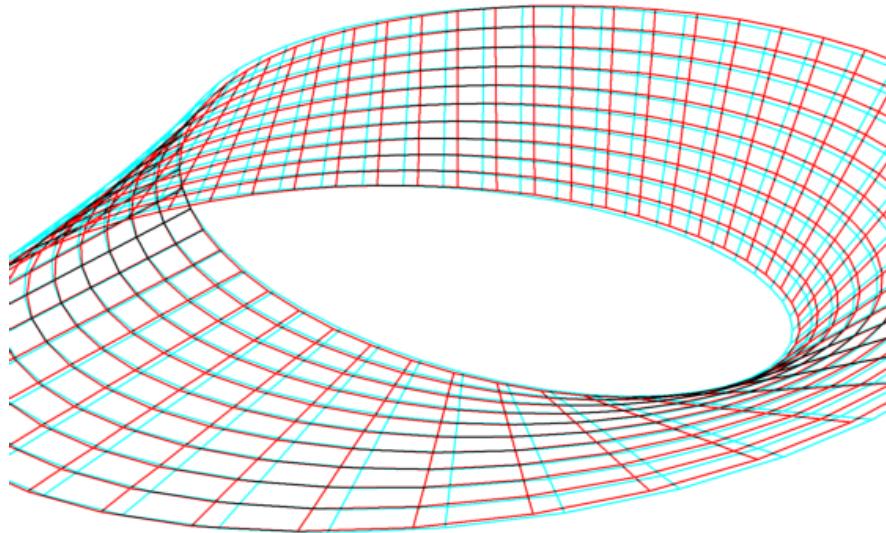
```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```



Euler juga dapat menggambar permukaan yang diparameterkan, ketika parameter-parameter tersebut adalah nilai-nilai x , y , dan z dari gambar kisi berbentuk persegi di dalam ruang.

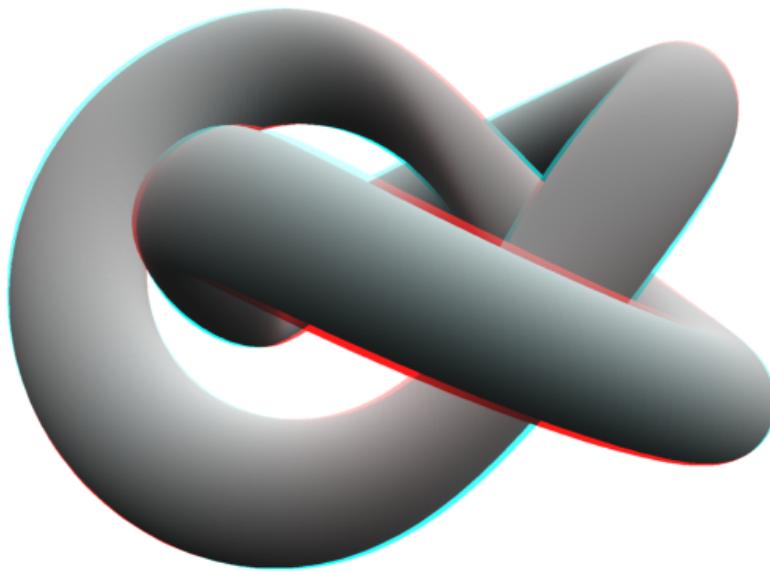
Untuk demonstrasi berikutnya, kami menyiapkan parameter u dan v , dan menghasilkan koordinat ruang dari keduanya.

```
>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 adalah contoh yang lebih rumit, yang sangat 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):
```



Grafik Statistik

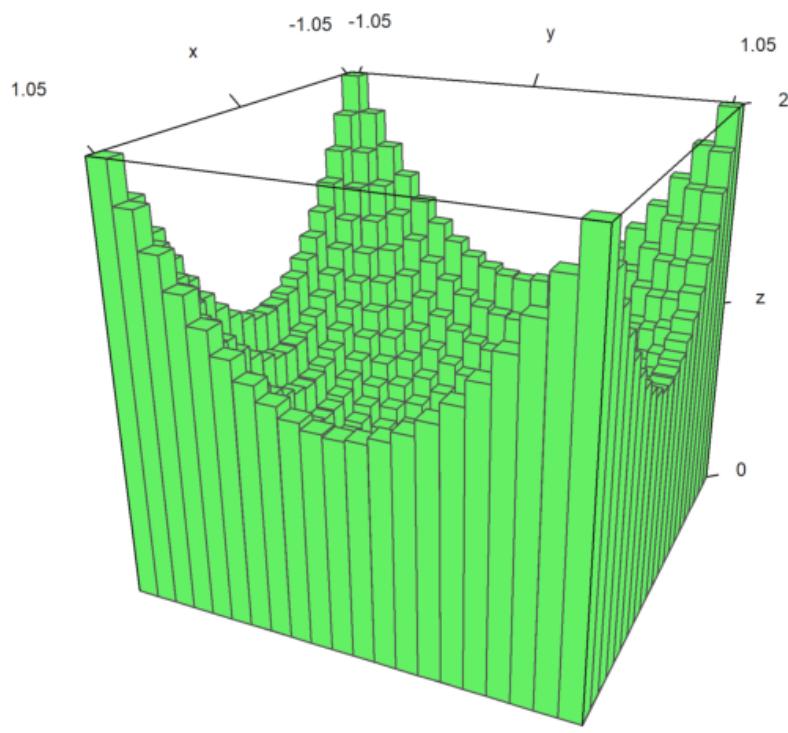
Grafik batang juga mungkin. Untuk ini, kita harus menyediakan

- x: vektor baris dengan n+1 elemen
- y: vektor kolom dengan n+1 elemen
- z: matriks nxn dari nilai-nilai.

z bisa lebih besar, tetapi hanya nilai-nilai nxn yang akan digunakan.

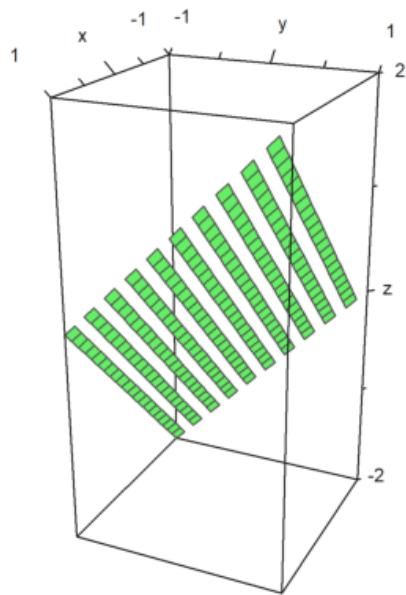
Pada contoh ini, pertama-tama kita menghitung nilai-nilai. Kemudian kita menyesuaikan x dan y, sehingga vektor-vektor tersebut berpusat pada nilai-nilai yang digunakan.

```
>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);
```



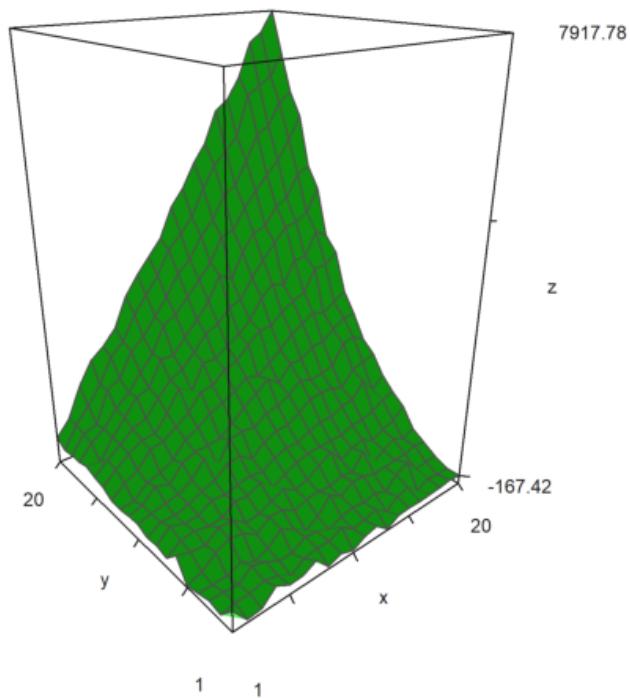
Mungkin untuk membagi plot permukaan menjadi dua atau lebih bagian.

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20);
```

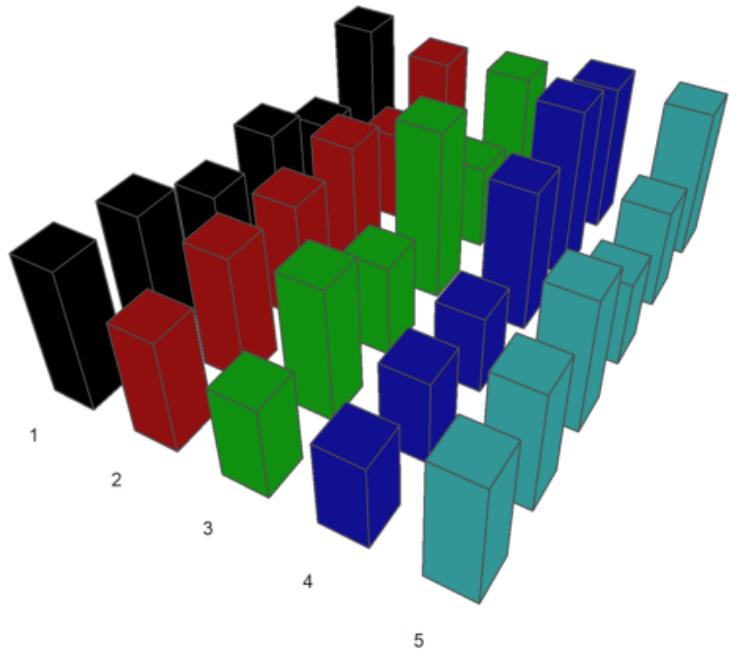


Jika Anda memuat atau menghasilkan matriks data M dari sebuah file dan perlu membuat plotnya dalam tiga dimensi (3D), Anda dapat mengubah skala matriks tersebut menjadi [-1,1] dengan menggunakan perintah "scale(M)", atau mengubah skala matriks tersebut dengan menggunakan perintah ">zscale". Hal ini dapat dikombinasikan dengan faktor-faktor skala individu yang akan diterapkan secara 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):
```

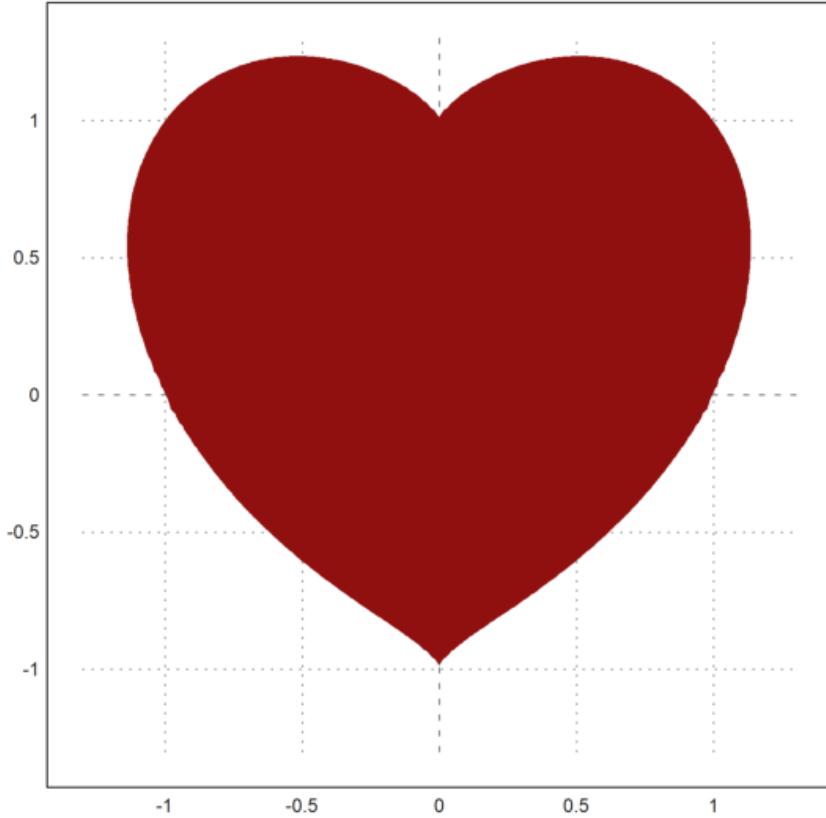


```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v',scols=1:5,ccols=[1:5]):
```



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):
```



```
>ekspressi &= (x^2+y^2-1)^3-x^2*y^3; $ekspressi
```

$$(y^2 + x^2 - 1)^3 - x^2 y^3$$

Kami ingin memutar kurva hati sekitar sumbu y. Berikut adalah ekspresi yang mendefinisikan hati:

$$f(x, y) = (x^2 + y^2 - 1)^3 - x^2 \cdot y^3.$$

Selanjutnya, kami mengatur

$$x = r \cos(a), \quad y = r \sin(a).$$

```
>function fr(r,a) &= ekspressi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

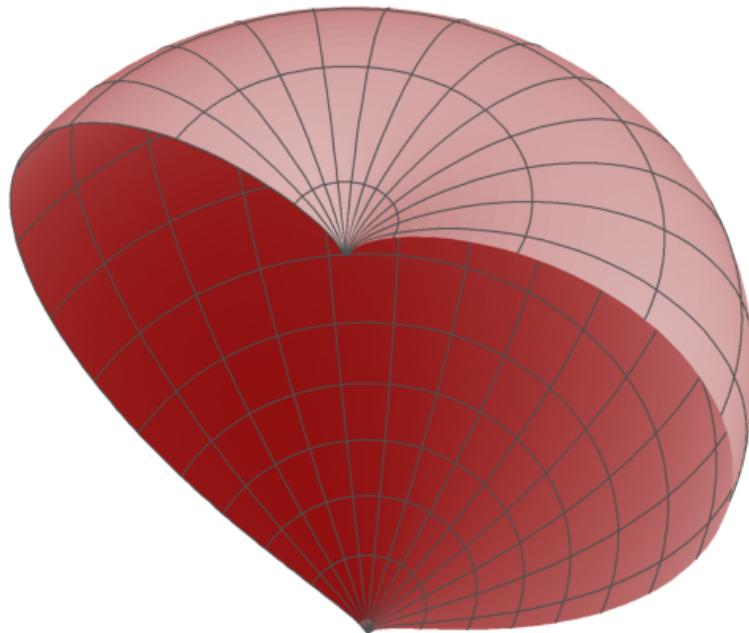
$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2\sin a) r^5}{16}$$

Ini memungkinkan untuk mendefinisikan fungsi numerik, yang menyelesaikan untuk r, jika a diberikan. Dengan fungsi itu, kita dapat menggambar hati yang berputar 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 adalah plot 3D dari gambar di atas yang diputar sekitar sumbu z. Kami mendefinisikan fungsi yang menggambarkan objek tersebut.

```

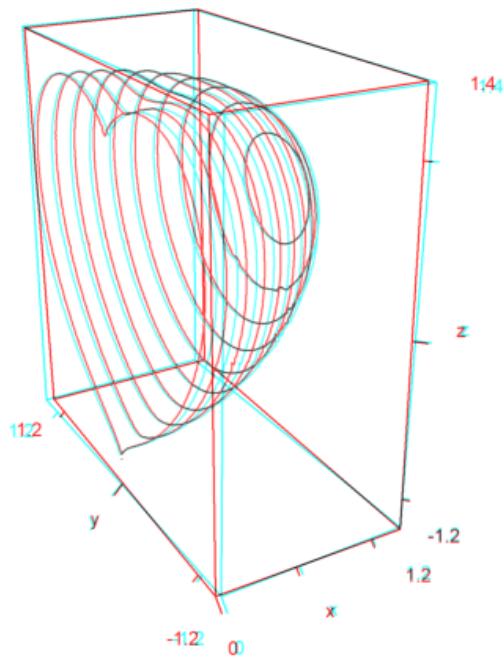
>function f(x,y,z) ...
r=x^2+y^2;
return (r+z^2-1)^3-r*z^3;
endfunction

```

```

>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

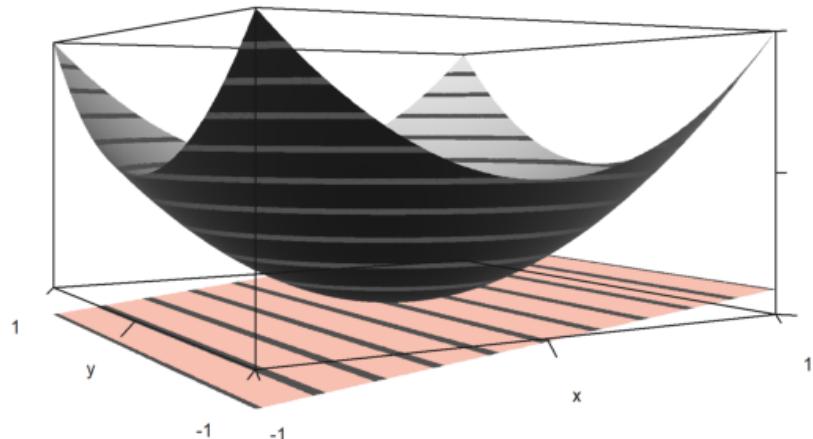
Fungsi plot3d ini bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih dasar, mungkin untuk mendapatkan plot bingkai dari objek apa pun yang Anda suka.

Meskipun Euler bukan program 3D, ini dapat menggabungkan beberapa objek dasar. Kami mencoba untuk memvisualisasikan paraboloid dan tangennya.

```
>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 bingkai, dan mengatur tampilan.

```
>framedplot("myplot", [-1,1,-1,1,0,1],height=0,angle=-30°, ...
> center=[0,0,-0.7],zoom=3):
```

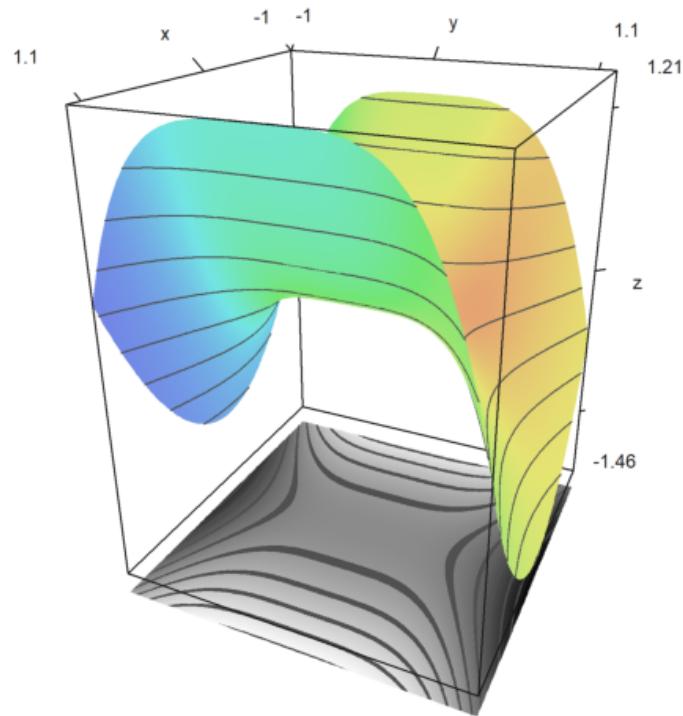


Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa `plot3d()` menyetel jendela ke `fullwindow()` secara default, tetapi `plotcontourplane()` berasumsi demikian.

```
>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):
```



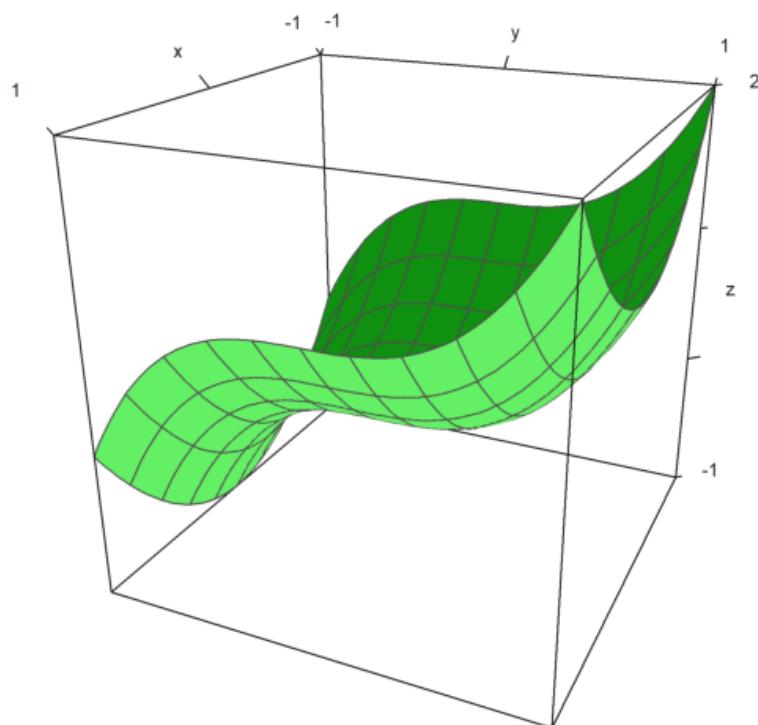
Animasi

Euler dapat menggunakan frame untuk melakukan pra-komputasi animasi.

Salah satu fungsi yang memanfaatkan teknik ini adalah memutar. Itu dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi ini memanggil `addpage()` untuk setiap plot baru. Akhirnya fungsi tersebut menganimasikan plotnya.

Silakan pelajari sumber rotasi untuk melihat lebih detail.

```
>function testplot () := plot3d("x^2+y^3"); ...
>rotate("testplot"); testplot():
```



Menggambar Povray

Dengan bantuan berkas Euler povray.e, Euler dapat menghasilkan berkas Povray. Hasilnya sangat bagus untuk dilihat.

Anda perlu menginstal Povray (32bit atau 64bit) dari <http://povray.org/>,

dan letakkan subdirektori "bin" dari Povray ke dalam path lingkungan, atau atur variabel "defaultpovray" dengan path lengkap yang mengarah ke "pvengine.exe".

Antarmuka Povray Euler menghasilkan berkas Povray dalam direktori rumah pengguna, dan memanggil Povray untuk mengurai berkas-berkas ini. Nama berkas default adalah current.pov, dan direktori default adalah eulerhome(), biasanya c:\Users\Username\Euler. Povray menghasilkan berkas PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan berkas-berkas ini, gunakan povclear().

Fungsi pov3d berada dalam semangat yang sama seperti plot3d. Ini dapat menghasilkan grafik dari fungsi $f(x,y)$, atau permukaan dengan koordinat X,Y,Z dalam matriks, termasuk garis-garis level opsional. Fungsi ini secara otomatis memulai raytracer, 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-objek tersebut. Untuk menggunakan fungsi-fungsi ini, mulailah berkas Povray dengan povstart(). Kemudian gunakan writeln(...) untuk menulis objek-objek ke berkas adegan. Terakhir, akhiri berkas dengan povend(). Secara default, raytracer akan dijalankan, dan PNG akan dimasukkan ke dalam notebook Euler.

Fungsi objek memiliki parameter yang disebut "look", yang memerlukan string dengan kode Povray untuk tekstur dan penyelesaian objek. Fungsi povlook() dapat digunakan untuk menghasilkan string ini. Ini memiliki parameter untuk warna, transparansi, Phong Shading, dll.

Perhatikan bahwa alam semesta Povray memiliki sistem koordinat yang berbeda. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat terus berpikir dalam sistem koordinat Euler dengan z mengarah ke atas secara vertikal, dan sumbu x, y, z sesuai dengan tangan kanan.

Anda perlu memuat berkas povray.

```
>load povray;
```

Pastikan direktori bin Povray ada dalam path. Jika tidak, edit variabel berikut agar berisi path ke eksekusi povray.

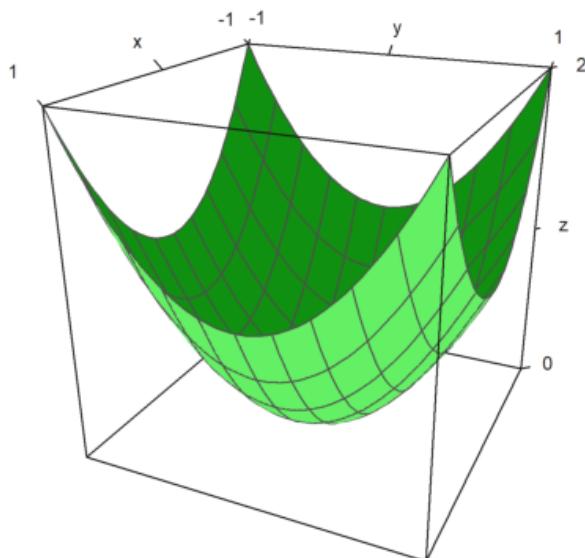
```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

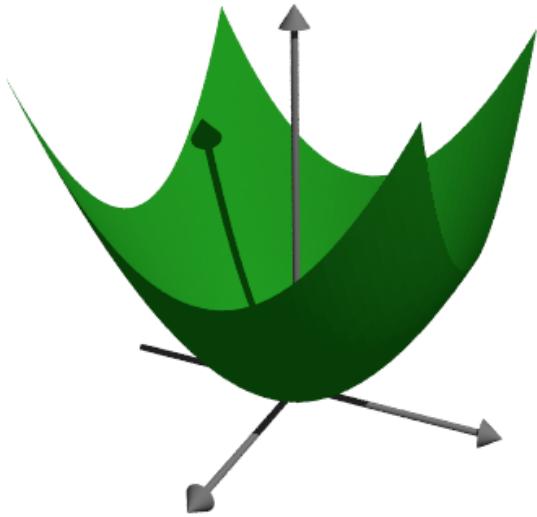
Untuk kesan pertama, kita akan membuat grafik dari sebuah fungsi sederhana. Perintah berikut akan menghasilkan sebuah file povray dalam direktori pengguna Anda, dan menjalankan Povray untuk mengejar sinar dari file tersebut.

Jika Anda menjalankan perintah berikut, GUI Povray seharusnya akan terbuka, menjalankan file, dan kemudian otomatis menutupnya. Demi alasan keamanan, Anda akan diminta apakah Anda ingin mengizinkan file exe ini berjalan. Anda dapat menekan tombol batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin juga harus menekan OK dalam jendela Povray untuk mengakui dialog awal dari Povray.

```
>plot3d("x^2+y^2", zoom=2) :
```

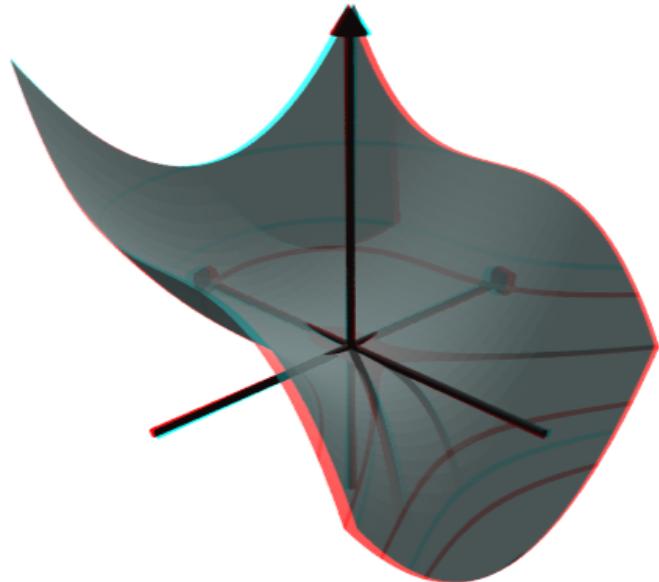


```
>pov3d("x^2+y^2", zoom=3);
```



Kita dapat membuat fungsi tersebut transparan dan menambahkan selesai lainnya. Kita juga dapat menambahkan garis-garis level ke plot fungsi.

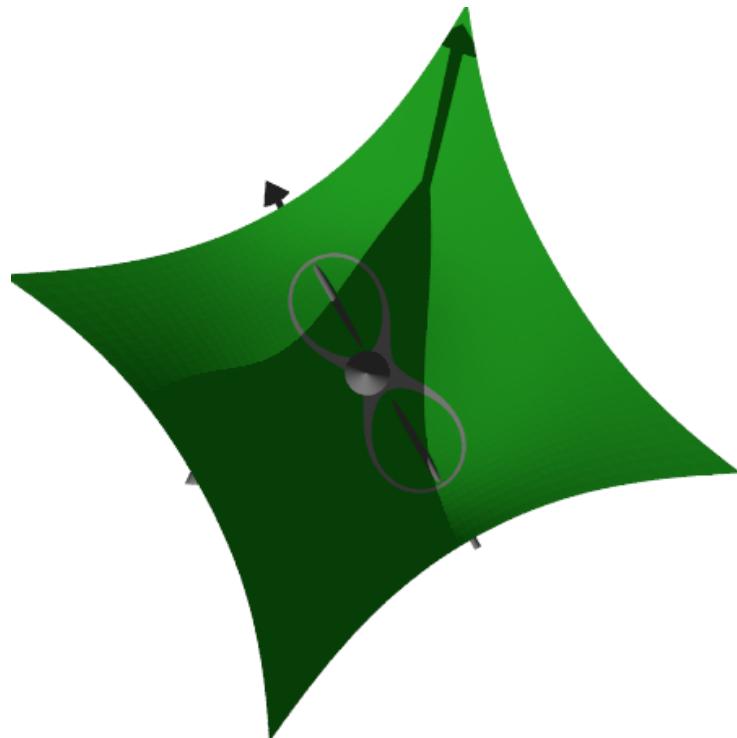
```
>pov3d("x^2+y^3",axiscolor=red,angle=-45°,>anaglyph, ...
> look=povlook(cyan,0.2),level=-1:0.5:1,zoom=3.8);
```



Terkadang perlu mencegah penskalaan fungsi dan menyesuaikan skala fungsi secara manual.

Kita memplot himpunan titik-titik dalam 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=90°,n=50, ...
> <fscale,zoom=3);
```

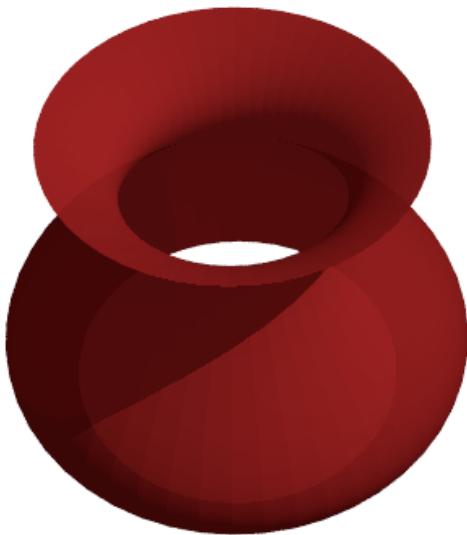


Plotting dengan Koordinat

Daripada menggunakan fungsi, kita dapat melakukan plotting dengan menggunakan koordinat. Seperti dalam plot3d, kita memerlukan tiga matriks untuk mendefinisikan objek tersebut.

Dalam contoh ini, kita memutar suatu fungsi sekitar sumbu z.

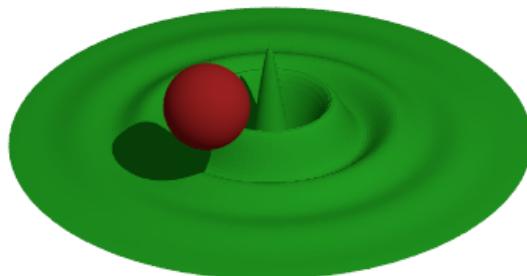
```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,50)'; ...
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light=[10,5,15]);
```



Dalam contoh berikut, kita membuat grafik gelombang yang meredam. Kita menghasilkan gelombang tersebut dengan menggunakan bahasa matriks Euler.

Kita juga menunjukkan bagaimana objek tambahan dapat ditambahkan ke dalam sebuah adegan pov3d. Untuk pembuatan objek, lihat contoh-contoh berikut. Perhatikan bahwa plot3d akan menyesuaikan skala plot agar muat dalam 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=4,axis=0,height=30°,add=povsphere([0.5,0,0.25],0.15,povlook(red)), ...
> w=500,h=300);
```



Dengan metode pencahayaan canggih dari Povray, sedikit sekali titik dapat menghasilkan permukaan yang sangat halus. Hanya di batas-batas dan dalam bayangan trik tersebut mungkin menjadi jelas.

Untuk ini, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>Z &= x^2*y^3
```

$$\begin{matrix} 2 & 3 \\ x & y \end{matrix}$$

Persamaan permukaannya adalah $[x, y, Z]$. Kami menghitung dua turunan terhadap x dan y dari ini dan mengambil hasil perkalian silang sebagai normalnya.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

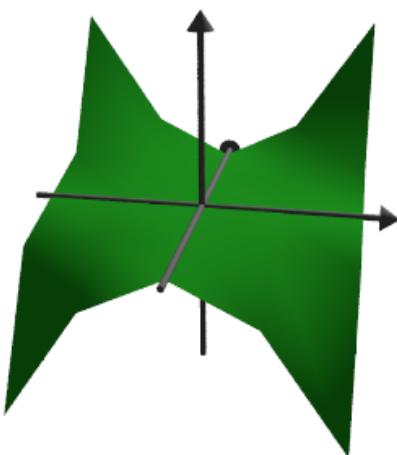
Kami mendefinisikan normal sebagai hasil perkalian silang dari turunan-turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

$$\begin{matrix} 3 & 2 & 2 \\ [-2x^y, -3x^y, 1] \end{matrix}$$

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 adalah simpul Trefoil yang dibuat oleh A. Busser dalam Povray. Ada versi yang ditingkatkan dari ini dalam contoh-contoh.

See: Examples\Trefoil Knot | Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kita menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung vektor normal untuk kita. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolik.

```
>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 hasil perkalian silang dari kedua turunan tersebut.

```
>dn &= crossproduct(dx,dy);
```

Sekarang kita mengevaluasinya secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

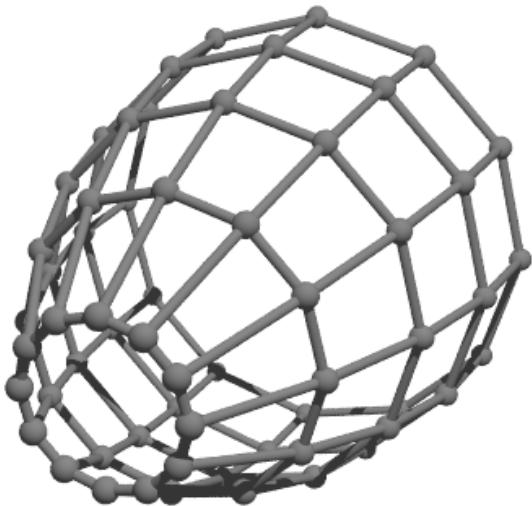
Vektor normal adalah evaluasi dari ekspresi simbolis dn[i] untuk i=1,2,3. Sintaks untuk ini adalah "&"ekspre-si"(parameter). Ini merupakan alternatif dari metode dalam contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolis NX, NY, NZ terlebih dahulu.

```
>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));
```



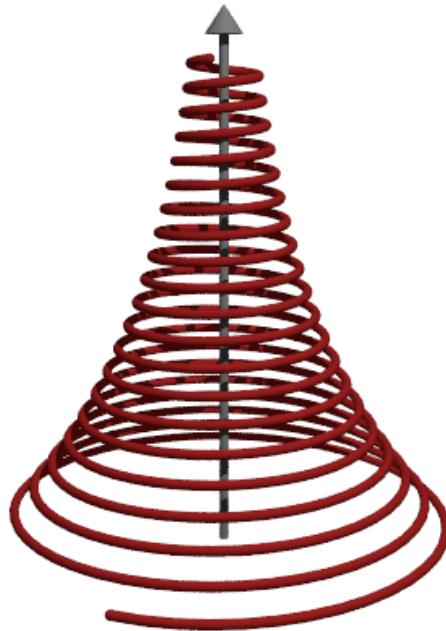
Kita juga dapat menghasilkan sebuah grid dalam 3D.

```
>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-kurva menjadi mungkin.

```
>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, kita menggunakan pov3d untuk memplot permukaan. Antarmuka povray dalam Euler juga dapat menghasilkan objek Povray. Objek-objek ini disimpan sebagai string dalam Euler, dan perlu ditulis ke file Povray.

Kita memulai output dengan povstart().

```
>povstart (zoom=4);
```

Pertama, kita mendefinisikan tiga silinder, dan menyimpannya dalam bentuk string dalam Euler.

Fungsi povx() dll. hanya mengembalikan vektor [1,0,0], yang bisa digunakan sebagai gantinya.

```
>c1=povcylinder (-povx,povx,1,povlook(red)); ...
>c2=povcylinder (-povy,povy,1,povlook(yellow)); ...
>c3=povcylinder (-povz,povz,1,povlook(blue)); ...
```

Teks tersebut mengandung kode Povray, yang pada saat itu tidak perlu kita pahami.

```
>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 pada objek dalam tiga warna yang berbeda.

Hal ini dilakukan dengan menggunakan `povlook()`, yang mengembalikan sebuah string dengan kode Povray yang relevan. Kami dapat menggunakan warna Euler default, atau mendefinisikan warna sendiri. Kami juga dapat menambahkan transparansi, atau mengubah cahaya ambien.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

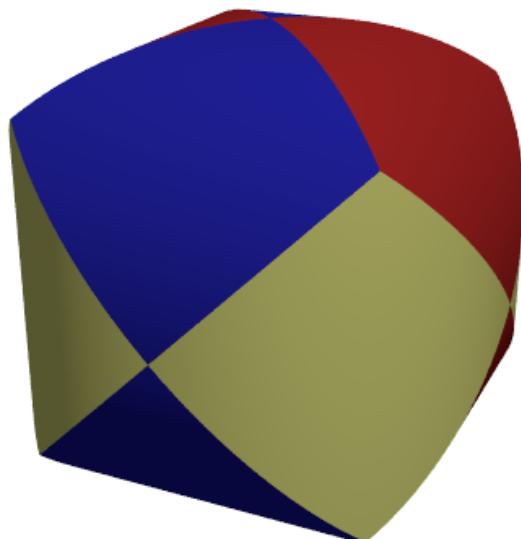
```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> } }
finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek perpotongan, dan menulis hasilnya ke dalam file.

```
>writeln(povintersection([c1,c2,c3]));
```

Perpotongan dari tiga silinder sulit untuk dibayangkan, jika Anda belum pernah melihatnya sebelumnya.

```
>povend;
```



Berikut adalah fungsi-fungsi berikut yang menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan bagaimana Euler mengatasi objek Povray sederhana. Fungsi `povbox()` mengembalikan sebuah string yang berisi koordinat kotak, tekstur, dan finishing.

```
>function onebox(x,y,z,d) := povbox([x,y,z],[x+d,y+d,z+d],povlook());
>function fractal (x,y,z,h,n) ...
```

```

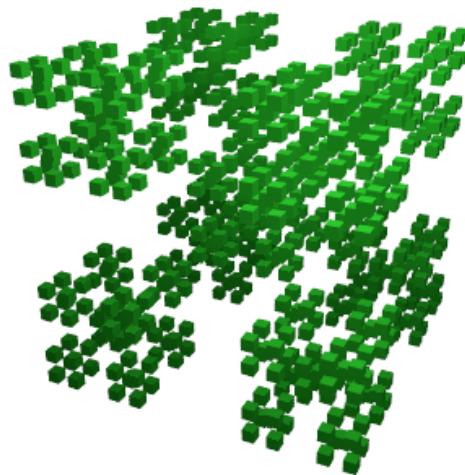
if n==1 then writeln(onebox(x,y,z,h));
else
  h=h/3;
  fractal(x,y,z,h,n-1);
  fractal(x+2*h,y,z,h,n-1);
  fractal(x,y+2*h,z,h,n-1);
  fractal(x,y,z+2*h,h,n-1);
  fractal(x+2*h,y+2*h,z,h,n-1);
  fractal(x+2*h,y,z+2*h,h,n-1);
  fractal(x,y+2*h,z+2*h,h,n-1);
  fractal(x+2*h,y+2*h,z+2*h,h,n-1);
  fractal(x+h,y+h,z+h,h,n-1);
endif;
endfunction

```

```

>povstart(fade=10,<shadow>;
>fractal(-1,-1,-1,2,4);
>povend();

```



Perbedaan memungkinkan pemisahan satu objek dari yang lain. Seperti perpotongan, ini adalah bagian dari objek CSG dalam Povray.

```
>povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kami mendefinisikan sebuah objek dalam Povray, daripada menggunakan sebuah string dalam Euler. Definisi ditulis ke file secara langsung.

Koordinat sebuah kotak dengan nilai -1 hanya berarti [-1,-1,-1].

```
>povdefine("mycube", povbox(-1,1));
```

Kita dapat menggunakan objek ini dalam povobject(), yang mengembalikan sebuah string seperti biasanya.

```
>c1=povobject("mycube", povlook(red));
```

Kami menghasilkan sebuah kubus kedua, lalu memutar dan mengubah ukurannya sedikit.

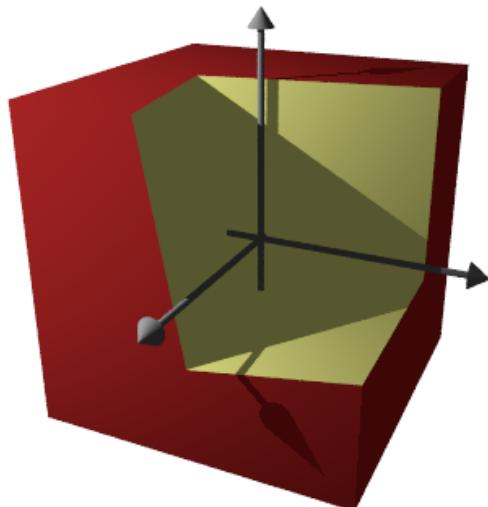
```
>c2=povobject("mycube", povlook(yellow), translate=[1,1,1], ...
>    rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita mengambil 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 dalam plot3d. Hasilnya terlihat jauh lebih baik, meskipun sintaks untuk fungsi-fungsi ini sedikit berbeda. Anda tidak dapat menggunakan hasil 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))*pow(pow(y,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);
```

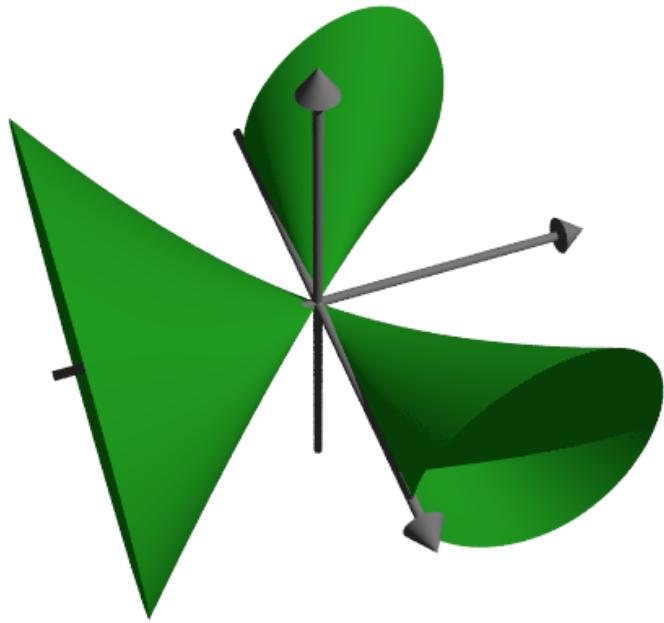
```
>povstart(angle=25°,height=10°);
>writeln(povsurface("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,2,""));
>povend();
```



```
>povstart(angle=70°,height=50°,zoom=4);
```

Buatlah permukaan implisit. Perhatikan sintaks yang berbeda dalam ungkapan ini.

```
>writeln(povsurface ("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...
>writeAxes(); ...
>povend();
```



Objek Jaringan (Mesh)

Dalam contoh ini, kami akan menunjukkan bagaimana membuat objek jaringan (mesh) dan menggambar objek tersebut dengan informasi tambahan.

Kami ingin memaksimalkan nilai xy dengan syarat $x+y=1$ dan menunjukkan sentuhan tangensial pada garis-garis level.

```
>povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kita tidak dapat menyimpan objek dalam string seperti sebelumnya, karena terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan declare. Fungsi povtriangle() melakukan ini secara otomatis. Ia dapat menerima vektor normal seperti pov3d().

Berikut ini definisi objek mesh, dan segera 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 mendefinisikan dua cakram, yang akan berpotongan dengan permukaan.

```
>cl=povdisc([0.5,0.5,0],[1,1,0],2); ...
>l1=povdisc([0,0,1/4],[0,0,1],2);
```

Tulis permukaannya dikurangi kedua cakram.

```
>writeln(povdifference(mesh,povunion([cl,l1]),povlook(green)));
```

Tuliskan kedua perpotongan tersebut.

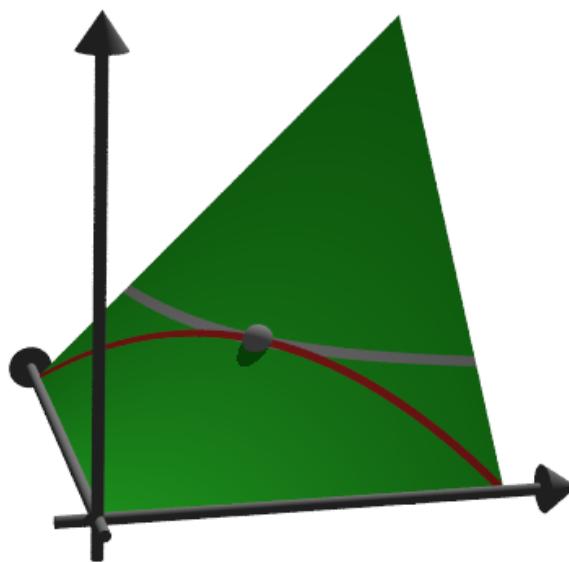
```
>writeln(povintersection([mesh,cl],povlook(red))); ...
>writeln(povintersection([mesh,l1],povlook(gray)));
```

Tulis poin maksimal.

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));
```

Tambahkan sumbu dan selesai.

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...
>povend();
```

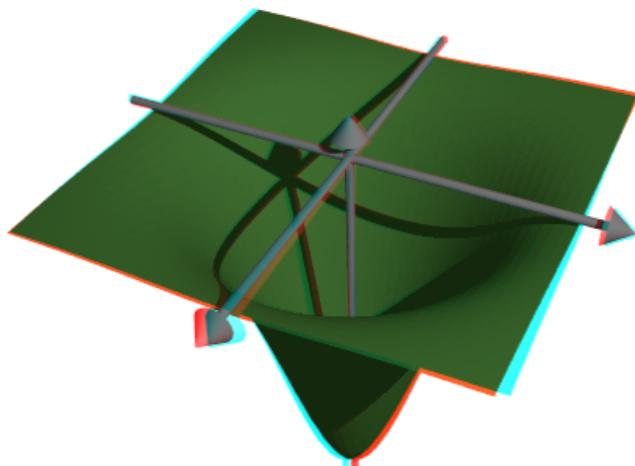


Anaglyph di Povray

Untuk menghasilkan anaglyph untuk kacamata merah/cyan, Povray harus dijalankan dua kali dari posisi kamera berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph().

Tentu saja, Anda memerlukan kacamata berwarna merah/cyan untuk melihat contoh berikut 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);
```

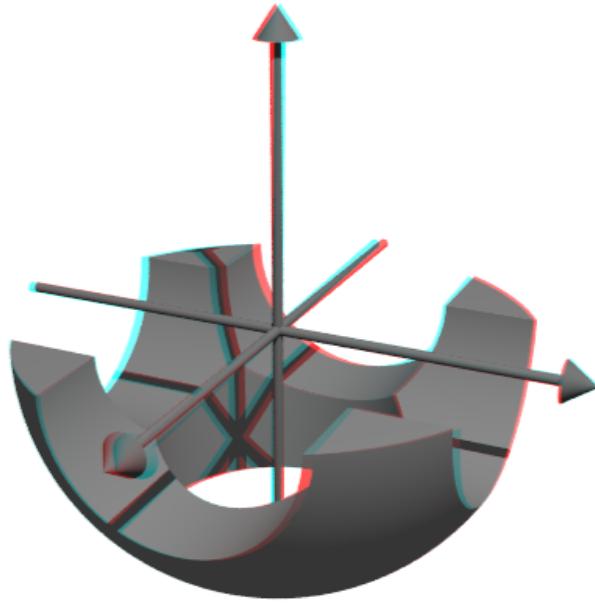


Jika Anda membuat adegan dengan objek, Anda perlu memasukkan pembuatan adegan ke dalam fungsi, dan menjalankannya dua kali dengan nilai berbeda untuk parameter anaglyph.

```
>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. Parameternya seperti gabungan povstart() dan povend().

```
>povanaglyph("myscene",zoom=4.5);
```



Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak objek. Namun Anda tidak dibatasi pada hal ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau merupakan objek yang benar-benar baru.

Kami mendemonstrasikan torus. Perintah Povray untuk ini adalah "torus". Jadi kami mengembalikan string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat pada titik asal.

```
>function povdonat (r1,r2,look "") ...
```

```
    return "torus {" + r1 + "," + r2 + look + "}";
endfunction
```

Ini 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 objek-objek tersebut ke dalam sebuah adegan. 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, kesalahan tersebut tidak ditampilkan. Oleh karena itu Anda harus menggunakan

```
>povend(<keluar);
```

jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray terbuka.

```
>povend(h=320,w=480);
```



Berikut adalah contoh yang lebih rumit. Kami memecahkannya

$$Ax \leq b, \quad x \geq 0, \quad c.x \rightarrow \text{Max.}$$

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 punya solusinya.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0, 1, 0.5]
```

Ya, sudah.

Selanjutnya kita mendefinisikan dua objek. Plane adalah yang pertama

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look="") ...
```

```
    return povplane(a,b,look)
endfunction
```

Kemudian kita mendefinisikan perpotongan semua setengah ruang dan sebuah 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 dapat merencanakan adegannya.

```
>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 sekitar optimal.

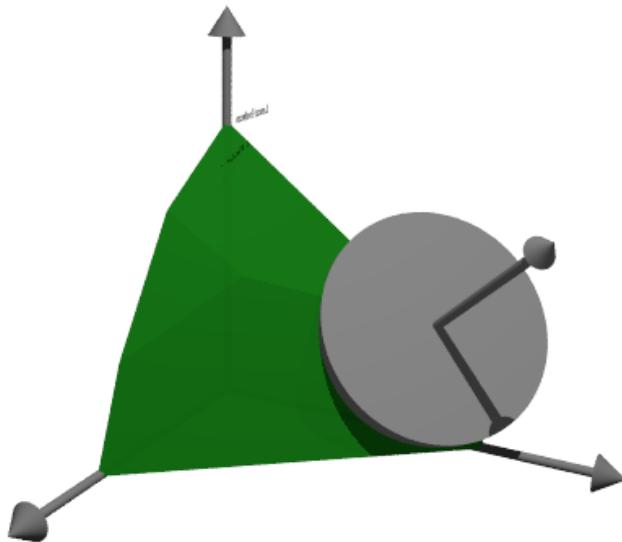
```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
>    povlook(red,0.9)));
```

Dan kesalahan ke arah optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah 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();
```



Lebih Banyak Contoh

Anda dapat menemukan beberapa contoh Povray di Euler di file berikut.

See: Examples/Dandelin Spheres

See: Examples/Donut Math

See: Examples/Trefoil Knot

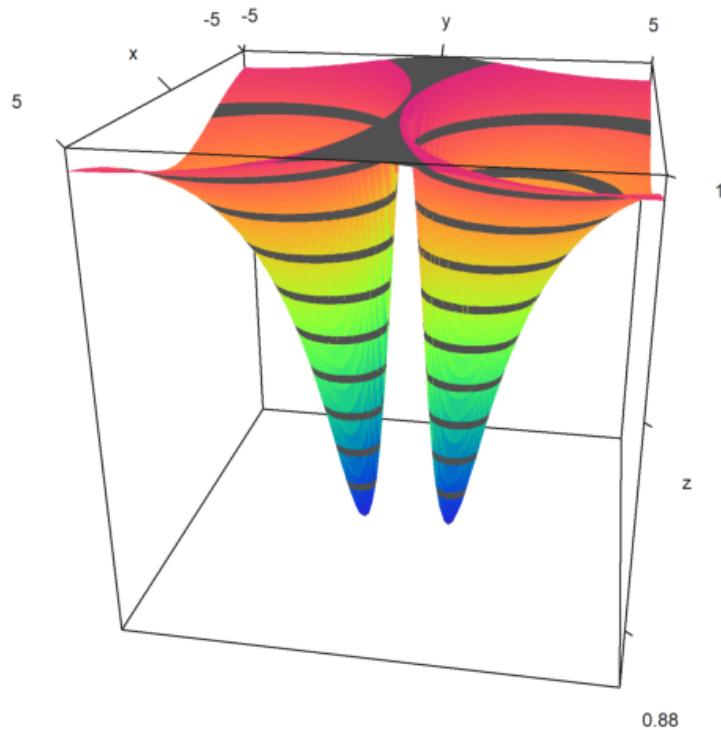
See: Examples/Optimization by Affine Scaling

Contoh Soal

1. Sketsakan plot kontur untuk fungsi berikut.

$$z = -1 + \cos\left(\frac{1}{1+x^2+y^2}\right)$$

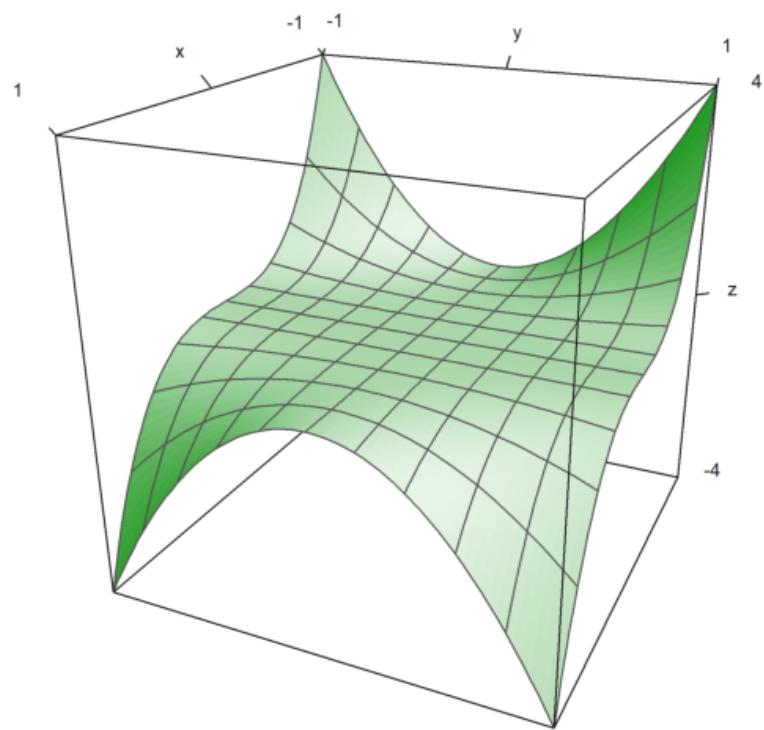
```
>plot3d("exp(-1+cos(y/(1+x^2+y^2)))",r=5,n=100,level="thick", ...
>>contour,>spectral,angle=100°):
```



2. Sketsakan plot kontur untuk fungsi berikut.

$$z = -4x^3y^2$$

```
>plot3d("-4*x^3*y^2",color=green,hue=true,grid=10):
```



3. Sketsakan bentuk berikut.

$$\frac{-4x^2 + 4y^2}{2} + 100$$

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

```
>pov3d("-exp( (-4*x^2-4*y^2) /2)+100", zoom=4);
```



```
>reset();
```

BAB 5

PEKAN 9-10: MENGGUNAKAN EMT UNTUK KALKULUS

article

eumat

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 beberapa contoh cara mendefinisikan fungsi:

$$f(x) = 2x^2 + e^{\sin(x)}.$$

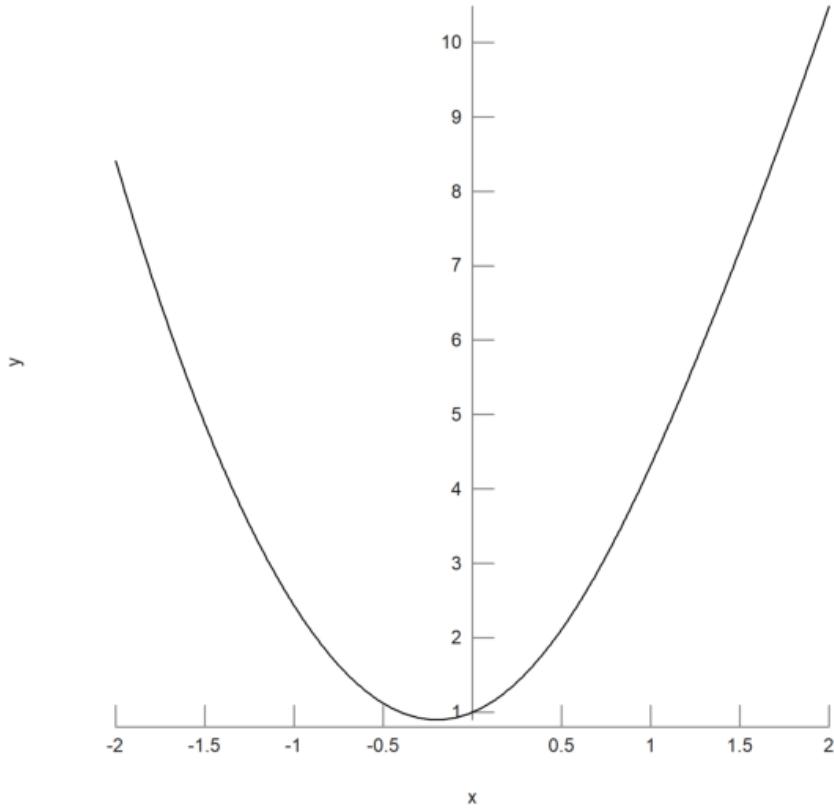
```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
```

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 ...
^

```
>plot2d("2*x^2+exp(sin(x))",grid=7,xl="x",yl="y"):
```



Silakan Anda plot kurva fungsi di atas!

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

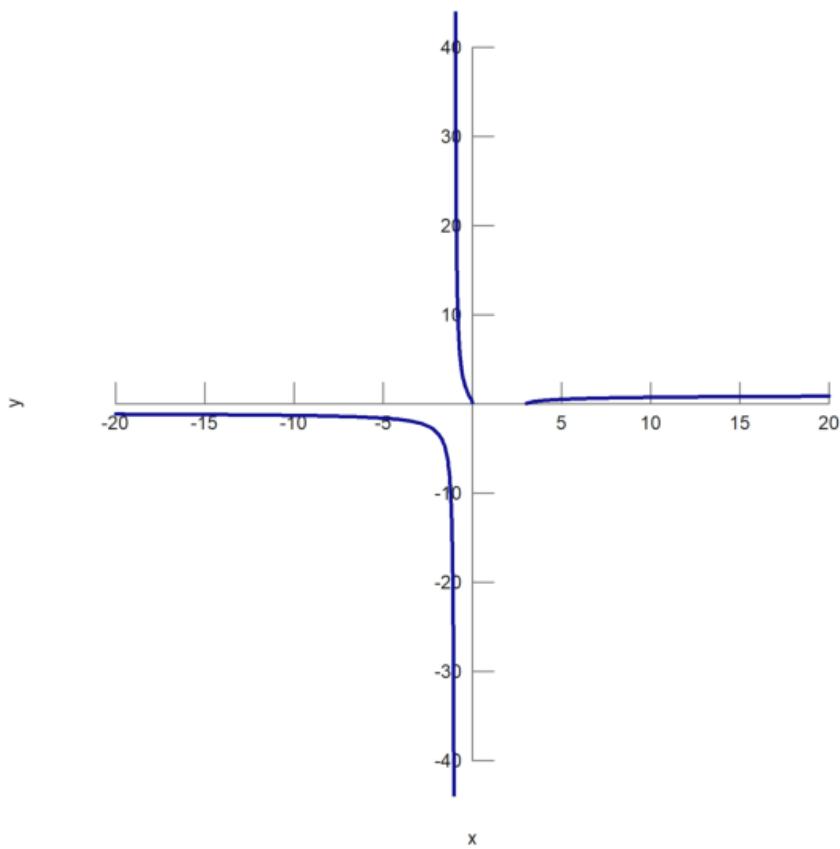
```
>g(0)
```

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 ...
^
```

```
>plot2d("sqrt(x^2-3*x) / (x+1)", -20, 20, -40, 40, grid=7, color=blue, thickness=2, xl="x", yl="y"):
```



Silakan Anda plot kurva fungsi di atas!

```
>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.

