

APLIKOM

Yulianda Fagita Kurniawati

November 2023

BAB 1

Pendahuluan dan Pengenalan Cara Kerja EMT

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

- Bagian numerik, GUI, dan komunikasi dengan Maxima telah dikembangkan oleh R. Grothmann, seorang profesor matematika di Universitas Eichstätt, Jerman. Banyak algoritma numerik dan pustaka software open source yang digunakan di dalamnya.
- Maxima adalah program open source yang matang dan sangat kaya untuk perhitungan simbolik dan aritmatika tak terbatas. Software ini dikelola oleh sekelompok pengembang di internet.
- Beberapa program lain (LaTeX, Povray, Tiny C Compiler, Python) dapat digunakan di Euler untuk memungkinkan perhitungan yang lebih cepat maupun tampilan atau grafik yang lebih baik.

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

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

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

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

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

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

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

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

Judul

Sub-Judul

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

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

maxima: 'integrate(x^3,x) = integrate(x^3,x) + C

http://www.euler-math-toolbox.de

See: http://www.google.de | Google

image: hati.png

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

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

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

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.

```
>x := 25
```

25

```
>x^2
```

625

```
>y := 30
```

30

```
>x * y, sin(y), x^y(x*3)
```

```
750
-0.988031624093
Unexpected "(".
Index () not allowed in strict mode!
In Euler files, use relax to avoid this.
Error in:
x * y, sin(y), x^y(x*3) ...
^
```

```
>x * y, tan(y), x^y*6
```

```
750
-6.40533119665
5.20417042793e+42
```

```
>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 $^\circ$. 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.
-

Satuan

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

>1km

1000

>&diff(log(x)/x^2, x)

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

>&diff(log(x)/x^3, x+2)

Maxima said:

diff: second argument must be a variable; found x+2
-- an error. To debug this try: debugmode(true);

Error in:

&diff(log(x)/x^3, x+2) ...
 ^

```
>&diff(log(x^2)/x^2)
```

$$\left(\frac{2}{x^3} - \frac{4 \log(x)}{x^3} \right) \text{de1}(x)$$

```
>2miles -> yard
```

3520

```
>2kg -> gr * 2gr
```

Variable gr not found!

Error in:

2kg -> gr * 2gr ...
 ^

```
>3mm * 25cm -> mm
```

0.75

```
>0.75year -> days
```

273.931875

```
>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
```

10

	BAB 1. PENDAHULUAN DAN PENGENALAN CARA KERJA EMT
r	1.25
x	25
Y	30
a	4
b	15
c	2
D	1.73655549868123

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

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

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

```
>listvar m$
```

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

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

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

Variable D not found!

Error in:

D ...
 ^

Menampilkan Panduan

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

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

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

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

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

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

```
>random(10)
```

```
[0.270906, 0.704419, 0.217693, 0.445363, 0.308411, 0.914541,  
0.193585, 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 kompleks 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
```

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 dikерjakan oleh Maxima.

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

$$(b + a)^2$$

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

$$\begin{aligned} &b^2 + 2ab + a^2 \\ &(x + 2)(x + 3) \end{aligned}$$

```
>&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
```

$$\begin{array}{r}
 2 \quad 2 \\
 a - b \\
 \hline
 b + a
 \end{array}$$

$$a - b$$

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

3628800

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

3628800

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

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

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

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

$$\begin{array}{c}
 3 \quad 3 \\
 2 \quad 5
 \end{array}$$

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

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

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
```

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

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

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

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

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

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

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
```

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

$$\frac{b^2 + (2a - 1)b + a^2 - a + 1}{b^2 + 2a^2 - a + 1}$$

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

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

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

$$\frac{2x^3 - 3x^2 + 6x}{6}$$

Tampilan Matematika Simbolik dengan LaTeX

Anda dapat menampilkan hasil perhitungan simbolik secara lebih bagus menggunakan LaTeX. Untuk melakukan hal ini, tambahkan tanda dolar (\$) di depan tanda & pada setiap perintah Maxima.

Perhatikan, hal ini hanya dapat menghasilkan tampilan yang diinginkan apabila komputer Anda sudah terpasang software LaTeX.

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

$$(b+a)^2$$

```
>$&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.

```
>mean([10, 8, 7, 6, 4, 7, 7, 12])
```

7.625

```
>median([5,5,9,8,3,12,4,10,6,6,8,10,4,4,2])
```

6

```
>gauss([3 4; 2 3], [4; 5])
```

```
Need , or ; between matrix elements in strict mode
Error in:
gauss([3 4; 2 3], [4; 5]) ...
^
```

```
>gauss([7 4; 2 3], [15; 10])
```

Need , or ; between matrix elements in strict mode

Error in:

```
gauss([7 4; 2 3], [15; 10]) ...  
^
```

```
>gauss([7 4; 2 3]; [15; 10])
```

Need , or ; between matrix elements in strict mode

Error in:

```
gauss([7 4; 2 3]; [15; 10]) ...  
^
```

BAB 2

EMT untuk Perhitungan Aljabar

2.1 EMT untuk Perhitungan Aljabar

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

- Membaca dengan 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 yang diikuti dengan titik koma ";" atau koma ",". Titik koma mencegah pencetakan hasil. Koma setelah perintah terakhir dapat dihilangkan.

Baris perintah berikut ini hanya akan mencetak hasil dari ekspresi, bukan penugasan atau perintah format.

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

16.7551608191

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

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

50.2654824574

100.530964915

Baris perintah dieksekusi sesuai urutan pengguna menekan tombol return. Jadi, Anda mendapatkan nilai baru setiap kali Anda mengeksekusi baris kedua.

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

0.540302305868

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

0.857553215846

Jika dua baris dihubungkan dengan "...", kedua baris tersebut akan selalu dieksekusi secara bersamaan.

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

1.41666666667
1.41421568627
1.41421356237

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

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

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

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

81

Euler mendukung perulangan dalam baris perintah, selama perulangan tersebut masuk ke dalam satu baris tunggal atau beberapa baris. Dalam program, tentu saja pembatasan ini tidak berlaku. Untuk informasi lebih lanjut, baca pengantar berikut ini.

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

1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237

Tidak masalah untuk 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 bisa digunakan.

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

Thought so!

Ketika Anda menjalankan perintah, kursor dapat berada di posisi mana pun dalam 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.

Apabila Anda menggerakkan kursor di sepanjang garis, pasangan tanda kurung atau tanda kurung pembuka dan penutup akan menyalin terang. Juga, perhatikan baris status. Setelah tanda kurung pembuka dari fungsi `sqrt()`, baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol return.

```
>sqrt(sin(10°)/cos(20°))
```

0.429875017772

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

Anda dapat mengklik dua kali pada perintah mana pun untuk membuka bantuan untuk perintah ini. Coba klik dua kali `exp` com- mand 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. Selain itu, Anda dapat menyalin tanda kurung yang disorot.

2.1. EMT UNTUK PERHITUNGAN ALJABAR

25

Euler mengetahui fungsi matematika yang biasa. Seperti yang telah Anda lihat di atas, fungsi trigonometri bekerja dalam radian atau derajat. Untuk mengonversi ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilai, atau gunakan fungsi rad(x). Fungsi akar kuadrat disebut sqrt dalam Euler. Tentu saja, $x^{(1/2)}$ juga dapat digunakan.

Untuk mengatur variabel, gunakan "=" atau ":=". Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir. Spasi tidak menjadi masalah. Tetapi spasi di antara perintah sangat diharapkan.

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma menekan output perintah. Pada akhir baris perintah, "," diasumsikan sebagai "", jika ";" tidak ada.

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

30.65625

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

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

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

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

And aharus memasukkannya kedalam baris

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

23.2671801626

Letakkan tanda kurung di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyorot ekspresi yang diselesaikan oleh tanda kurung penutup. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi. Hasil komputasi ini adalah angka floating point. Secara default dicetak dengan akurasi sekitar 12 digit. Pada baris perintah berikut ini, kita juga mempelajari bagaimana kita dapat 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, ekspresi tersebut harus mengandung tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, mengatur tanda kurung adalah ide yang bagus. Perhatikan bahwa EMT menampilkan tanda kurung pembuka dan penutup saat mengedit baris perintah.

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

14.4978445072

Operator numerik Euler meliputi

+ unary atau operator plus

- unary atau operator minus

*, /

. produk matriks

pangkat a^b untuk a positif atau bilangan bulat b (a**b juga dapat digunakan)

n! operator faktorial

dan masih banyak lagi.

Berikut ini 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
```

2.1. EMT UNTUK PERHITUNGAN ALJABAR

beberapa perintah memiliki alias seperti ln untuk log.

27

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

```
2  
0.5
```

```
>sin(30°)
```

```
0.5
```

Pastikan untuk menggunakan tanda kurung (tanda kurung bulat), apabila 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 utama dalam Euler adalah bilangan real. Bilangan 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)
```

```
5.555555555554*16^-1
```

String

String pada 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 dikonversi menjadi string dalam kasus tersebut.

```
>"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 cm².

Fungsi cetak juga mengonversi angka ke string. Fungsi ini dapat mengambil sejumlah digit dan sejumlah tempat (0 untuk output padat), dan secara optimal satu unit.

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

Golden Ratio : 1.61803

Ada string khusus tidak ada, yang tidak mencetak. Dikembalikan oleh beberapa fungsi, ketika hasilnya tidak penting. (Dikembalikan secara otomatis, jika fungsi tidak memiliki pernyataan pengembalian).

```
>none
```

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

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

1234.5

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

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

affe
charlie
bravo

2.1. EMT UNTUK PERHITUNGAN ALJABAR

29

Vektor string kosong dilambangkan dengan [none]. Vektor string dapat digabungkan.

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

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

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

String Unicode dapat digabungkan seperti string lainnya.

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

```
= 45°
```

I

Dalam komentar, entitas yang sama seperti , dll. dapat digunakan. Ini mungkin merupakan alternatif yang cepat untuk Latex. (Detail lebih lanjut tentang komentar di bawah). Ada beberapa fungsi untuk membuat atau menganalisis string unicode. Fungsi strtochar() akan mengenali string Unicode, dan menerjemahkannya dengan benar.

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

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

Hasilnya adalah sebuah vektor angka Unicode. Fungsi kebalikannya adalah chartoutf().

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

```
Ü is a German letter
```

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

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

```
We have =.
```

```
>u"hnliches"
```

Ähnliches

Nilai Boolean

Nilai Boolean direpresentasikan dengan 1 = benar atau 0 = salah dalam 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 sebuah vektor. Pada contoh, kita menggunakan kondisional `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]
```

Format Ouput

Format output default EMT mencetak 12 digit. Untuk memastikan bahwa kita melihat format default, kita atur ulang formatnya.

```
>defformat; pi
```

```
3.14159265359
```

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

```
>longest pi
```

```
3.141592653589793
```

Berikut ini adalah representasi heksadesimal internal dari angka ganda

```
>printhex(pi)
```

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

Format output dapat diubah secara permanen dengan perintah format.

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

```
0.33333  
3.14159  
0.84147
```

Format defaultnya adalah(12)

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

0.333333333333

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

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

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

Format default untuk skalar adalah format(12). Tetapi ini dapat diubah

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

3.1416

Fungsi "longestformat" juga menetapkan format skalar.

```
>longestformat; pi
```

3.141592653589793

Sebagai referensi, berikut ini daftar format output yang paling penting.

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

Akurasi internal EMT adalah sekitar 16 tempat desimal, yang merupakan standar IEEE. Angka disimpan dalam format internal ini. Tetapi format output EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

3.141592653589793

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

3.14159

Standarnya adalah deformat().

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

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

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

4.934802200544679

Ada juga operator singkat untuk mencetak hasil dalam format pecahan. Kami sudah menggunakan di atas.

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

25/12

Karena format internal menggunakan cara biner untuk menyimpan angka, maka nilai 0,1 tidak akan terwakili dengan tepat. Kesalahan bertambah sedikit, seperti yang Anda lihat dalam perhitungan berikut ini.

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

-1.110223024625157e-16

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

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

0

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

Variabel global dapat digunakan dalam evaluasi.

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

12.56637061435917

Parameter ditetapkan ke x, y, dan z dalam urutan tersebut. Parameter tambahan dapat ditambahkan dengan 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 menyatakan bahwa koleksi panggilan (dibahas di tempat lain) dapat berisi ekspresi. Jadi kita dapat membuat contoh di atas sebagai berikut.

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

45

2.1. EMT UNTUK PERHITUNGAN ALJABAR

35

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

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

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

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

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

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

Hal ini memungkinkan untuk memanipulasi ekspresi dalam variabel lain untuk fungsi EMT yang membutuhkan ekspresi dalam "x". Cara paling dasar untuk mendefinisikan fungsi sederhana adalah dengan menyimpan rumusnya dalam ekspresi simbolik atau numerik. Jika variabel utama 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 berbentuk simbolik. Hal ini diperlukan, jika ekspresi mengandung fungsi-fungsi, yang hanya dikenal di kernel numerik, bukan di Maxima.

Matematika Simbolik

EMT melakukan matematika simbolik dengan bantuan Maxima. Untuk detailnya, mulailah dengan tutorial berikut ini, atau telusuri referensi untuk Maxima. Para ahli dalam Maxima harus mencatat bahwa ada perbedaan dalam sintaks antara sintaks asli Maxima dan sintaks default ekspresi simbolik dalam EMT. Matematika simbolik diintegrasikan dengan mulus ke dalam Euler dengan &. Setiap ekspresi yang dimulai dengan & adalah ekspresi simbolik. Ekspresi ini dievaluasi dan dicetak oleh Maxima. Pertama-tama, Maxima memiliki aritmatika "tak terbatas" yang dapat menangani angka yang sangat besar.

```
>$& 44 !
```

26582715747884487680436258110146158903196385280000000000

dengan cara ini, anda dapat menghitung hasil yang besar secara tepat. mari kita hitung

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

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

2481256778

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

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

2481256778

2.1. EMT UNTUK PERHITUNGAN ALJABAR

37

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik dua kali pada fungsi tersebut. Sebagai contoh, coba klik dua kali pada "&binomial" di baris perintah sebelumnya. Ini akan membuka dokumentasi Maxima yang disediakan oleh pembuat program tersebut.

Anda akan mengetahui bahwa fungsi-fungsi berikut juga dapat digunakan.

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

```
>$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 dapat menggunakan solusi dari sebuah persamaan dalam persamaan lain. Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Alasan untuk ini adalah bendera simbolis khusus dalam string.

Seperti yang telah Anda lihat pada contoh sebelumnya dan contoh berikut, jika Anda telah menginstal LaTeX, Anda dapat mencetak ekspresi simbolik dengan Latex. Jika tidak, perintah berikut ini 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 tidak memiliki LaTeX yang terinstal.

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

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

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

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

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

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

$$\begin{aligned} &x^4 + 4x^3 + 6x^2 + 4x + 1 \\ &4(x+1)^3 \end{aligned}$$

Sekali lagi, % mengacu pada hasil sebelumnya. Untuk mempermudah, kita menyimpan solusi ke dalam sebuah 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))
```

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

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

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

```
>::: 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 adalah seorang ahli dalam Maxima, Anda mungkin ingin menggunakan sintaks asli Maxima. Anda dapat melakukan ini dengan ":::".

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

$$\begin{matrix} 2 \\ g \end{matrix}$$

```
>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 pada perintah berikut ini, sisi kanan dari &= dievaluasi sebelum penugasan 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 evaluasi ekspresi dengan nilai variabel tertentu, Anda dapat menggunakan operator "with".

Baris perintah berikut ini juga menunjukkan bahwa Maxima dapat mengevaluasi sebuah ekspresi secara numerik dengan float().

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

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

$$2.20079141499189e+7$$

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

$$x (x^2 + 6x + 6) e^x$$

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

```
>tex(fx)
```

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

```
>fx(0.5)
```

0.206090158838

Dalam ekspresi simbolik, hal ini tidak dapat dilakukan, karena Maxima tidak mendukungnya. Sebagai gantinya, gunakan sintaks "with" (bentuk yang lebih baik dari perintah at(...) pada Maxima)

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

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

Penugasan ini juga bisa bersifat simbolis

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

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

Perintah solve menyelesaikan ekspresi simbolik untuk sebuah variabel di Maxima. Hasilnya adalah sebuah 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 "solve" numerik di Euler, yang membutuhkan nilai awal, dan secara opsional nilai target

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

1.56155281281

42

BAB 2. EMT UNTUK PERHITUNGAN ALJABAR

Nilai numerik dari solusi simbolik dapat dihitung dengan evaluasi hasil simbolik. Euler akan membaca penugasan $x = \text{dst}$. Jika Anda tidak membutuhkan hasil numerik untuk perhitungan lebih lanjut, Anda juga bisa 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 sebuah simbol penyelesaian yang spesifik, dapat menggunakan "with" dan sebuah index.

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

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

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

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]
```

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

2

Ekspresi simbolik dapat memiliki bendera, yang menunjukkan perlakuan khusus di Maxima. Beberapa flag dapat digunakan sebagai perintah juga, namun ada juga yang tidak. Bendera ditambahkan dengan "!" (bentuk yang lebih baik dari "ev(...,flags)")

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

```
> $& 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". Fungsi dapat berupa fungsi satu baris atau fungsi multibaris. Fungsi satu baris dapat berupa numerik atau simbolik.

Fungsi satu baris numerik didefinisikan dengan ":="

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

Sebagai gambaran umum, kami menunjukkan semua definisi yang mungkin untuk fungsi satu baris. Sebuah fungsi dapat dievaluasi seperti halnya fungsi Euler bawaan.

```
> f(2)
```

4.472135955

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

```
> 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. Alih-alih ekspresi, kita hanya perlu memberikan nama fungsi. Berbeda dengan ekspresi simbolik atau numerik, nama fungsi harus disediakan dalam bentuk string

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

0.786151377757

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

Anda masih dapat memanggil fungsi bawaan sebagai "...", jika itu adalah fungsi dalam inti Euler.

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

0.707106781187

Sebaiknya kita hilangkan definisi ulang tentang sin ini.

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

0.707106781187

Parameter Default

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

16

Mengurnya akan menimpa nilai default.

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

80

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

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

16

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

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

24

Tetapi, parameter yang ditetapkan akan mengesampingkan nilai global. Jika argumen tidak terdapat dalam daftar parameter yang telah ditentukan sebelumnya, argumen tersebut harus dideklarasikan dengan ":="!

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

20

Fungsi simbolik didefinisikan dengan "&=". Fungsi-fungsi ini didefinisikan dalam Euler dan Maxima, dan dapat digunakan di kedua bahasa tersebut. Ekspresi pendefinisan 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
```

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

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

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

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

$$178.635099908$$

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

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

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

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

$$0.703467422498$$

```
> 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(%)
```

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

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

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

$$625$$

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

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

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

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

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

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

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

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

$$16x^7 + 64x^6 + 24x^5 - 120x^4 - 15x^3 + 102x^2 - 52x + 8$$

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

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

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

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

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

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

$$x^3 - x$$

Dengan := fungsi tersebut berupa angka. Contoh yang baik adalah integral pasti seperti

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

yang tidak dapat dievaluasi secara simbolis.

Jika kita mendefinisikan ulang fungsi dengan kata kunci "map", fungsi ini dapat digunakan untuk vektor x. Secara internal, fungsi ini dipanggil untuk semua nilai x satu kali, dan hasilnya disimpan dalam sebuah vektor.

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

Sekarang, fungsi ini dapat dipanggil dengan atau tanpa parameter "base".

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

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

```
2
```

Sering kali, kita ingin menggunakan fungsi untuk vektor di satu tempat, dan untuk masing-masing elemen 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)
```

$$b^2 - a b + b + a^2$$

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

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

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

```
17
```

Ada juga fungsi yang murni simbolis, yang tidak dapat digunakan secara numerik

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

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

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

$$y^4 - 6x^2y^2 + x^4$$

0

Tetapi tentu saja, semua itu bisa digunakan dalam ekspresi simbolis atau dalam definisi fungsi simbolis.

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

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

Untuk meringkas

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

Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolis.

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

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

1.41421356237

Hal ini juga bisa digunakan untuk ekspresi simbolis.

Perhatikan fungsi berikut ini

```
> $&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, di mana 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 sebuah ekspresi simbolik dalam bentuk simbolik mengembalikan sebuah daf- tar solusi. Kami menggunakan pemecah simbolik 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 sebuah 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])}
```

$$\frac{(\sqrt{5} - 1)^2}{4}$$

0

Menyelesaikan sistem persamaan secara simbolik dapat dilakukan dengan vektor persamaan dan pemecah simbolik solve(). Jawabannya adalah sebuah daftar dari 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. Tetapi 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 mengoper parameter tambahan ke f() adalah dengan menggunakan sebuah daftar yang berisi nama fungsi dan parameternya (cara lainnya adalah dengan menggunakan parameter titik koma).

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

2.54116291558

Hal ini juga dapat dilakukan dengan ekspresi. Namun, elemen daftar bernama harus digunakan. (Lebih lanjut tentang daftar dalam 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\fourier_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])
```

$$\begin{aligned} & [6 < x, x < 8, y < -11] \text{ or } [8 < x, y < -11] \\ & \text{or } [x < 8, 13 < y] \text{ or } [x = y, 13 < y] \text{ or } [8 < x, x < y, 13 < y] \\ & \text{or } [y < x, 13 < y] \end{aligned}$$

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

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

Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler. Vektor dan matriks dimasukkan dengan tanda kurung siku, elemen dipisahkan dengan koma, baris dipisahkan dengan titik koma

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

1
3

2
4

Hasil kali matriks dilambangkan dengan sebuah titik

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

3

4

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

[3, 4]

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

-2

1

1.5

-0.5

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

11

25

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

1

0

0

1

Poin utama dari bahasa matriks adalah bahwa semua fungsi dan operator bekerja elemen demi elemen.

```
>A.A
```

7

10

15

22

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

1

4

9

16

>A.A.A

$$\begin{array}{r} 37 \\ 81 \end{array} \quad \begin{array}{r} 54 \\ 118 \end{array}$$

>power(A, 3) //perpangkatan matriks

$$\begin{array}{r} 37 \\ 81 \end{array} \quad \begin{array}{r} 54 \\ 118 \end{array}$$

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

$$\begin{array}{r} 1 \\ 1 \end{array} \quad \begin{array}{r} 1 \\ 1 \end{array}$$

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

$$\begin{array}{r} 0.333333 \\ 0.75 \end{array} \quad \begin{array}{r} 0.666667 \\ 1 \end{array}$$

>A\b // hasil kali invers A dan b, A^{-1}b

$$\begin{array}{r} -2 \\ 2.5 \end{array}$$

>inv(A).b

$$\begin{array}{r} -2 \\ 2.5 \end{array}$$

>A\A //A^{-1}A

$$\begin{array}{r} 1 \\ 0 \end{array} \quad \begin{array}{r} 0 \\ 1 \end{array}$$

```
>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, tetapi perkalian elemen demi elemen. Hal yang sama 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 akan diperluas dengan cara alami.

```
>2*A
```

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

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

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

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

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

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

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

2.1. EMT UNTUK PERHITUNGAN ALJABAR

59

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

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

1	2
1	2

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

1	4
3	8

Hal ini juga berlaku untuk dua vektor di mana satu vektor adalah vektor baris dan yang lainnya adalah vektor kolom. Kami 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 sebuah tabel nilai.

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

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

Sekali lagi, ingatlah bahwa ini bukan produk matriks!

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

55

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

55

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

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

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

Sebagai contoh, 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, di mana 1 berarti benar

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

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

Dari vektor seperti itu, "nonzeros" memilih elemen yang bukan nol. Dalam hal ini, kita mendapatkan indeks semua elemen yang lebih besar dari 50.

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

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

Tentu saja, kita dapat menggunakan vektor indeks ini untuk mendapatkan nilai 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 dari angka 1 sampai 1000, yaitu 5 modulo 11 dan 3 modulo 13.

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

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

2.1. EMT UNTUK PERHITUNGAN ALJABAR

61

EMT tidak sepenuhnya efektif untuk komputasi bilangan bulat. EMT menggunakan floating point presisi ganda secara internal. Akan tetapi, hal ini sering kali sangat berguna. Kita dapat memeriksa bilangan prima. Mari kita cari tahu, berapa banyak kuadrat ditambah 1 yang merupakan bilangan prima.

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

112

Fungsi nonzeros() hanya bekerja 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 menetapkan elemen ke suatu nilai.

```
>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-elemen pada indeks ke entri-entri dari beberapa matriks lain.

```
>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-elemen dalam vektor.

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

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

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

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

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

Kita bisa menggunakan ini untuk mengekstrak nilai maksimal dalam setiap baris.

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

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

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

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

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

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

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

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

2.1. EMT UNTUK PERHITUNGAN ALJABAR

63

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

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

1	2	3
1	2	3

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

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

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

Jika keduanya tidak memiliki jumlah baris yang sama, matriks yang lebih pendek diisi dengan 0. Ada pengecualian untuk aturan ini. Bilangan real yang dilampirkan pada matriks akan digunakan sebagai kolom yang diisi dengan bilangan real tersebut.

```
>A | 1
```

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

Ini mungkin untuk membuat matriks vektor baris dan kolom.

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

1	2	3
1	2	3

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

1	1
2	2
3	3

Tujuan utama dari hal ini adalah untuk menginterpretasikan vektor ekspresi untuk vektor kolom

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

1	1
2	4
3	9

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut ini

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

2	
4	
[2,	4]
4	

Untuk vektor, ada length().

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

9

Ada banyak fungsi lain yang menghasilkan matriks.

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

1	1
1	1

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

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

[6, 6, 6, 6, 6]

Matriks angka acak juga dapat dihasilkan dengan acak (distribusi seragam) atau normal (distribusi Gauß)

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

0.66566	0.831835
0.977	0.544258

Berikut ini adalah fungsi lain yang berguna, yang merestrukturisasi elemen-elemen matriks menjadi matriks lain

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

1	2	3
4	5	6
7	8	9

Dengan fungsi berikut, kita dapat menggunakan fungsi ini dan fungsi dup untuk menulis fungsi rep(), yang mengulang sebuah 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-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() membalik urutan baris atau kolom dari sebuah matriks. Misalnya, 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 menghapus elemen dengan indeks di i dari vektor v

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

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

perhatikan bahwa vektor i dalam drop(v,i) merujuk pada indeks elemen dalam v, bukan nilai elemen. jika anda ingin menghapus elemen, anda harus menemukan elemen-elemen tersebut terlebih dahulu. fungsi indexof(v,x) dapat digunakan untuk menemukan elemen x dalam vektor terurut 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 menyertakan indeks di luar jangkauan (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, kami menetapkan diagonal bawah (-1) ke 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 kita tidak mengubah matriks A. Kita mendapatkan sebuah matriks baru sebagai hasil dari setdiag().

Berikut adalah sebuah fungsi yang mengembalikan sebuah 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 sebuah matriks juga dapat diekstrak dari matriks. Untuk mendemonstrasikan hal ini, kami merestrukturisasi vektor 1:9 menjadi matriks 3x3.

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

1	2	3
4	5	6
7	8	9

Sekarang, kita bisa mengekstrak diagonal.

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

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

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

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

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

Hampir semua fungsi dalam Euler juga dapat digunakan untuk input matriks dan vektor, jika hal ini masuk akal.

Sebagai contoh, 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 sebuah fungsi (alternatif lainnya 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 dari 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. Misalkan, vektor kolom dikalikan vektor baris akan diperluas menjadi matriks, jika sebuah operator diterapkan. Berikut ini, `v'` adalah vektor yang ditransposisikan (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

2.1. EMT UNTUK PERHITUNGAN ALJABAR

69

Perhatikan, bahwa ini sangat berbeda dengan hasil kali matriks. Hasil kali matriks dilambangkan dengan sebuah titik "." dalam 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 . menyatakan perkalian matriks, dan A' menyatakan transpose. Matriks 1x1 dapat digunakan seperti halnya bilangan real.

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

5

25

Untuk mentransposisi matriks, kita menggunakan apostrof.

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

1

2

3

4

Jadi kita dapat menghitung matriks A dikali vektor b.

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

30

70

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

```
>v' . v
```

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

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

```
>v . v'
```

30

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

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

30

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ini adalah ringkasan dari aturan-aturan tersebut.

- Fungsi yang diterapkan ke vektor atau matriks diterapkan ke setiap elemen.
- Operator yang beroperasi pada dua matriks dengan ukuran yang sama diterapkan secara berpasangan pada elemenelemen matriks.
- Jika kedua matriks memiliki dimensi yang berbeda, keduanya diperluas dengan cara yang masuk akal, sehingga keduanya memiliki ukuran yang sama.

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

Berikut ini adalah kasus sederhana dengan operator ^.

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

[1, 4, 9]

Ini adalah kasus yang lebih rumit. Vektor baris dikalikan vektor kolom memperluas keduanya dengan menduplikasi.

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

1	2	3
2	4	6
3	6	9

2.1. EMT UNTUK PERHITUNGAN ALJABAR

71

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan produk *!

```
>v.v'
```

14

Ada banyak fungsi untuk matriks. Kami memberikan daftar singkat. Anda harus membaca dokumentasi untuk informasi lebih lanjut mengenai perintah-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 dari sebuah 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 dari matriks.

```
>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

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

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

2

3

5

6

8

9

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

```
>A{4}
```

4

Sebuah matriks juga dapat diratakan, dengan menggunakan fungsi redim(). Hal ini diimplementasikan dalam fungsi flatten().

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

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

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

```
>deffformat; 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.

Pada contoh berikut, kita menghitung $t[j]^i$ untuk i dari 1 hingga n . Kita mendapatkan sebuah matriks, di mana setiap baris adalah tabel dari t^i untuk satu i . Dengan kata lain, matriks tersebut memiliki elemen elemen

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

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

Integrasi numerik integrate() hanya bekerja untuk batas interval skalar. Jadi kita perlu membuat vektornya.

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

Kata kunci "map" membuat vektor fungsi. Fungsi ini sekarang akan bekerja untuk vektor angka.

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

[0, 2.05045, 13.7251, 113.336, 1241.03]

Sub Matriks dan Elemen Matriks

Untuk mengakses elemen matriks, gunakan notasi kurung

```
>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 lengkap dari sebuah matriks

```
>A[2]
```

[4, 5, 6]

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

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

2

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

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

[4, 5, 6]

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

Di sini kita menginginkan baris pertama dan kedua dari 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, tetapi menghitung versi susunan ulang dari A.

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

7	8	9
4	5	6
1	2	3

Trik indeks juga dapat digunakan pada 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

Sebagai alternatif, 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-elemen dari A dengan memberikan sebuah submatriks dari A ke suatu nilai. Hal ini sebenarnya mengubah matriks A yang tersimpan.

>A[1, 1]=4

4	2	3
4	5	6
7	8	9

Kita juga dapat menetapkan nilai pada deretan A.

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

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

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

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

5	6	-1
7	8	6
7	8	9

Selain itu, beberapa jalan pintas diperbolehkan

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

0	0	-1
0	0	6
7	8	9

Peringatan: Indeks di luar batas akan mengembalikan matriks kosong, atau pesan kesalahan, tergantung pada pengaturan sistem. Standarnya adalah pesan kesalahan. Namun, ingatlah bahwa indeks negatif dapat digunakan untuk mengakses elemen-elemen matriks yang dihitung dari akhir

```
>A[4]
```

Row index 4 out of bounds!

Error in:

A[4] ...
 ^

Menyortir dan Mengacak

Fungsi `mengurutkan()` mengurutkan vektor baris

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

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

Sering kali diperlukan untuk mengetahui indeks vektor yang telah diurutkan dalam vektor aslinya. Hal ini dapat digunakan untuk menyusun ulang vektor lain dengan cara yang sama.

Mari kita acak sebuah vektor.

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

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

Indeks berisi urutan v yang tepat.

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

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

Hal ini juga berlaku 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 entri ganda agak acak

```
>ind
```

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

Fungsi unik mengembalikan daftar terurut dari elemen unik vektor.

```
>unique(s)
```

```
a  
aa  
d  
e
```

Aljabar Linier

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

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

$$\begin{array}{r} -4 \\ 4.5 \end{array}$$

Sebagai contoh lain, kita membuat matriks 200x200 dan jumlah barisnya. Kemudian kita selesaikan $Ax = b$ dengan menggunakan matriks kebalikannya. Kita mengukur kesalahan sebagai deviasi maksimal dari semua elemen dari 1, yang tentu saja merupakan solusi yang benar.

```
>A=normal(200,200); b=sum(A); longest totalmax(abs(inv(A).b-1))
```

$$2.677857935395878e-13$$

Jika sistem tidak memiliki solusi, pencocokan linier meminimalkan norma kesalahan $Ax-b$

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

$$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

Determinan dari matriks ini adalah 0

```
>det(A)
```

$$0$$

Matriks Simbolik

Maxima memiliki matriks simbolik. Tentu saja, Maxima dapat digunakan untuk masalah aljabar linier sederhana. Kita bisa mendefinisikan matriks untuk Euler dan Maxima dengan `&:=`, lalu menggunakan dalam ekspresi simbolik.

Bentuk [...] yang biasa untuk mendefinisikan matriks dapat digunakan dalam 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(%)
```

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

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

```
>$&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)
```

$$(1 - x) (2 - x) - a b$$

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

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

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

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

2.1. EMT UNTUK PERHITUNGAN ALJABAR

81

Untuk mengekstrak vektor eigen tertentu, diperlukan pengindeksan yang cermat

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

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

$$[1, - 1]$$

Matriks simbolik dapat dievaluasi dalam Euler secara numerik sama halnya ekspresi simbolik lainnya

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

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

Dalam ekspresi simbolis, gunakan dengan.

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

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

Akses ke baris matriks simbolik bekerja seperti halnya matriks numerik.

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

$$[1, a]$$

Ekspresi simbolik dapat berisi sebuah penugasan. Dan itu mengubah matriks A.

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

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

Terdapat fungsi-fungsi simbolik dalam Maxima untuk membuat vektor dan matriks. Untuk hal 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)
```

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

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

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

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

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

Euler juga memiliki fungsi yang kuat xinv(), yang membuat usaha yang lebih besar dan mendapatkan hasil yang lebih tepat.

Perhatikan, bahwa dengan &:= matriks B telah didefinisikan sebagai simbolik dalam ekspresi simbolik dan sebagai numerik dalam ekspresi numerik. Jadi kita dapat menggunakannya di sini.

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

$$\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}$$

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 mengetahui 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 simbolik hanyalah sebuah string yang berisi ekspresi. Jika kita ingin mendefinisikan nilai untuk ekspresi simbolik dan ekspresi numerik, kita harus menggunakan "&:=".

```
> A &:= [1, pi; 4, 5]
```

$$\begin{matrix} 1 & 3.14159 \\ 4 & 5 \end{matrix}$$

Masih ada perbedaan antara bentuk numerik dan bentuk simbolik. Ketika mentransfer matriks ke bentuk simbolik, perkiraan pecahan untuk bilangan real akan digunakan.

```
> $&A
```

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

Untuk menghindari hal ini, ada fungsi "mxmset(variable)"

```
> mxmset (A); $&A
```

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

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

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

$1.4142135623730950488016887242097_B \times 10^0$

1.414213562373095

Ketepatan angka floating point yang besar dapat diubah

```
>&fpprec:=100; &bfloat(pi)
```

3.14159265358979323846264338327950288419716939937510582097494\
4592307816406286208998628034825342117068b0

Variabel numerik dapat digunakan dalam ekspresi simbolik apa pun dengan menggunakan "@var".

Perhatikan bahwa hal 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 simbolik untuk menunjukkan kepada Anda bagaimana Euler dapat digunakan untuk memecahkan masalah kehidupan nyata. Asumsikan Anda memiliki modal awal sebesar 5000 (katakanlah dalam dolar).

```
>K=5000
```

5000.00

2.1. EMT UNTUK PERHITUNGAN ALJABAR

85

Sekarang kita asumsikan suku bunga 3% per tahun. Mari kita tambahkan satu suku bunga sederhana dan hitung hasilnya.

```
>K*1.03
```

5150.00

Euler juga akan memahami sintaks berikut ini.

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

5150.00

Tetapi akan lebih mudah untuk menggunakan faktor

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

1.03
5150.00

Untuk 10 tahun, kita cukup mengalikan faktor-faktor tersebut dan mendapatkan nilai akhir dengan suku bunga majemuk.

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

6719.58

Untuk tujuan kita, kita bisa menetapkan format ke 2 digit setelah titik desimal.

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

6719.58

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

```
>"Mulai dari " + K + "$ Anda akan mendapatkan " + round(K*q^10,2) + "$."
```

Mulai dari 5000\$ Anda akan mendapatkan 6719.58\$.

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

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

Real 1 x 11 matrix

5000.00	5150.00	5304.50	5463.64	...
---------	---------	---------	---------	-----

Bagaimana keajaiban ini bekerja? Pertama, ekspresi 0:10 mengembalikan sebuah vektor bilangan bulat

```
>short 0:10
```

[0,	1,	2,	3,	4,	5,	6,	7,	8,	9,	10]
-----	----	----	----	----	----	----	----	----	----	-----

Kemudian semua operator dan fungsi dalam 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 hingga q^{10} . Ini dikalikan dengan K, dan kita mendapatkan vektor nilai.

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

Tentu saja, cara yang realistik untuk menghitung suku bunga ini adalah dengan membuatkan ke sen terdekat setelah setiap tahun. Mari kita tambahkan fungsi untuk ini.

```
>function oneyear (K) := round(K*q, 2)
```

Mari kita bandingkan kedua hasil tersebut, dengan dan tanpa pembulatan.

```
>longest oneyear(1234.57), longest 1234.57*q
```

1271.61
1271.6071

2.1. EMT UNTUK PERHITUNGAN ALJABAR

87

Sekarang tidak ada rumus sederhana untuk tahun ke-n, dan kita harus mengulang selama bertahun-tahun. Euler menyediakan banyak solusi untuk ini.

Cara termudah adalah iterasi fungsi, yang mengulang fungsi tertentu beberapa kali.

```
>VKr=iterate("oneyear",5000,10)
```

Real 1 x 11 matrix

5000.00 5150.00 5304.50 5463.64 ...

Kami dapat mencetaknya dengan cara yang ramah, menggunakan format kami dengan angka desimal yang tetap.

```
>VKr'
```

5000.00
5150.00
5304.50
5463.64
5627.55
5796.38
5970.27
6149.38
6333.86
6523.88
6719.60

Untuk mendapatkan elemen tertentu dari vektor, kita menggunakan indeks dalam tanda kurung siku

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

5150.00
5000.00 5150.00 5304.50

Ternyata, kita juga dapat 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

Memecahkan Persamaan

Sekarang kita mengambil fungsi yang lebih canggih, yang menambahkan sejumlah uang setiap tahun.

```
>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 bahwa uangnya berkurang. Jelas, jika kita hanya mendapatkan 150 bunga di tahun pertama, tetapi menghapus 200 bunga, kita akan kehilangan uang setiap tahun. Bagaimana kita dapat menentukan jumlah tahun uang tersebut akan bertahan? Kita harus menulis perulangan untuk ini. Cara termudah adalah dengan melakukan perulangan yang cukup lama.

```
>VKR=iterate("onepay",5000,50)
```

Real 1 x 51 matrix

5000.00	4950.00	4898.50	4845.45	...
---------	---------	---------	---------	-----

Dengan menggunakan bahasa matriks, kita dapat menentukan nilai negatif pertama dengan cara berikut

```
>min (nonzeros (VKR<0))
```

48.00

Alasannya adalah karena nonzeros(VKR<0) mengembalikan vektor indeks i, di mana VKR[i]<0, dan min menghitung indeks minimal.

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

Fungsi iterate() memiliki satu trik lagi. Fungsi ini dapat menerima sebuah kondisi akhir sebagai argumen. Kemudian ia akan mengembalikan nilai dan jumlah iterasi.

```
>{x,n}=iterate ("onipay",5000,till="x<0"); x, n,
```

-19.83

47.00

Mari kita coba menjawab pertanyaan yang lebih ambigu. Anggaplah kita tahu bahwa nilainya adalah 0 setelah 50 tahun. Berapakah tingkat suku bunganya?

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

Langkah pertama adalah mendefinisikan sebuah 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

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

Tetapi kita tidak lagi menggunakan nilai global R dalam ekspresi kita. Fungsi-fungsi seperti iterate() memiliki trik khusus dalam Euler. Anda bisa mengoper nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R. Selain itu, kita hanya tertarik pada nilai terakhir. Jadi kita mengambil indeks [-1].

Mari kita coba tes

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

-19.83

Sekarang kita bisa menyelesaikan masalah kita

```
>solve("f(5000,-200,x,50)",3)
```

3.15

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

Kita dapat menggunakan fungsi yang sama untuk menyelesaikan pertanyaan berikut: Berapa banyak yang dapat kita keluarkan per tahun agar modal awal habis setelah 20 tahun dengan asumsi suku 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 untuk Masalah Suku Bunga

Kita bisa menggunakan bagian simbolik dari Euler untuk mempelajari masalah ini. Pertama, kita mendefinisikan fungsi onepay() secara simbolik.

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

$$R + qK$$

Sekarang kita bisa mengulanginya.

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

$$q(q(q(R + qK) + R) + R)$$

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

2.1. EMT UNTUK PERHITUNGAN ALJABAR

91

Kita melihat sebuah pola. Setelah n periode, kita memiliki

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

Rumus tersebut adalah rumus untuk jumlah geometris, yang diketahui oleh Maxima.

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

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

Ini sedikit rumit. Penjumlahan dievaluasi dengan flag "simpsum" untuk menguranginya menjadi hasil bagi.

Mari kita buat sebuah fungsi untuk ini.

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

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

Fungsi ini melakukan hal yang sama seperti fungsi f kita sebelumnya. Tetapi fungsi ini lebih efektif.

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

```
-19.82504734650985  
-19.82504734652684
```

Sekarang kita bisa menggunakan untuk menanyakan waktu n. Kapan modal kita habis? Perkiraan awal kita adalah 30 tahun.

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

20.51

Jawaban ini mengatakan bahwa itu akan menjadi negatif setelah 21 tahun.

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

Asumsikan kita mendapatkan pinjaman sebesar K , dan membayar n kali pembayaran sebesar R (dimulai setelah tahun pertama) sehingga menyisakan sisa utang sebesar Kn (pada saat pembayaran terakhir).

Rumus untuk ini adalah sebagai berikut

```
>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
```

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

Kita dapat menyelesaikan laju R secara simbolis

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

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

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

Tentu saja, kita memiliki batas berikut.

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

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

2.1. EMT UNTUK PERHITUNGAN ALJABAR

93

Jelas, tanpa bunga kita harus membayar kembali 10 suku bunga 500. Persamaan ini juga dapat diselesaikan untuk n. Akan terlihat lebih baik jika kita menerapkan beberapa penyederhanaan.

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

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

R.2 Exercise Set

49. Menyederhanakan persamaan

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

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

$$\{2 a^{\{10\}+6} b^{\{-3\}+\{-8\}} c^{12}\}^{\{-5\}}$$

disederhanakan dengan memfaktorkan dan mengeliminasi penyebut dengan pembilangnya

50. Menyederhanakan persamaan berikut

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

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

$$\{5 p^{\{12\}+\{8\}} q^{\{6\}+\{-14\}} r^{\{22\}+\{-15\}}\}^{\{-4\}}$$

87. hitunglah:

$$5.3 + 8.3(2) + 4(6 - 2)$$

$$>\$ & (5 * 3) + (8 * 3^2) + 4 * (6 - 2)$$

103

91. hitunglah:

$$\left(\frac{4(8 - 6)^2 - 4 \cdot 3 + 2.8}{3^1 + 19^0} \right)$$

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

5

92. hitunglah:

$$\left[\frac{4(8 - 6)^2 + 4](3 - 2.8)}{2^2(2^3 + 5)} \right]$$

$$>\$ & ((4 * (8 - 6)^2 + 4) * (3 - 2 * 8)) / (2^2 * (2^3 + 5))$$

-5

R.3 Exercise set

No. 7

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

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

$$6y - 5x - 19$$

$$(2x^2 + 12xy - 11) + (6x^2 - 2x + 4) + (-x^2 - y - 2)$$

```
> $& (2*x^2+12*x*y-11)-(6*x^2-2*x+4)+(-x^2-y-2)
```

$$12xy - y - 5x^2 + 2x - 17$$

No. 9

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

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

$$-5x^3 + 3x^2 + 6x - 2$$

No. 14

$$(8y^5)(9y)$$

```
> $& (8*y^5)*(9*y)
```

$$72y^6$$

No. 33

$$(x + 3)^2$$

```
> $& (x+3)^2 | ratsimp
```

$$x^2 + 6x + 9$$

96
No. 10
Faktorkan

BAB 2. EMT UNTUK PERHITUNGAN ALJABAR

$$x^3 + 3x^2 + 6x + 18$$

```
> $&factor(x^3+3*x^2+6*x+18)
```

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

No. 11
Faktorkan

$$y^3 - y^2 + 2y - 2$$

```
> $&factor(y^3-y^2+2*y-2)
```

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

No. 19
Faktorkan

$$w^2 - 7w + 10$$

```
> $&factor(w^2-7*w+10)
```

$$(w - 5) (w - 2)$$

No. 21
Faktorkan

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

```
> $&factor(x^2+6*x+5)
```

$$(x + 1) (x + 5)$$

No. 81
Faktorkan

$$2n^2 + 9n - 56$$

```
> $&factor(2*n^2+9*n-56)
```

$$(n + 8) (2n - 7)$$

R.5

No. 31
Selesaikan

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

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

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

No. 34
Selesaikan

$$3(2n - 5) - 7 = 4(n - 9)$$

```
> $&solve(3*(2*n-5)-7=4*(n-9))
```

$$[n = -7]$$

98
No. 54
Selesaikan

BAB 2. EMT UNTUK PERHITUNGAN ALJABAR

$$y^2 - 81 = 0$$

```
>$&solve(y^2-81=0)
```

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

No. 81
Selesaikan

$$3[5 - 3(4 - t)] - 2 = 5[3(5t - 4) + 8] - 26$$

```
>$&solve(3*[5-3*(4-t)]-2=5*[3*(5*t-4)+8]-26)
```

$$[[23 - 66 t] = 0]$$

No. 85
Selesaikan

$$(5x^2 + 6x)(12x^2 - 5x - 2) = 0$$

```
>$&solve((5*x^2+6*x)*(12*x^2-5*x-2)=0)
```

$$\left[x = -\frac{1}{4}, x = \frac{2}{3}, x = -\frac{6}{5}, x = 0 \right]$$

R.6

No. 9

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

```
>${&(x^2-4) / (x^2-4*x+4)} | ratsimp
```

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

No. 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
```

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

No. 16

$$\frac{6-x}{x^2 - 36}$$

```
>${&(6-x) / (x^2-36)} | ratsimp
```

$$-\frac{1}{x+6}$$

No. 32

$$\frac{7}{12y} - \frac{1}{12y}$$

```
>${&(7 / (12*y) - 1 / (12*y))}
```

$$\frac{1}{2y}$$

$$\frac{4}{3a+4} - \frac{3a}{3a+4}$$

```
>${\&} (4 / (3*a+4) - 3*a / (3*a+4)) | ratsimp
```

$$\frac{4 - 3a}{3a + 4}$$

2.3

```
>function f(x) &= 3*x+1; function g(x) &= x^2-2*x-6; function h(x) &= x^3;
```

No. 1

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

```
>${\&} f(x) \text{ with } x=g(x), ${\&} ratsimp(%), ${\%} \text{ with } x=-1
```

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

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

$$-8$$

No. 2

$$(g \circ f)(-2)$$

```
>${\&} g(x) \text{ with } x=f(x), ${\&} ratsimp(%), ${\%} \text{ with } x=-2
```

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

$$9x^2 - 7$$

29

No. 3

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

```
> $&h(x) with x=f(x), $&ratsimp(%), $% with x=1
```

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

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

64

No. 5

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

```
> $&g(x) with x=f(x), $&ratsimp(%), $% with x=5
```

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

$$9x^2 - 7$$

218

No. 8

$$(h \circ g)(3)$$

```
> $&h(x) with x=g(x), $&ratsimp(%), $% with x=3
```

$$x^6 - 6x^5 - 6x^4 + 64x^3 + 36x^2 - 216x - 216$$

-27

3.1 Exercise Set

```
>&powerdisp:true
```

true

No. 11

$$(-5 + 3i) + (7 + 8i)$$

```
>$& (-5+3*I) + (7+8*I)
```

$$2 + 11i$$

No. 13

$$(4 - 9i) + (1 - 3i)$$

```
>$& (4-9*I) + (1-3*I)
```

$$5 - 12i$$

No. 20

$$(7 - \sqrt{-36}) + (2 + \sqrt{-9})$$

```
> $& (7-\sqrt{-36}) + (2+\sqrt{-9})
```

$$9 - 3i$$

No. 21

$$(10 + 7i) - (5 + 3i)$$

```
> $& (10+7*I) - (5+3*I)
```

$$5 + 4i$$

No. 22

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

```
> $& (-3-4*I) - (8-I)
```

$$-11 - 3i$$

3.4 Exercise Set

No. 1
Selesaikan

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

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

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

104
No. 2
Selesaikan

BAB 2. EMT UNTUK PERHITUNGAN ALJABAR

$$\frac{1}{3} - \frac{5}{6} = \frac{1}{x}$$

```
>$&solve ((1/3)+(5/6)=(1/x))
```

$$\left[x = \frac{6}{7} \right]$$

No. 6
Selesaikan

$$\frac{1}{t} + \frac{1}{2t} + \frac{1}{3t} = 5$$

```
>$&solve ((1/t)+(1/2*t)+(1/3*t)=5)
```

$$\left[t = \frac{15 - \sqrt{195}}{5}, t = \frac{15 + \sqrt{195}}{5} \right]$$

No. 8

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

```
>$&solve ((2/x-2)=(3/x+2))
```

$$\left[x = -\frac{1}{4} \right]$$

No. 10

$$x - \frac{12}{x} = 1$$

```
>$&solve (x-(12/x)=1)
```

$$[x = -3, x = 4]$$

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

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

No. 23

$$|x + 3| - 2 = 8$$

```
>$&fourier_elim(abs(x+3)-2= 8, [x])
```

$$[x = 7] \vee [x = -13]$$

No. 27

$$|4x - 3| + 1 = 7$$

```
>$&fourier_elim(abs(4*x-3)+1= 7, [x])
```

$$\left[x = \frac{9}{4}\right] \vee \left[x = -\frac{3}{4}\right]$$

No. 29

$$12 - |x + 6| = 5$$

```
>$&fourier_elim(abs(12-(x+6))= 5, [x])
```

$$[x = 11] \vee [x = 1]$$

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

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

$$[x = 1] \vee [x = 0]$$

No. 45

$$|x + 8| < 9$$

```
>${&}fourier_elim(abs(x+8)<9, [x])
```

$$[-17 < x, x < 1]$$

4.1

No. 23

Apakah 4, 5, dan -2 solusi dari polinomial

$$f(x) = x^3 - 9x^2 + 14x + 24$$

```
>function f(x) &= x^3-9*x^2+14*x+24;  
>${&}f(x) with x=4
```

$$0$$

```
>${&}f(x) with x=5
```

$$-6$$

```
> $&f(x) with x=-2
```

-48

No. 24

Apakah 2, 3, dan -1 solusi dari

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

```
>function f(x) &= 2*x^3-3*x^2+x+6;
>$&f(x) with x=2
```

12

```
> $&f(x) with x=3
```

36

```
> $&f(x) with x=-1
```

0

No. 25

Apakah 2, 3, dan -1 solusi dari polinomial

$$g(x) = x^4 - 6x^3 + 8x^2 + 6x - 9$$

```
>function g(x) &= x^4-6*x^3+8*x^2+6*x-9;
>$&f(x) with x=2
```

12

```
>$&f(x) with x=3
```

36

```
>$&f(x) with x=-1
```

0

No. 26

Apakah 1, -2, dan 3 solusi dari polinomial

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

```
>function g(x) &= x^4-x^3-3*x^2+5*x-2;
>$&f(x) with x=1
```

6

```
>$&f(x) with x=-2
```

-24

```
>$&f(x) with x=3
```

36

No. 62

$$(x - 2)(x + 5) > x(x - 3)$$

```
>${&fourier_elim( ((x-2)*(x+5)>x*(x-3)), [x] )}
```

$$\left[\frac{5}{3} < x \right]$$

BAB 3

Menggambar Grafik 2D dengan EMT

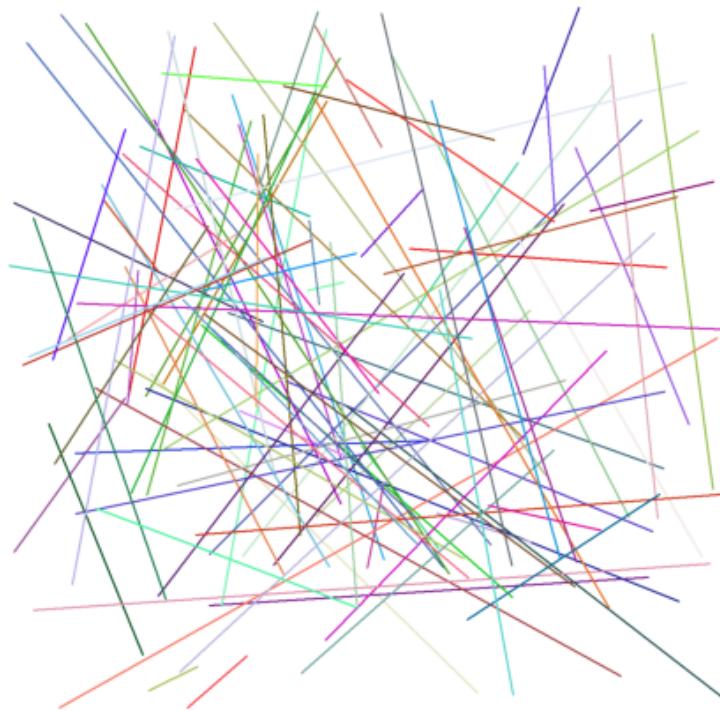
Notebook ini menjelaskan tentang cara menggambar berbagai kurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi plot2d() untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

Plot Dasar

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

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

```
>clg; // membersihkan layar
>window(0,0,1024,1024); // menggunakan semua jendela
>setplot(0,1,0,1); // mengatur koordinat plot
>hold on; // mulai mode timpa
>n=100; X=random(n,2); Y=random(n,2); // mendapatkan nilai acak
>colors=rgb(random(n),random(n),random(n)); // mendapatkan warna acak
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
>hold off; // mengakhiri mode timpa
>insimg; // menyisipkan ke lembar catatan
```



```
>reset;
```

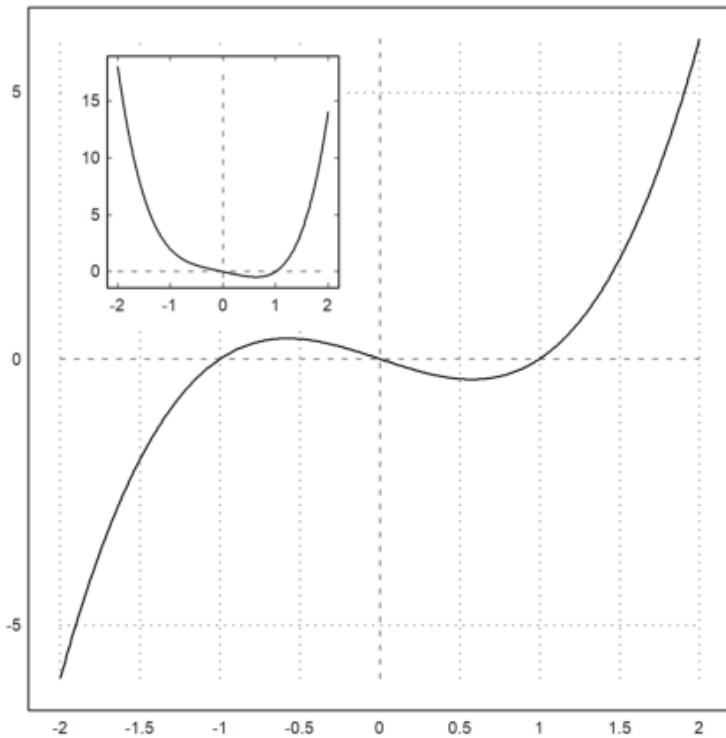
Anda harus menahan grafik, karena perintah plot() akan menghapus jendela plot. Untuk menghapus semua yang telah kita lakukan, kita menggunakan reset().

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

Sebagai contoh lain, kita menggambar plot sebagai inset dalam plot lain. Hal ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan beberapa margin untuk hal ini sesuai kebutuhan.

Perhatikan bahwa kita menyimpan dan mengembalikan jendela penuh, dan menahan plot saat ini ketika kita membuat inse

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



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

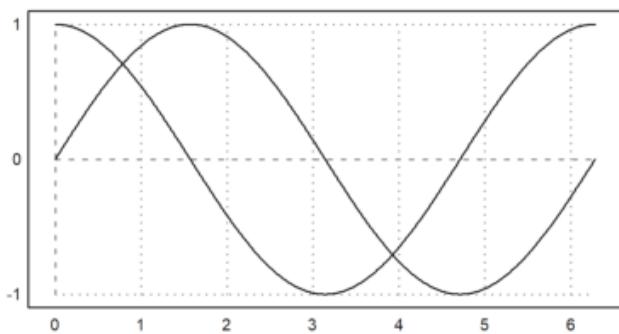
Plot dengan beberapa angka dicapai dengan cara yang sama. Ada fungsi utility figure() untuk ini.

Aspek Plot

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

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

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



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

Fungsi reset() memulihkan default plot, termasuk rasio aspek.

Plot 2D di Euler

EMT Math Toolbox memiliki plot dalam bentuk 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d.

Fungsi ini dapat memplot fungsi dan data. Hal ini memungkinkan untuk memplot di Maxima menggunakan Gnuplot atau di Python menggunakan Math Plot Lib.

Euler dapat memplot plot 2D dari

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

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

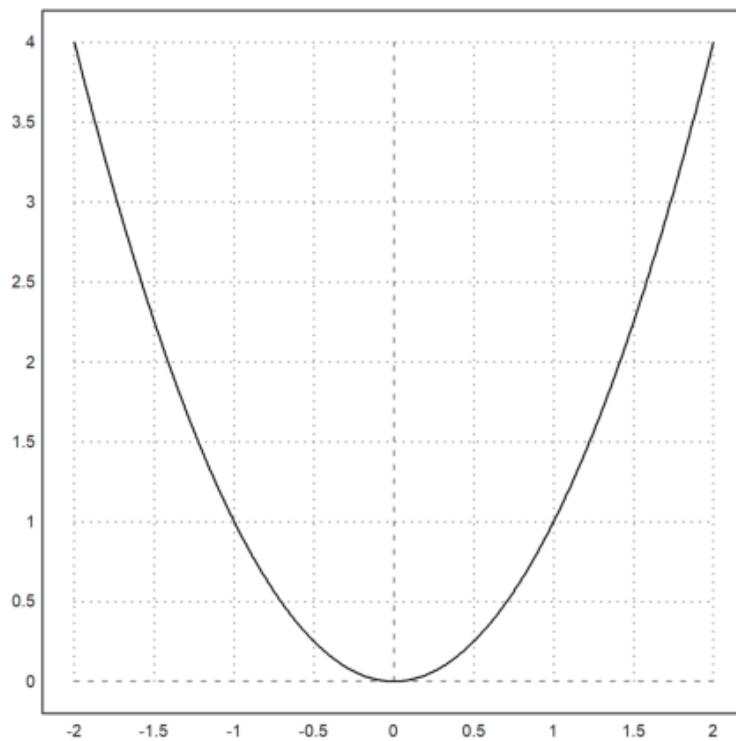
Plot Ekspresi atau Variabel

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

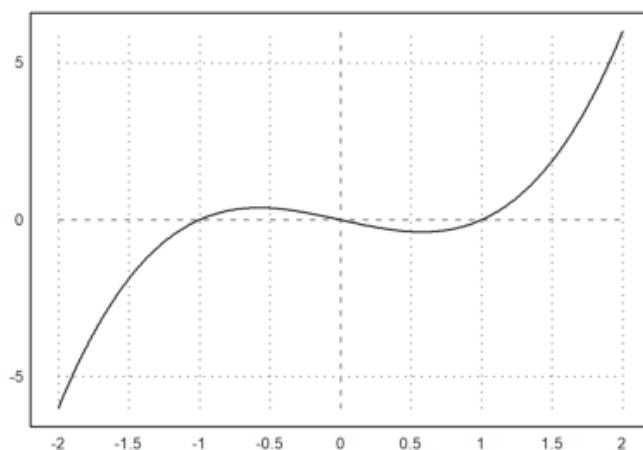
Berikut ini adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsi.

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

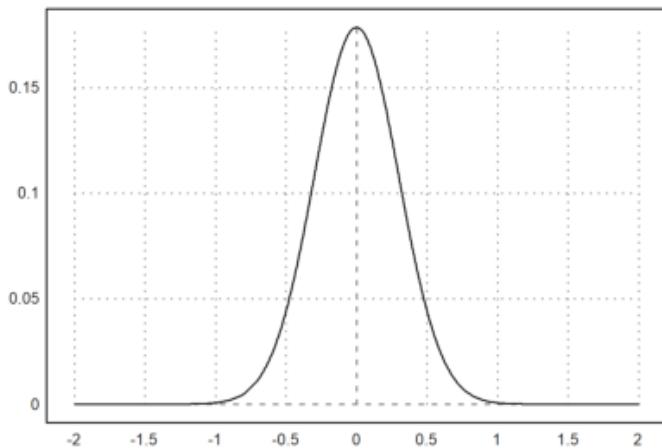
```
>plot2d("x^2"):
```



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



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

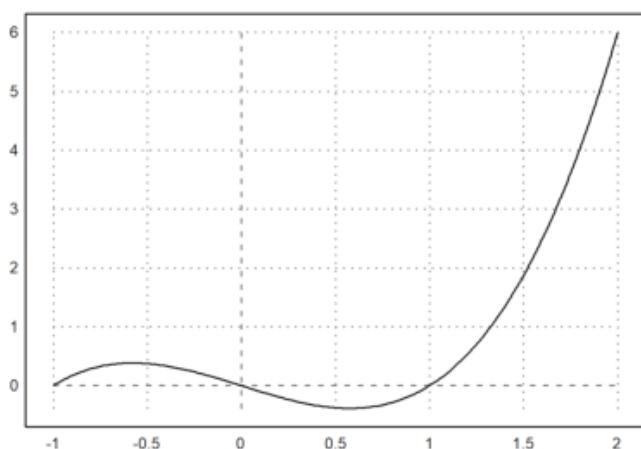


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai batas X (dan Y) di belakang ekspresi yang digambar.

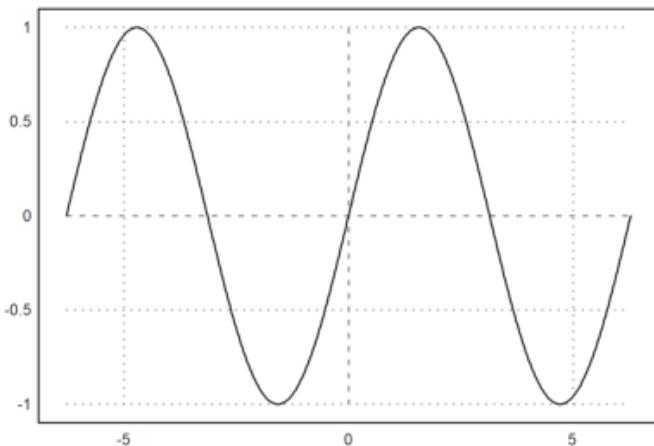
Rentang plot ditetapkan dengan parameter yang ditetapkan berikut ini

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

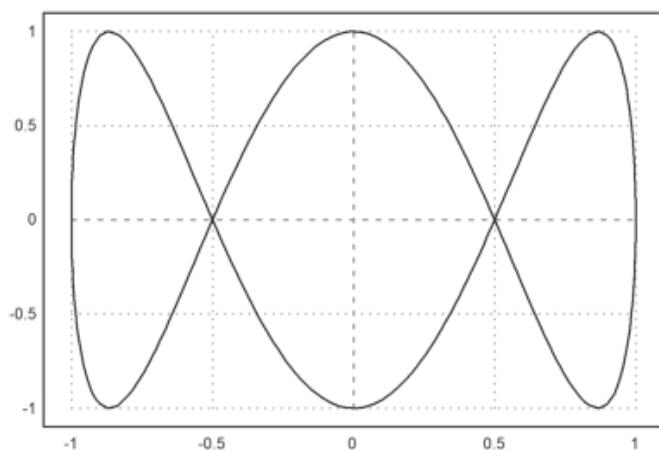
```
>plot2d("x^3-x",-1,2):
```



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



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



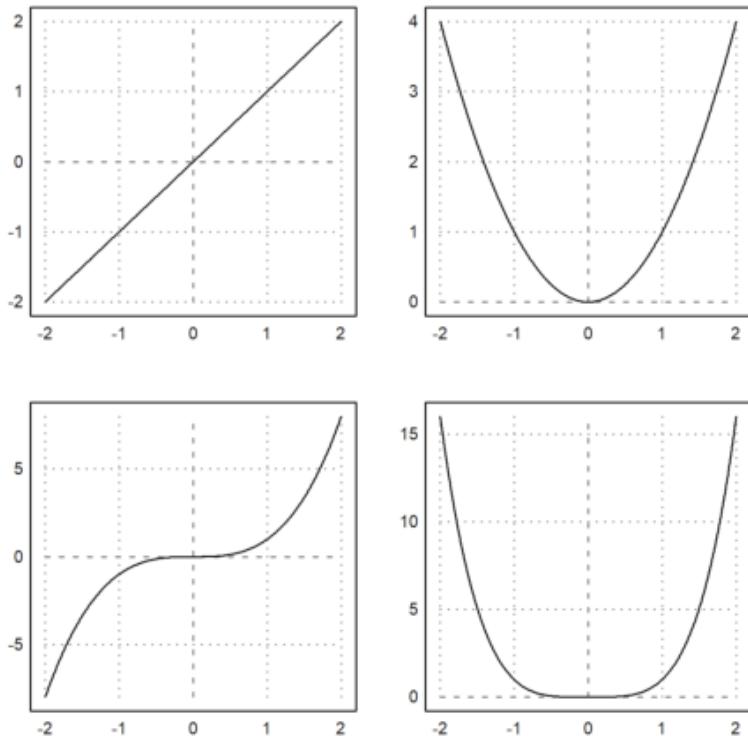
Alternatif untuk tanda titik dua adalah perintah insimg(lines), yang menyisipkan plot yang menempati sejumlah baris teks tertentu.