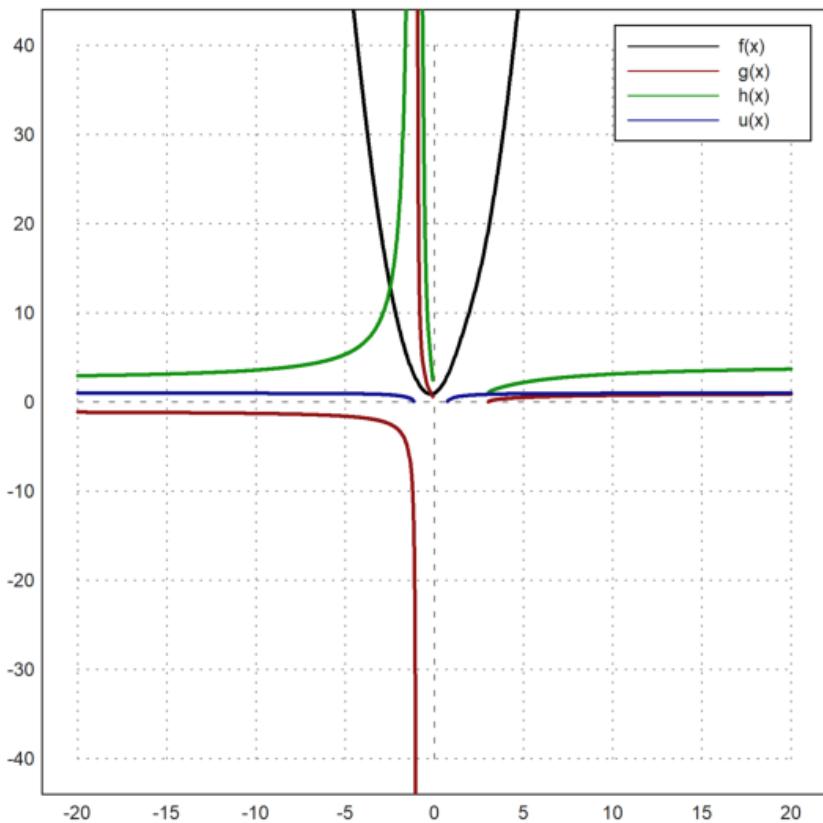
```
>function f(x) := 2*x^2+exp(sin(x));  
>function g(x) := sqrt(x^2-3*x)/(x+1);  
>function h(x) &= f(g(x))
```

f(g(x))

```
>function u(x) &= g(f(x))
```

g(f(x))

```
>plot2d(["f", "g", "h", "u"], -20, 20, -40, 40, color=1:4, thickness=2); ...  
>labelbox(["f(x)", "g(x)", "h(x)", "u(x)"], styles="-", ...  
> colors=1:4, w=0.25);
```



```
>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
```

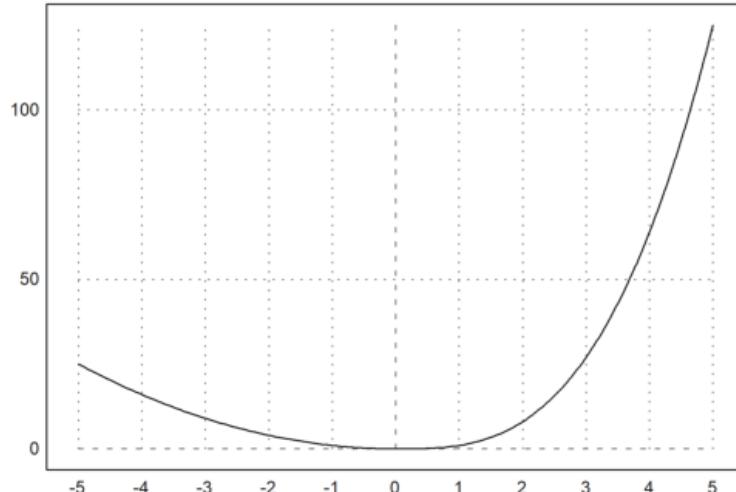
```
>f(-2)
```

```
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
```

$$2^{\frac{x}{E}}$$

```
>$f(a) // nilai fungsi secara simbolik
```

$$2e^a$$

```
>f(E) // nilai fungsi berupa bilangan desimal
```

30.308524483

```
>$f(E), $float(%)
```

30.30852448295852

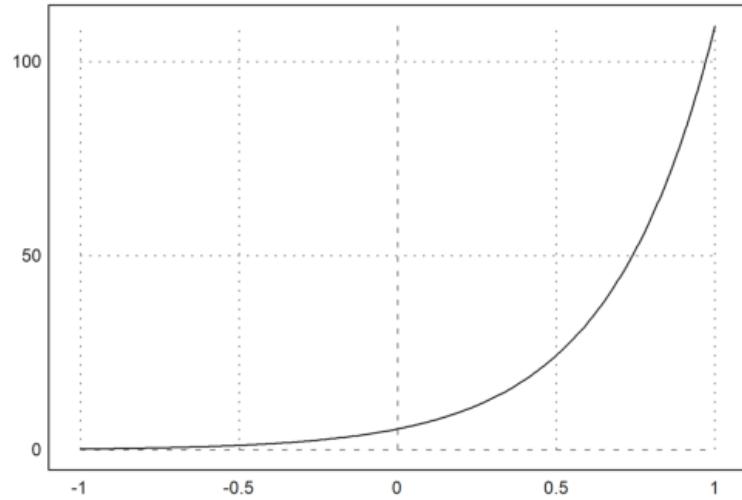
```
>function g(x) &= 3*x+1
```

3 x + 1

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

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

```
>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

$$\text{untuk } f(x) = \frac{x}{x-1} \text{ dan } g(x) = \sqrt{1+x^2}$$

```
>function f(x) &= x/(x-1);
>function g(x) &= sqrt(1+x^2);
>function h(x) &= f(g(x))
```

$$\begin{aligned} &\frac{\sqrt{x^2 + 1}}{x^2 - 1} \\ &----- \\ &\frac{\sqrt{x^2 + 1} - 1}{x^2 - 1} \end{aligned}$$

```
>function u(x) &= g(f(x))
```

$$\begin{aligned} &\frac{x^2}{(x^2 - 1)^{1/2} + 1} \\ &----- \\ &(x^2 - 1)^{1/2} \end{aligned}$$

```
>f(2), f(3), f(5)
```

2
1.5
1.25

```
>g(1), g(4), g(7)
```

1.41421356237
4.12310562562
7.07106781187

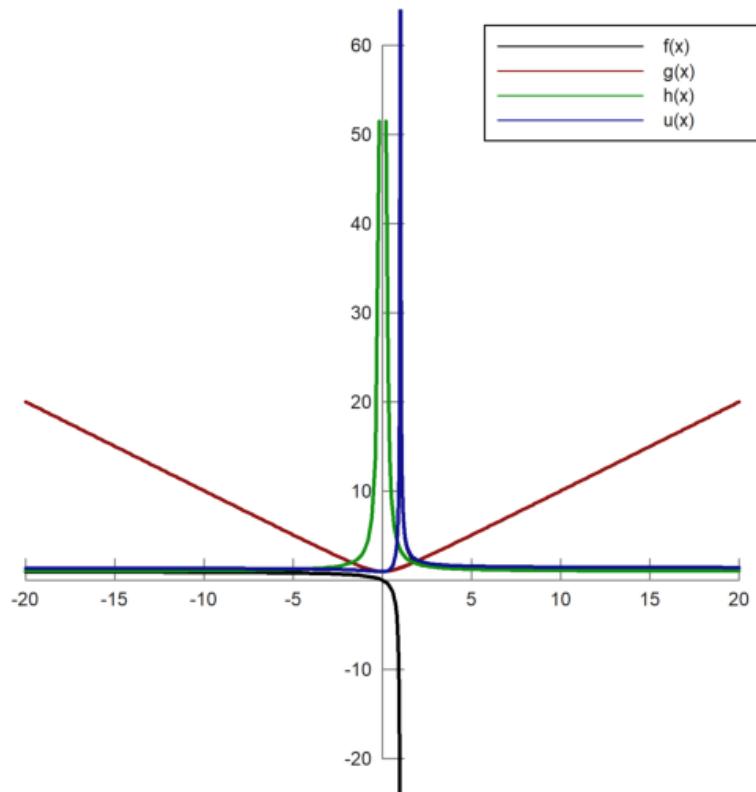
```
>f(g(sqrt(8)))
```

1.5

```
>g(f(1/10))
```

1.00615390424

```
>aspect(1); plot2d(["f","g","h","u"],-20,20,-20,60,color=1:4,thickness=2,grid=7); ...  
>labelbox(["f(x)","g(x)","h(x)","u(x)"],styles="-", ...  
> colors=1:4,w=0.35):
```



Soal 2

untuk $f(x) = x^2 + x$ dan $g(x) = \frac{2}{x+3}$

```
>function f(x) &= x^2+x;
>function g(x) &= 2/(x+3);
>function h(x) &= f(g(x))
```

$$\frac{x^2}{x+3} + \frac{4}{(x+3)^2}$$

```
>function u(x) &= g(f(x))
```

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

```
>f(3), f(4), f(7)
```

12
20
56

```
>f(5), f(8), f(9)
```

30
72
90

```
>f(g(1))
```

0.75

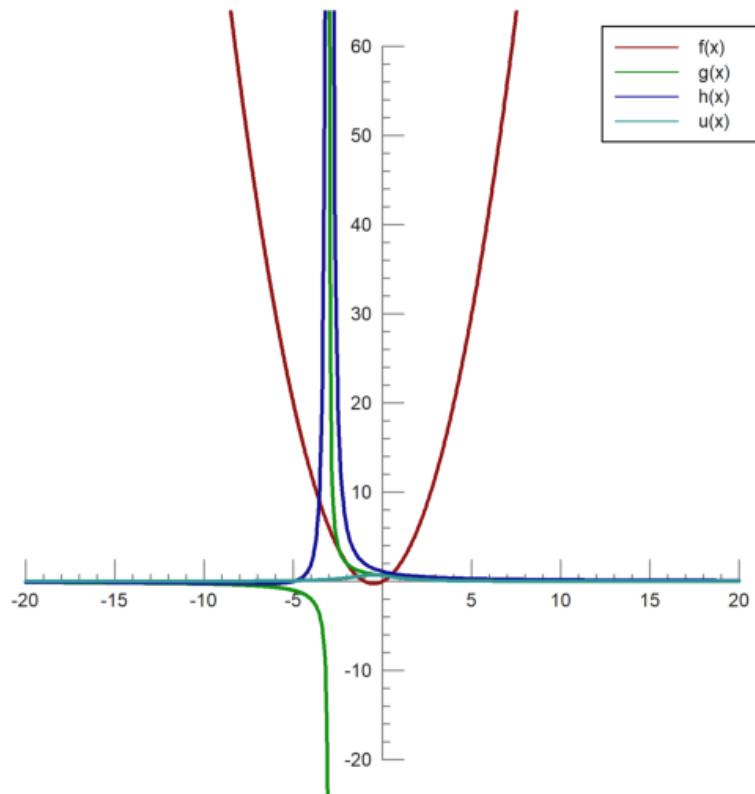
```
>g(f(-5))
```

0.0869565217391

```

>aspect(1); plot2d(["f","g","h","u"],-20,20,-20,60,color=2:5,thickness=2,grid=8); ...
>labelbox(["f(x)","g(x)","h(x)","u(x)"],styles="-", ...
>    colors=2:5,w=0.2):

```



Soal 3

untuk $f(x) = x^3 + 2x + 2$ dan $g(x) = \frac{2}{x-1}$

```

>function f(x) &= x^3+2*x+2;
>function g(x) &= 2/(x-1);
>function h(x) &= f(g(x))

```

$$\frac{4}{x-1} + \frac{8}{(x-1)^3} + 2$$

```
>function u(x) &= g(f(x))
```

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

```
>f(7), f(1), f(0)
```

359
5
2

```
>g(1/2), g(sqrt(4)), g(5)
```

-4
2
0.5

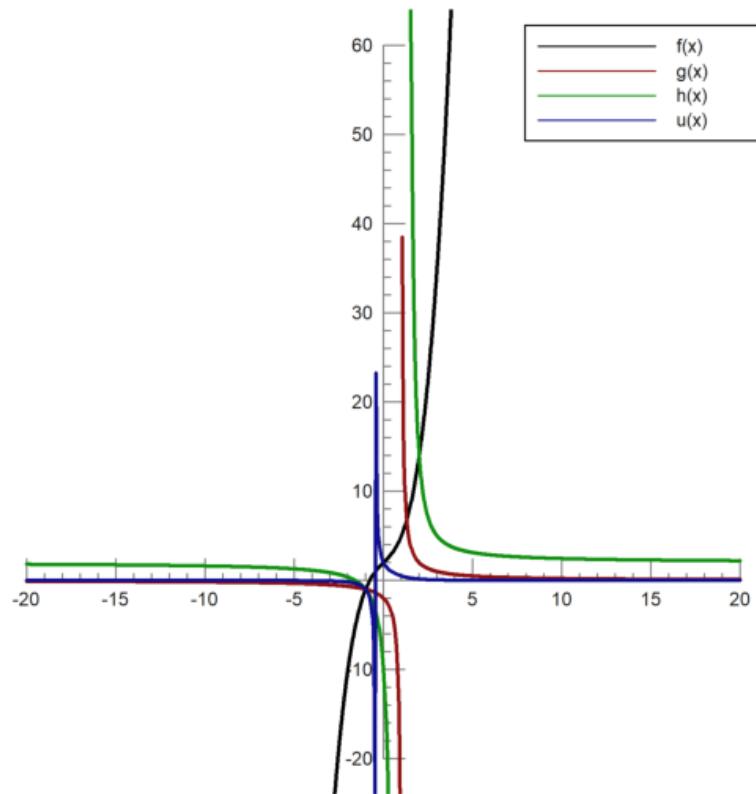
```
>f(g(sqrt(1/2)))
```

-330.048773235

```
>g(f(5/2))
```

0.0924855491329

```
>aspect(1); plot2d(["f","g","h","u"], -20, 20, -20, 60, color=1:4, thickness=2, grid=8); ...
>labelbox(["f(x)", "g(x)", "h(x)", "u(x)"], styles="-", ...
>    colors=1:4, w=0.3);
```



Soal 4

untuk $f(x) = \sqrt{x^2 - 1}$ dan $g(x) = \frac{2}{x}$

```
>function f(x) &= sqrt(x^2-1);
>function g(x) &= 2/x;
>function h(x) &= f(g(x))
```

$$\frac{\sqrt{(\frac{2}{x})^2 - 1}}{x}$$

```
>function u(x) &= g(f(x))
```

$$\frac{2}{\sqrt{x^2 - 1}}$$

```
>f(9), f(4), f(16)
```

```
8.94427191  
3.87298334621  
15.9687194227
```

```
>g(-1/2), g(-9), g(12)
```

```
-4  
-0.222222222222  
0.166666666667
```

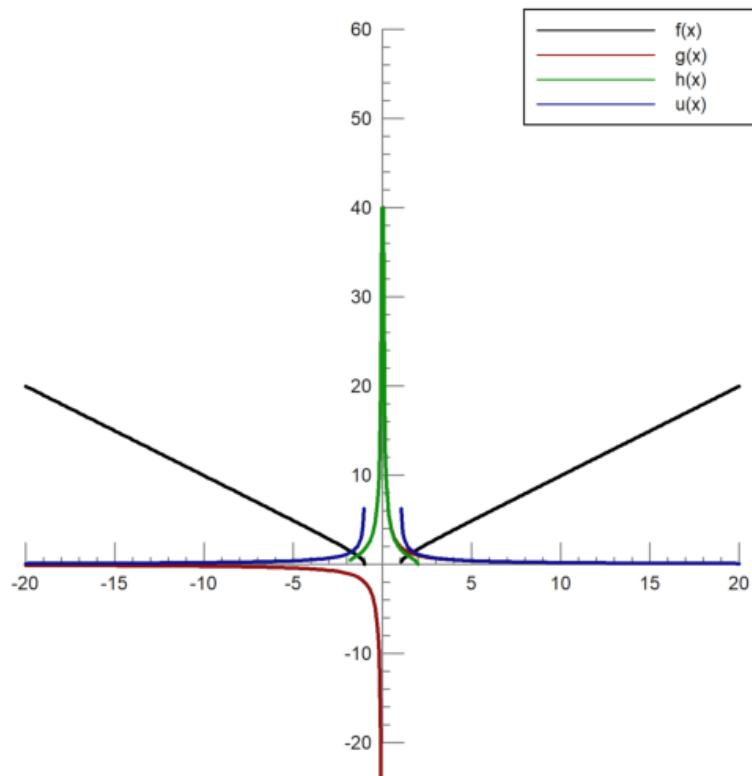
```
>f(g(2))
```

```
0
```

```
>g(f(-7))
```

```
0.288675134595
```

```
>aspect(1); plot2d(["f", "g", "h", "u"], -20, 20, -20, 60, color=1:4, thickness=2, grid=8); ...  
>labelbox(["f(x)", "g(x)", "h(x)", "u(x)"], styles="-", ...  
> colors=1:4, w=0.3):
```



Soal 5

untuk $f(x) = x - 4$ dan $g(x) = \sin(x).\cos(x)$

```
>function f(x) &= x-4;
>function g(x) &= sin(x)*cos(x);
>function h(x) &= f(g(x))
```

$$\cos(x) \sin(x) - 4$$

```
>function u(x) &= g(f(x))
```

$$\cos(x - 4) \sin(x - 4)$$

```
>f(4), f(21), f(17)
```

0
17
13

```
>g(9), g(30), g(45)
```

-0.375493623386
-0.152405310551
0.4469983318

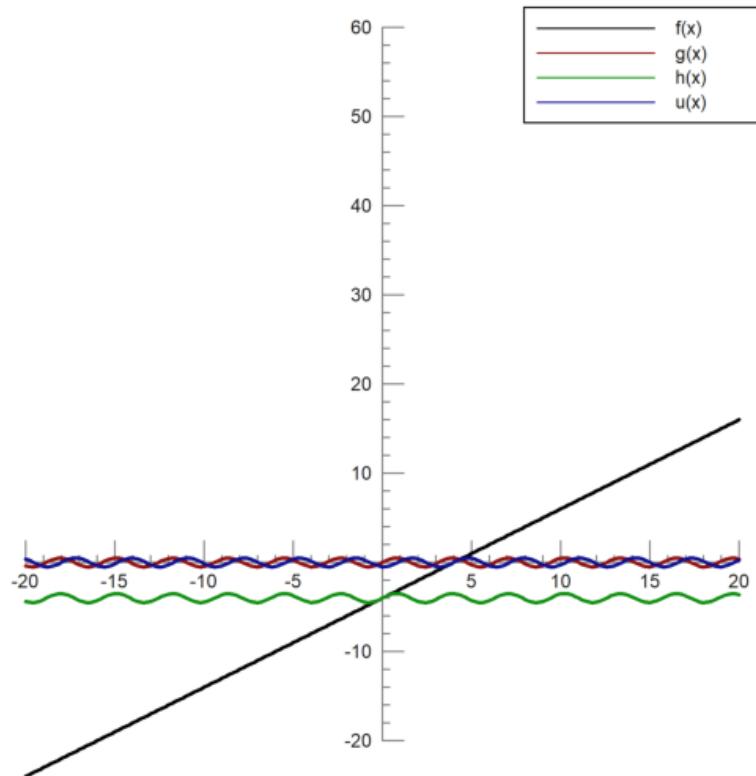
```
>f(g(1/5))
```

-3.80529082885

```
>g(f(30))
```

0.49331379602

```
>aspect(1); plot2d(["f", "g", "h", "u"], -20, 20, -20, 60, color=1:4, thickness=2, grid=8); ...
>labelbox(["f(x)", "g(x)", "h(x)", "u(x)"], styles="-", ...
>    colors=1:4, w=0.3);
```



```
>reset();
```

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))
```

$$\lim_{x \rightarrow \infty} \frac{\sqrt{x^2 - 3x}}{x + 1} = 1$$

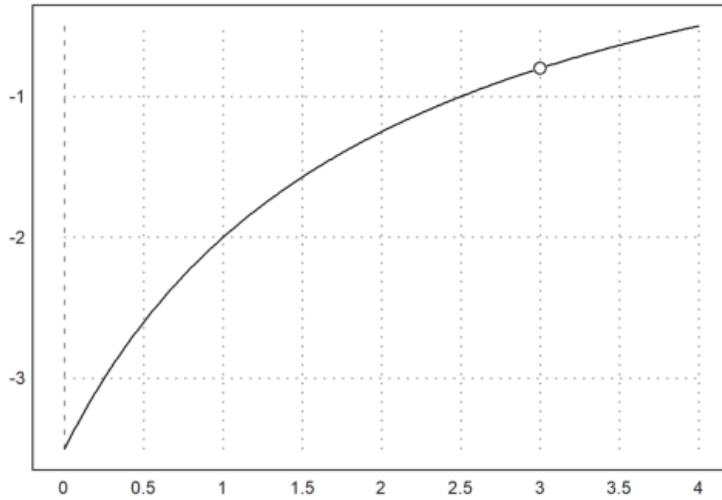
```
>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

$$-\frac{4}{5}$$

$$\lim_{x \rightarrow 3} \frac{x^3 - 13x^2 + 51x - 63}{x^3 - 4x^2 - 3x + 18} = -\frac{4}{5}$$

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(3, -4/5, >points)
```



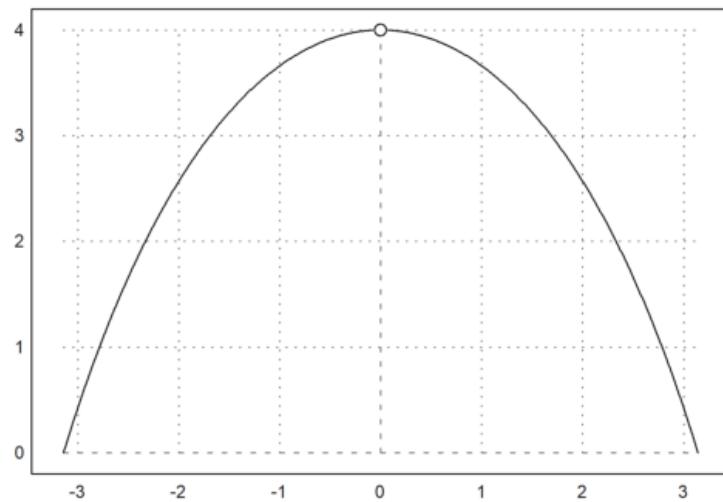
```
>$limit(2*x*sin(x)/(1-cos(x)), x, 0)
```

4

$$2 \left(\lim_{x \rightarrow 0} \frac{x \sin x}{1 - \cos x} \right) = 4$$

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) :
```



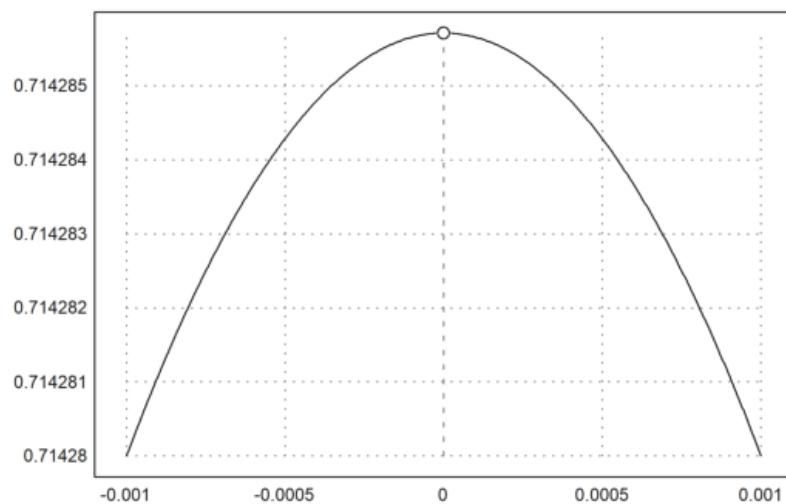
```
> $limit(cot(7*h)/cot(5*h), h, 0)
```

$$\frac{5}{7}$$

$$\lim_{h \rightarrow 0} \frac{\cot(7h)}{\cot(5h)} = \frac{5}{7}$$

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", >add):
```

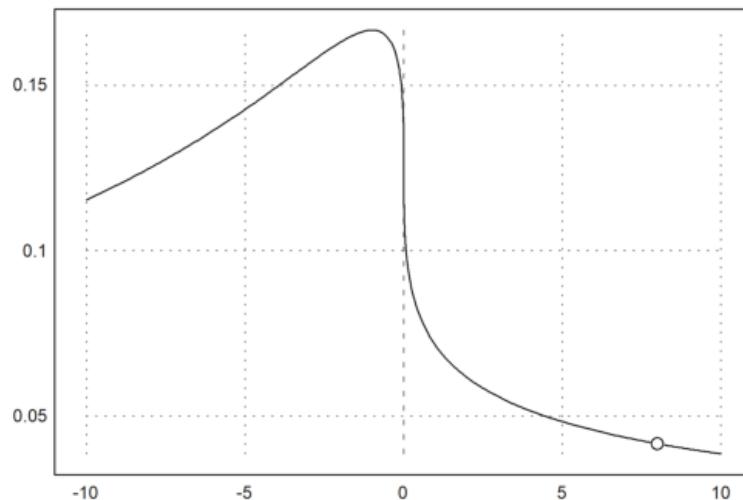


```
> $showev('limit((x/8)^(1/3)-1)/(x-8), x, 8)
```

$$\lim_{x \rightarrow 8} \frac{\frac{x^{\frac{1}{3}}}{2} - 1}{x - 8} = \frac{1}{24}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x/8)^(1/3)-1)/(x-8)", -10, 10); plot2d(8, 1/24, >points, style="ow", >add) :
```

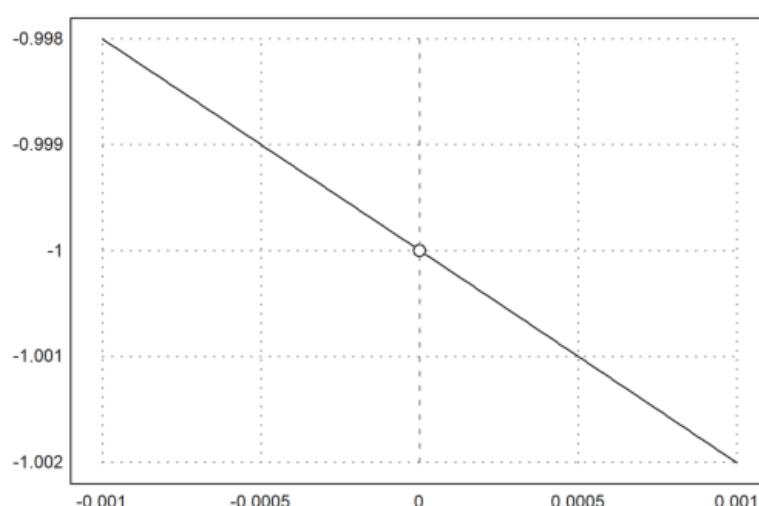


```
>$showev('limit(1/(2*x-1), x, 0))
```

$$\lim_{x \rightarrow 0} \frac{1}{2x - 1} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("1/(2*x-1)", -0.001, 0.001); plot2d(0, -1, >points, style="ow", >add) :
```

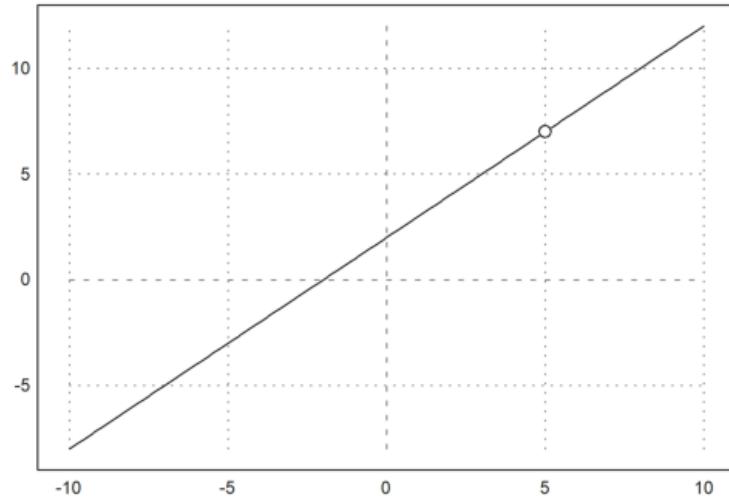


```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

$$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(x^2-3*x-10)/(x-5)",-10,10); plot2d(5,7,>points,style="ow",>add):
```

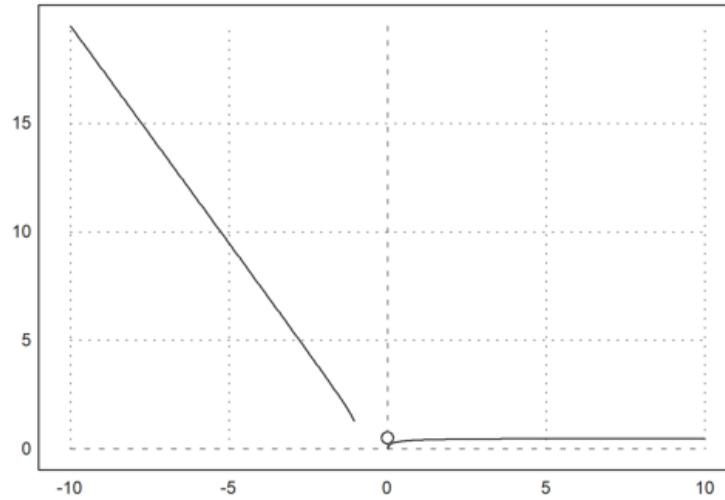


```
>$showev('limit(sqrt(x^2+x)-x,x,inf))
```

$$\lim_{x \rightarrow \infty} \sqrt{x^2 + x} - x = \frac{1}{2}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("sqrt(x^2+x)-x",-10,10); plot2d(0,1/2,>points,style="ow",>add):
```



```
>$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.

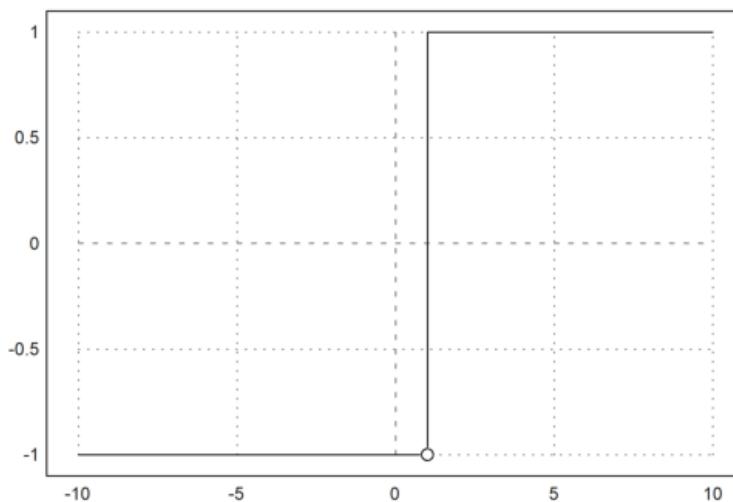
```
>$showev('limit(abs(x-1)/(x-1),x,1,plus))
```

$$\lim_{x \downarrow 1} \frac{|x-1|}{x-1} = 1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

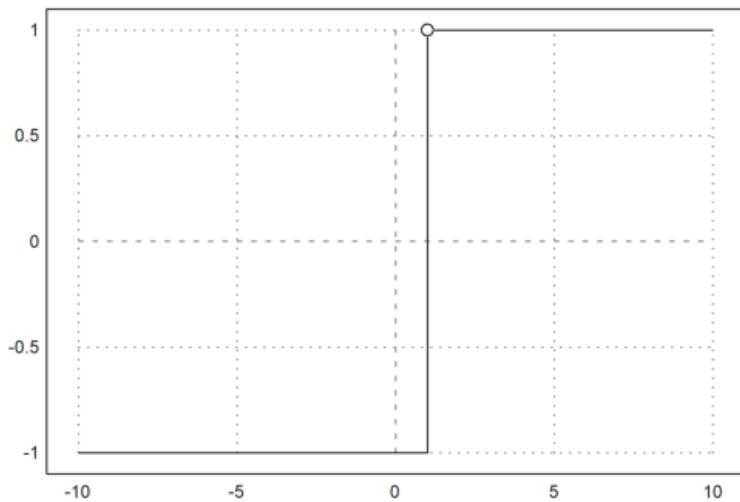
a. Grafik limit untuk x menuju 1 dari kiri.

```
>plot2d("abs(x-1)/(x-1)",-10,10); plot2d(1,-1,>points,style="ow",>add):
```



b. Grafik limit untuk x menuju 1 dari kanan.

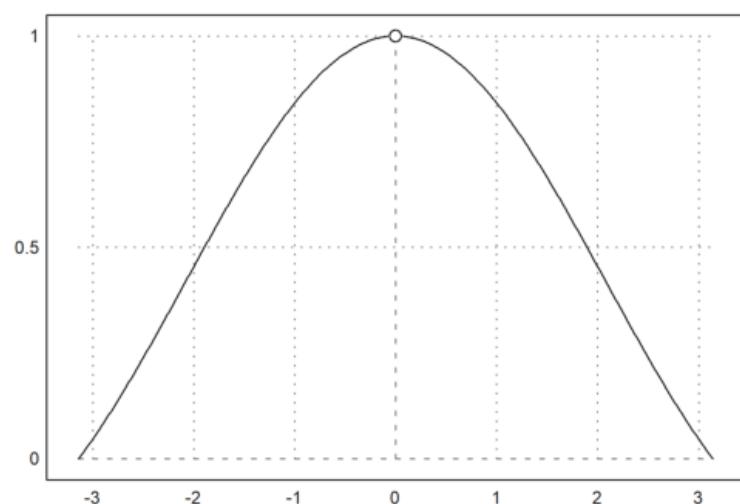
```
>plot2d("abs(x-1)/(x-1)",-10,10); plot2d(1,1,>points,style="ow",>add):
```



```
>$showev('limit(sin(x)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

```
>plot2d("sin(x)/x",-pi,pi); plot2d(0,1,>points,style="ow",>add):
```

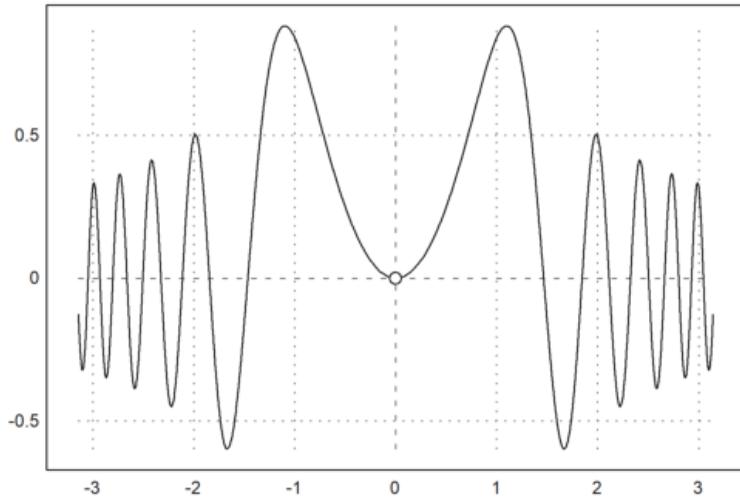


```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("sin(x^3)/x",-pi,pi); plot2d(0,0,>points,style="ow",>add) :
```



```
>$showev('limit(log(x), x, minf))
```

$$\lim_{x \rightarrow -\infty} \log x = \text{infinity}$$

```
>$showev('limit((-2)^x,x, inf))
```

$$\lim_{x \rightarrow \infty} (-2)^x = \text{infinity}$$

```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t \uparrow 2} t - \sqrt{2-t} = 2$$

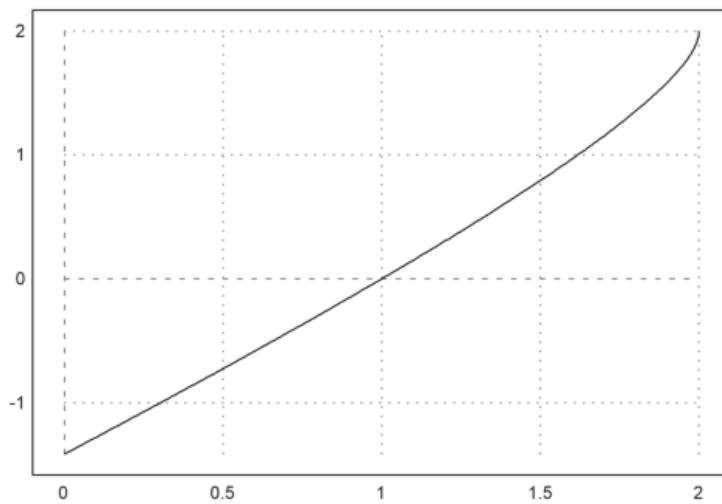
```
>$showev('limit(t-sqrt(2-t),t,2,plus))
```

$$\lim_{t \downarrow 2} t - \sqrt{2-t} = 2$$

```
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
```

$$\lim_{t \downarrow 5} t - \sqrt{2-t} = 5 - \sqrt{3}i$$

```
>plot2d("x-sqrt(2-x)", 0, 2):
```

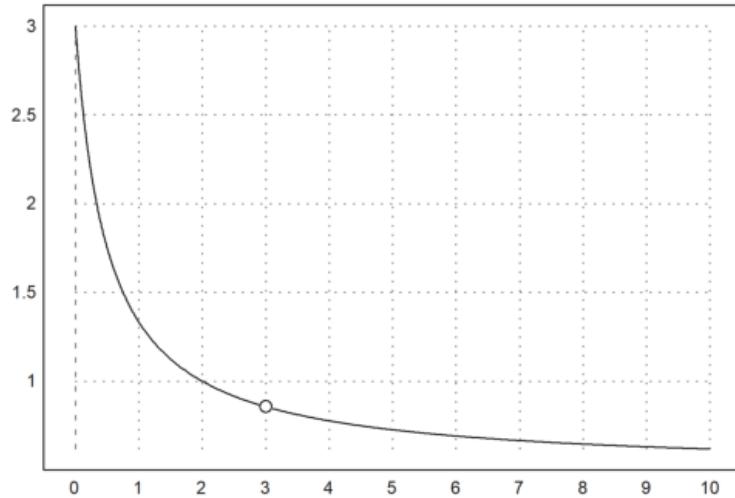


```
>$showev('limit((x^2-9)/(2*x^2-5*x-3), x, 3))
```

$$\lim_{x \rightarrow 3} \frac{x^2 - 9}{2x^2 - 5x - 3} = \frac{6}{7}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(x^2-9)/(2*x^2-5*x-3)", 0, 10); plot2d(3, 6/7, >points, style="ow", >add):
```

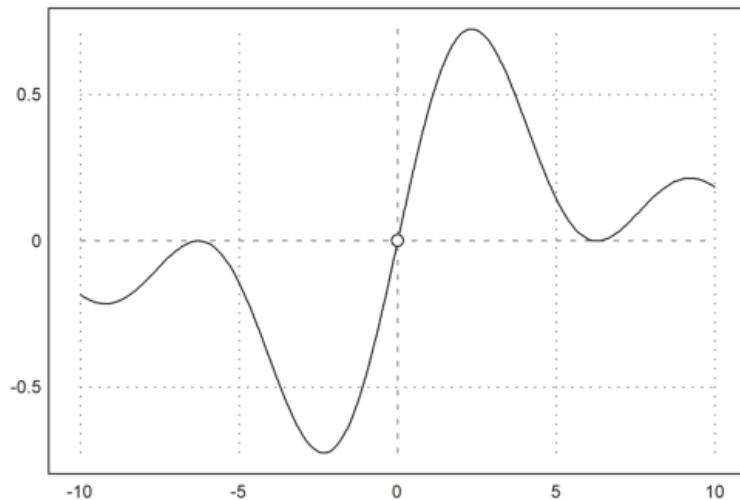


```
>$showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1-cos(x))/x", -10, 10); plot2d(0, 0, >points, style="ow", >add):
```

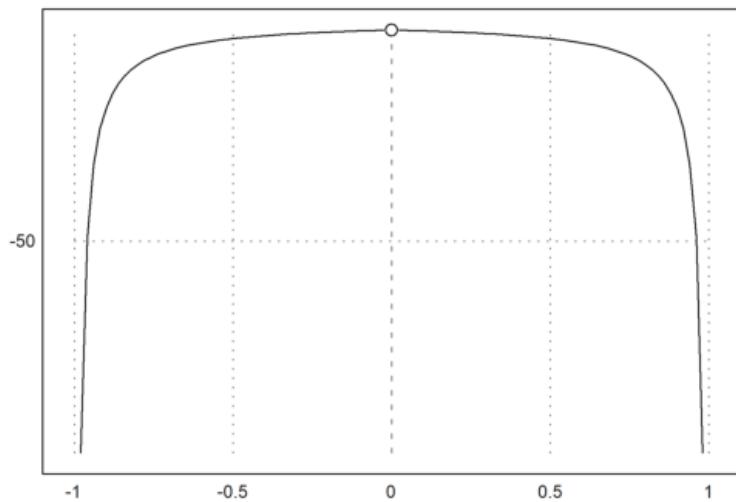


```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

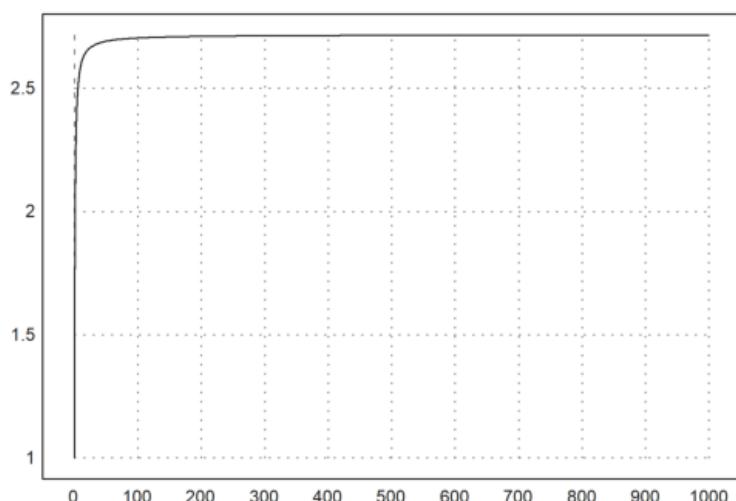
```
>plot2d("(x^2+abs(x))/(x^2-abs(x))", -1, 1); plot2d(0, -1, >points, style="ow", >add) :
```



```
>$showev('limit((1+1/x)^x, x, inf))
```

$$\lim_{x \rightarrow \infty} \left(\frac{1}{x} + 1 \right)^x = e$$

```
>plot2d("(1+1/x)^x", 0, 1000) :
```



```
>$showev('limit((1+k/x)^x,x,inf))
```

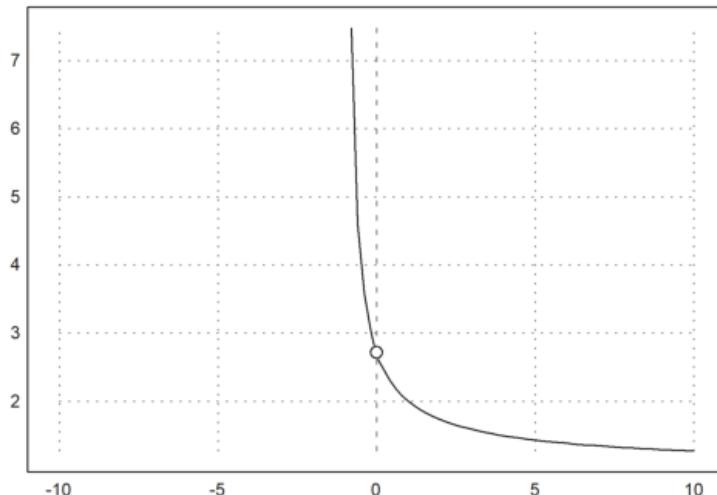
$$\lim_{x \rightarrow \infty} \left(\frac{k}{x} + 1 \right)^x = e^k$$

```
>$showev('limit((1+x)^(1/x),x,0))
```

$$\lim_{x \rightarrow 0} (x+1)^{\frac{1}{x}} = e$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1+x)^(1/x)",-10,10); plot2d(0,E,>points,style="ow",>add):
```



```
>$showev('limit((x/(x+k))^x,x,inf))
```

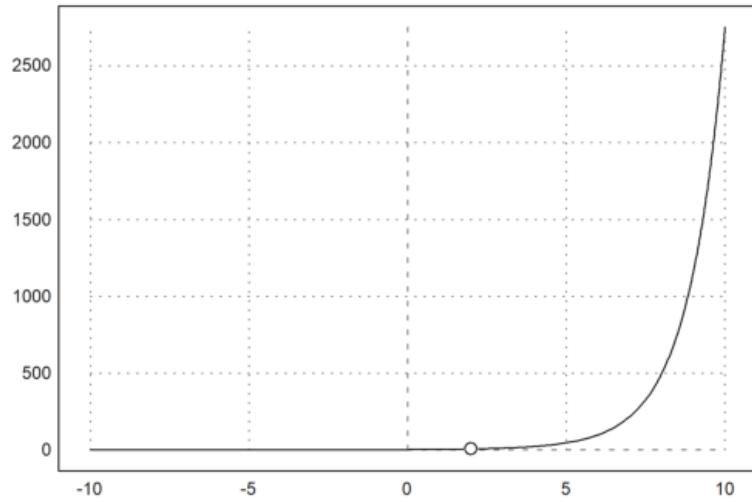
$$\lim_{x \rightarrow \infty} \left(\frac{x}{x+k} \right)^x = e^{-k}$$

```
>$showev('limit((E^x-E^2)/(x-2),x,2))
```

$$\lim_{x \rightarrow 2} \frac{e^x - e^2}{x - 2} = e^2$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(E^x-E^2)/(x-2)",-10,10); plot2d(2,E^2,>points,style="ow",>add):
```



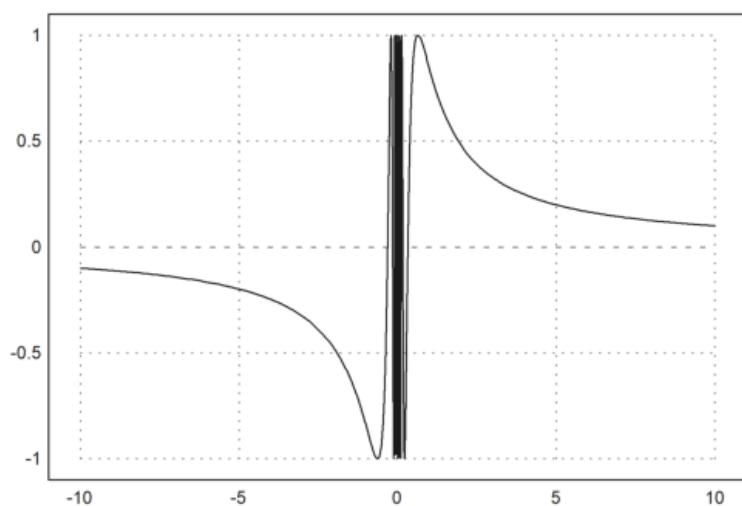
```
>$showev('limit(sin(1/x),x,0))
```

$$\lim_{x \rightarrow 0} \sin\left(\frac{1}{x}\right) = \text{ind}$$

```
>$showev('limit(sin(1/x),x,inf))
```

$$\lim_{x \rightarrow \infty} \sin\left(\frac{1}{x}\right) = 0$$

```
>plot2d("sin(1/x)",-10,10):
```



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.

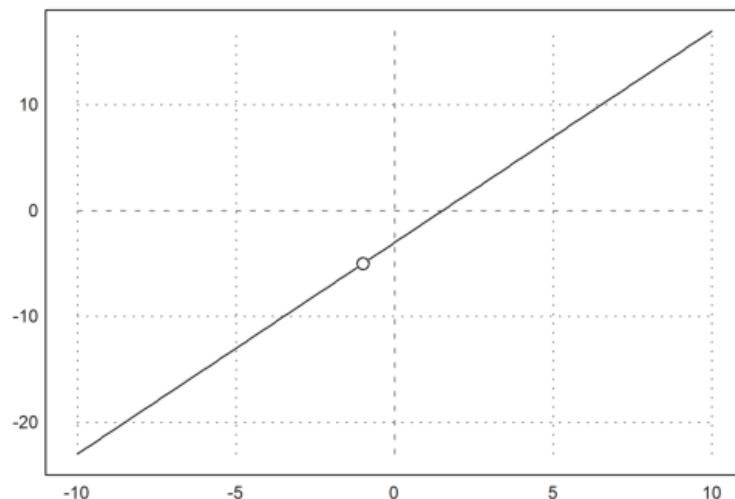
Soal 1

$$\lim_{x \rightarrow -1} \frac{2x^2 - x - 3}{x + 1}$$

```
>$showev('limit((2*x^2-x-3)/(x+1),x,-1))
```

$$\lim_{x \rightarrow -1} \frac{2x^2 - x - 3}{x + 1} = -5$$

```
>plot2d("(2*x^2-x-3)/(x+1)",-10,10); plot2d(-1,-5,>points,style="ow",>add):
```



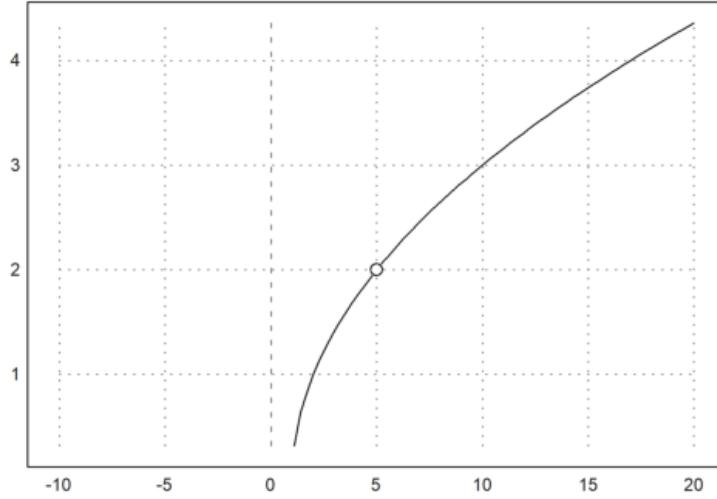
Soal 2

$$\lim_{x \rightarrow 5} \sqrt{x - 1}$$

```
>$showev('limit(sqrt(x-1),x,5))
```

$$\lim_{x \rightarrow 5} \sqrt{x - 1} = 2$$

```
>plot2d("sqrt(x-1)",-10,20); plot2d(5,2,>points,style="ow",>add):
```



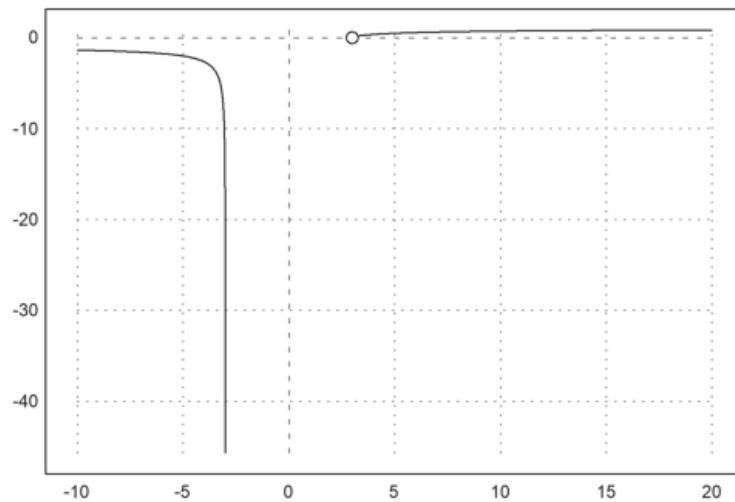
Soal 3

$$\lim_{x \downarrow 3} \frac{x-3}{\sqrt{x^2-9}}$$

```
>$showev('limit((x-3)/(sqrt(x^2-9)),x,3,plus))
```

$$\lim_{x \downarrow 3} \frac{x-3}{\sqrt{x^2-9}} = 0$$

```
>plot2d("(x-3)/(sqrt(x^2-9))", -10, 20); plot2d(3, 0, >points, style="ow", >add):
```



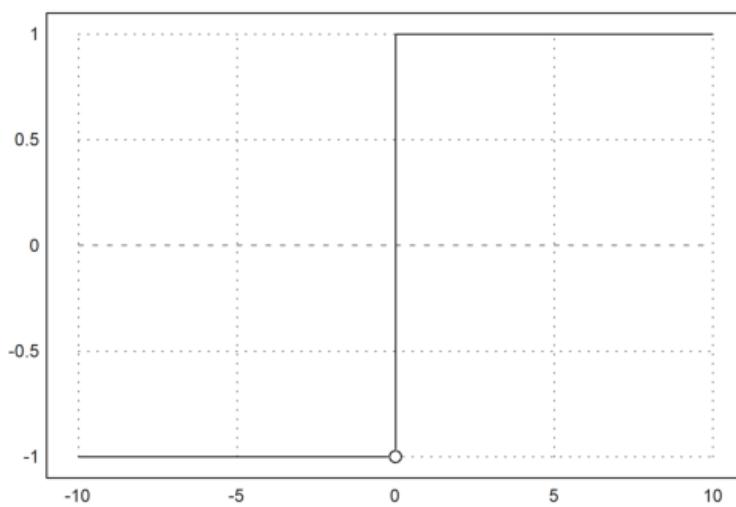
Soal 4

$$\lim_{x \uparrow 0} \frac{x}{|x|}$$

```
>showev('limit(x/abs(x),x,0,minus))
```

$$\lim_{x \uparrow 0} \frac{x}{|x|} = -1$$

```
>plot2d("x/abs(x)",-10,10); plot2d(0,-1,>points,style="ow",>add):
```



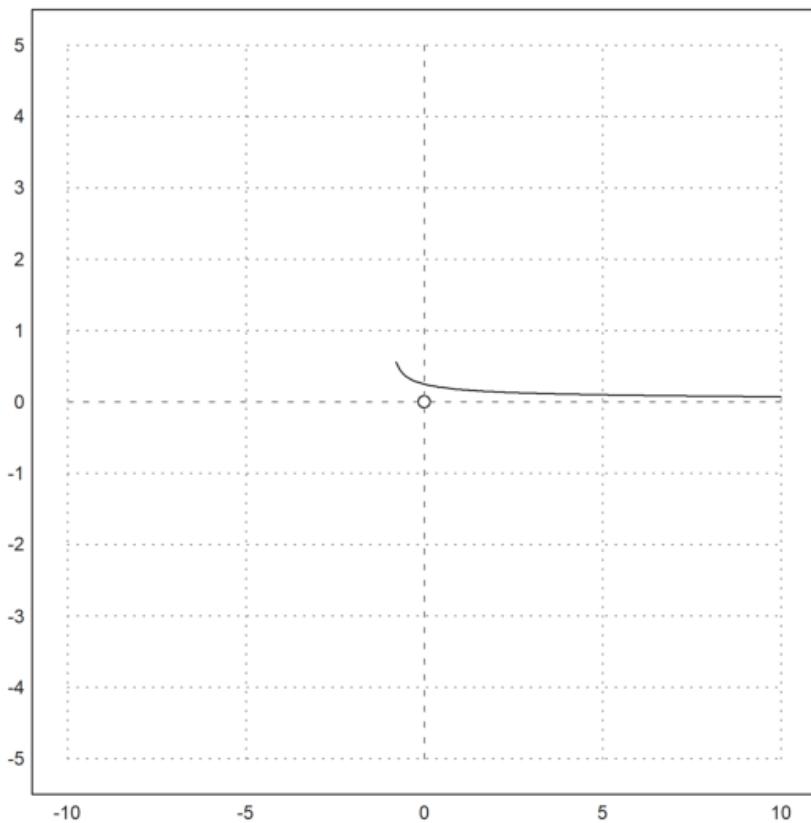
Soal 5

$$\lim_{x \rightarrow \infty} \frac{\sqrt{x+1}}{4x+4}$$

```
>showev('limit(sqrt(1+x)/(4+4*x),x,inf))
```

$$\lim_{x \rightarrow \infty} \frac{\sqrt{x+1}}{4x+4} = 0$$

```
>aspect(1); plot2d("sqrt(1+x)/(4+4*x)",-10,10,-5,5); plot2d(0,0,>points,style="ow",>add):
```



```
>reset();
```

Turunan Fungsi

Definisi turunan:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

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
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2}{h} = 2x$$

```
>p &= expand((x+h)^2-x^2)|simplify; $p // pembilang dijabarkan dan disederhanakan
```

$$2hx + h^2$$

```
>q &=ratsimp(p/h); $q // ekspresi yang akan dihitung limitnya disederhanakan
```

$$2x + h$$

```
>$limit(q,h,0) // nilai limit sebagai turunan
```

$$2x$$

```
>$showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} = nx^{n-1}$$

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.

$$\begin{aligned}(x+h)^n &= x^n + \binom{n}{1}x^{n-1}h + \binom{n}{2}x^{n-2}h^2 + \dots + h^n \\ \lim_{h \rightarrow 0} \frac{x^n + \binom{n}{1}x^{n-1}h + \binom{n}{2}x^{n-2}h^2 + \dots + h^n - x^n}{h} &= \\ \lim_{h \rightarrow 0} \frac{\binom{n}{1}x^{n-1}h + \binom{n}{2}x^{n-2}h^2 + \dots + h^n}{h} &= \\ \lim_{h \rightarrow 0} \binom{n}{1}x^{n-1} + \binom{n}{2}x^{n-2}h + \dots + h^{n-1} &= \\ \binom{n}{1}x^{n-1} &= \\ nx^{n-1} &\end{aligned}$$

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

$$\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h} = \cos x$$

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.

$$\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin(x)}{h}$$

$$\sin(x+h) = \sin(x)\cos(h) + \cos(x)\sin(h)$$

$$\lim_{h \rightarrow 0} \frac{(\sin(x)\cos(h) + \cos(x)\sin(h)) - \sin(x)}{h}$$

$$\lim_{h \rightarrow 0} \frac{\sin(x)(\cos(h) - 1) + \cos(x)\sin(h)}{h}$$

$$\lim_{h \rightarrow 0} (\sin(x) \frac{\cos(h) - 1}{h} + \cos(x) \frac{\sin(h)}{h})$$

$$\lim_{h \rightarrow 0} (\sin(x) \frac{\cos(h) - 1}{h} + \cos(x) \frac{\sin(h)}{h})$$

$$\sin(x) \lim_{h \rightarrow 0} \frac{\cos(h) - 1}{h} + \cos(x) \lim_{h \rightarrow 0} \frac{\sin(h)}{h}$$

$$\sin(x).0 + \cos(x).1$$

$$\cos(x)$$

```
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
```

$$\lim_{h \rightarrow 0} \frac{\log(x+h) - \log x}{h} = \frac{1}{x}$$

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.

$$\log(x+h) - \log(x) = \log\left(\frac{x+h}{x}\right)$$

$$\lim_{h \rightarrow 0} \frac{\log\left(\frac{x+h}{x}\right)}{h}$$

$$\lim_{h \rightarrow 0} \frac{\log(1 + \frac{h}{x})}{h}$$

$$\lim_{h \rightarrow 0} \frac{\log(1 + \frac{h}{x})}{h} = \frac{1}{x}$$

```
>$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x
```

$$\lim_{h \rightarrow 0} \frac{\frac{1}{x+h} - \frac{1}{x}}{h} = -\frac{1}{x^2}$$

```
>&assume(x>0); $showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x
```

Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is x an integer?

Use assume!
Error in:
... ssume(x>0); \$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(...
^

Maxima bermasalah dengan limit:

$$\lim_{h \rightarrow 0} \frac{e^{x+h} - e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{e^h - 1}{h} = 1$$

```
>$showev('factor(E^(x+h)-E^x))
```

$$factor(e^{x+h} - e^x) = (e^h - 1) e^x$$

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

$$\left(\lim_{h \rightarrow 0} \frac{e^h - 1}{h} \right) e^x = e^x$$

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

$$\begin{matrix} x \\ x \end{matrix}$$

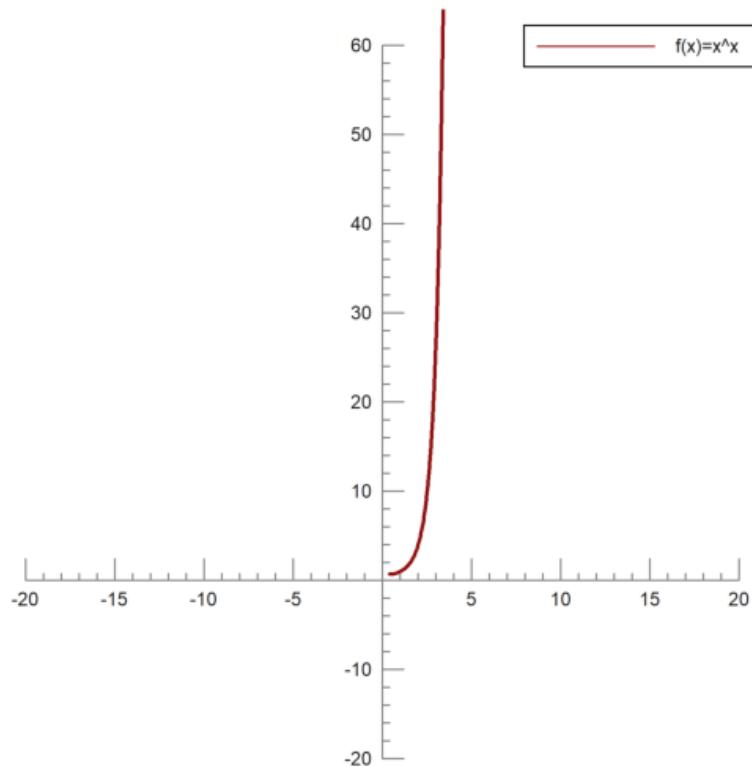
```
>$showev('limit(f(x),x,0))
```

$$\lim_{x \rightarrow 0} x^x = 1$$

Silakan Anda gambar kurva

$$y = x^x.$$

```
>&assume(x>0); aspect(1); plot2d("f",-20,20,-20,60,color=2,thickness=2,grid=8); ...
>labelbox(["f(x)=x^x"],styles="-", ...
> colors=2,w=0.3):
```



```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = x^x (\log x + 1)$$

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h}.$$

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
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = x^x (\log x + 1)$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>&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)
```

$$\lim_{h \rightarrow 0} \frac{\arcsin(x+h) - \arcsin x}{h} = \frac{1}{\sqrt{1-x^2}}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

$$\lim_{h \rightarrow 0} \frac{\tan(x+h) - \tan x}{h} = \frac{1}{\cos^2 x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

$$\tan(A) - \tan(B) = \frac{\sin(A-B)}{\cos(A)\cos(B)}$$

$$\lim_{h \rightarrow 0} \frac{\sin(x+h-x)}{h\cos(x+h)\cos(x)}$$

$$\lim_{h \rightarrow 0} \frac{\sin(h)}{h\cos(x+h)\cos(x)}$$

$$\lim_{h \rightarrow 0} \frac{\sin(h)}{h} \frac{1}{\cos(x)} \frac{1}{\cos(x+h)}$$

$$1 \frac{1}{\cos(x)} \frac{1}{\cos(x)}$$

$$\frac{1}{\cos^2(x)}$$

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

$\sinh(x)$

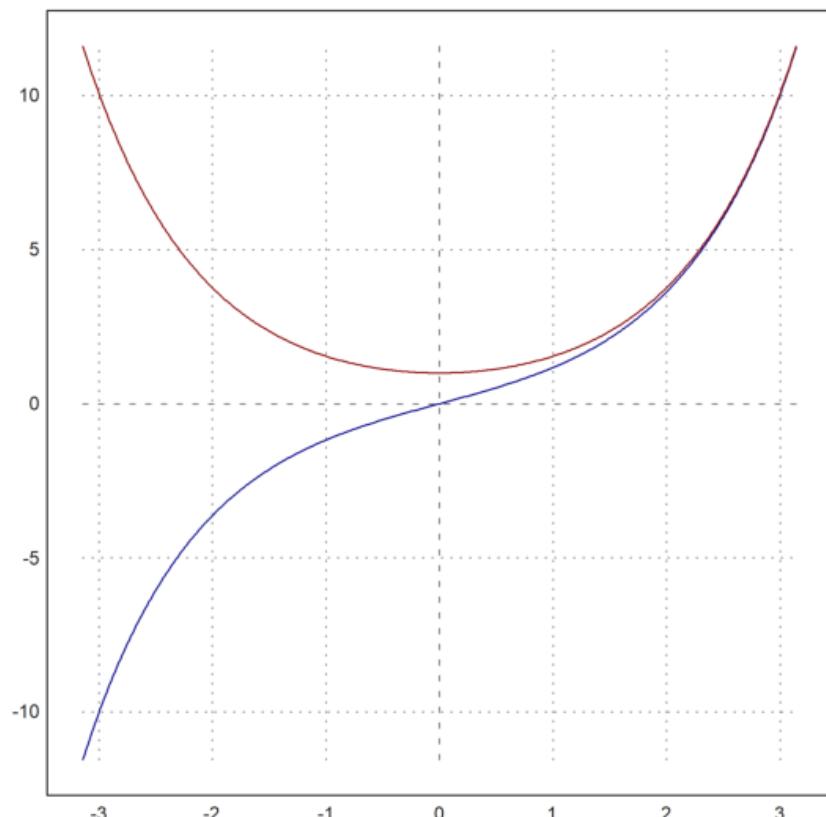
```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

$$\frac{e^{-x} (e^{2x} + 1)}{2}$$

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
```

$$\sin^2(3x^5 + 7)$$

```
>diff(f,3), diffc(f,3)
```

1198.32948904

1198.72863721

Apakah perbedaan diff dan diffc?

Jawab:

"diff" merupakan perintah untuk menghitung turunan biasa dari suatu fungsi terhadap variabel tertentu. "diff" menghasilkan turunan fungsi sebagai suatu expresi matematika.

"diffc" merupakan perintah yang digunakan untuk menghitung turunan suatu fungsi pada titik tertentu. "diffc" menghasilkan nilai numerik turunan pada titik yang ditentukan.