Dalam opsi, plot dapat diatur untuk muncul

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

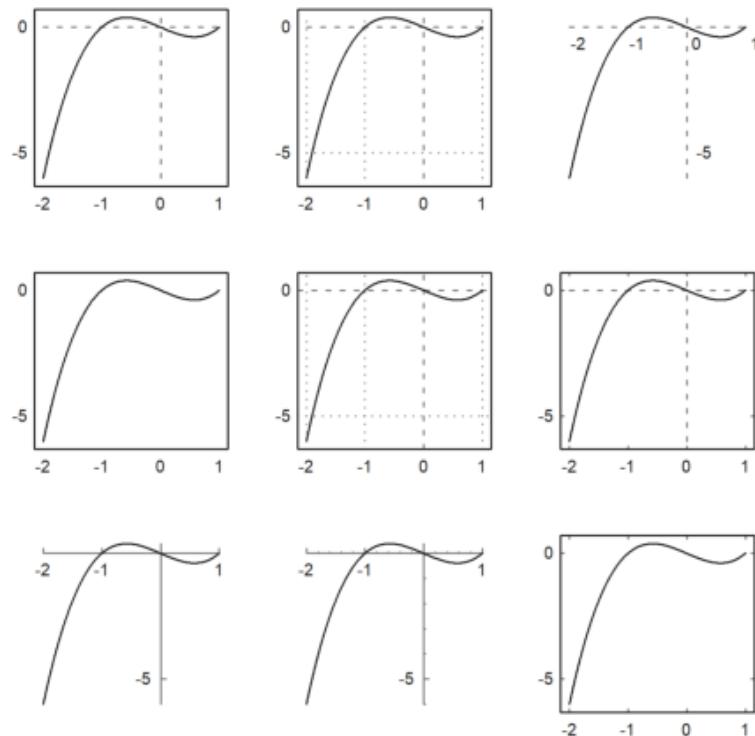
Lebih banyak gaya dapat dicapai dengan perintah plot tertentu. Dalam hal apa pun, tekan tombol tabulator untuk melihat plot, jika disembunyikan. Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Pada contoh, kita memplot x^1 hingga x^4 menjadi 4 bagian jendela. figure(0) mengatur ulang jendela default

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



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

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

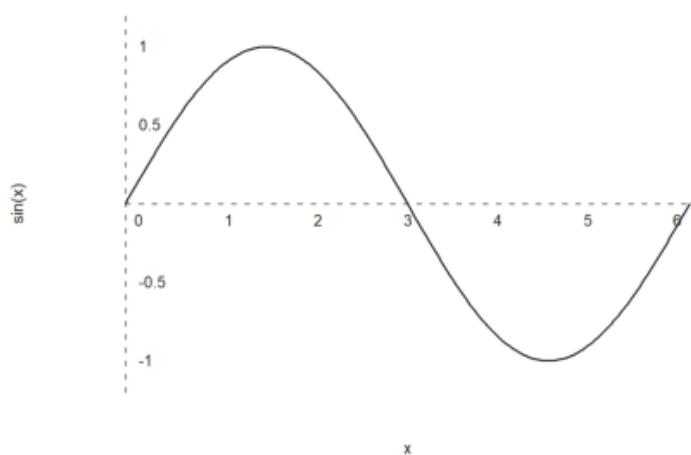


Jika argumen untuk `plot2d()` adalah sebuah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot.

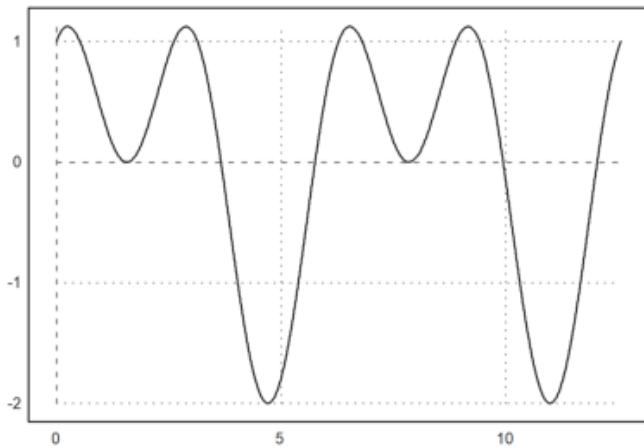
Atau, a, b, c, d dapat ditetapkan sebagai parameter yang ditetapkan sebagai a=... dst.

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):
```

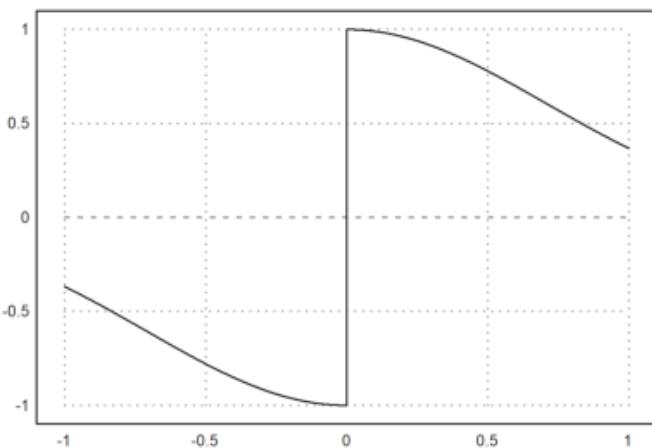


Gambar yang dihasilkan dengan menyisipkan plot ke dalam jendela teks disimpan di direktori yang sama dengan buku catatan, secara default dalam subdirektori bernama "images". Gambar-gambar tersebut juga digunakan oleh ekspor HTML.

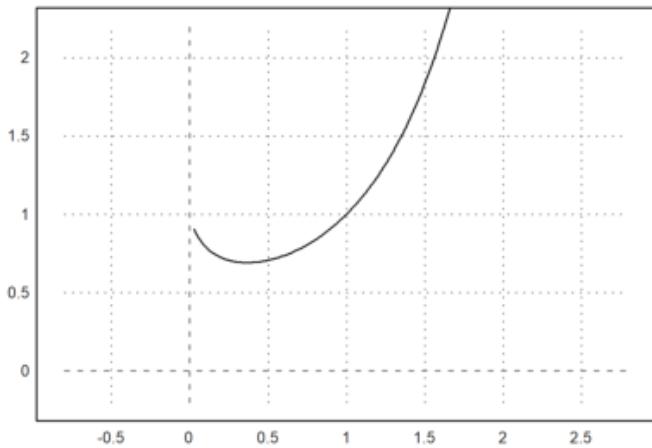
Anda cukup menandai gambar apa pun dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengekspor grafik saat ini dengan fungsi-fungsi dalam menu File.

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

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



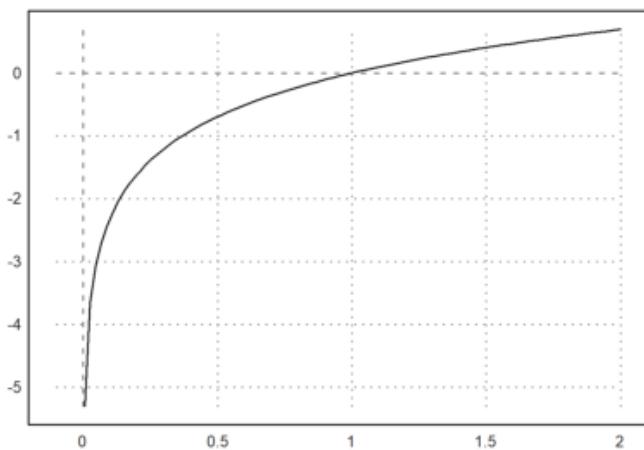
```
>plot2d("x^x", r=1.2, cx=1, cy=1) :
```



Perhatikan bahwa x^x tidak didefinisikan untuk $x \leq 0$. Fungsi plot2d menangkap kesalahan ini, dan mulai memplot segera setelah fungsi didefinisikan.

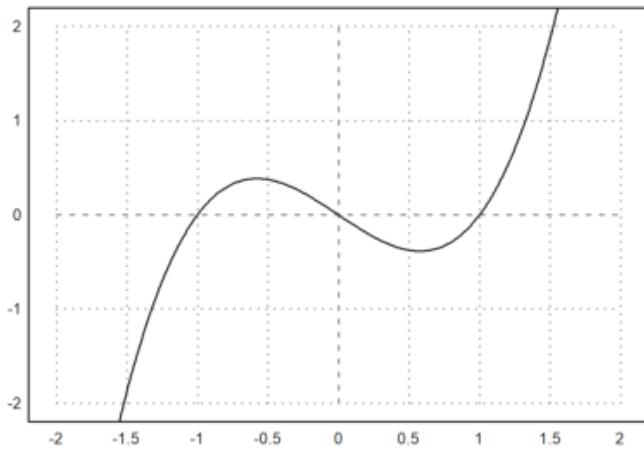
Hal ini berlaku untuk semua fungsi yang mengembalikan NAN di luar jangkauan definisinya

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

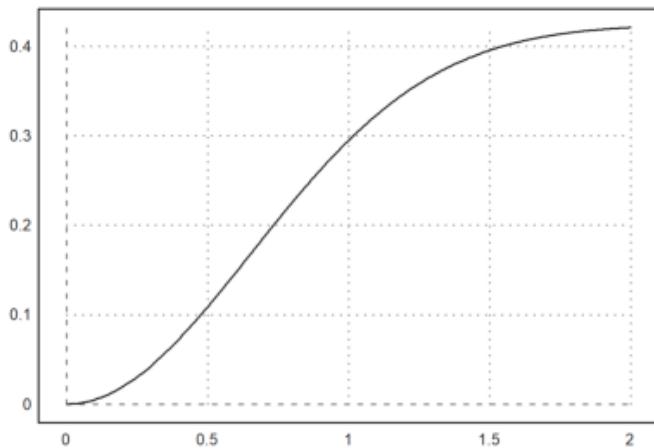


Parameter square=true (atau >square) memilih rentang y secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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

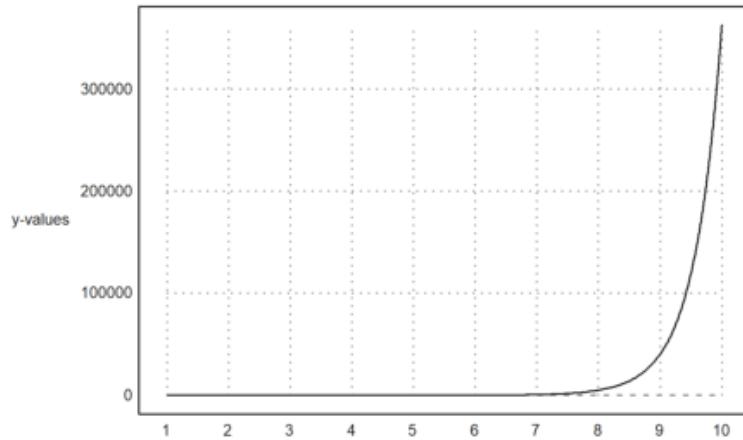


```
>plot2d('' integrate("sin(x)*exp(-x^2)", 0, x)'', 0, 2): // plot integral
```



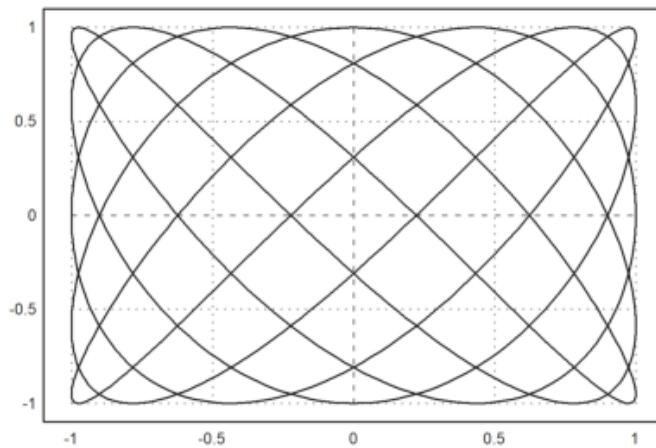
Jika Anda membutuhkan lebih banyak ruang untuk label-y, panggil shrinkwindow() dengan parameter lebih kecil, atau tetapkan nilai positif untuk "smaller" pada plot2d()

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

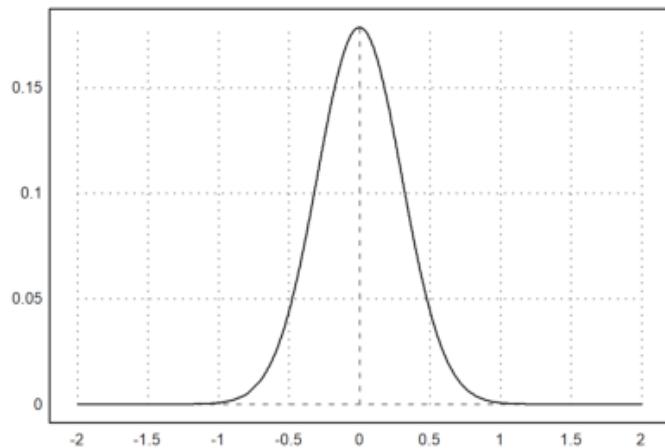


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana

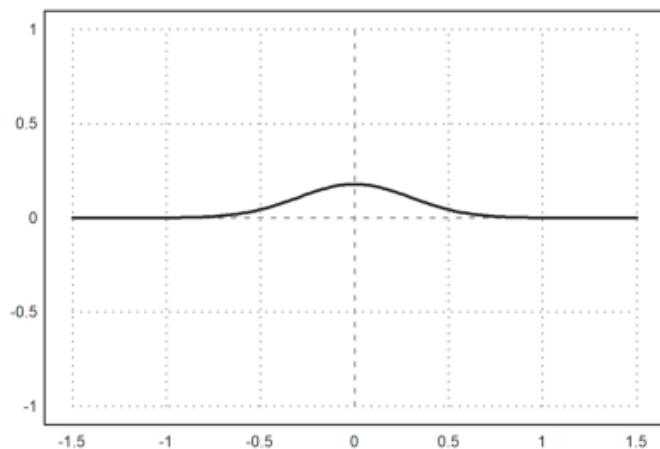
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



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

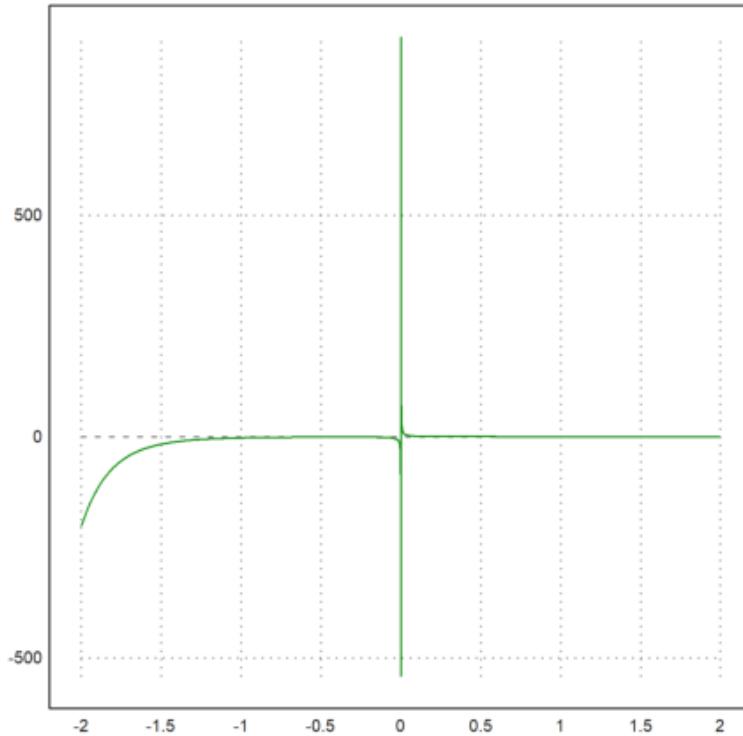


```
>plot2d(expr,r=1,thickness=2): // plot dalam square di sekitar (0,0)
```

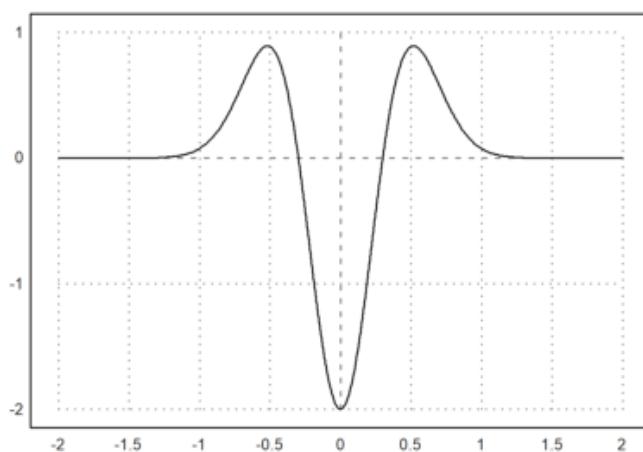


contoh lain

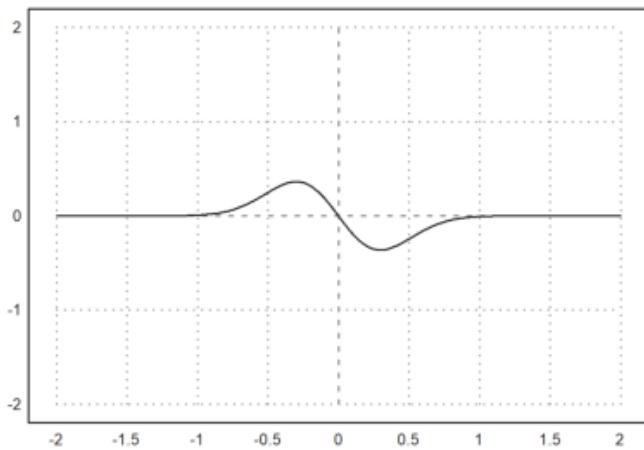
```
>expr &= exp(x^2-2*x-2)/x;
>plot2d(expr,-2,2,color=green):
```



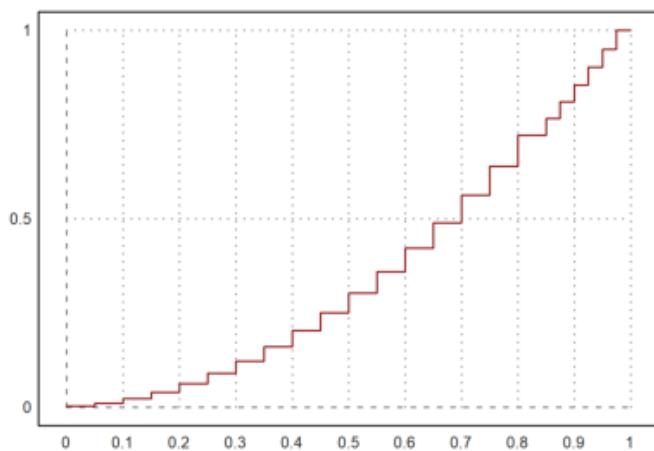
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot dalam persegi panjang
```



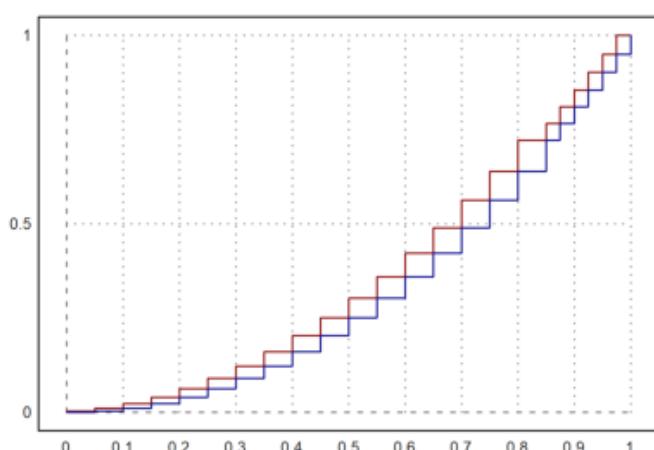
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // menetapkan plot dalam bentuk pe
```



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



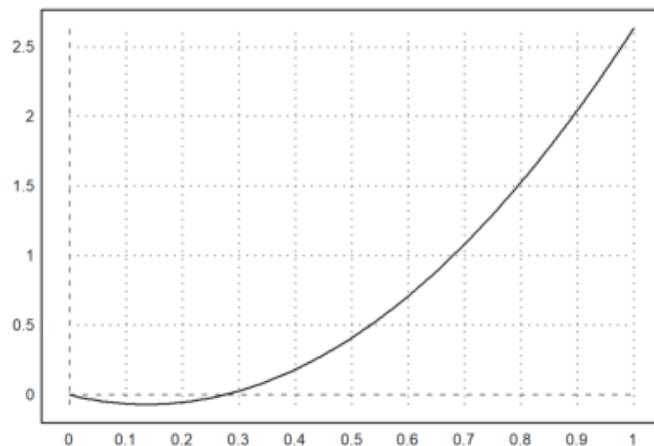
```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```



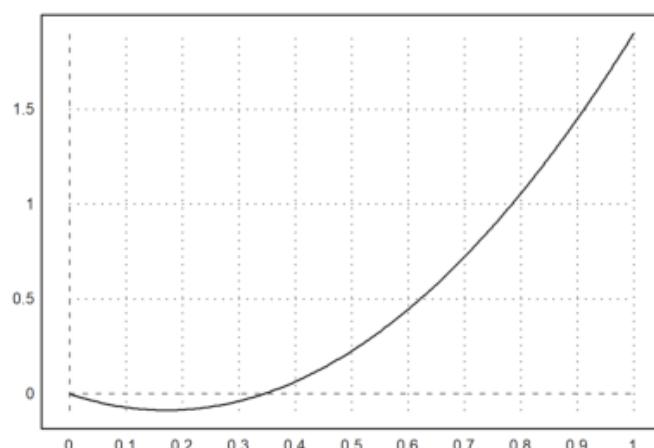
Fungsi plot yang paling penting untuk plot planar adalah `plot2d()`. Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat pada awal program.

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

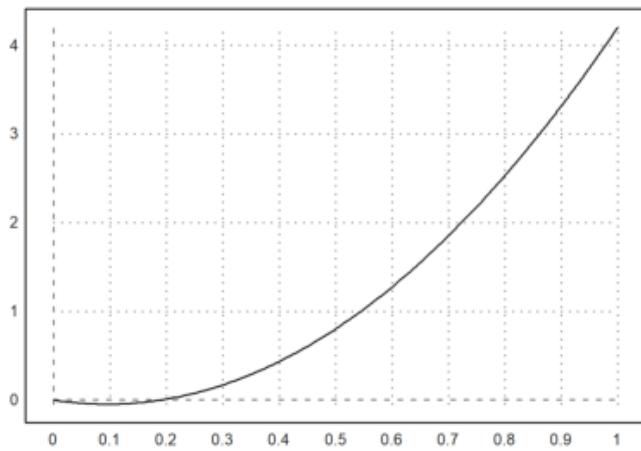
```
>function f(x,a) := x^2/a+a*x^2-x; // mendefinisikan fungsi
>a=0.3; plot2d("f",0,1;a): // plot dengan a=0.3
```



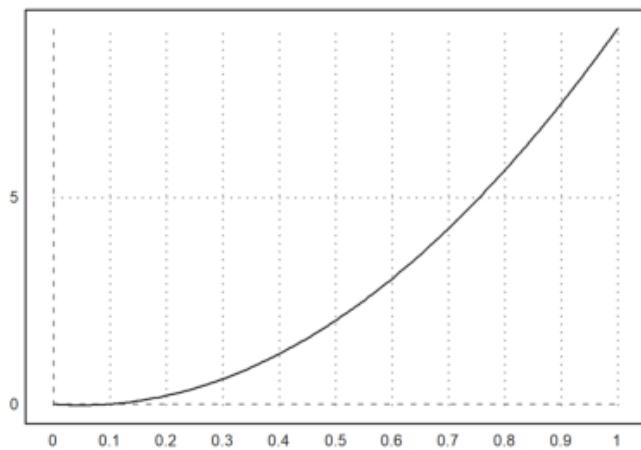
```
>plot2d("f",0,1;0.4): // plot dengan a=0.4
```



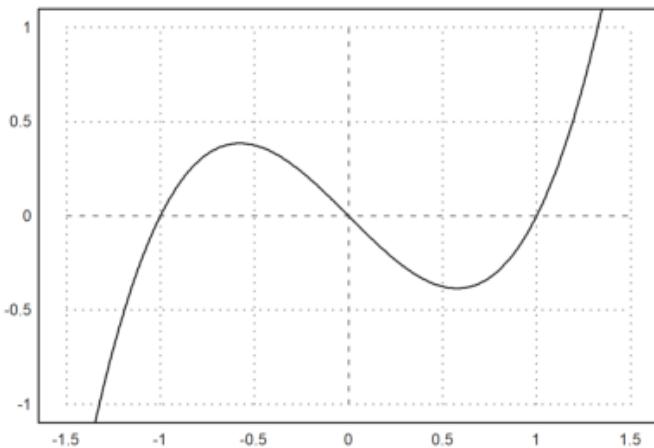
```
>plot2d({{"f",0.2}},0,1): // plot dengan a=0.2
```



```
>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 adalah rangkuman dari fungsi yang diterima

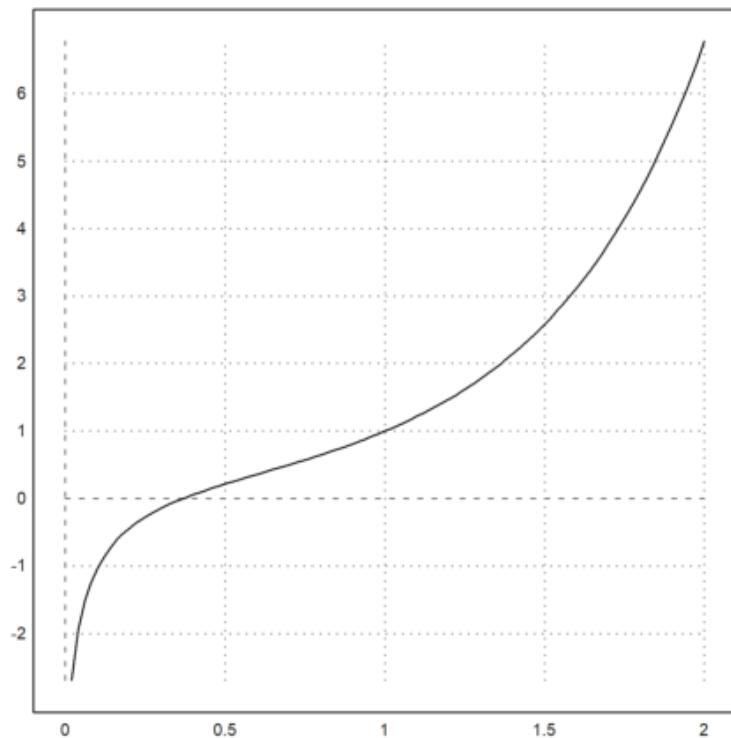
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolis dengan nama sebagai "f"
- fungsi simbolik hanya dengan nama f

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

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

$$\frac{d}{dx} (x^{\log(x) + 1})$$

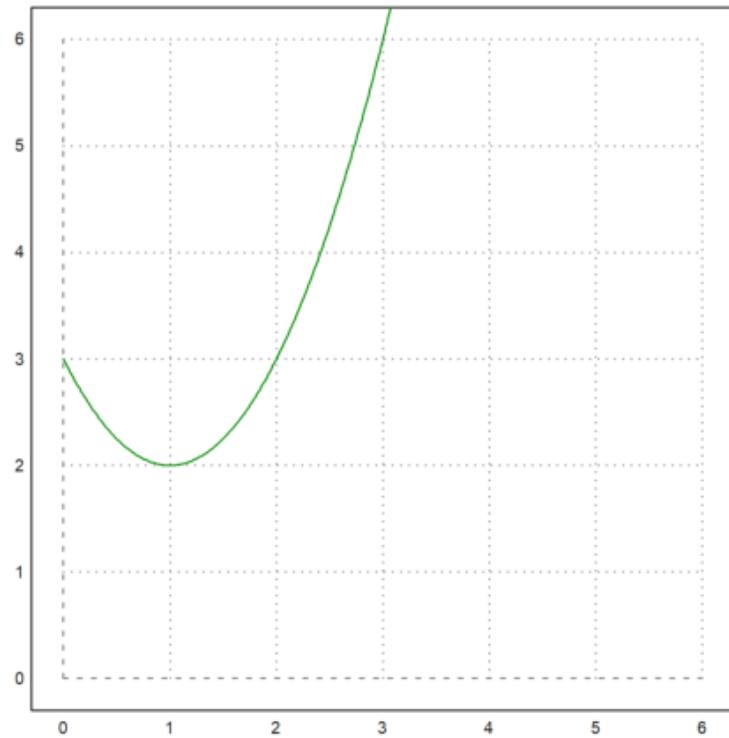
```
>plot2d(f, 0, 2):
```



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

contoh lain

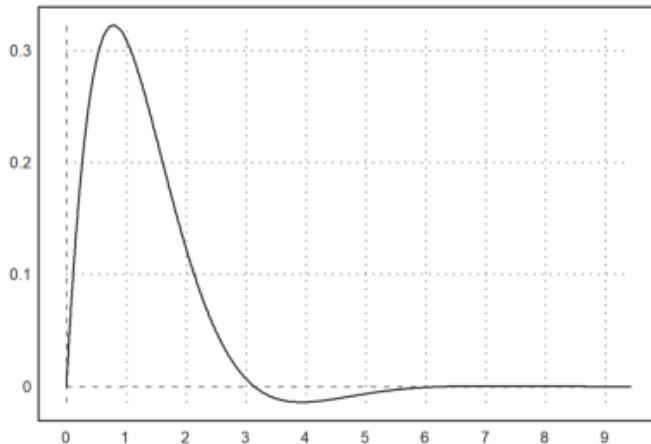
```
>function f(x) &= (x^2+2*x+3);  
>plot2d(f, r=3, cx=3, cy=3, color=green):
```



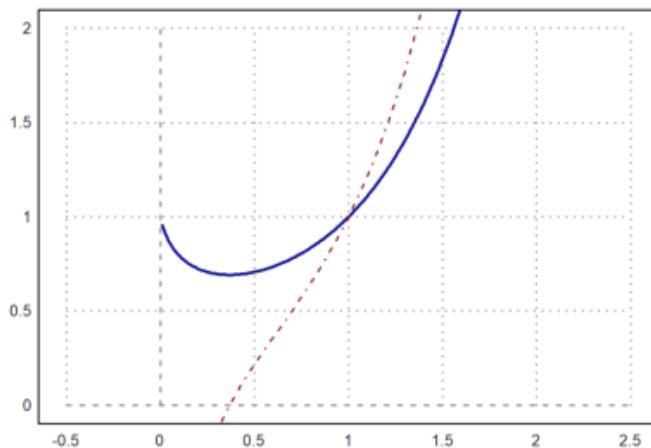
```
>expr &= sin(x)*exp(-x)
```

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

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



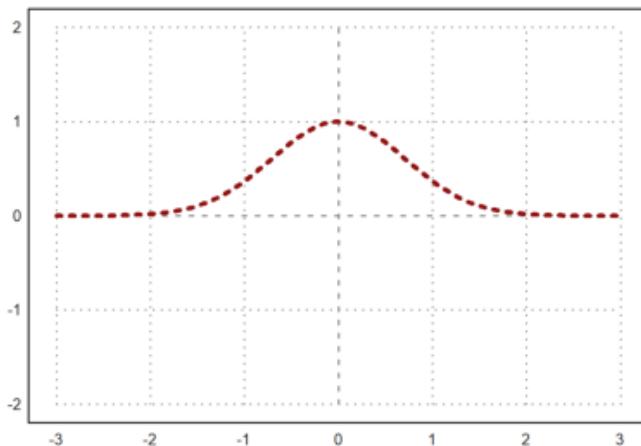
Untuk gaya garis, terdapat berbagai opsi.

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

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

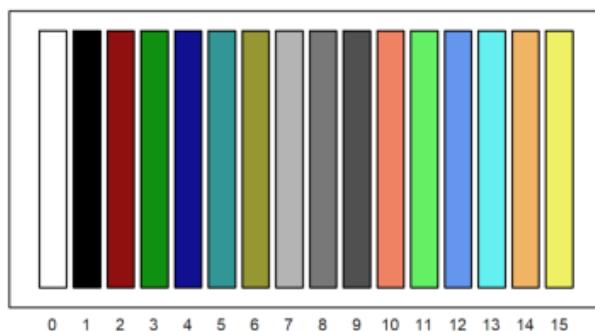
- 0..15: indeks warna default.
- konstanta warna: putih, hitam, merah, hijau, biru, cyan, olive, abu-abu muda, abu-abu, abu-abu tua, oranye, hijau muda, biru kehijauan, biru muda, oranye muda, kuning
- rgb (merah, hijau, biru): parameter dalam bentuk real dalam [0,1].

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



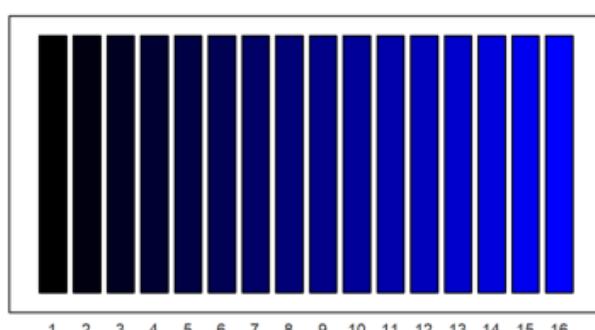
Berikut ini adalah pemandangan warna EMT yang sudah ditetapkan sebelumnya

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



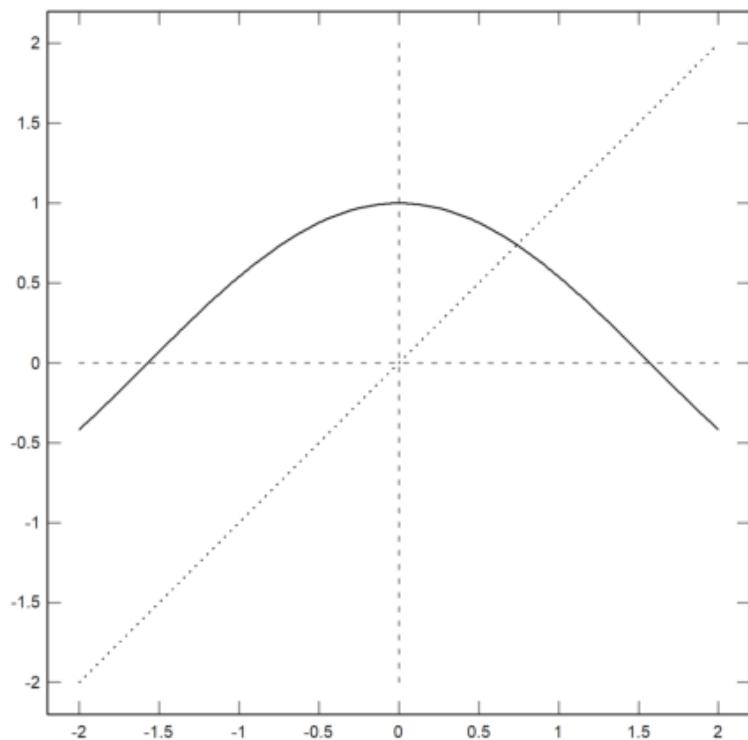
Tetapi Anda dapat menggunakan warna apapun.

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

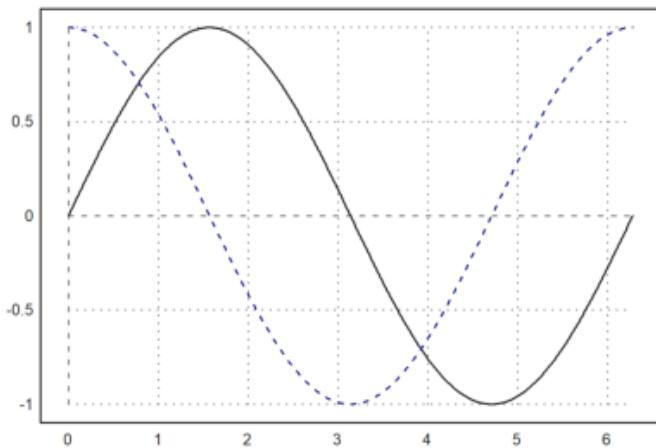


Memplot lebih dari satu fungsi (beberapa fungsi) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu caranya adalah dengan menggunakan >add untuk beberapa pemanggilan ke plot2d secara bersamaan, kecuali pemanggilan pertama. Kita telah menggunakan fitur ini pada contoh di atas.

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



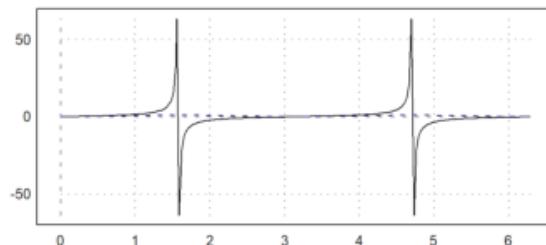
```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--"
```



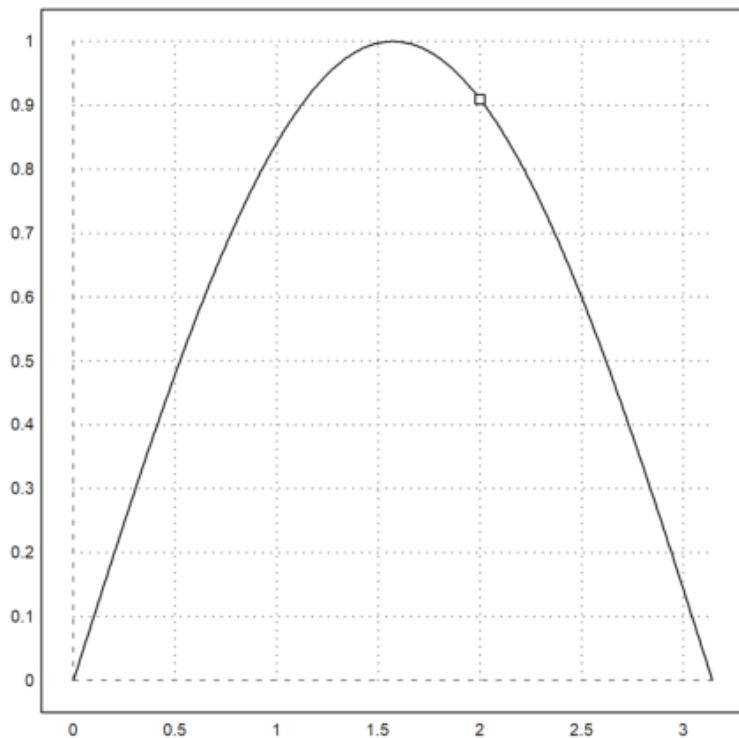
Salah satu kegunaan `>add` adalah untuk menambahkan titik pada kurva.

contoh lain

```
>aspect(2.4); plot2d("tan(x)",0,2pi); plot2d("sin(x)^2",color=blue,style="-")
```

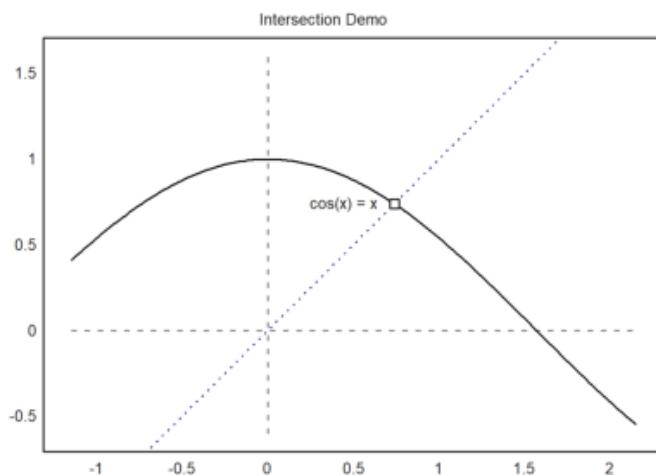


```
>plot2d("sin(x)",0,pi); plot2d(2,sin(2),>points,>add):
```



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

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



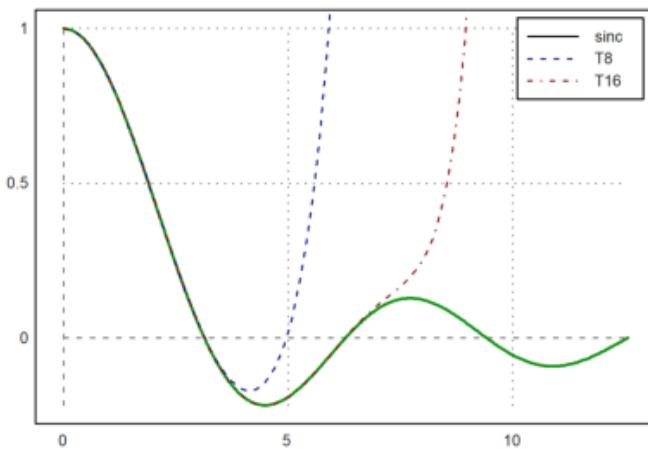
Dalam demo berikut ini, kami memplot fungsi $\text{sinc}(x) = \sin(x)/x$ dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolik. Plot ini dilakukan dalam perintah multi-baris berikut ini dengan tiga kali pemanggilan `plot2d()`. Pemanggilan kedua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

Kami menambahkan kotak label yang menjelaskan fungsinya.

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

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

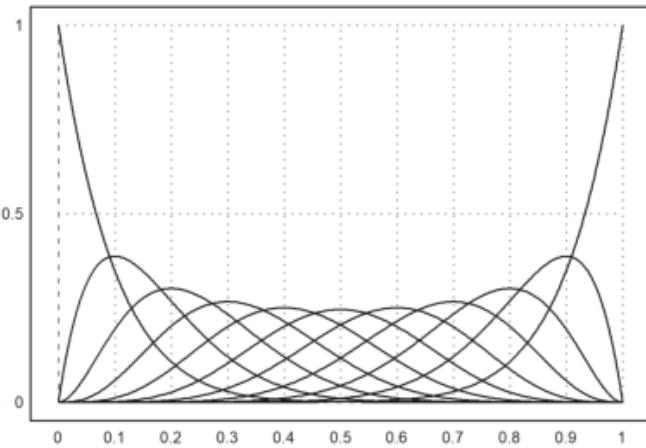
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
> plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
> plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
> labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
> colors=[black,blue,red]):
```



Pada contoh berikut, kita menghasilkan Polinomial-Bernstein

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

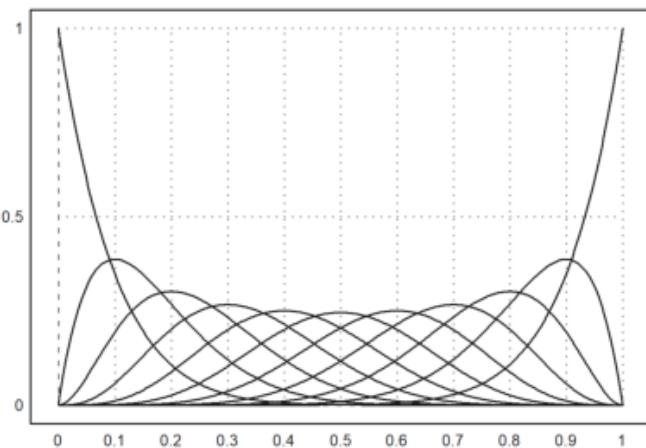
```
>plot2d("(1-x)^10",0,1); // plot fungsi pertama
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```



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

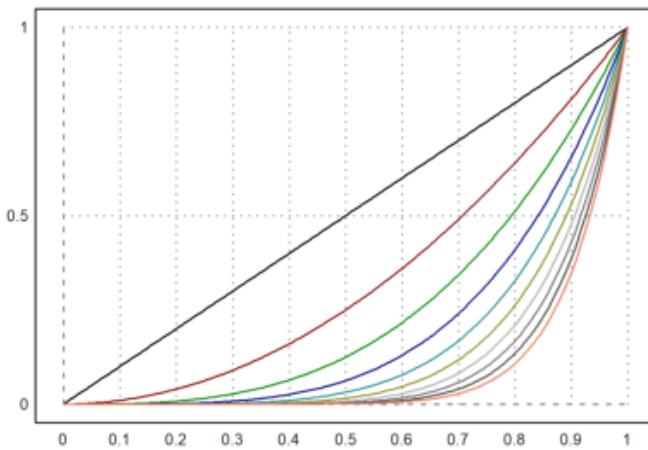
Kita membuat sebuah matriks nilai dengan satu Bernstein-Polynomial di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihatlah pengantar tentang bahasa matriks untuk mempelajari lebih lanjut.

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



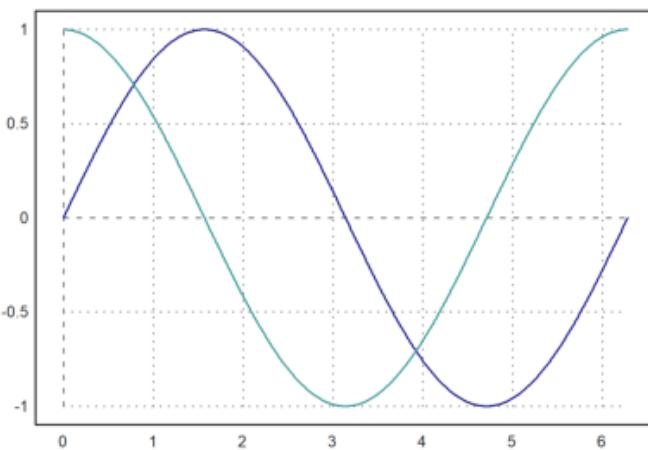
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks

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

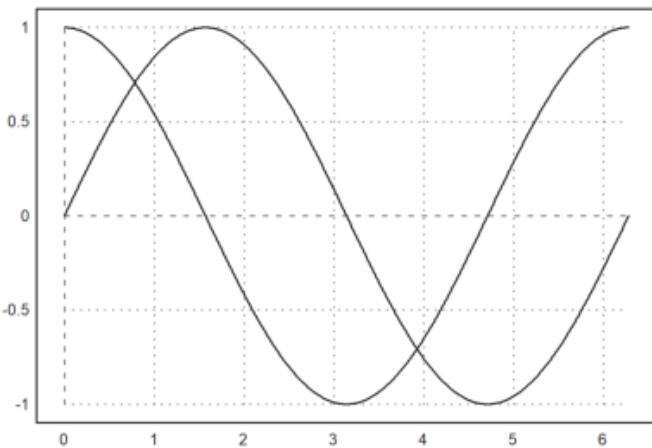


Metode lainnya adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan larik warna, larik gaya, dan larik ketebalan dengan panjang yang sama.

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



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



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

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

```

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

```

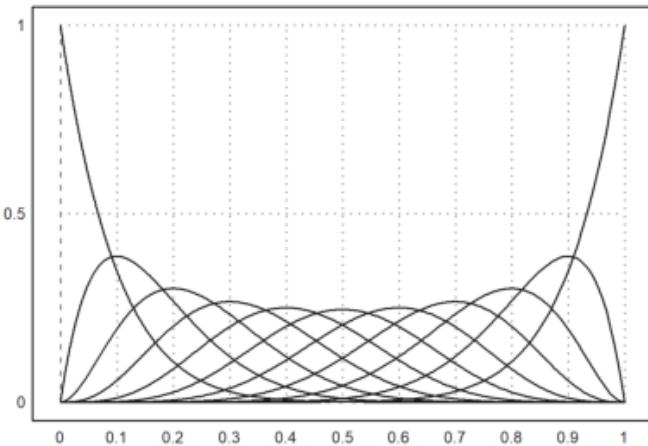
```
>mxm2str(v) // mendapatkan vektor string dari vektor simbolik
```

```

(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
```

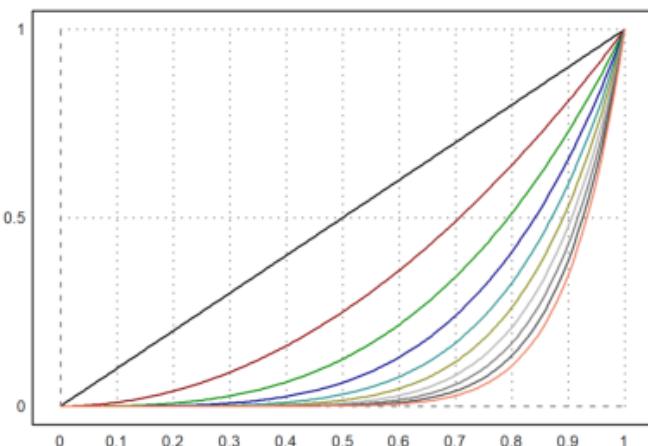


Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

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

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

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

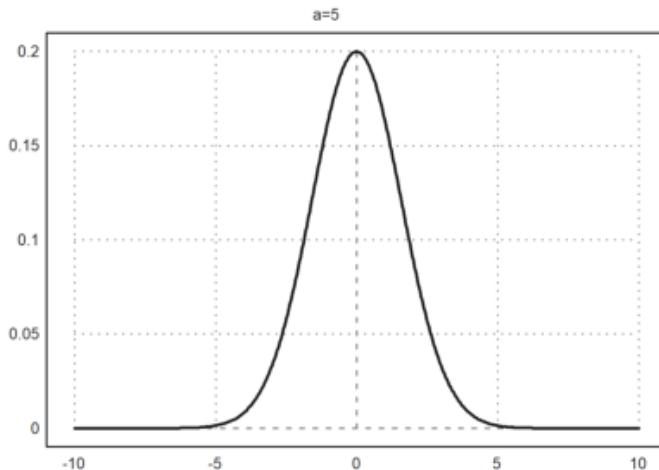


Ekspresi dan fungsi satu baris dapat melihat variabel global.

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

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

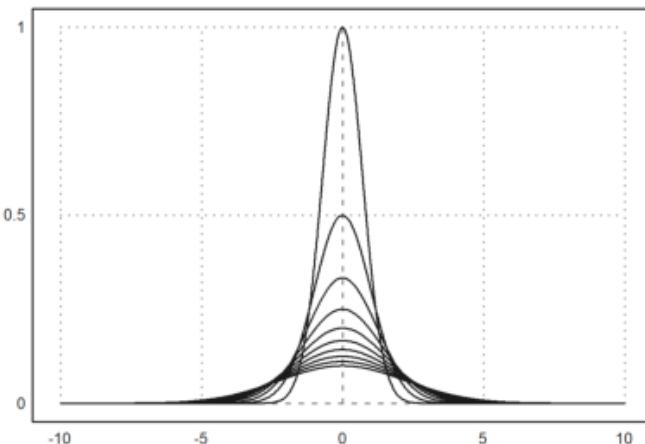
```
>function f(x,a) := 1/a*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
```



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

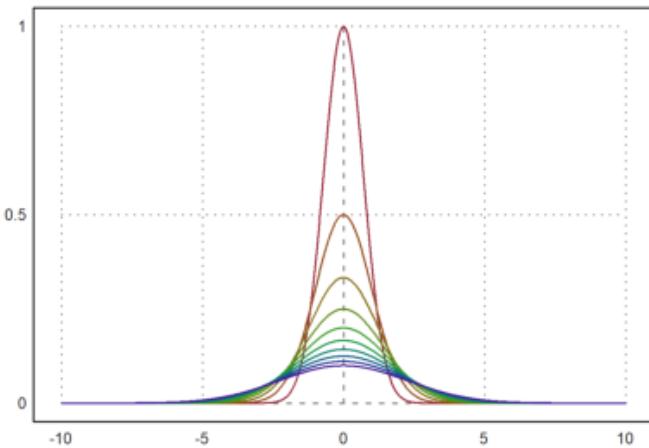
Pada contoh berikut ini, kita menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loop).

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



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

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



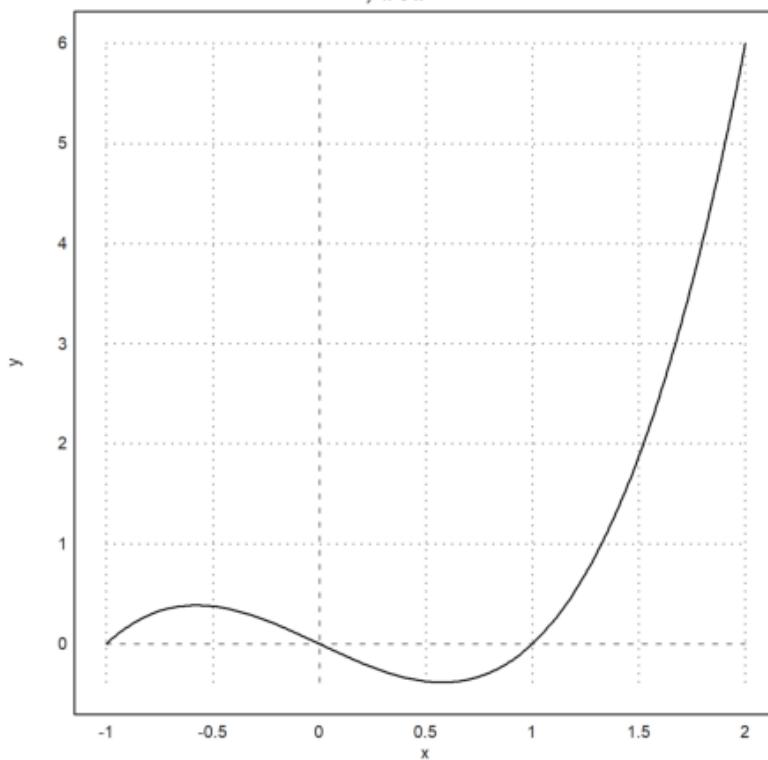
Label Teks

Dekorasi sederhana dapat berupa

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

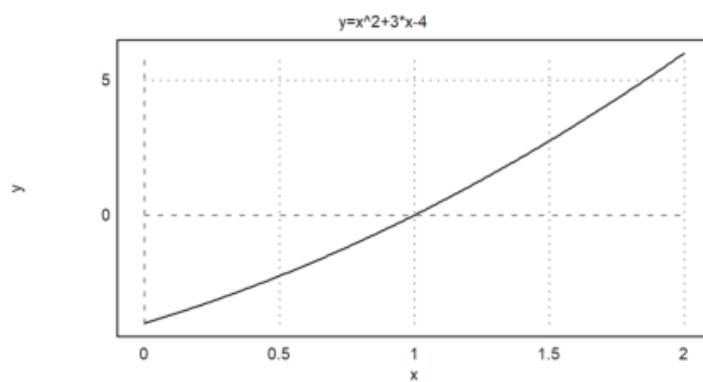
Perintah label akan memplot ke dalam plot saat ini pada koordinat plot (x,y). Perintah ini dapat menerima argumen posisi.

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

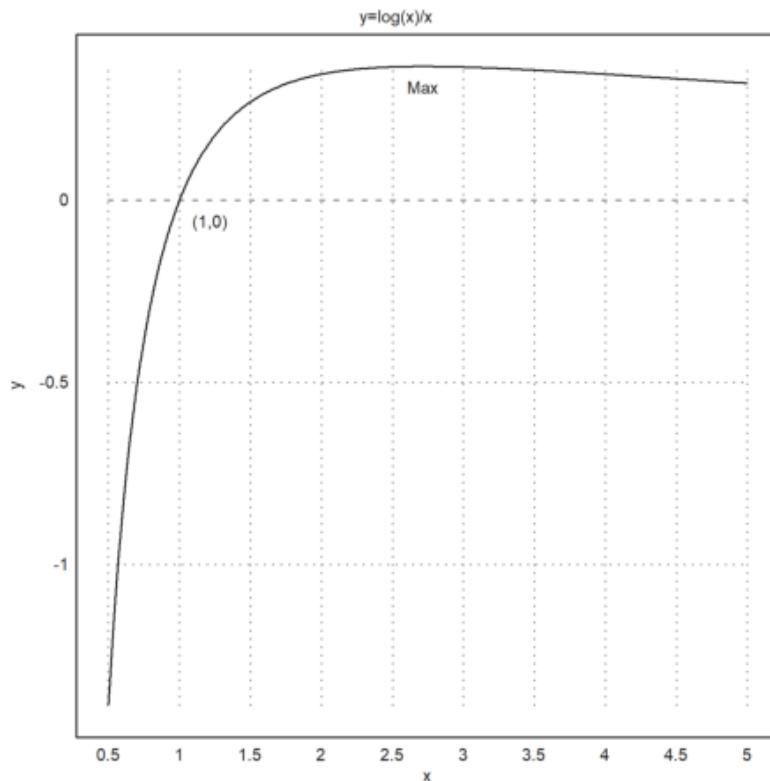


contoh lain

```
>aspect(2); plot2d("x^2+3*x-4", 0, 2, title="y=x^2+3*x-4", yl="y", xl="x") :
```

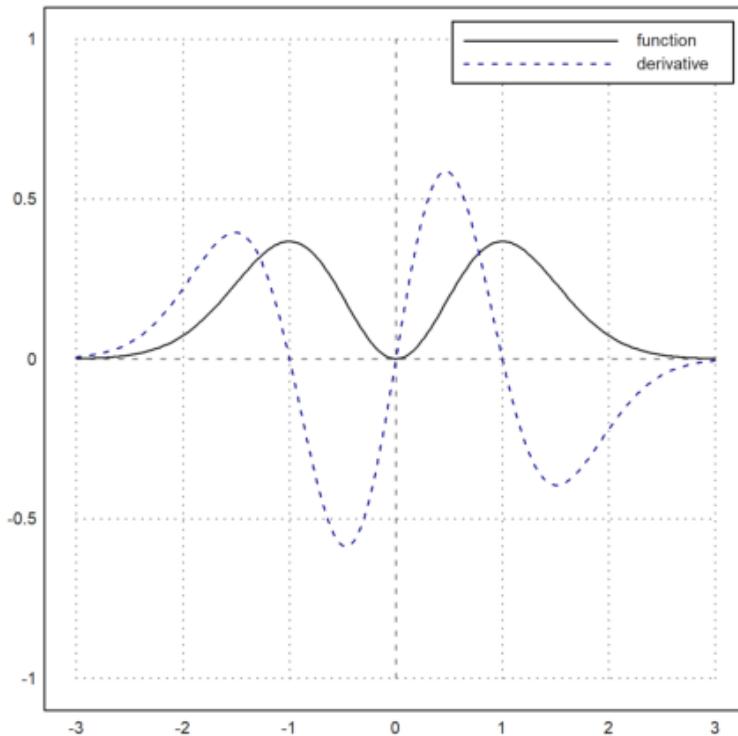


```
>expr := "log(x)/x"; ...
> plot2d(expr, 0.5, 5, title="y="+expr, xl="x", yl="y"); ...
> label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



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

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

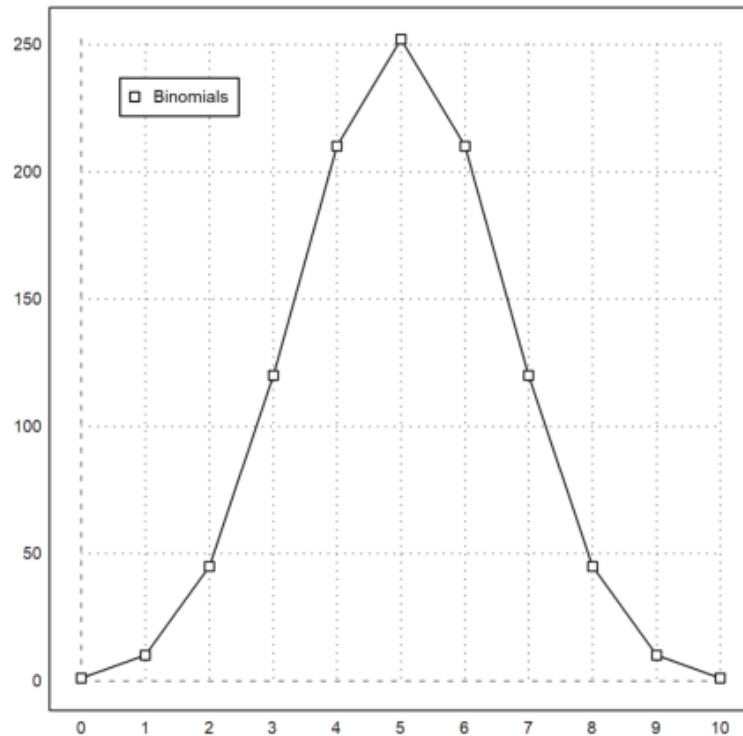


Kotak tersebut berlabuh di kanan atas secara default, tetapi `>left` menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat mana pun yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafik. Lebarnya adalah otomatis.

Untuk plot titik, kotak label juga dapat digunakan. Tambahkan sebuah parameter `>point`, atau sebuah vektor bendera, satuuntuk setiap label.

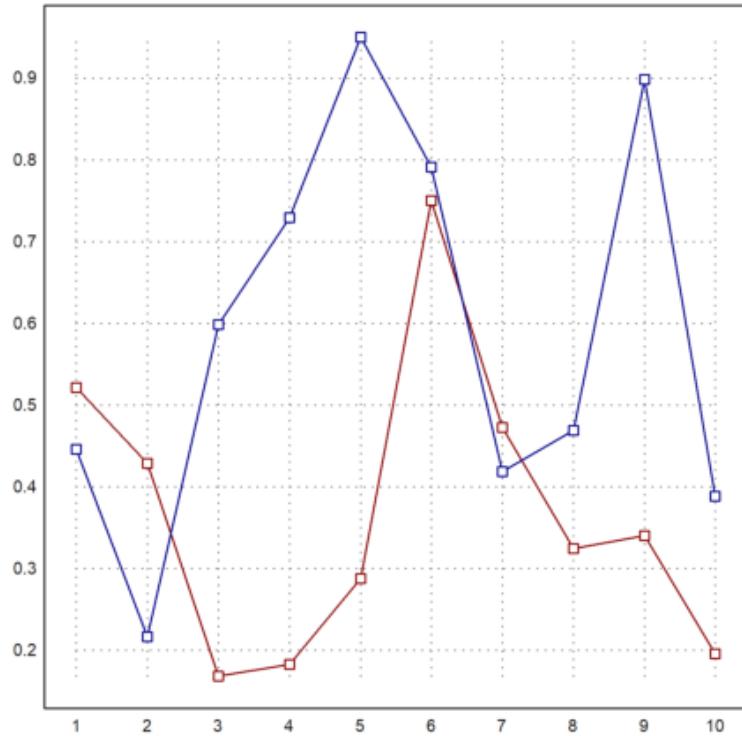
Pada contoh berikut ini, hanya ada satu fungsi. Jadi kita dapat menggunakan string dan bukan vektor string. Kami mengatur warna teks menjadi hitam untuk contoh ini.

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



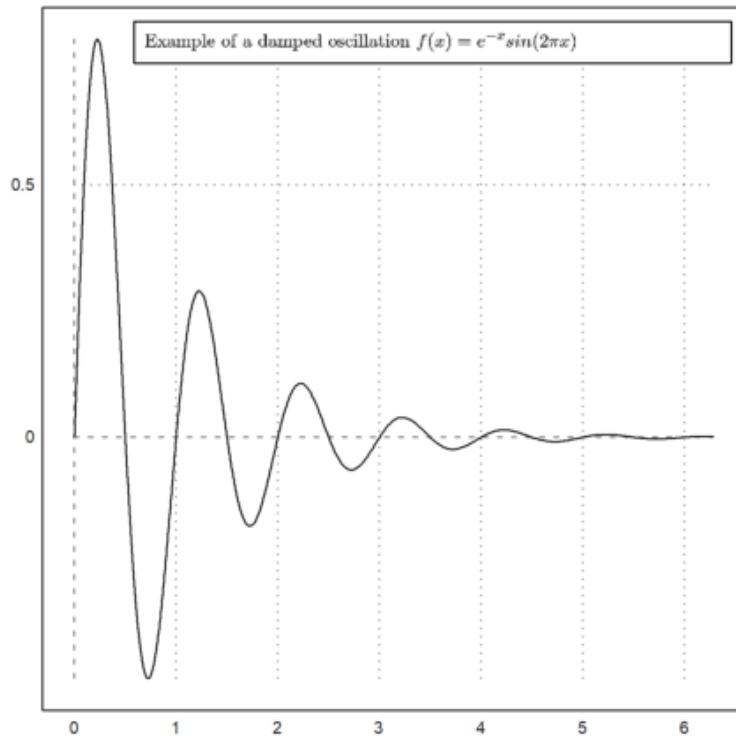
Gaya plot ini juga tersedia di statplot(). Seperti pada plot2d() warna dapat diatur untuk setiap baris plot. Terdapat lebih banyak plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

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



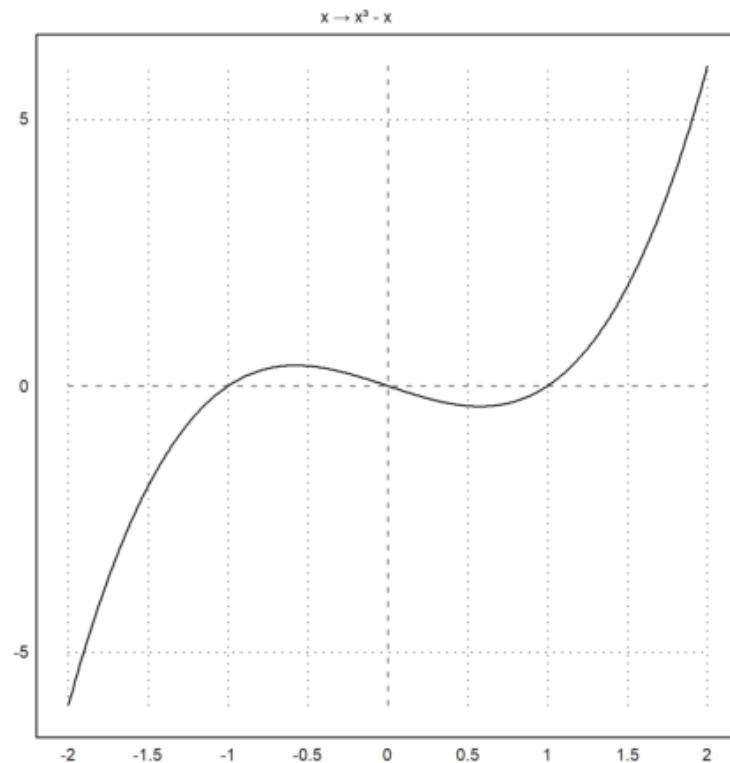
Fitur yang serupa adalah fungsi `textbox()`. Lebarnya secara default adalah lebar maksimal baris teks. Tetapi, ini juga dapat diatur oleh pengguna.

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



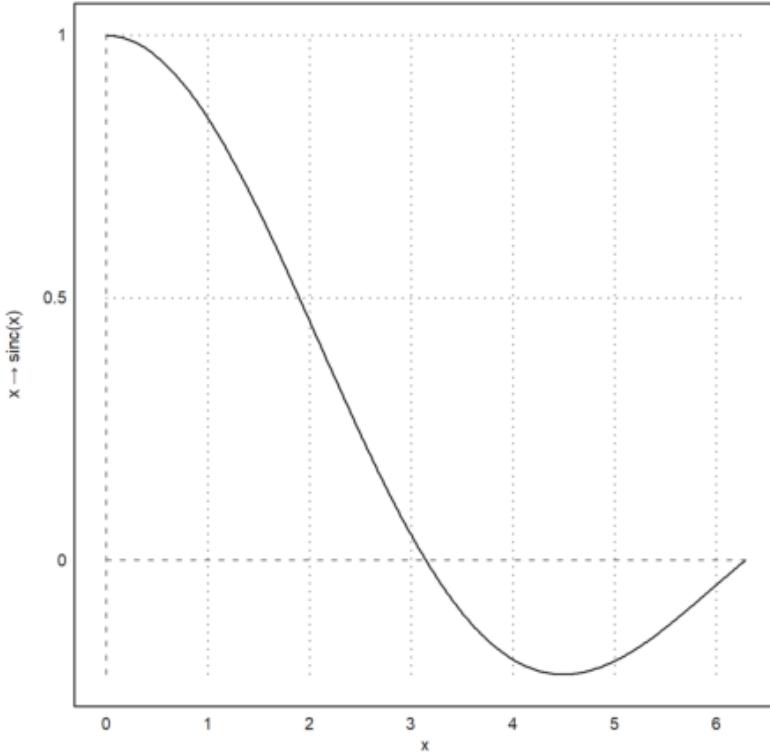
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

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



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

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



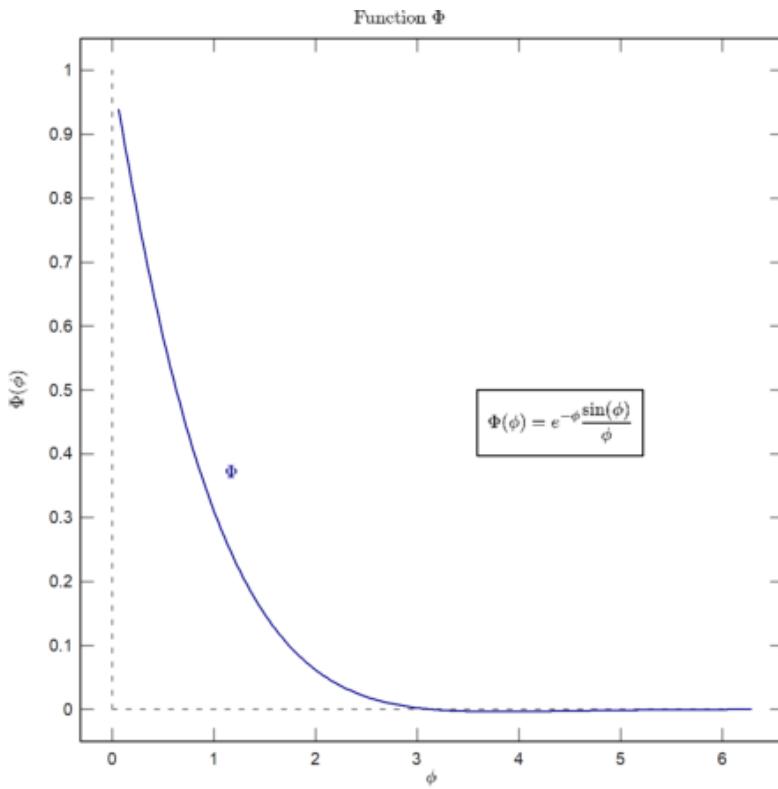
LaTex

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

Perlu diperhatikan bahwa penguraian LaTeX berjalan lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil `latex()` sebelum perulangan sekali dan menggunakan hasilnya (gambar dalam matriks RGB).