```
>$showev('diff(f(x),x))
```

$$\frac{d}{dx} \sin^2(3x^5 + 7) = 30x^4 \cos(3x^5 + 7) \sin(3x^5 + 7)$$

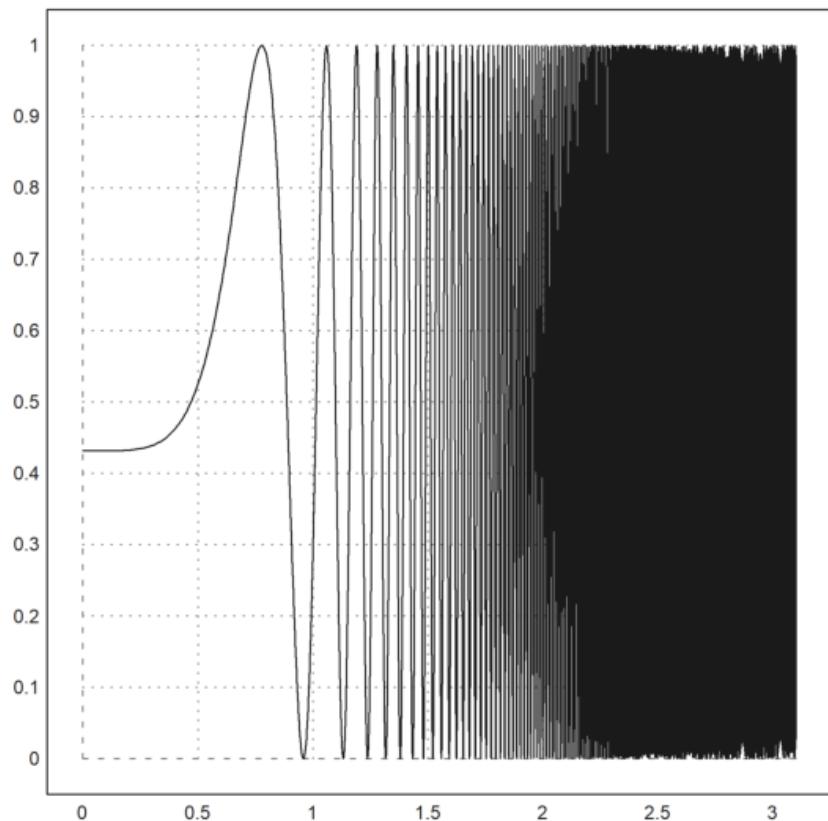
```
>$% with x=3
```

$$\%at\left(\frac{d}{dx} \sin^2(3x^5 + 7), x = 3\right) = 2430 \cos 736 \sin 736$$

```
>$float(%)
```

$$\%at\left(\frac{d^{1.0}}{dx^{1.0}} \sin^2(3.0x^5 + 7.0), x = 3.0\right) = 1198.728637211748$$

```
>plot2d(f,0,3.1):
```



```
>function f(x) &=5*cos(2*x)-2*x*sin(2*x) // mendefinisikan fungsi f
```

$$5 \cos(2x) - 2x \sin(2x)$$

```
>function df(x) &=diff(f(x),x) // fd(x) = f'(x)
```

$$-12 \sin(2x) - 4x \cos(2x)$$

```
>$'f(1)=f(1), $float(f(1)), $'f(2)=f(2), $float(f(2)) // nilai f(1) dan f(2)
```

$$-0.2410081230863468$$

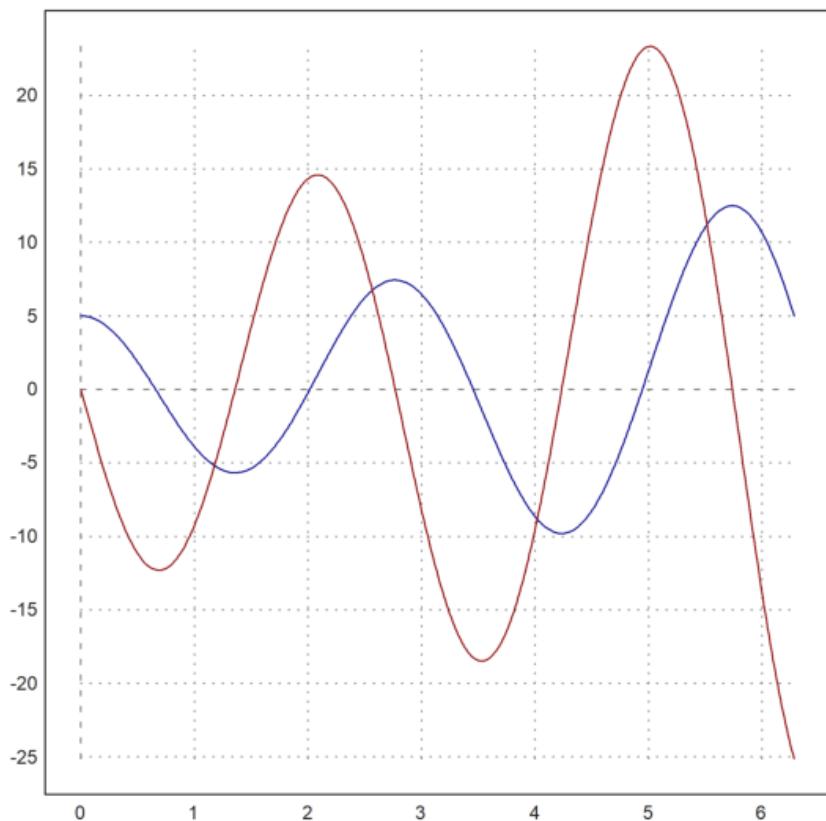
```
>xp=solve("df(x)",1,2,0) // solusi  $f'(x)=0$  pada interval [1, 2]
```

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 turunannya
```



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

$$f(x) = 5x - 4$$

Penyelesaian

```
>function f(x) &= 5*x-4;  
>$showev('limit((f(x+h)-f(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{5(x+h) - 5x}{h} = 5$$

```
>$showev('diff(f(x),x))
```

$$\frac{d}{dx} (5x - 4) = 5$$

Penyelesaian secara manual

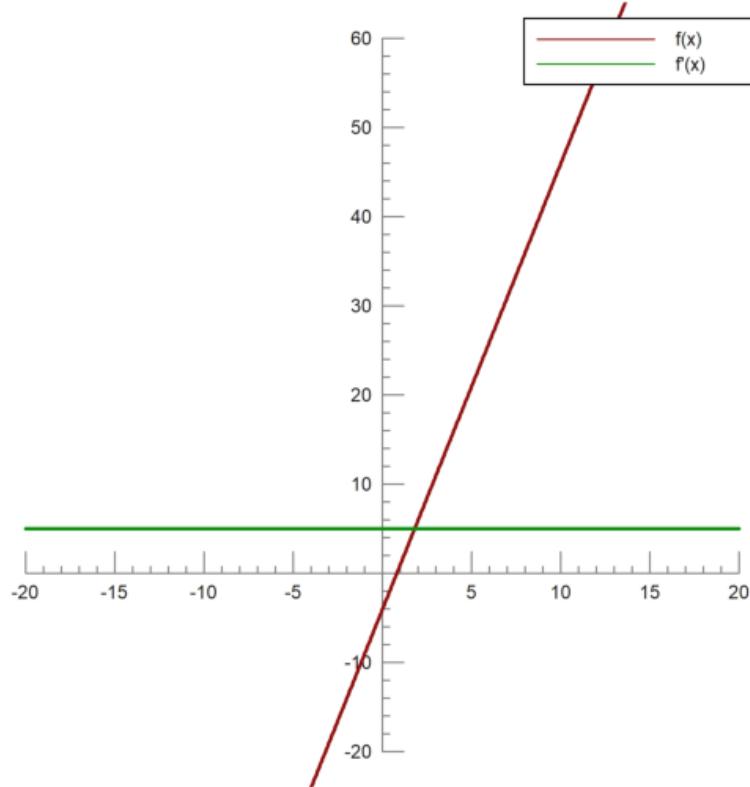
$$\lim_{h \rightarrow 0} \frac{5(x+h) - 4 - (5x - 4)}{h}$$

$$\lim_{h \rightarrow 0} \frac{5x + 5h - 4 - 5x + 4}{h}$$

$$\lim_{h \rightarrow 0} \frac{5h}{h}$$

$$5$$

```
>aspect(1); plot2d(["f",&diff(f(x),x)],-20,20,-20,60,color=2:3,thickness=2,grid=8); ...  
>labelbox(["f(x)", "f'(x)"], styles="-", ...  
> colors=2:3, w=0.3):
```



Soal 2

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

```
>function f(x) &= 2*x^3;
>$showev('limit((f(x+h)-f(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{2(x+h)^3 - 2x^3}{h} = 6x^2$$

```
>$showev('diff(f(x),x))
```

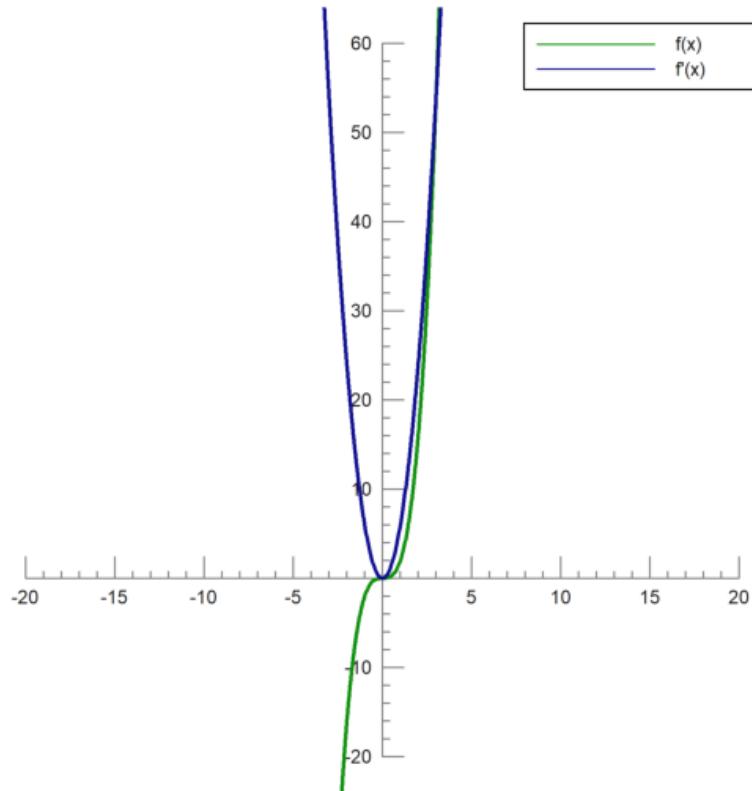
$$\frac{d}{dx} (2x^3) = 6x^2$$

Penyelesaian secara manual

$$\begin{aligned}
&\lim_{h \rightarrow 0} \frac{2(x+h)^3 - (2x^3)}{h} \\
&\lim_{h \rightarrow 0} \frac{2x^3 + 6hx^2 + 6h^2x + 2h^3 - 2x^3}{h} \\
&\lim_{h \rightarrow 0} 6x^2 + 6hx + 2h^2
\end{aligned}$$

$$6x^2$$

```
>aspect(1); plot2d(["f",&diff(f(x),x)],-20,20,-20,60,color=3:4,thickness=2,grid=8); ...
>labelbox(["f(x)","f'(x)"],styles="-", ...
>    colors=3:4,w=0.3):
```



Soal 3

$$f(x) = \frac{3}{5x}$$

```
>function f(x) &= 3/(5*x);
>$showev('limit((f(x+h)-f(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{\frac{3}{5(x+h)} - \frac{3}{5x}}{h} = -\frac{3}{5x^2}$$

```
>$showev('diff(f(x),x))
```

$$\frac{d}{dx} \left(\frac{3}{5x} \right) = -\frac{3}{5x^2}$$

Penyelesaian secara manual

$$\lim_{h \rightarrow 0} \frac{\frac{3}{5(x+h)} - \frac{3}{5x}}{h}$$

$$\lim_{h \rightarrow 0} \frac{\frac{3x-3(x+h)}{5x(x+h)}}{h}$$

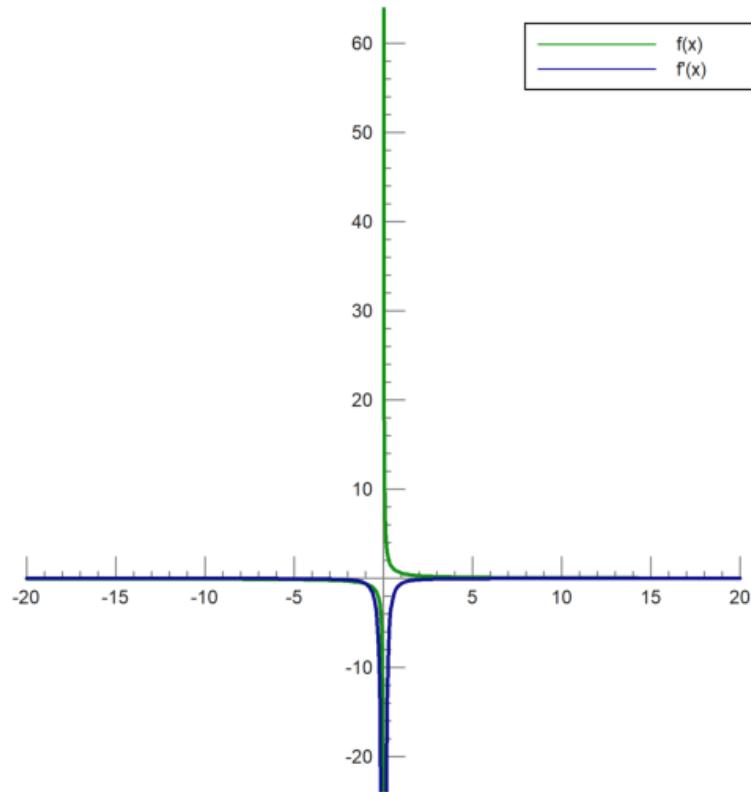
$$\lim_{h \rightarrow 0} \frac{\frac{3x-3x-3h}{5x^2+5xh}}{h}$$

$$\lim_{h \rightarrow 0} \frac{\frac{-3h}{5x^2+5xh}}{h}$$

$$\lim_{h \rightarrow 0} \frac{-3}{5x^2 + 5xh}$$

$$\frac{-3}{5x^2}$$

```
>aspect(1); plot2d(["f",&diff(f(x),x)],-20,20,-20,60,color=3:4,thickness=2,grid=8); ...
>labelbox(["f(x)","f'(x)"],style="-", ...
>    colors=3:4,w=0.3):
```



Soal 4

$$f(x) = \frac{2}{x+6}$$

```
>function f(x) &= 2/(x+6);
>$showev('limit((f(x+h)-f(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{\frac{2}{x+h+6} - \frac{2}{x+6}}{h} = -\frac{2}{x^2 + 12x + 36}$$

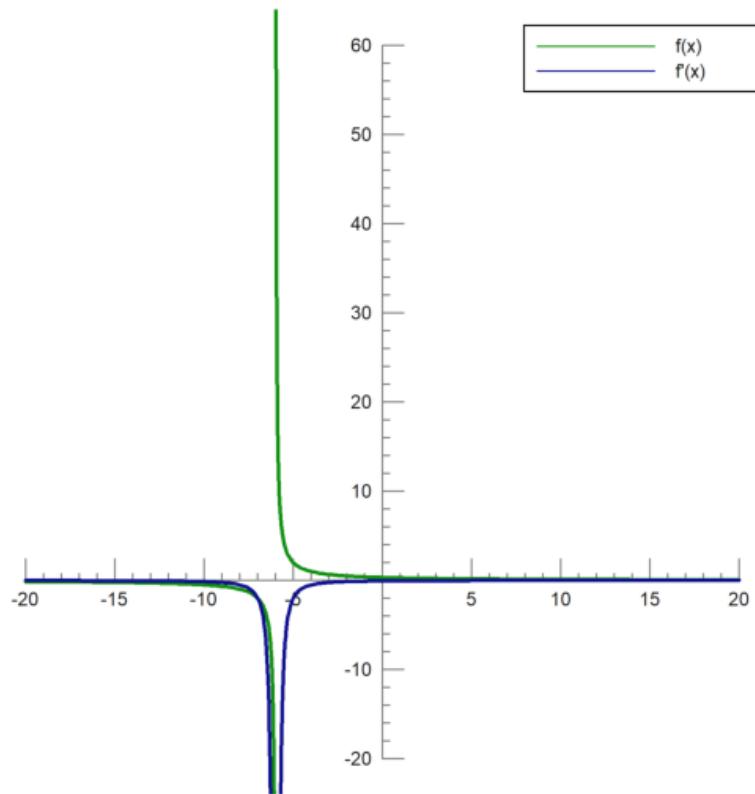
```
>$showev('diff(f(x),x))
```

$$\frac{d}{dx} \left(\frac{2}{x+6} \right) = -\frac{2}{(x+6)^2}$$

Penyelesaian secara manual

$$\begin{aligned} & \lim_{h \rightarrow 0} \frac{\frac{2}{x+h+6} - \frac{2}{x+6}}{h} \\ & \lim_{h \rightarrow 0} \frac{\frac{2(x+6) - 2(x+h+6)}{(x+h+6)(x+6)}}{h} \\ & \lim_{h \rightarrow 0} \frac{\frac{2x-12-2x-2h-12}{(x+h+6)(x+6)}}{h} \\ & \lim_{h \rightarrow 0} \frac{\frac{-2h}{(x+h+6)(x+6)}}{h} \\ & \lim_{h \rightarrow 0} \frac{-2}{(x+h+6)(x+6)} \\ & \lim_{h \rightarrow 0} \frac{-2}{x^2 + 6x + xh + 6h + 6x + 36} \\ & \frac{-2}{x^2 + 12x + 36} \end{aligned}$$

```
>aspect(1); plot2d(["f",&diff(f(x),x)],-20,20,-20,60,color=3:4,thickness=2,grid=8); ...
>labelbox(["f(x)","f'(x)"],styles="--", ...
> colors=3:4,w=0.3):
```



Soal 5

$$f(x) = \sqrt{3x}$$

```
>function f(x) &= sqrt(3*x);
>$showev('limit((f(x+h)-f(x))/h, h, 0))
```

$$\lim_{h \rightarrow 0} \frac{\sqrt{3}\sqrt{x+h} - \sqrt{3}\sqrt{x}}{h} = \frac{\sqrt{3}}{2\sqrt{x}}$$

```
>$showev('diff(f(x), x))
```

$$\frac{d}{dx} (\sqrt{3}\sqrt{x}) = \frac{\sqrt{3}}{2\sqrt{x}}$$

Penyelesaian secara manual

$$\lim_{h \rightarrow 0} \frac{\sqrt{3}\sqrt{x+h} - \sqrt{3}\sqrt{x}}{h}$$

$$\lim_{h \rightarrow 0} \frac{\sqrt{3x+3h} - \sqrt{3x}}{h}$$

$$\lim_{h \rightarrow 0} \frac{\sqrt{3x+3h} - \sqrt{3x}}{h} * \frac{\sqrt{3x+3h} + \sqrt{3x}}{\sqrt{3x+3h} + \sqrt{3x}}$$

$$\lim_{h \rightarrow 0} \frac{3x + 3h - 3x}{h(\sqrt{3x+3h} + \sqrt{3x})}$$

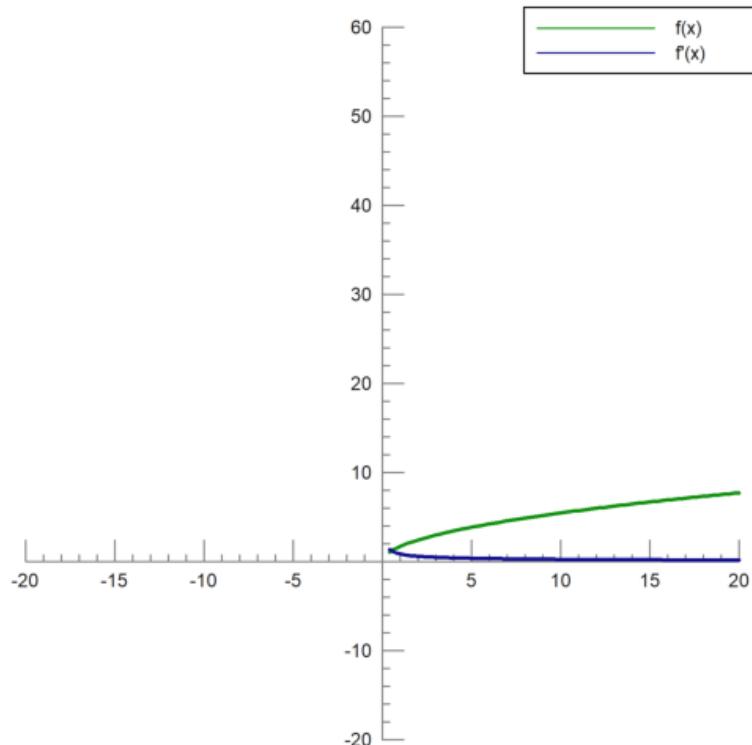
$$\lim_{h \rightarrow 0} \frac{3h}{h(\sqrt{3x+3h} + \sqrt{3x})}$$

$$\lim_{h \rightarrow 0} \frac{3}{\sqrt{3x+3h} + \sqrt{3x}}$$

$$\lim_{h \rightarrow 0} \frac{3}{\sqrt{3x} + \sqrt{3x}}$$

$$\lim_{h \rightarrow 0} \frac{3}{2\sqrt{3x}}$$

```
>aspect(1); plot2d(["f",&diff(f(x),x)],-20,20,-20,60,color=3:4,thickness=2,grid=8); ...
>labelbox(["f(x)","f'(x)"],styles="-", ...
> colors=3:4,w=0.3):
```



```
>reset();
```

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))
```

$$\int x^n dx = \frac{x^{n+1}}{n+1}$$

```
>$showev('integrate(1/(1+x),x))
```

$$\int \frac{1}{x+1} dx = \log(x+1)$$

```
>$showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2 + 1} dx = \arctan x$$

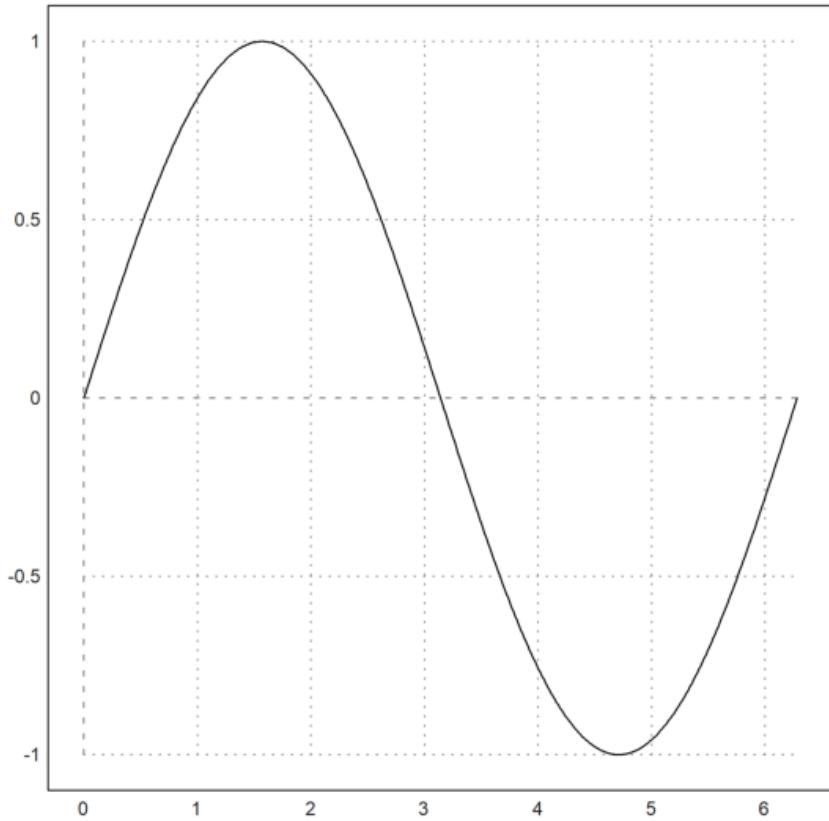
```
>$showev('integrate(1/sqrt(1-x^2),x))
```

$$\int \frac{1}{\sqrt{1-x^2}} dx = \arcsin x$$

```
>$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))
```

$$\int_a^b x^n \, dx = \frac{b^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
```

$$\int x^2 \sqrt{2x+1} \, dx = \frac{(2x+1)^{\frac{7}{2}}}{28} - \frac{(2x+1)^{\frac{5}{2}}}{10} + \frac{(2x+1)^{\frac{3}{2}}}{12}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
```

$$\int_0^2 x^2 \sqrt{2x+1} \, dx = \frac{25^{\frac{5}{2}}}{21} - \frac{2}{105}$$

```
>${ratsimp(%)
```

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{25^{\frac{7}{2}} - 2}{105}$$

```
>${showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^\pi - 1) \sin a + (e^\pi + 1) \cos a$$

```
>${factor(%)
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^\pi - 1) (\sin a - \cos a)$$

```
>function map f(x) &= E^(-x^2)
```

$$\frac{e^{-x^2}}{2}$$

```
>${showev('integrate(f(x),x))
```

$$\int e^{-x^2} dx = \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

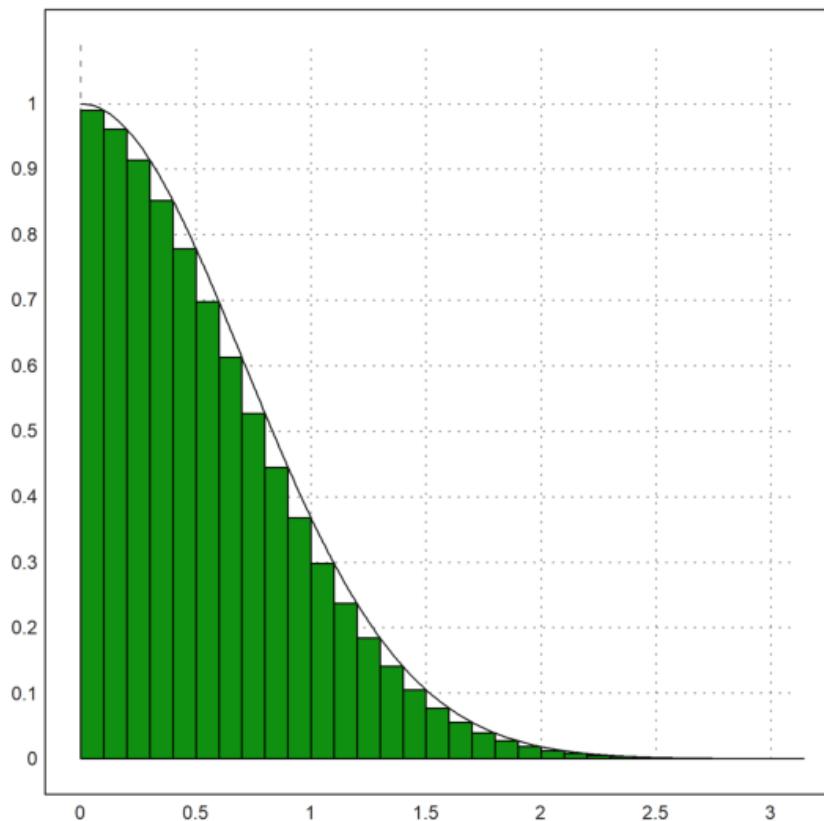
$$\operatorname{erf}(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}} dx.$$

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

$$\int_0^{\pi} e^{-x^2} dx$$

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-nilai x
>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 $x = 0.1, 0.2, 0.3, \dots, 3.2$.

```
>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!

$$\int_0^{\infty} e^{-x^2} dx$$

Untuk membuktikan bahwa hasilnya seperti di atas, kita dapat menggunakan rumus integral Gaussian yang sangat terkenal. Rumus ini adalah:

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

maka

$$\begin{aligned}\int_0^{\infty} e^{-x^2} dx &= \frac{1}{2} \int_{-\infty}^{\infty} e^{-x^2} dx \\ \int_0^{\infty} e^{-x^2} dx &= \frac{1}{2} \sqrt{\pi} \\ \int_0^{\infty} e^{-x^2} dx &= \frac{\sqrt{\pi}}{2}\end{aligned}$$

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

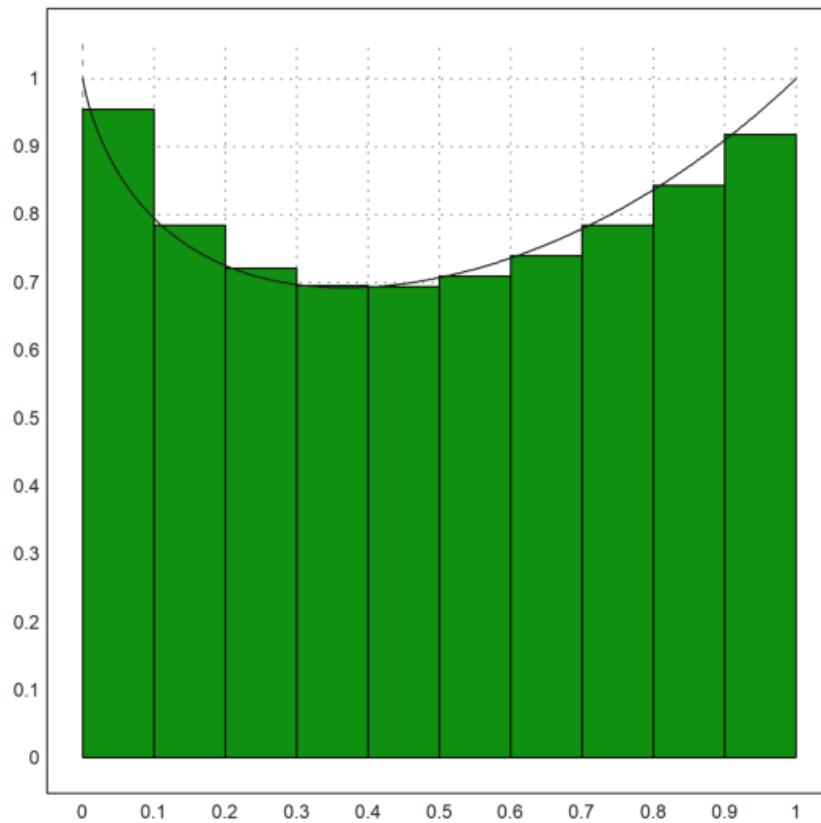
```
>function f(x) &= x^x
```

$$\begin{matrix} x \\ x \end{matrix}$$

```
>$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));
```

$$\int_0^1 x^x \, dx = 0.7834935879025506$$

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
>function f(x) &= sin(3*x^5+7)^2
```

$$\sin^2(3x^5 + 7)$$

```
>integrate(f,0,1)
```

```
0.542581176074
```

```
>&showev('integrate(f(x),x,0,1))
```

```
1
/
[      2      5           1           pi
gamma(-) sin(14) sin(--)
      5           10
I   sin (3 x  + 7) dx = -----
]                               1/5
/
0
4/5           1           4/5           1
- (((6 gamma_incomplete(-, 6 I) + 6 gamma_incomplete(-, - 6 I))
      5           5           5
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(%)
```

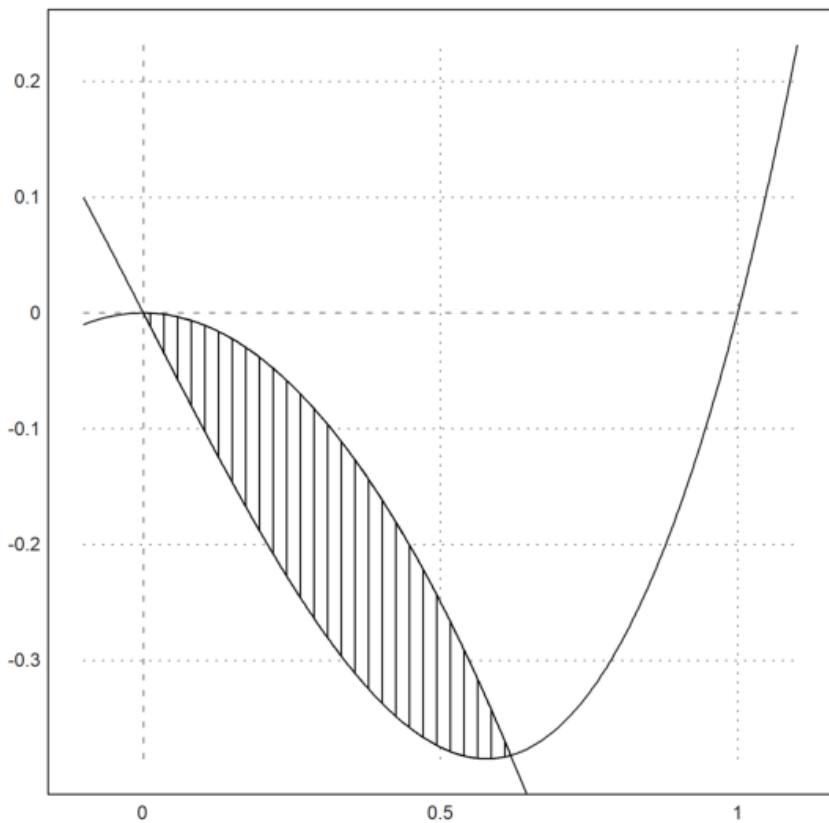
```
1.0
/
[      2      5
I   sin (3.0 x  + 7.0) dx =
]
/
0.0
0.09820784258795788 - 0.00833333333333333
(0.3090169943749474 (0.1367372182078336
(4.192962712629476 I gamma_incomplete(0.2, 6.0 I)
- 4.192962712629476 I gamma_incomplete(0.2, - 6.0 I))
+ 0.9906073556948704 (4.192962712629476 gamma_incomplete(0.2, 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)
```

$$\int_0^1 x e^{-x} dx = 1 - 2e^{-1}$$

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 daerah antara 2 kurva
```



```
>a=solve("x^3-x+x^2",0), b=solve("x^3-x+x^2",1) // absis titik-titik potong kedua kurva
```

```
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 kurva secara eksak
```

$$\left[x = \frac{-\sqrt{5} - 1}{2}, x = \frac{\sqrt{5} - 1}{2}, x = 0 \right]$$

```
>$showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara eksak
```

$$\int_0^{\frac{\sqrt{5}-1}{2}} -x^3 - x^2 + x \, dx = \frac{13 - 5^{\frac{3}{2}}}{24}$$

```
> $float(%)
```

$$\int_{0.0}^{0.6180339887498949} -1.0x^3 - 1.0x^2 + x \, dx = 0.07581917135421037$$

Panjang Kurva

Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

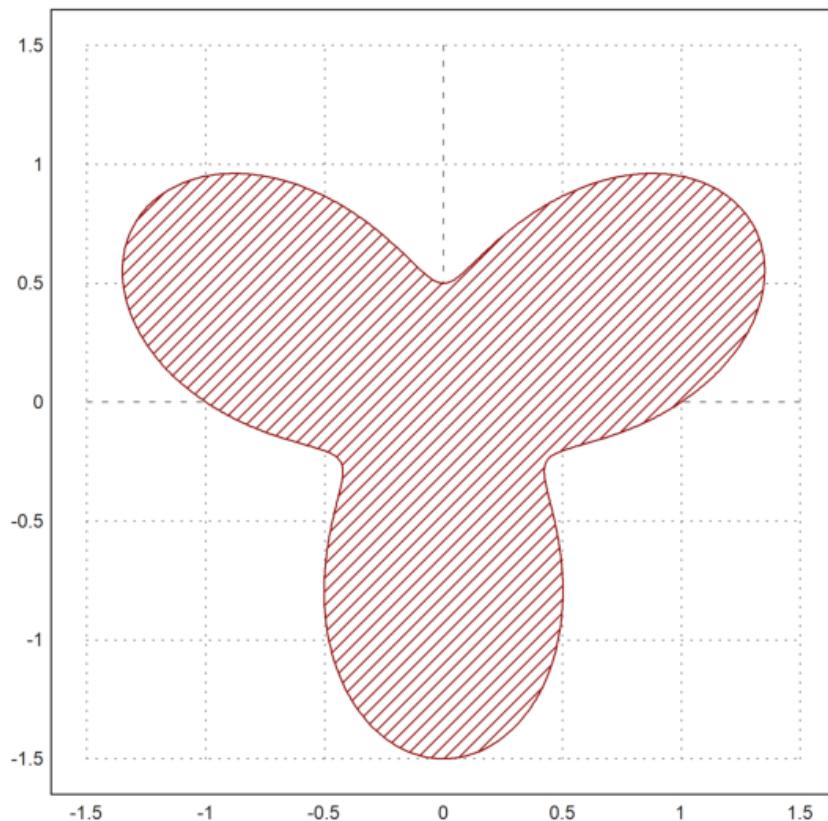
$$\gamma(t) = (r(t) \cos(t), r(t) \sin(t))$$

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}, \quad 0 \leq t \leq 2\pi.$$

```
>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 terlebih dahulu
```



```
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
```

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

```
>function fx(t) &= r(t)*cos(t); $' fx(t)=fx(t)
```

$$fx(t) = \cos t \left(\frac{\sin(3t)}{2} + 1 \right)$$

```
>function fy(t) &= r(t)*sin(t); $' fy(t)=fy(t)
```

$$fy(t) = \sin t \left(\frac{\sin(3t)}{2} + 1 \right)$$

```
>function ds(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $' ds(t)=ds(t)
```

$$ds(t) = \frac{\sqrt{4 \cos(6t) + 4 \sin(3t) + 9}}{2}$$

```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
```

$$\frac{\int_0^{2\pi} \sqrt{4 \cos(6x) + 4 \sin(3x) + 9} dx}{2}$$

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara umerik dengan perintah EMT.

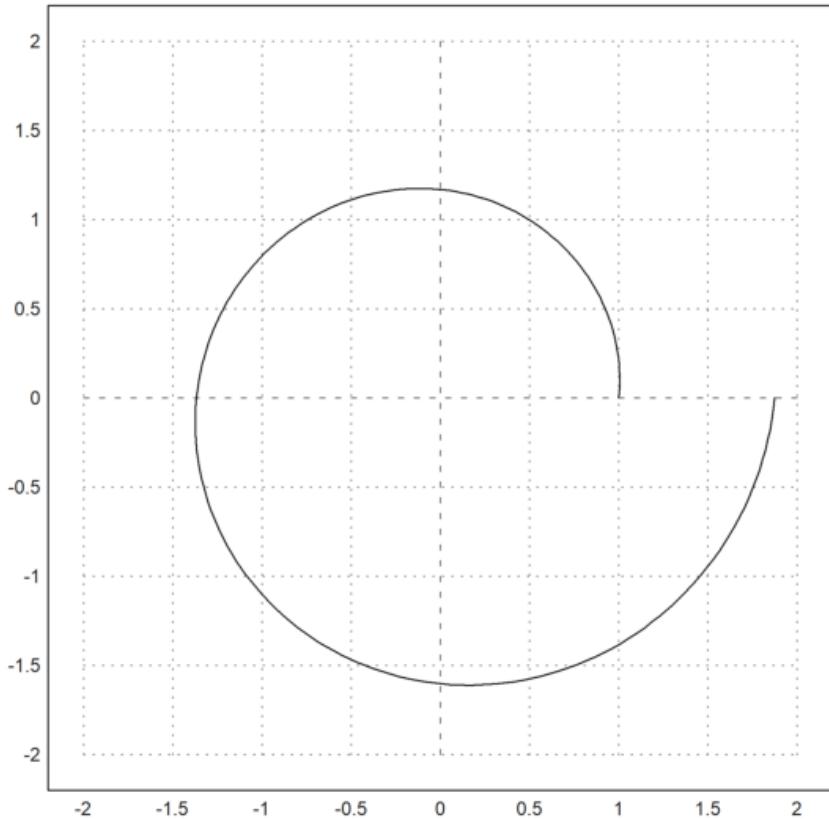
```
>integrate("ds(x)",0,2*pi)
```

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)", x=2, xmin=0, xmax=2*pi):
```



```
>&kill(a) // hapus expresi a
```

done

```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
```

$$fx(t) = e^{at} \cos t$$

```
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
```

$$fy(t) = e^{at} \sin t$$

```
>function df(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t)
```

$$df(t) = \sqrt{a^2 + 1} e^{at}$$

```
>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

Soal:

Tunjukkan bahwa keliling lingkaran dengan jari-jari r adalah $K=2\pi r$.

Penyelesaian:

Akan ditunjukkan bahwa keliling lingkaran dengan jari-jari r adalah $K=2\pi r$
Persamaan lingkaran dengan pusat $(0,0)$ dengan jari-jari R

$$x^2 + y^2 = R^2$$

engan menggunakan turunan implisit dari persamaan lingkaran di atas, diperoleh

$$2xdx + 2ydy = 0$$

Jika hanya jika

$$\frac{dy}{dx} = -\frac{x}{y}$$

apat dilihat rumus keliling lingkaran dengan pusat $(0,0)$ menggunakan integral,

$$\begin{aligned} \frac{1}{4}K &= \int_0^R \sqrt{(dx)^2 + (dy)^2} dx \\ &= \int_0^R \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \end{aligned}$$

substitusikan hasilnya

$$\frac{1}{4}K = \int_0^R \frac{1}{\sqrt{1 - (\frac{x}{R})^2}} dx$$

dengan memisalkan,

$$\frac{x}{R} = \sin(u)$$

,maka

$$dx = R\cos(u)du$$

$$x = 0$$

$$u = 0$$

dan

$$x = R$$

$$u = \frac{\pi}{2}$$

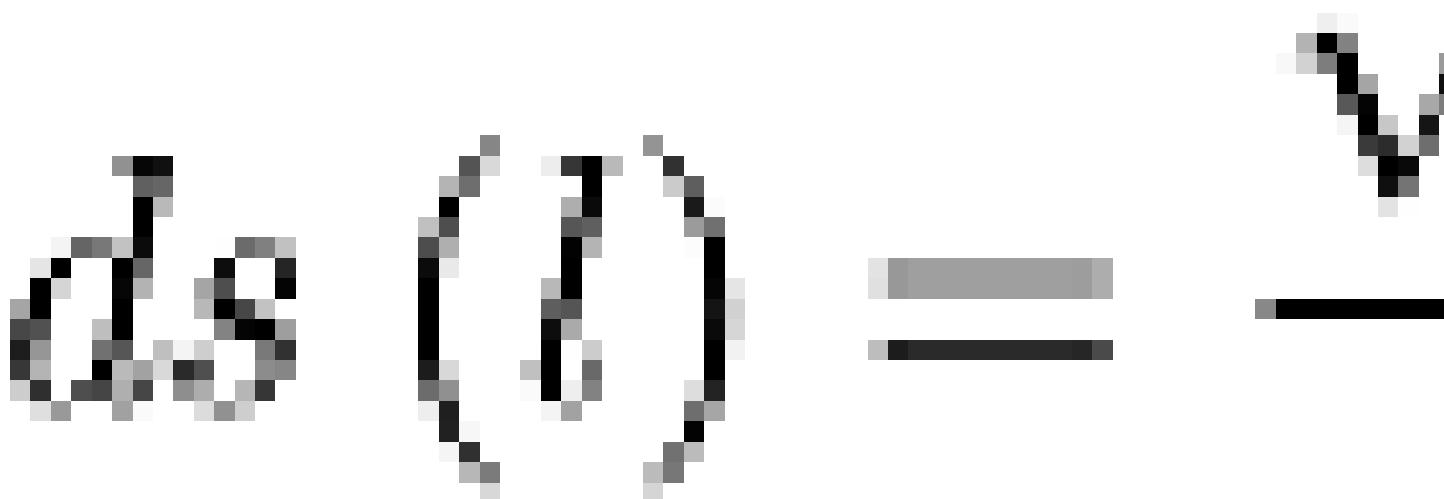
oleh karena itu,

$$\begin{aligned}\frac{1}{4}K &= \int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{1-\sin^2 u}} R \cos(u) \, du \\ &= \frac{1}{2}\pi R\end{aligned}$$

sehingga rumus keliling lingkarannya adalah

$$K = 2\pi R$$

```
>plot2d("x^2", xmin=-1, xmax=1):
```



```
>showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))
```

$$\int_{-1}^1 \sqrt{4x^2 + 1} dx = \frac{\operatorname{asinh} 2 + 2\sqrt{5}}{2}$$

```
>float(%)
```

$$\int_{-1.0}^{1.0} \sqrt{4.0x^2 + 1.0} dx = 2.957885715089195$$

```
>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

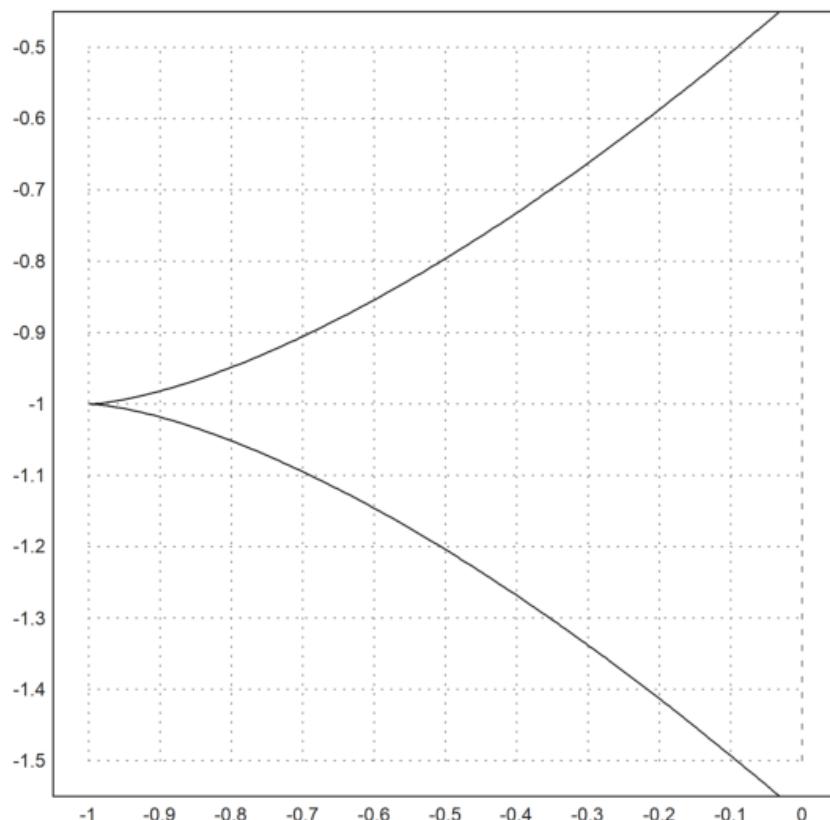
Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

$$x^3 + y^3 - 3xy = 0.$$

```
>z &= x^3+y^3-3*x*y; $z
```

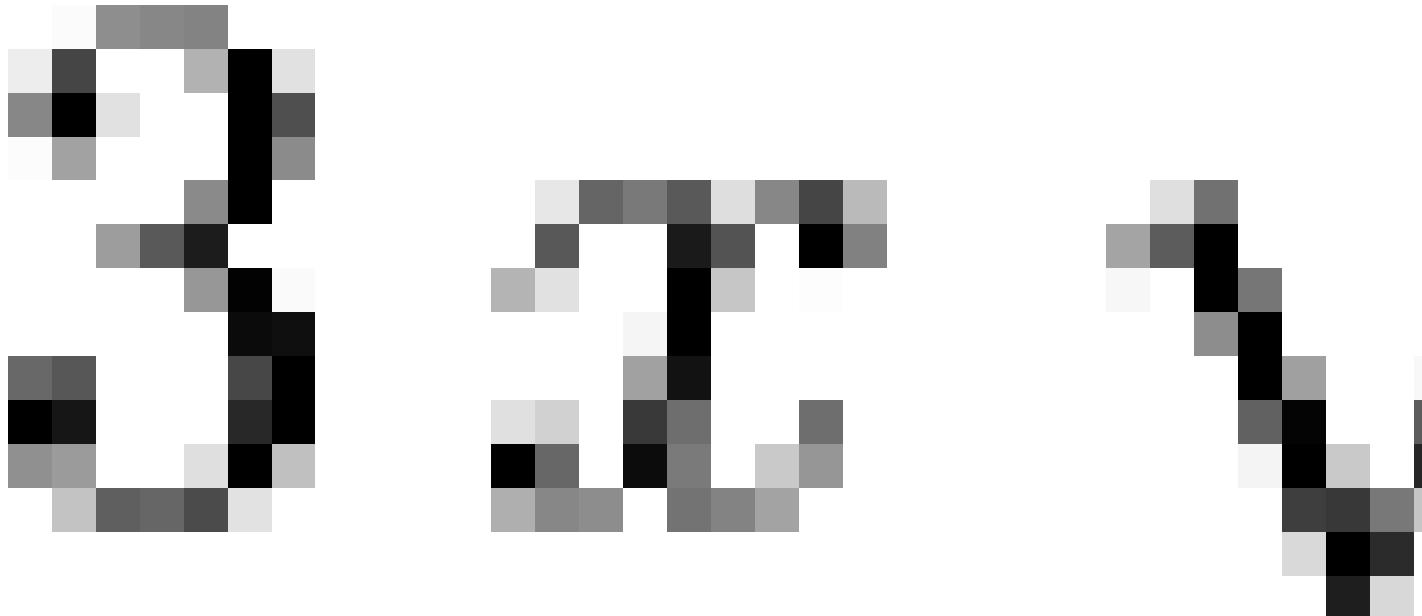
$$y^3 - 3xy + x^3$$

```
>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
```

$$\left[x = \frac{3l}{l^3 + 1}, x = 0 \right]$$

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0x^2 + 4.0} dx = 2.337835372767141$$

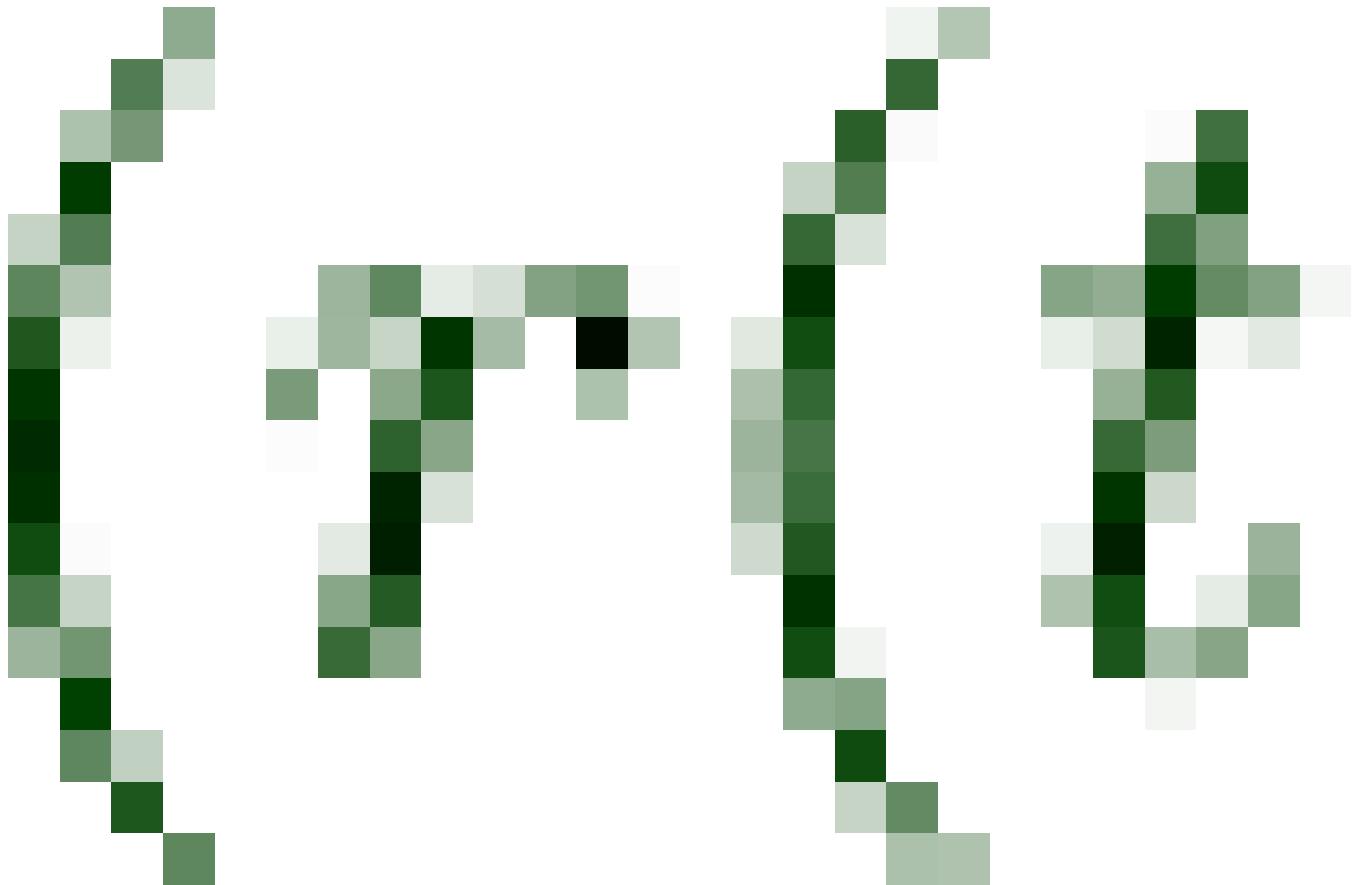
Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); $' f(l)=f(l)
```

$$f(l) = \frac{3l}{l^3 + 1}$$

Fungsi tersebut juga dapat digunakan 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(l)
```

$$ds(l) = \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}}$$

```
>$integrate(ds(l), l, 0, 1)
```

$$\int_0^1 \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}} dl$$

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral tersebut 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 hampiran integral yang
```

4.91748872168

Perhitungan di atas 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 parabola sebelumnya
```

2.95788571509

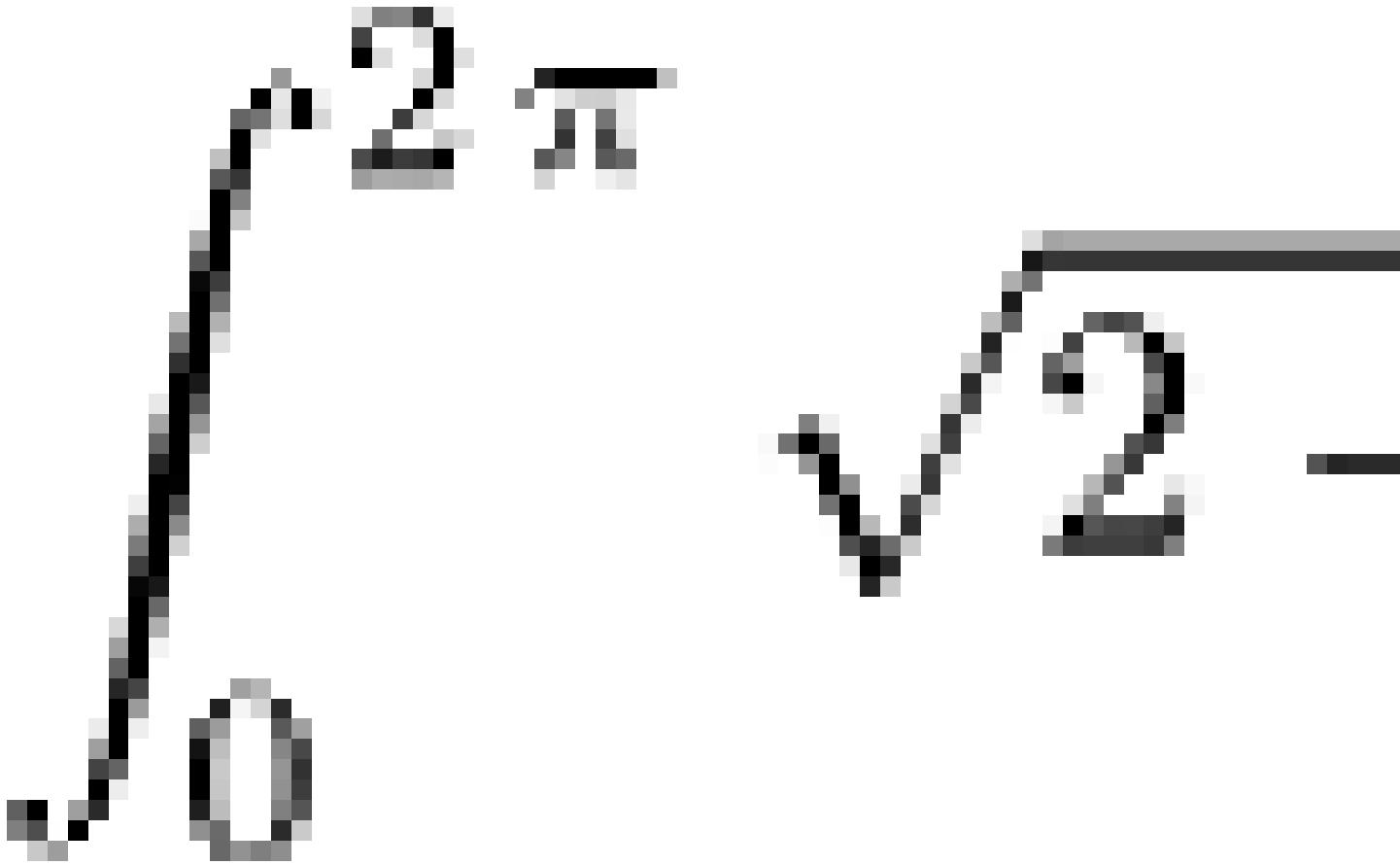
Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)", mxm("x*f(x)", 0, 1)) // cek contoh terakhir, bandingkan hasilnya
```

4.91748872168

Kita hitung panjang spiral Archimedes 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)
```

$$\sqrt{\left(\frac{d}{dx}(\cos x - x \sin x)\right)^2 + \left(\frac{d}{dx}(\sin x + x \cos x)\right)^2}$$

```
>sol &= ds(x*cos (x),x*sin(x)); $sol // Kita gunakan untuk menghitung panjang kurva terakhir
```

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

```
>$sol | trigreduce | expand, $integrate(% ,x,0,2*pi), %()
```

$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$

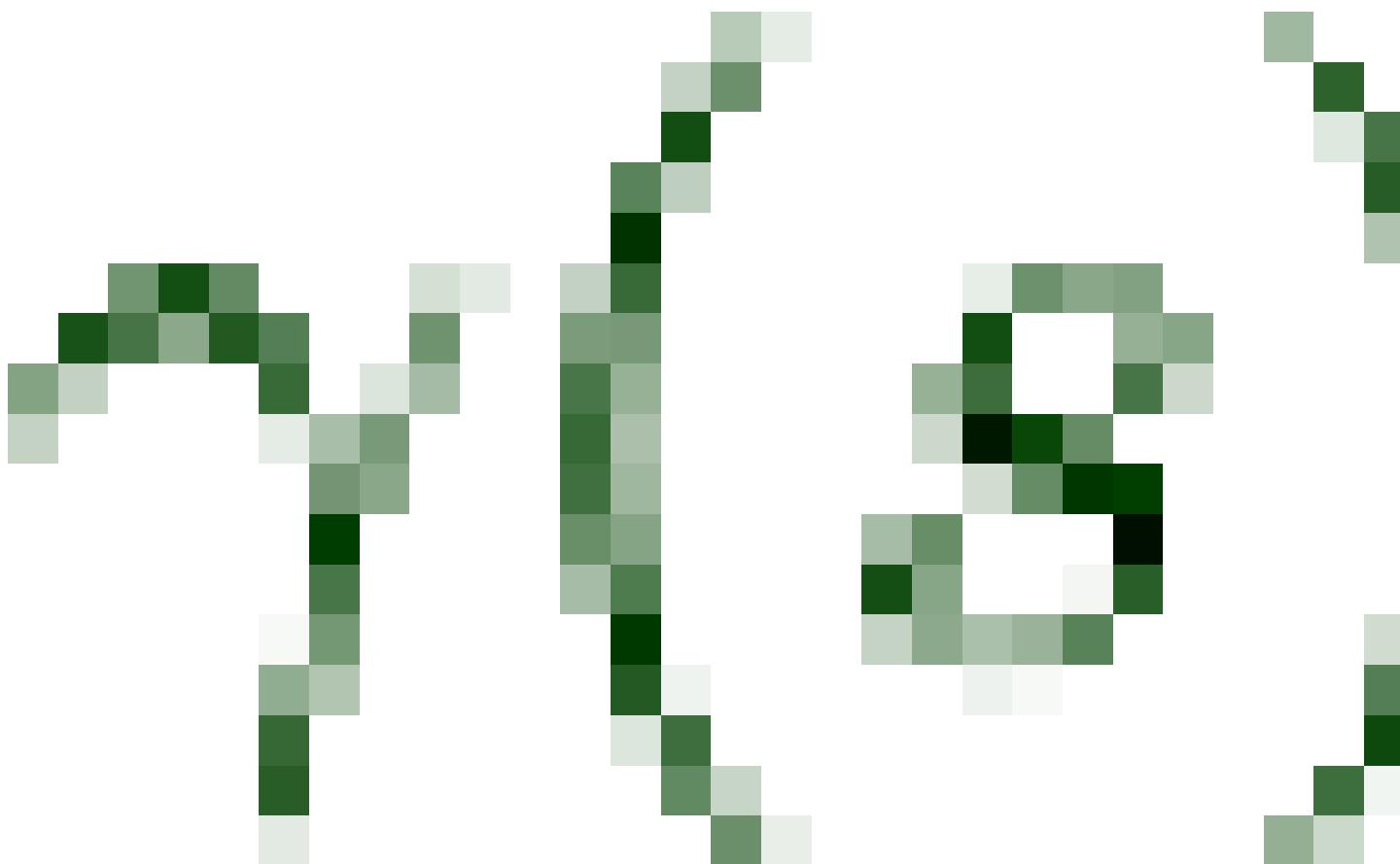
$$\kappa = \frac{1}{R}$$

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
```

$$3x\sqrt{9x^2 + 4}$$

```
>$showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%)// panjang kurva di atas
```

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0x^2 + 4.0} dx = 2.337835372767141$$

$\mathbf{T}(s) = \gamma'(s),$
 $\mathbf{T}'(s) = 1$ (konstanta) $\Rightarrow \mathbf{T}'(s) \cdot \mathbf{T}(s) = 0$
 $s(s) = \|\mathbf{T}'(s)\| = \|\gamma''(s)\| = \sqrt{(x'(s))^2 + (y'(s))^2}.$

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))
```

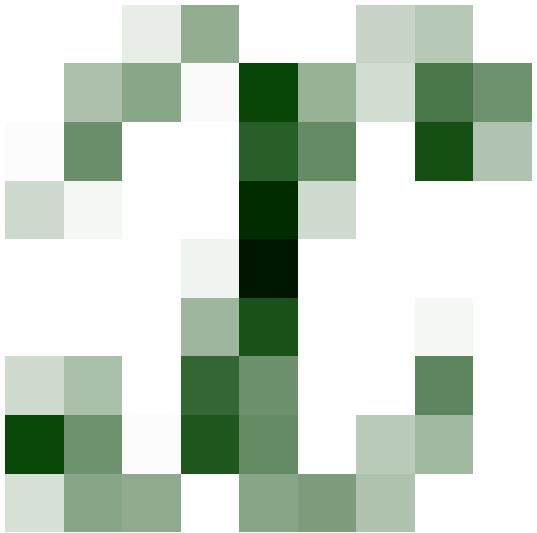
$$r(t - \sin(t))$$

```
>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):
```



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 panjang kurva s
```

$$\sqrt{\sin^2 x + \cos^2 x - 2 \cos x + 1} = \sqrt{2 - 2 \cos x}$$

```
>ds &= trigsimp(ds); $ds
```

$$\sqrt{2 - 2 \cos x}$$

```
>$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu putaran penuh
```

$$\int_0^{2\pi} \sqrt{2 - 2 \cos x} dx = 8$$

```
>integrate(mxm("ds"),0,2*pi) // hitung secara numerik
```

```
>romberg(mxm("ds"), 0, 2*pi) // cara lain hitung secara numerik
```

Perhatikan, seperti terlihat pada gambar, panjang sikloid lebih besar daripada keliling lingkarannya, yakni:

$$2\pi.$$

Kurvatur (Kelengkungan) Kurva



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 2π sejauh $2\pi R$.)

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 $\mathbf{T}'(s)$ 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

$$x = r \cos t, \quad y = r \sin t.$$

```
>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 // elemen panjang
```

$r \ t$

```
>&kill(s); fx &= r*cos(s/r); fy &= r*sin(s/r); // definisi ulang persamaan parametrik terhadap s
>k &= trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur lingkaran dengan s
```

$$\frac{1}{r}$$

Untuk representasi parametrik umum, misalkan

$$x = x(t), \quad y = y(t)$$

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

$$\begin{aligned}\kappa &= \frac{d\phi}{ds} = \frac{\frac{d\phi}{dt}}{\frac{ds}{dt}} \quad (\phi \text{ adalah sudut kemiringan garis singgung dan } s \text{ adalah panjang kurva}) \\ &= -\frac{\frac{d\phi}{dt}}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}} = \frac{\frac{d\phi}{dt}}{\sqrt{x'(t)^2 + y'(t)^2}}.\end{aligned}$$

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

$$x = x(t), \quad y = y(t)$$

adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}.$$

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kutub

$$x = r(\theta) \cos \theta, \quad y = r(\theta) \sin \theta,$$

maka rumus kurvaturnya adalah

$$\kappa(\theta) = \frac{r(\theta)^2 + 2r'(\theta)^2 - r(\theta)r''(\theta)}{(r'(\theta)^2 + r''(\theta)^2)^{3/2}}.$$

(Silakan Anda turunkan rumus tersebut!)

Contoh:

Lingkaran dengan pusat (0,0) dan jari-jari r dapat dinyatakan dengan persamaan parametrik

$$x = r \cos t, \quad y = r \sin t.$$

Nilai kelengkungan lingkaran tersebut adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}} = \frac{r^2}{r^3} = \frac{1}{r}.$$

Hasil cocok dengan definisi kurvatur suatu kelengkungan.

Kurva

$$y = f(x)$$

dapat dinyatakan ke dalam persamaan parametrik

$$x = t, \quad y = f(t), \quad \text{dengan } x'(t) = 1, \quad x''(t) = 0,$$

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) // kelengkungan p
```

$$k(x) = \frac{2a}{((2ax + b)^2 + 1)^{\frac{3}{2}}}$$

```
>function f(x) &= x^2+x+1; $y=f(x) // akan kita plot kelengkungan parabola untuk a=b=c=1
```

$$y = x^2 + x + 1$$

```
>function k(x) &= (diff(f(x),x,2)) / (1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) // kelengkungan p
```

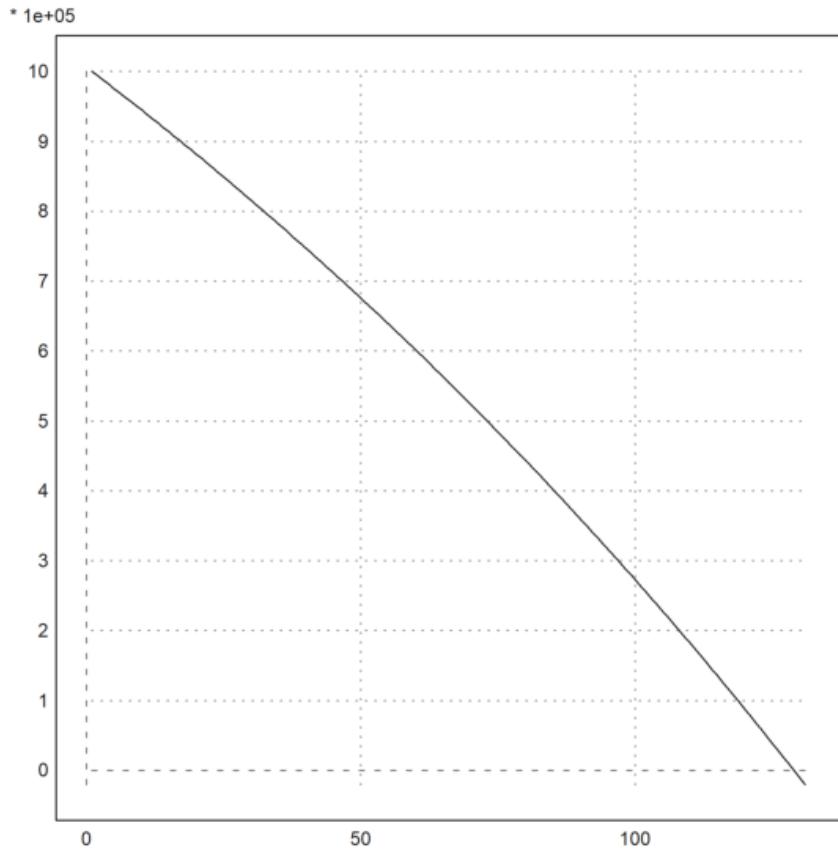
$$k(x) = \frac{2}{((2x + 1)^2 + 1)^{\frac{3}{2}}}$$

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

$$(-1/2, 3/4), \quad 1/k(2) = 1/2,$$

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, color=green, >add);
>plot2d("-1/2+1/k(-1/2)*cos(x)", "5/4+1/k(-1/2)*sin(x)", xmin=0, xmax=2pi, >add, color=blue);
```



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

$$\kappa = \frac{F_y^2 F_{xx} - 2F_x F_y F_{xy} + F_x^2 F_{yy}}{(F_x^2 + F_y^2)^{3/2}}.$$

(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 &=diff(diff(F(x,y),
```

$$2 \ a \ x + b$$

$$2 \ a$$

$$- 1$$

$$0$$

$$0$$

```
>function k(x) &= (Fy^2*Fxx-2*Fx*Fy*Fxy+Fx^2*Fyy)/(Fx^2+Fy^2)^(3/2); $'k(x)=k(x) // kurvat
```

$$k(x) = \frac{2a}{\left((2ax+b)^2+1\right)^{\frac{3}{2}}}$$

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 $t=t_0$ dan $x=b$ bersesuaian dengan $t=t_1$, 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 implisit $F(x,y)=0$

- 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 oskulasi-nya.