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

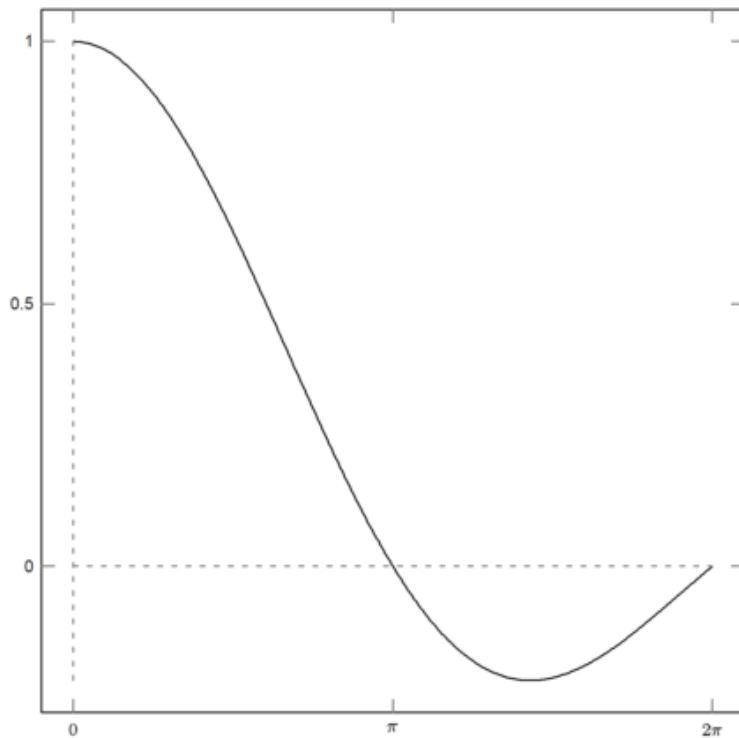
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
> title=latex("\text{Function } \$\Phi\$"), ...
> xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
>textbox( ...
> latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
>label(latex("\Phi",color=blue),1,0.4):
```



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

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

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



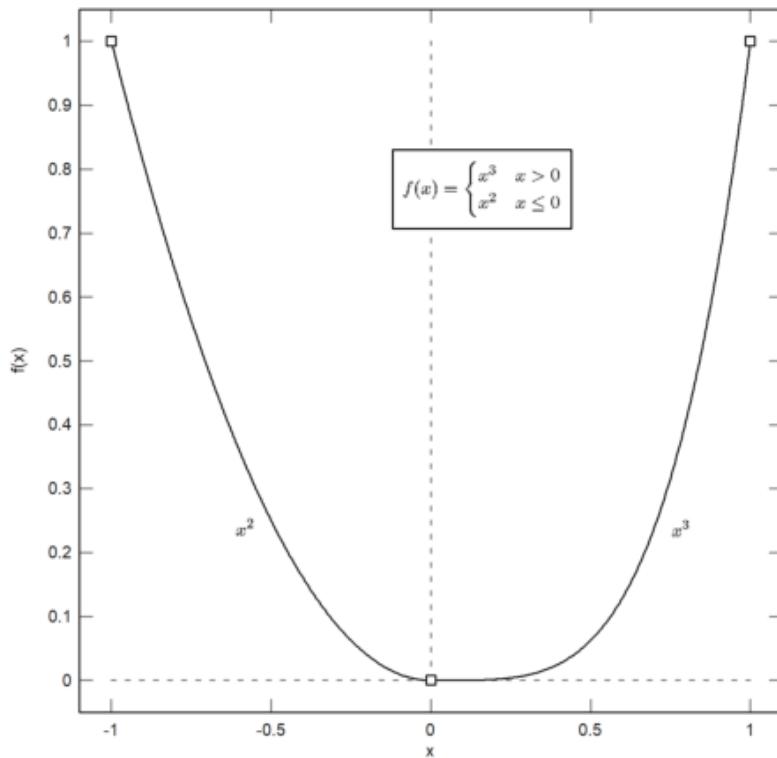
Tentu saja, fungsi juga dapat digunakan.

```
>function map f(x) ...
if x>0 then return x^4
else return x^2
endif
endfunction
```

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

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

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



Interaksi Pengguna

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

- zoom dengan + atau -
- memindahkan plot dengan tombol cursor
- pilih jendela plot dengan mouse
- mengatur ulang tampilan dengan spasi
- keluar dengan kembali

Tombol spasi akan mengatur ulang plot ke jendela plot asli. Saat memplot data, bendera `>user` hanya akan menunggu penekanan tombol.

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

```
Row index 2 out of bounds!
Try "trace errors" to inspect local variables after errors.
plot2d:
  if auto then k2[2]=k1[2]; endif;
```

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

```
Interrupt with Escape key!
```

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

```
plot2d:
```

```
k1=_mouse("Press return, cursor key, +/-, space, click with m ...
```

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

Fungsi bawaan mousedrag() menunggu peristiwa mouse atau keyboard. Fungsi ini melaporkan mouse ke bawah, mouse bergerak atau mouse ke atas, dan penekanan tombol. Fungsi dragpoints() memanfaatkan hal ini, dan mengizinkan pengguna untuk menyeret titik manapun di dalam plot.

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

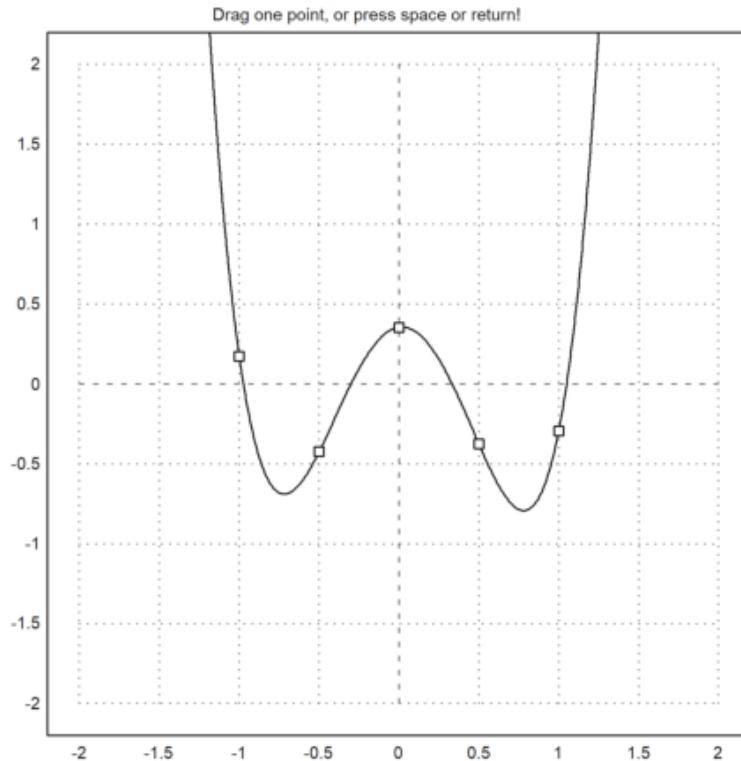
```
>function plotf(xp,yp,select) ...
```

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

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

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

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

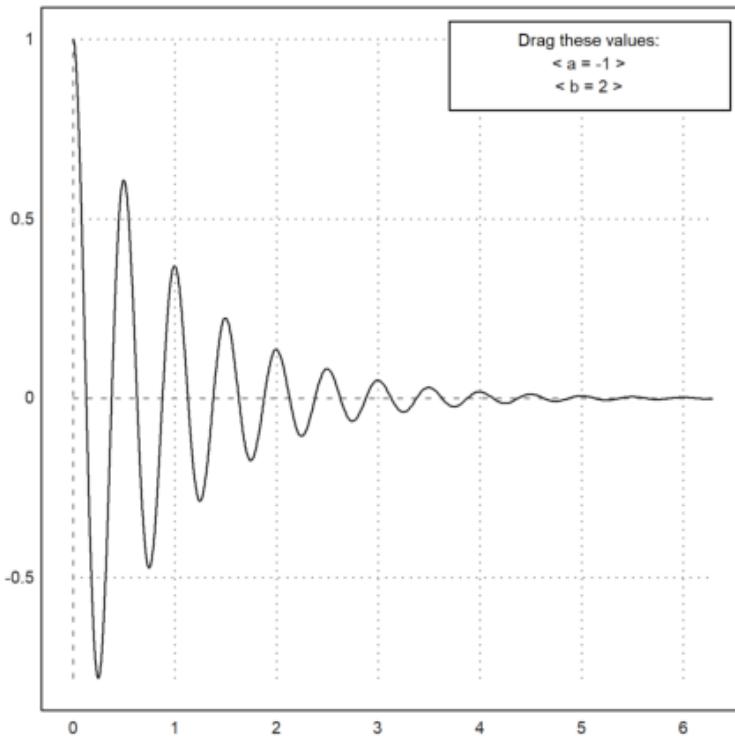


Ada juga fungsi yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini. Pertama, kita memerlukan fungsi plot.

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

Kemudian kita membutuhkan nama untuk parameter, nilai awal dan matriks rentang nx2, dan secara opsional, sebuah garis judul. Terdapat slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

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



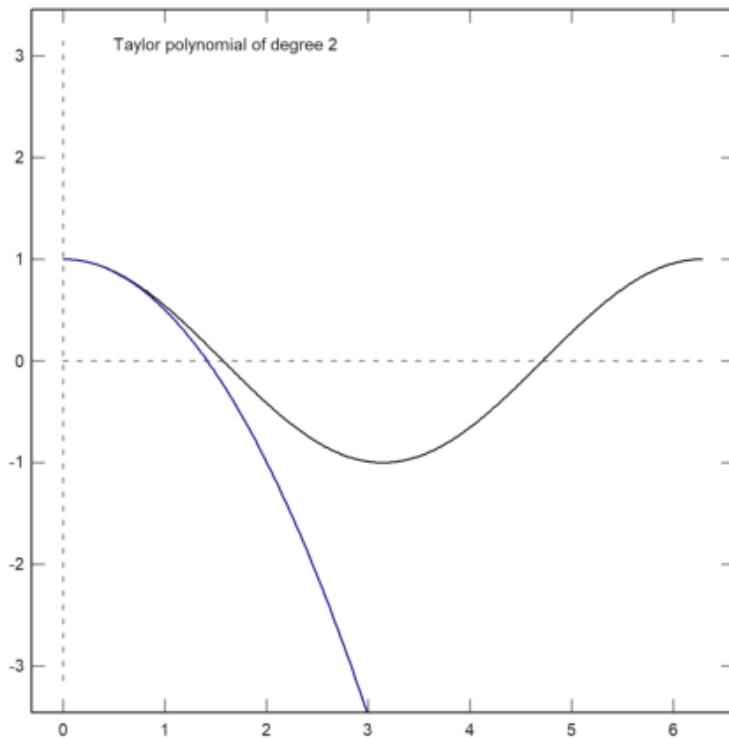
Anda dapat membatasi nilai yang diseret menjadi bilangan bulat. Sebagai contoh, kita menulis fungsi plotf, yang memplot polinomial Taylor dengan derajat n ke fungsi kosinus.

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

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

Sekarang kita membiarkan derajat n bervariasi dari 0 sampai 20 dalam 20 stop. Hasil dari dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk menyisipkan plot ke dalam notebook.

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



Berikut ini adalah demo sederhana dari fungsi ini. Pengguna dapat menggambar di atas jendela plot, meninggalkan jejak titik.

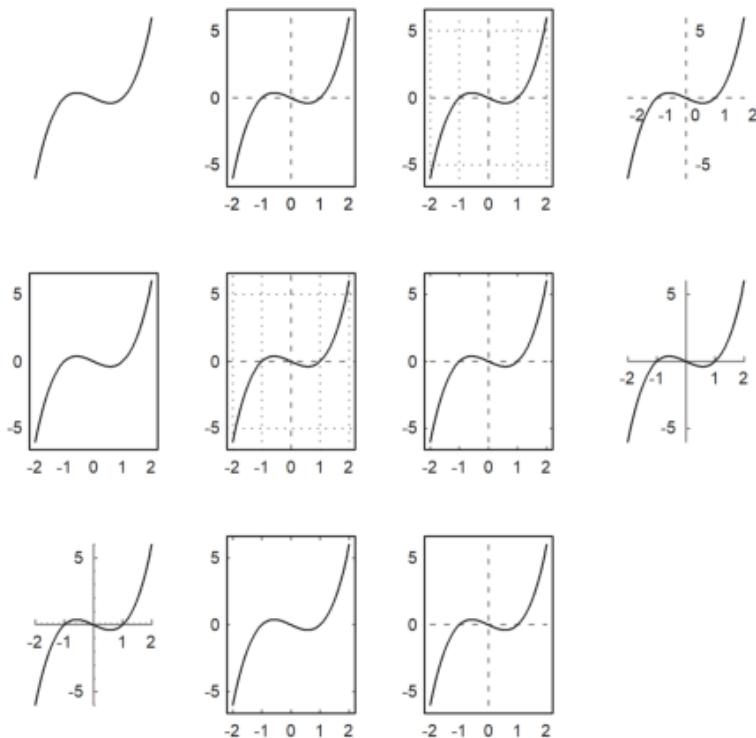
```
>function dragtest ...
```

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

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

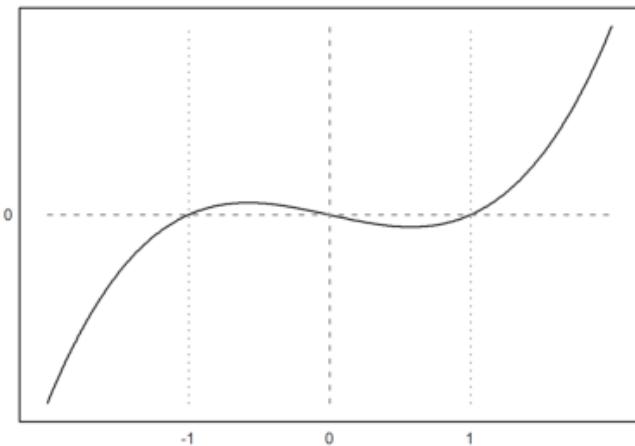
Secara default, EMT menghitung tick sumbu otomatis dan menambahkan label pada setiap tick. Hal ini dapat diubah dengan parameter kisi-kisi. Gaya default sumbu dan label dapat dimodifikasi. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang ke gaya default, gunakan reset()

```
>aspect();
>figure(3,4); ...
> figure(1); plot2d("x^3-x",grid=0); ... // tidak ada grid, bingkai, atau s
>figure(2); plot2d("x^3-x",grid=1); ... // x-y-axis
>figure(3); plot2d("x^3-x",grid=2); ... // default ticks
>figure(4); plot2d("x^3-x",grid=3); ... // sumbu x-y dengan label di dalamn
>figure(5); plot2d("x^3-x",grid=4); ... // tidak ada centang, hanya label
>figure(6); plot2d("x^3-x",grid=5); ... // default, tidak ada margin
>figure(7); plot2d("x^3-x",grid=6); ... // hanya sumbu
>figure(8); plot2d("x^3-x",grid=7); ... // hanya sumbu, centang pada sumbu
>figure(9); plot2d("x^3-x",grid=8); ... // hanya sumbu, centang lebih halus
>figure(10); plot2d("x^3-x",grid=9); ... // default, centang kecil didalamn
>figure(11); plot2d("x^3-x",grid=10); ....// tidak ada centang, hanya sumbu
>figure(0):
```



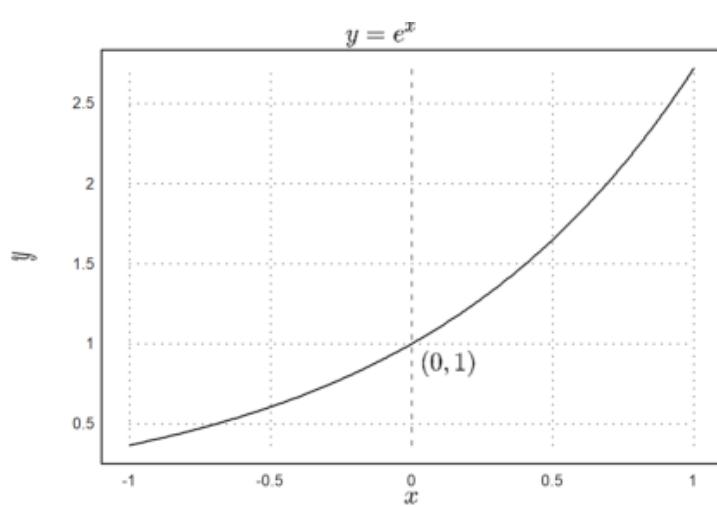
Parameter `<frame>` mematikan bingkai, dan `framecolor=blue` menetapkan bingkai ke warna biru. Jika Anda menginginkan tanda centang Anda sendiri, Anda dapat menggunakan `style=0`, dan menambahkan semuanya nanti.

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



Untuk judul plot dan label sumbu, lihat contoh berikut.

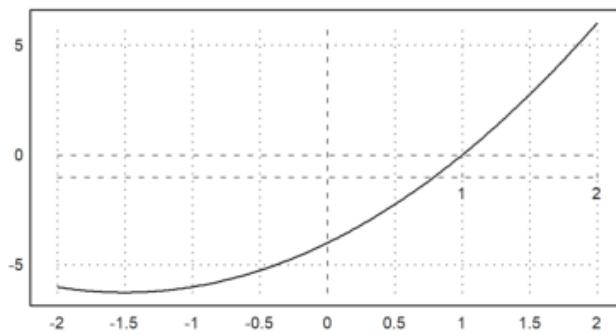
```
>plot2d("exp(x)",-1,1);
>textcolor(black); // set warna teks menjadi hitam
>title(latex("y=e^x")); // judul diatas plot
>xlabel(latex("x")); // "x" untuk sumbu x
>ylabel(latex("y"),>vertical); // vertikal "y" untuk sumbu y
>label(latex("(0,1)'),0,1,color=blue): // label titik a
```



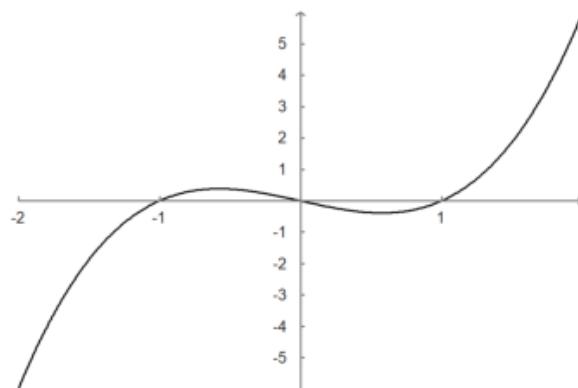
Sumbu dapat digambar secara terpisah dengan sumbu $x()$ dan sumbu $y()$.

contoh lain

```
>plot2d("x^2+3*x-4") ;  
>xaxis(-1,xx=0:5);  
>yaxis(0,yy=2:1) :
```



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

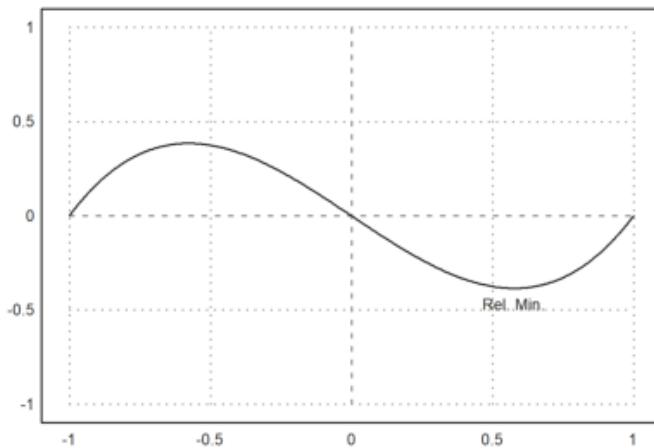


Teks pada plot dapat diatur dengan `label()`. Pada contoh berikut ini, "lc" berarti lower center. Ini mengatur posisi label relatif terhadap koordinat plot.

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

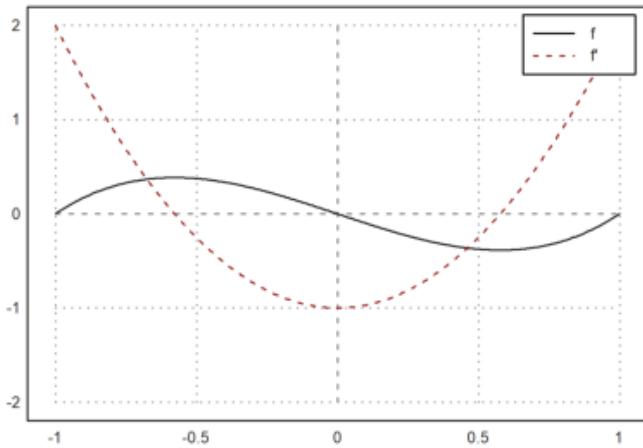
$$x^3 - x$$

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

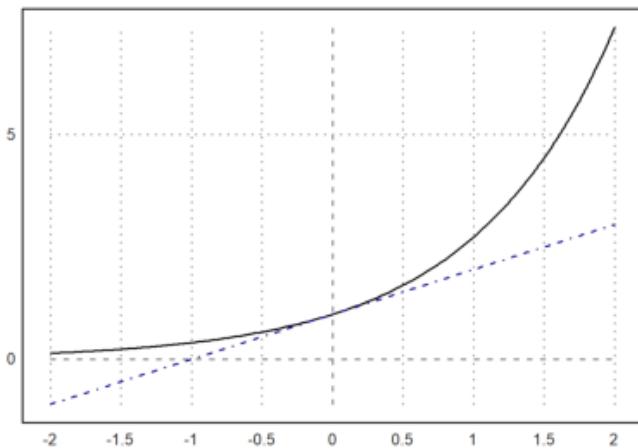


Terdapat juga kotak teks

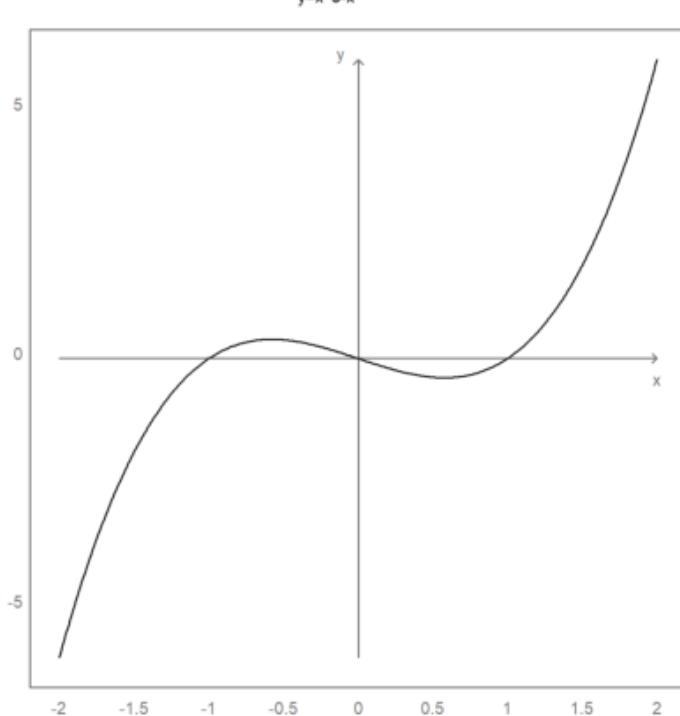
```
>plot2d(&f(x),-1,1,-2,2); // fungsi
>plot2d(&diff(f(x),x),>add,style="--",color=red); // turunan
>labelbox(["f","f'"],["-", "--"],[black,red]): // kotak label
```



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

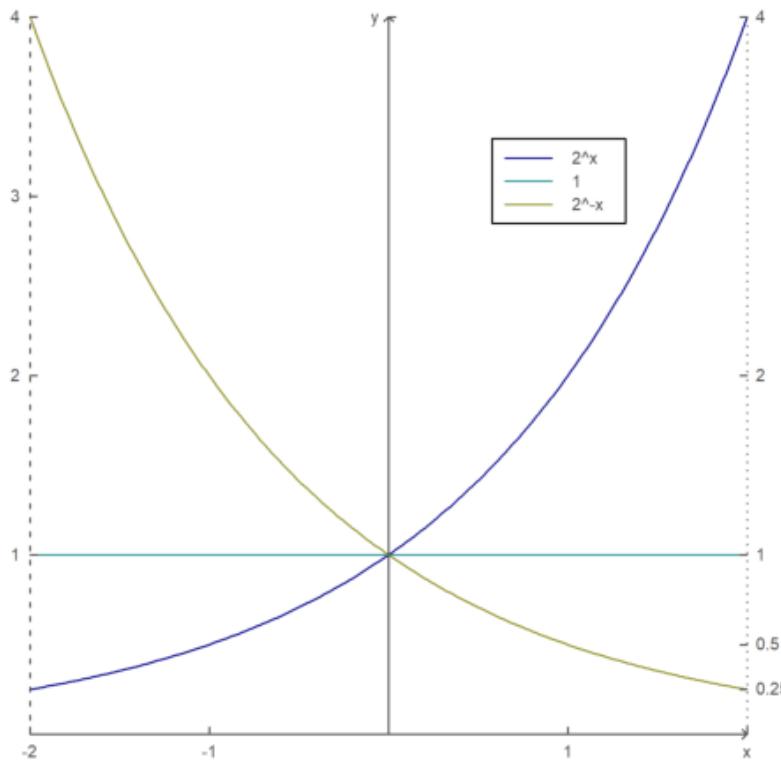


```
>gridstyle("->", color=gray, textcolor=gray, framecolor=gray); ...
> plot2d("x^3-x", grid=1); ...
> setttitle("y=x^3-x", color=black); ...
> label("x", 2, 0, pos="bc", color=gray); ...
> label("y", 0, 6, pos="cl", color=gray); ...
> reset():
```



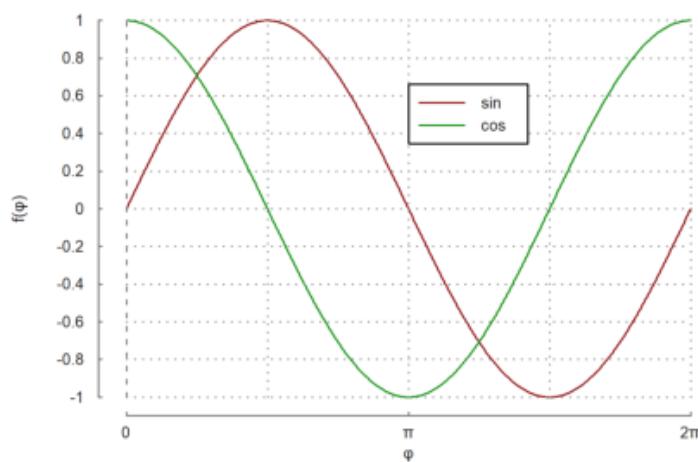
Untuk kontrol yang lebih banyak lagi, sumbu x dan sumbu y dapat dilakukan secara manual. Perintah `fullwindow()` memperluas jendela plot karena kita tidak lagi membutuhkan tempat untuk label di luar jendela plot. Gunakan `shrinkwindow()` atau `reset()` untuk mengatur ulang ke default.

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



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

```
>aspect(1.5);
>plot2d(["sin(x)", "cos(x")], 0, 2pi, color=[red, green], <grid, <frame); ...
> xaxis(-1.1, (0:2)*pi, xt=["0", u"\u03c0", "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)"):
```



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

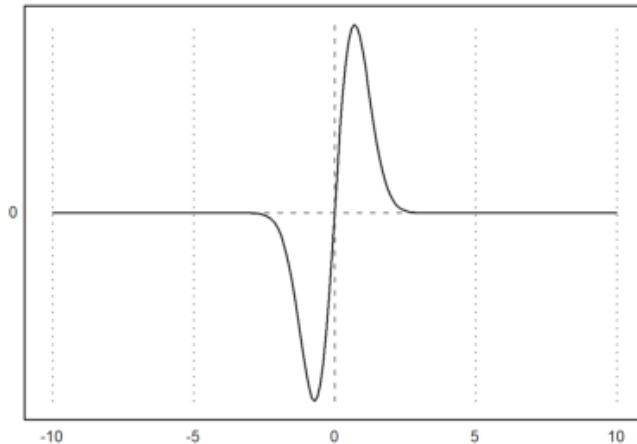
Memplot ekspresi hanyalah singkatan dari plot data. Untuk plot data, Anda memerlukan satu atau lebih baris nilai x , dan satu atau lebih baris nilai y . Dari rentang dan nilai x , fungsi `plot2d` akan menghitung data untuk diplot, secara default dengan evaluasi adaptif dari fungsi tersebut. Untuk plot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

Tetapi Anda dapat memasukkan data secara langsung.

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

Berikut adalah contoh dengan satu baris untuk x dan y .

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

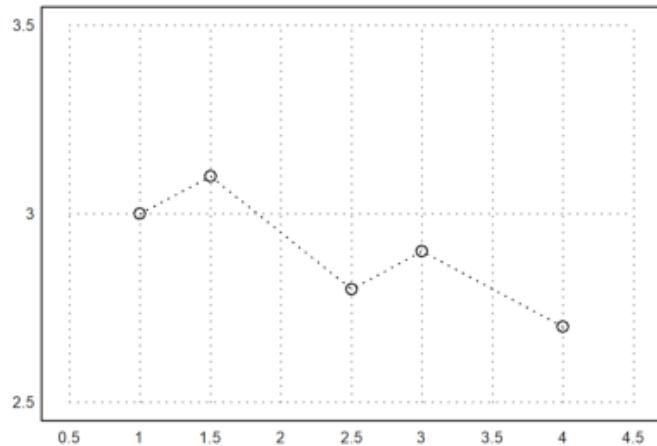


Data juga dapat diplot sebagai titik. Gunakan `poin=true` untuk ini. Plot ini berfungsi seperti poligon, namun hanya menggambar sudut-sudutnya saja.

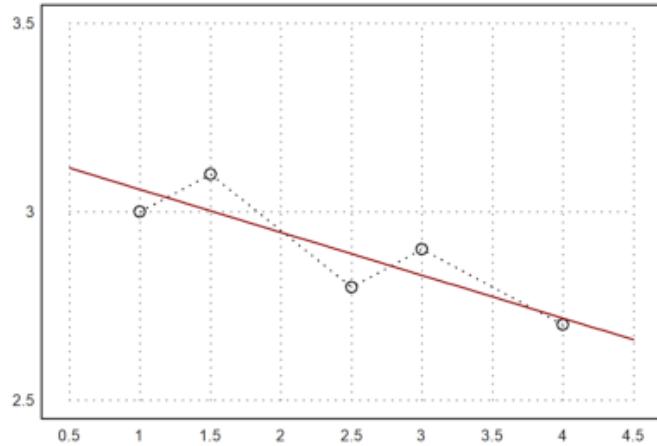
`- style = "...":` Pilih dari "[]", "<>", "o", ".", "..", "+", "*", "[]", "<>", "o", ".", "..", "|".

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

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



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



Menggambar Daerah Yang Dibatasi Kurva

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

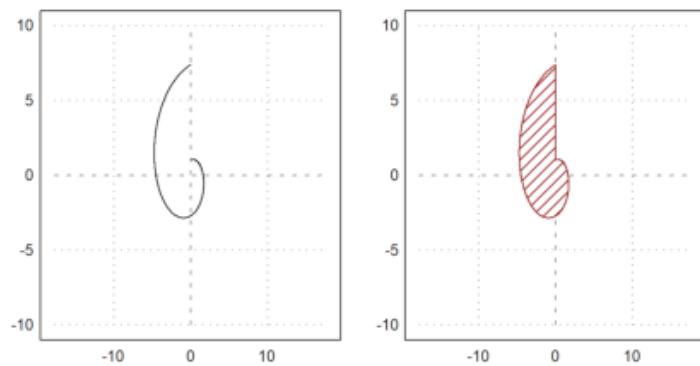
- filled = true mengisi plot.

- style = "...": Pilih dari "", "/", "\", "\", "/".

- fillcolor: Lihat di atas untuk warna yang tersedia.

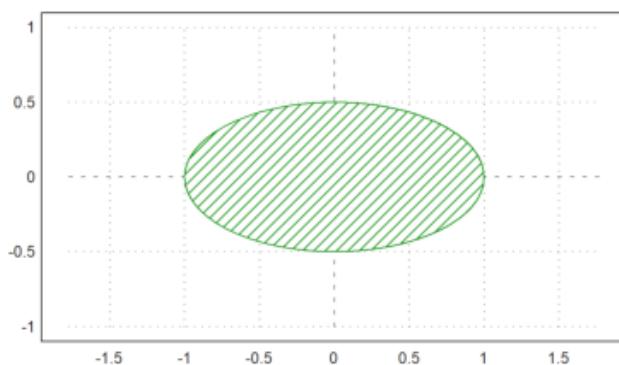
Warna isian ditentukan oleh argumen "fillcolor", dan pada opsional <outline mencegah menggambar garis batas untuk semua gaya kecuali yang default

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

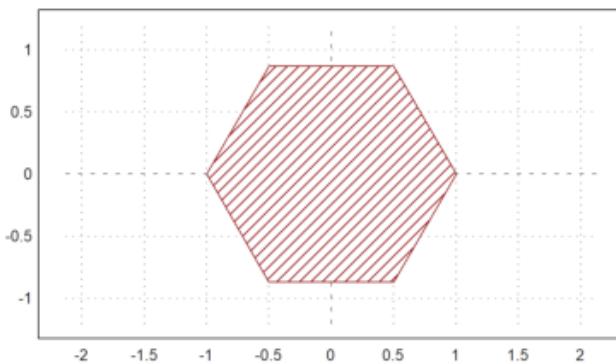


Pada contoh berikut ini, kami memplot elips yang terisi dan dua segi enam yang terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian yang berbeda.

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

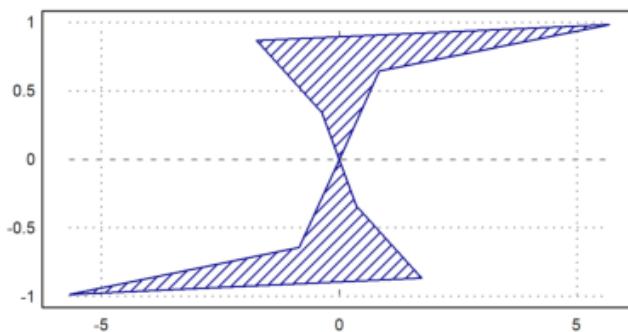


```
>t=linspace(0,2pi,6); ...
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.2):
```

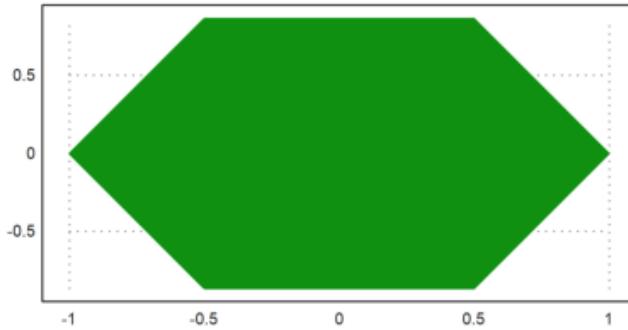


contoh lain

```
>x=linspace(0,2pi,9); plot2d(tan(x),sin(x),>filled,style="/",fillcolor=blue)
```

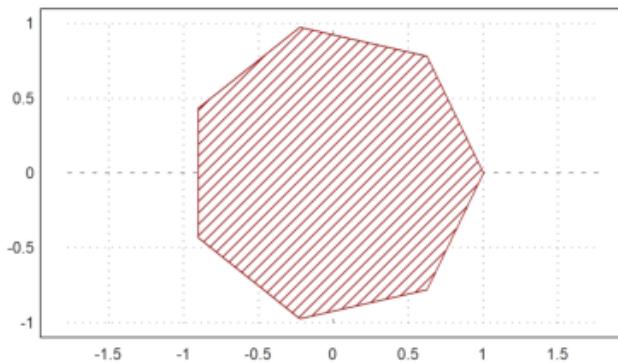


```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



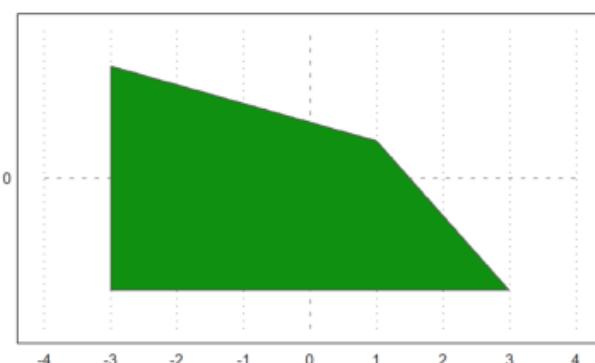
Contoh lainnya adalah septagon, yang kita buat dengan 7 titik pada lingkaran satuan.