Barisan dan Deret

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan 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 berurutan (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

$$x_{k+1} = g(x_k), \quad x_0 = x_0, \quad k = 1, 2, 3, \dots, n.$$

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]
```

$$\left[A = \frac{5a}{4r} \right]$$

$$[A = 2500.0]$$

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, $x[1], \dots, x[n-1]$ 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)
```

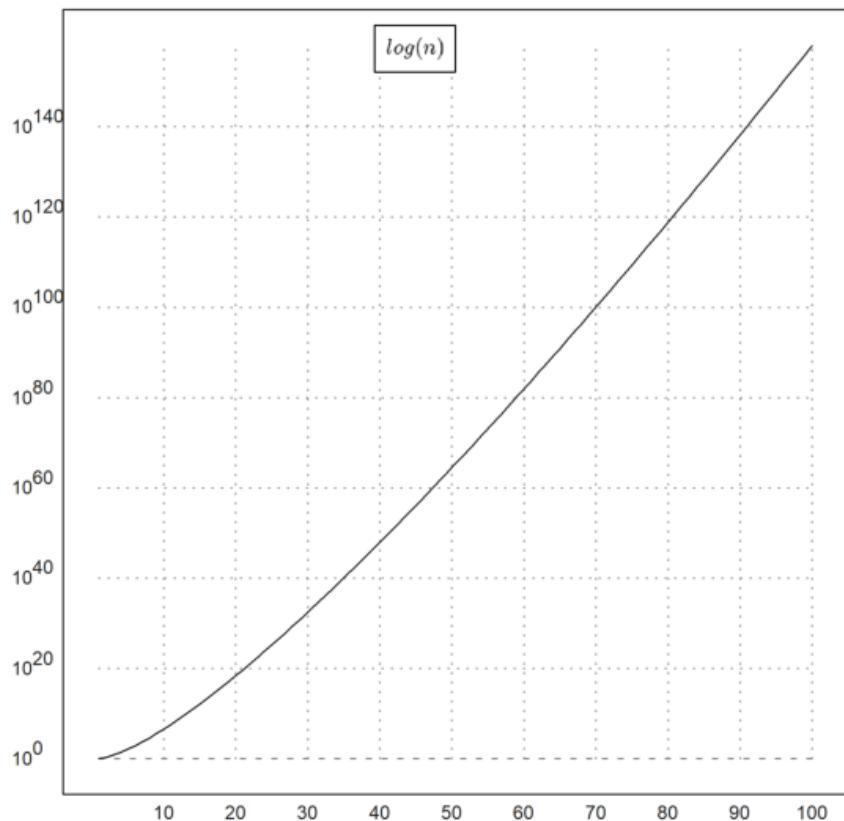
```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]
```

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

$$\begin{array}{cccccc} 1 & & 2 & & 5 & & 13 \\ & 1 & & 3 & & 8 & & 21 & \dots \end{array}$$

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "matrixpower()". Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak $\log_2(n)$.

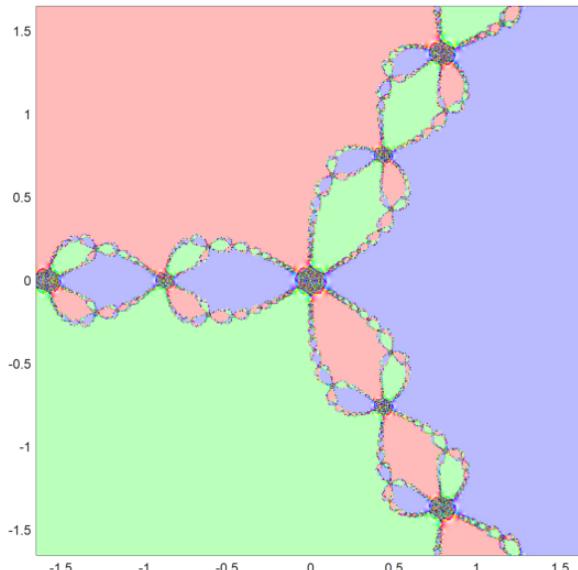
$$x_n = A \cdot x_{n-1} = A^2 \cdot x_{n-2} = A^3 \cdot x_{n-3} = \dots = A^{n-1} \cdot x_1.$$

```
>sequence("matrixpower(A, n). [1;1]", 1, 6)
```

Real 2 x 6 matrix

$$\begin{array}{cccccc} 1 & & 5 & & 13 & & 34 \\ & 1 & & 8 & & 21 & & 55 & \dots \end{array}$$

Spiral Theodorus



Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

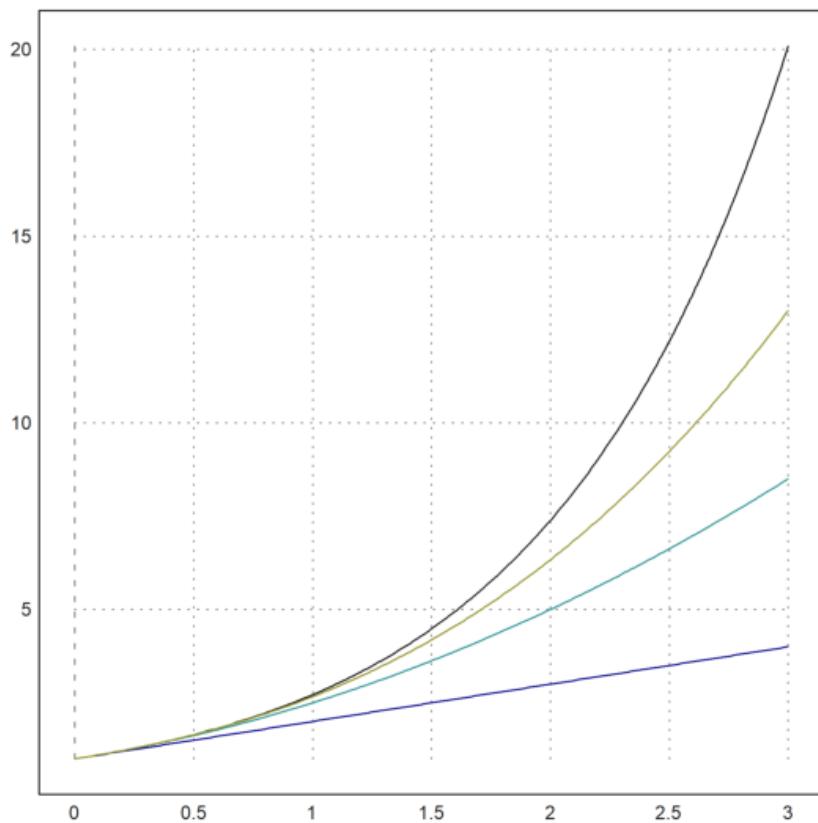
$$x_n = \left(1 + \frac{i}{\sqrt{n-1}}\right) x_{n-1}, \quad x_1 = 1,$$

yang menghasilkan barisan bilangan kompleks.

```
>function g(n) := 1+I/sqrt(n)
```

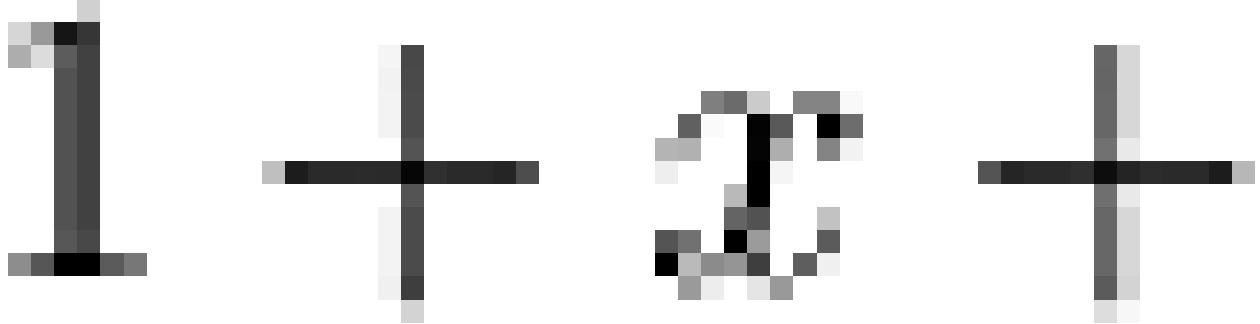
Rekursinya dapat dijalankan sebanyak 17 untuk menghasilkan barisan 17 bilangan kompleks, kemudian di-gambar bilangan-bilangan kompleksnya.

```
>x=sequence("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox(latex("Spiral\ Theodorus"),0.4
```



Selanjutnya dihubungkan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end:
```



>

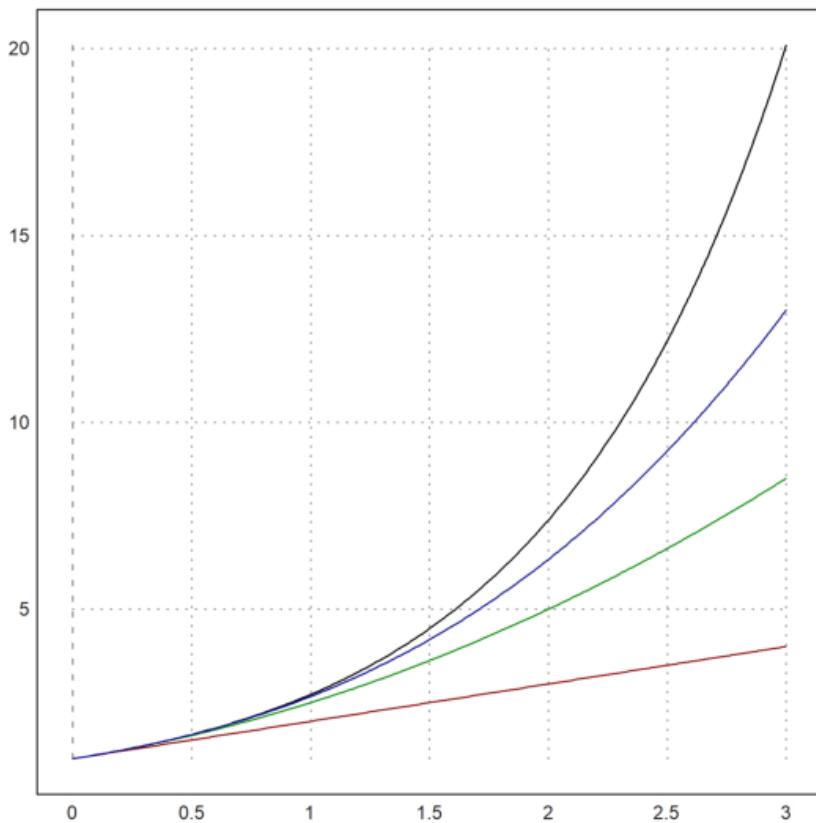
Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memerlukan indeks dan bilangan kompleks. Dalam hal ini digunakan vektor kolom pada bidang.

```
>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 interval awal (1, 2)
```

~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 meningkatkan 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 dipercepat dengan 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

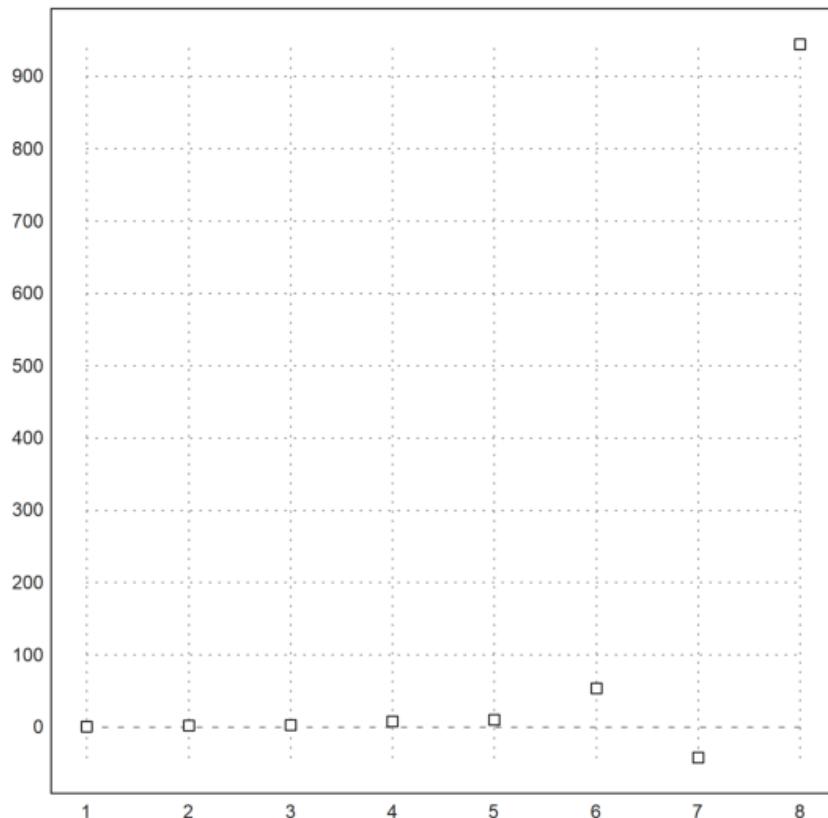
Penggabungan matriks menggunakan tanda " $|$ " dapat digunakan untuk menyimpan semua hasil iterasi.

```
>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
```

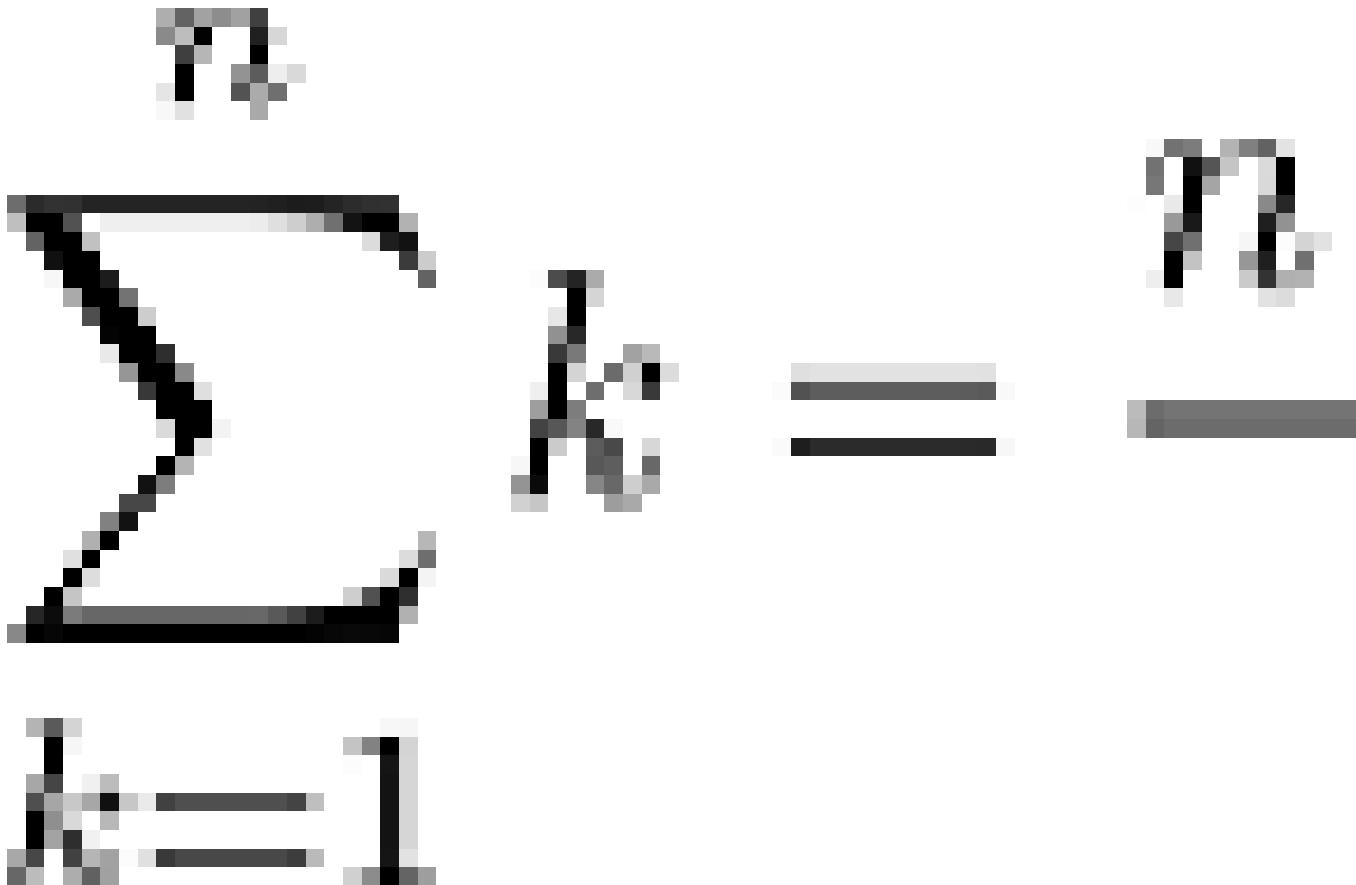
Misalnya, berikut adalah iterasi untuk mendapatkan hampiran akar kuadrat 2, cukup cepat, konvergen pada iterasi ke-5, jika dimulai dari hampiran awal 2.

```
>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 penambahannya 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
```

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

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.

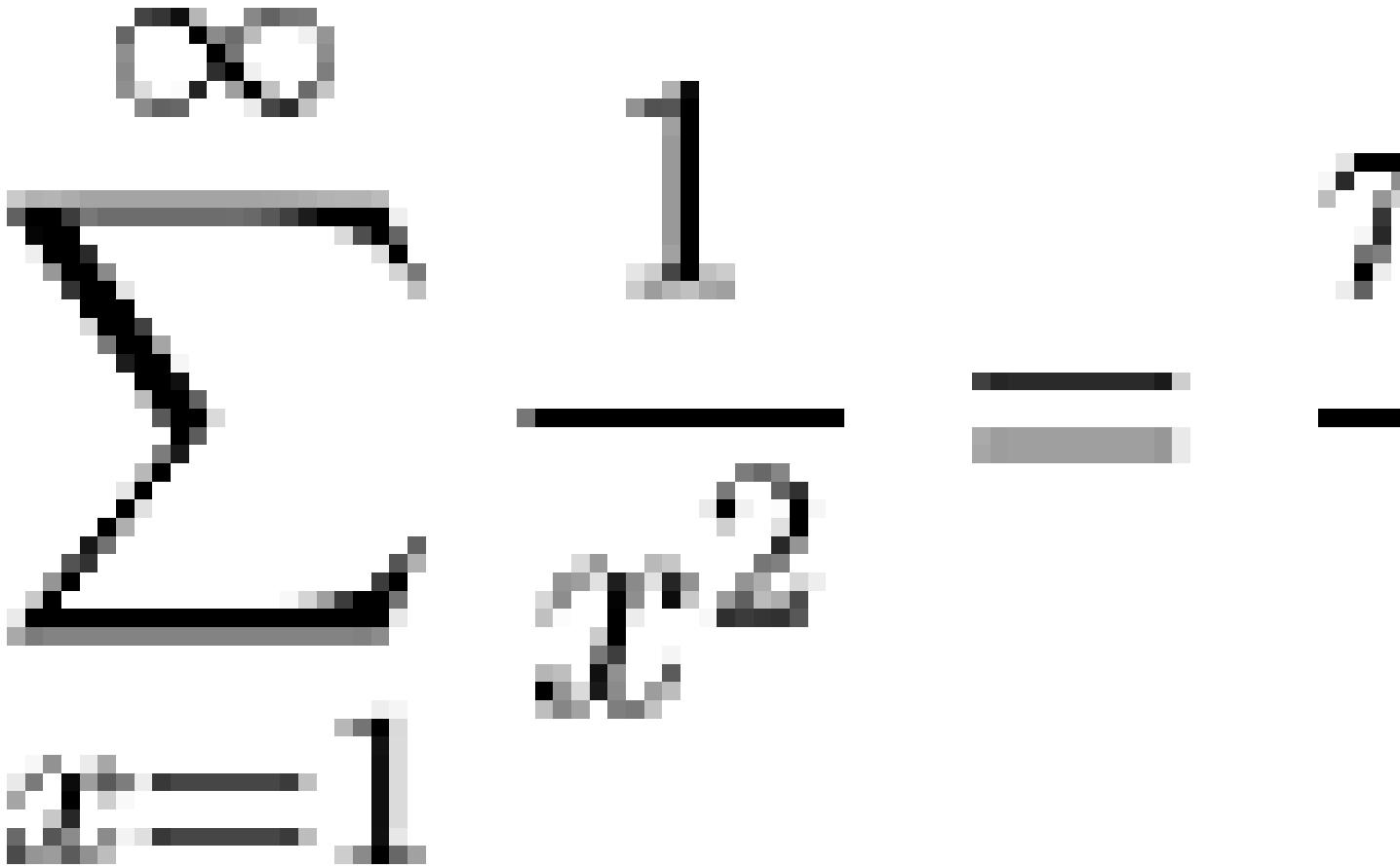
```
>z=&solve(p)()
```

```
[ -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 berpangkat terkecil
```

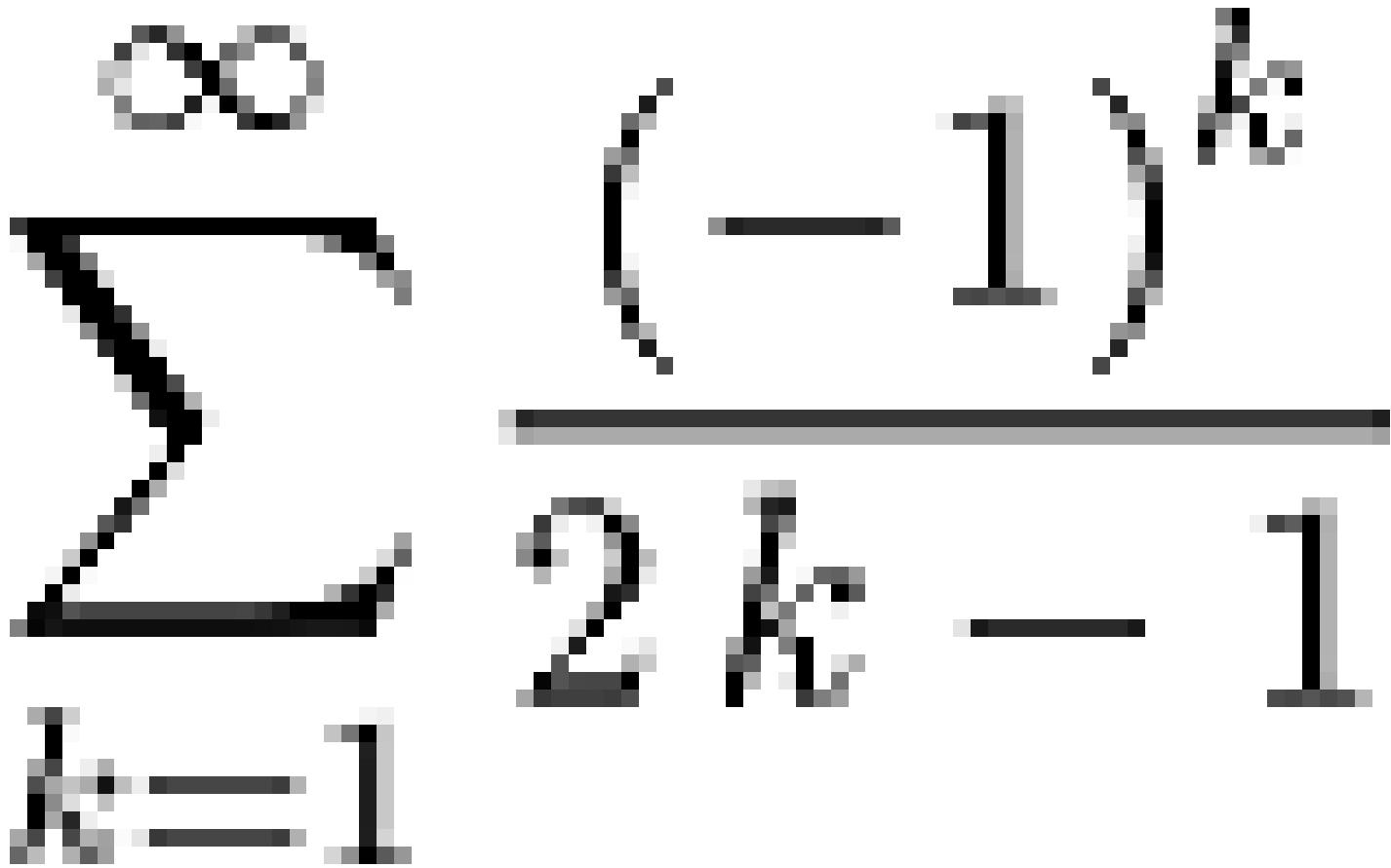
```
true
```

```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Taylor untuk e^x
```

$$\left[1 + x, 1 + x + \frac{x^2}{2}, 1 + x + \frac{x^2}{2} + \frac{x^3}{6} \right]$$

Untuk mengubah barisan deret tersebut menjadi vektor string di EMT digunakan fungsi mxm2str(). Selanjutnya, vektor string/ekspresi hasilnya dapat digambar seperti menggambar vektor ekspresi pada EMT.

```
>plot2d("exp(x)",0,3); // plot fungsi aslinya, e^x  
>plot2d(mxm2str("deret"),>add,color=4:6): // plot ketiga deret taylor hampiran fungsi ters
```

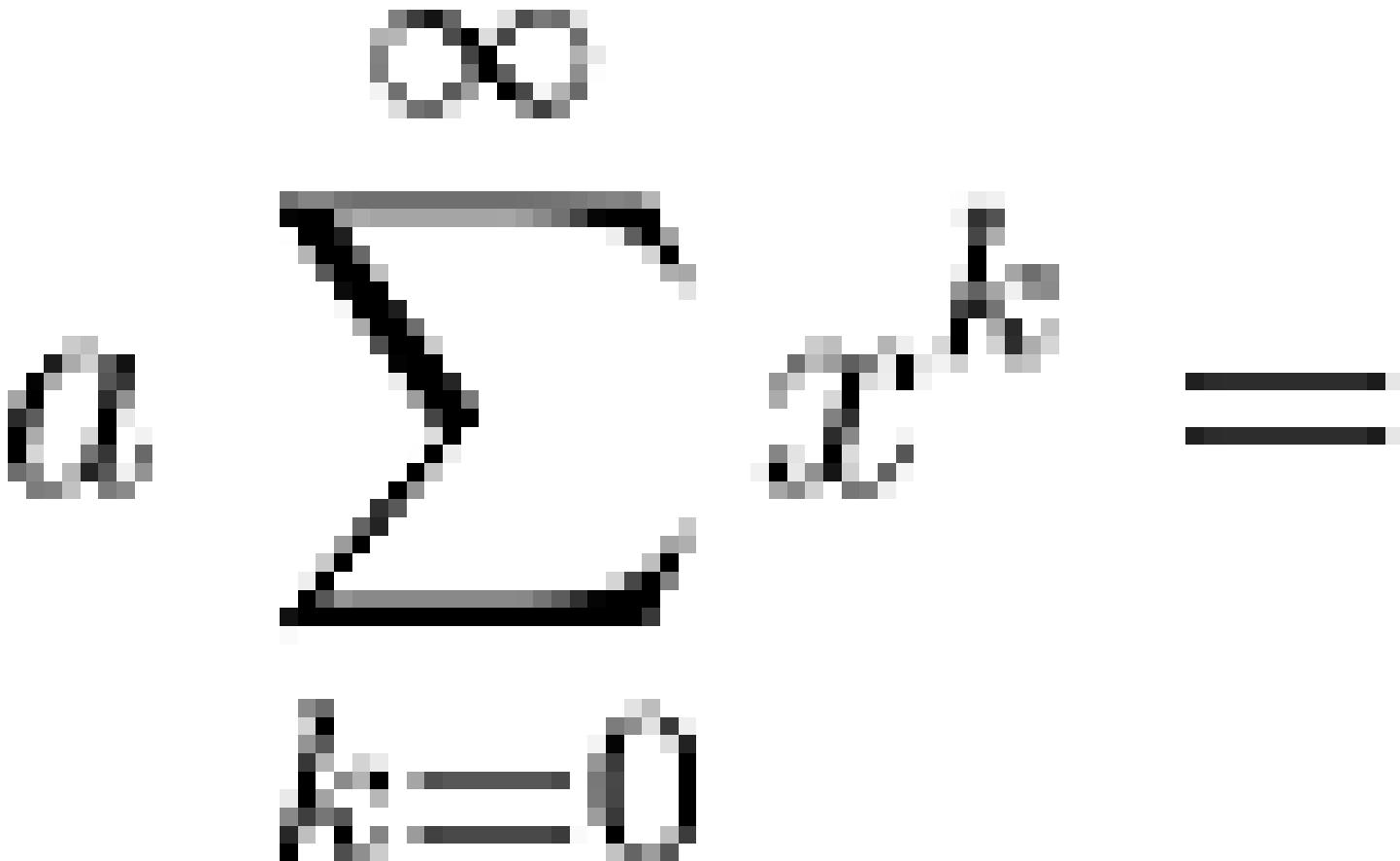


Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

```
>$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):
```



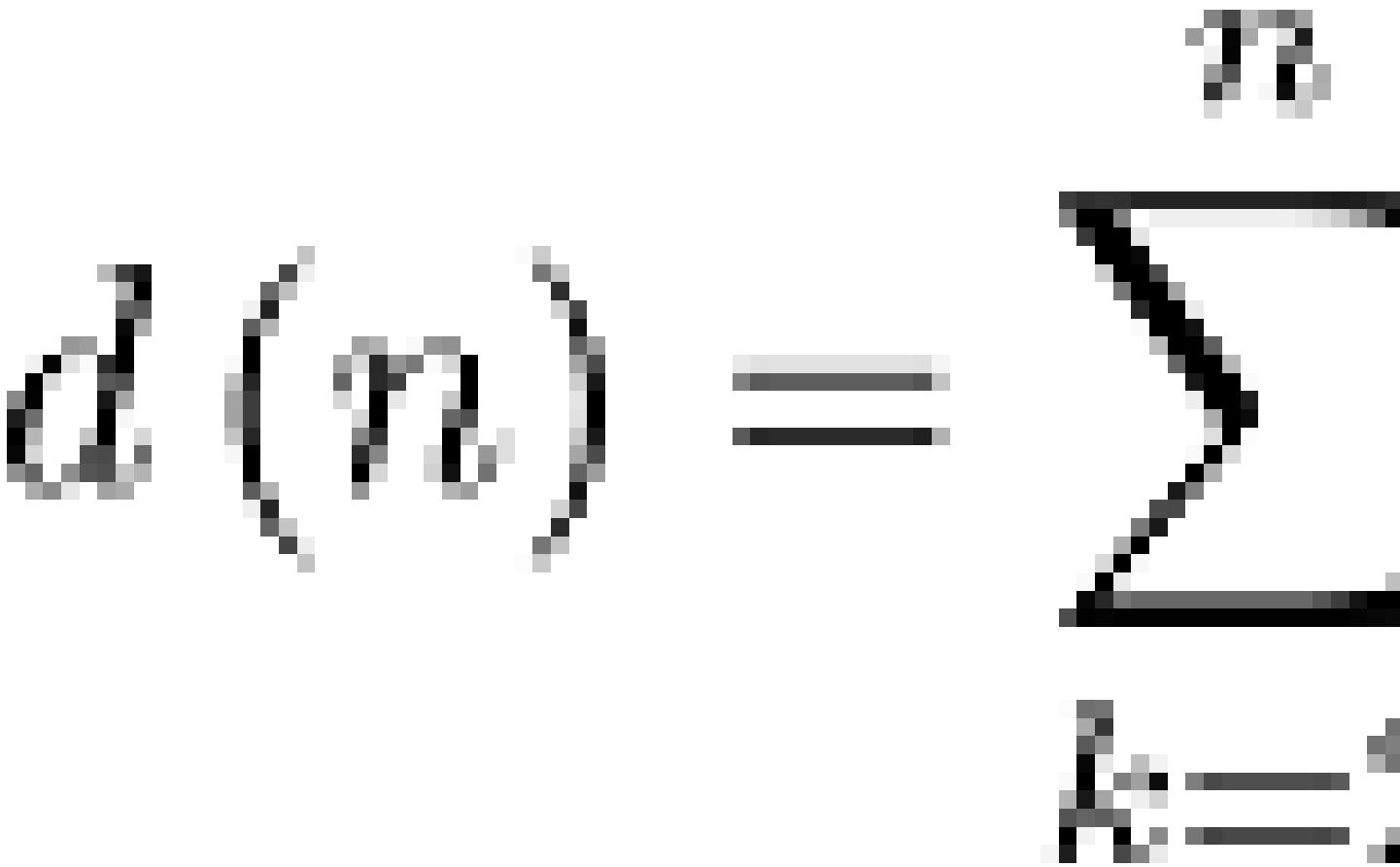
```
>sum(sin(k*x)/k,k,1,5)
```

$$\sin x + \frac{\sin(2x)}{2} + \frac{\sin(3x)}{3} + \frac{\sin(4x)}{4} + \frac{\sin(5x)}{5}$$

Berikut adalah cara menggambar kurva

$$y = \sin(x) + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \dots$$

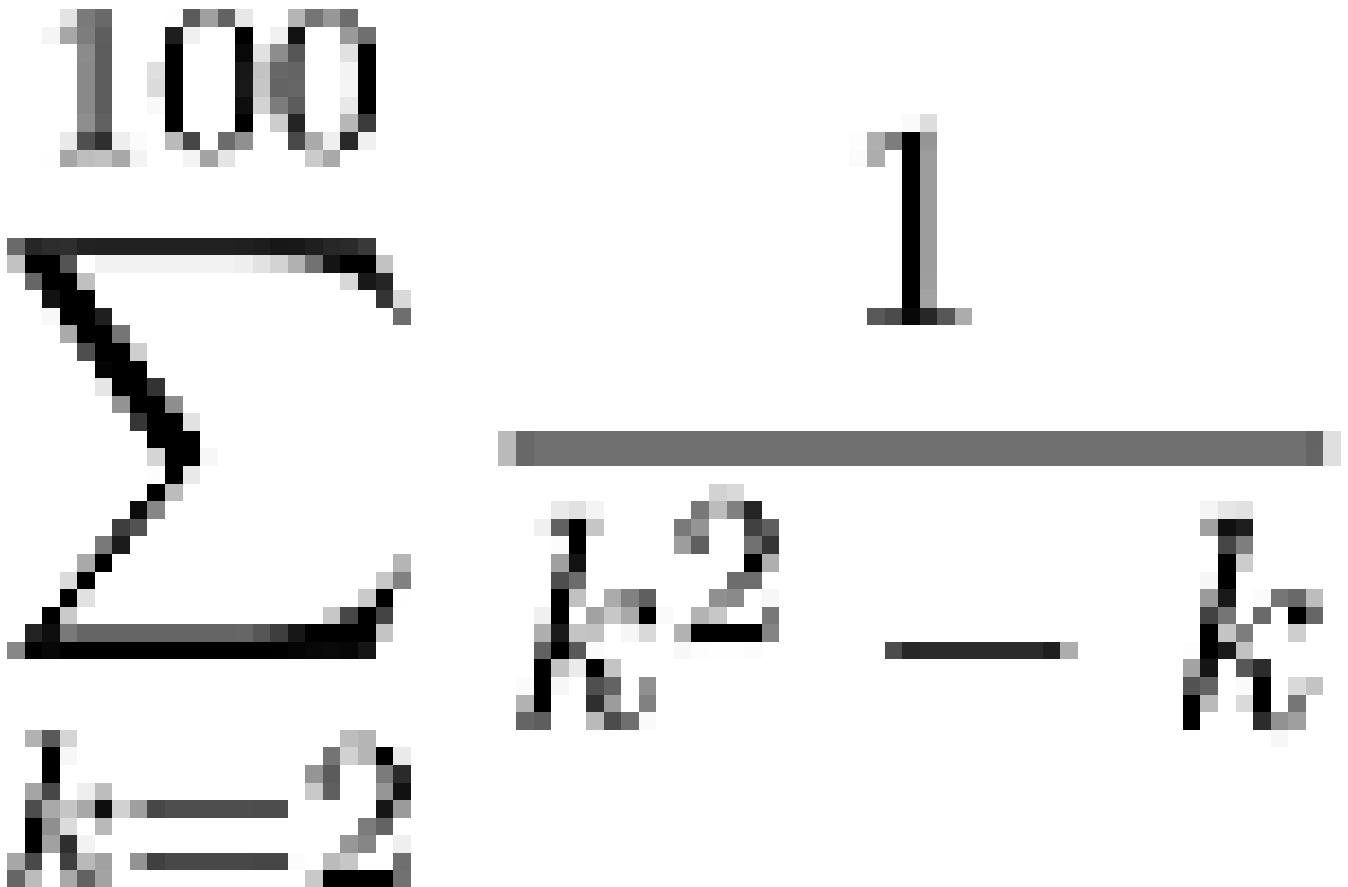
```
>plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi):
```



Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan kita akan menggambar kurva

$$y = \sum_{k=1}^{100} \frac{\sin(kx)}{k}, \quad 0 \leq x \leq 2\pi.$$

```
>x=linspace(0,2pi,1000); k=1:100; y=sum(sin(k*x')/k)'; plot2d(x,y);
```



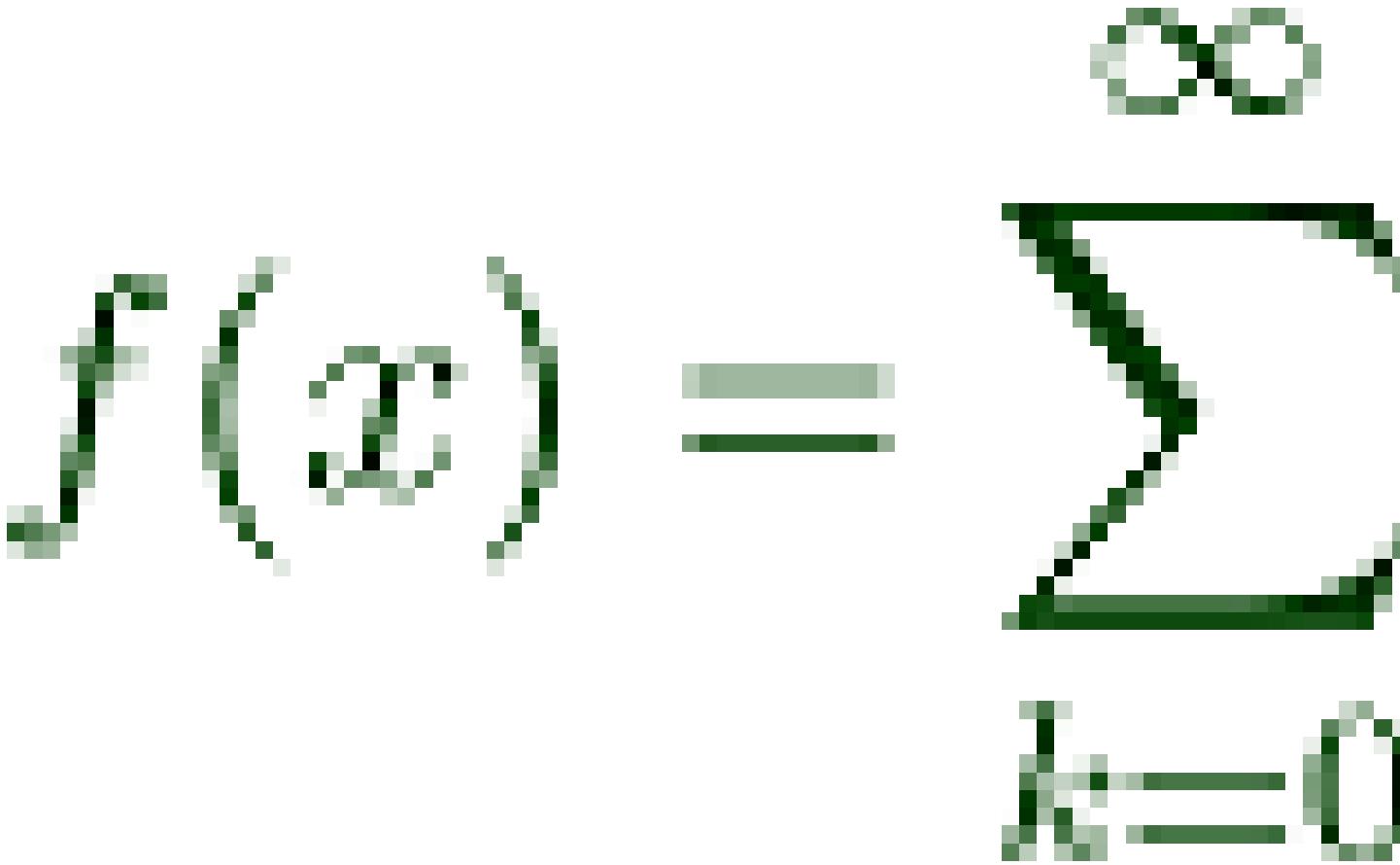
Tabel Fungsi

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:menghitung deret'}
```

$$\sum_{k=1}^n k = \frac{n(1+n)}{2}$$

```
>${' sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))'}
```

$$\sum_{k=0}^{\infty} \frac{1}{k+3^k} = \sum_{k=0}^{\infty} \frac{1}{k+3^k}$$

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: menghitung nilai e'}
```

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

```
>${' sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))'}
```

$$\sum_{k=1}^{\infty} \frac{(-1)^{-1+k}}{k} = -\sum_{x=1}^{\infty} \frac{(-1)^x}{x}$$

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), simpsum=true)
```

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{-1 + 2k} = \sum_{k=1}^{\infty} \frac{(-1)^k}{-1 + 2k}$$

```
> $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), simpsum=true), &forget
```

$$a \sum_{k=0}^{\infty} x^k = \frac{a}{1-x}$$

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)
```

$$d(n) = \sum_{k=2}^n \frac{1}{-k + k^2}$$

```
> $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 suku ke-11
```

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040} + \frac{x^8}{40320} + \frac{x^9}{362880} + \frac{x^{10}}{3628800}$$

```
>$' log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1
```

$$\log x = -1 - \frac{(-1+x)^2}{2} + \frac{(-1+x)^3}{3} - \frac{(-1+x)^4}{4} + \frac{(-1+x)^5}{5} - \frac{(-1+x)^6}{6} + \frac{(-1+x)^7}{7} - \frac{(-1+x)^8}{8} + \frac{(-1+x)^9}{9} - \frac{(-1+x)^{10}}{10} + \dots$$

BAB 6

PEKAN 11-12: MENGGUNAKAN EMT UNTUK GEOMETRI

article
eumat

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 geometri 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
```

koordinat

```
setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas
```

sumbu-x dan y adalah -r sd r

```
plotPoint (P, "P"): menggambar titik P dan diberi label "P"  
plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label
```

"AB" sejauh d

```
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 vektor v 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
angleBisection(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 lingkaran 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 dengan titik A pada
```

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

$\sin(\alpha)^2$ dengan

alpha sudut yang menghadap sisi a.
crosslaw(a,b,c,sa) : persamaan 3 quads dan 1 spread pada segitiga

dengan panjang sisi a, b, c.

triplespread(sa,sb,sc) : persamaan 3 spread sa,sb,sc yang memebntuk suatu segitiga

doublespread(sa) : Spread sudut rangkap Spread 2ϕ , dengan
 $sa=\sin(\phi)^2$ spread a.

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 mari kita atur tiga titik dan gambarkan.

```
>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 melibatkan fungsi-fungsi untuk membuat garis dan lingkaran. Format untuk garis adalah [a, b, c], yang mewakili garis dengan persamaan $ax + by = c$.

```
>lineThrough(B,C) // garis yang melalui B dan C
```

[-1, 2, 2]

Hitung garis tegak lurus melalui titik A pada garis BC.

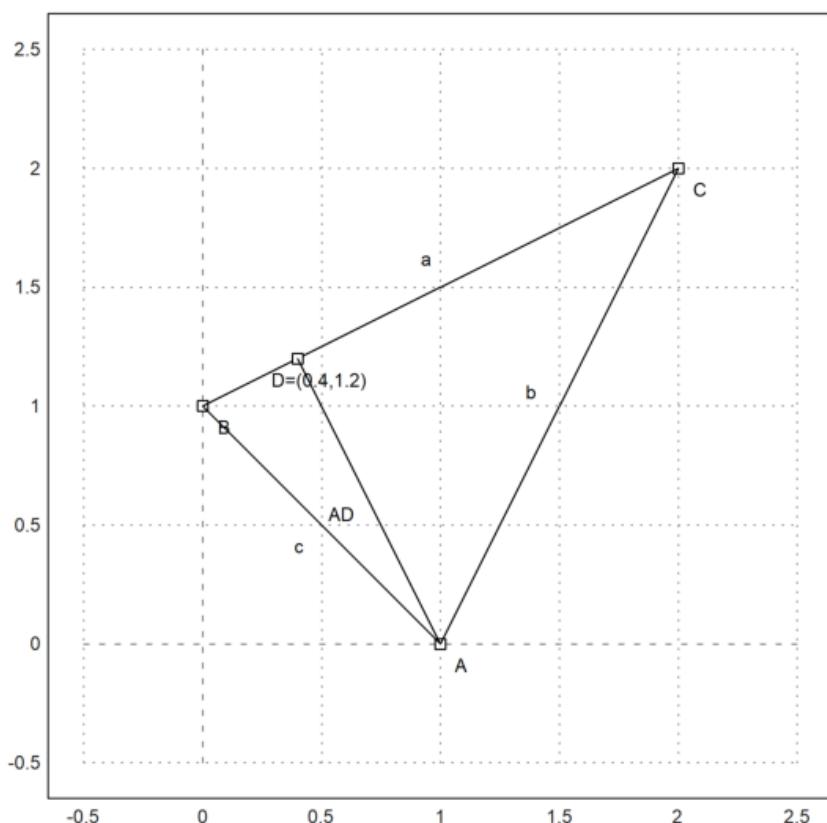
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan temukan titik perpotongannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Gambarkan.

```
>plotPoint(D,value=1); // koordinat D ditampilkan  
>aspect(1); plotSegment(A,D); // tampilkan semua gambar hasil plot...()
```



Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2} AD \cdot BC.$$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
```

Bandingkan dengan rumus determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langsung dengan fungsi
```

1.5

Cara lain menghitung luas segitiga ABC:

```
>distance(A,D)*distance(B,C)/2
```

1.5

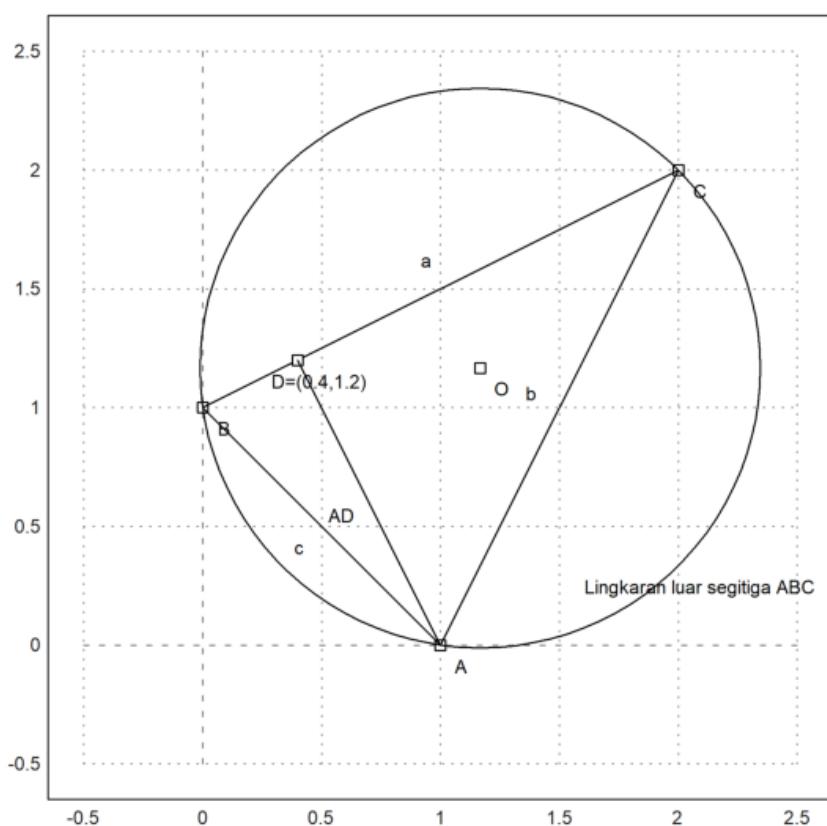
Sudut di titik C.

```
>degrprint(computeAngle(B,C,A))
```

$36^\circ 52' 11.63''$

Sekarang, lingkaran melingkari segitiga.

```
>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
```

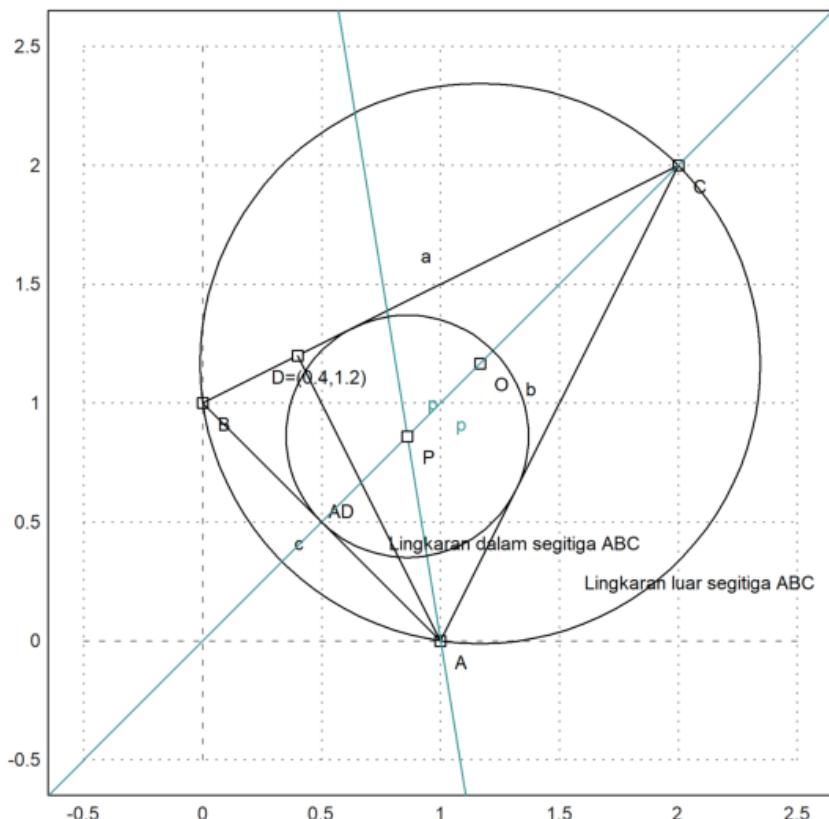
```
[0.86038, 0.86038]
```

Tambahkan segalanya ke dalam plot.

```
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut  
>plotPoint(P,"P"); // gambar titik potongnya  
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

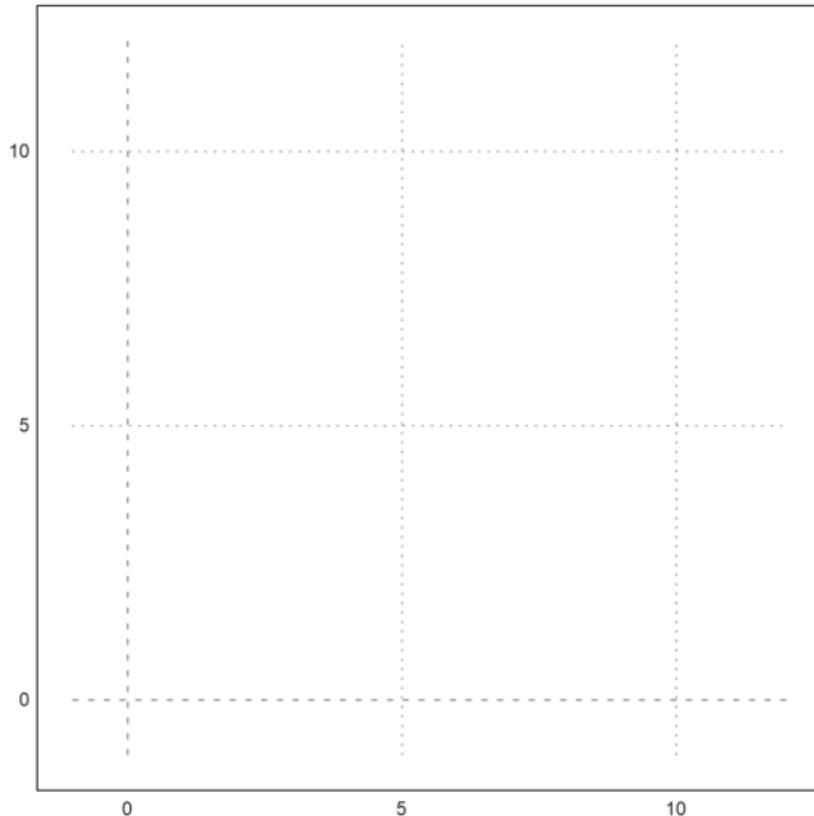
```
0.509653732104
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"); // gambar lingkaran dal
```

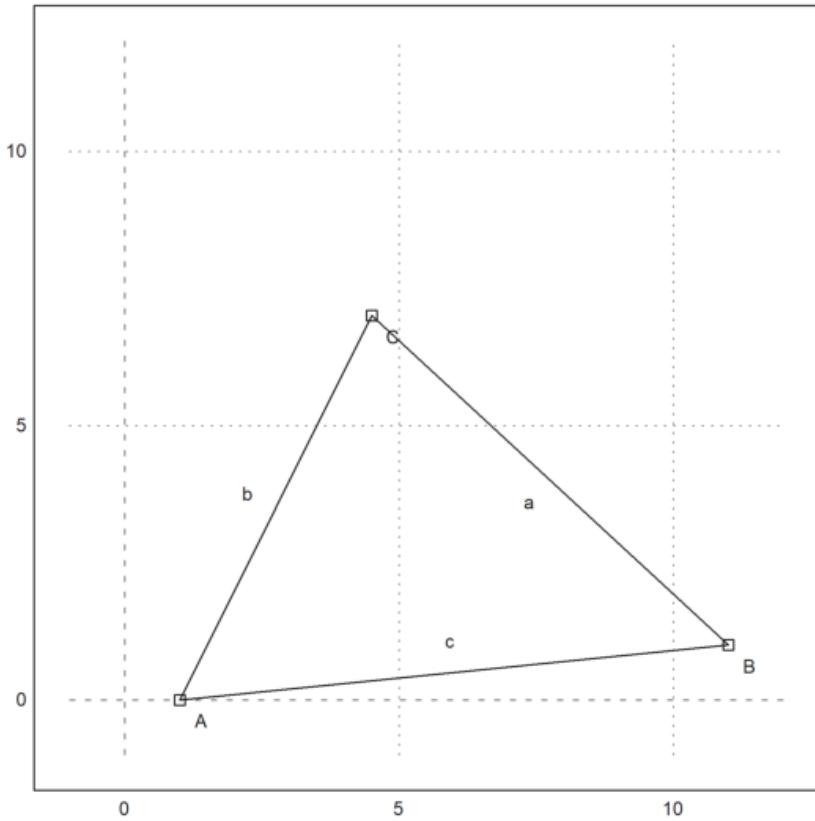


1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

```
>setPlotRange(-1,12,-1,12):
```



```
>A=[1,0]; plotPoint(A,"A");
>B=[11,1]; plotPoint(B,"B");
>C=[4.5,7]; plotPoint(C,"C");
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
>plotSegment(A,C,"b");
```



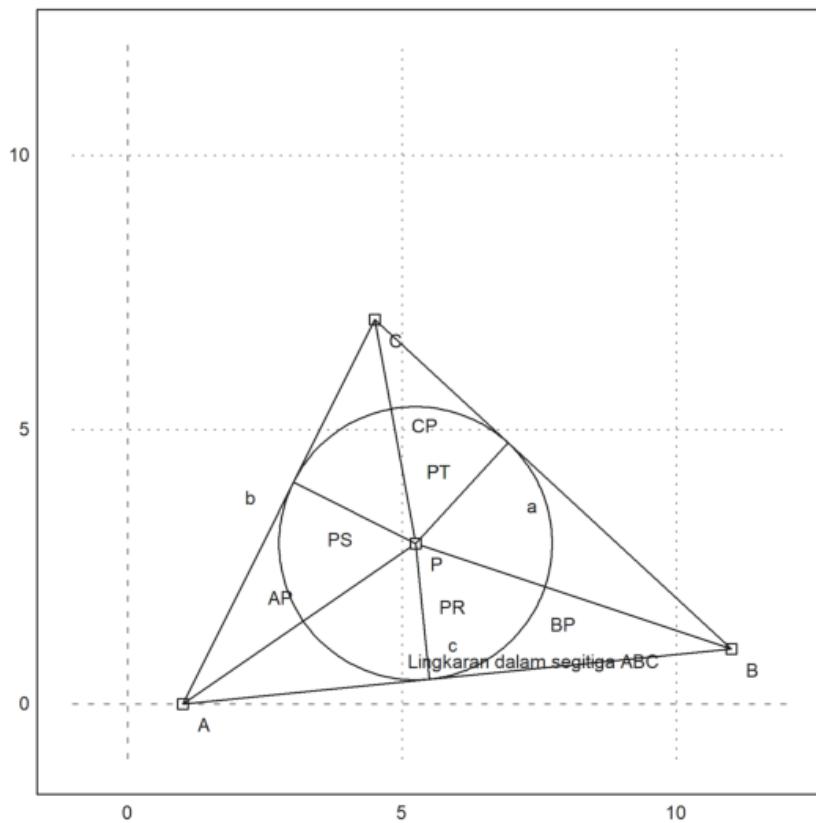
```
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>i=angleBisector(A,B,C); // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

[5.24507, 2.9255]

```
>plotSegment(A,P); plotSegment(B,P); plotSegment(C,P);
>plotPoint(P,"P"); // gambar titik potongnya
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

2.4885846425

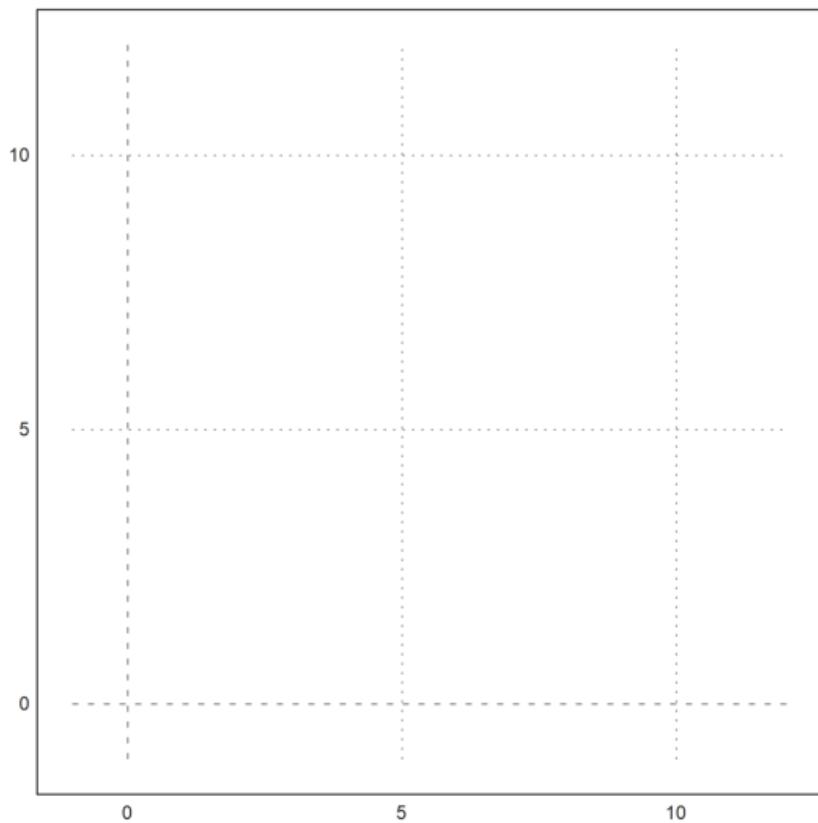
```
>h=perpendicular(P,lineThrough(A,B));
>R=lineIntersection(h,lineThrough(A,B));
>plotSegment(P,R);
>j=perpendicular(P,lineThrough(A,C));
>S=lineIntersection(j,lineThrough(A,C));
>plotSegment(P,S);
>k=perpendicular(P,lineThrough(C,B));
>T=lineIntersection(k,lineThrough(C,B));
>plotSegment(P,T);
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"); // gambar lingkaran dal
```



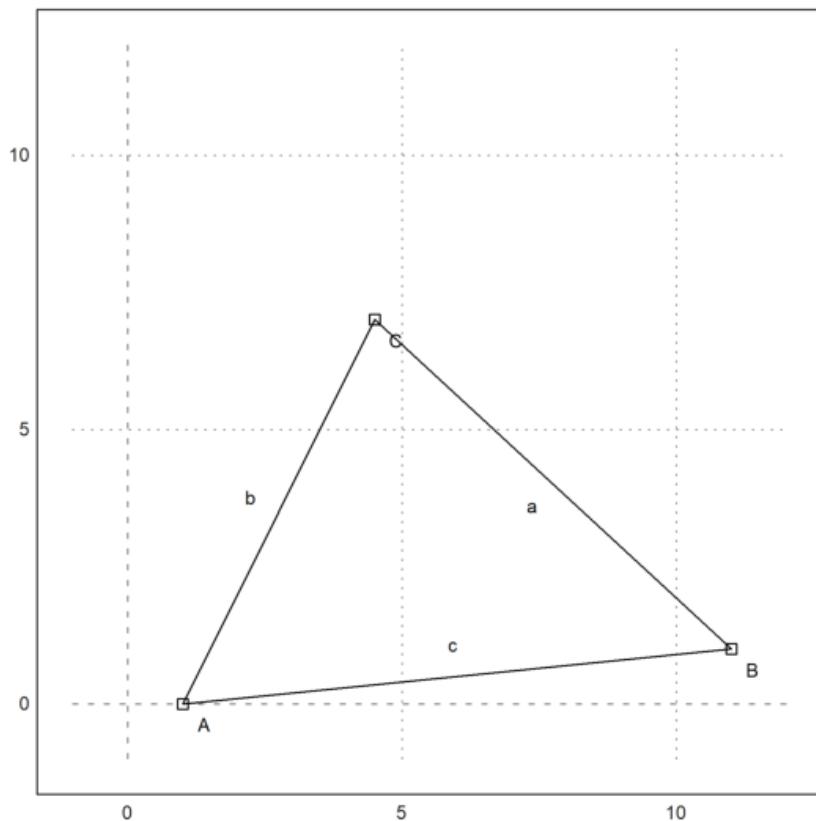
```
>reset();
```

2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.

```
>setPlotRange(-1,12,-1,12);
```



```
>A=[1,0]; plotPoint(A, "A");
>B=[11,1]; plotPoint(B, "B");
>C=[4.5,7]; plotPoint(C, "C");
>plotSegment(A,B, "c");
>plotSegment(B,C, "a");
>plotSegment(A,C, "b");
```



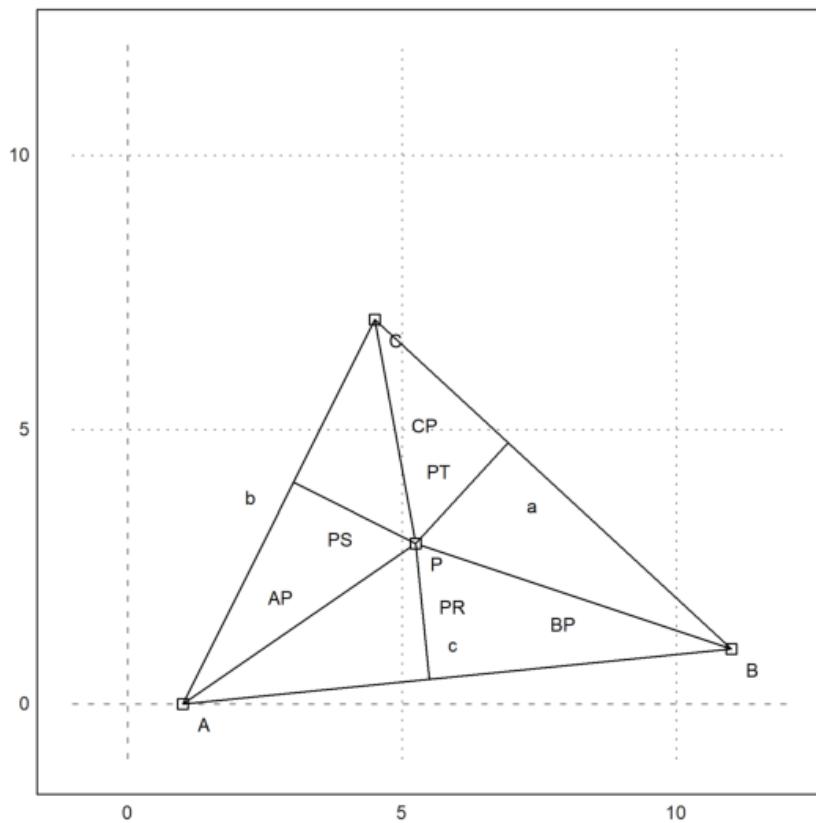
```
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>i=angleBisector(A,B,C); // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

[5.24507, 2.9255]

```
>plotSegment(A,P); plotSegment(B,P); plotSegment(C,P);
>plotPoint(P,"P"); // gambar titik potongnya
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

2.4885846425

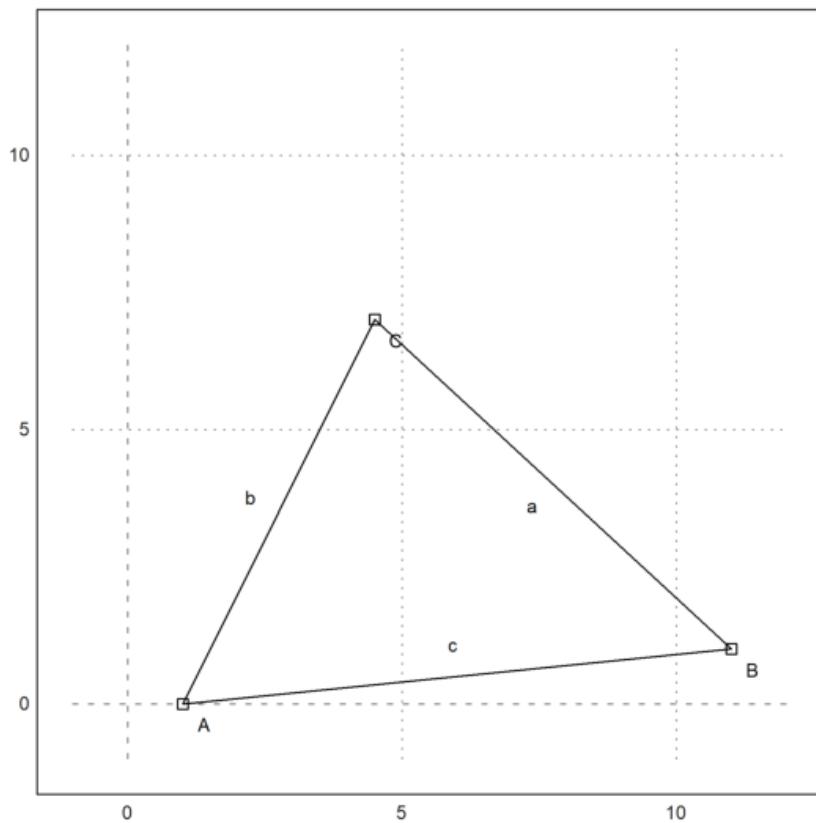
```
>h=perpendicular(P,lineThrough(A,B));
>R=lineIntersection(h,lineThrough(A,B));
>plotSegment(P,R);
>j=perpendicular(P,lineThrough(A,C));
>S=lineIntersection(j,lineThrough(A,C));
>plotSegment(P,S);
>k=perpendicular(P,lineThrough(C,B));
>T=lineIntersection(k,lineThrough(C,B));
>plotSegment(P,T):
```



```
>reset();
```

3. Hitung luas segitiga tersebut.

```
>setPlotRange(-1,12,-1,12);
>A=[1,0]; plotPoint(A,"A");
>B=[11,1]; plotPoint(B,"B");
>C=[4.5,7]; plotPoint(C,"C");
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
>plotSegment(A,C,"b");
```



```
>areaTriangle(A,B,C) // Menghitung luas segitiga
```

33.25

```
>reset();
```

4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

5. Gambar jari-jari lingkaran dalam.

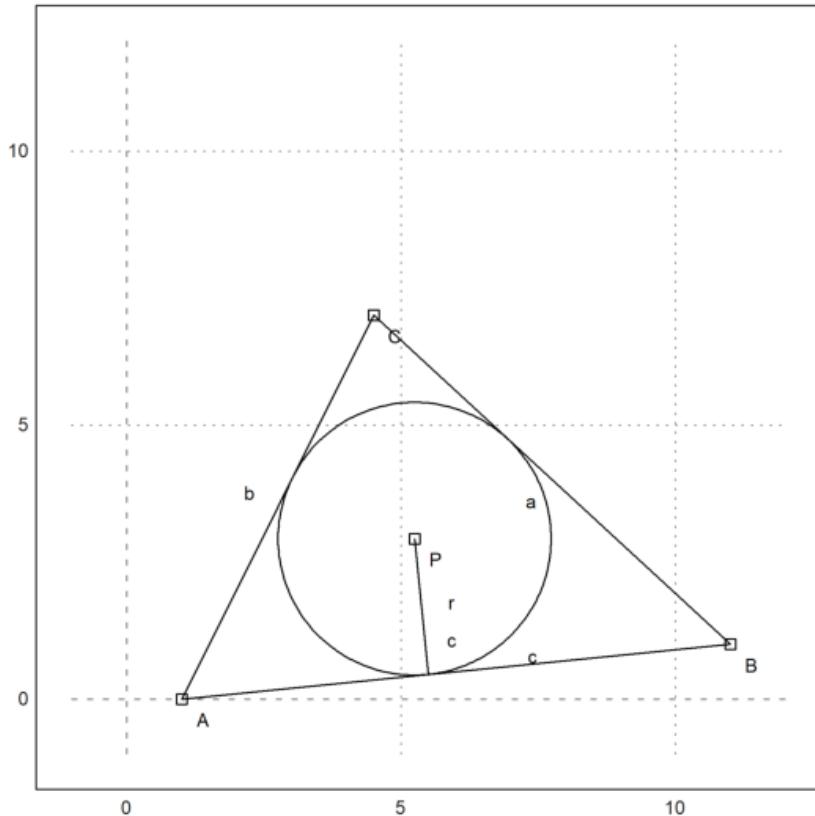
```
>setPlotRange(-1,12,-1,12);
>A=[1,0]; plotPoint(A,"A");
>B=[11,1]; plotPoint(B,"B");
>C=[4.5,7]; plotPoint(C,"C");
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
>plotSegment(A,C,"b");
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>i=angleBisector(A,B,C); // garis bagi <CAB
>P=lineIntersection(l,g)
```

[5.24507, 2.9255]

```
>plotPoint(P, "P");
>r=norm(P-projectToLine(P,lineThrough(A,B))) //jari-jari lingkaran dalam
```

2.4885846425

```
>h=perpendicular(P,lineThrough(A,B));
>D=lineIntersection(h,lineThrough(A,B));
>plotSegment(P,D, "r");
>plotCircle(circleWithCenter(P,r)):
```



```
>reset();
```

6. Hitung luas lingkaran luar dan luas lingkaran dalam segitiga ABC. Adakah hubungan antara luas kedua lingkaran tersebut dengan luas segitiga ABC?

```
>setPlotRange(-5,12,-5,12);
>A=[1,0]; plotPoint(A,"A");
>B=[11,1]; plotPoint(B,"B");
>C=[4.5,7]; plotPoint(C,"C");
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
```

```

>plotSegment(A,C,"b");
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
>R=getCircleRadius(c); // jari-jari lingkaran luar
>O=getCircleCenter(c); // titik pusat lingkaran c
>O, R // Titik pusat dan jari-jari lingkaran luar

```

[5.85526, 1.94737]
5.23123542767

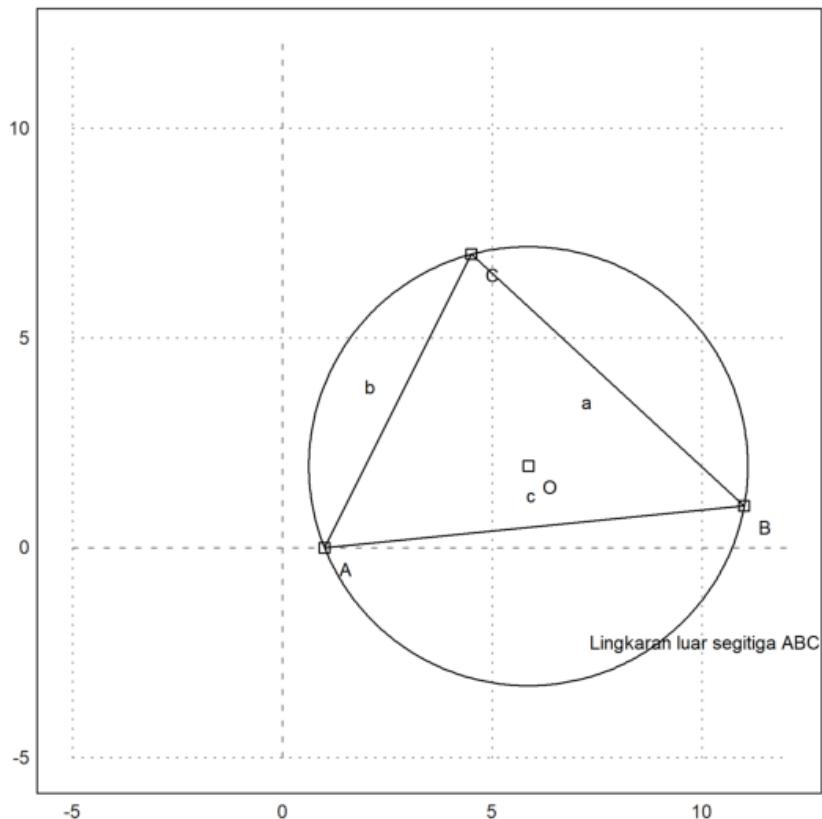
```
>pi*r^2 // Luas lingkaran luar
```

19.4560514507

```

>plotPoint(O,"O"); // Menggambar titik "O"
>plotCircle(c,"Lingkaran luar segitiga ABC"):

```



```

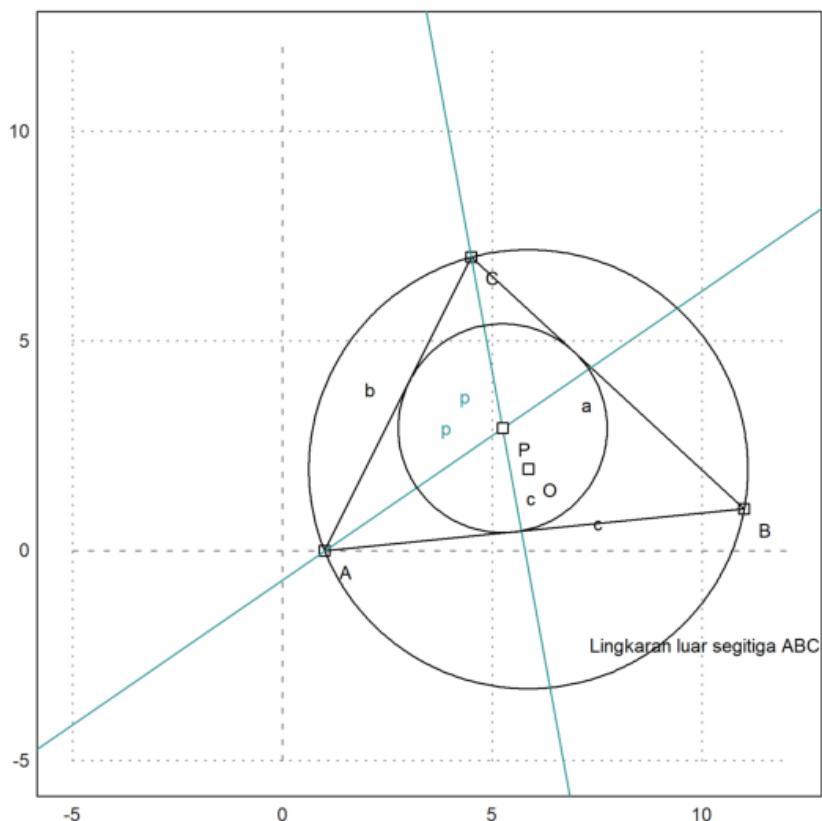
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>i=angleBisector(A,B,C); // garis bagi <CAB
>P=lineIntersection(l,g);
>color(5); plotLine(l); plotLine(g); color(1);
>plotPoint(P,"P");

```

```
>r=norm(P-projectToLine(P,lineThrough(A,B))); // Jari-jari lingkaran dalam
>pi*r^2 // Luas lingkaran dalam
```

19.4560514507

```
>plotCircle(circleWithCenter(P,r)):
```



```
>reset();
```

Contoh 2: Geometri Smbolik

Kita dapat menghitung geometri yang tepat dan simbolis menggunakan Maxima.

Berkas geometry.e menyediakan fungsi yang sama (dan lebih) dalam Maxima. Namun, kita dapat menggunakan perhitungan simbolis sekarang.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Fungsi-fungsi untuk garis dan lingkaran berfungsi seperti fungsi-fungsi Euler, tetapi memberikan perhitungan simbolis.

```
>c &= lineThrough(B,C) // c=BC
```

```
[ - 1, 2, 2]
```

Kita dapat dengan mudah mendapatkan persamaan untuk garis.

```
>$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) // persamaan garis melalui
```

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

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

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
```

$$(x_1 - 1) y - x y_1 = -y_1$$

```
>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
```

$$\begin{matrix} 2 & 6 \\ [-, -] \\ 5 & 5 \end{matrix}$$

```
>$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 melalui A, B, C
```

$$\left[\frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}} \right]$$

```
>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 bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 ga
```

$$\left[\frac{\sqrt{2}\sqrt{5}+2}{6}, \frac{\sqrt{2}\sqrt{5}+2}{6} \right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

$$[0.86038, 0.86038]$$

Perpotongan garis dan lingkaran

Tentu saja, kita juga dapat menggabungkan 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);
```

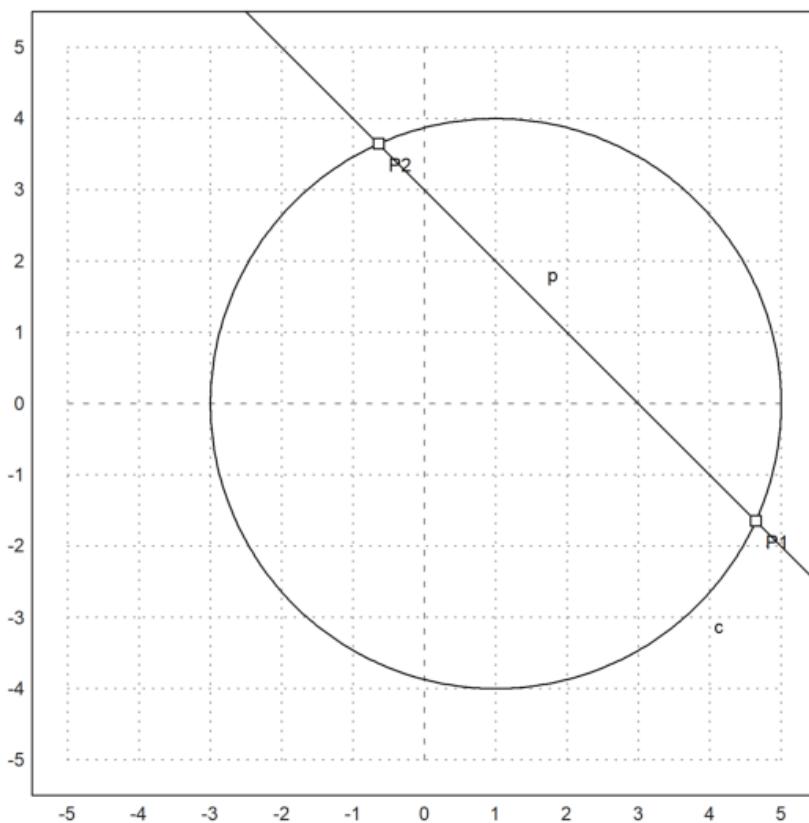
Hasil dari perpotongan garis dengan lingkaran adalah dua titik dan jumlah titik potongnya.

```
>{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 berlaku di 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 garis l
```

$$\left[\left[\sqrt{7} + 2, 1 - \sqrt{7} \right], \left[2 - \sqrt{7}, \sqrt{7} + 1 \right] \right]$$

Akan ditunjukkan bahwa sudut-sudut yang menghadap bsuusr yang sama adalah sama besar.

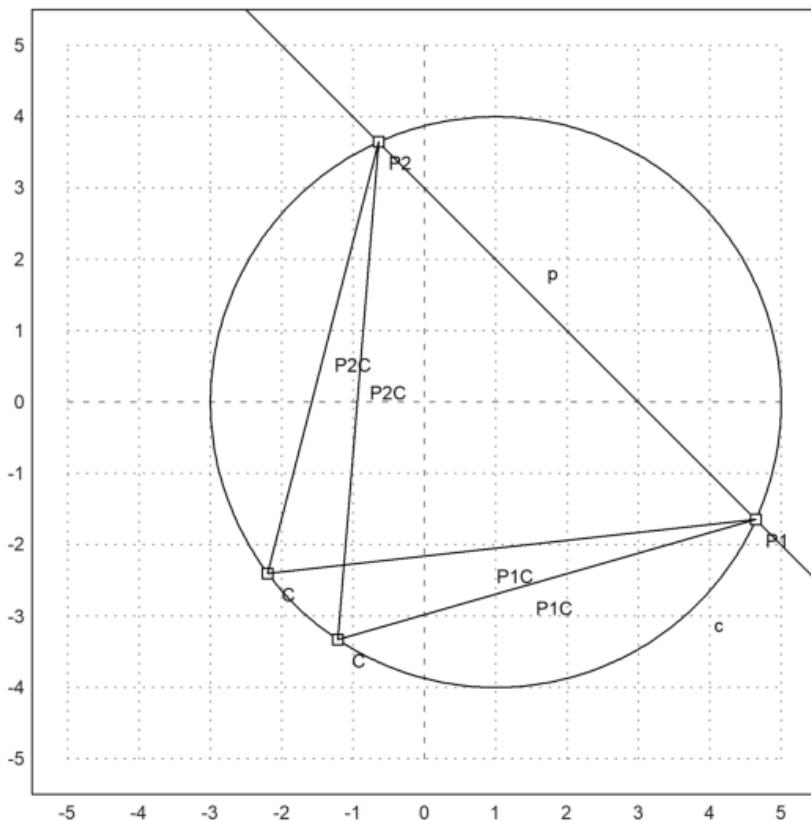
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>degrprint(computeAngle(P1,C,P2))
```

$69^\circ 17' 42.68''$

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>degrprint(computeAngle(P1,C,P2))
```

$69^\circ 17' 42.68''$

```
>insimg;
```

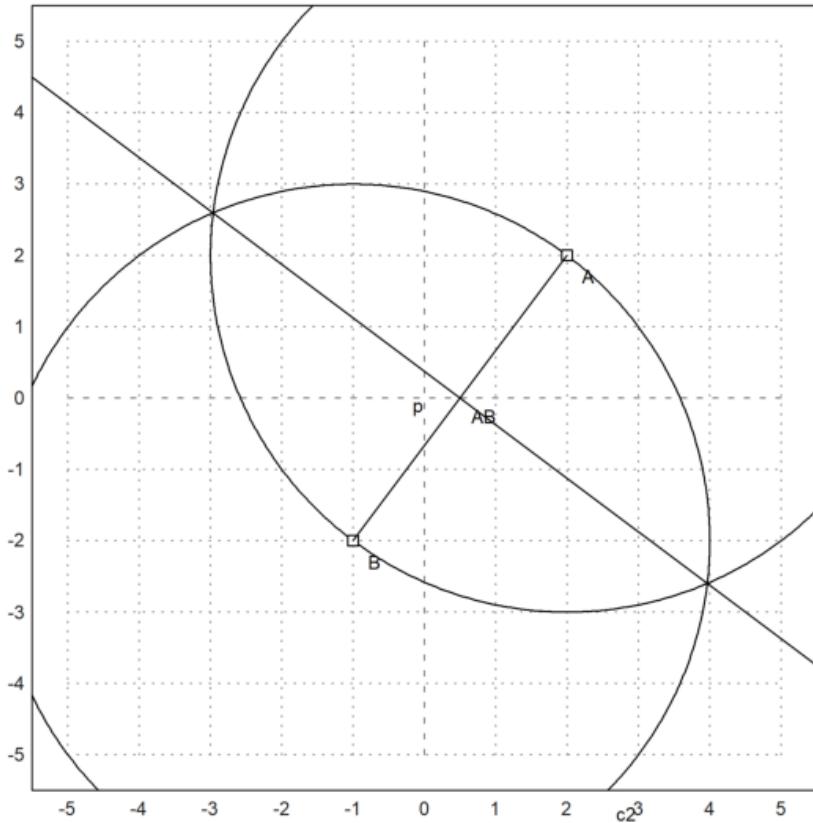


Garis Sumbu

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 melalui 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, kita lakukan 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 titik potongnya cukup rumit. Tetapi kita bisa menyederhanakan jika kita menyelesaikannya untuk nilai y.

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)
```

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

Ini memang sama dengan garis tengah tegak lurus, yang dihitung dengan cara yang sama sekali berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

```
>h &=getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
```

$$\left[y = \frac{(b_2 - a_2)x - a_1 b_2 + a_2 b_1}{b_1 - a_1} \right]$$

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:

$$L = \frac{1}{4} \sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}$$

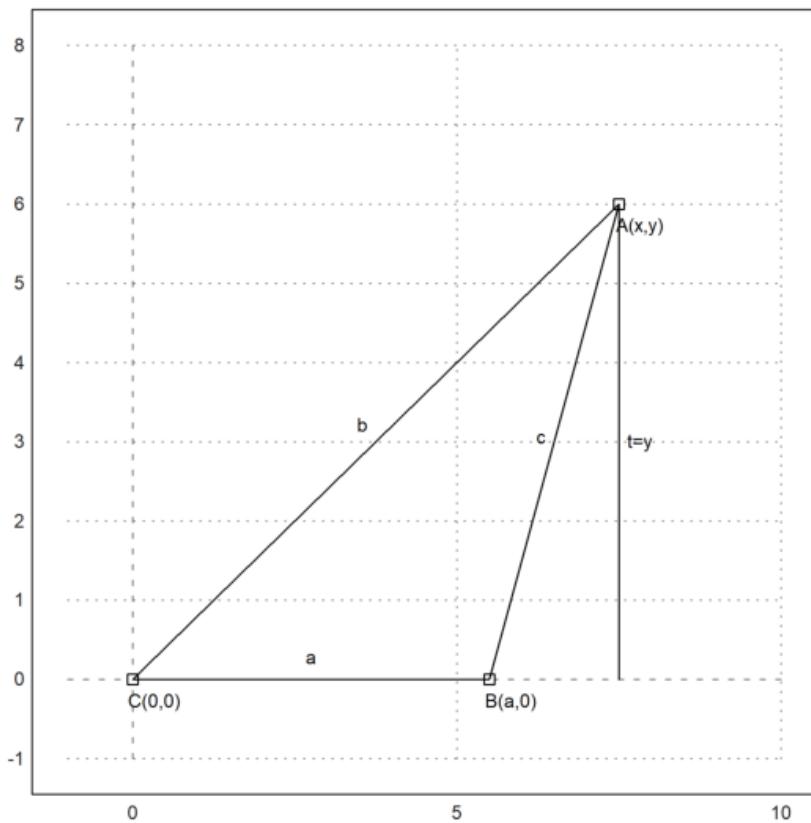
Untuk membuktikan hal ini kita misalkan C(0,0), B(a,0) dan A(x,y), b=AC, c=AB. Luas segitiga ABC adalah

$$L_{\triangle ABC} = \frac{1}{2}a \times y.$$

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], "B(a,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])
```

$$\begin{aligned}
 & -c^2 + b^2 + a^2 \\
 & [x = \frac{-c^2 + b^2 + a^2}{2a}, y = \\
 & \quad \frac{\sqrt{(-c^2 + 2b^2 + a^2)^2 + 2a^2c^2 - b^2 + 2ab^2 - a^4}}{2a}], \\
 & -\frac{-c^2 + b^2 + a^2}{2a} \\
 & [x = \frac{-c^2 + b^2 + a^2}{2a}, y = \\
 & \quad \frac{\sqrt{(-c^2 + 2b^2 + a^2)^2 + 2a^2c^2 - b^2 + 2ab^2 - a^4}}{2a}]
 \end{aligned}$$

Kita akan mencari solusi y dari masalah ini.

```
>ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2))
```

$$y = \frac{\sqrt{(-c+b+a)(c-b+a)(c+b-a)(c+b+a)}}{2a}$$

Kita mendapatkan rumus Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H(a,b,c) = \frac{\sqrt{(-c+b+a)(c-b+a)(c+b-a)(c+b+a)}}{4}$$

```
>$'Luas=H(2,5,6) // luas segitiga dengan panjang sisi-sisi 2, 5, 6
```

$$Luas = \frac{3\sqrt{39}}{4}$$

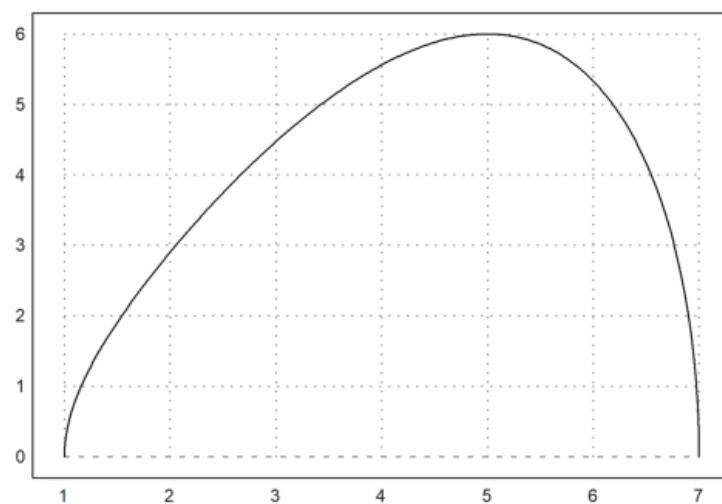
Tentu saja, setiap segitiga siku-siku adalah kasus yang sudah dikenal.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

6

Dan jelas pula bahwa segitiga ini memiliki luas maksimal dengan dua sisi 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7); // Kurva luas segitiga dengan panjang sisi 3, 4, x
```



Kasus umum juga berlaku.

```
>$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 temukan himpunan semua titik di mana $b + c = d$ untuk beberapa konstanta d . Sudah diketahui bahwa ini adalah sebuah elips.

```
>s1 &= subst(d-c,b,sol[2]); $s1
```

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

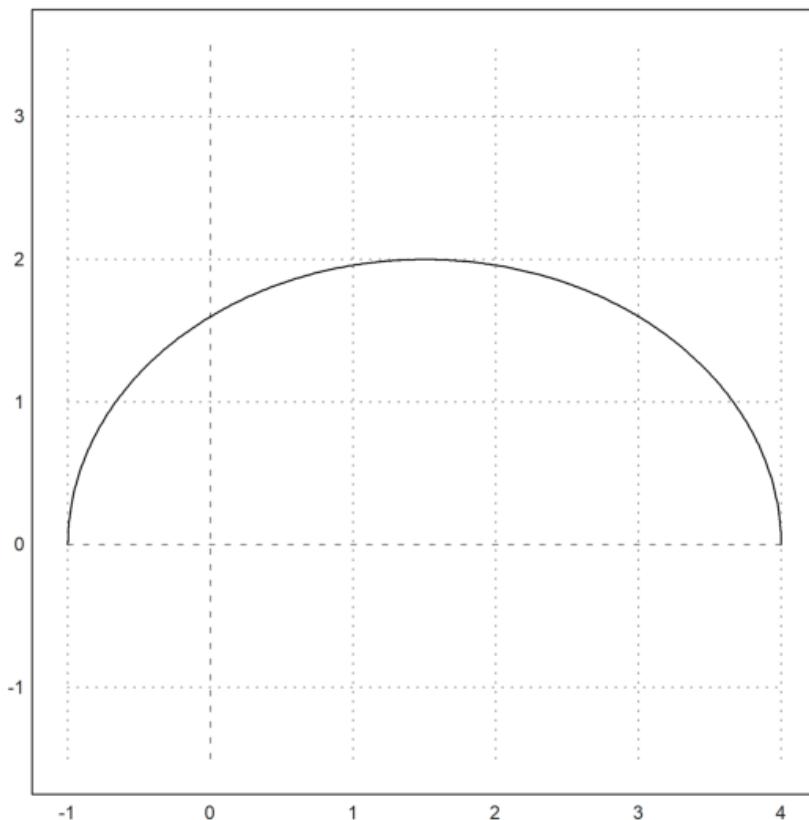
Dan buatlah fungsi-fungsi dari ini.

```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); $fy(a,c,d)
```

$$\begin{aligned} &\frac{\sqrt{-(d-c)^4 + 2c^2(d-c)^2 + 2a^2(d-c)^2 - c^4 + 2a^2c^2 - a^4}}{2a} \\ &\frac{\sqrt{-(d-c)^4 + 2c^2(d-c)^2 + 2a^2(d-c)^2 - c^4 + 2a^2c^2 - a^4}}{2a} \end{aligned}$$

Sekarang kita bisa menggambar himpunan ini. Sisi b bervariasi dari 1 hingga 4. Sudah diketahui bahwa kita akan mendapatkan sebuah 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

$$\frac{(x - x_m)^2}{u^2} + \frac{(y - y_m)^2}{v^2} = 1,$$

di mana (x_m, y_m) adalah pusatnya, dan u dan v adalah separuh sumbu-sumbunya.

```
>$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)/2])
```

1

Kita lihat bahwa tinggi dan oleh karena itu luas dari segitiga ini maksimal saat $x=0$. Dengan demikian, luas segitiga dengan $a+b+c=d$ maksimal jika segitiga tersebut berbentuk segitiga sama sisi. Kami ingin mendapatkan 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
```

$$\left[\frac{d(d-2a)(d-2b)}{8} - \frac{(-d+2b+2a)d(d-2b)}{8} = 0, \frac{d(d-2a)(d-2b)}{8} - \frac{(-d+2b+2a)d(d-2a)}{8} = 0 \right]$$

Kita mendapatkan beberapa minimum, yang sesuai dengan segitiga yang salah satu sisinya adalah 0, dan solusinya adalah $a=b=c=d/3$.

```
>$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, yang memaksimalkan $H(a,b,c)^2$ dengan mempertimbangkan $a+b+d=d$.

```
>&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])
```

$$\begin{aligned} & \left[\left[a = 0, b = \frac{d}{2}, c = \frac{d}{2}, la = 0 \right], \right. \\ & \left[\left[a = \frac{d}{2}, b = 0, c = \frac{d}{2}, la = 0 \right], \left[a = \frac{d}{2}, b = \frac{d}{2}, c = 0, la = 0 \right], \right. \\ & \left. \left[\left[a = \frac{d}{3}, b = \frac{d}{3}, c = \frac{d}{3}, la = \frac{d}{108} \right] \right] \right] \end{aligned}$$

Kita dapat membuat plot situasi tersebut.
Pertama, tentukan titik-titiknya di Maxima.

```
>A &= at([x,y],sol[2]); $A
```

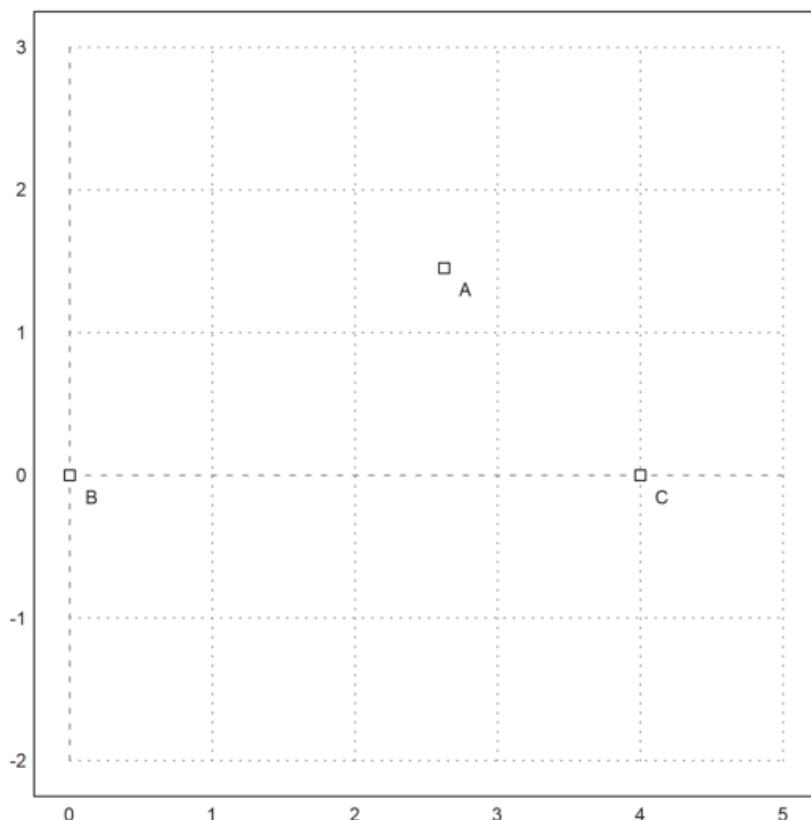
$$\left[\frac{-c^2 + b^2 + a^2}{2a}, \frac{\sqrt{-c^4 + 2b^2c^2 + 2a^2c^2 - b^4 + 2a^2b^2 - a^4}}{2a} \right]$$

```
>B &= [0,0]; $B, C &= [a,0]; $C
```

$$[a, 0]$$

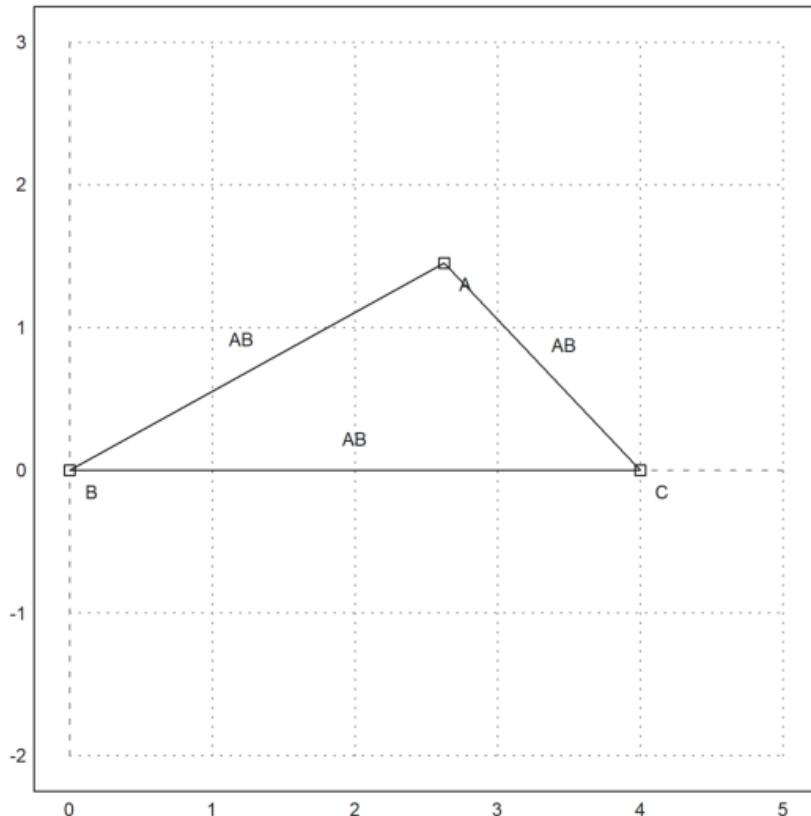
Kemudian tentukan rentang plot, dan gambarkan titik-titik tersebut.

```
>setPlotRange(0,5,-2,3); ...
>a=4; b=3; c=2; ...
>plotPoint(mxmeval("B"), "B"); plotPoint(mxmeval("C"), "C"); ...
>plotPoint(mxmeval("A"), "A");
```



Gambarkan segmennya.

```
>plotSegment(mxmeval("A"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("A"));
```



Hitung garis tengah tegak lurus di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan pusat lingkaran.

```
>U &= lineIntersection(h,g);
```

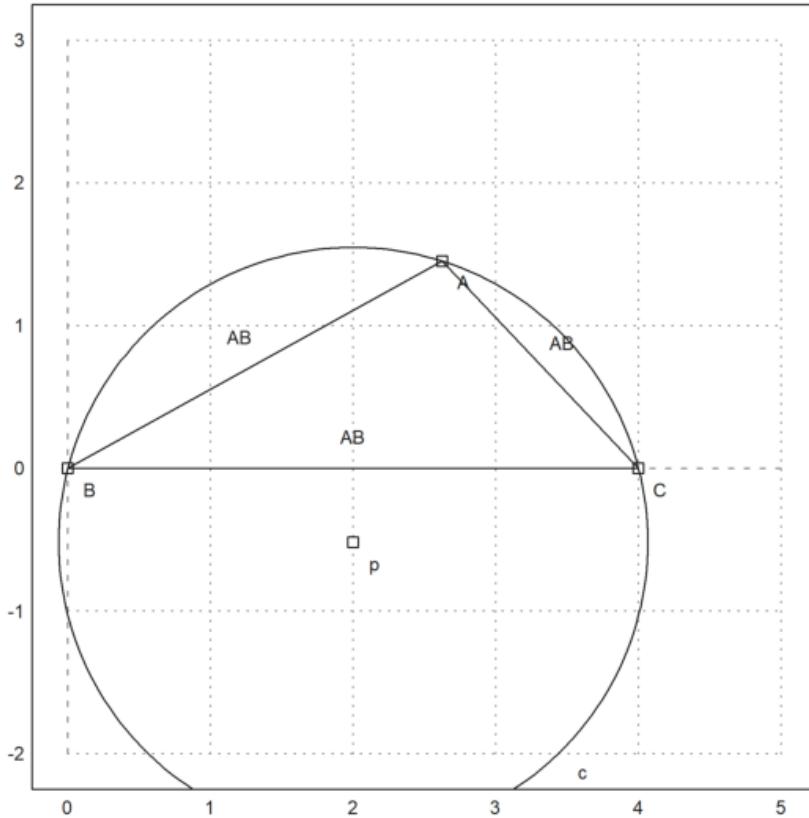
Kita akan mendapatkan rumus untuk jari-jari lingkaran luar.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

$$\frac{iac}{\sqrt{c-b-a}\sqrt{c-b+a}\sqrt{c+b-a}\sqrt{c+b+a}}$$

Tambahkan ini ke dalam plot.

```
>plotPoint(U()); ...
>plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
```



Menggunakan geometri, kita mendapatkan rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk jari-jari. Kita bisa memeriksa apakah ini benar dengan Maxima. Maxima akan menyederhanakan ini hanya jika kita mengkuadratkannya.

```
>$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 segitiga apa pun yang tidak sama sisi. Ini adalah garis pusat segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk ortosenter, sirkumsenter, centroid, titik Exeter, dan pusat dari lingkaran sembilan titik segitiga.

Untuk mendemonstrasikan ini, kita menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kita menentukan sudut-sudut segitiga dalam Euler. Kita menggunakan definisi yang terlihat dalam ekspresi simbolik.

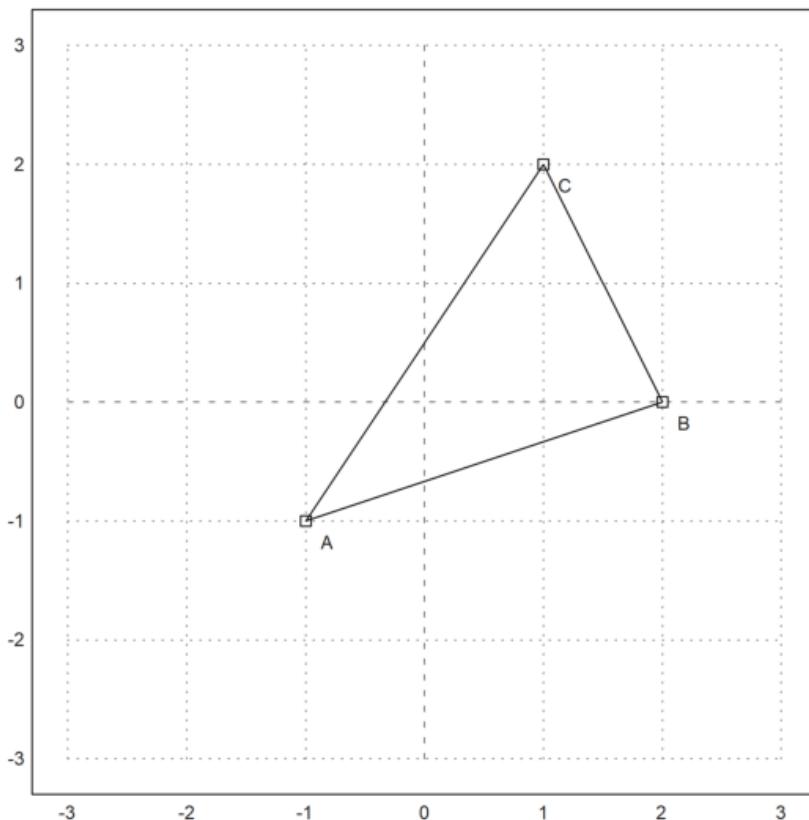
```
>A:=[-1,-1]; B:=[2,0]; C:=[1,2];
```

Untuk memplot objek-objek geometri, kita menyiapkan area plot, dan menambahkan titik-titik ke dalamnya. Semua plot objek-objek geometri ditambahkan ke dalam plot saat ini.

```
>setPlotRange(3); plotPoint(A, "A"); plotPoint(B, "B"); plotPoint(C, "C");
```

Kita juga dapat menambahkan sisi-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 mendapatkan rumus untuk garis ini.

```
>$getLineEquation(c,x,y)
```

$$3y - x = -2$$

Untuk bentuk Hesse, kita perlu menentukan sebuah titik, sehingga titik tersebut berada di sisi positif dari bentuk Hesse. Memasukkan titik tersebut menghasilkan jarak positif ke garis tersebut.

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$$\frac{7}{\sqrt{10}}$$

$$\frac{7}{\sqrt{10}}$$

Sekarang kita akan menghitung lingkaran luar dari ABC.

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

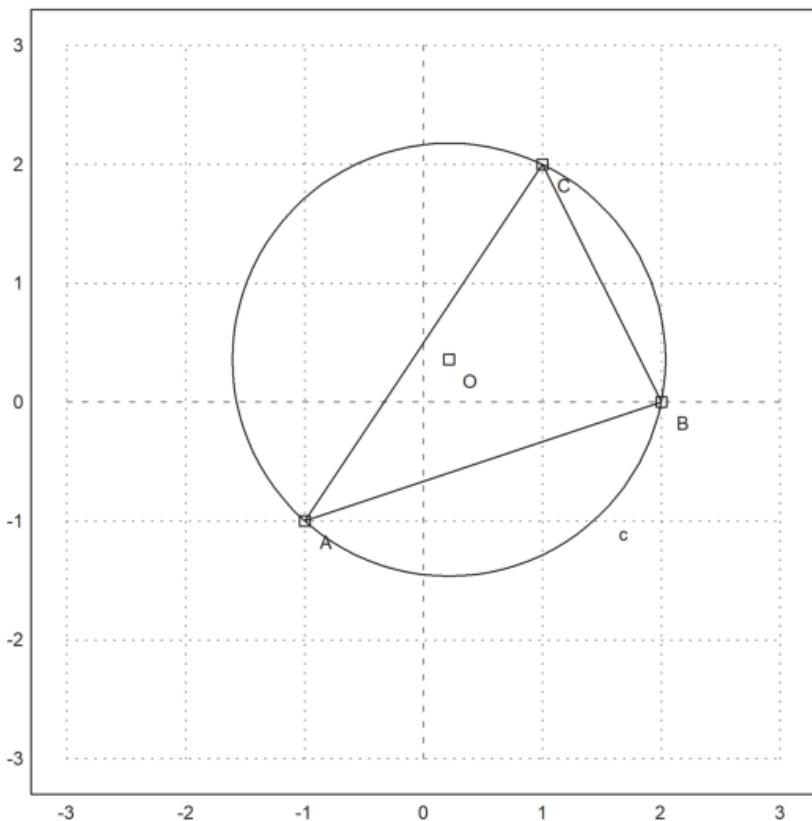
$$\left(y - \frac{5}{14}\right)^2 + \left(x - \frac{3}{14}\right)^2 = \frac{325}{98}$$

```
>O &= getCircleCenter(LL); $O
```

$$\left[\frac{3}{14}, \frac{5}{14}\right]$$

Gambar lingkaran dan pusatnya. Cu dan U adalah simbolis. Kita akan mengevaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O"):
```



Kita dapat menghitung titik potong dari tinggi dalam ABC (ortosentrum) secara numerik dengan perintah berikut.

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...
> perpendicular(B,lineThrough(A,C))); $H
```

$$\left[\frac{11}{7}, \frac{2}{7} \right]$$

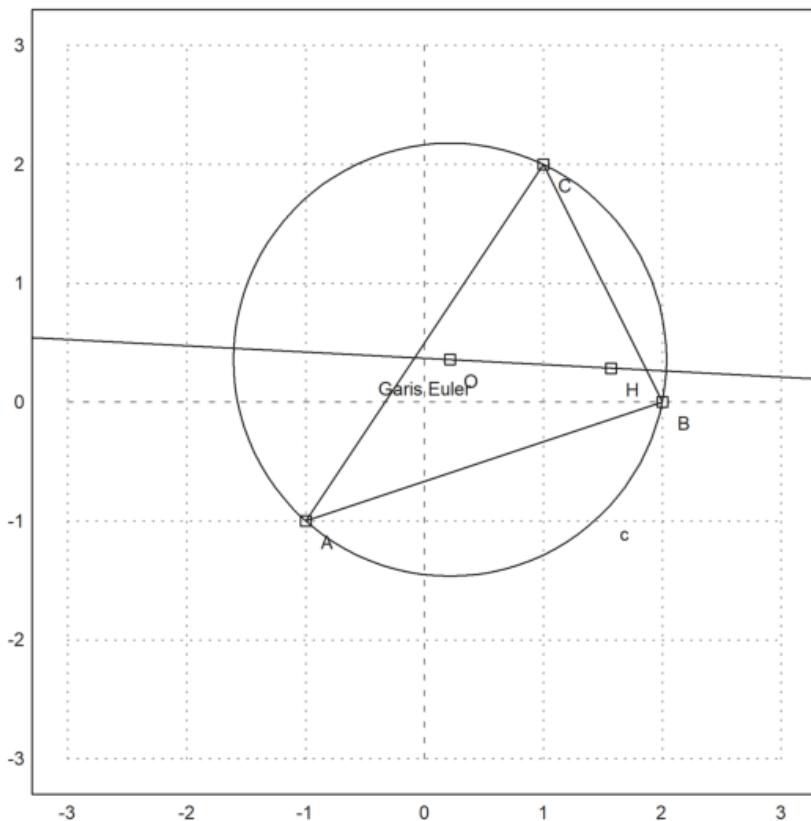
Sekarang kita dapat menghitung garis Euler dari segitiga ini.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$-\frac{19y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan garis ini ke plot kita.

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler");
```

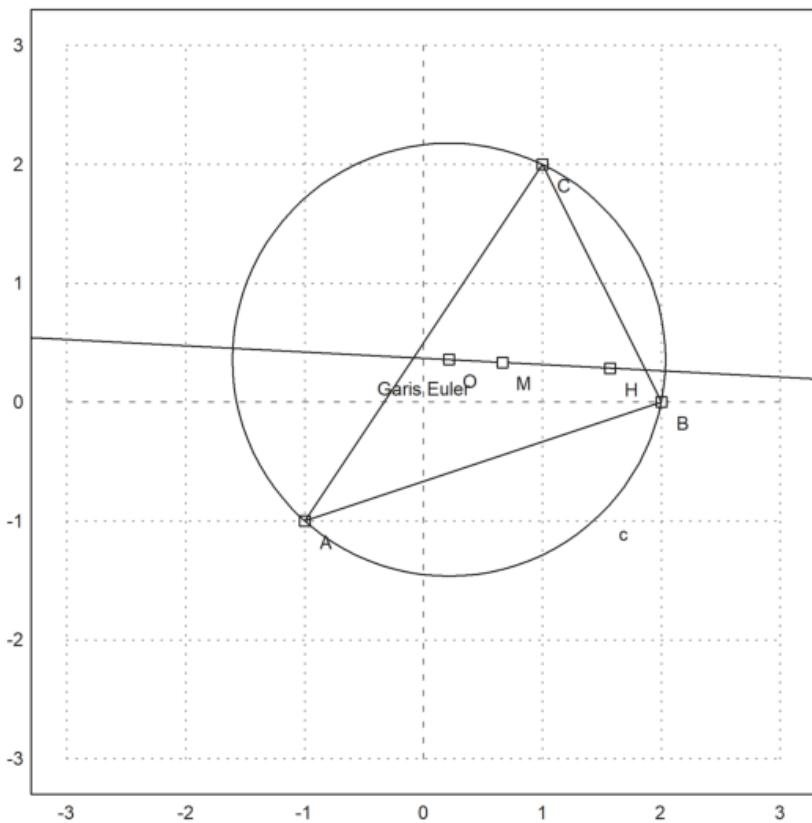


Pusat gravitasi seharusnya berada pada garis ini.

```
>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
```



Teori memberitahu kita bahwa $MH=2*MO$. Kita perlu menyederhanakannya dengan radcan untuk mencapai hal ini.

```
>$distance(M,H)/distance(M,O) | radcan
```

2

Fungsi-fungsi ini mencakup fungsi untuk sudut juga.

```
>$computeAngle(A,C,B), degprint(%())
```

$$\arccos\left(\frac{4}{\sqrt{5}\sqrt{13}}\right)$$

$60^\circ 15' 18.43''$

Persamaan untuk pusat lingkaran dalam tidak begitu bagus.

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)) | radcan; $Q
```

$$\left[\frac{\left(2^{\frac{3}{2}} + 1\right) \sqrt{5} \sqrt{13} - 15 \sqrt{2} + 3}{14}, \frac{(\sqrt{2} - 3) \sqrt{5} \sqrt{13} + 5 2^{\frac{3}{2}} + 5}{14} \right]$$

Mari kita juga menghitung ekspresi untuk jari-jari lingkaran dalam.

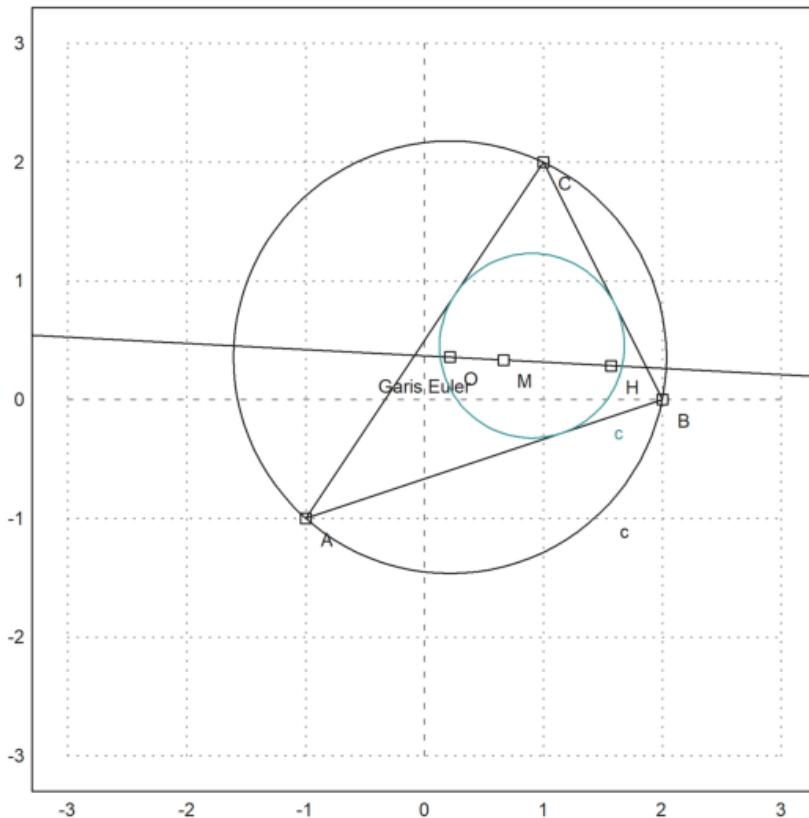
```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{(-41\sqrt{2} - 31)\sqrt{5}\sqrt{13} + 115\sqrt{2} + 614}}{7\sqrt{2}}$$

```
>LD &= circleWithCenter(Q,r); // Lingkaran dalam
```

Tambahkan ini ke plot kita.

```
>color(5); plotCircle(LD());:
```



Parabola

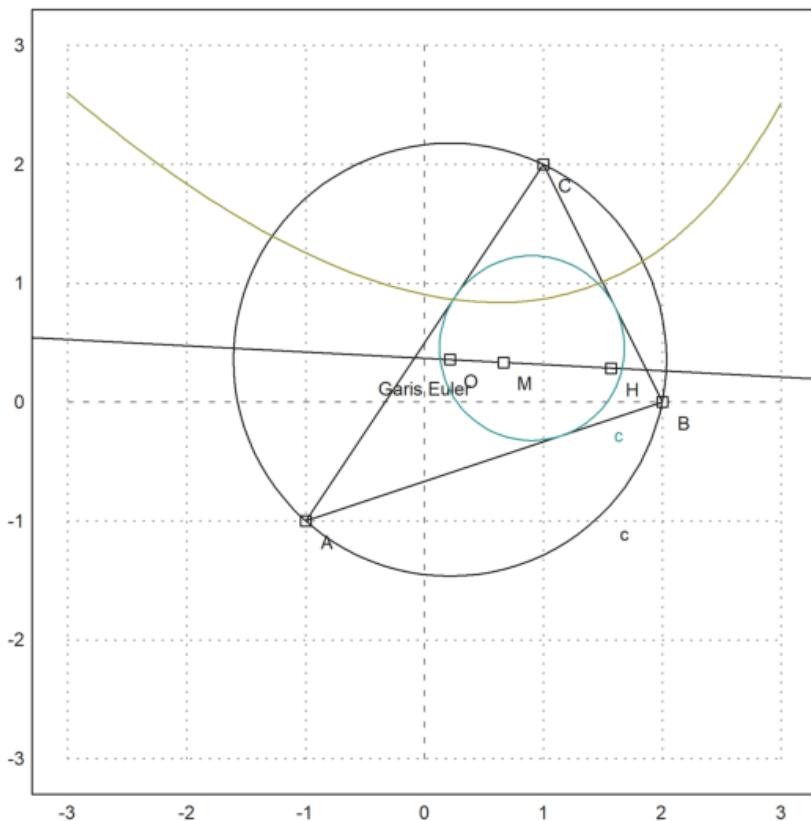
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3y - x + 2}{\sqrt{10}} - \sqrt{(2-y)^2 + (1-x)^2} = 0$$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```



Ini seharusnya menjadi suatu fungsi, tetapi pemecah masalah bawaan Maxima hanya dapat menemukan solusi jika kita mengkuadratkan persamaannya. Akibatnya, kita mendapatkan solusi palsu.

```
>akar &= solve(getHesseForm(lineThrough(A, B), x, y, C)^2-distance([x, y], C)^2, y)
```

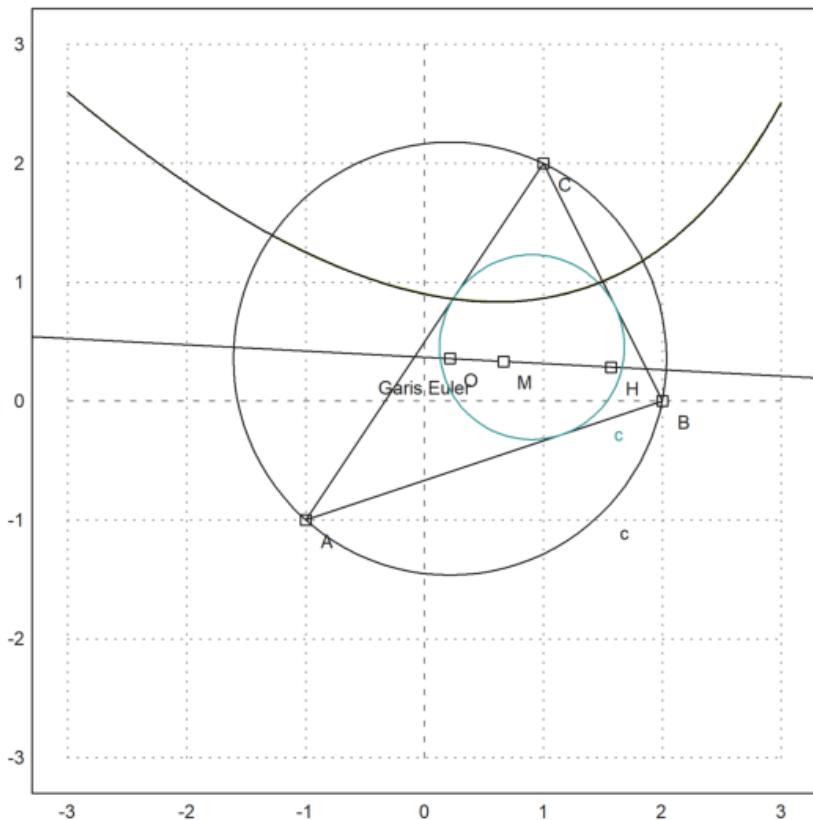
```
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26,
y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 26]
```

Solusi pertama adalah

$$y = -3x + \sqrt{70} \sqrt{9 - 2x} + 26$$

Dengan menambahkan solusi pertama ke plot, dapat dilihat bahwa itu memang merupakan jalur yang kita cari. Teori mengatakan bahwa ini adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]), add=1):
```



```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan kurva di atas
```

$$g(x) = -3x - \sqrt{70} \sqrt{9 - 2x} + 26$$

```
>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(%) // jatakan 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 oleh pidato N.J.Wildberger. Dalam bukunya "Divine Proportions," Wildberger mengusulkan untuk menggantikan konsep klasik tentang jarak dan sudut dengan kuadran dan spread. Dengan menggunakan ini, memungkinkan untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional."

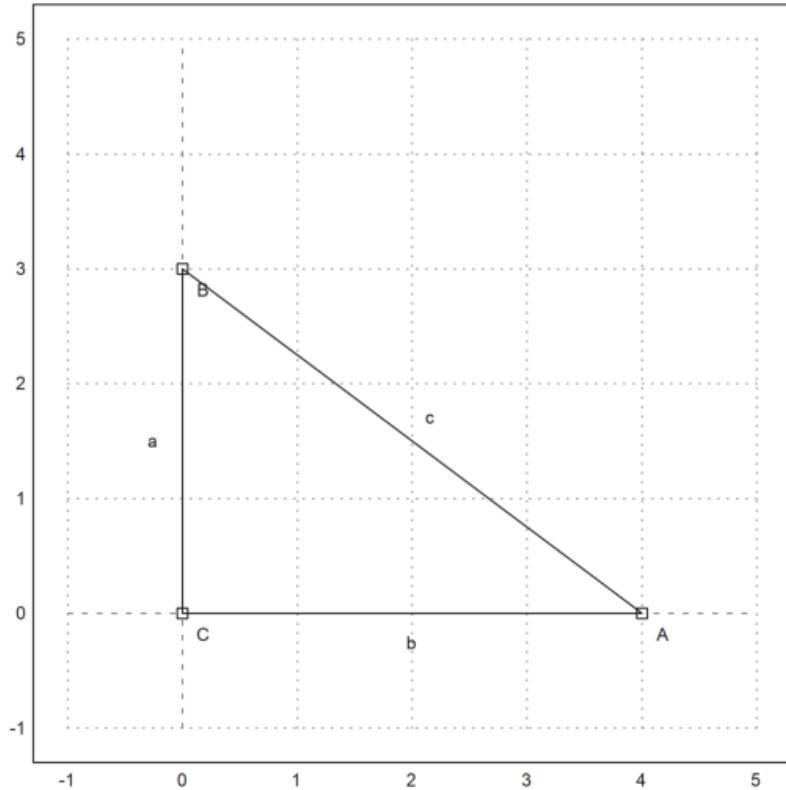
Pada berikut ini, saya akan memperkenalkan konsep-konsep tersebut dan menyelesaikan beberapa masalah. Saya menggunakan komputasi simbolik Maxima di sini, yang menyembunyikan keuntungan utama trigonometri rasional bahwa perhitungan dapat dilakukan hanya dengan kertas dan pensil. Anda diundang untuk memeriksa hasilnya tanpa komputer.