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



Berikut ini adalah himpunan nilai maksimal dari empat kondisi linier yang kurang dari atau sama dengan 3. Ini adalah $A[k].v \leq 3$ untuk semua barisan A . Untuk mendapatkan sudut-sudut yang bagus, kita menggunakan n yang relatif besar.

```
>A=[2,1;1,2;-1,0;0,-1];  
>function f(x,y) := max([x,y].A');  
>plot2d("f",r=4,level=[0;3],color=green,n=111):
```

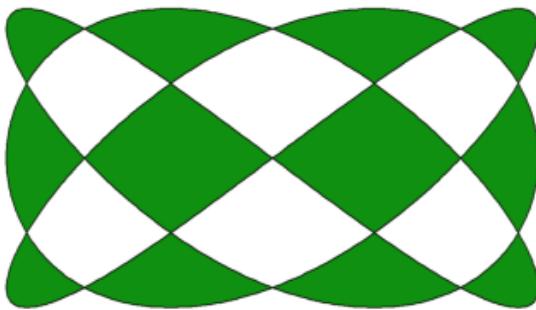


Poin utama dari bahasa matriks adalah bahwa bahasa ini memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

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

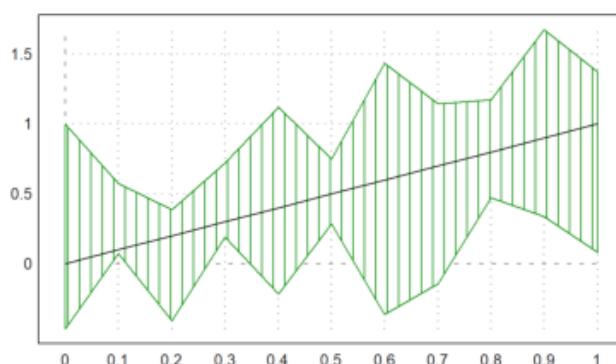
Kita sekarang memiliki vektor nilai x dan y. `plot2d()` dapat memplot nilai-nilai ini sebagai sebuah kurva yang menghubungkan titik-titik. Plot dapat diisi. Dalam kasus ini, hal ini memberikan hasil yang bagus karena aturan penggulungan, yang digunakan untuk isi.

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



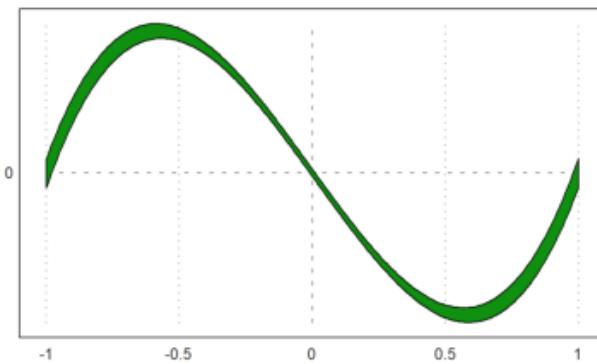
Vektor interval diplot terhadap nilai x sebagai wilayah yang terisi antara nilai bawah dan atas interval. Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tetapi juga dapat digunakan untuk memplot kesalahan statistik.

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



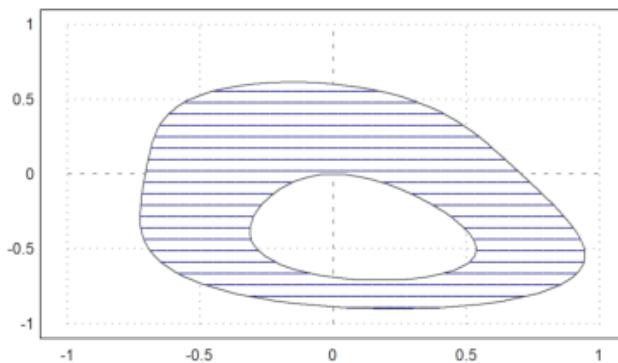
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka `plot2d` akan memplot rentang interval yang terisi di bidang, gaya isian sama dengan gaya poligon.

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



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

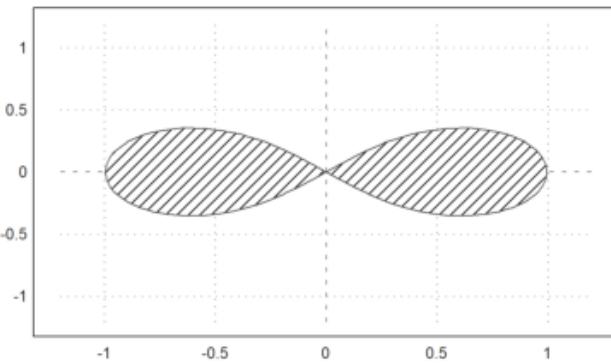
```
>expr := "2*x^2+x*y+3*y^4+y"; // mendefinisikan persamaan f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



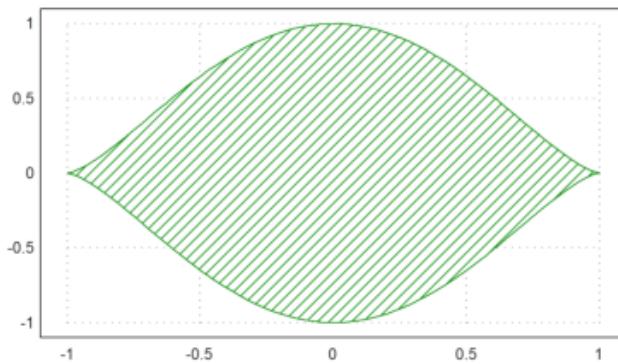
Kita juga dapat mengisi rentang nilai seperti

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

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



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



Grafik Fungsi Parametrik

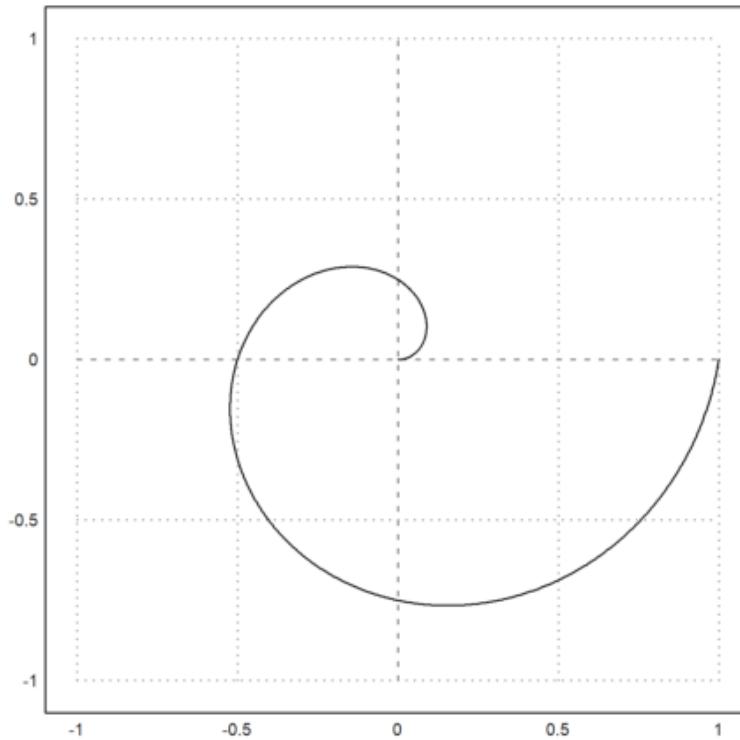
Nilai x tidak perlu diurutkan. (x,y) hanya menggambarkan sebuah kurva. Jika x diurutkan, kurva tersebut adalah grafik dari sebuah fungsi.

Dalam contoh berikut, kami memplot spiral

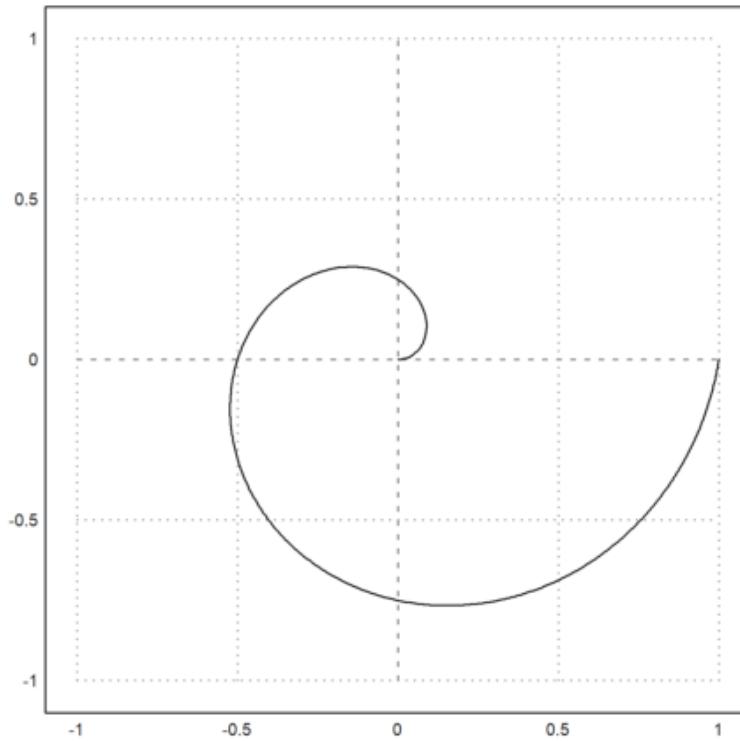
$$\gamma(t) = t \cdot (\cos(2\pi t), \sin(2\pi t))$$

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

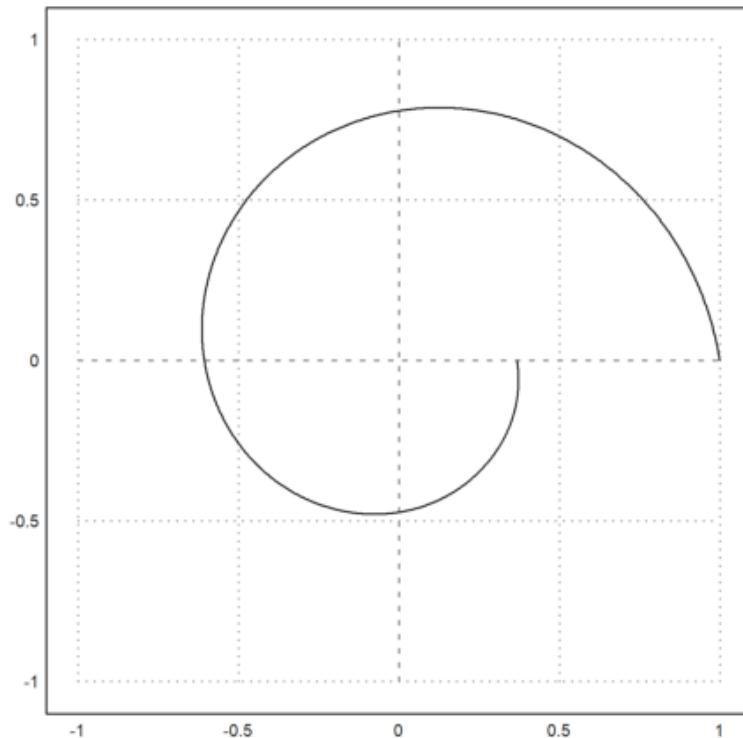
```
>t=linspace(0,1,1000); ...
>plot2d(t*cos(2*pi*t),t*sin(2*pi*t),r=1):
```



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



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



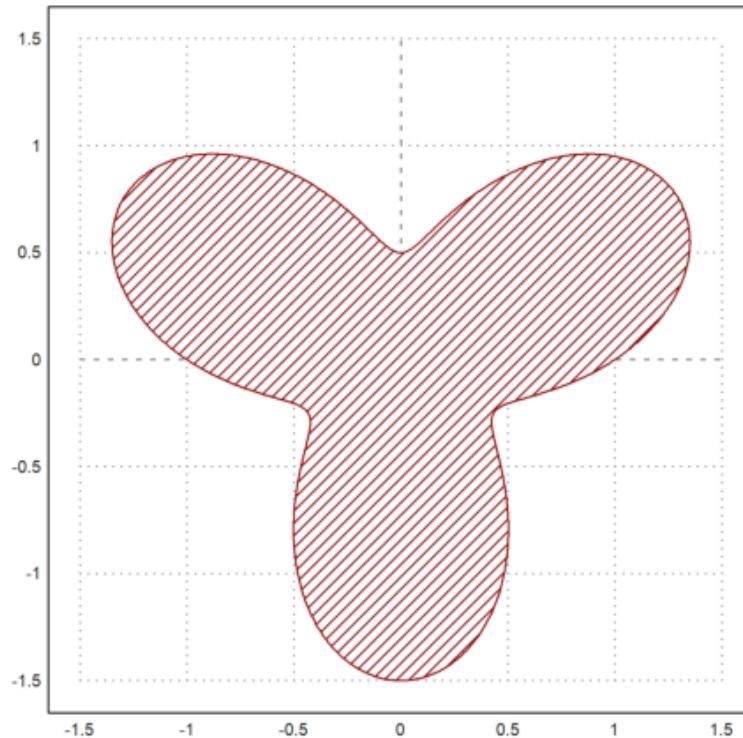
Dalam contoh berikut, kami memplot kurva

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

dengan

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

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



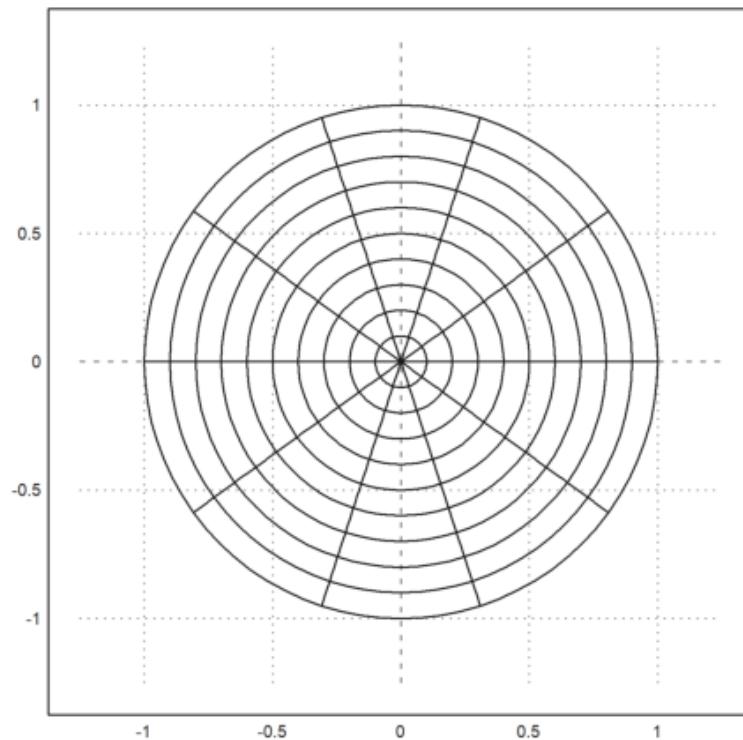
Menggambar Grafik Bilangan Kompleks

Larik bilangan kompleks juga dapat diplot. Kemudian titik-titik kisi akan dihubungkan. Jika sejumlah garis kisi ditentukan (atau vektor 1×2 garis kisi) pada argumen cgrid, hanya garis-garis kisi tersebut yang akan terlihat.

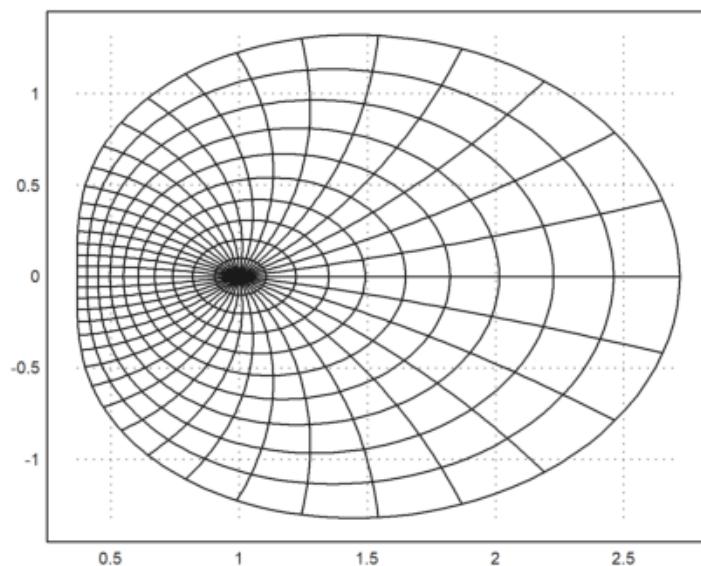
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi-kisi pada bidang kompleks.

Pada contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva kisi-kisi.

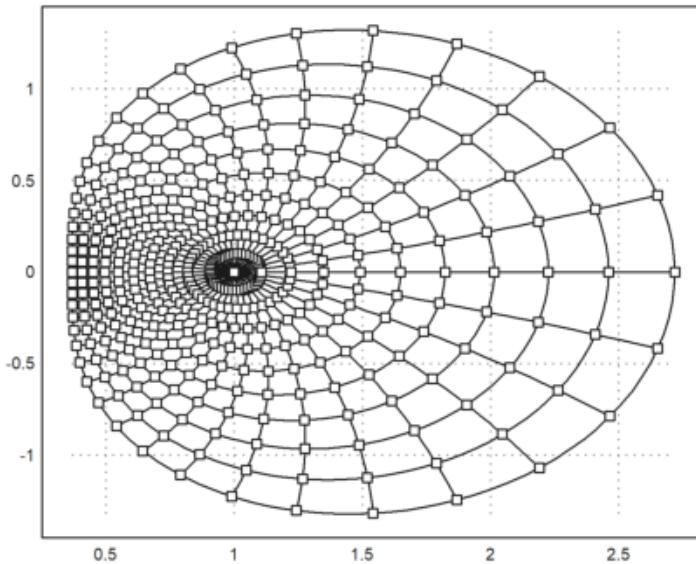
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```



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



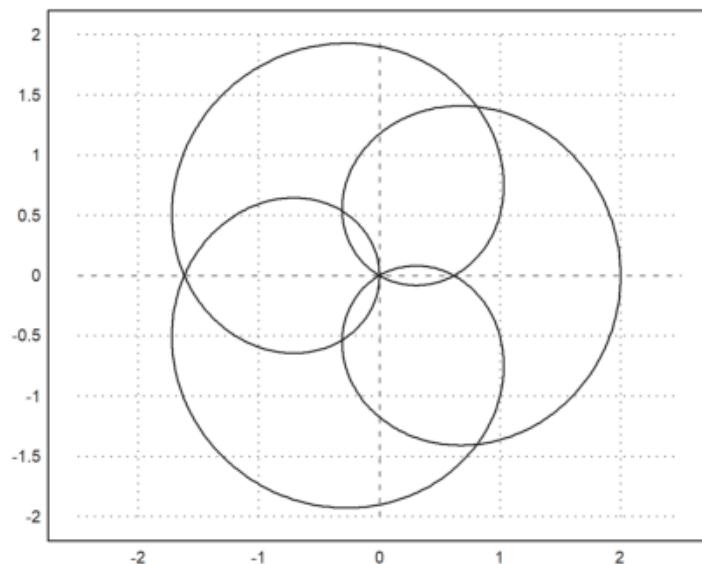
```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```



Vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian riil dan bagian imajiner. Dalam contoh, kami memplot lingkaran satuan dengan

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

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



Terdapat banyak fungsi yang diperuntukkan pada plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

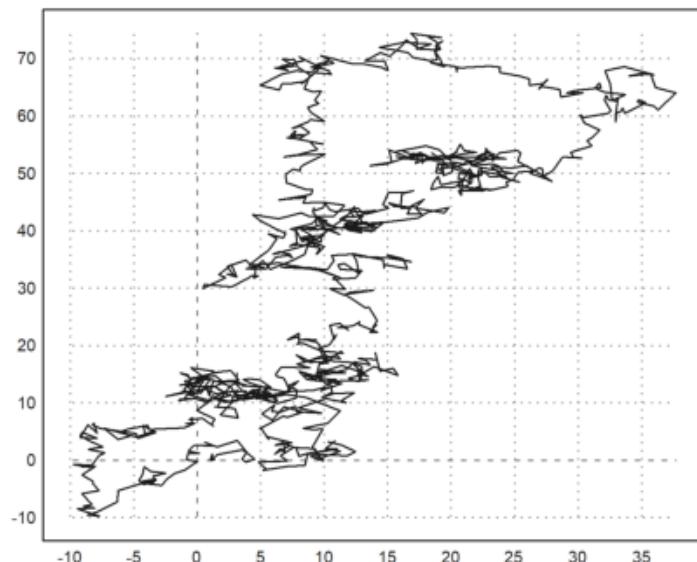
Jumlah kumulatif dari nilai berdistribusi normal 0-1 menghasilkan jalan acak.

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

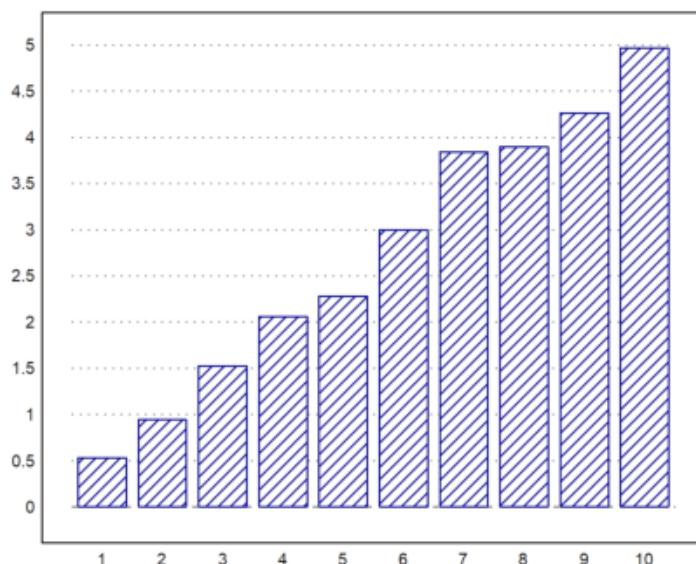


Dengan menggunakan dua baris, ini menunjukkan cara dalam dua dimensi

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

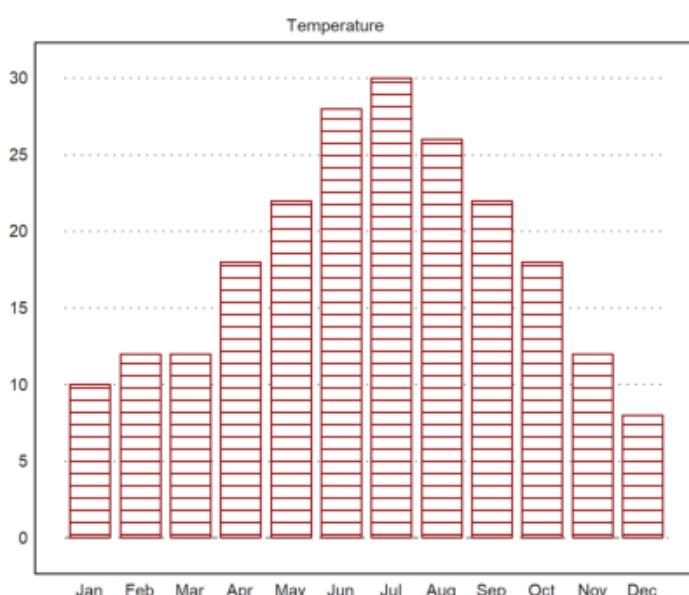


```
>columnsplot(cumsum(random(10)), style="/", color=blue) :
```

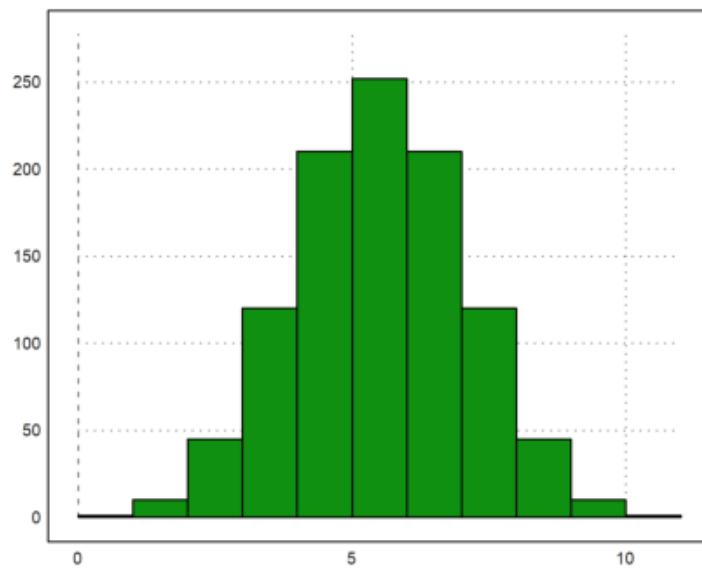


Ini juga dapat menampilkan string sebagai label

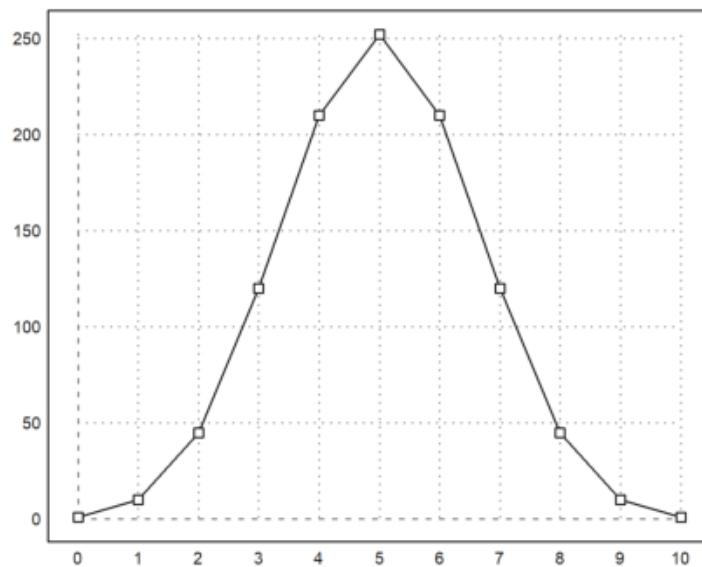
```
>months=["Jan", "Feb", "Mar", "Apr", "May", "Jun", ...
> "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values, lab=months, color=red, style="-");
>title("Temperature"):
```



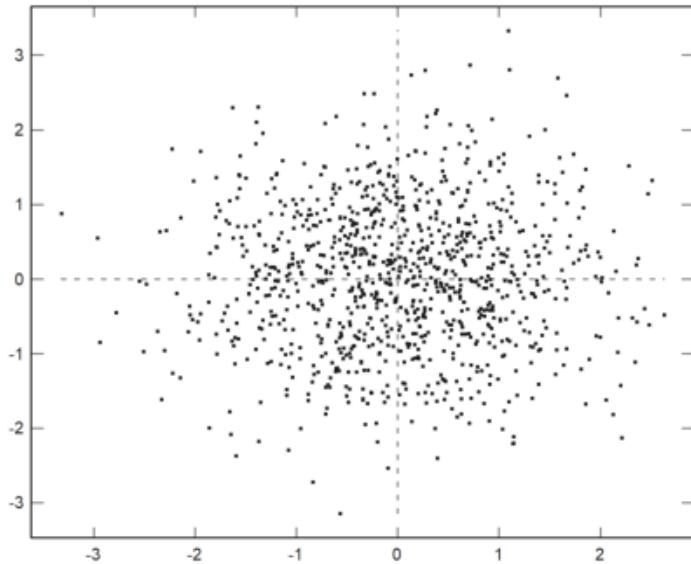
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



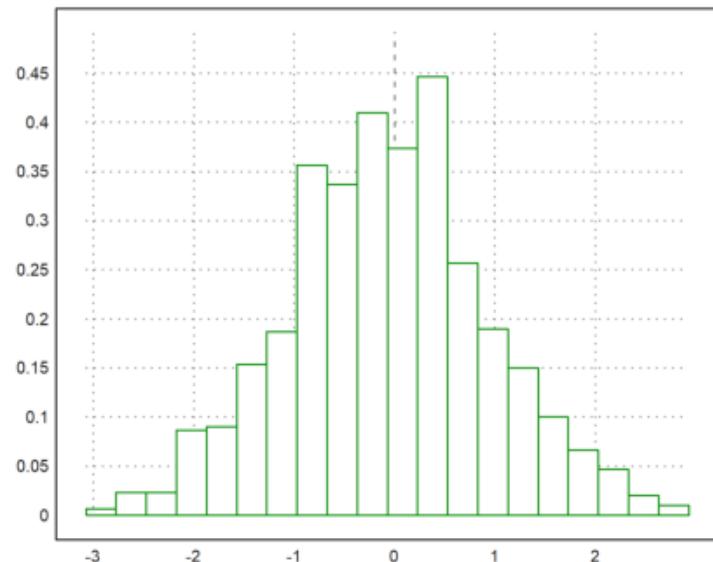
```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



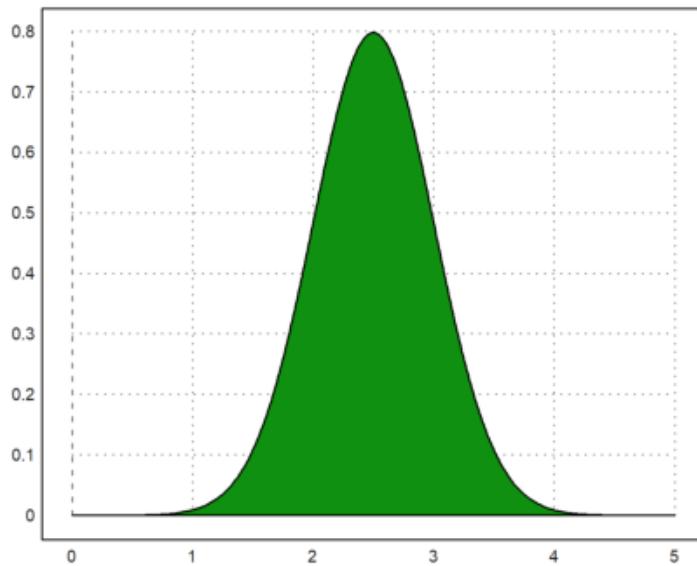
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=". . ."):
```