Intinya adalah bahwa perhitungan simbolik rasional sering menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya dievaluasi menjadi pendekatan numerik saja

```
>load geometry;
```

Sebagai pengantar pertama, kita akan menggunakan segitiga siku-siku dengan proporsi Mesir terkenal 3, 4, dan 5. Perintah berikut adalah perintah Euler untuk menggambar geometri datar yang terdapat dalam berkas 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"); ...
>insimsg(30);
```



Tentu saja,

$$\sin(w_a) = \frac{a}{c}$$

di mana w_a adalah sudut di A. Cara biasa untuk menghitung sudut ini adalah dengan mengambil invers dari fungsi sinus. Hasilnya adalah sudut yang sulit dimengerti, yang hanya bisa dicetak dalam bentuk perkiraan numerik.

```
>wa := arcsin(3/5); degprint(wa)
```

$36^\circ 52' 11.63''$

Trigonometri rasional berusaha menghindari hal ini.

Konsep pertama trigonometri rasional adalah kuadran, yang menggantikan jarak. Sebenarnya, itu hanya kuadrat jarak. Dalam hal berikut, a, b, dan c menunjukkan kuadran dari sisi-sisi tersebut.

Teorema Pythagoras hanya menjadi $a+b=c$ kemudian.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

$$25 = 25$$

Konsep kedua trigonometri rasional adalah spread. Spread mengukur pembukaan antara garis. Nilainya adalah 0 jika garis-garis sejajar dan 1 jika garis-garis tegak lurus. Ini adalah kuadrat dari sinus sudut antara dua garis tersebut.

Spread dari garis AB dan AC dalam gambar di atas didefinisikan sebagai

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

di mana a dan c adalah kuadran dari segitiga siku-siku dengan satu sudut di A.

```
>sa &= a/c; $sa
```

$$\frac{9}{25}$$

Ini lebih mudah dihitung daripada sudut, tentu saja. Tetapi Anda kehilangan sifat bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengonversi nilai perkiraan sudut wa ke dalam spread, dan mencetaknya dalam bentuk pecahan.

```
>fracprint(sin(wa)^2)
```

9/25

Hukum kosinus trigonometri klasik diterjemahkan ke dalam "hukum silang" berikut.

$$(c + b - a)^2 = 4bc, (1 - s_a)$$

Di sini, a, b, dan c adalah kuadran dari sisi-sisi segitiga, dan sa adalah spread di sudut A. Sisi a, seperti biasa, berlawanan dengan sudut A.

Hukum-hukum ini diimplementasikan dalam berkas geometri.e yang kita muat ke dalam Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

$$(cc + bb - aa)^2 = 4 bb cc (1 - saa)$$

Dalam kasus ini kita mendapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024 = 1024$$

Mari kita gunakan hukum silang ini untuk mencari sudut di A. Untuk melakukannya, kita akan menghasilkan hukum silang untuk kuadrans a, b, dan c, lalu menyelesaiakannya untuk sudut yang tidak diketahui. Anda dapat melakukannya dengan mudah secara manual, tetapi saya menggunakan Maxima. Tentu saja, kita akan mendapatkan hasil yang sudah kita ketahui.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

$$\left[x = \frac{9}{25} \right]$$

$$\left[x = \frac{9}{25} \right]$$

Kita sudah mengetahuinya sebelumnya. Definisi sudut adalah kasus khusus dari hukum silang.

Kita juga dapat menyelesaiakannya untuk a, b, c yang umum. Hasilnya adalah rumus yang menghitung sudut segitiga berdasarkan kuadrans dari tiga sisi.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[x = \frac{-cc^2 - (-2bb - 2aa)cc - bb^2 + 2aa\cdot bb - aa^2}{4bb\cdot cc} \right]$$

Kita dapat membuat fungsi dari hasil tersebut. Sebuah fungsi seperti itu sudah didefinisikan dalam file geometry.e milik Euler.

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi-sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya adalah bilangan rasional, yang tidak mudah diperoleh jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat.

```
>deprint(arcsin(sqrt(6/7)))
```

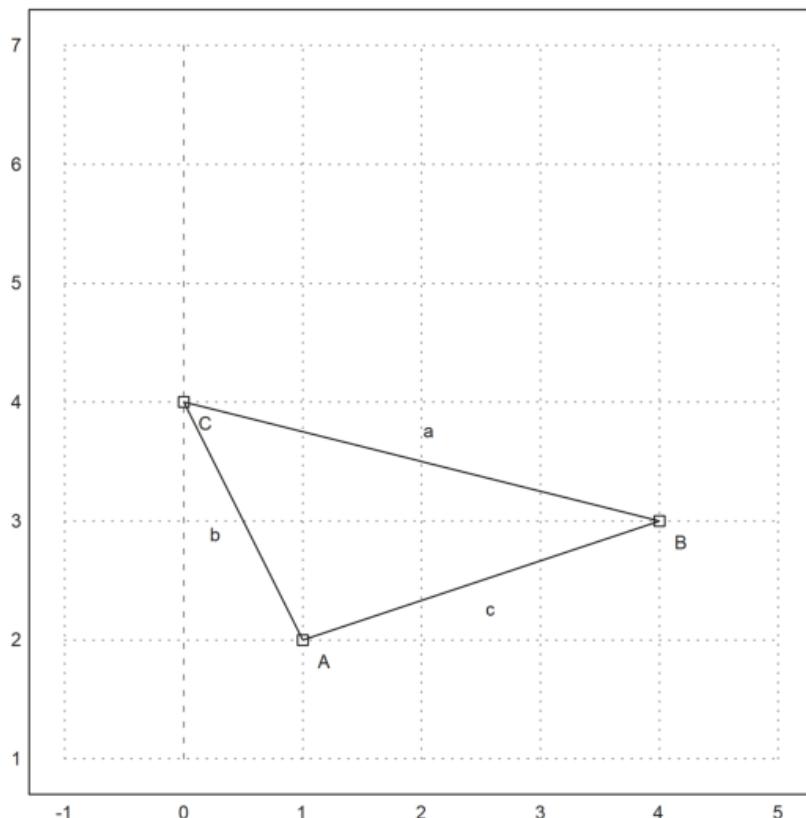
$$67^\circ 47' 32.44''$$

[Contoh Lainnya](#)

Sekarang, mari kita mencoba contoh yang lebih canggih.

Kita menetapkan 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 Pythagoras, mudah untuk menghitung jarak antara dua titik. Saya pertama kali menggunakan fungsi jarak dari file Euler untuk geometri. Fungsi jarak menggunakan geometri klasik.

```
>$distance(A,B)
```

$$\sqrt{10}$$

Euler juga berisi fungsi untuk kuadran antara dua titik.

Pada contoh berikut, karena $c + b$ bukanlah a , segitiga ini bukanlah segitiga siku-siku.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan hasil kali titik dari dua vektor. Hasilnya adalah perkiraan floating point.

$$A = \langle 1, 2 \rangle \quad B = \langle 4, 3 \rangle, \quad C = \langle 0, 4 \rangle$$

$$\mathbf{a} = C - B = \langle -4, 1 \rangle, \quad \mathbf{c} = A - B = \langle -3, -1 \rangle, \quad \beta = \angle ABC$$

$$\mathbf{a} \cdot \mathbf{c} = |\mathbf{a}| \cdot |\mathbf{c}| \cos \beta$$

$$\cos \angle ABC = \cos \beta = \frac{\mathbf{a} \cdot \mathbf{c}}{|\mathbf{a}| \cdot |\mathbf{c}|} = \frac{12 - 1}{\sqrt{17} \sqrt{10}} = \frac{11}{\sqrt{17} \sqrt{10}}$$

```
>wb &= computeAngle(A, B, C); $wb, $(wb/pi*180)()
```

$$\arccos \left(\frac{11}{\sqrt{10} \sqrt{17}} \right)$$

32.4711922908

Dengan pensil dan kertas, kita bisa melakukan hal yang sama dengan hukum silang. Kita memasukkan kuadran a, b, dan c ke dalam hukum silang dan menyelesaikan untuk x.

```
>$crosslaw(a, b, c, x), $solve(% , x), // (b+c-a)^=4b.c(1-x)
```

$$\left[x = \frac{49}{50} \right]$$

$$\left[x = \frac{49}{50} \right]$$

Itulah yang dilakukan oleh fungsi spread 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. Ia mengurangi istilah $\sin(\arccos(...))$ menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukannya.

```
>$sin(computeAngle(A, B, C))^2
```

$$\frac{49}{170}$$

Setelah kita memiliki spread di B, kita dapat menghitung tinggi ha di sisi a. Ingatlah bahwa

$$s_b = \frac{h_a}{c}$$

menurut definisi.

```
>ha &= c*sb; $ha
```

$$\frac{49}{17}$$

Gambar berikut ini telah diproduksi dengan program geometri C.a.R., yang dapat menggambar kuadran dan spread.

image: (20) Rational_Geometry_CaR.png

Menurut definisi, panjang ha adalah akar kuadratnya.

```
>$sqrt(ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang kita bisa menghitung luas segitiga. Jangan lupa, kita berurusan dengan kuadran!

```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{7}{2}$$

Rumus determinan biasa memberikan 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 menghitung sebaran di titik B untuk segitiga dengan sisi-sisi a, b, dan c. Kemudian kita menghitung luasnya yang dikuadratkan (mungkin "kuadrea"?), faktorkan dengan Maxima, dan kita akan mendapatkan rumus terkenal Heron.

Diakui bahwa 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 sudut-sudut yang tersebar adalah bahwa mereka tidak hanya ditambahkan seperti sudut-sudut biasa.

Namun, tiga sudut dari sebuah segitiga memenuhi aturan "triple spread" berikut ini.

```
>&remvalue(sa, sb, sc); $triplespread(sa, sb, sc)
```

$$(sc + sb + sa)^2 = 2 (sc^2 + sb^2 + sa^2) + 4 sa sb sc$$

Aturan ini berlaku untuk tiga sudut apa pun yang jumlahnya mencapai 180° .

$$\alpha + \beta + \gamma = \pi$$

Karena sudut-sudut dari

$$\alpha, \pi - \alpha$$

sama besar, aturan triple spread juga benar jika

$$\alpha + \beta = \gamma$$

Karena sudut negatif memiliki sudut yang sama, aturan triple spread juga berlaku jika

$$\alpha + \beta + \gamma = 0$$

Sebagai contoh, kita dapat menghitung sudut spread dari sudut 60° . Nilainya adalah $3/4$. Persamaan ini memiliki solusi kedua, di mana semua spread adalah 0.

```
>$solve(triplespread(x, x, x), x)
```

$$\left[x = \frac{3}{4}, x = 0 \right]$$

Sudut spread dari 90° jelas adalah 1. Jika dua sudut jumlahnya menjadi 90° , maka spread mereka memenuhi persamaan triple spread dengan a, b, dan 1. Dengan perhitungan berikut, kita mendapatkan $a+b=1$.

```
>$triplespread(x,y,1), $solve(%,x)
```

$$[x = 1 - y]$$

Karena sudut spread dari 180° -t sama dengan sudut spread dari t, rumus triple spread juga berlaku jika salah satu sudut adalah hasil penjumlahan atau selisih dari dua sudut lainnya.

Jadi kita dapat menemukan sudut spread dari sudut dua kali lipat. Perlu diingat bahwa ada dua solusi lagi. Kita dapat membuatnya sebagai suatu fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))
```

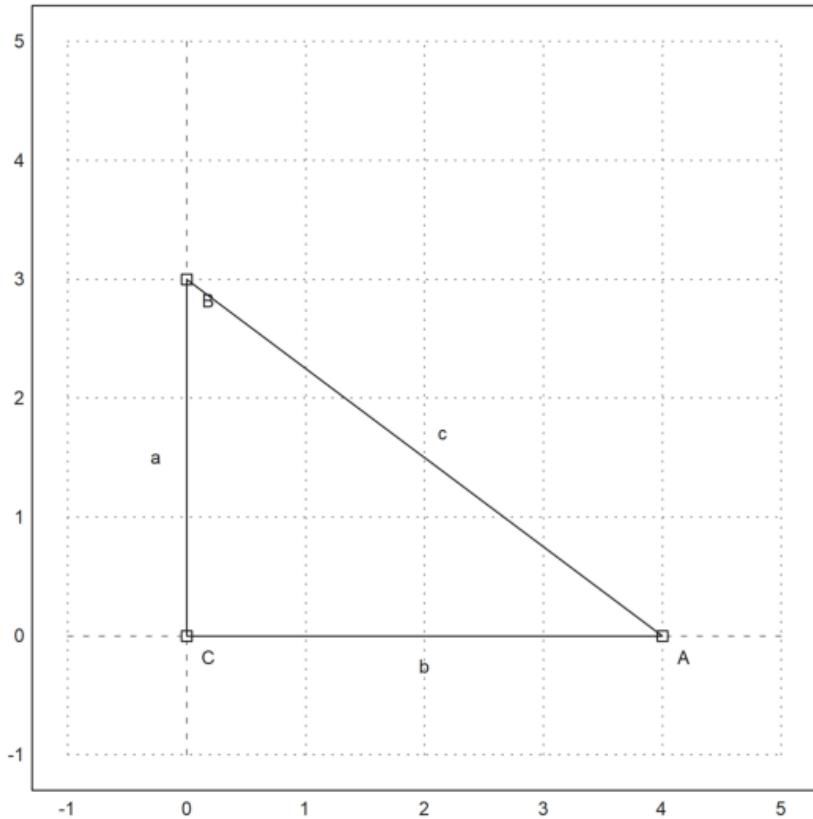
$$[x = 4a - 4a^2, x = 0]$$

$$- 4 (a - 1) a$$

Sudut Bisector

Situasinya sudah kita ketahui.

```
>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 penyegmentasi sudut di A. Namun, kita ingin menyelesaiakannya untuk nilai umum a, b, c.

```
>&remvalue(a,b,c);
```

Pertama, kita hitung sudut yang tersegmentasi di A, menggunakan rumus spread triple.

Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut yang tersegmentasi 180° -wa.

```
>$triplespread(x,x,a/(a+b)), $solve(% ,x), sa2 &= rhs(%[1]); $sa2
```

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

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

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

Mari kita periksa untuk segiempat Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

Kita dapat mencetak sudut dalam Euler, setelah mentransfer spread ke dalam radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

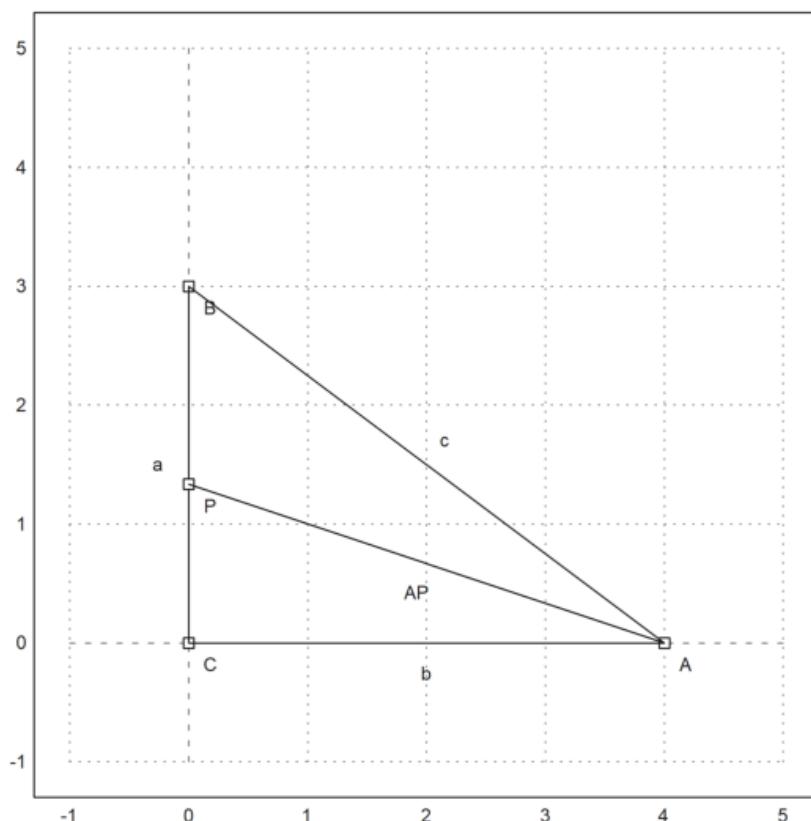
$18^\circ 26' 5.82''$

Titik P adalah hasil dari perpotongan penyegmentasi sudut dengan sumbu y.

```
>P := [0, tan(wa2)*4]
```

[0, 1.33333]

```
>plotPoint(P, "P"); plotSegment(A, P);
```



Mari kita periksa sudut-sudut dalam contoh khusus kita.

```
>computeAngle(C, A, P), computeAngle(P, A, B)
```

0.321750554397
0.321750554397

Sekarang kita hitung panjang penyegmentasi AP.

Kita 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 setiap segitiga. Kuadratkan, ini diterjemahkan menjadi "hukum spread" yang disebut

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

di mana a, b, c adalah kuadrans.

Karena spread CPA adalah $1-sa^2$, kita mendapatkan dari itu bisa/ $1-b/(1-sa^2)$ dan dapat menghitung bisa (kuadrans penyegmentasi sudut).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$\frac{2b(b+a)}{\sqrt{b}\sqrt{b+a}+b+a}$$

Mari kita periksa rumus ini untuk 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
```

$$-\frac{b(\sqrt{b}\sqrt{b+a}-b-a)}{\sqrt{b}\sqrt{b+a}+b+a}$$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

```
1.33333333333
```

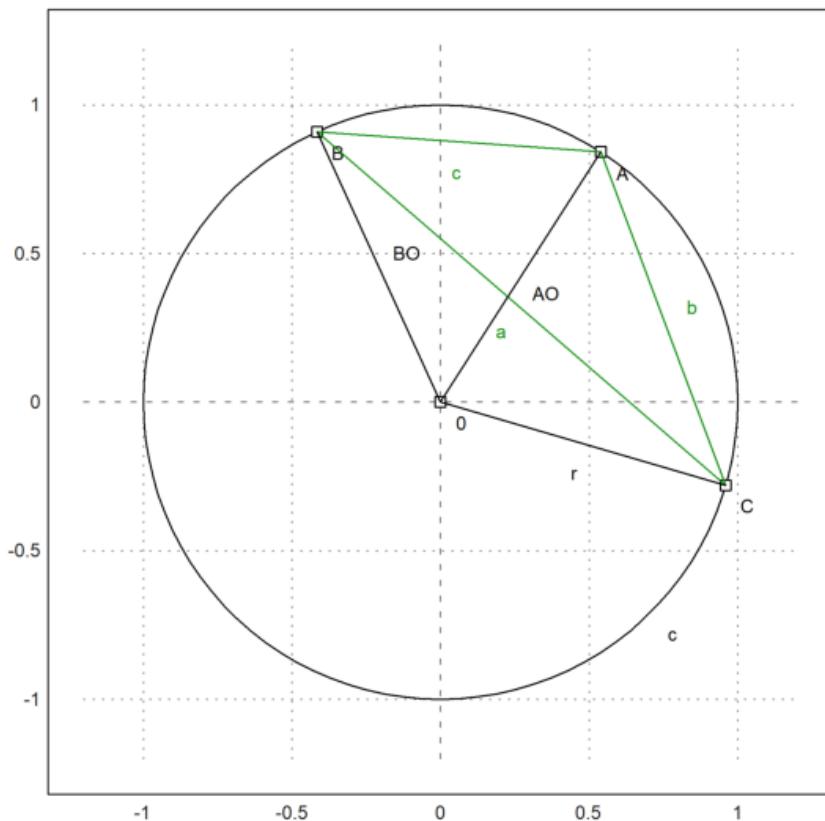
Sudut Chord

Mari kita lihat 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,"O"); ...
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
>insimg;

```



Kita dapat menggunakan Maxima untuk menyelesaikan formula spread triple untuk sudut-sudut di pusat O untuk r. Dengan demikian, kita mendapatkan formula untuk radius kuadrat perisirkel dalam hal kuadran dari sisi-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)),r)[4]); $rabc

```

$$-\frac{abc}{c^2 - 2bc + a(-2c - 2b) + b^2 + a^2}$$

Kita dapat menjadikannya sebagai fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasilnya untuk titik-titik A, B, C kita.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Jari-jarinya memang 1.

```
>periradius(a,b,c)
```

1

Faktanya adalah bahwa spread CBA hanya bergantung pada b dan c. Ini disebut teorema sudut chord.

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Faktanya, penyebarannya adalah $b/(4r)$, dan kita melihat bahwa sudut tali busur b adalah setengah sudut pusatnya.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

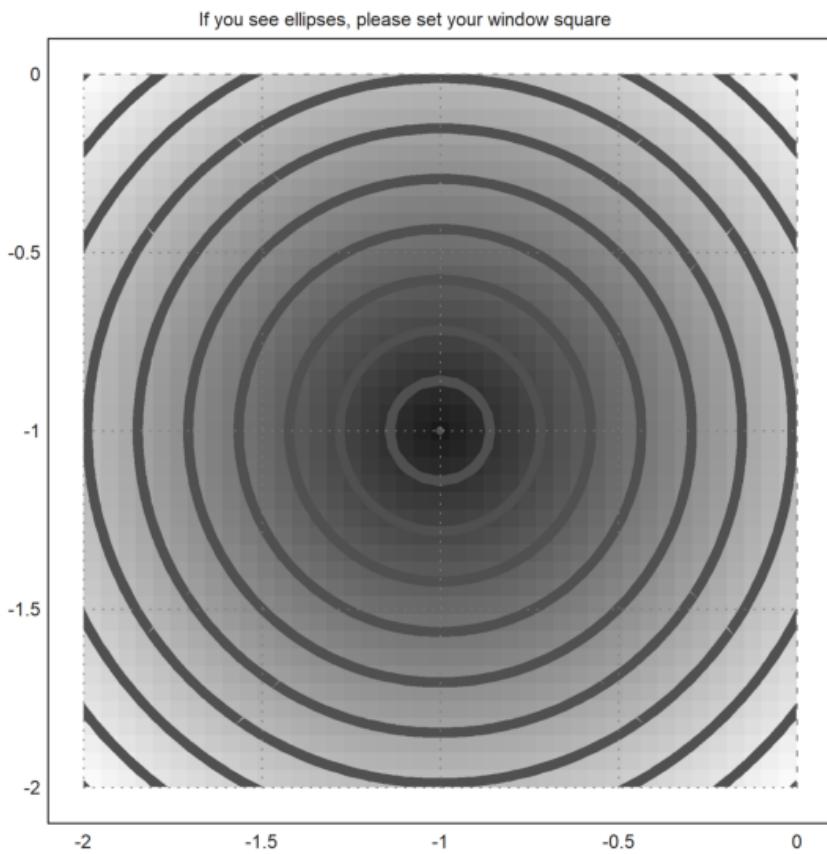
0

Contoh 6: Jarak Minimal pada Bidang

Catatan Awal

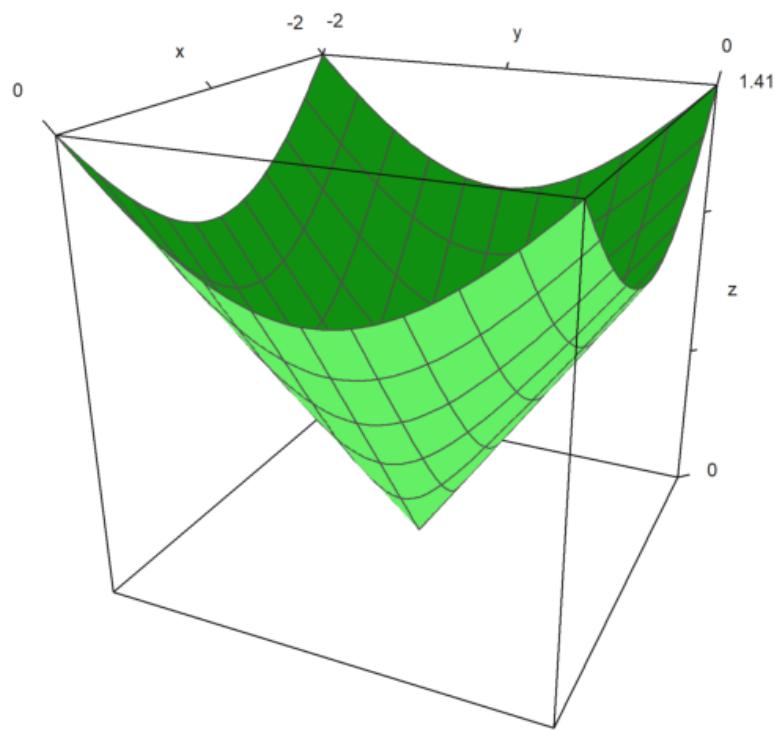
Fungsi yang menghubungkan titik M di bidang, menghasilkan jarak AM antara titik tetap A dan M, memiliki garis level yang cukup sederhana: lingkaran yang berpusat di A.

```
>&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 cukup sederhana: bagian atas kerucut:

```
>plot3d("d1", xmin=-2, xmax=0, ymin=-2, ymax=0):
```

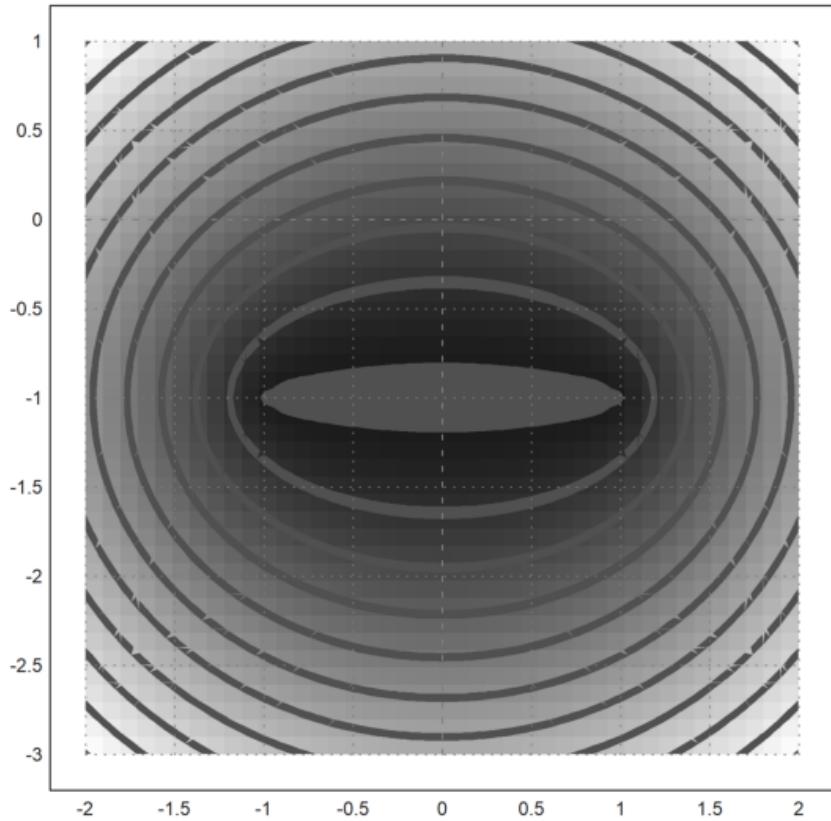


Tentu saja, minimum 0 dicapai di A.

Dua Titik

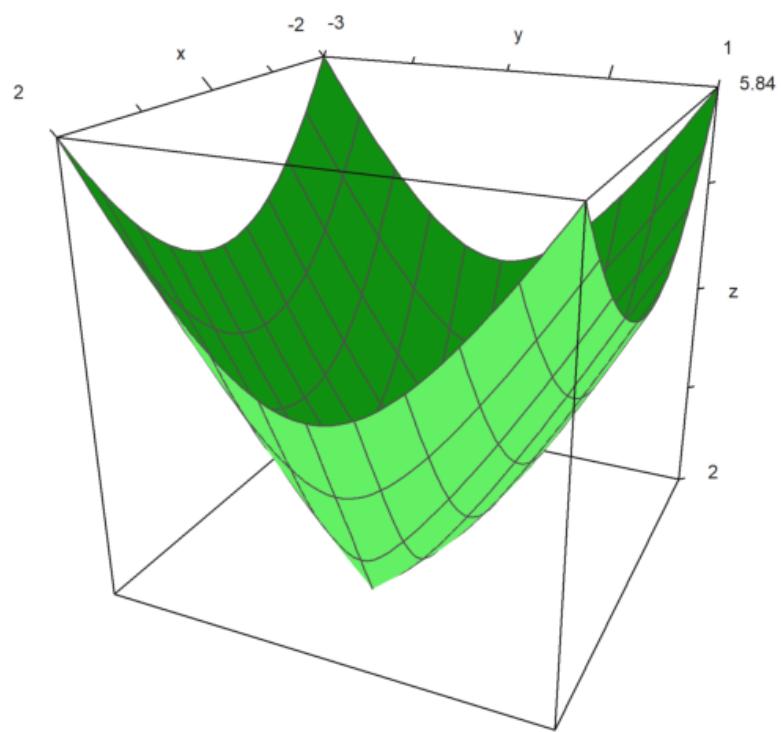
Sekarang kita melihat fungsi MA+MB di mana A dan B adalah dua titik (tetap). Ini adalah "fakta yang sudah umum diketahui" bahwa kurva levelnya adalah ellips, dengan fokusnya adalah A dan B; kecuali untuk minimum AB yang konstan di segmen [AB]:

```
>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):
```



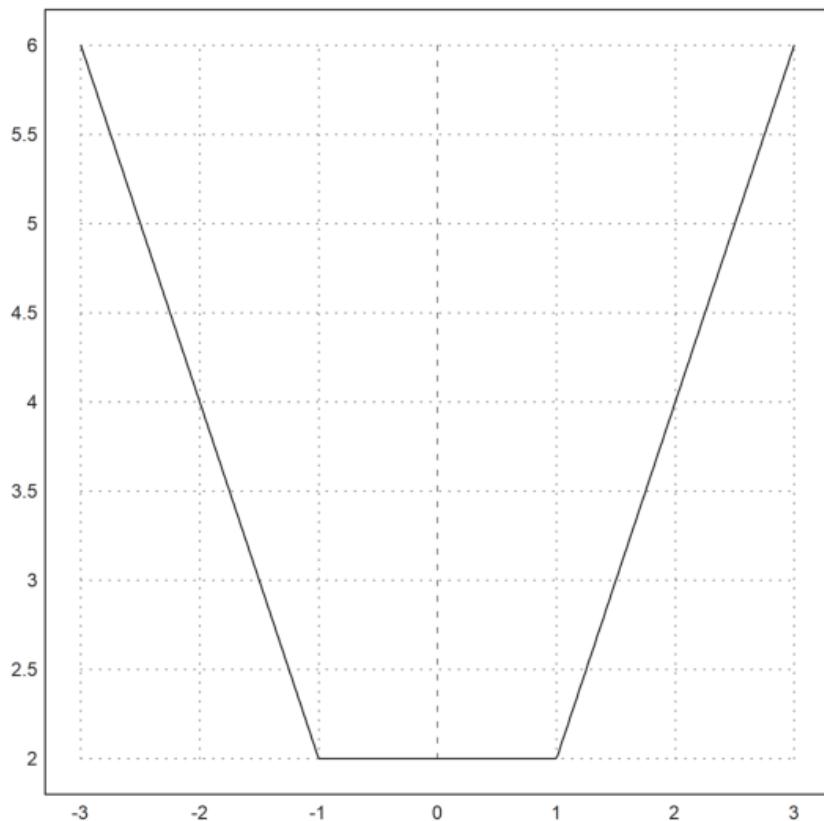
Grafiknya lebih menarik:

```
>plot3d("d2", xmin=-2, xmax=2, ymin=-3, ymax=1):
```



Pembatasan pada garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



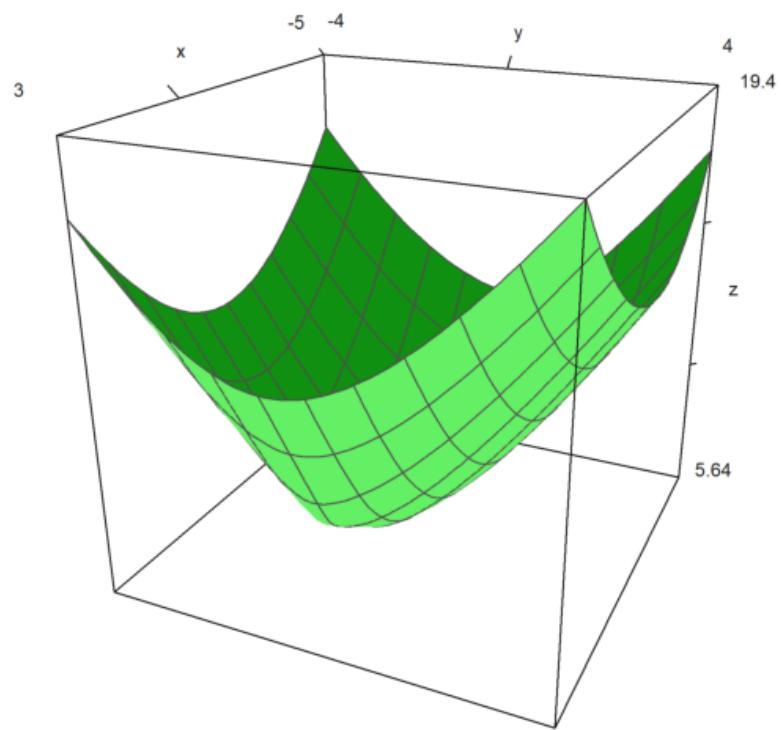
Tiga Titik

Sekarang hal-hal menjadi sedikit kurang sederhana: Kurang diketahui bahwa $MA+MB+MC$ mencapai nilai minimumnya di satu titik pada bidang, tetapi untuk menentukannya tidaklah mudah:

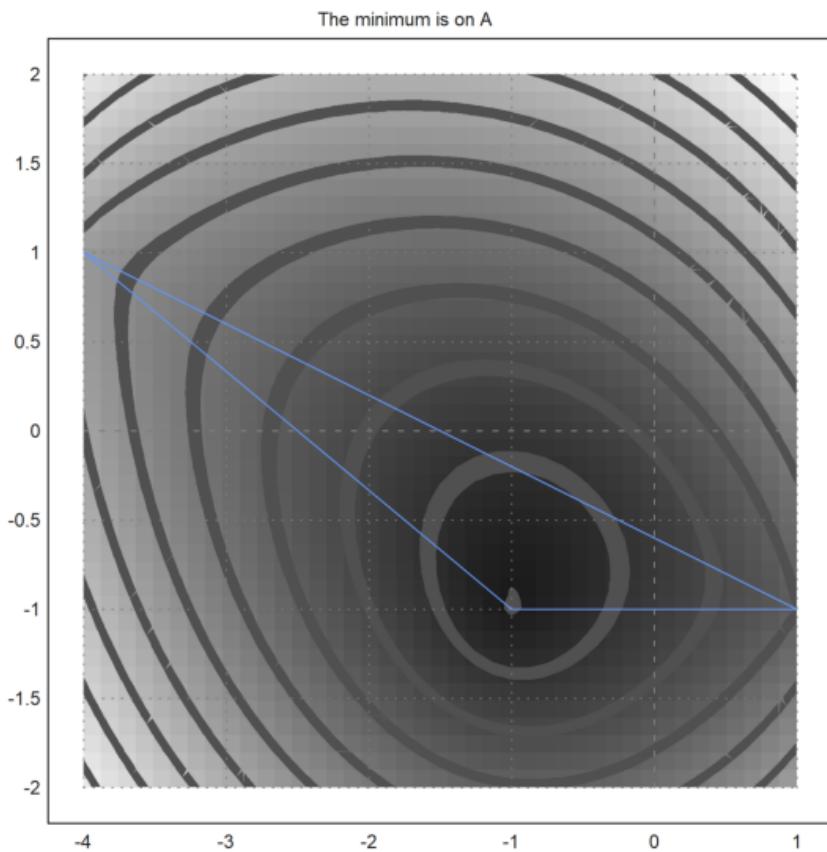
- 1) Jika salah satu sudut segitiga ABC lebih dari 120° (misalnya sudut di titik A), maka nilai minimum dicapai pada titik ini (misalnya $AB+AC$).

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);
>insimsg;
```

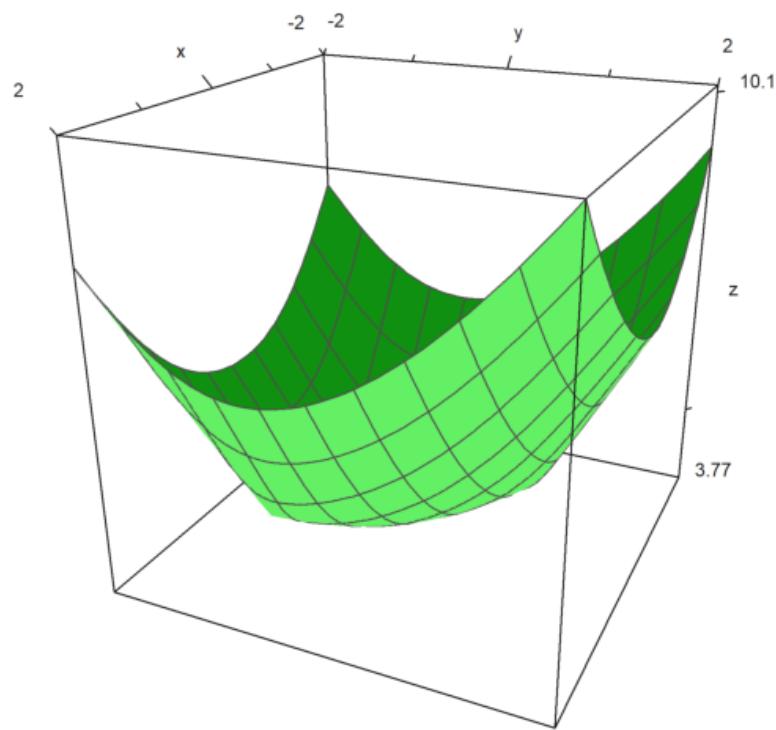


```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

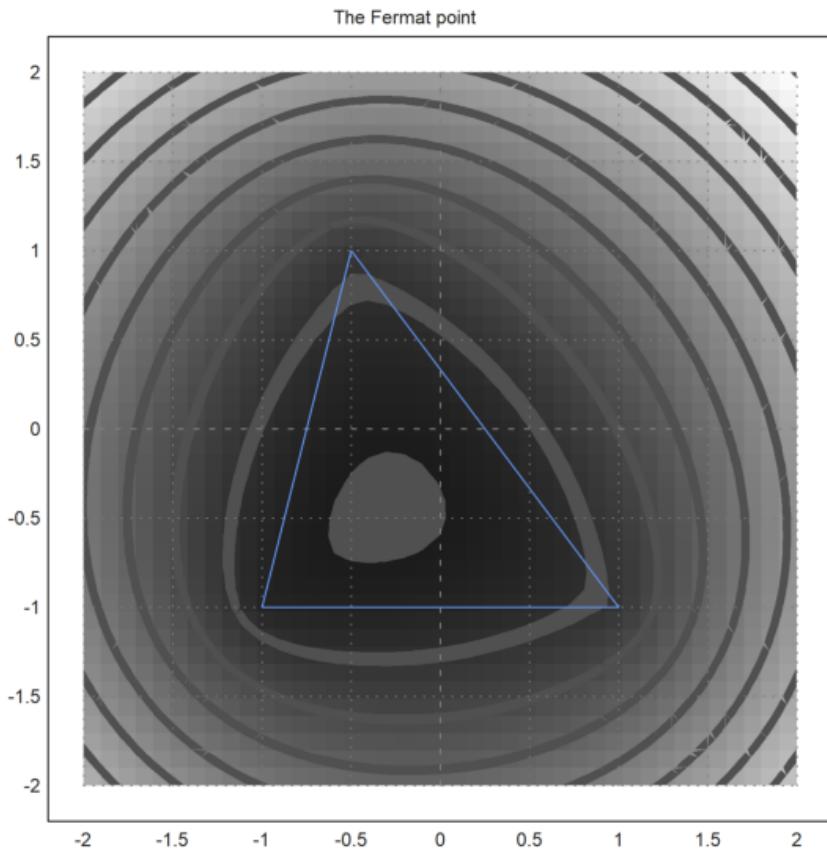


- 2) Namun, jika semua sudut segitiga ABC lebih kecil dari 120° , nilai minimumnya terletak pada titik F di dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (yaitu 120° masing-masing):

```
>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;
```



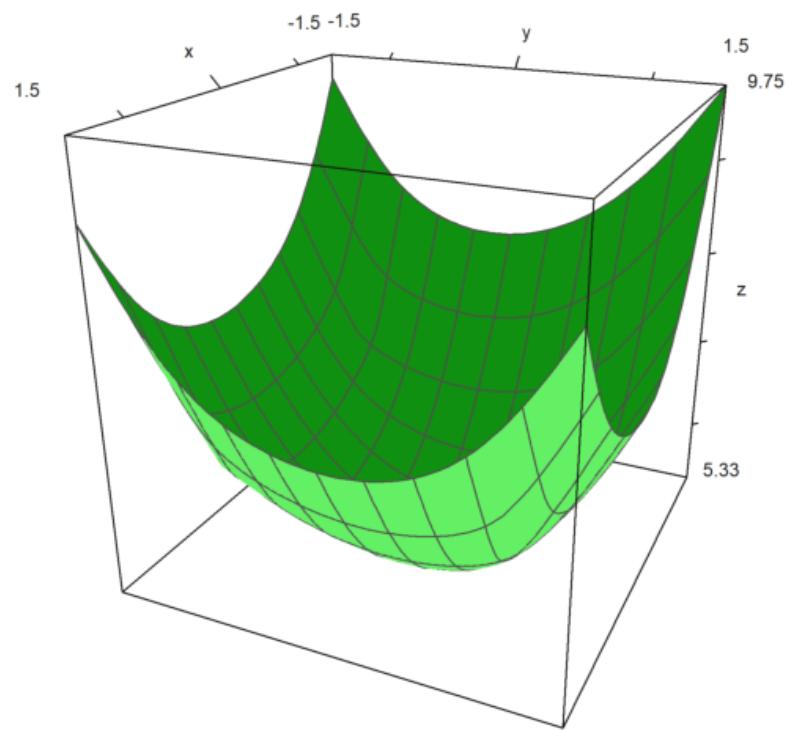
Ini adalah kegiatan menarik untuk menggambarkan gambar di atas dengan perangkat lunak geometri; sebagai contoh, saya tahu sebuah perangkat lunak yang ditulis dalam bahasa Java yang memiliki instruksi "garis kontur"...

Semua yang disebutkan di atas telah ditemukan oleh seorang hakim Prancis bernama Pierre de Fermat; ia menulis surat kepada dilettante lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di departemen pajak penghasilan. Oleh karena itu, titik unik F sedemikian rupa sehingga $FA+FB+FC$ minimal disebut titik Fermat dari segitiga. Namun, tampaknya beberapa tahun sebelumnya, seorang Italia bernama Torricelli telah menemukan titik ini sebelum Fermat melakukannya! Bagaimanapun, tradisinya adalah menandai titik ini sebagai F...

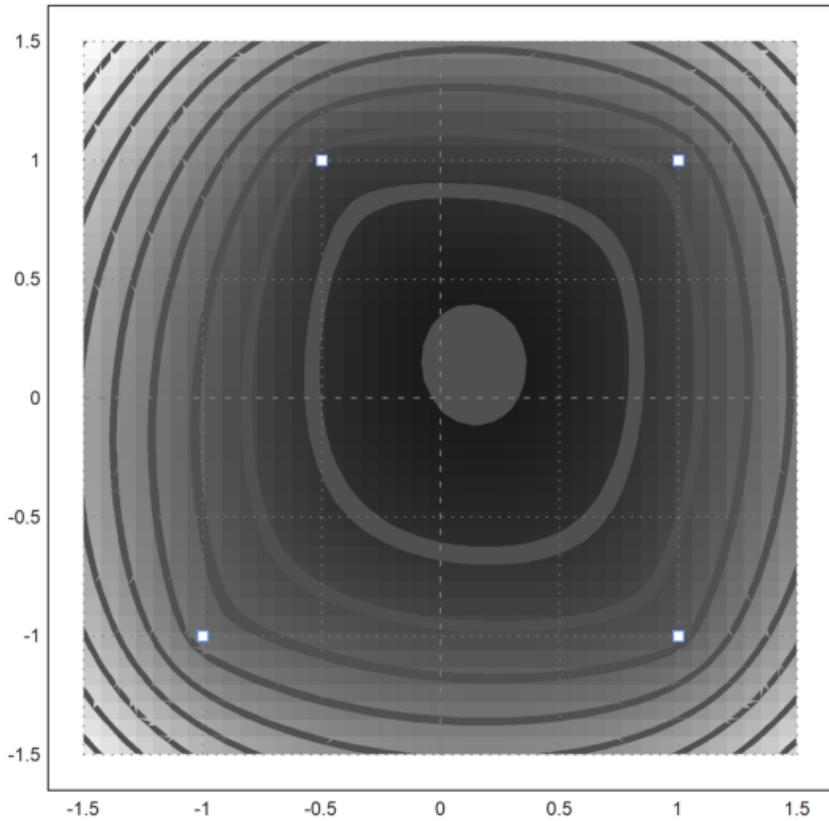
Empat Titik

Langkah berikutnya adalah menambahkan titik keempat D dan mencoba meminimalkan $MA+MB+MC+MD$; misalkan Anda adalah operator TV kabel dan ingin menemukan di mana Anda harus menempatkan antena Anda agar Anda dapat menyediakan layanan ke 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;
```



Tetapi ada nilai minimum dan itu tidak terletak di salah satu titik sudut A, B, C, atau D:

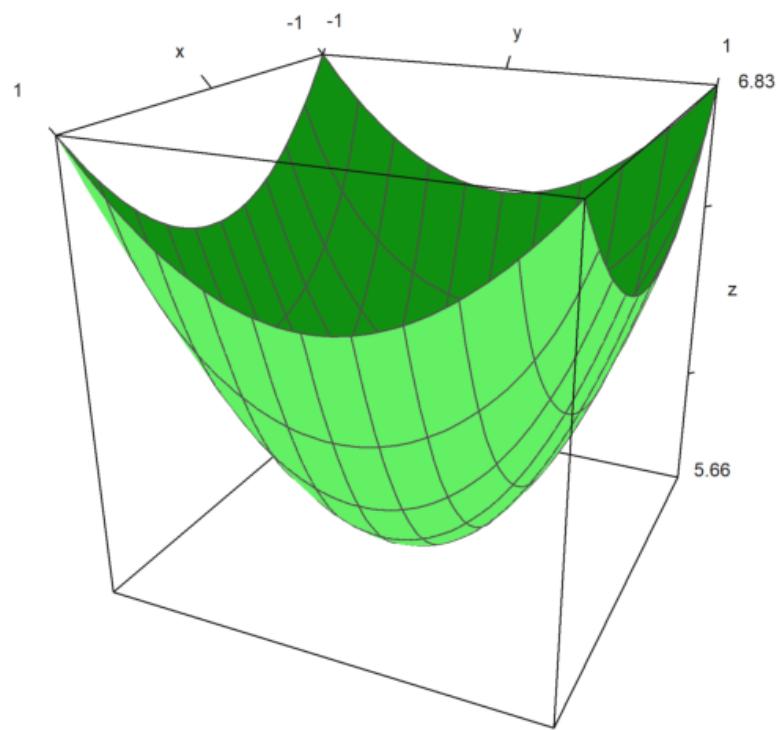
```
>function f(x):=d4(x[1],x[2])
>neldermin("f", [0.2,0.2])
```

```
[0.142858, 0.142857]
```

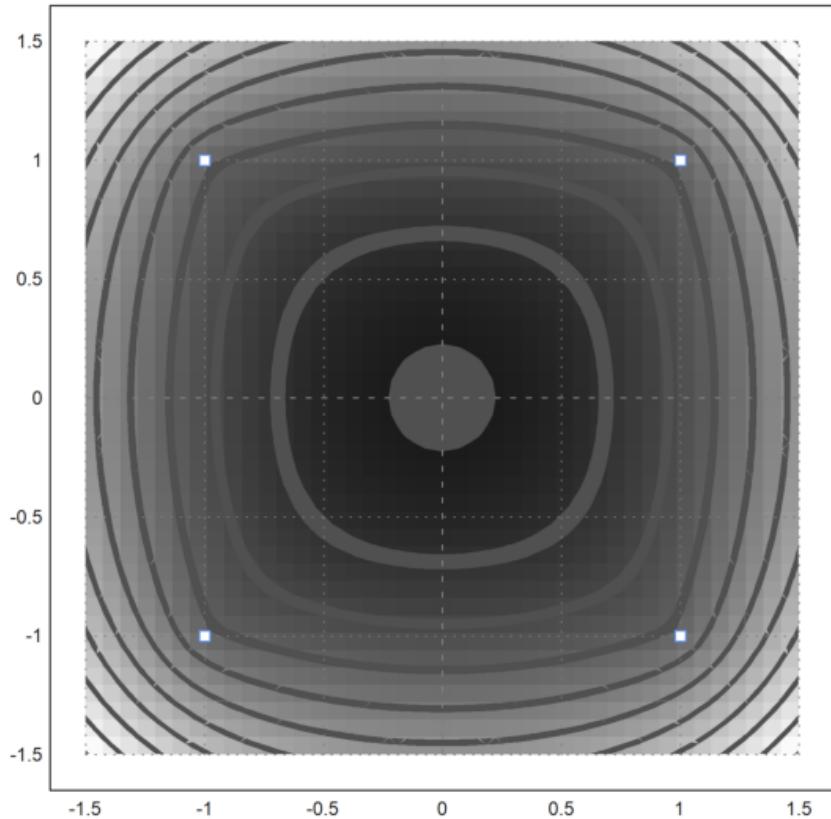
Sepertinya dalam kasus ini, koordinat titik optimalnya adalah rasional atau mendekati rasional...

Sekarang ABCD adalah sebuah persegi dan kami mengharapkan bahwa titik optimal akan berada di tengah-tengah 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 memiliki Povray terinstal, dan pvenGINE.exe berada dalam jalur program.

Pertama, kita menghitung jari-jari bola.

Jika Anda melihat gambar di bawah ini, Anda akan melihat bahwa kita memerlukan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file geometry.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 ]
```

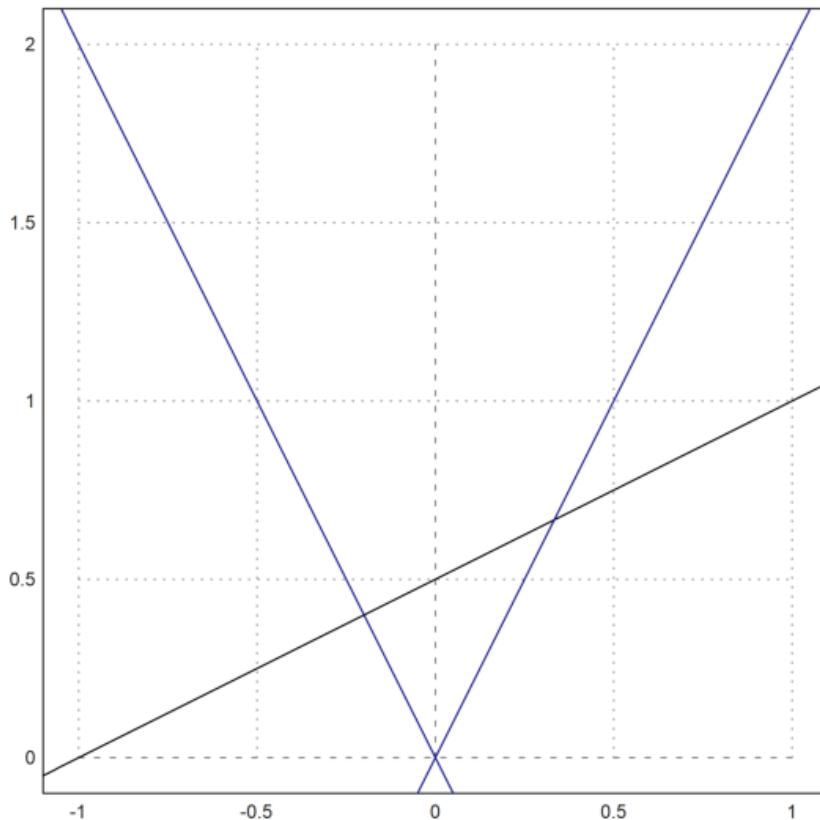
Kemudian, garis ketiga.

```
>g &= lineThrough([-1,0],[1,1])
```

```
[ - 1, 2, 1 ]
```

Kami menggambar semua yang telah kita selesaikan.

```
>setPlotRange(-1,1,0,2);  
>color(black); plotLine(g(), "")  
>a:=2; color(blue); plotLine(g1(), ""), plotLine(g2(), ""):
```



Sekarang kita mengambil titik umum pada sumbu y.

```
>P &= [0,u]
```

```
[ 0, u ]
```

Menghitung jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

$$\sqrt{\left(\frac{a^2 u}{a^2 + 1} - u\right)^2 + \frac{a^2 u^2}{(a^2 + 1)^2}}$$

Menghitung jarak ke g.

```
>d &= distance(P,projectToLine(P,g)); $d
```

$$\sqrt{\left(\frac{u + 2}{5} - u\right)^2 + \frac{(2 u - 1)^2}{25}}$$

Dan menemukan pusat dari dua lingkaran, di mana 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-solusi simbolis dan menemukan kedua pusat dan kedua jarak.

```
>u := sol()
```

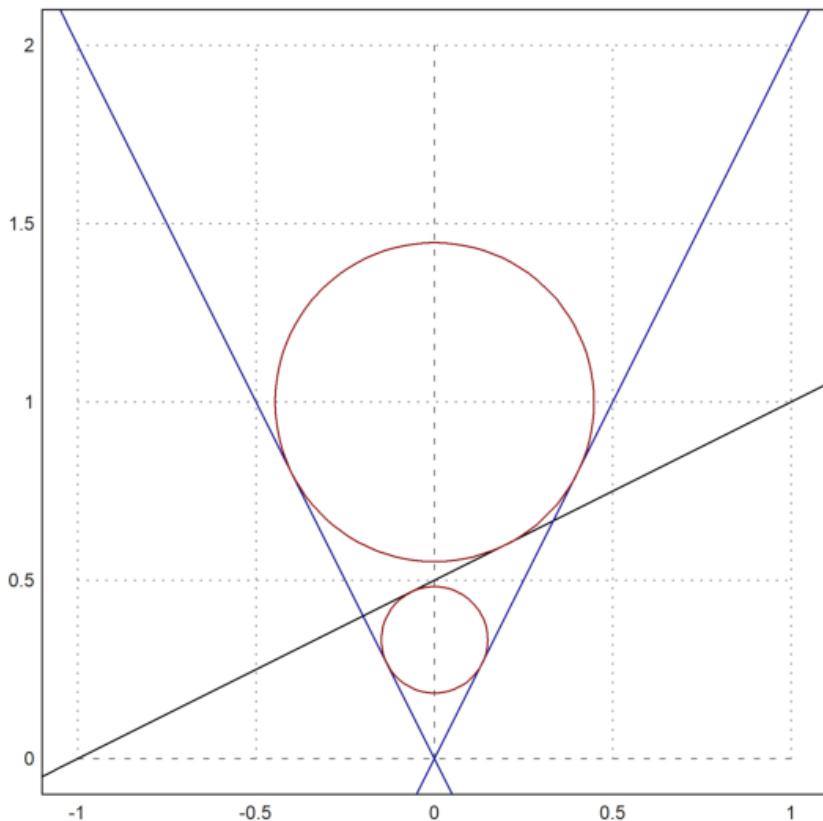
```
[0.333333, 1]
```

```
>dd := d()
```

```
[0.149071, 0.447214]
```

Gambar lingkaran-lingkaran ke dalam gambar tersebut.

```
>color(red);
>plotCircle(circleWithCenter([0,u[1]],dd[1]),"");
>plotCircle(circleWithCenter([0,u[2]],dd[2]),"");
>insimg;
```



Plot dengan Povray

Selanjutnya kita akan membuat plot dengan Povray. Perhatikan bahwa Anda dapat mengubah perintah mana pun dalam urutan perintah Povray berikut ini, dan jalankan semua perintah dengan menekan Shift-Return. Pertama, kita muatkan fungsi-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

Kita atur suasana dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Selanjutnya, kita tulis dua bola ke file Povray.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

Dan kerucut, yang transparan.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

Kita hasilkan sebuah bidang yang 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));
```

Kemudian, kita hasilkan dua titik pada lingkaran, di mana bola menyentuh kerucut.

```
>function turnz(v) := return [-v[2],v[1],v[3]]  
>P1=projectToLine([0,u[1]],g()); P1=turnz([P1[1],0,P1[2]]);  
>writeln(povpoint(P1,povlook(yellow)));  
>P2=projectToLine([0,u[2]],g()); P2=turnz([P2[1],0,P2[2]]);  
>writeln(povpoint(P2,povlook(yellow)));
```

Kemudian, kita hasilkan 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 menghitung 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)));
```

Kita menghubungkan titik-titik dengan segmen garis.

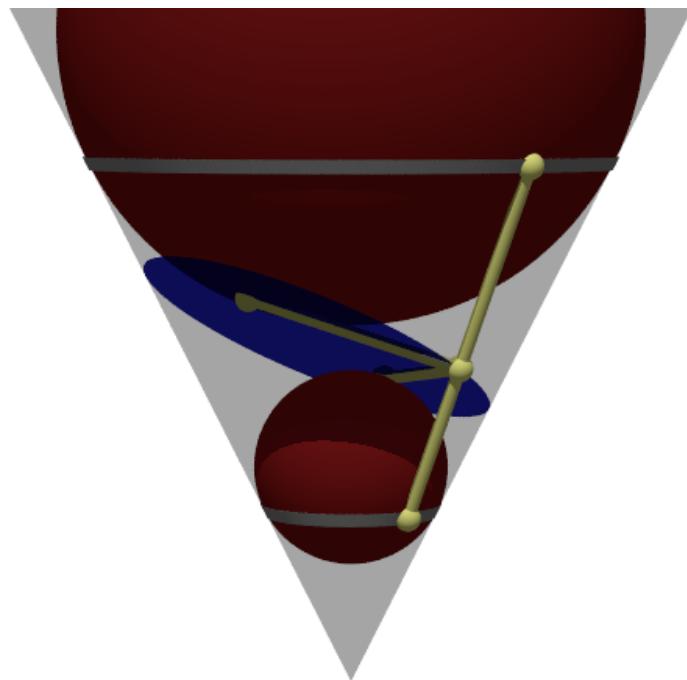
```
>writeln(povsegment(P1,P2,povlook(yellow)));  
>writeln(povsegment(P5,P3,povlook(yellow)));  
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Selanjutnya, kita hasilkan 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]-defaultpointsiz/2],[0,0,P1[3]+defaultpointsiz/2],1);  
>writeln(povintersection([pcw,pc1],povlook(gray)));  
>pc2=povcylinder([0,0,P2[3]-defaultpointsiz/2],[0,0,P2[3]+defaultpointsiz/2],1);  
>writeln(povintersection([pcw,pc2],povlook(gray)));
```

Mulai program Povray.

```
>povend();
```



Untuk mendapatkan Anaglyph dari ini, kita perlu meletakkan semuanya ke dalam fungsi `scene`. Fungsi ini akan digunakan dua kali nanti.

```
>function scene () ...
```

```
global a,u,dd,g,g1,defaultpointszie;
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,dp[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))');
```

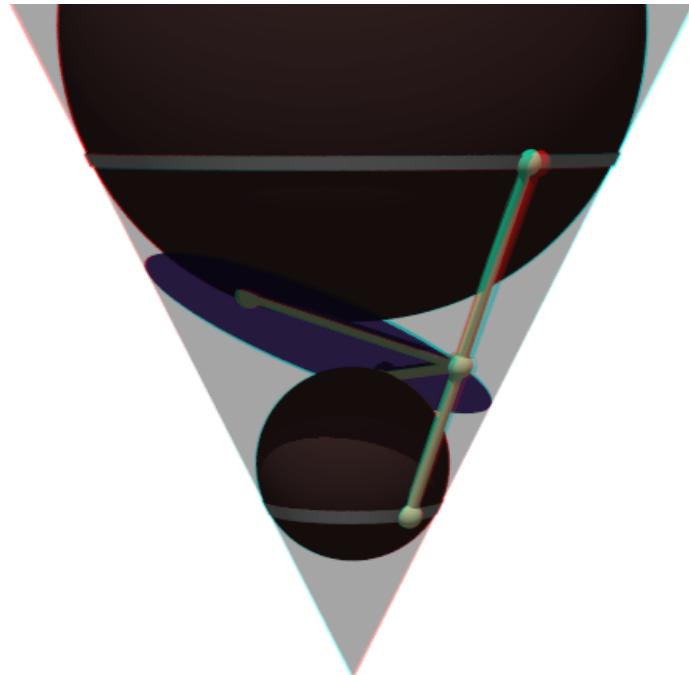
```

pcw=povcone([0,0,0],0,[0,0,a],1.01);
pcl=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pcl],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction

```

Anda memerlukan kacamata merah/cyan untuk mengapresiasi efek berikutnya.

```
>povanaglyph("scene", zoom=11, center=[0,0,0.5], height=10°, angle=140°);
```



Contoh 8: Geometri Bumi

Dalam notebook ini, kita ingin melakukan beberapa perhitungan bola. Fungsi-fungsi ini terdapat dalam file "spherical.e" di folder contoh. Kami perlu memuat file tersebut 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 adalah koordinat untuk Kampus FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569, 1.92657]
```

Anda dapat mencetak posisi ini dengan menggunakan sposprint (cetak posisi bola).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

S $7^{\circ}46.467'$ E $110^{\circ}23.050'$

Mari tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,24.533)];  
>sposprint(Solo), sposprint(Semarang),
```

S $7^{\circ}34.333'$ E $110^{\circ}49.683'$
S $6^{\circ}59.050'$ E $110^{\circ}24.533'$

Pertama-tama kita menghitung vektor dari satu kota ke kota lainnya pada bola ideal. Vektor ini adalah [heading, jarak] dalam radian. Untuk menghitung jarak di Bumi, kita mengalikannya dengan jari-jari Bumi pada lintang 7° .

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth( $7^{\circ}$ )->km // perkiraan jarak FMIPA-Solo
```

$65^{\circ}20'26.60''$
53.8945384608

Ini adalah perkiraan yang baik. Rutinitas-rutinitas berikut menggunakan perkiraan yang lebih baik lagi. Pada jarak yang begitu pendek, hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->"km"
```

88.0114026318km

Ada fungsi untuk heading, yang memperhitungkan bentuk elips Bumi. Sekali lagi, kami mencetaknya dengan cara yang lebih canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

65.34°

Sudut dari segitiga melebihi 180° pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,Semarang,Solo);
```

$180^{\circ}0'10.77''$

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga yang kecil, ini tidak akurat karena kesalahan pengurangan dalam asum- π .

```
>(asum-pi)*rearth( $48^{\circ}$ )^2->" km^2"
```

2116.02948749 km 2

Ada fungsi untuk ini, yang menggunakan lintang rata-rata segitiga untuk menghitung jari-jari Bumi, dan mengatasi kesalahan pembulatan untuk segitiga yang sangat kecil.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi esarea()
```

2123.64310526 km²

Kita juga bisa menambahkan vektor ke posisi. Sebuah vektor berisi heading dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kita menggunakan svector. Untuk menambahkan vektor ke posisi, kita menggunakan saddvector.

```
>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 ideal. Sama dengan di Bumi.

```
>sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(Solo),
```

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 perkiraan yang baik.

```
>esdist(Tugu,Monas)->"km"
```

431.565659488km

Heading-nya sama dengan 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 asli. Ini terjadi karena kita tidak menghitung fungsi inversnya dengan tepat, tetapi mengambil perkiraan jari-jari Bumi sepanjang jalur.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

S 6°10.500' E 106°48.717'

Kesalahan ini tidak besar, bagaimanapun.

```
>sposprint(Monas),
```

S 6°10.500' E 106°48.717'

Tentu saja, kita tidak dapat berlayar dengan heading yang sama dari satu tujuan ke tujuan lain, jika kita ingin mengambil jalur terpendek. Bayangkan, Anda terbang ke timur laut dari titik mana pun di Bumi. Kemudian Anda akan berputar-putar ke kutub utara. Garis besar tidak mengikuti heading konstan!