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

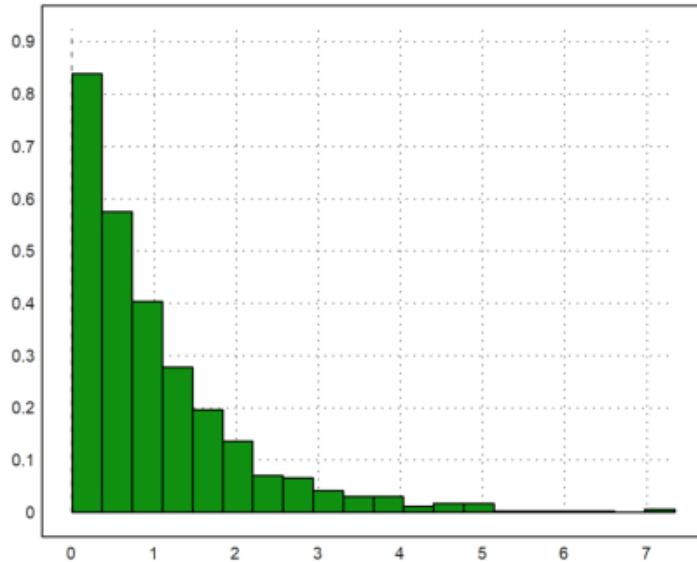


```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



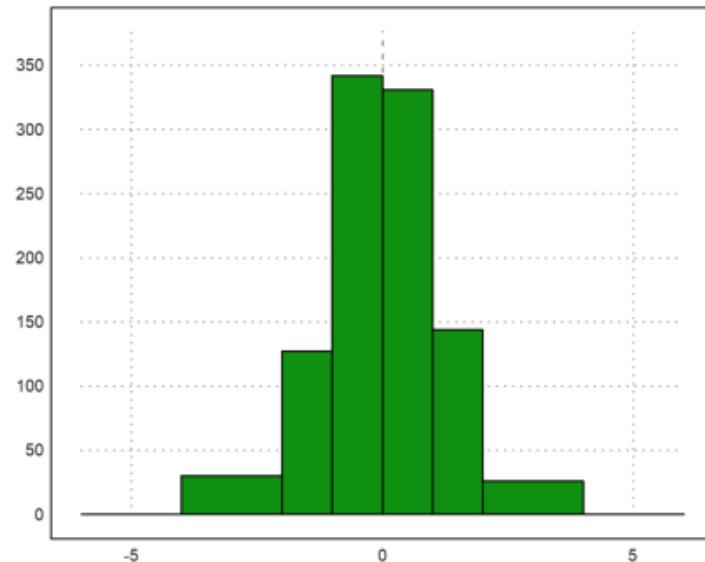
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

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



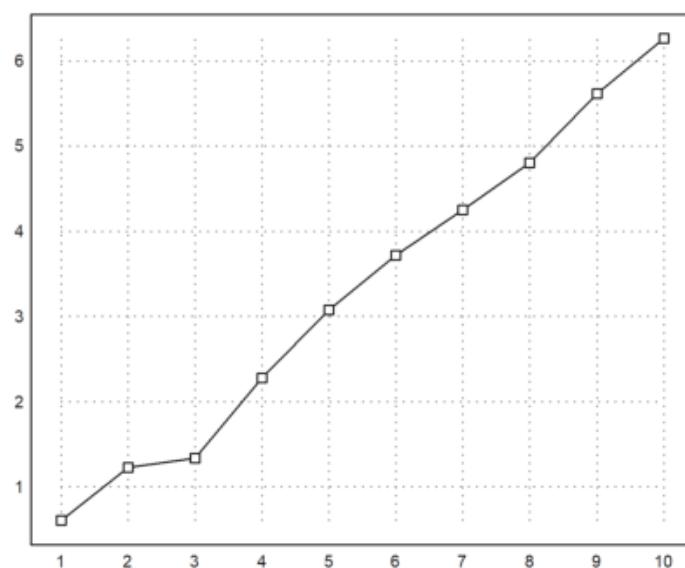
Atau dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom

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

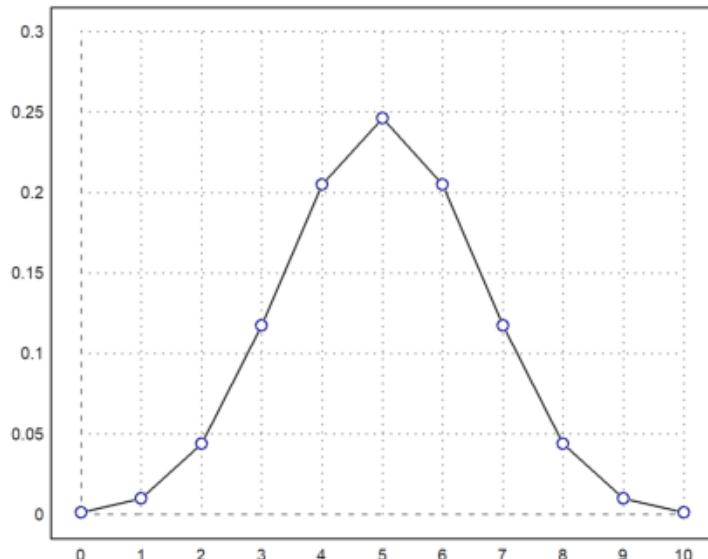


Fungsi statplot() menetapkan gaya dengan string sederhana

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



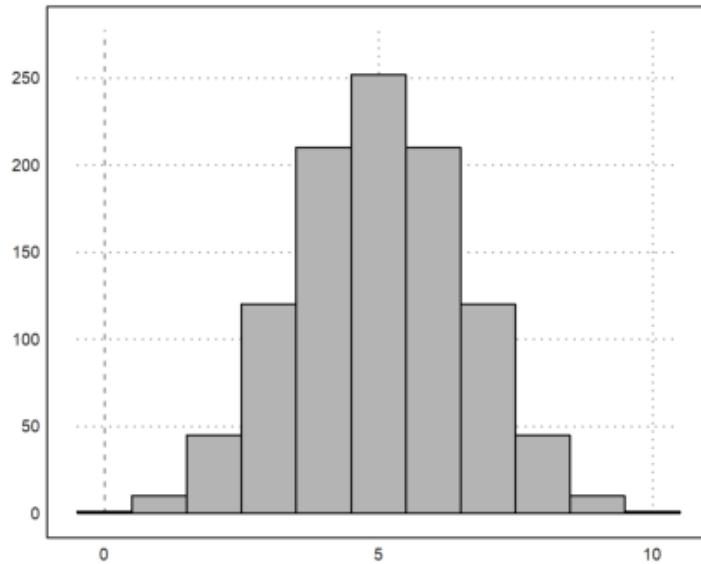
```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



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

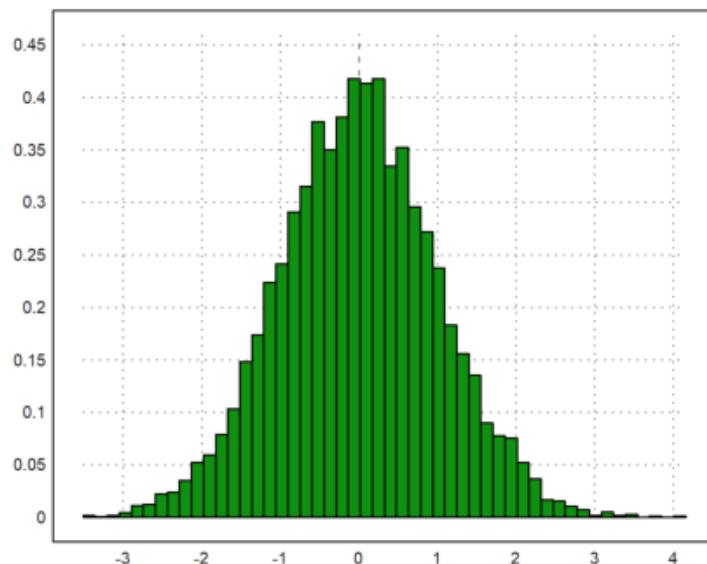
Fill styles dapat digunakan seperti di atas.

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

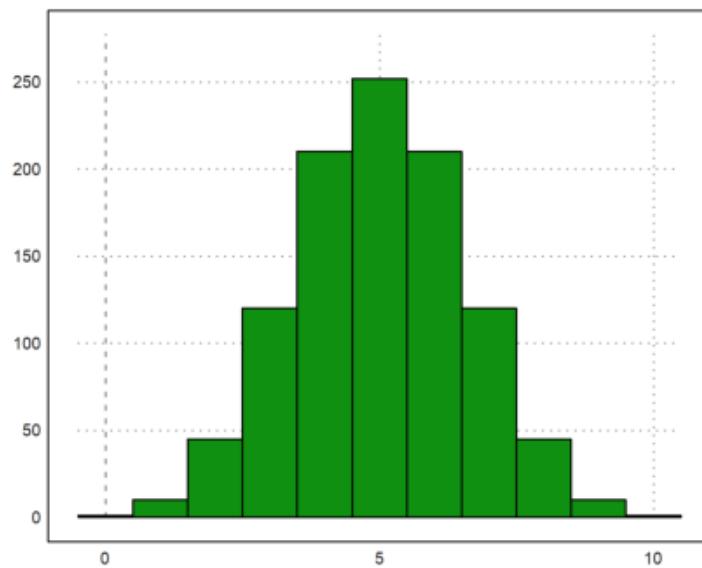


Data untuk plot batang (bar = 1) dan histogram (histogram = 1) dapat diberikan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribution (atau distribusi = n). Histogram dari nilai xv akan dihitung secara otomatis dengan >histogram. Jika >even ditentukan,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

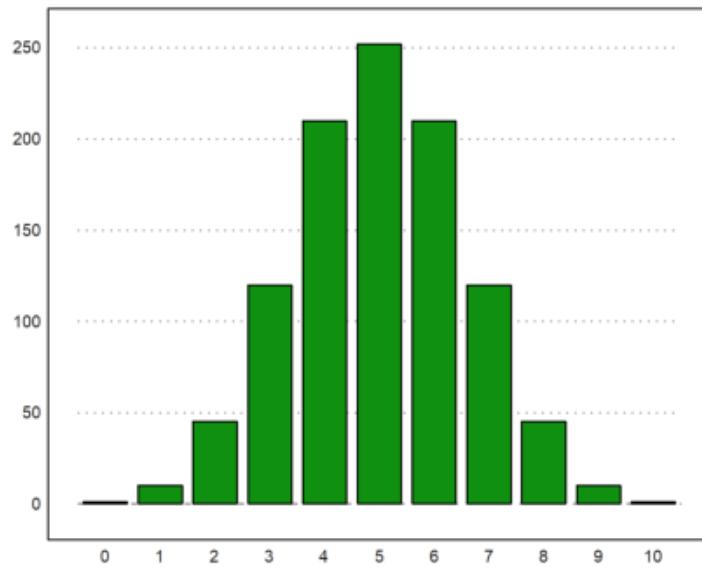
```
>plot2d(normal(10000),distribution=50):
```



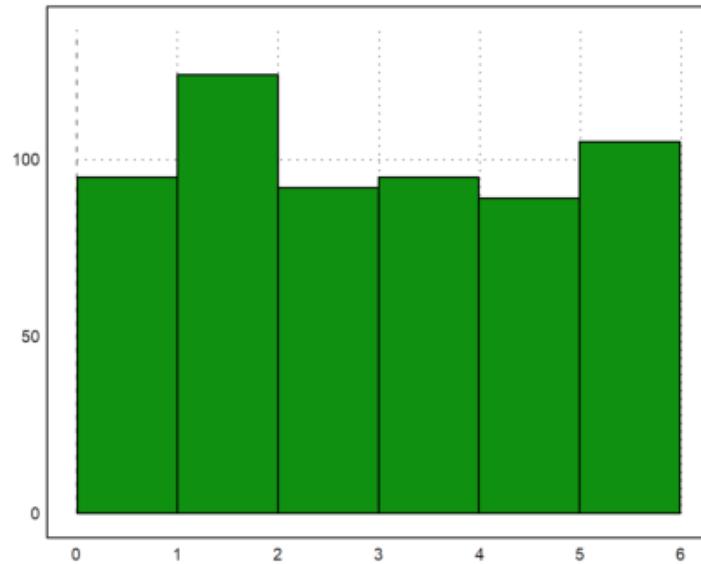
```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```



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

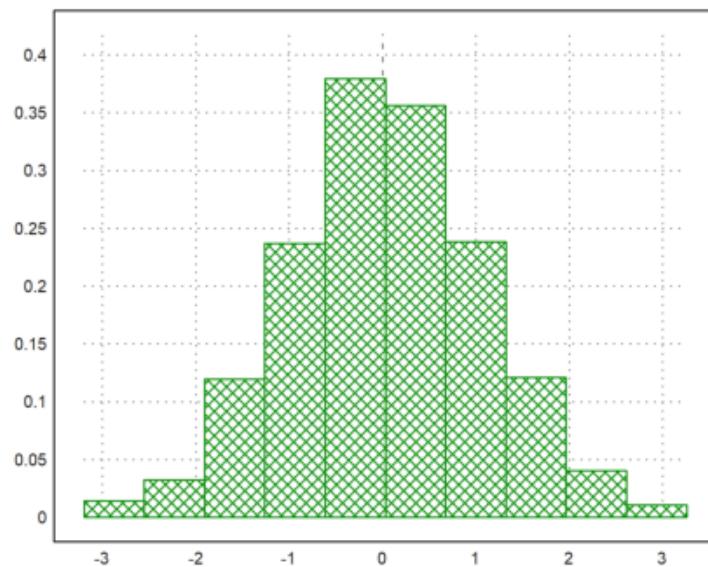


```
>plot2d(random(600)*6,histogram=6) :
```



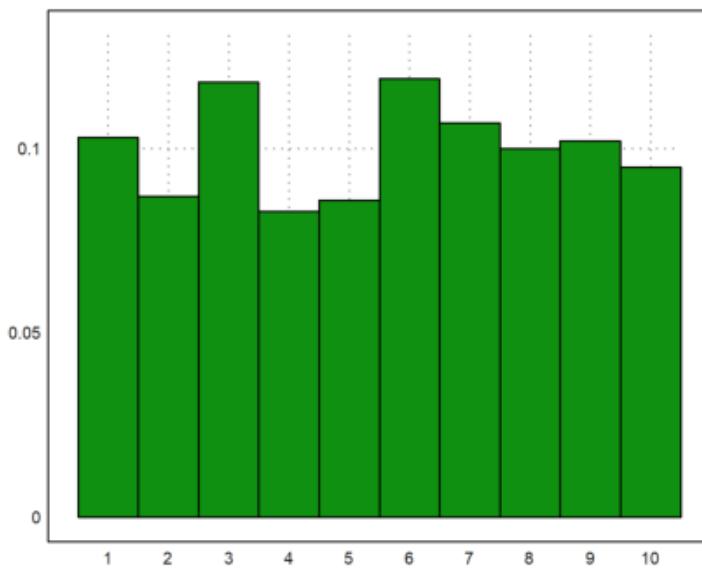
Untuk distribusi, ada parameter `distribution=n`, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

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



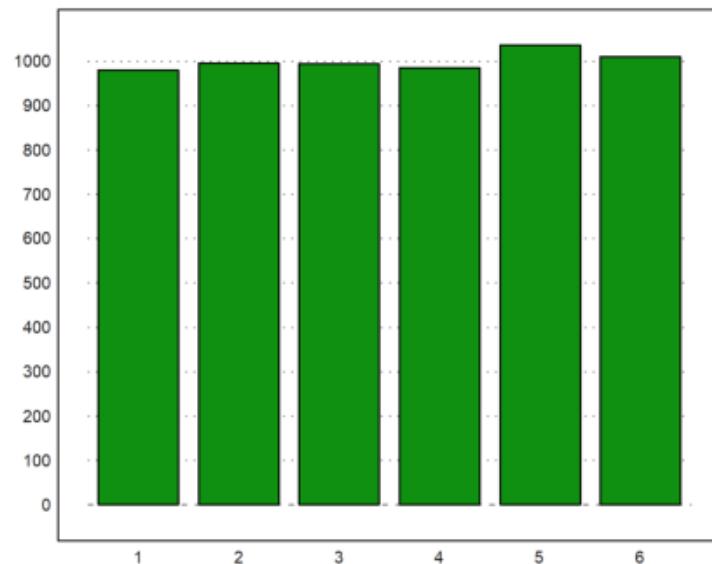
Dengan parameter `even=true`, ini akan menggunakan interval bilangan bulat.

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

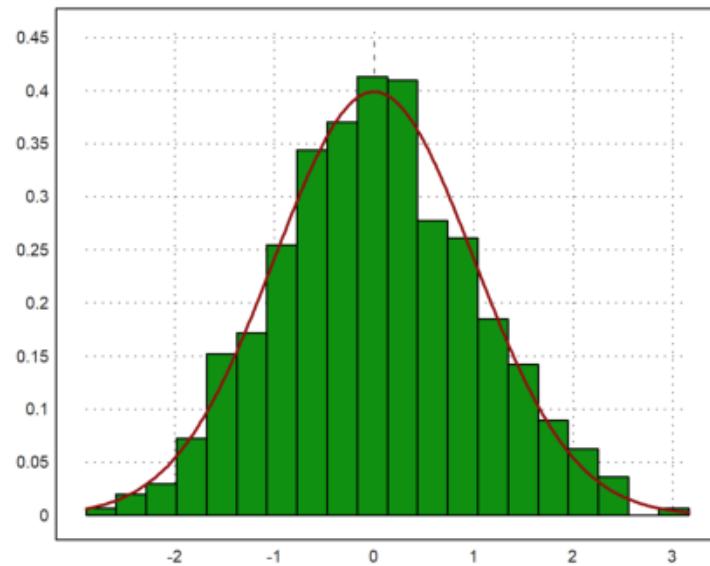


Perhatikan bahwa ada banyak plot statistik yang mungkin berguna. Lihatlah tutorial tentang statistik.

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

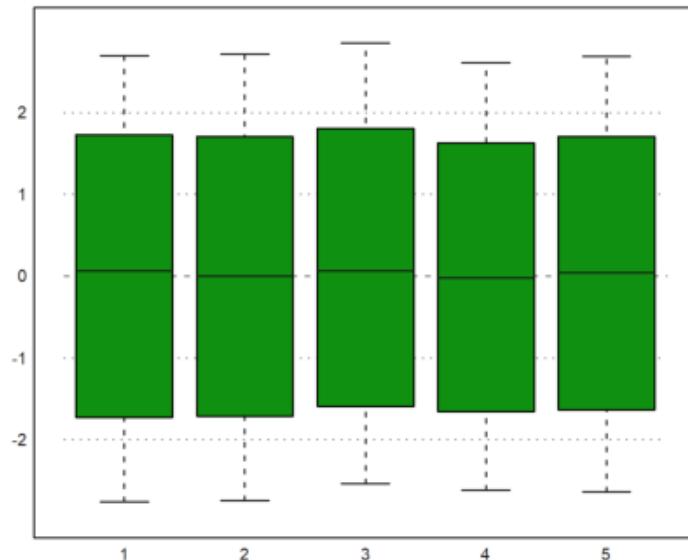


```
>plot2d(normal(1,1000),>distribution); ...
> plot2d("qnormal(x)",color=red,thickness=2,>add):
```



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

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



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

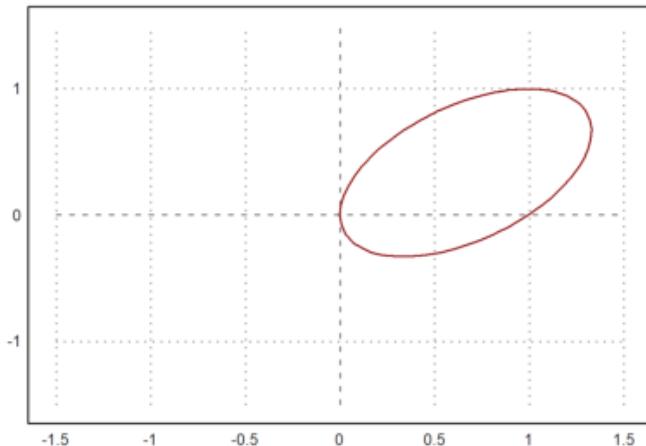
Euler dapat menandai garis level

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

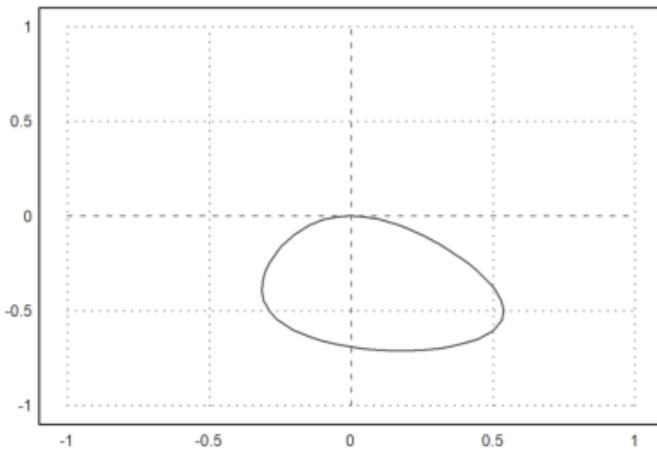
dari fungsi apa pun.

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

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

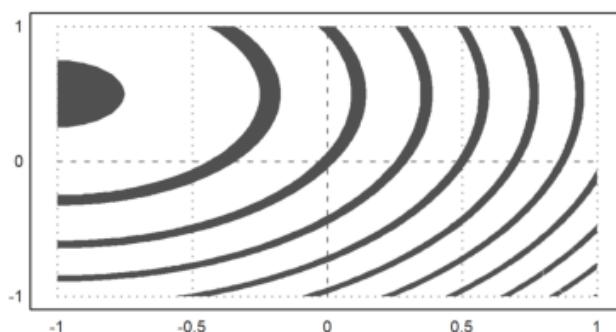


```
>expr := "2*x^2+x*y+3*y^4+y"; // mendefinisikan ekspresi f(x,y)
>plot2d(expr, level=0); // solusi dari f(x,y)=0
```

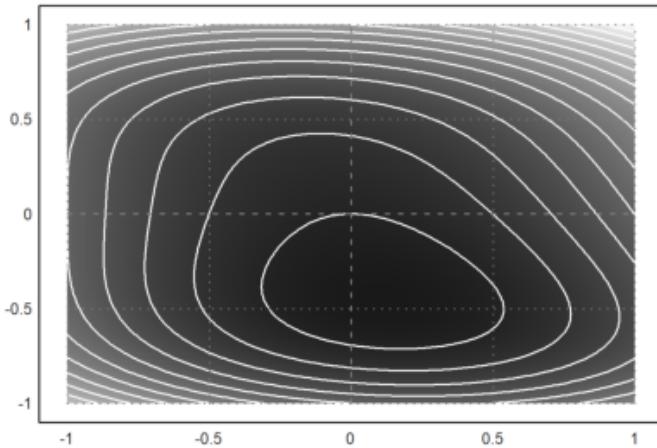


contoh lain

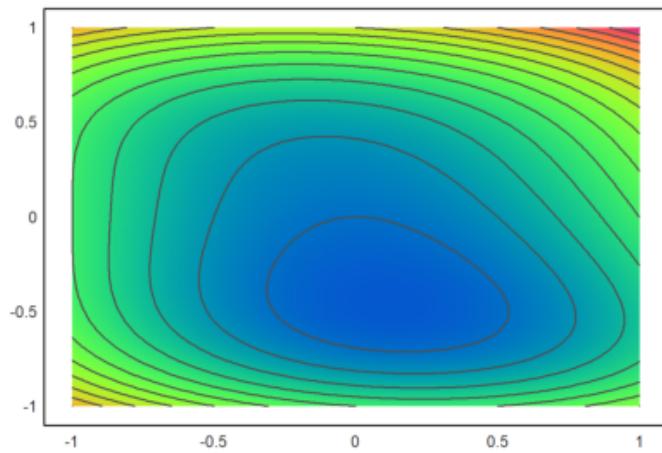
```
>plot2d("x^2+x*y+y^2-y",>contour,>spectral):
```



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

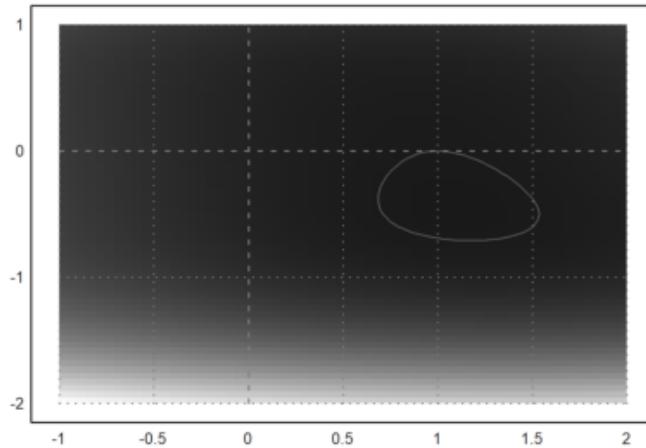


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // lebih baik
```

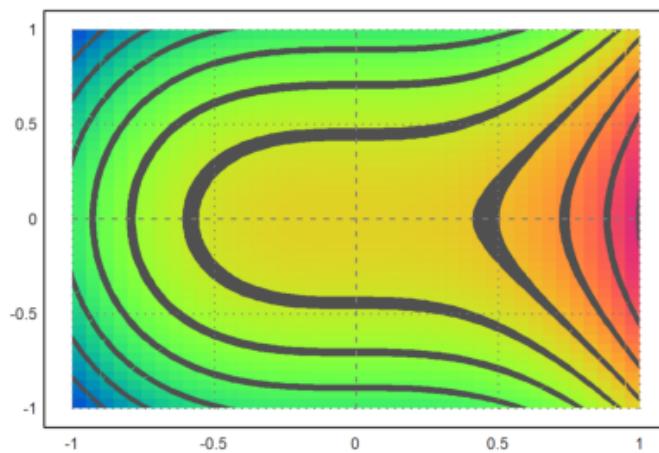


Hal ini juga berlaku untuk plot data. Tetapi Anda harus menentukan rentang untuk label sumbu.

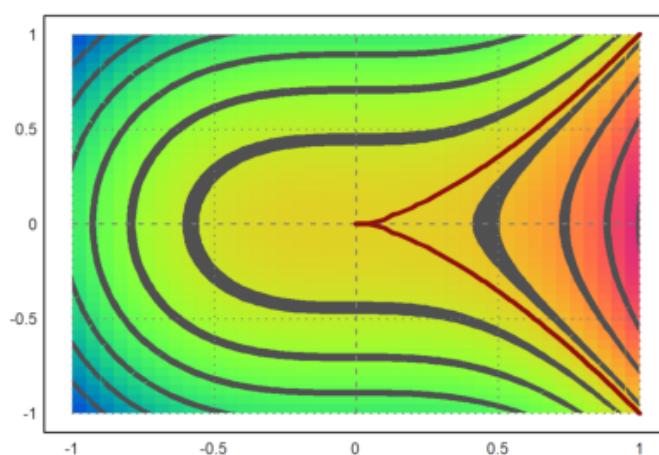
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



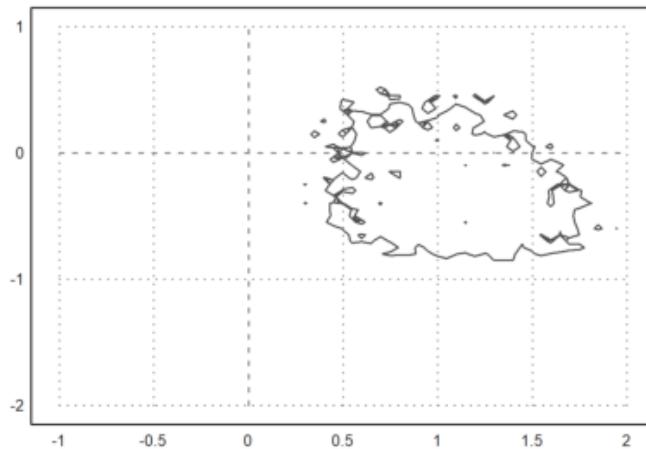
```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



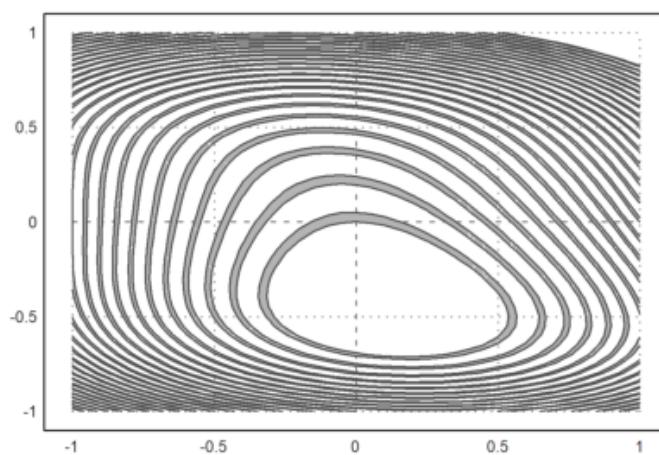
```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



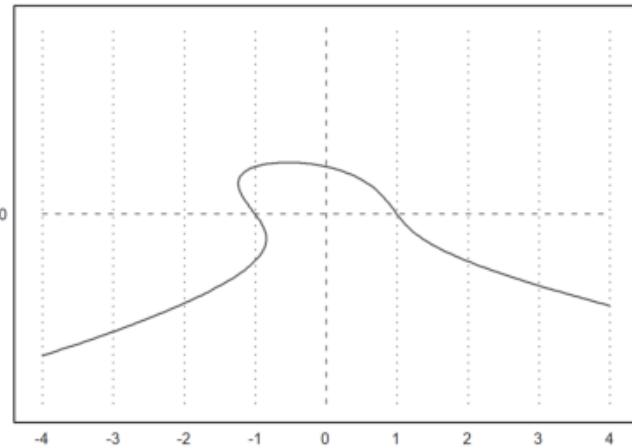
```
>z=z+normal(size(z))*0.2;  
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



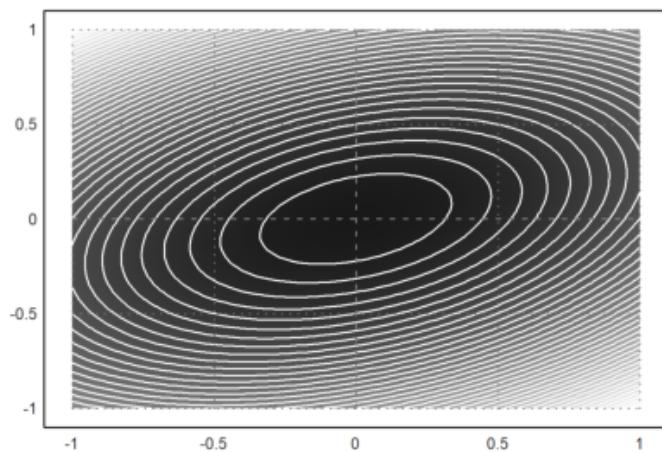
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



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



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



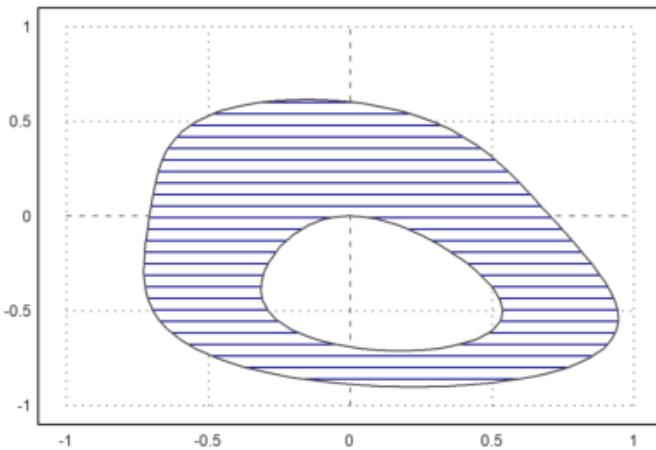
Dimungkinkan juga untuk mengisi set

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

dengan rentang level.

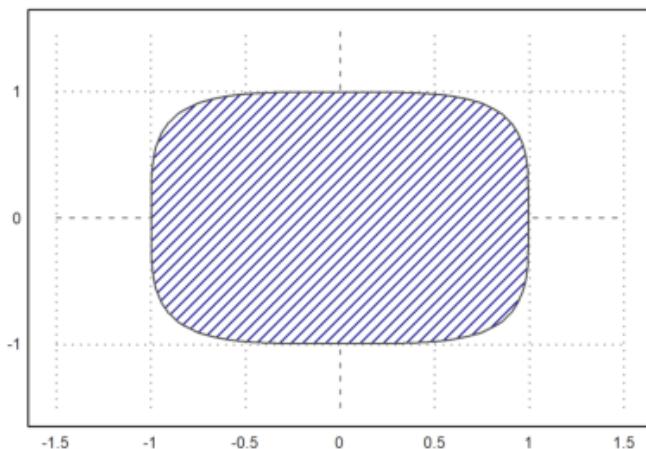
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas

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

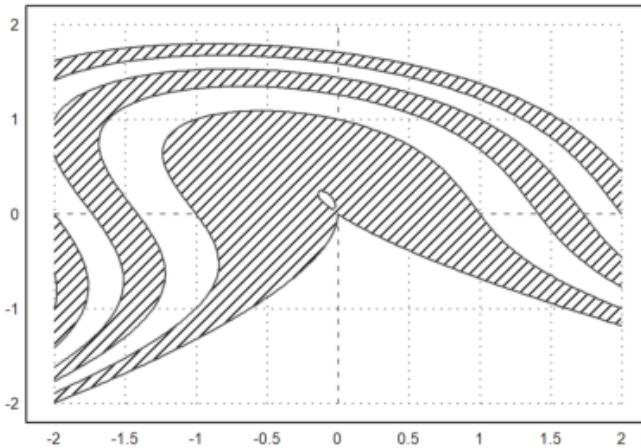


Plot implisit juga dapat menunjukkan rentang level. Maka level harus berupa matriks 2xn interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Sebagai alternatif, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level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

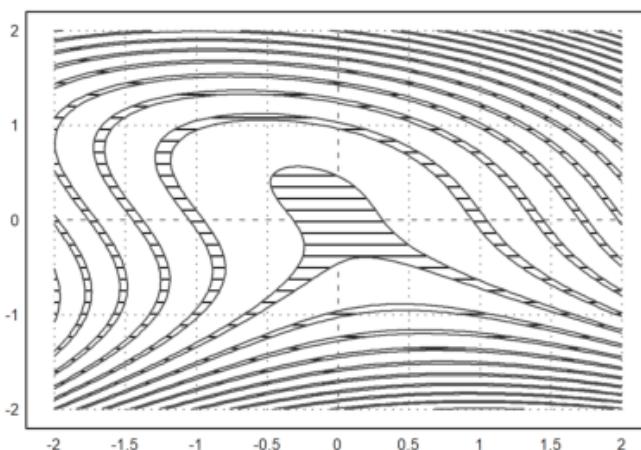
```
>plot2d("x^4+y^4", r=1.5, level=[0;1], color=blue, style="/"):
```



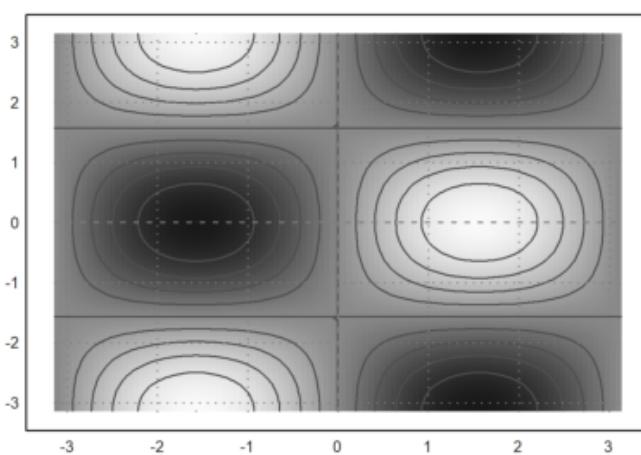
```
>plot2d("x^2+y^3+x*y", level=[0,2,4;1,3,5], style="/", r=2, n=100):
```



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



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

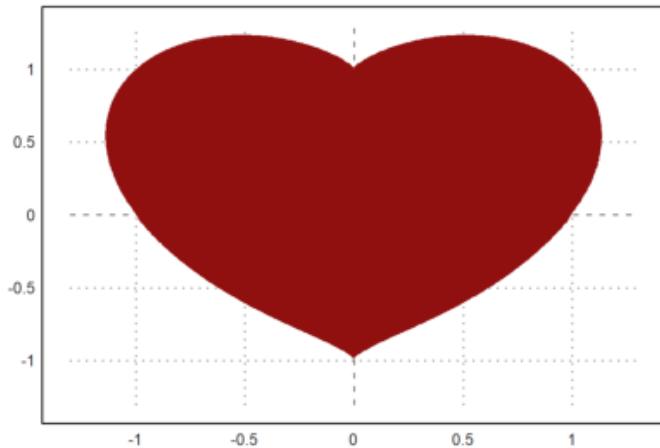


Anda juga dapat menandai suatu wilayah

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

Hal ini dilakukan dengan menambahkan level dengan dua baris.

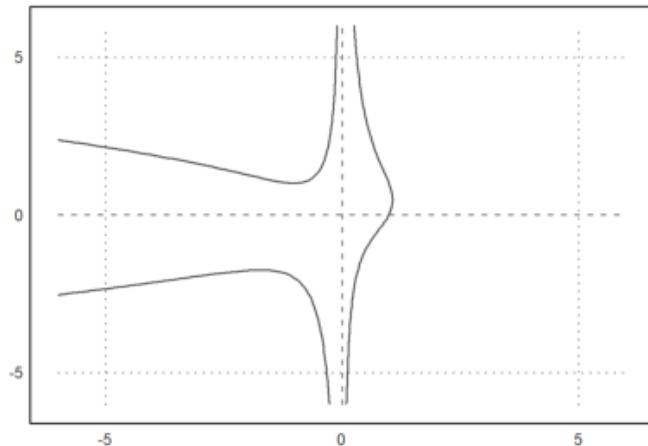
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3", r=1..3, ...
>  style="#", color=red, <outline, ...
>  level=[-2;0], n=100) :
```