Perhitungan berikut menunjukkan bahwa kita jauh dari tujuan yang benar, jika kita menggunakan heading yang sama selama perjalanan.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali satu persepuluhan jarak, menggunakan heading ke Monas yang kita dapatkan di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya jauh dari benar.

```
>sposprint(p), skmpprint(esdist(p,Monas))
```

S 6°11.250' E 106°48.372'
1.529km

Sebagai contoh lain, mari kita ambil dua titik di Bumi pada lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

alur terpendek dari P1 ke P2 bukanlah lingkaran lintang 30°, tetapi jalur yang lebih pendek dimulai 10° lebih ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

79.69°

Tetapi, jika kita mengikuti bacaan kompas ini, kita akan berputar-putar ke kutub utara! Jadi kita harus menyesuaikan heading kita sepanjang jalan. Untuk tujuan kasar, kita menyesuaikannya setiap 1/10 dari jarak total.

```
>p=P1; dist=esdist(P1,P2); ...
> loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
79.69°
81.67°
83.71°
85.78°
87.89°
90.00°
92.12°
94.22°
96.29°
98.33°
```

Jaraknya tidak benar, karena kita akan menambahkan sedikit kesalahan jika kita mengikuti heading yang sama terlalu lama.

```
>skmpprint(esdist(p,P2))
```

```
0.203km
```

Kita mendapatkan perkiraan yang baik, jika kita menyesuaikan heading kita setelah setiap 1/100 dari jarak total dari Tugu ke Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...
> loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;
>skmpprint(esdist(p,Monas))
```

```
0.000km
```

Untuk tujuan navigasi, kita bisa mendapatkan rangkaian posisi GPS sepanjang garis besar ke Monas dengan fungsi `navigate`.

```
>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'
```

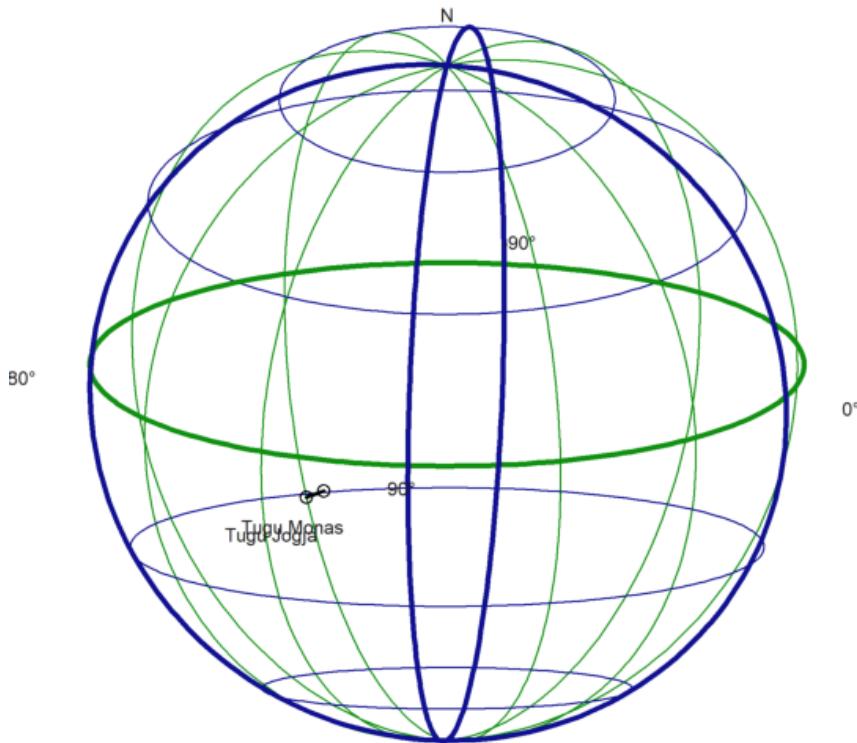
Kita menulis sebuah fungsi yang memplotkan Bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
```

```
useglobal;
plotearth;
plotpos(Tugu, "Tugu Jogja"); plotpos(Monas, "Tugu Monas");
plotposline(v);
endfunction
```

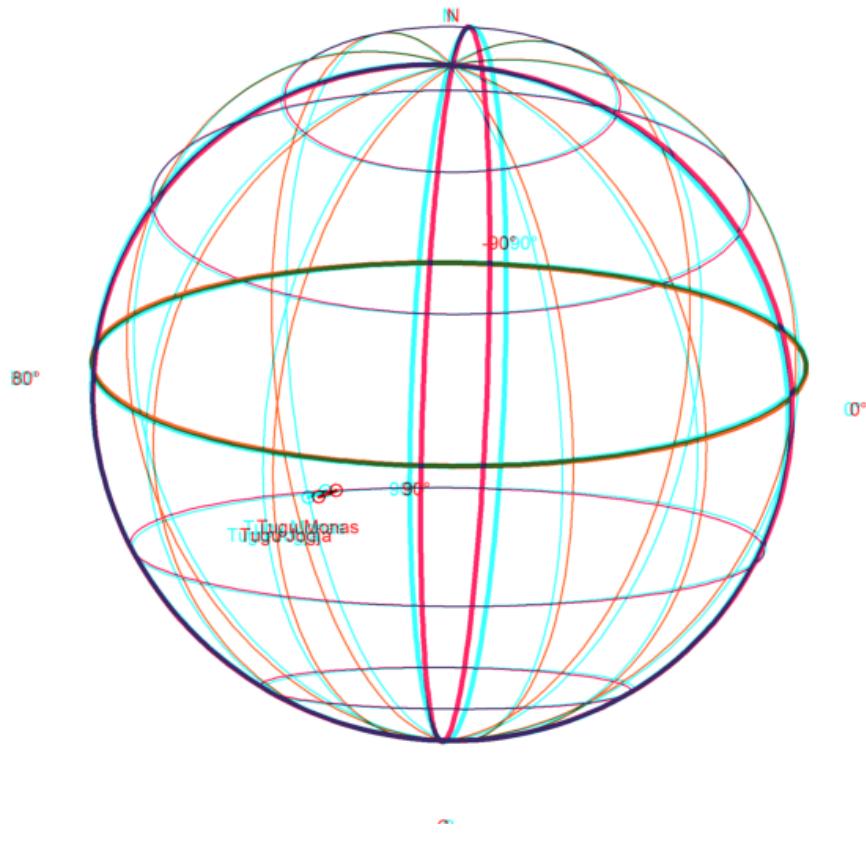
Sekarang plot semuanya.

```
>plot3d("testplot", angle=25, height=6, >own, >user, zoom=4) :
```



Atau gunakan plot3d untuk mendapatkan tampilan anaglyph. Ini terlihat sangat bagus dengan kacamata merah/cyan.

```
>plot3d("testplot", angle=25, height=6, distance=5, own=1, anaglyph=1, zoom=4) :
```



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.

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.

```
>A &=[3, 2]; B &=[1, -1]; C &=[4, 2];
>r&=(lineThrough(A, B)); $r
```

$$[3, -2, 5]$$

```
>$getLineEquation(r,x,y); $solve(% ,y)
```

$$\left[y = \frac{3x - 5}{2} \right]$$

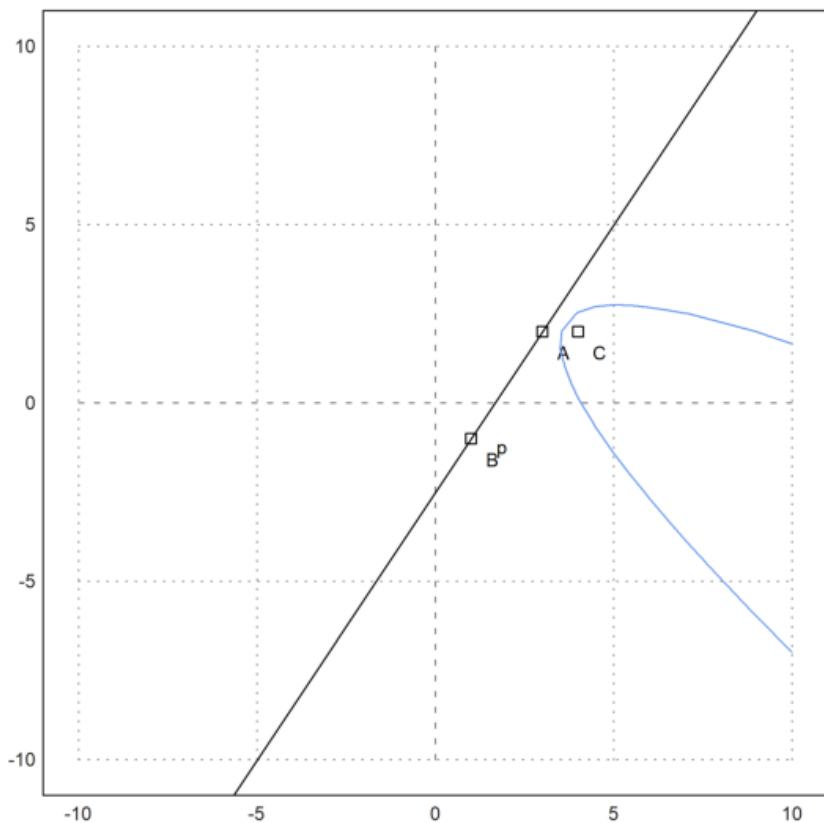
```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{-2y + 3x - 5}{\sqrt{13}} - \sqrt{(2-y)^2 + (4-x)^2} = 0$$

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y); $akar
```

$$\left[y = \frac{-\sqrt{13}\sqrt{2x-7} - 2x + 12}{3}, y = \frac{\sqrt{13}\sqrt{2x-7} - 2x + 12}{3} \right]$$

```
>A=[3,2]; B=[1,-1]; C=[4,2];
>setPlotRange (-10,10,-10,10); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
>plotLine(lineThrough(A,B));
>plot2d(p,level=0,add=1,contourcolor=12):
```



```
>reset();
```

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-sisinya 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.

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).

Penyelesaian

Titik Fokus

$$F_1 = (-c, 0) = (-1, 0)$$

$$F_2 = (c, 0) = (7, 0)$$

Melalui titik $(0, 12/5)$

```
>$load(draw):  
>F1 &=[-1,0]; F2 &=[7,0]; C &=[0, 12/5];  
>d1 &= sqrt((0+(-1))^2+(12/5)^2); $d1 // d1 = jarak F1 ke titik (0,12/5)
```

$$\frac{13}{5}$$

```
>d2 &= sqrt((0-7)^2+(12/5)^2); $d2 // d2 = jarak F2 ke titik (0,12/5)
```

$$\frac{37}{5}$$

```
>a &= (d1+d2)/2 // menentukan nilai a
```

5

```
>xp &= (-1+7)/2 // titik pusat ellips xp
```

3

```
>yp &= (0-0)/2 // titik pusat ellips yp
```

0

Titik pusat ellips (xp, yp) = (3, 0)

```
>c &= (-1+7)/2 // jarak fokus ke pusat
```

3

```
>b &= sqrt(a^2-c^2) // menentukan nilai b
```

4

```
>$ (x-xp)^2/a^2+(y-yp)^2/b^2=1 // persamaan ellips
```

$$\frac{y^2}{16} + \frac{(x-3)^2}{25} = 1$$

```
>$draw2d(ellipse(3, 0, 5, 4, 360, -360):
>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).

```
>A &=[-1,-1]; B &=[2,0]; C &=[1,2];
>r&=(lineThrough(A,B)); $r
```

$$[-1, 3, -2]$$

```
>$getLineEquation(r,x,y); $solve(% ,y)
```

$$\left[y = \frac{x - 2}{3} \right]$$

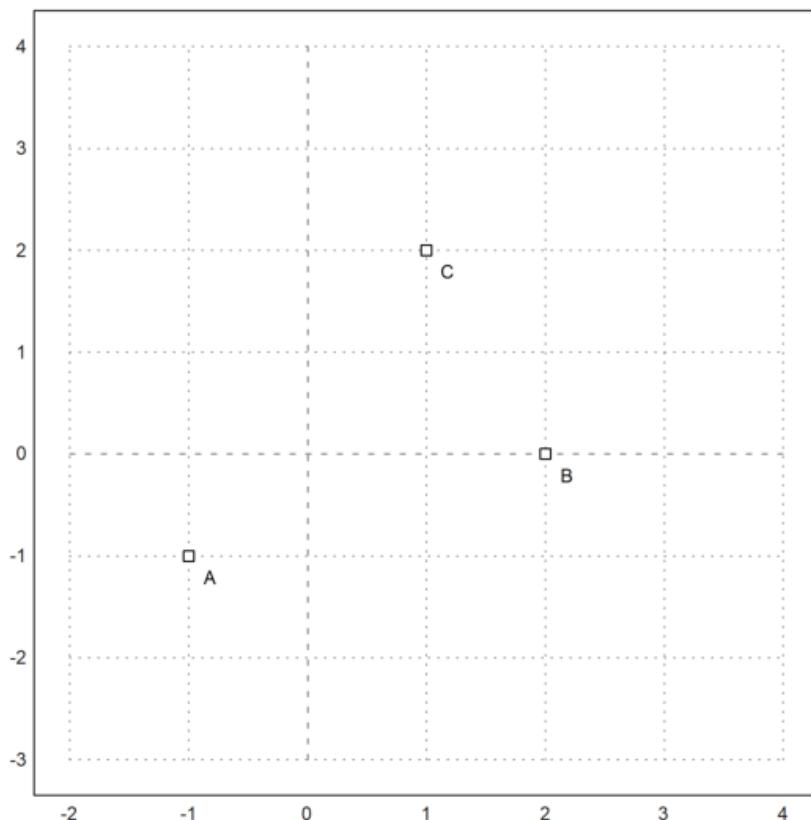
```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3y - x + 2}{\sqrt{10}} - \sqrt{(2-y)^2 + (1-x)^2} = 0$$

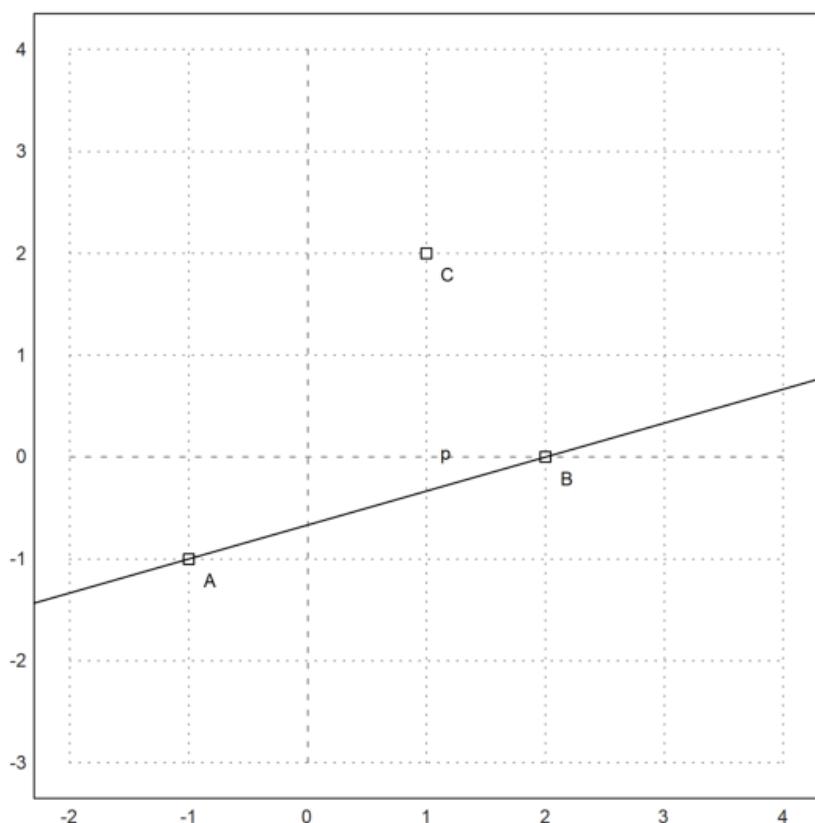
```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y); $akar
```

$$\left[y = -3x - \sqrt{70}\sqrt{9-2x} + 26, y = -3x + \sqrt{70}\sqrt{9-2x} + 26 \right]$$

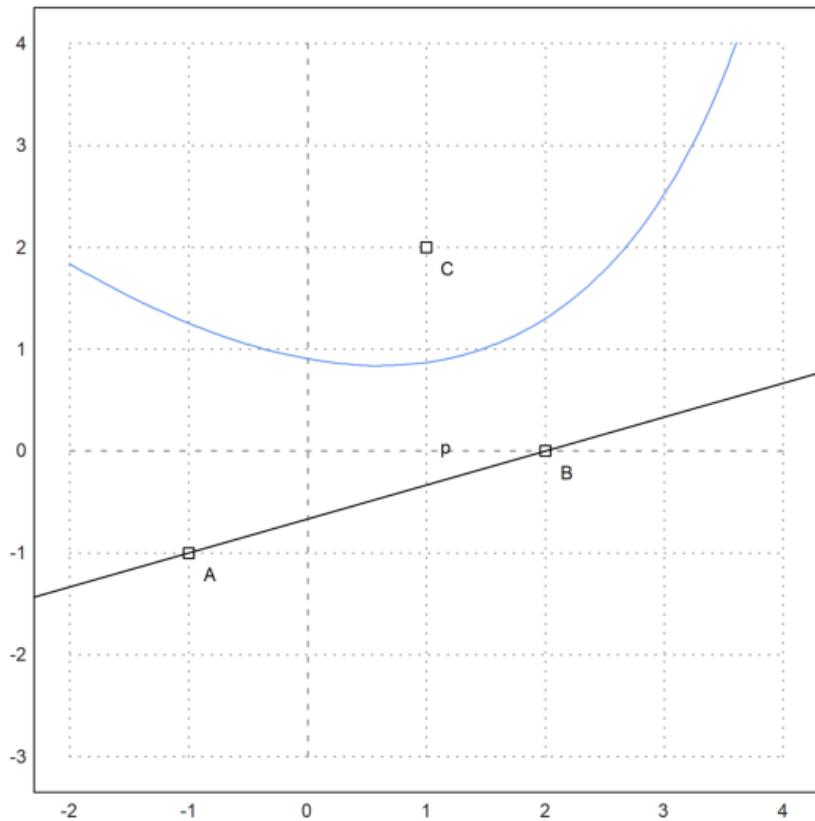
```
>A=[-1,-1]; B=[2,0]; C=[1,2];
>setPlotRange (-2,4,-3,4); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```



```
>plotLine(lineThrough(A,B)):
```



```
>plot2d(p,level=0,add=1,contourcolor=12):
```



```
>reset();
```

BAB 7

PEKAN 13-14: MENGGUNAKAN EMT UNTUK STATISTIKA

article

eumat

EMT untuk Statistika

Di dalam notebook ini, kita menampilkan plot, uji statistik utama, dan distribusi di Euler.

Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar untuk statistik. Jadi, mungkin Anda memerlukan beberapa latar belakang untuk memahami detailnya.

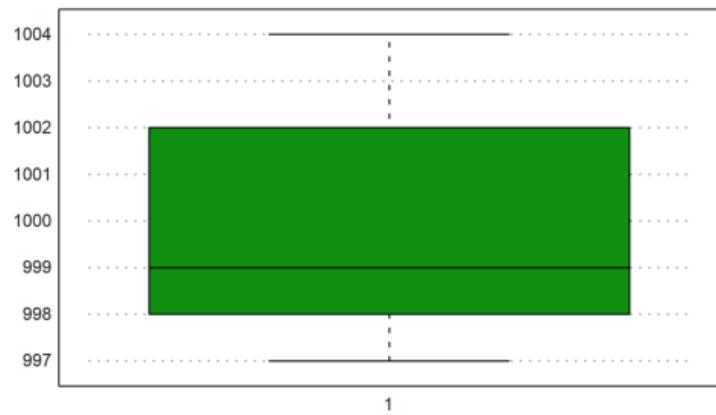
Anggaplah pengukuran berikut. Kita ingin menghitung nilai rata-rata dan standar deviasi yang diukur.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...  
>median(M), mean(M), dev(M),
```

```
999  
999.9  
2.72641400622
```

Kita dapat membuat plot box-and-whiskers untuk data tersebut. Dalam kasus kita, tidak ada outlier.

```
>aspect(1.75); boxplot(M):
```



Kita akan menghitung probabilitas bahwa sebuah nilai lebih besar dari 1005, dengan mengasumsikan nilai yang diukur berasal dari distribusi normal.

Semua fungsi untuk distribusi di Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

$$\text{normaldis}(x, m, d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} dt.$$

Kita mencetak hasilnya dalam % dengan akurasi 2 digit menggunakan fungsi print.

```
>print((1-normaldis(1005, mean(M), dev(M)))*100, 2, unit=" %")
```

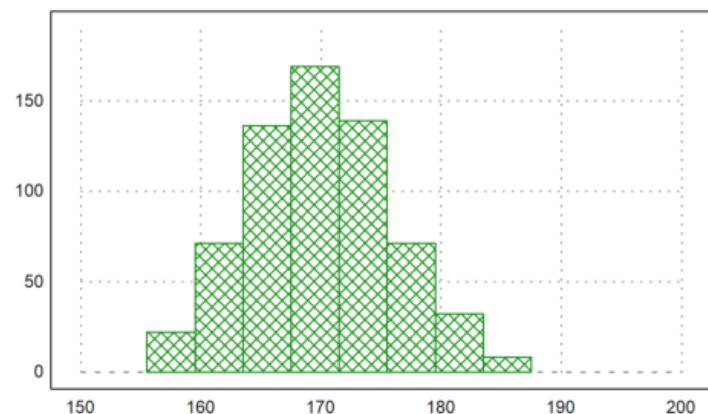
3.07 %

Untuk contoh selanjutnya, kita mengasumsikan jumlah pria dalam rentang ukuran tertentu.

```
>r=155.5:4:187.5; v=[22, 71, 136, 169, 139, 71, 32, 8];
```

Berikut adalah plot dari distribusinya.

```
>plot2d(r, v, a=150, b=200, c=0, d=190, bar=1, style="/") :
```



Kita bisa memasukkan data mentah seperti ini ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita seharusnya berisi tiga kolom: Awal rentang, akhir rentang, jumlah pria dalam rentang tersebut.

Tabel bisa dicetak dengan judul. Kita menggunakan vektor string untuk mengatur judul.

```
>T:=r[1:8]' | r[2:9]' | v'; writetable(T, labc = ["BB", "BA", "Frek"])
```

BB	BA	Frek
155.5	159.5	22
159.5	163.5	71
163.5	167.5	136
167.5	171.5	169
171.5	175.5	139
175.5	179.5	71
179.5	183.5	32
183.5	187.5	8

Jika kita memerlukan nilai rata-rata dan statistik lainnya dari ukuran, kita perlu menghitung titik tengah dari rentang tersebut. Kita dapat menggunakan dua kolom pertama dari tabel kita untuk ini.

Simbol "|" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan opsi "labc" untuk menentukan judul kolom.

```
>(T[,1]+T[,2])/2 // the midpoint of each interval
```

```
157.5  
161.5  
165.5  
169.5  
173.5  
177.5  
181.5  
185.5
```

Tetapi lebih mudah untuk menggabungkan rentang dengan vektor [1/2,1/2].

```
>M=fold(r, [0.5, 0.5])
```

```
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
```

Sekarang kita dapat menghitung rata-rata dan deviasi dari sampel dengan frekuensi yang diberikan.

```
>{m, d}=meandev(M, v); m, d,
```

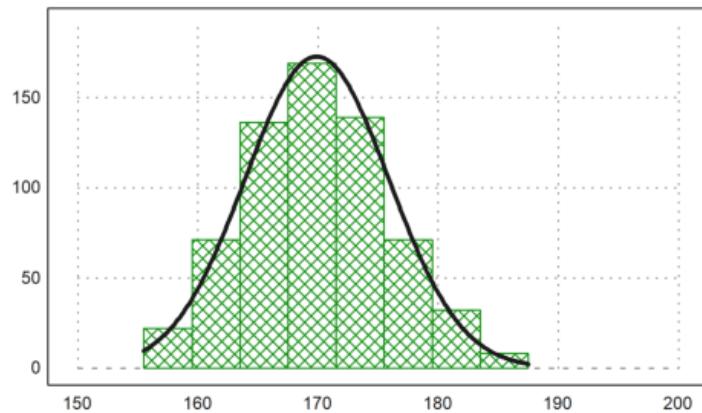
```
169.901234568  
5.98912964449
```

Mari kita tambahkan distribusi normal dari nilai ke plot batang di atas. Rumus untuk distribusi normal dengan mean m dan standar deviasi d adalah:

$$y = \frac{1}{d\sqrt{2\pi}} e^{\frac{-(x-m)^2}{2d^2}}.$$

Karena nilainya antara 0 dan 1, untuk menggambarkannya pada plot batang, nilainya harus dikalikan dengan 4 kali total jumlah data.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...
>  xmin=min(r),xmax=max(r),thickness=3,add=1):
```



Tabel

Di dalam direktori notebook ini, Anda akan menemukan sebuah file dengan sebuah tabel. Data ini mewakili hasil dari sebuah survei. Berikut adalah empat baris pertama dari file tersebut. Data ini berasal dari buku online Jerman "Einführung in die Statistik mit R" oleh A. Handl.

```
>printfile("table.dat", 4);
```

```
Person Sex Age Titanic Evaluation Tip Problem
1 m 30 n . 1.80 n
2 f 23 y g 1.80 n
3 f 26 y g 1.80 y
```

Tabel ini terdiri dari 7 kolom angka atau token (string). Kami ingin membaca tabel dari file tersebut. Pertama, kami akan menggunakan terjemahan kami sendiri untuk token-token ini.

Untuk ini, kami akan mendefinisikan set token-token. Fungsi strtokens() akan menerima vektor string token dari string yang diberikan.

```
>mf:=[ "m", "f" ]; yn:=[ "y", "n" ]; ev:=strtokens("g vg m b vb");
```

Sekarang kita akan membaca tabel dengan terjemahan ini.

Argumen tok2, tok4, dsb. adalah terjemahan dari kolom-kolom tabel. Argumen-argumen ini tidak ada dalam parameter daftar fungsi readtable(), jadi Anda perlu menyediakannya dengan ":=".

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
>load over statistics;
```

Untuk mencetaknya, kami perlu menentukan set token yang sama. Kami hanya akan mencetak empat baris pertama.

```
>writetable(MT[1:10], labc=hd, wc=5, tok2:=mf, tok4:=yn, tok5:=ev, tok7:=yn);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n

Tanda titik "." mewakili nilai yang tidak tersedia.

Jika kita tidak ingin menentukan token-token untuk terjemahan sebelumnya, kita hanya perlu menentukan kolom-kolom yang berisi token dan bukan angka.

```
>ctok=[2, 4, 5, 7]; {MT, hd, tok}=readtable("table.dat", ctok=ctok);
```

Fungsi readtable() sekarang akan mengembalikan set token.

```
>tok
```

```
m  
n  
f  
y  
g  
vg
```

Tabel ini berisi entri dari file dengan token diterjemahkan menjadi angka.

String khusus NA="." diinterpretasikan sebagai "Tidak Tersedia", dan menjadi NAN (bukan angka) dalam tabel. Terjemahan ini dapat diubah dengan parameter NA dan NAvl.

```
>MT[1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Berikut adalah isi dari tabel dengan angka yang belum diterjemahkan.

```
>writetable(MT, wc=5)
```

1	1	30	2	.	1.8	2
2	3	23	4	5	1.8	2
3	3	26	4	5	1.8	4
4	1	33	2	.	2.8	2
5	1	37	2	.	1.8	2
6	1	28	4	5	2.8	4
7	3	31	4	6	2.8	2
8	1	23	2	.	0.8	2
9	3	24	4	6	1.8	4
10	1	26	2	.	1.8	2
11	3	23	4	6	1.8	4
12	1	32	4	5	1.8	2
13	1	29	4	6	1.8	4
14	3	25	4	5	1.8	4
15	3	31	4	5	0.8	2
16	1	26	4	5	2.8	2
17	1	37	2	.	3.8	2
18	1	38	4	5	.	2
19	3	29	2	.	3.8	2
20	3	28	4	6	1.8	2
21	3	28	4	1	2.8	4
22	3	28	4	6	1.8	4
23	3	38	4	5	2.8	2
24	3	27	4	1	1.8	4
25	1	27	2	.	2.8	4

Untuk kemudahan, Anda dapat meletakkan output dari readtable() ke dalam sebuah daftar.

```
>Table={{readtable("table.dat",ctok=ctok)}},
```

Dengan menggunakan kolom-kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dsb. atau menggunakan daftar Table.

```
>writetable(Table,ctok=ctok,wc=5);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n
11	f	23	y	vg	1.8	y
12	m	32	y	g	1.8	n
13	m	29	y	vg	1.8	y
14	f	25	y	g	1.8	y
15	f	31	y	g	0.8	n
16	m	26	y	g	2.8	n
17	m	37	n	.	3.8	n

```

18   m  38      y      g   .
19   f  29      n      .  3.8  n
20   f  28      y      vg  1.8  n
21   f  28      y      m   2.8  y
22   f  28      y      vg  1.8  y
23   f  38      y      g   2.8  n
24   f  27      y      m   1.8  y
25   m  27      n      .  2.8  y

```

Fungsi `tablecol()` mengembalikan nilai-nilai dari kolom-kolom tabel, melewati baris-baris dengan nilai NAN (".") dalam file, dan indeks-indeks kolom yang berisi nilai-nilai ini.

```
>{c,i}=tablecol(MT,[5,6]);
```

Kita dapat menggunakan untuk mengekstrak kolom-kolom dari tabel untuk membuat tabel baru.

```
>j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok)
```

Person	Evaluation	Tip
2	g	1.8
3	g	1.8
6	g	2.8
7	vg	2.8
9	vg	1.8
11	vg	1.8
12	g	1.8
13	vg	1.8
14	g	1.8
15	g	0.8
16	g	2.8
20	vg	1.8
21	m	2.8
22	vg	1.8
23	g	2.8
24	m	1.8

Tentu saja, kita perlu mengekstrak tabel itu sendiri dari daftar Table dalam hal ini.

```
>MT=Table[1];
```

Tentu saja, kita juga bisa menggunakan untuk menentukan nilai rata-rata dari sebuah kolom atau nilai statistik lainnya.

```
>mean(tablecol(MT,6))
```

2.175

Fungsi `getstatistics()` mengembalikan elemen-elemen dalam sebuah vektor, dan jumlah mereka. Kami merekapkannya pada nilai-nilai "m" dan "f" dalam kolom kedua dari tabel kami.

```
>{xu,count}=getstatistics(tablecol(MT,2)); xu, count,
```

```
[1, 3]  
[12, 13]
```

Kami dapat mencetak hasilnya dalam sebuah tabel baru.

```
>writetable(count', labr=tok[xu])
```

m	12
f	13

Fungsi selecttable() mengembalikan tabel baru dengan nilai-nilai dalam satu kolom yang dipilih dari sebuah vektor indeks. Pertama, kita mencari indeks dari dua nilai dalam tabel token.

```
>v:=indexof(tok, ["g", "vg"])
```

```
[5, 6]
```

Sekarang kita dapat memilih baris-baris dari tabel yang memiliki salah satu dari nilai dalam vektor v dalam baris kelima mereka.

```
>MT1:=MT[selectrows(MT, 5, v)]; i:=sortedrows(MT1, 5);
```

Sekarang kita bisa mencetak tabel, dengan nilai-nilai yang diekstrak dan diurutkan dalam kolom kelima.

```
>writetable(MT1[i], labc=hd, ctok=ctok, tok=tok, wc=7);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
6	m	28	y	g	2.8	y
18	m	38	y	g	.	n
16	m	26	y	g	2.8	n
15	f	31	y	g	0.8	n
12	m	32	y	g	1.8	n
23	f	38	y	g	2.8	n
14	f	25	y	g	1.8	y
9	f	24	y	vg	1.8	y
7	f	31	y	vg	2.8	n
20	f	28	y	vg	1.8	n
22	f	28	y	vg	1.8	y
13	m	29	y	vg	1.8	y
11	f	23	y	vg	1.8	y

Untuk statistik berikutnya, kami ingin menghubungkan dua kolom tabel. Jadi kita akan mengekstrak kolom 2 dan 4 dan mengurutkan tabelnya.

```
>i=sortedrows(MT,[2,4]); ...
> writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)
```

m	n
m	n
m	n
m	n
m	n
m	n
m	n
m	y
m	Y
m	Y
m	Y
m	Y
f	n
f	y
f	Y
f	Y
f	Y
f	Y
f	Y
f	Y
f	Y
f	Y
f	Y
f	Y
f	Y
f	Y

Dengan getstatistics(), kita juga bisa menghubungkan jumlah dalam dua kolom tabel satu sama lain.

```
>MT24=tablecol(MT,[2,4]); ...
>{xu1,xu2,count}=getstatistics(MT24[1],MT24[2]); ...
>writetable(count,labr=tok[xu1],labc=tok[xu2])
```

	n	y
m	7	5
f	1	12

Tabel dapat ditulis ke dalam sebuah file.

```
>filename="test.dat"; ...
>writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
```

Kemudian kita bisa membaca tabel dari file tersebut.

```
>{MT2,hd,tok2,hdr}=readtable(filename,>clabs,>rlabs); ...
>writetable(MT2,labr=hdr,labc=hd)
```

	n	y
m	7	5
f	1	12

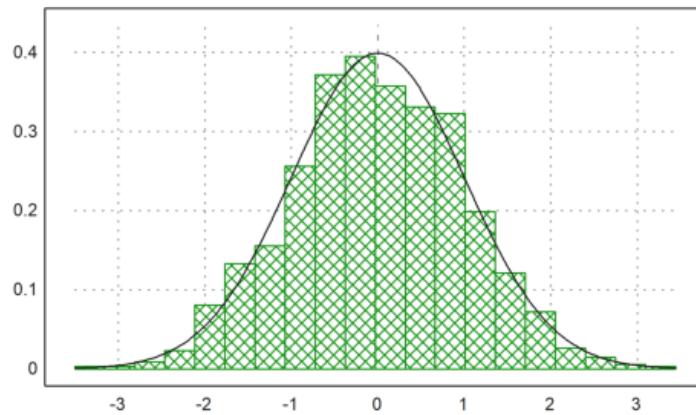
Dan menghapus file tersebut.

```
>fileremove(filename);
```

Distribusi

Dengan plot2d, ada metode yang sangat mudah untuk membuat grafik distribusi data eksperimental.

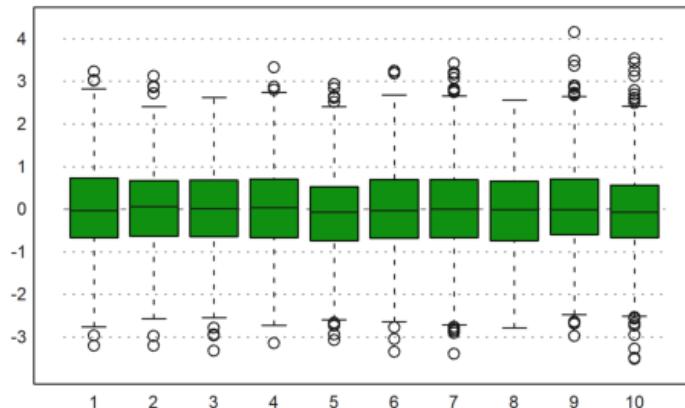
```
>p=normal(1,1000); //1000 random normal-distributed sample p  
>plot2d(p,distribution=20,style="/"); // plot the random sample p  
>plot2d("qnormal(x,0,1)",add=1); // add the standard normal distribution plot
```



Harap perhatikan perbedaan antara diagram batang (sampel) dan kurva normal (distribusi sebenarnya). Masukkan kembali tiga perintah tersebut untuk melihat hasil pengambilan sampel lainnya.

Berikut adalah perbandingan dari 10 simulasi nilai yang didistribusikan secara normal sebanyak 1000 kali menggunakan diagram kotak. Diagram ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, serta outliers.

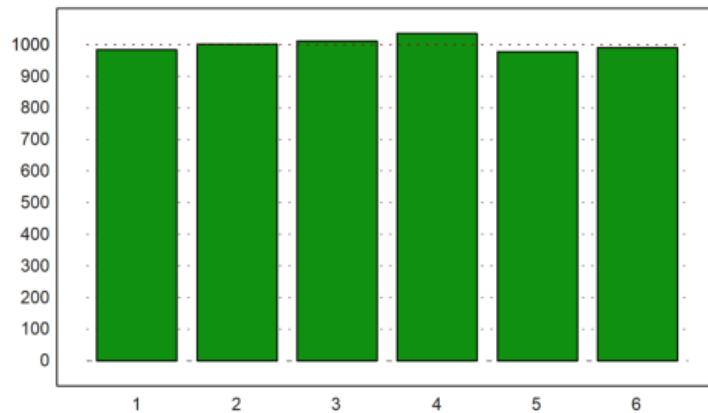
```
>p=normal(10,1000); boxplot(p);
```



Untuk menghasilkan bilangan bulat acak, Euler memiliki intrandom. Mari kita simulasi lemparan dadu dan buat grafik distribusinya.

Kami menggunakan fungsi getmultiplicities(v, x), yang menghitung seberapa sering elemen-elemen v muncul dalam x. Kemudian kami membuat grafik hasilnya menggunakan columnsplot().

```
>k=intrandom(1,6000,6); ...  
>columnsplot (getmultiplicities(1:6,k)); ...  
>ygrid(1000,color=red):
```



Meskipun intrandom(n, m, k) mengembalikan bilangan bulat yang terdistribusi merata dari 1 hingga k, mungkin untuk menggunakan distribusi bilangan bulat lainnya dengan randpint().

Pada contoh berikut, probabilitas untuk 1, 2, 3 masing-masing adalah 0,4, 0,1, 0,5.

```
>randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,%)
```

```
[378, 102, 520]
```

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Lihatlah referensi untuk informasi lebih lanjut.

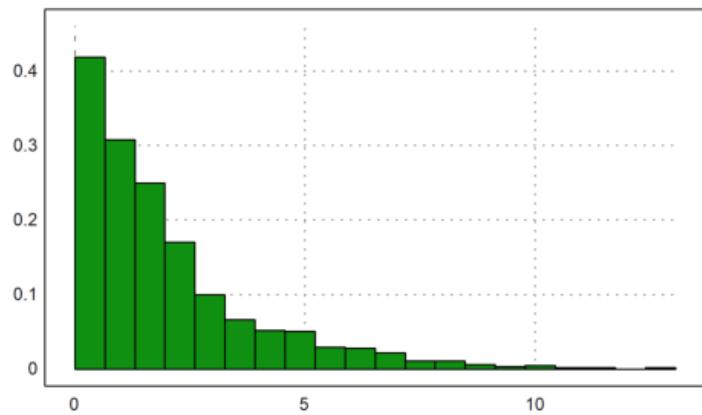
Sebagai contoh, kita mencoba distribusi eksponensial. Variabel acak kontinu X dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

dengan parameter

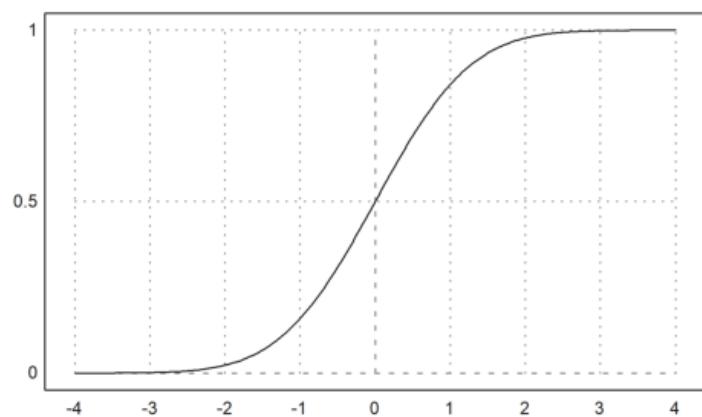
$$\lambda = \frac{1}{\mu}, \quad \mu \text{ adalah rata-rata, dan dilambangkan oleh } X \sim \text{Eksponensial}(\lambda).$$

```
>plot2d(randexponential(1,1000,2),>distribution):
```



Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversnya.

```
>plot2d("normaldis", -4, 4) :
```



Berikut adalah salah satu cara untuk membuat grafik kuantil.

```
>plot2d("qnormal(x,1,1.5)",-4,6); ...
>plot2d("qnormal(x,1,1.5)",a=2,b=5,>add,>filled) :
```

$$\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} dt.$$

Probabilitas berada di area hijau adalah sebagai berikut.

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

0.248662156979

Ini dapat dihitung secara numerik dengan integral berikut.

$$\int_2^5 \frac{1}{1.5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-1}{1.5})^2} dx.$$

```
>gauss("qnormal(x,1,1.5)",2,5)
```

0.248662156979

Mari kita bandingkan distribusi binomial dengan distribusi normal yang memiliki mean dan deviasi yang sama. Fungsi invbindis() menyelesaikan interpolasi linier antara nilai integer.

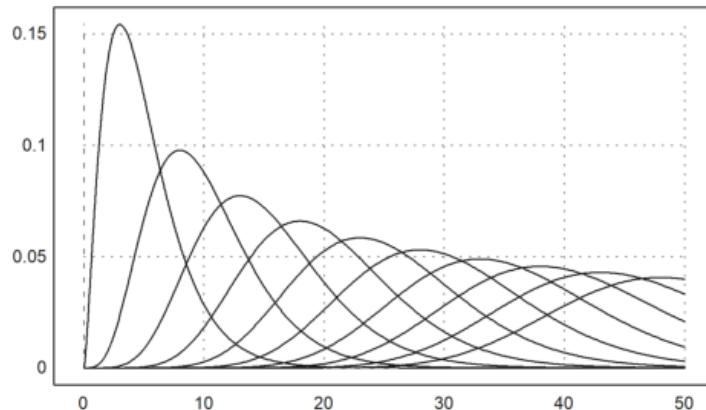
```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

525.516721219

526.007419394

Fungsi qdis() adalah densitas distribusi chi-square. Seperti biasa, Euler memetakan vektor ke fungsi ini. Dengan demikian, kita dengan mudah mendapatkan grafik semua distribusi chi-square dengan derajat 5 hingga 30 dengan cara berikut.

```
>plot2d("qchidis(x,(5:5:50)')",0,50):
```



Euler memiliki fungsi akurat untuk mengevaluasi distribusi. Mari kita periksa chidis() dengan integral.

Penamaan konsisten. Misalnya,

- distribusi chi-square adalah chidis(),
- fungsi inversnya adalah invchidis(),
- densitasnya adalah qchidis().

Komplemen dari distribusi (batas atas) adalah chicdis().

```
>chidis(1.5,2), integrate("qchidis(x,2)",0,1.5)
```

0.527633447259

0.527633447259

Distribusi Diskrit

Untuk mendefinisikan distribusi diskrit Anda sendiri, Anda dapat menggunakan metode berikut. Pertama, kita atur fungsi distribusi.

```
>wd = 0 | ((1:6)+[-0.01,0.01,0,0,0,0])/6
```

```
[0, 0.165, 0.335, 0.5, 0.666667, 0.833333, 1]
```

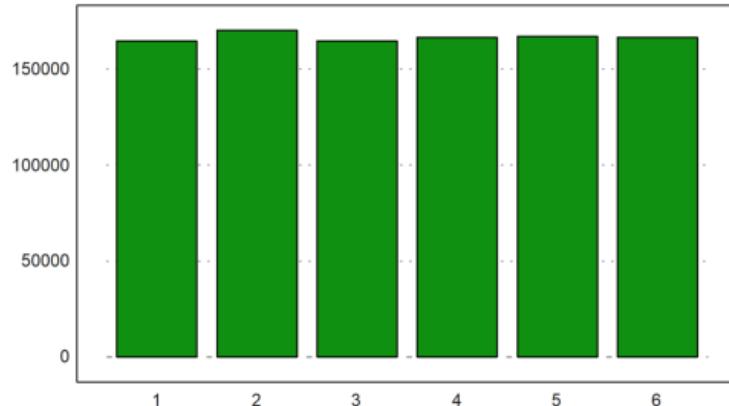
Artinya, dengan probabilitas $wd[i+1]-wd[i]$, kita menghasilkan nilai acak i.

Ini hampir merupakan distribusi seragam. Mari kita definisikan generator angka acak untuk ini. Fungsi $find(v, x)$ menemukan nilai x dalam vektor v. Ini berfungsi juga untuk vektor x.

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahan ini begitu halus sehingga kita hanya melihatnya dengan sangat banyak iterasi.

```
>columnsplot (getmultiplicities(1:6,wrongdice(1,1000000))):
```



Berikut adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai-nilai 1...K dalam v. Kami menerima hasilnya jika untuk semua frekuensi

$$\left| f_i - \frac{1}{K} \right| < \frac{\delta}{\sqrt{n}}.$$

```
>function checkrandom (v, delta=1) ...
```

```
K=max(v); n=cols(v);
fr=getfrequencies(v,1:K);
return max(fr/n-1/K)<delta/sqrt(n);
endfunction
```

Memang, fungsi menolak distribusi seragam.

```
>checkrandom(wrongdice(1,1000000))
```

0

Dan itu menerima generator acak bawaan.

```
>checkrandom(intrandom(1,1000000,6))
```

1

Kita dapat menghitung distribusi binomial. Pertama ada binomials() , yang mengembalikan probabilitas i atau kurang sukses dari percobaan n.

```
>bindis(410,1000,0.4)
```

0.751401349654

Fungsi invers Beta digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p. Tingkat default adalah alpha.

Arti dari interval ini adalah bahwa jika p berada di luar interval, hasil yang diamati dari 410 dari 1000 adalah langka.

```
>clopperpearson(410,1000)
```

[0.37932, 0.441212]

Perintah berikut adalah cara langsung untuk mendapatkan hasil di atas. Tetapi untuk n yang besar, penjumlahan langsung tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

0.751401349655

Invbinsum() menghitung invers dari binomials().

```
>invbindis(0.75,1000,0.4)
```

409.932733047

Dalam Bridge, kita mengasumsikan 5 kartu yang luar biasa (dari 52) dalam dua tangan (26 kartu). Mari kita hitung probabilitas distribusi yang lebih buruk dari 3:2 (misalnya, 0:5, 1:4, 4:1, atau 5:0).

```
>2*hypergeomsum(1,5,13,26)
```

0.321739130435

Ada juga simulasi distribusi multinomial.

```
>randmultinomial(10,1000,[0.4,0.1,0.5])
```

381	100	519
376	91	533
417	80	503
440	94	466
406	112	482
408	94	498
395	107	498
399	96	505
428	87	485
400	99	501

Membuat Plot Data

Untuk membuat plot data, kita mencoba hasil pemilihan umum Jerman sejak tahun 1990, 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];
```

Untuk partai-partai, kita menggunakan deretan nama-nama.

```
>P := ["CDU/CSU", "SPD", "FDP", "Gr", "Li"];
```

Mari cetak persentasenya dengan rapi.

Pertama, kita ekstrak kolom-kolom yang diperlukan. Kolom 3 hingga 7 adalah kursi masing-masing partai, dan kolom 2 adalah total jumlah kursi. Kolom ini adalah tahun dari pemilihan.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Kemudian kita cetak statistik dalam bentuk tabel. Kita menggunakan nama-nama sebagai judul kolom, dan tahun sebagai judul untuk baris. Lebar default untuk kolom adalah `wc=10`, tetapi kita lebih suka keluaran yang lebih padat. Kolom-kolom akan diperluas untuk label-label kolom, jika diperlukan.

```
>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

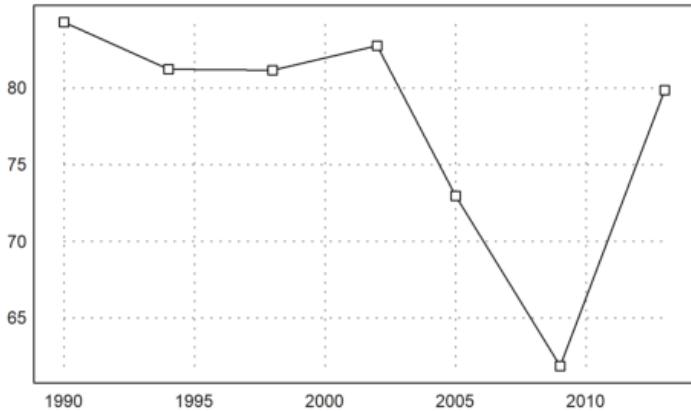
Perkalian matriks berikutnya mengekstrak jumlah persentase dari dua partai besar yang menunjukkan bahwa partai-partai kecil telah mendapatkan tempat di parlemen hingga tahun 2009.

```
>BT1:=(BT.[1;1;0;0;0])'*100
```

```
[84.29, 81.25, 81.1659, 82.7529, 72.9642, 61.8971, 79.8732]
```

Ada juga plot statistik sederhana. Kita menggunakan untuk menampilkan garis dan titik secara bersamaan. Alternatifnya adalah memanggil plot2d dua kali dengan >add.

```
>statplot(YT,BT1,"b"):
```



Tentukan beberapa warna untuk setiap partai.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

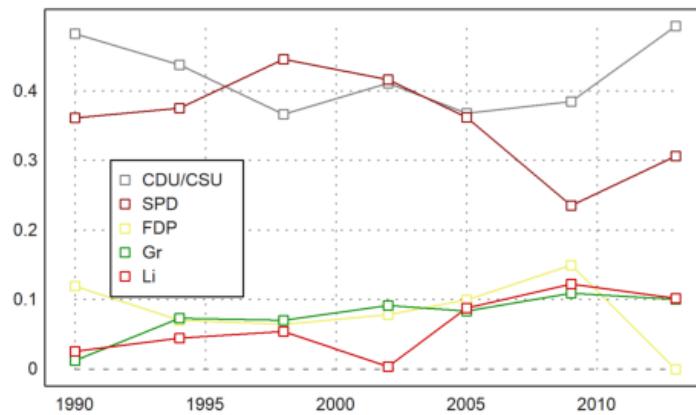
Sekarang kita bisa membuat plot hasil pemilihan tahun 2009 dan perubahannya ke dalam satu plot menggunakan figure. Kita bisa menambahkan vektor kolom ke setiap plot.

```
>figure(2,1); ...
>figure(1); columnsplot(BW[6,3:7],P,color=CP); ...
>figure(2); columnsplot(BW[6,3:7]-BW[5,3:7],P,color=CP); ...
>figure(0):
```



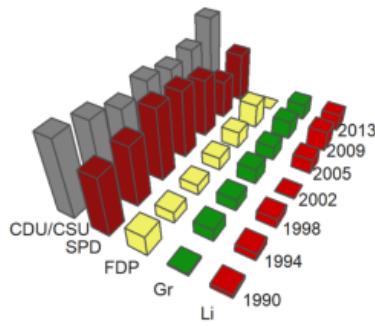
Plot data menggabungkan baris data statistik dalam satu plot.

```
>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):
```



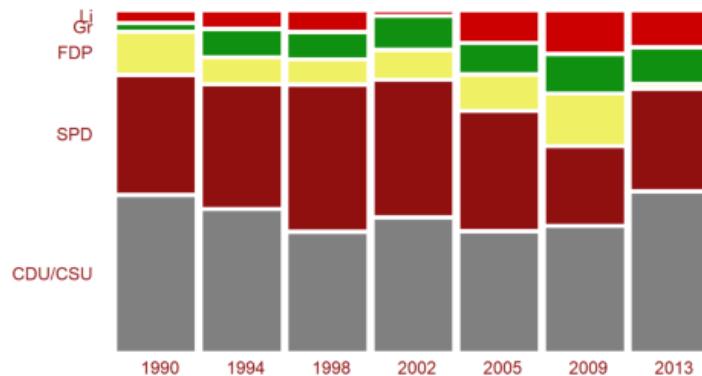
Plot kolom 3D menunjukkan baris data statistik dalam bentuk kolom. Kami menyediakan label untuk baris dan kolom. Sudut adalah sudut pandang.

```
>columnsplot3d(BT,scols=P,srows=YT, ...
> angle=30°,ccols=CP):
```



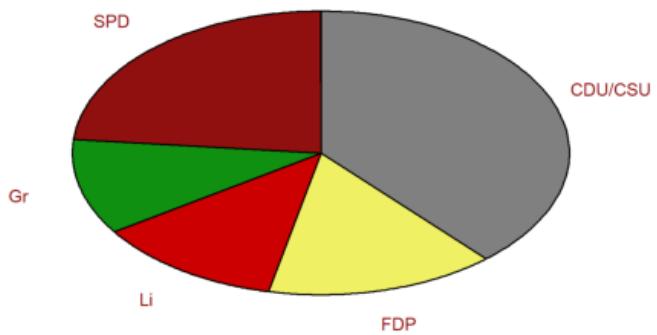
Representasi lainnya adalah plot mozaik. Perhatikan bahwa kolom plot mewakili kolom matriks di sini. Karena panjang label CDU/CSU, kita mengambil jendela yang lebih kecil dari biasanya.

```
>shrinkwindow(>smaller); ...
>mosaicplot(BT', srows=YT, scols=P, color=CP, style="#");
>shrinkwindow():
```



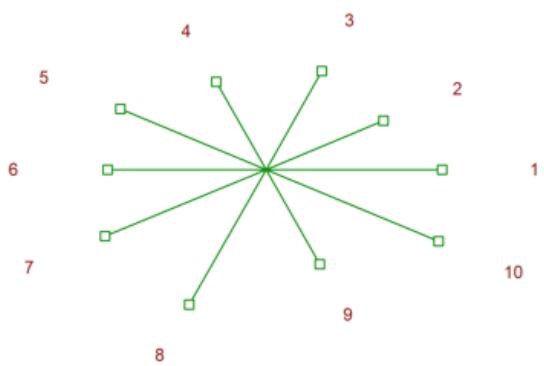
Kita juga bisa membuat diagram lingkaran. Karena hitam dan kuning membentuk koalisi, kita mengurutkan ulang elemennya.

```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```



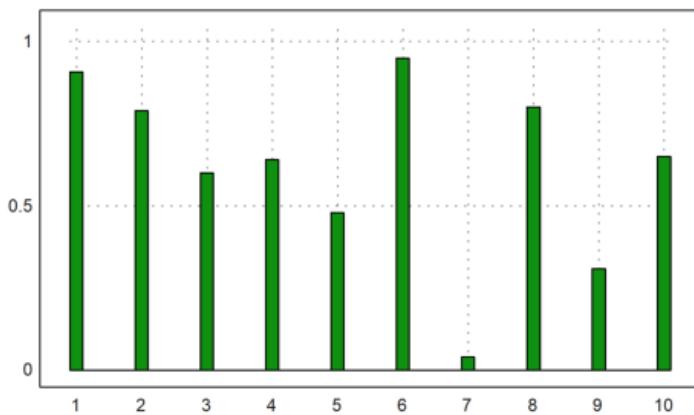
Berikut adalah jenis plot lainnya.

```
>starplot(normal(1,10)+4,lab=1:10,>rays) :
```



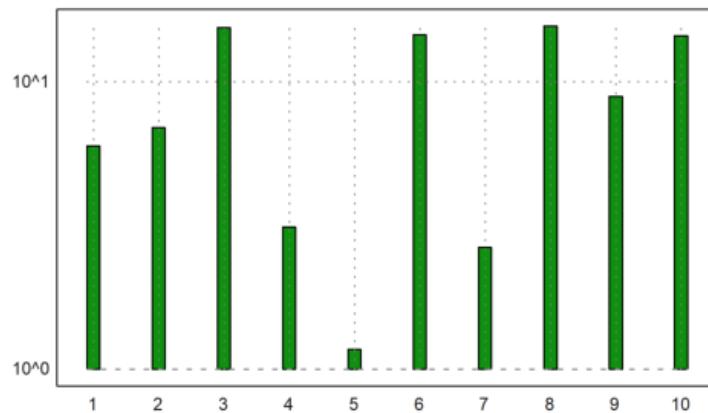
Beberapa plot dalam plot2d bagus untuk statistik. Berikut adalah plot impuls dari data acak, terdistribusi merata dalam [0,1].

```
>plot2d(makeimpulse(1:10,random(1,10)),>bar) :
```



Tetapi untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

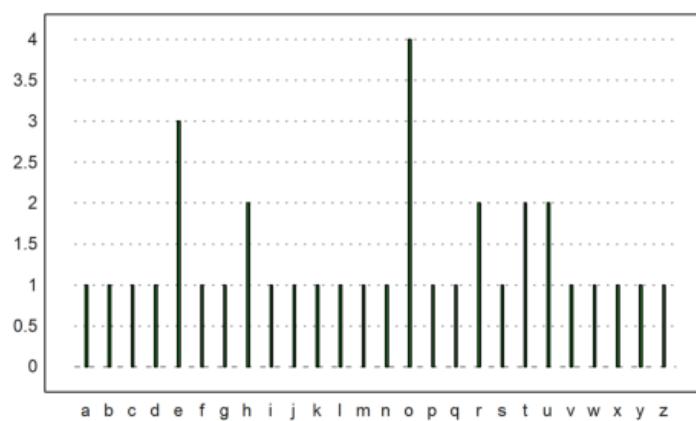
```
>logimpulseplot(1:10,-log(random(1,10))*10):
```



Fungsi `columnsplot()` lebih mudah digunakan, karena hanya membutuhkan vektor nilai. Selain itu, dapat menetapkan labelnya menjadi apa pun yang kita inginkan, seperti yang sudah kami tunjukkan dalam tutorial ini.

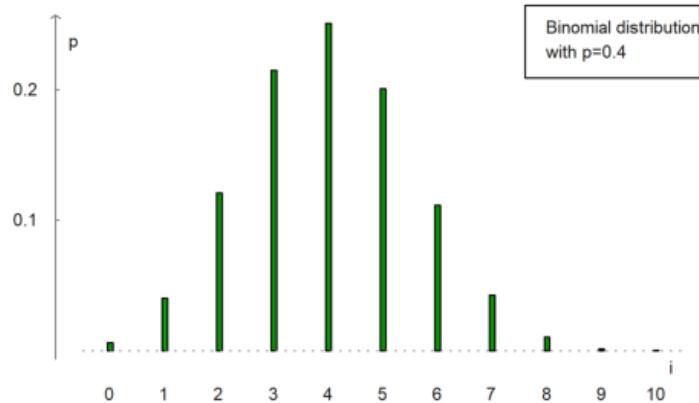
Berikut adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan membuat statistiknya.

```
>v=char("the quick brown fox jumps over the lazy dog"); ...
>w=ascii("a"):ascii("z"); x=getmultiplicities(w,v); ...
>cw=[]; for k=w; cw=cw|char(k); end; ...
>columnsplot(x,lab=cw,width=0.05):
```



Juga mungkin untuk secara manual mengatur sumbu-sumbu

```
>n=10; p=0.4; i=0:n; x=bin(n,i)*p^i*(1-p)^(n-i); ...
>columnsplot(x,lab=i,width=0.05,<frame,<grid); ...
>yaxis(0,0:0.1:1,style="->",>left); xaxis(0,style="."); ...
>label("p",0,0.25), label("i",11,0); ...
>textbox(["Binomial distribution","with p=0.4"]):
```



Berikut adalah cara untuk membuat plot frekuensi angka dalam sebuah vektor.

Kita membuat vektor angka acak 1 hingga 6.

```
>v:=intrandom(1,10,10)
```

```
[8, 5, 8, 8, 6, 8, 8, 3, 5, 5]
```

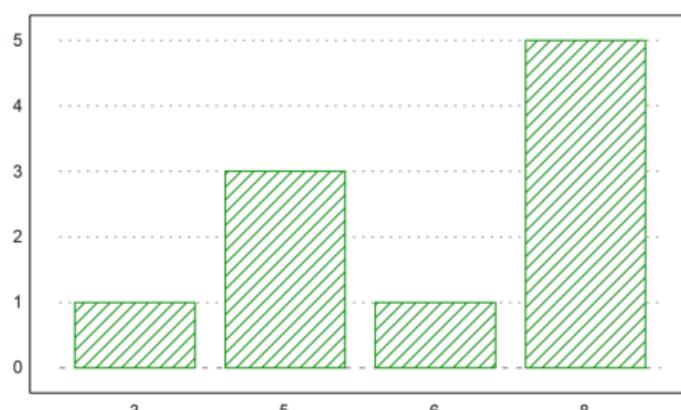
Kemudian ekstrak angka-angka unik dalam v.

```
>vu:=unique(v)
```

```
[3, 5, 6, 8]
```

Dan plot frekuensinya dalam plot kolom.

```
>columnsplot(getmultiplicities(vu,v),lab=vu,style="/"):
```



Kami ingin menunjukkan fungsi untuk distribusi empiris dari nilai-nilai.

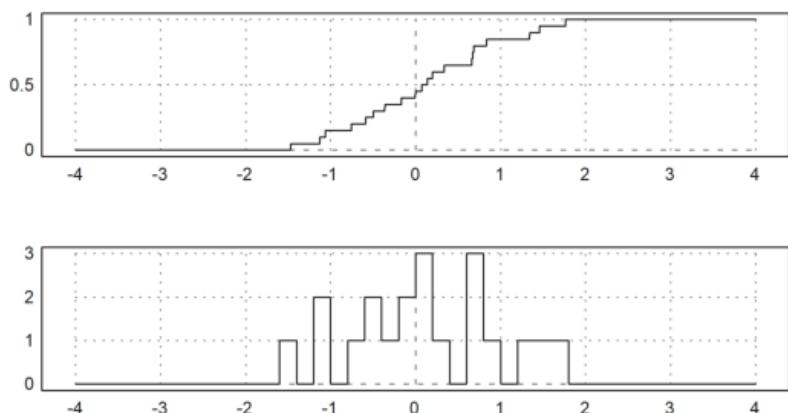
```
>x=normal(1,20);
```

Fungsi empdist(x,vs) membutuhkan array nilai yang terurut. Jadi kita harus mengurutkan x sebelum kita bisa menggunakannya.

```
>xs=sort(x);
```

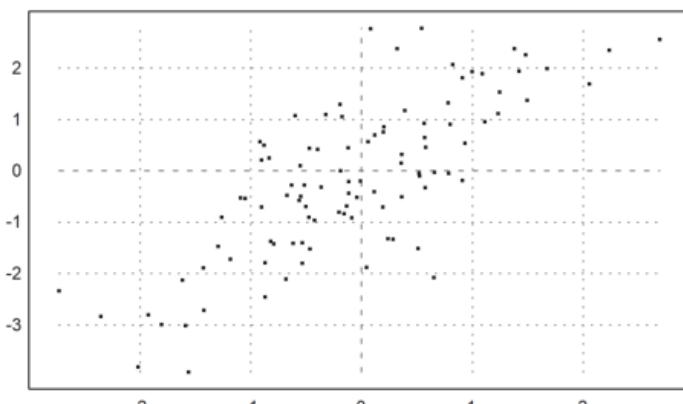
Kemudian kita membuat plot distribusi empiris dan beberapa batang densitas dalam satu plot. Alih-alih plot batang untuk distribusi, kita menggunakan plot sawtooth kali ini.

```
>figure(2,1); ...
>figure(1); plot2d("empdist",-4,4;xs); ...
>figure(2); plot2d(histo(x,v=-4:0.2:4,<bar)); ...
>figure(0):
```



Plot pencar mudah dilakukan di Euler dengan plot titik biasa. Grafik berikut menunjukkan bahwa X dan X+Y jelas berkorelasi positif.

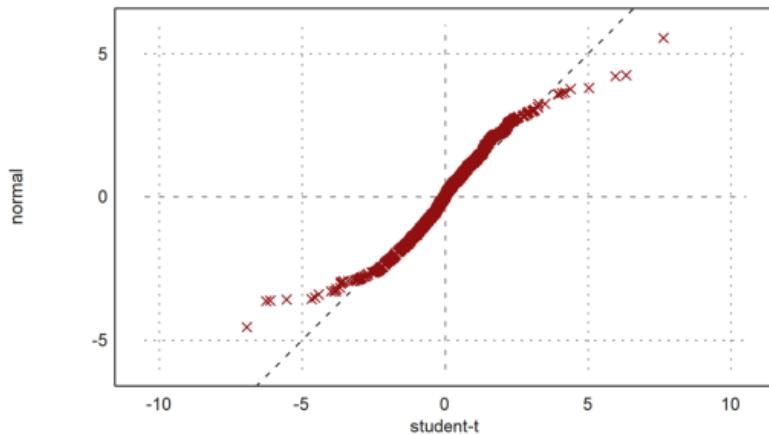
```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style="."):
```



Seringkali, kita ingin membandingkan dua sampel dari distribusi yang berbeda. Ini bisa dilakukan dengan plot kuantil-kuantil.

Untuk uji coba, kita mencoba distribusi t-student dan distribusi eksponensial.

```
>x=randt(1,1000,5); y=randnormal(1,1000,mean(x),dev(x)); ...
>plot2d("x",r=6,style="--",yl="normal",xl="student-t",>vertical); ...
>plot2d(sort(x),sort(y),>points,color=red,style="x",>add):
```

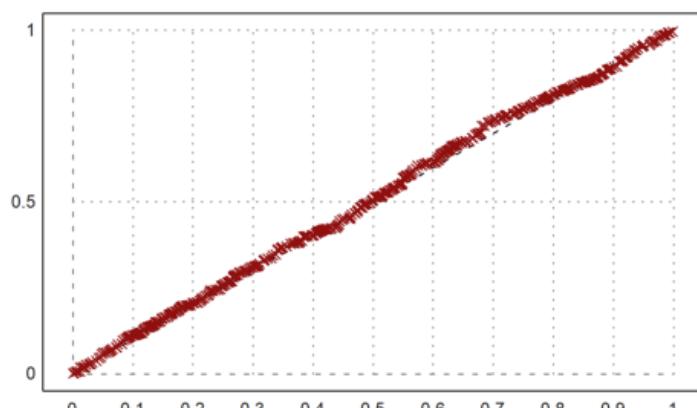


Plot dengan jelas menunjukkan bahwa nilai yang terdistribusi normal cenderung lebih kecil di ujung ekstrim. Jika kita memiliki dua distribusi berukuran berbeda, kita bisa memperluas yang lebih kecil atau mengecilkan yang lebih besar. Fungsi berikutnya baik untuk keduanya. Ini mengambil nilai median dengan persentase antara 0 dan 1.

```
>function medianexpand (x,n) := median(x,p=linspace(0,1,n-1));
```

Mari membandingkan dua distribusi yang sama.

```
>x=random(1000); y=random(400); ...
>plot2d("x",0,1,style="--"); ...
>plot2d(sort(medianexpand(x,400)),sort(y),>points,color=red,style="x",>add):
```



Regressi dan Korelasi

Regresi linear dapat dilakukan dengan menggunakan fungsi polyfit() atau berbagai fungsi fit lainnya. Pertama, kita mencari garis regresi untuk data univariat dengan polyfit(x, y, 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

Kita ingin membandingkan fit tanpa bobot dan fit dengan bobot. Pertama, koefisien fit linier.

```
>p=polyfit(x,y,1)
```

```
[0.733333, 0.812121]
```

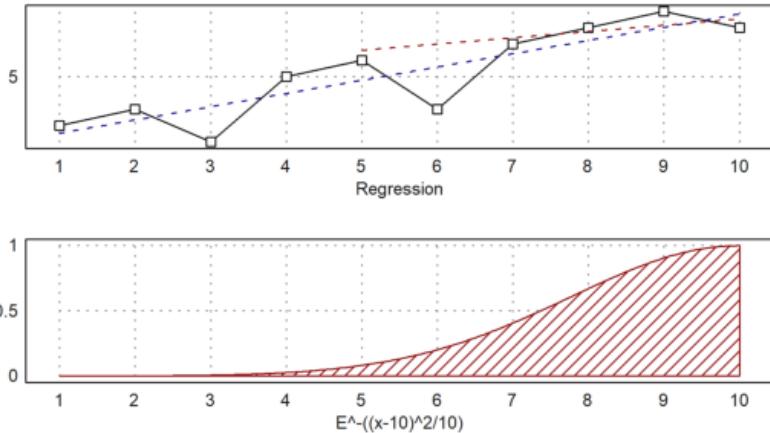
Sekarang, koefisien dengan bobot yang menekankan pada nilai terakhir.

```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

```
[4.71566, 0.38319]
```

Kita masukkan semua ke dalam satu plot untuk titik-titik dan garis regresi, serta untuk bobot yang digunakan.

```
>figure(2,1); ...
>figure(1); statplot(x,y,"b",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):
```



Sebagai contoh lain, kita membaca hasil survei mahasiswa, usia mereka, usia orangtua mereka, dan jumlah saudara dari sebuah file.

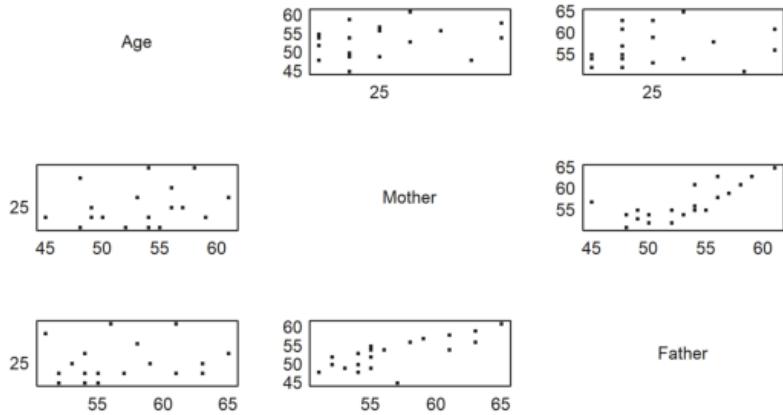
Tabel ini berisi "m" dan "f" di kolom kedua. Kita menggunakan variabel tok2 untuk mengatur terjemahan yang tepat daripada membiarkan readtable() mengumpulkan terjemahan.

```
>{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

Bagaimana usia bergantung satu sama lain? Kesimpulan awal bisa didapat dari scatterplot berpasangan.

```
>scatterplots(tablecol(MS,3:5),hd[3:5]):
```



Jelas bahwa usia ayah dan ibu saling bergantung. Mari tentukan dan gambarkan garis regresi.

```
>cs:=MS[, 4:5]'; ps:=polyfit(cs[1], cs[2], 1)
```

```
[17.3789, 0.740964]
```

Ini jelas adalah model yang salah. Garis regresi akan menjadi $s=17+0,74t$, di mana t adalah usia ibu dan s adalah usia ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak begitu besar.

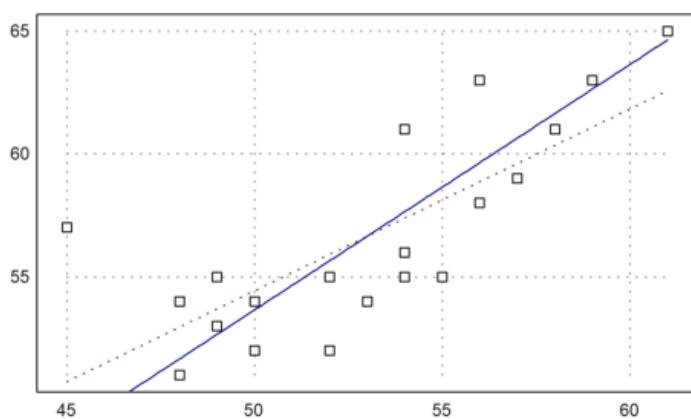
Sebaliknya, kita curigai sebuah fungsi seperti $s=a+t$. Kemudian, a adalah rata-rata dari $s-t$. Itu adalah perbedaan usia rata-rata antara ayah dan ibu.

```
>da:=mean(cs[2]-cs[1])
```

```
3.65
```

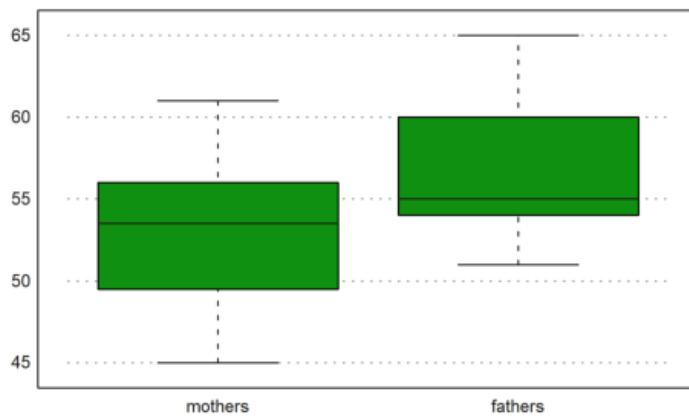
Mari plot ini ke dalam satu scatter plot.

```
>plot2d(cs[1],cs[2],>points); ...
>plot2d("evalpoly(x,ps)",color=red,style=".",>add); ...
>plot2d("x+da",color=blue,>add):
```



Berikut adalah box plot dari dua usia tersebut. Ini hanya menunjukkan bahwa usia mereka berbeda.

```
>boxplot(cs, ["mothers", "fathers"]):
```



Menarik bahwa perbedaan median tidak sebesar perbedaan mean.

```
>median(cs[2])-median(cs[1])
```

1.5

Koefisien korelasi menunjukkan korelasi positif.

```
>correl(cs[1],cs[2])
```

0.7588307236

Korelasi peringkat adalah ukuran untuk urutan yang sama dalam kedua vektor. Ini juga cukup positif.

```
>rankcorrel(cs[1],cs[2])
```

0.758925292358

Membuat Fungsi Baru

Tentu saja, bahasa EMT dapat digunakan untuk memprogram fungsi-fungsi baru. Sebagai contoh, kita mendefinisikan fungsi kemiringan (skewness).

$$sk(x) = \frac{\sqrt{n} \sum_i (x_i - m)^3}{(\sum_i (x_i - m)^2)^{3/2}}$$

di mana m adalah rata-rata dari x.

```
>function skew (x:vector) ...
```

```
m=mean(x);  
return sqrt(cols(x))*sum((x-m)^3)/(sum((x-m)^2))^(3/2);  
endfunction
```

Seperti yang Anda lihat, kita dapat dengan mudah menggunakan bahasa matriks untuk mendapatkan implementasi yang sangat singkat dan efisien. Mari kita coba fungsi ini.

```
>data=normal(20); skew(normal(10))
```

```
-0.198710316203
```

Berikut adalah fungsi lain, yang disebut koefisien kemiringan Pearson.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)
>skew1(data)
```

```
-0.0801873249135
```

Simulasi Monte Carlo

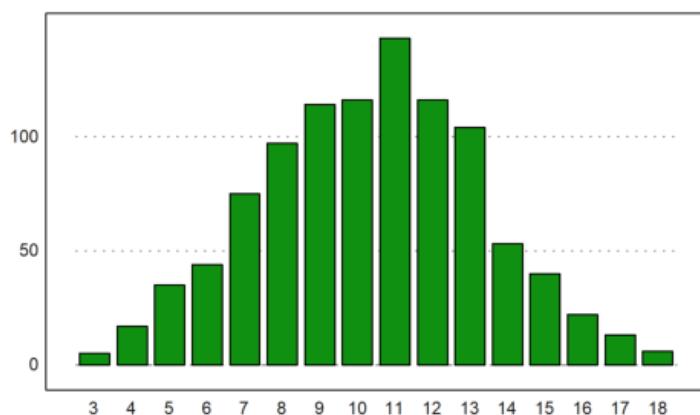
Euler dapat digunakan untuk mensimulasikan peristiwa acak. Kita sudah melihat contoh sederhana di atas. Berikut adalah contoh lain, yang mensimulasikan 1000 kali lemparan tiga dadu, dan bertanya tentang distribusi jumlahnya.

```
>ds:=sum(intrandom(1000,3,6)); fs=getmultiplicities(3:18,ds)
```

```
[5, 17, 35, 44, 75, 97, 114, 116, 143, 116, 104, 53, 40,
22, 13, 6]
```

Sekarang kita bisa membuat plot ini.

```
>columnsplot(fs,lab=3:18):
```



Menentukan distribusi yang diharapkan tidak semudah itu. Kami menggunakan rekursi secara lebih dalam untuk ini.

Fungsi berikut menghitung jumlah cara angka k dapat diwakili sebagai jumlah n angka dalam rentang 1 hingga m. Ini bekerja secara rekursif dengan cara yang jelas.

```
>function map countways (k; n, m) ...
if n==1 then return k>=1 && k<=m
else
  sum=0;
  loop 1 to m; sum=sum+countways(k-#,n-1,m); end;
  return sum;
end;
endfunction
```

Berikut adalah hasil untuk tiga lemparan dadu.

```
>countways(5:25,5,5)
```

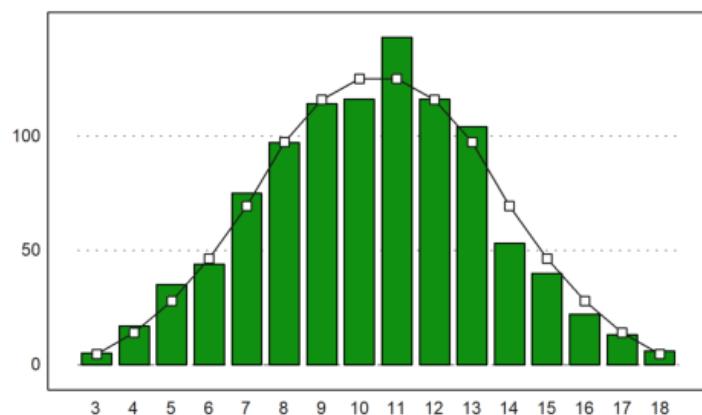
```
[1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320,
255, 185, 121, 70, 35, 15, 5, 1]
```

```
>cw=countways(3:18,3,6)
```

```
[1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3,
1]
```

Kita menambahkan nilai yang diharapkan ke plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```



Untuk simulasi lain, deviasi nilai rata-rata dari n variabel acak yang terdistribusi normal antara 0-1 adalah $1/\sqrt{n}$.

```
>longformat; 1/sqrt(10)
```

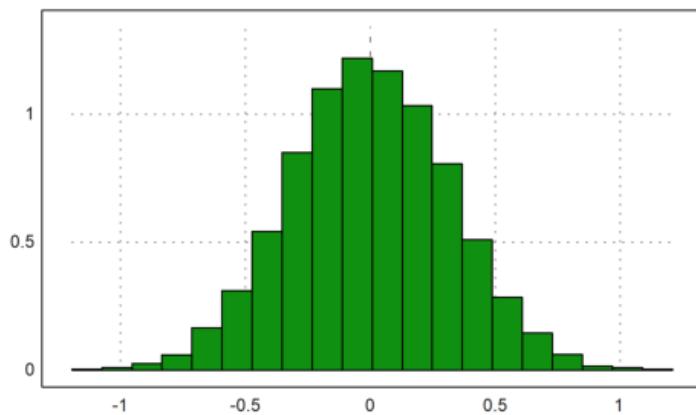
0.316227766017

Mari kita periksa ini dengan simulasi. Kami menghasilkan 10000 kali vektor acak sebanyak 10.

```
>M=normal(10000,10); dev(mean(M)')
```

0.319493614817

```
>plot2d(mean(M)',>distribution):
```



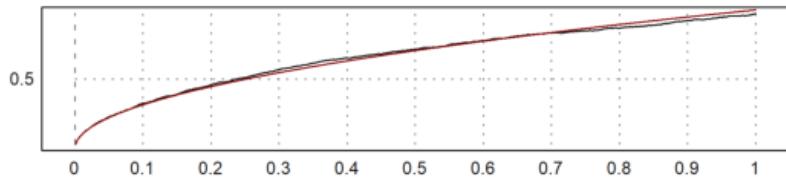
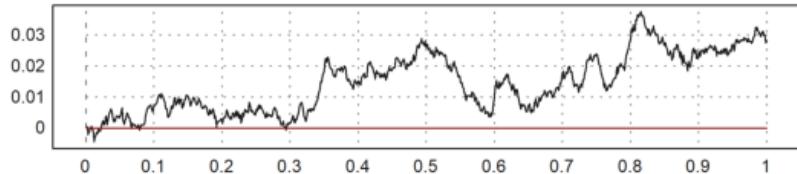
Deviasi median dari 10 angka acak yang terdistribusi normal antara 0-1 memiliki deviasi yang lebih besar.

```
>dev(median(M)')
```

0.374460271535

Karena kita dapat dengan mudah menghasilkan langkah-langkah acak, kita dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami kemudian membuat plot deviasi standar dan rata-rata langkah ke-n dari proses-proses ini bersama dengan nilai yang diharapkan dalam warna merah.

```
>n=1000; m=1000; M=cumsum(normal(n,m)/sqrt(m)); ...
>t=(1:n)/n; figure(2,1); ...
>figure(1); plot2d(t,mean(M)'); plot2d(t,0,color=red,>add); ...
>figure(2); plot2d(t,dev(M)'); plot2d(t,sqrt(t),color=red,>add); ...
>figure(0):
```



Tests

Uji adalah alat penting dalam statistik. Di Euler, banyak uji diimplementasikan. Semua uji ini menghasilkan galat yang kita terima jika kita menolak hipotesis nol.

Sebagai contoh, kita menguji lemparan dadu untuk distribusi seragam. Pada 600 lemparan, kita mendapatkan nilai-nilai berikut, yang kita masukkan ke dalam uji chi-square.

```
>chitest([90,103,114,101,103,89],dup(100,6)')
```

0.498830517952

Uji chi-square juga memiliki mode, yang menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya seharusnya hampir sama. Parameter `>p` mengartikan vektor `y` sebagai vektor probabilitas.

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

0.526

Galat ini terlalu besar. Jadi kita tidak dapat menolak distribusi seragam. Ini tidak membuktikan bahwa dadu kita adil. Tetapi kita tidak bisa menolak hipotesis kita.

Selanjutnya, kita menghasilkan 1000 lemparan dadu menggunakan generator angka acak dan melakukan uji yang sama.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

0.528028118442

Mari uji untuk nilai rata-rata 100 dengan uji t.

```
>s=200+normal([1,100])*10; ...
>ttest(mean(s),dev(s),100,200)
```

0.0218365848476

Fungsi ttest() membutuhkan nilai rata-rata, deviasi, jumlah data, dan nilai rata-rata yang akan diuji.

Sekarang mari kita periksa dua pengukuran untuk nilai rata-rata yang sama. Kita menolak hipotesis bahwa mereka memiliki nilai rata-rata yang sama, jika hasilnya $<0,05$.

```
>tcomparedata(normal(1,10),normal(1,10))
```

0.38722000942

Jika kita menambahkan bias ke salah satu distribusi, kita mendapatkan lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

5.60009101758e-07

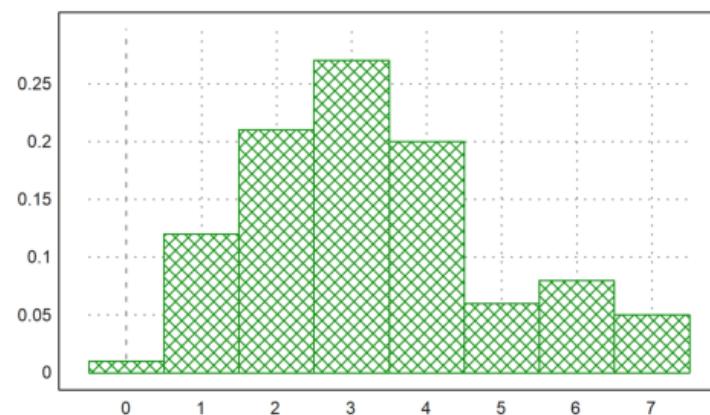
Dalam contoh berikutnya, kita menghasilkan 20 lemparan dadu acak 100 kali dan menghitung jumlah satu. Harus ada $20/6=3,3$ satu secara rata-rata.

```
>R=random(100,20); R=sum(R*6<=1); mean(R)
```

3.28

Kemudian kita membandingkan jumlah satu dengan distribusi binomial. Pertama, kita plot distribusi satu.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\\"/"):
```



```
>t=count(R,21);
```

Kemudian kita menghitung nilai yang diharapkan.

```
>n=0:20; b=bin(20,n)*(1/6)^n*(5/6)^(20-n)*100;
```

Kita harus mengumpulkan beberapa angka untuk mendapatkan kategori yang cukup besar.

```
>t1=sum(t[1:2])|t[3:7]|sum(t[8:21]); ...
>b1=sum(b[1:2])|b[3:7]|sum(b[8:21]);
```

Uji chi-square menolak hipotesis bahwa distribusi kita adalah distribusi binomial, jika hasilnya <0,05.

```
>chitest(t1,b1)
```

0.53921579764

Contoh berikutnya berisi hasil dua kelompok orang (pria dan wanita, katakanlah) yang memilih salah satu dari enam partai.

```
>A=[23,37,43,52,64,74;27,39,41,49,63,76]; ...
> writetable(A,wc=6,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	23	37	43	52	64	74
f	27	39	41	49	63	76

Kita ingin menguji independensi suara dari jenis kelamin. Uji tabel chi-square melakukan ini. Hasilnya terlalu besar untuk menolak independensi. Jadi kita tidak bisa mengatakan apakah pemilihan tergantung pada jenis kelamin dari data ini.

```
>tabletest(A)
```

0.990701632326

Berikut adalah tabel yang diharapkan, jika kita mengasumsikan frekuensi pemilihan yang diamati.

```
>writetable(expectedtable(A),wc=6,dc=1,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	24.9	37.9	41.9	50.3	63.3	74.7
f	25.1	38.1	42.1	50.7	63.7	75.3

Kita dapat menghitung koefisien kontingensi yang sudah dikoreksi. Karena sangat mendekati 0, kita menyimpulkan bahwa pemilihan tidak bergantung pada jenis kelamin.

```
>contingency(A)
```

0.0427225484717

Beberapa Test Tambahan

Selanjutnya, kita menggunakan analisis varians (uji F) untuk menguji tiga sampel data yang terdistribusi secara normal untuk nilai rata-rata yang sama. Metodenya disebut ANOVA (analisis varians). Di dalam Euler, fungsi varanalysis() digunakan.

```
>x1=[109,111,98,119,91,118,109,99,115,109,94]; mean(x1),
```

106.545454545

```
>x2=[120,124,115,139,114,110,113,120,117]; mean(x2),
```

119.111111111

```
>x3=[120,112,115,110,105,134,105,130,121,111]; mean(x3)
```

116.3

```
>varanalysis(x1,x2,x3)
```

0.0138048221371

Ini berarti kita menolak hipotesis nilai rata-rata yang sama. Kita melakukannya dengan tingkat kesalahan sebesar 1,3%.

Ada juga uji median, yang menolak sampel data dengan distribusi rata-rata yang berbeda dengan menguji median dari sampel yang digabung.

```
>a=[56,66,68,49,61,53,45,58,54];
>b=[72,81,51,73,69,78,59,67,65,71,68,71];
>mediantest(a,b)
```

0.0241724220052

Uji lain tentang kesetaraan adalah uji peringkat. Ini jauh lebih tajam dibandingkan uji median.

```
>ranktest(a,b)
```

0.00199969612469

Pada contoh berikut, kedua distribusi memiliki nilai rata-rata yang sama.

```
>ranktest(random(1,100),random(1,50)*3-1)
```

0.129608141484

Mari kita coba mensimulasikan dua perlakuan, yaitu a dan b, yang diterapkan pada orang yang berbeda.

```
>a=[8.0,7.4,5.9,9.4,8.6,8.2,7.6,8.1,6.2,8.9];  
>b=[6.8,7.1,6.8,8.3,7.9,7.2,7.4,6.8,6.8,8.1];
```

Uji tanda menentukan apakah a lebih baik daripada b.

```
>signtest(a,b)
```

0.0546875

Ini terlalu banyak kesalahan. Kita tidak dapat menolak bahwa a sama baiknya dengan b.

Uji Wilcoxon lebih tajam dibandingkan uji ini, tetapi bergantung pada nilai kuantitatif dari perbedaan.

```
>wilcoxon(a,b)
```

0.0296680599405

Mari kita coba dua uji lagi menggunakan rangkaian data yang dihasilkan.

```
>wilcoxon(normal(1,20),normal(1,20)-1)
```

0.0068706451766

```
>wilcoxon(normal(1,20),normal(1,20))
```

0.275145971064

Bilangan Acak

Berikut adalah uji coba untuk generator bilangan acak. Euler menggunakan generator yang sangat baik, jadi kita tidak perlu mengharapkan masalah apa pun.

Pertama, kita menghasilkan sepuluh juta bilangan acak dalam rentang [0,1].

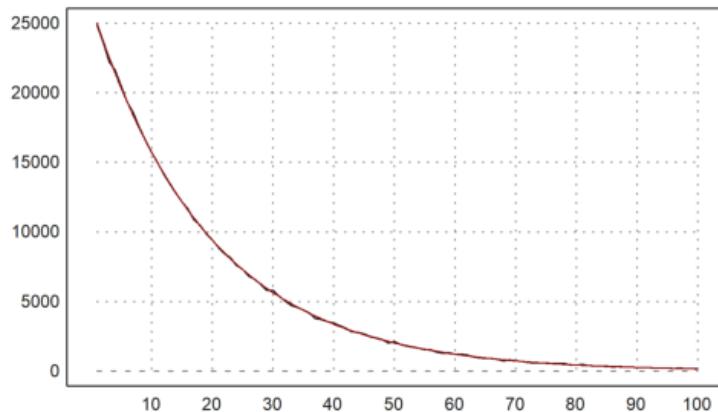
```
>n:=10000000; r:=random(1,n);
```

Selanjutnya, kita menghitung jarak antara dua bilangan kurang dari 0,05.

```
>a:=0.05; d:=differences(nonzeros(r<a));
```

Terakhir, kita plot jumlah kali setiap jarak terjadi, dan membandingkannya dengan nilai yang diharapkan.

```
>m=getmultiplicities(1:100,d); plot2d(m); ...
> plot2d("n*(1-a)^(x-1)*a^2",color=red,>add):
```



Hapus data.

```
>remvalue n;
```

Pengantar untuk Pengguna Proyek R

Jelas, EMT tidak bersaing dengan R sebagai paket statistik. Namun, ada banyak prosedur statistik dan fungsi yang tersedia di EMT juga. Jadi, EMT mungkin memenuhi kebutuhan dasar. Pada dasarnya, EMT dilengkapi dengan paket numerik dan sistem aljabar komputer.

Notebook ini untuk Anda jika Anda akrab dengan R, tetapi perlu mengetahui perbedaan sintaks antara EMT dan R. Kami berusaha memberikan gambaran umum tentang hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami akan melihat cara pertukaran data antara kedua sistem ini.

Perhatikan bahwa ini masih dalam tahap pengembangan.

Sintaks Dasar

Hal pertama yang dipelajari dalam R adalah membuat vektor. Dalam EMT, perbedaan utamanya adalah operator ":" dapat menggunakan ukuran langkah. Selain itu, ia memiliki kekuatan ikatan rendah.

```
>n=10; 0:n/20:n-1
```

```
[0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5,
7, 7.5, 8, 8.5, 9]
```

Fungsi c() tidak ada. Mungkin menggunakan vektor untuk menggabungkan hal-hal.

Contoh berikutnya, seperti banyak contoh lainnya, berasal dari "Pengenalan ke R" yang disertakan dalam proyek R. Jika Anda membaca PDF ini, Anda akan menemukan bahwa saya mengikuti jalannya dalam tutorial ini.

```
>x=[10.4, 5.6, 3.1, 6.4, 21.7]; [x,0,x]
```

```
[10.4, 5.6, 3.1, 6.4, 21.7, 0, 10.4, 5.6, 3.1, 6.4, 21.7]
```

Operator titik dua dengan ukuran langkah EMT digantikan oleh fungsi seq() dalam R. Kita dapat menulis fungsi ini dalam EMT.

```
>function seq(a,b,c) := a:b:c; ...  
>seq(0,-0.1,-1)
```

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

Fungsi rep() dari R tidak ada dalam EMT. Untuk input vektor, dapat ditulis sebagai berikut.

```
>function rep(x:vector,n:index) := flatten(dup(x,n)); ...  
>rep(x,2)
```

```
[10.4, 5.6, 3.1, 6.4, 21.7, 10.4, 5.6, 3.1, 6.4, 21.7]
```

Perhatikan bahwa "=" atau ":=" digunakan untuk penugasan. Operator "->" digunakan untuk satuan dalam EMT.

```
>125km -> " miles"
```

```
77.6713990297 miles
```

Operator "<->" untuk penugasan memang menyesatkan, dan bukan ide bagus dari R. Berikut akan membandingkan a dan -4 dalam EMT.

```
>a=2; a<-4
```

```
0
```

Dalam R, "a<-4<3" berfungsi, tetapi "a<-4<-3" tidak. Saya juga memiliki ambiguitas serupa di EMT, tetapi mencoba menghilangkannya satu per satu.

EMT dan R memiliki vektor tipe boolean. Tetapi dalam EMT, angka 0 dan 1 digunakan untuk mewakili false dan true. Dalam R, nilai true dan false masih dapat digunakan dalam aritmetika biasa seperti di EMT.

```
>x<5, %*x
```

```
[0, 0, 1, 0, 0]  
[0, 0, 3.1, 0, 0]
```

EMT menghasilkan kesalahan atau menghasilkan NAN tergantung pada flag "errors".

```
>errors off; 0/0, isNaN(sqrt(-1)), errors on;
```

```
NAN  
1
```

String sama dalam R dan EMT. Keduanya berada dalam locale saat ini, bukan dalam Unicode.

Dalam R, ada paket untuk Unicode. Dalam EMT, string dapat menjadi string Unicode. String Unicode dapat diterjemahkan ke encoding lokal dan sebaliknya. Selain itu, u"..." dapat berisi entitas HTML.

```
>u"\u00e9; Ren\u00e9 Grothmann"
```

© René Grothmann

Berikut mungkin atau mungkin tidak tampil dengan benar pada sistem Anda sebagai A dengan titik dan garis di atasnya. Ini tergantung pada font yang Anda gunakan.

```
>chartoutf([480])
```

Penggabungan string dilakukan dengan "+" atau "|". Ini dapat mencakup angka, yang akan dicetak dalam format saat ini.

```
>"pi = "+pi
```

```
pi = 3.14159265359
```

Pengindeksan

Sebagian besar, ini akan berfungsi seperti di R.

Namun, EMT akan menginterpretasikan indeks negatif dari bagian belakang vektor, sementara R menginterpretasikan x[n] sebagai x tanpa elemen ke-n.

```
>x, x[1:3], x[-2]
```

```
[10.4, 5.6, 3.1, 6.4, 21.7]  
[10.4, 5.6, 3.1]  
6.4
```

Perilaku R dapat dicapai di EMT dengan menggunakan drop().

```
>drop(x, 2)
```

```
[10.4, 3.1, 6.4, 21.7]
```

Vektor logis tidak diperlakukan secara berbeda sebagai indeks di EMT, berbeda dengan R. Anda perlu meng-ekstrak elemen-elemen non-nol terlebih dahulu di EMT.

```
>x, x>5, x[nonzeros(x>5)]
```

```
[10.4, 5.6, 3.1, 6.4, 21.7]
[1, 1, 0, 1, 1]
[10.4, 5.6, 6.4, 21.7]
```

Sama seperti di R, vektor indeks dapat berisi pengulangan.

```
>x[[1,2,2,1]]
```

```
[10.4, 5.6, 5.6, 10.4]
```

Tetapi nama untuk indeks tidak mungkin di EMT. Untuk paket statistik, ini seringkali diperlukan untuk memudahkan akses ke elemen-elemen vektor.

Untuk meniru perilaku ini, kita dapat mendefinisikan fungsi sebagai berikut.

```
>function sel (v,i,s) := v[indexof(s,i)]; ...
>s=["first","second","third","fourth"]; sel(x,[ "first","third"],s)
```

```
Trying to overwrite protected function sel!
Error in:
function sel (v,i,s) := v[indexof(s,i)]; ... ...
^

Trying to overwrite protected function sel!
Error in:
function sel (v,i,s) &= v[indexof(s,i)]; ... ...
^

Trying to overwrite protected function sel!
Error in:
function sel (v,i,s) := v[indexof(s,i)]; ... ...
^
```

```
Trying to overwrite protected function sel!
Error in:
function sel (v,i,s) := v[indexof(s,i)]; ... ...
^

[10.4, 3.1]
```

Tipe Data

EMT memiliki lebih banyak jenis data yang tetap dibandingkan dengan R. Tentu saja, dalam R terdapat vektor yang dapat tumbuh. Anda dapat mengatur vektor numerik kosong v dan memberikan nilai pada elemen v[17]. Hal ini tidak mungkin dilakukan dalam EMT.

Berikut adalah salah satu yang sedikit tidak efisien.

```
>v=[]; for i=1 to 10000; v=v|i; end;
```

EMT sekarang akan membuat vektor dengan v dan i ditambahkan pada tumpukan dan menyalin vektor tersebut kembali ke variabel global v.

Lebih efisien jika Anda telah menentukan vektor sebelumnya.

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Untuk mengubah jenis data dalam EMT, Anda dapat menggunakan fungsi seperti complex().

```
>complex(1:4)
```

```
[ 1+0i , 2+0i , 3+0i , 4+0i ]
```

Konversi ke string hanya mungkin untuk jenis data dasar. Format saat ini digunakan untuk penggabungan string sederhana. Tetapi ada fungsi seperti print() atau frac().

Untuk vektor, Anda dapat dengan mudah menulis fungsi Anda sendiri.

```
>function tostr (v) ...
```

```
s="[";  
loop 1 to length(v);  
  s=s+print(v[#],2,0);  
  if #<length(v) then s=s+","; endif;  
end;  
return s+"]";  
endfunction
```

```
>tostr(linspace(0,1,10))
```

```
[0.00,0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1.00]
```

Untuk berkomunikasi dengan Maxima, ada fungsi convertmxm(), yang juga dapat digunakan untuk memformat vektor untuk output.

```
>convertm xm(1:10)
```

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

Untuk Latex, perintah tex dapat digunakan untuk mendapatkan perintah Latex.

```
>tex(&[1,2,3])
```

```
\left[ 1 , 2 , 3 \right]
```

Faktor dan Tabel

Pada pengantar ke dalam bahasa pemrograman R, terdapat contoh dengan yang disebut faktor.

Berikut adalah daftar wilayah dari 30 negara bagian.

```
>austates = ["tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa", ...
>"qld", "vic", "nsw", "vic", "qld", "qld", "sa", "tas", ...
>"sa", "nt", "wa", "vic", "qld", "nsw", "nsw", "wa", ...
>"sa", "act", "nsw", "vic", "vic", "act"];
```

Misalkan, kita memiliki pendapatan yang sesuai di setiap negara bagian.

```
>incomes = [60, 49, 40, 61, 64, 60, 59, 54, 62, 69, 70, 42, 56, ...
>61, 61, 61, 58, 51, 48, 65, 49, 49, 41, 48, 52, 46, ...
>59, 46, 58, 43];
```

Sekarang, kita ingin menghitung rata-rata pendapatan di wilayah-wilayah tersebut. Sebagai program statistik, R memiliki fungsi `factor()` dan `tapply()` untuk hal ini.

EMT dapat melakukannya dengan mencari indeks wilayah di dalam daftar unik wilayah.

```
>auterr=sort(unique(austates)); f=indexofsorted(auterr,austates)
```

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

Pada titik tersebut, kita dapat menulis fungsi loop kita sendiri untuk melakukan hal-hal hanya untuk satu faktor.

Atau kita bisa meniru fungsi `tapply()` dengan cara berikut.

```
>function map_tappl (i; f$:call, cat, x) ...
u=sort(unique(cat));
f=indexof(u,cat);
return f$(x[nonzeros(f==indexof(u,i))]);
endfunction
```

Ini agak tidak efisien, karena menghitung wilayah-wilayah unik untuk setiap `i`, tetapi ini berfungsi.

```
>tappl(auterr,"mean",austates,incomes)
```

```
[44.5, 57.3333333333, 55.5, 53.6, 55, 60.5, 56, 52.25]
```

Perhatikan bahwa ini berfungsi untuk setiap vektor wilayah.

```
>tappl(["act","nsw"],"mean",austates,incomes)
```

```
[44.5, 57.3333333333]
```

Sekarang, paket statistik EMT mendefinisikan tabel seperti di R. Fungsi `readtable()` dan `writetable()` dapat digunakan untuk input dan output.

Jadi kita dapat mencetak pendapatan rata-rata negara bagian di wilayah dengan cara yang ramah.

```
>writetable(tappl(auterr,"mean",austates,incomes),labc=auterr,wc=7)
```

act	nsw	nt	qld	sa	tas	vic	wa
44.5	57.33	55.5	53.6	55	60.5	56	52.25

Kita juga bisa mencoba meniru perilaku R sepenuhnya.

Faktor-faktor seharusnya jelas disimpan dalam sebuah koleksi dengan jenis dan kategori (negara bagian dan wilayah dalam contoh kita). Untuk EMT, kita menambahkan indeks yang telah dihitung sebelumnya.

```
>function makef (t) ...
```

```
## Factor data
## Returns a collection with data t, unique data, indices.
## See: tapply
u=sort(unique(t));
return ({t,u,indexofsorted(u,t)}};
endfunction
```

```
>statef=makef(austates);
```

Sekarang, elemen ketiga dari koleksi akan berisi indeks.

```
>statef[3]
```

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

Sekarang kita dapat meniru fungsi tapply() dengan cara berikut. Ini akan mengembalikan tabel sebagai koleksi data tabel dan judul kolom.

```
>function tapply (t:vector,tf,f$:call) ...
```

```
## Makes a table of data and factors
## tf : output of makef()
## See: makef
uf=tf[2]; f=tf[3]; x=zeros(length(uf));
for i=1 to length(uf);
  ind=nonzeros(f==i);
  if length(ind)==0 then x[i]=NAN;
  else x[i]=f$(t[ind]);
  endif;
end;
return {{x,uf}};
endfunction
```

Kami tidak menambahkan banyak pemeriksaan tipe di sini. Satu-satunya tindakan pencegahan berkaitan dengan kategori (faktor) tanpa data. Tetapi kita harus memeriksa panjang yang benar dari t dan kebenaran dari koleksi tf.

Tabel ini dapat dicetak sebagai tabel dengan writetable().

```
>writetable(tapply(incomes,statef,"mean"),wc=7)
```

act	nsw	nt	qld	sa	tas	vic	wa
44.5	57.33	55.5	53.6	55	60.5	56	52.25

Arrays

EMT hanya memiliki dua dimensi untuk array. Jenis data ini disebut matriks. Namun, akan mudah menulis fungsi untuk dimensi yang lebih tinggi atau pustaka C untuk ini.

R memiliki lebih dari dua dimensi. Di R, array adalah vektor dengan kolom dimensi.

Dalam EMT, vektor adalah matriks dengan satu baris. Ini dapat diubah menjadi matriks dengan menggunakan redim().

```
>shortformat; X=redim(1:20,4,5)
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Ekstraksi baris dan kolom, atau sub-matriks, mirip dengan di R.

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

2	3
7	8
12	13
17	18

Namun, di R, mungkin untuk mengatur daftar indeks tertentu dari vektor ke suatu nilai. Hal yang sama hanya mungkin dilakukan di EMT dengan menggunakan perulangan.

```
>function setmatrixvalue (M, i, j, v) ...
```

```
loop 1 to max(length(i),length(j),length(v))
  M[i{#},j{#}] = v{#};
end;
endfunction
```

Kami memperlihatkan ini untuk menunjukkan bahwa matriks dikirimkan melalui referensi di EMT. Jika Anda tidak ingin mengubah matriks asli M, Anda perlu menyalinnya dalam fungsi.

```
>setmatrixvalue(X,1:3,3:-1:1,0); X,
```

1	2	0	4	5
6	0	8	9	10
0	12	13	14	15
16	17	18	19	20

Produk luar di EMT hanya dapat dilakukan antara vektor. Ini otomatis karena bahasa matriks. Salah satu vektor perlu menjadi vektor kolom dan yang lainnya vektor baris.

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

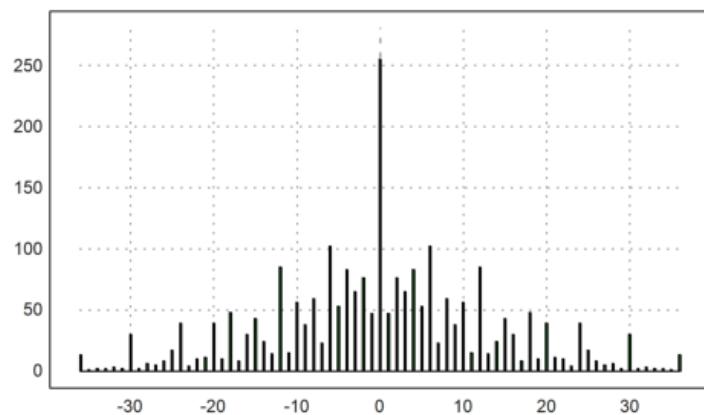
1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Dalam PDF pengantar untuk R, ada contoh yang menghitung distribusi ab-cd untuk a, b, c, d yang dipilih secara acak dari 0 hingga n. Solusi di R adalah membentuk matriks 4 dimensi dan menjalankan fungsi table() di atasnya.

Tentu saja, ini dapat dicapai dengan menggunakan perulangan. Tetapi perulangan tidak efektif di EMT atau R. Di EMT, kita bisa menulis perulangan di C dan itu akan menjadi solusi tercepat.

Namun, kita ingin meniru perilaku R. Untuk ini, kita perlu meratakan perkalian ab dan membuat matriks ab-cd.

```
>a=0:6; b=a'; p=flatten(a*b); q=flatten(p-p'); ...
>u=sort(unique(q)); f=getmultiplicities(u,q); ...
>statplot(u,f,"h"):
```



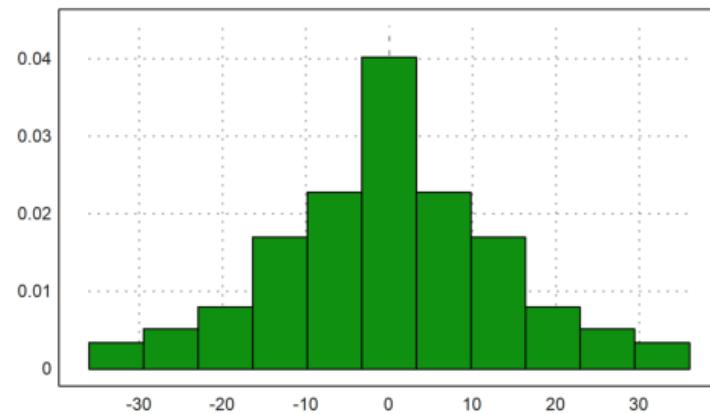
Selain kelipatan tepat, EMT dapat menghitung frekuensi dalam vektor.

```
>getfrequencies(q,-50:10:50)
```

```
[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]
```

Cara paling mudah untuk memplot ini sebagai distribusi adalah sebagai berikut.

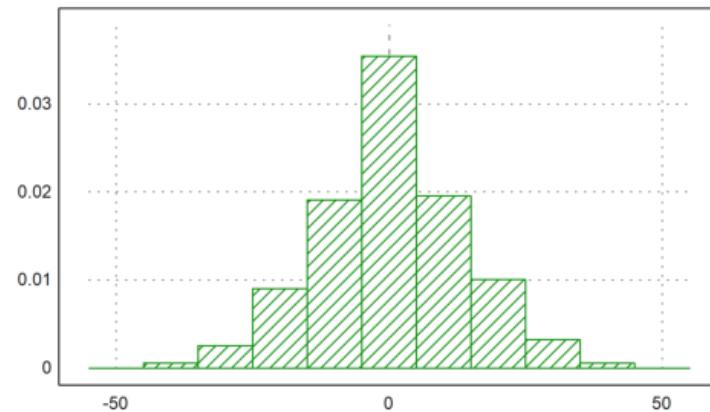
```
>plot2d(q,distribution=11) :
```



Namun, juga mungkin untuk menghitung jumlah dalam interval yang dipilih sebelumnya. Tentu saja, yang berikut menggunakan getfrequencies() secara internal.

Karena fungsi histo() mengembalikan frekuensi, kita perlu menyesuaikan agar integral di bawah grafik batang menjadi 1.

```
>{x,y}=histo(q,v=-55:10:55); y=y/sum(y)/differences(x); ...
>plot2d(x,y,>bar,style="/") :
```



Lists

EMT memiliki dua jenis daftar. Salah satunya adalah daftar global yang dapat diubah, dan yang lainnya adalah tipe daftar yang tidak dapat diubah. Kami tidak peduli tentang daftar global di sini.

Tipe daftar yang tidak dapat diubah disebut koleksi dalam EMT. Ini berperilaku seperti struktur dalam bahasa C, tetapi elemennya hanya diberi nomor dan tidak dinamai.

```
>L={ {"Fred", "Flintstone", 40, [1990, 1992] } }
```

```
Fred  
Flintstone  
40  
[1990, 1992]
```

Saat ini, elemen-elemen tersebut tidak memiliki nama, meskipun nama dapat ditetapkan untuk tujuan khusus. Mereka diakses dengan nomor.

```
>(L[4])[2]
```

```
1992
```

File Input dan Output (Membaca dan Menulis Data)

Anda seringkali ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberi tahu Anda tentang banyak cara untuk mencapai hal ini. Fungsi sederhana adalah writematrix() dan readmatrix().

Mari kita tunjukkan bagaimana cara membaca dan menulis vektor bilangan riil ke dalam file.

```
>a=random(1,100); mean(a), dev(a),
```

```
0.49815  
0.28037
```

Untuk menulis data ke dalam file, kita menggunakan fungsi writematrix().

Karena pengantar ini kemungkinan berada di direktori di mana pengguna tidak memiliki akses tulis, kita menulis data ke direktori home pengguna. Untuk notebook sendiri, ini tidak diperlukan, karena file data akan ditulis ke dalam direktori yang sama.

```
>filename="test.dat";
```

Sekarang kita menulis vektor kolom a' ke dalam file. Ini menghasilkan satu angka di setiap baris file.

```
>writematrix(a',filename);
```

Untuk membaca data, kita menggunakan readmatrix().

```
>a=readmatrix(filename)';
```

Dan hapus file tersebut.

```
>fileremove(filename);  
>mean(a), dev(a),
```

```
0.49815  
0.28037
```

Fungsi writematrix() atau writetable() dapat dikonfigurasi untuk bahasa lain.

Contohnya, jika Anda memiliki sistem Indonesia (titik desimal dengan koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan koma). File "test.csv" berikut seharusnya muncul di folder Anda saat ini.

```
>filename="test.csv"; ...  
>writematrix(random(5,3),file=filename,separator=",");
```

Anda sekarang dapat membuka file ini langsung dengan Excel berbahasa Indonesia.

```
>fileremove(filename);
```

Terkadang kita memiliki string dengan token seperti berikut.

```
>s1:="f m m f m m m f f f m m f"; ...  
>s2:="f f f m m f f";
```

Untuk memecah ini, kita mendefinisikan vektor token.

```
>tok:=[ "f", "m" ]
```

```
f  
m
```

Kemudian kita dapat menghitung jumlah kali setiap token muncul dalam string, dan menempatkan hasilnya ke dalam sebuah tabel.

```
>M:=getmultiplicities(tok,strtokens(s1))_ ...  
> getmultiplicities(tok,strtokens(s2));
```

Tulis tabel dengan header token.

```
>writetable(M,labc=tok,labr=1:2,wc=8)
```

	f	m
1	6	7
2	5	2

Untuk statistik, EMT dapat membaca dan menulis tabel.

```
>file="test.dat"; open(file,"w"); ...
>writeln("A,B,C"); writematrix(random(3,3)); ...
>close();
```

File terlihat seperti ini.

```
>printfile(file)
```

```
A,B,C
0.7003664386138074,0.1875530821001213,0.3262339279660414
0.5926249243193858,0.1522927283984059,0.368140583062521
0.8065535209872989,0.7265910840408142,0.7332619844597152
```

Fungsi readtable() dalam bentuk sederhananya dapat membaca ini dan mengembalikan kumpulan nilai dan baris judul.

```
>L=readtable(file,>list);
```

Kumpulan ini dapat dicetak dengan writetable() ke notebook, atau ke file.

```
>writetable(L,wc=10,dc=5)
```

A	B	C
0.70037	0.18755	0.32623
0.59262	0.15229	0.36814
0.80655	0.72659	0.73326

Matriks nilai adalah elemen pertama dari L. Perhatikan bahwa mean() di EMT menghitung nilai rata-rata dari baris matriks.

```
>mean(L[1])
```

```
0.40472
0.37102
0.75547
```

File CSV

Pertama, mari kita menulis matriks ke dalam sebuah file. Untuk output, kita menghasilkan sebuah file di direktori kerja saat ini.

```
>file="test.csv"; ...
>M=random(3,3); writematrix(M,file);
```

Ini adalah isi dari file tersebut

```
>printfile(file)
```

```
0.8221197733097619,0.821531098722547,0.7771240608094004  
0.8482947121863489,0.3237767724883862,0.6501422353377985  
0.1482301827518109,0.3297459716109594,0.6261901074210923
```

File CSV ini dapat dibuka di sistem berbahasa Inggris ke dalam Excel dengan mengklik dua kali. Jika Anda mendapatkan file tersebut di sistem Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal.

Namun, titik desimal adalah format default untuk EMT juga. Anda dapat membaca sebuah matriks dari file dengan menggunakan fungsi readmatrix().

```
>readmatrix(file)
```

```
0.82212 0.82153 0.77712  
0.84829 0.32378 0.65014  
0.14823 0.32975 0.62619
```

Mungkin untuk menulis beberapa matriks ke dalam satu file. Perintah open() dapat membuka sebuah file untuk penulisan dengan parameter "w". Defaultnya adalah "r" untuk membaca.

```
>open(file,"w"); writematrix(M); writematrix(M'); close();
```

Matriks-matriks tersebut dipisahkan oleh baris kosong. Untuk membaca matriks-matriks tersebut, buka file dan panggil readmatrix() beberapa kali.

```
>open(file); A=readmatrix(); B=readmatrix(); A==B, close();
```

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

Di Excel atau spreadsheet serupa, Anda dapat mengekspor sebuah matriks sebagai CSV (comma separated values). Pada Excel 2007, gunakan "save as" dan "other formats", kemudian pilih "CSV". Pastikan tabel saat ini hanya berisi data yang ingin Anda ekspor.

Berikut adalah contohnya.

```
>printfile("excel-data.csv")
```

```
0;1000;1000  
1;1051,271096;1072,508181  
2;1105,170918;1150,273799  
3;1161,834243;1233,67806  
4;1221,402758;1323,129812  
5;1284,025417;1419,067549  
6;1349,858808;1521,961556
```

```
7;1419,067549;1632,31622  
8;1491,824698;1750,6725  
9;1568,312185;1877,610579  
10;1648,721271;2013,752707
```

Seperti yang dapat Anda lihat, sistem Jerman saya menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubah ini dalam pengaturan sistem atau di Excel, tetapi ini tidak diperlukan untuk membaca matriks ke dalam EMT.

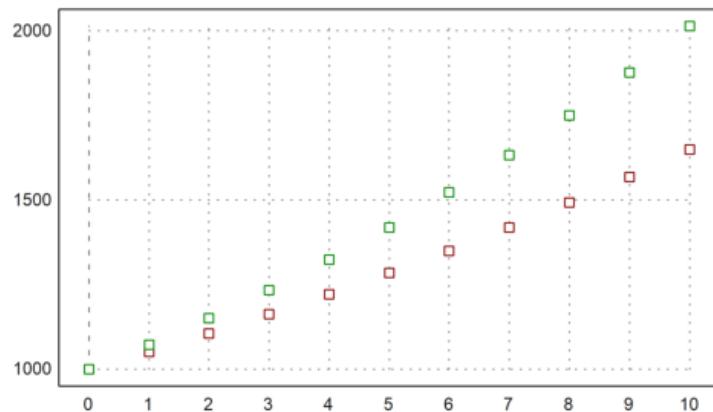
Cara paling mudah untuk membaca ini ke dalam Euler adalah dengan menggunakan `readmatrix()`. Semua koma digantikan oleh titik dengan parameter `>comma`. Untuk CSV berbahasa Inggris, cukup abaikan parameter ini.

```
>M=readmatrix("excel-data.csv",>comma)
```

0	1000	1000
1	1051.3	1072.5
2	1105.2	1150.3
3	1161.8	1233.7
4	1221.4	1323.1
5	1284	1419.1
6	1349.9	1522
7	1419.1	1632.3
8	1491.8	1750.7
9	1568.3	1877.6
10	1648.7	2013.8

Mari kita gambarkan ini.

```
>plot2d(M' [1],M' [2:3],>points,color=[red,green]'):
```



Ada cara lebih dasar untuk membaca data dari sebuah file. Anda dapat membuka file dan membaca angka-angka baris per baris. Fungsi `getvectorline()` akan membaca angka-angka dari sebuah baris data. Secara default, itu mengharapkan titik desimal. Tetapi juga dapat menggunakan koma desimal, jika Anda memanggil `setdecimaldot(",")` sebelum Anda menggunakan fungsi ini.

Fungsi berikut adalah contoh untuk ini. Ini akan berhenti pada akhir file atau baris kosong.

```
>function myload (file) ...
open(file);
M=[];
repeat
  until eof();
  v=getvectorline(3);
  if length(v)>0 then M=M_v; else break; endif;
end;
return M;
close(file);
endfunction
```

```
>myload(file)
```

```
0.82212  0.82153  0.77712
0.84829  0.32378  0.65014
0.14823  0.32975  0.62619
```

Juga mungkin untuk membaca semua angka dalam file tersebut dengan menggunakan getvector().

```
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
```

```
0.82212  0.82153  0.77712
0.84829  0.32378  0.65014
0.14823  0.32975  0.62619
```

Jadi sangat mudah untuk menyimpan sebuah vektor nilai, satu nilai dalam setiap baris, dan membaca kembali vektor ini.

```
>v=random(1000); mean(v)
```

```
0.50303
```

```
>writematrix(v',file); mean(readmatrix(file)')
```

```
0.50303
```

Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Sebagai contoh, kita menulis tabel dengan judul baris dan kolom ke dalam sebuah file.

```
>file="test.tab"; M=random(3,3); ...
>open(file,"w"); ...
>writetable(M,separator=",",labc=["one","two","three"]); ...
>close(); ...
>printfile(file)
```

```

one,two,three
0.09,      0.39,      0.86
0.39,      0.86,      0.71
0.2,       0.02,      0.83

```

Ini dapat diimpor ke dalam Excel.

Untuk membaca file di EMT, kita menggunakan readtable().

```

>{M,headings}=readtable(file,>clabs); ...
>writetable(M,labc=headings)

```

```

one      two      three
0.09    0.39    0.86
0.39    0.86    0.71
0.2     0.02    0.83

```

Menganalisis Sebuah Baris

Anda bahkan bisa mengevaluasi setiap baris secara manual. Misalkan, kita memiliki baris dengan format berikut.

```
>line="2020-11-03,Tue,1'114.05"
```

2020-11-03, Tue, 1'114.05

Pertama, kita dapat melakukan tokenisasi pada baris tersebut.

```
>vt=strtoks(line)
```

2020-11-03
Tue
1'114.05

Kemudian, kita dapat mengevaluasi setiap elemen dari baris tersebut menggunakan evaluasi yang sesuai.

```

>day(vt[1]), ...
>indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])), ...
>strrepl(vt[3], "'", "")()

```

7.3816e+05
2
1114

Dengan menggunakan ekspresi reguler, kita dapat mengekstrak hampir semua informasi dari sebuah baris data.

Anggaplah kita memiliki baris berikut dalam dokumen HTML.

```
>line="<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>"
```

<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>

Untuk mengekstrak ini, kita menggunakan ekspresi reguler yang mencari

- sebuah tanda kurung tutup >,
- setiap string yang tidak mengandung tanda kurung dengan

sub-matching "(...)";

- sebuah tanda kurung buka dan tanda kurung tutup menggunakan solusi terpendek,

- setiap string yang tidak mengandung tanda kurung,
- dan sebuah tanda kurung buka <.

Ekspresi reguler agak sulit untuk dipelajari tetapi sangat kuat.

```
>{pos,s,vt}=strxfind(line,">([^\>]+)<.+?>([^\>]+)<" );
```

Hasilnya adalah posisi kecocokan, string yang cocok, dan vektor string untuk sub-matching.

```
>for k=1:length(vt); vt[k](), end;
```

```
1145.5  
5.6
```

Berikut adalah sebuah fungsi yang membaca semua item numerik antara <td> dan </td>.

```
>function readtd (line) ...
```

```
v=[]; cp=0;  
repeat  
{pos,s,vt}=strxfind(line,"<td.*?>(.+?)</td>",cp);  
until pos==0;  
if length(vt)>0 then v=v|vt[1]; endif;  
cp=pos+strlen(s);  
end;  
return v;  
endfunction
```

```
>readtd(line+"<td>non-numerical</td>")
```

```
1145.45  
5.6  
-4.5  
non-numerical
```

Sebuah situs web atau file dengan URL dapat dibuka di EMT dan dapat dibaca baris per baris.

Pada contoh ini, kita membaca versi terkini dari situs EMT. Kita menggunakan ekspresi reguler untuk mencari "Versi ..." dalam suatu judul.

```
>function readversion () ...
urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html");
repeat
    until urleof();
    s=urlgetline();
    k=strfind(s,"Version ",1);
    if k>0 then substring(s,k,strfind(s,<,k)-1), break; endif;
end;
urlclose();
endfunction

>readversion
```

Version 2022-05-18

Input dan Output Variabel

Anda dapat menulis variabel dalam bentuk definisi Euler ke dalam file atau ke baris perintah.

```
>writevar(pi, "mypi");
mypi = 3.141592653589793;
```

Untuk pengujian, kita menghasilkan sebuah file Euler di direktori kerja EMT.

```
>file="test.e"; ...
>writevar(random(2,2), "M", file); ...
>printfile(file,3)
```

```
M = [ ..
0.5991820585590205, 0.7960280262224293;
0.5167243983231363, 0.2996684599070898];
```

Sekarang kita dapat memuat file tersebut. Ini akan mendefinisikan matriks M.

```
>load(file); show M,
```

```
M =
0.59918   0.79603
0.51672   0.29967
```

Selain itu, jika writevar() digunakan pada suatu variabel, itu akan mencetak definisi variabel dengan nama variabel tersebut.

```
>writevar(M); writevar(inch$)
```

```
M = [ ..  
0.5991820585590205, 0.7960280262224293;  
0.5167243983231363, 0.2996684599070898];  
inch$ = 0.0254;
```

Kita juga dapat membuka file baru atau menambahkan ke file yang sudah ada. Dalam contoh ini, kita menambahkan ke file yang telah dihasilkan sebelumnya.

```
>open(file,"a"); ...  
>writevar(random(2,2),"M1"); ...  
>writevar(random(3,1),"M2"); ...  
>close();  
>load(file); show M1; show M2;
```

```
M1 =  
0.30287 0.15372  
0.7504 0.75401  
M2 =  
0.27213  
0.053211  
0.70249
```

Untuk menghapus file-file tertentu, gunakan fileremove().

```
>fileremove(file);
```

Sebuah vektor baris dalam sebuah file tidak memerlukan koma, jika setiap angka berada di baris baru. Mari kita hasilkan file seperti itu, dengan menulis setiap baris satu per satu menggunakan writeln().

```
>open(file,"w"); writeln("M = ["); ...  
>for i=1 to 5; writeln(""+random()); end; ...  
>writeln("]"); close(); ...  
>printfile(file)
```

```
M = [  
0.344851384551  
0.0807510017715  
0.876519562911  
0.754157709472  
0.688392638934  
];
```

```
>load(file); M
```

```
[0.34485, 0.080751, 0.87652, 0.75416, 0.68839]
```

Latihan

Soal 1

Bilangan-bilangan berikut menyatakan hasil ujian akhir Metode Statistika:

23 60 79 32 57 74 52 70 82 36
80 77 81 95 41 65 92 85 55 76
52 10 64 75 78 25 80 98 81 67
41 71 83 54 64 72 88 62 74 43
60 78 89 76 84 48 84 90 15 79
34 67 17 82 69 74 63 80 85 61

Dengan nilai terendah 10, buat sebaran frekuensinya

Penyelesaian

- Menentukan range
 - range= nilai maksimum-nilai minimum
 - = 98-10
 - = 88
- Menentukan banyak kelas dengan aturan struges.
 - = $1+3,3 \log n$, n banyaknya data
 - = $1+3,3 \log 60$
 - = 6,8679
 - = 7
- Menentukan panjang kelas

$$p = \frac{\text{range}}{\text{banyak kelas}}$$

$$p = \frac{88}{7}$$

$$p = 12,571 = 13$$

Berdasarkan data di atas, nilai minimum 10, nilai maksimum 98, banyak kelas 7, dan panjang kelas 13 maka dapat dibuat tabel distribusi frekuensi dengan batas bawah kelas pertama 10 dan batas atas kelas ketujuh yaitu 100. Sehingga dapat ditentukan tepi bawah kelas pertama yaitu $10-0.5=9.5$ dan tepi atas kelas ketujuh yaitu $100+0.5=100.5$.

```
>r=9.5:13:100.5; v=[3,4,5,8,14,20,6];
>T:=r[1:7]' | r[2:8]' | v'; writetable(T,labc=["TB","TA","Frekuensi"])
```

TB	TA	Frekuensi
9.5	22.5	3
22.5	35.5	4
35.5	48.5	5
48.5	61.5	8
61.5	74.5	14
74.5	87.5	20
87.5	100.5	6

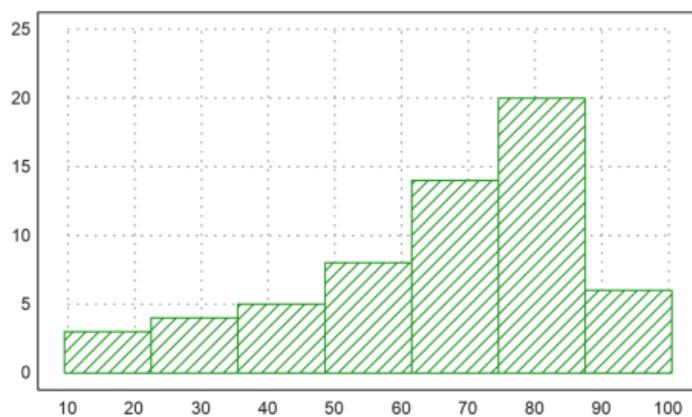
Mencari titik tengah

```
>(T[,1]+T[,2])/2 // Titik tengah interval
```

```
16  
29  
42  
55  
68  
81  
94
```

Berikut plot histogram dari data di atas

```
>plot2d(r,v,a=10,b=100,c=0,d=25,bar=1,style="/") :
```



Soal 2

Banyaknya izin mendirikan bangunan yang dikeluarkan bagi dua belas perusahaan adalah 4, 7, 0, 7, 11, 4, 1, 15, 3, 5, 8, dan 7. Dengan memandang data itu sebagai populasi, hitunglah

- (a) rata-ratanya
- (b) mediannya
- (c) modusnya

Penyelesaian

```
>data=[4,7,0,7,11,4,1,15,3,5,8,7];  
>dataurut=sort(data)
```

```
[0, 1, 3, 4, 4, 5, 7, 7, 7, 8, 11, 15]
```

```
>mean(dataurut) // nilai rata-ratanya
```

6

```
>median(dataurut) // mediannya
```

6

Modusnya, yaitu nilai yang terjadi dengan frekuensi paling tinggi, adalah 7

Soal 3

Nilai-nilai berikut diberikan oleh enam juri dalam suatu pertandingan senam: 7, 5, 9, 7, 8, dan 6. Hitung simpangan baku bagi populasi ini.

Penyelesaian

```
>data=[7,5,9,7,8,6];
>dataurut=sort(data)
```

```
[5, 6, 7, 7, 8, 9]
```

```
>a = mean(dataurut)
```

```
7
```

```
>dev = dataurut-a
```

```
[-2, -1, 0, 0, 1, 2]
```

```
>varians = mean(dev^2)
```

```
1.6667
```

```
>simpanganBaku = sqrt(varians)
```

```
1.291
```

Soal 4

Perbandingan harga dalam bungkus 200 gram di empat toko kelontong yang dipilih secara acak di San Diego menunjukkan kenaikan dari harga bulan sebelumnya 12, 15, 17, dan 20 sen. Hitunglah ragam contoh kenaikan harga kopi ini.

Penyelesaian

```
>data=[12,15,17,20];
>a = mean(data)
```

```
16
```

```
>dev = data-a
```

```
[-4, -1, 1, 4]
```

```
>varians = mean(dev^2)
```

```
8.5
```

Soal 5

Pegawai di sebuah pabrik memberikan sumbangan dalam dolar pada United Fund: 10, 40, 25, 5, 20, 10, 25, 50, 30, 10, 5, 15, 25, 50, 10, 30, 5, 25, 45, dan 15. Tentukan:

- (a) Kuartil
- (b) Persentil
- (c) Desil

Penyelesaian

```
>data=[10,40,25,5,20,10,25,50,30,10,5,15,25,50,10,30,5,25,45,15];  
>dataurut=sort(data)
```

```
[5,   5,   5,   10,   10,   10,   10,   15,   15,   15,   20,   25,   25,   25,   25,   30,  
30,   40,   45,   50,   50]
```

```
>quartiles(data)
```

```
[5,   10,   22.5,   30,   50]
```

Dalam output hitung yang dihasilkan dari 'quartiles(data)' dapat diketahui bahwa nilai Q1(kuartil bawah) = 10 , Q2(kuartil tengah(median)) = 22,5, dan Q3(kuartil atas)= 30. Lalu untuk nilai paling kanan dan paling kiri merupakan minimum dan maximum dari suatu data yang diketahui.