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

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

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

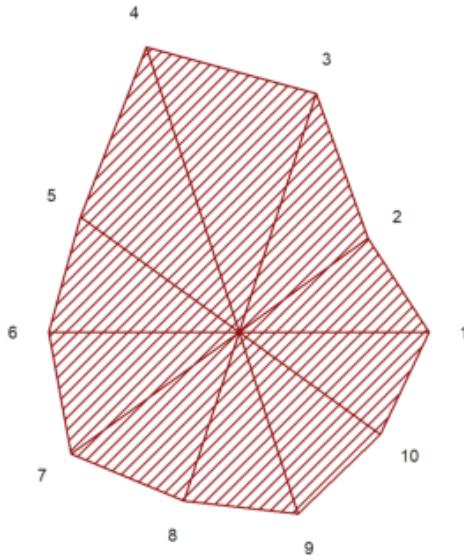


```
>function starplot1 (v, style="/", color=green, lab=none) ...
if !holding() then clg; endif;
w=window(); window(0,0,1024,1024);
h=holding(1);
r=max(abs(v))*1.2;
setplot (-r,r,-r,r);
n=cols(v); t=linspace(0,2pi,n);
v=v|v[1]; c=v*cos(t); s=v*sin(t);
cl=barcolor(color); st=barstyle(style);
loop 1 to n
  polygon([0,c[#],c[#+1]], [0,s[#],s[#+1]],1);
  if lab!=none then
    rlab=v[#]+r*0.1;
    {col,row}=toscreen(cos(t[#])*rlab,sin(t[#])*rlab);
    ctext(""+lab#[#],col,row-textheight()/2);
  endif;
end;
barcolor(cl); barstyle(st);
holding(h);
window(w);
endfunction
```

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

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

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

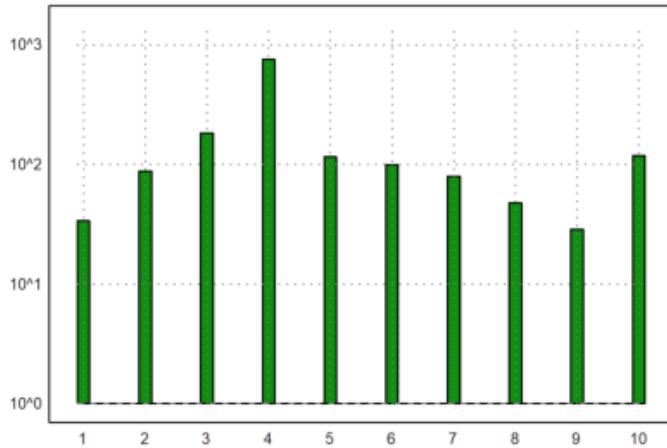


Terkadang, Anda mungkin ingin merencanakan sesuatu yang tidak dapat dilakukan oleh plot2d, tetapi hampir. Pada fungsi berikut, kita melakukan plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

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

Mari kita uji dengan nilai yang terdistribusi secara eksponensial

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



Mari kita menghidupkan kurva 2D dengan menggunakan plot langsung. Perintah plot(x,y) hanya memplot kurva ke dalam jendela plot. setplot(a,b,c,d) mengatur jendela ini. Fungsi wait(0) memaksa plot untuk muncul pada jendela grafis. Kalau tidak, penggambaran ulang dilakukan dalam interval waktu yang jarang.

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

```
t=linspace(0,2pi,500);
f=0;
c=framecolor(0);
l=linewidth(2);
setplot(-1,1,-1,1);
repeat
  clg;
  plot(sin(n*t),cos(m*t+f));
  wait(0);
  if testkey() then break; endif;
  f=f+0.02;
end;
framecolor(c);
linewidth(l);
endfunction
```

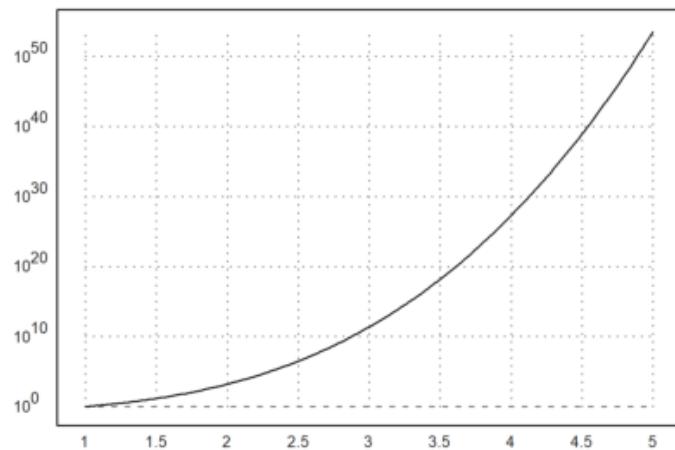
Tekan sembarang tombol untuk menghentikan animasi ini.

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

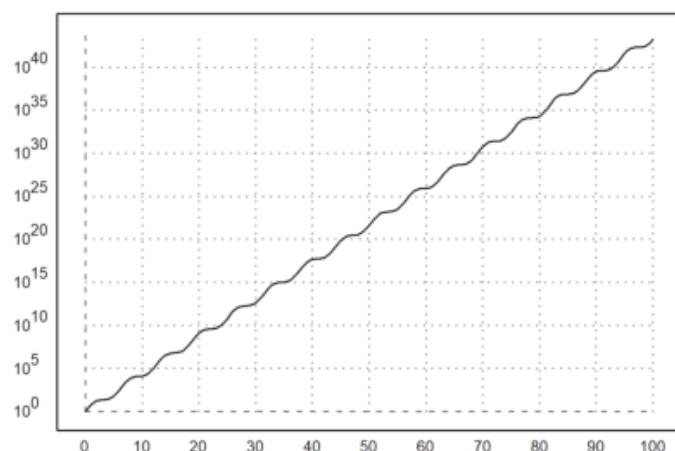
EMT menggunakan parameter "logplot" untuk skala logaritmik. Plot logaritmik dapat diplot menggunakan skala logaritmik dalam y dengan logplot = 1, atau menggunakan skala logaritmik dalam x dan y dengan logplot = 2, atau dalam x dengan logplot = 3.

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

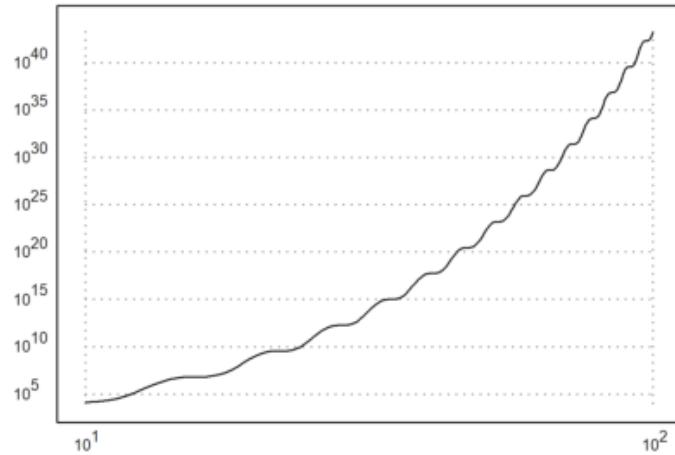
```
>plot2d("exp(x^3-x)*x^2", 1, 5, logplot=1) :
```



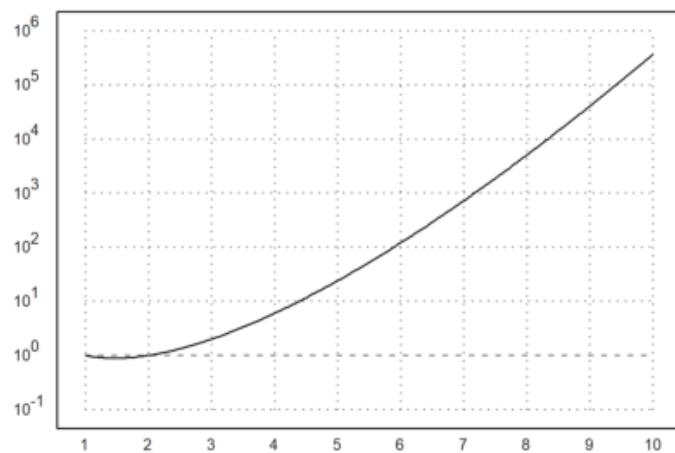
```
>plot2d("exp(x+sin(x))", 0, 100, logplot=1) :
```



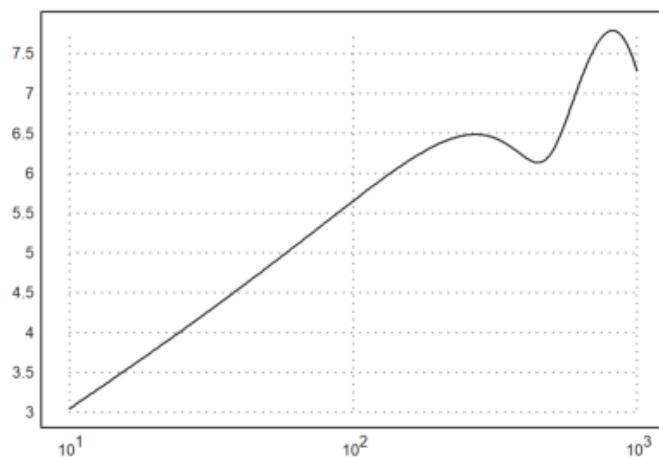
```
>plot2d("exp(x+sin(x))", 10, 100, logplot=2) :
```



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

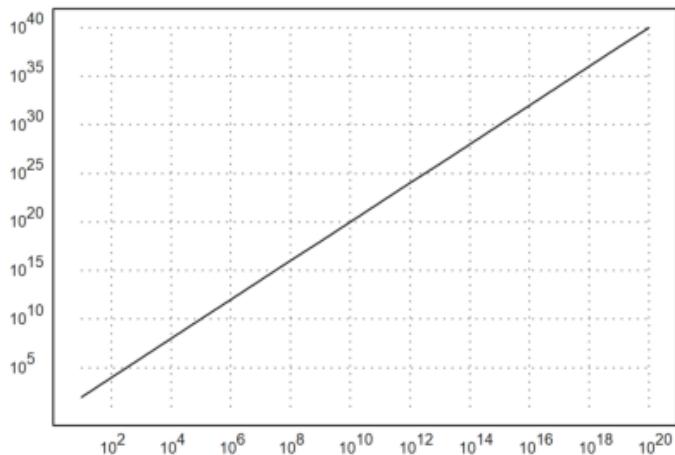


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Hal ini juga berlaku untuk plot data

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



Rujukan Lengkap Fungsi plot2d()

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

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

Parameter

x,y : persamaan, fungsi, atau vektor data

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

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

r dapat berupa vektor [rx, ry] atau vektor [rx1, rx2, ry1, ry2].

xmin, xmax : kisaran parameter untuk kurva

auto: Menentukan rentang y secara otomatis (default)

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

n : jumlah interval (standarnya adalah adaptif)

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

1 = sumbu saja,
 2 = kisi-kisi normal (lihat di bawah untuk jumlah garis kisi
 kisi)

3 = sumbu bagian dalam
 4 = tidak ada kisi-kisi
 5 = grid penuh termasuk margin
 6 = tanda centang pada bingkai
 7 = hanya sumbu
 8 = hanya sumbu, sub-centang

frame : 0 = tidak ada frame

framecolor: warna frame dan grid

margin : angka antara 0 dan 0.4 untuk margin di sekitar plot

color : warna kurva. Jika ini adalah vektor warna,

ini akan digunakan untuk setiap baris dari matriks plot. Dalam

kasus plot titik, ini harus berupa vektor kolom. Jika vektor baris atau matriks penuh warna digunakan untuk plot titik, vektor tersebut akan digunakan untuk setiap titik data.

thickness: ketebalan untuk curva

nilai ini bisa lebih kecil dari 1 untuk garis yang sangat tipis.

style: style plot untuk garis, markers, dan fills.

Untuk points gunakan "[]", "<>", ".", "..", "...", "*", "+", "|", "-", "o" "[]", "<>", "o" (filled shapes)
 "[]w", "<>w", "ow" (non-transparent)

Untuk garis gunakan "-", "--", "-.", ".-", "-.-", "->"

Untuk poligon terisi atau plot bar gunakan "", "O", "O", "/", "\", "\/", "+", "|", "-", "t"

points : memplot titik tunggal bukan segmen

addpoints : jika benar, memplot segmen dan titik

add : menambahkan plot ke plot yang ada

user : mengaktifkan interaksi pengguna untuk fungsi

delta : ukuran langkah untuk interaksi pengguna

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

histogram : memplot frekuensi x dalam n subinterval

distribution=n : memplot distribusi x dengan n subinterval

even : menggunakan nilai untuk histogram otomatis

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

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

level : plot garis level dari fungsi implisit dari dua variable

outline : menggambar batas rentang level.

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

hue : menambahkan warna rona ke plot level untuk menunjukkan nilai fungsi
 contour: gunakan plot level untuk level otomatis

nc: angka untuk garis level otomatis

title: judul plot (default "")

x1, y1: label untuk sumbu x dan y

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

vertical:

mengaktifkan atau menonaktifkan label vertikal. Ini mengubah

variabel

global label vertikal secara lokal untuk satu plot. Nilai 1 hanya menetapkan teks vertikal, nilai 2 menggunakan label numerik vertikal pada sumbu y.

filled: mengisi plot kurva

fillcolor: mengisi warna untuk bar dan kurva yang terisi

outline: batas untuk poligon yang terisi

logplot: set untuk plot logaritmik

```
1 = logplot di y
2 = logplot di xy
3 = logplot di x
```

own:

Sebuah string yang menunjuk ke sebuah rutinitas plot. Gunakan >user, anda akan mendapatkan unsur interaction yang sama seperti di plot2d. Jangkauan akan diatur sebelum setiap memanggil fungsi.

maps: ekspresi map(0 lebih cepat), fungsi selalu dipetakan

contourcolor: warna garis kontur

contourwidth: lebar garis kontur

clipping: mengalihkan clipping (defaultnya true)

title:

Ini dapat digunakan untuk menggambarkan plot. Judul akan muncul di atas plot. Selain itu, label untuk sumbu x dan y dapat ditambahkan dengan `x1= "string"` atau `y1 = "string"`. Label lain dapat ditambahkan dengan fungsi `label()` atau `labelbox()`. Judul dapat berupa sebuah unicode string atau gambar dengan formula LaTex.

`cgrid:`

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

Overview

Fungsi ini dapat merencanakan

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

Jika sebuah fungsi atau ekspresi untuk xv diberikan, `plot2d()` akan menghitung nilai dalam rentang yang diberikan menggunakan fungsi atau ekspresi tersebut. Ekspresi harus berupa ekspresi dalam variabel x. Kisarannya harus ditentukan dalam parameter a dan b kecuali jika rentang default harus digunakan. Rentang y akan dihitung secara otomatis, kecuali c dan d ditentukan, atau radius r, yang menghasilkan rentang r, r untuk x dan y. Untuk plot fungsi, `plot2d` akan menggunakan evaluasi adaptif dari fungsi secara default. Untuk mempercepat plot untuk fungsi-fungsi yang rumit, matikan ini dengan `<adaptive`, dan secara opsional kurangi jumlah interval n. Selain itu, `plot2d()` secara default akan menggunakan pemetaan. Yaitu, ia akan menghitung elemen plot untuk elemen. Jika ekspresi Anda atau fungsi Anda dapat menangani sebuah vektor x, Anda dapat menonaktifkannya dengan `<maps` untuk evaluasi yang lebih cepat.

Perhatikan bahwa plot adaptif selalu dihitung elemen per elemen. Jika fungsi atau ekspresi untuk xv dan yv ditentukan, `plot2d()` akan menghitung kurva dengan nilai xv sebagai koordinat x dan nilai yv sebagai koordinat y. Pada kasus ini, sebuah rentang haruslah didefinisikan untuk parameter menggunakan `xmin`, `xmax`. Ekspresi yang terdapat dalam string harus selalu berupa ekspresi dalam variabel parameter x.

BAB 4

Menggambar Plot 3D dengan EMT

Ini adalah pengenalan plot 3D di Euler. Kita memerlukan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

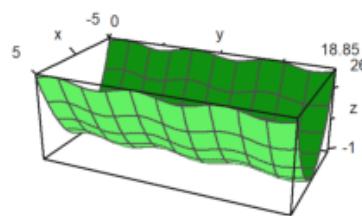
Euler menggambar fungsi-fungsi tersebut dengan menggunakan algoritme pengurutan untuk menyembunyikan bagian-bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Standarnya adalah dari kuadran x-y positif ke arah asal $x=y=z=0$, tetapi sudut= 0° terlihat dari arah sumbu-y. Sudut pandang dan ketinggian dapat diubah.

Euler dapat merencanakan

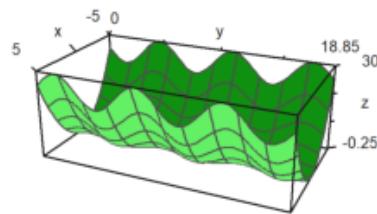
- permukaan dengan bayangan dan garis level atau rentang level,
- awan titik-titik,
- kurva parametrik,
- permukaan implisit.

Plot 3D dari sebuah fungsi menggunakan plot3d. Cara termudah adalah dengan memplot ekspresi dalam x dan y. Parameter r mengatur rentang plot di sekitar (0,0).

```
>aspect(1.5); plot3d("x^2+sin(y)", -5, 5, 0, 6*pi):
```



```
>plot3d("x^2+x*sin(y)", -5, 5, 0, 6*pi) :
```



Fungsi dari dua Variabel

Untuk grafik fungsi, gunakan

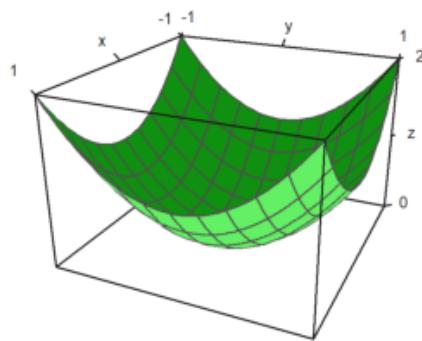
- ekspresi sederhana dalam x dan y,
- nama fungsi dari dua variabel
- atau matriks data.

Standarnya adalah kisi-kisi kawat yang terisi dengan warna yang berbeda pada kedua sisinya. Perhatikan, bahwa jumlah interval kisi-kisi default adalah 10, tetapi plot menggunakan jumlah default 40x40 persegi panjang untuk membangun permukaan. Hal ini dapat diubah.

- n = 40, n = [40,40]: jumlah garis kisi di setiap arah
- grid=10, grid=[10,10]: jumlah garis kisi di setiap arah.

Kami menggunakan default n=40 dan grid=10.

```
>plot3d("x^2+y^2") :
```



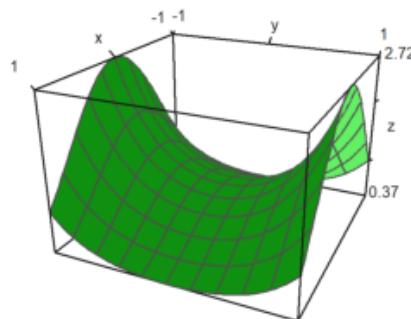
Interaksi pengguna dapat dilakukan dengan parameter `>user`.

Pengguna dapat menekan tombol berikut ini.

- kiri, kanan, atas, bawah: putar sudut pandang
- +,-: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l: sakelar untuk memutar sumber cahaya (lihat di bawah)
- spasi: setel ulang ke default
- return: 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: bujur sangkar simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

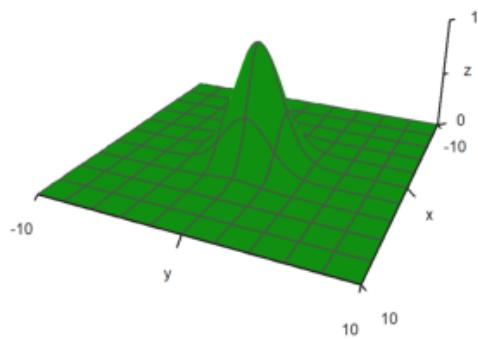
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: skala ke nilai fungsi (defaultnya adalah <fscale>).

scale: angka atau vektor 1x2 untuk menskalakan ke arah x dan y.

frame: jenis bingkai (default 1).

```
>plot3d("exp(-(x^2+y^2)/5)", r=10, n=80, fscale=4, scale=1.2, frame=3) :
```



Tampilan dapat diubah dengan berbagai cara.

- **distance:** jarak pandang ke plot.
- **zoom:** nilai zoom.
- **angle:** sudut ke sumbu y negatif dalam radian.
- **height:** ketinggian tampilan dalam radian.

Nilai default dapat diperiksa atau diubah dengan fungsi `view()`. Fungsi ini mengembalikan parameter sesuai urutan di atas.

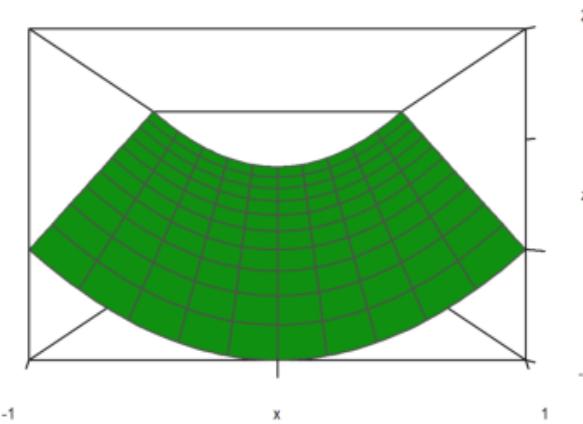
```
>view
```

```
[5, 2.6, 2, 0.4]
```

Jarak yang lebih dekat membutuhkan zoom yang lebih sedikit. Efeknya lebih seperti lensa sudut lebar.

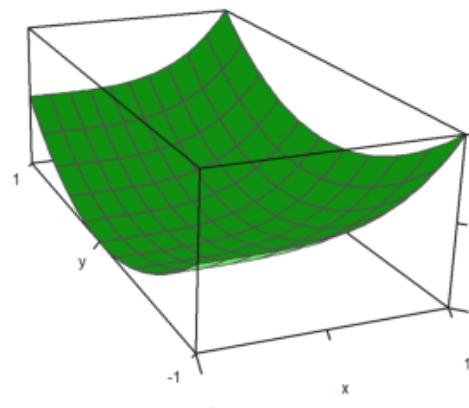
Pada contoh berikut ini, sudut = 0 dan tinggi = 0 terlihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^2+y", distance=3, zoom=2, angle=0, height=0) :
```



Plot terlihat selalu ke bagian tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter center.

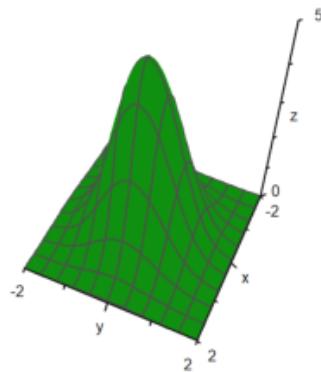
```
>plot3d("x^4+y^2", a=0, b=1, c=-1, d=1, angle=-20°, height=20°, ...
>  center=[0.4, 0, 0], zoom=5):
```



Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi, tidak perlu mengubah jarak atau melakukan zoom, tergantung pada ukuran plot. Namun demikian, label mengacu ke ukuran yang sesungguhnya.

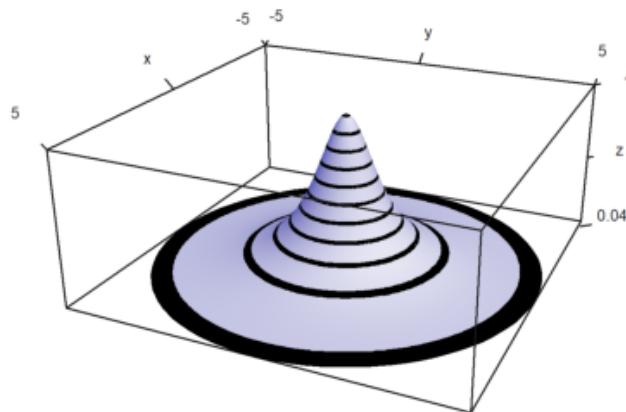
Jika Anda menonaktifkannya dengan scale=false, Anda harus berhati-hati, agar plot tetap muat ke dalam jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan bagian tengahnya.

```
>plot3d("5*exp(-x^2-y^2)", r=2, <fscale, <scale, distance=13, height=50°, ...
>  center=[0, 0, -2], frame=3):
```

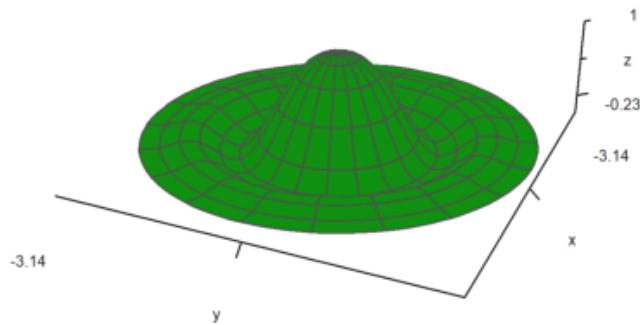


Plot polar juga tersedia. Parameter `polar=true` menggambar plot polar. Fungsi harus tetap merupakan fungsi dari x dan y . Parameter "fscale" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi akan diskalakan agar sesuai dengan kubus

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=blue):
```



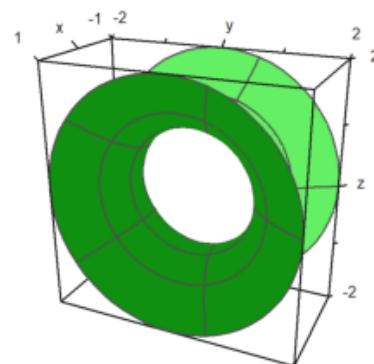
```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```



Parameter `rotate` memutar fungsi dalam x di sekitar sumbu x.

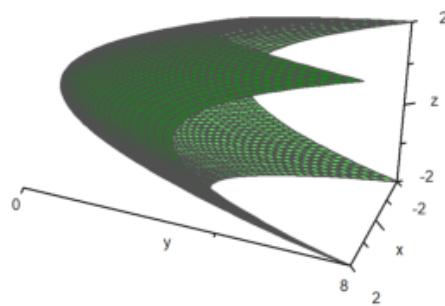
- `rotate = 1`: Menggunakan sumbu x
- `rotate = 2`: Menggunakan sumbu z

```
>plot3d("x^2+1", a=-1, b=1, rotate=true, grid=5):
```



Berikut ini adalah plot dengan tiga fungsi

```
>plot3d("x", "x^2+y^2", "y", r=2, zoom=3.5, frame=3):
```



Plot Kontur

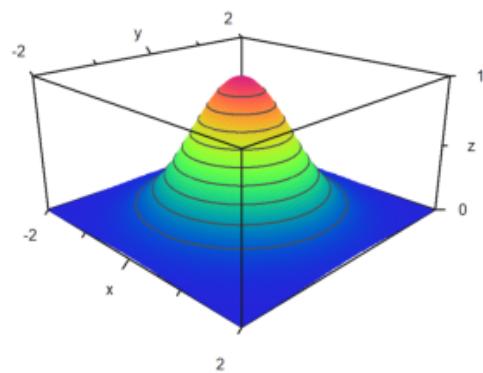
Untuk plot, Euler menambahkan garis kisi-kisi. Sebagai gantinya, dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona berwarna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan bayangan. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah / cyan.

- >hue: Mengaktifkan bayangan cahaya, bukan kabel.
- >contour: Memplot garis kontur otomatis pada plot.
- level=... (atau levels): Vektor nilai untuk garis kontur.

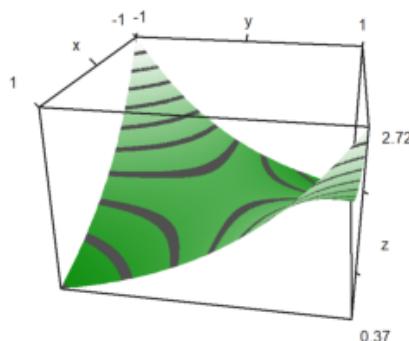
Standarnya adalah level = "auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat dalam plot, level sebenarnya adalah kisaran level.

Gaya default dapat diubah. Untuk plot kontur berikut ini, kami menggunakan grid yang lebih halus untuk titik-titik 100x100, skala fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...
> >contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°) :
```



```
>plot3d("exp(x*y)", angle=100°, >contour, color=green) :
```

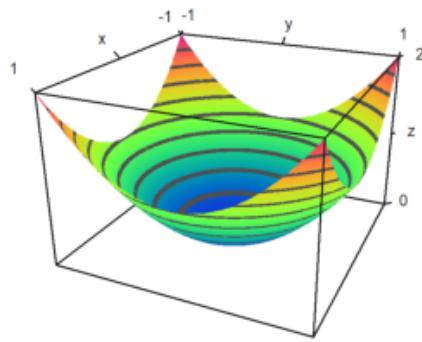


Bayangan default menggunakan warna abu-abu. Tetapi, kisaran warna spektral juga tersedia.

- >spectral: Menggunakan skema spektral default
- color =...: Menggunakan warna khusus atau skema spektral

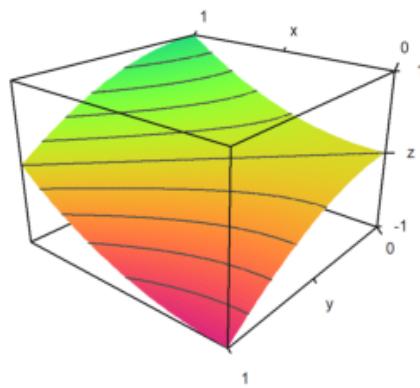
Untuk plot berikut ini, kami menggunakan skema spektral default dan menambah jumlah titik untuk mendapatkan tampilan yang sangat mulus

```
>plot3d("x^2+y^2", >spectral, >contour, n=100) :
```



Alih-alih garis level otomatis, kita juga dapat menetapkan nilai garis level. Hal ini akan menghasilkan garis level yang tipis, alih-alih rentang level.

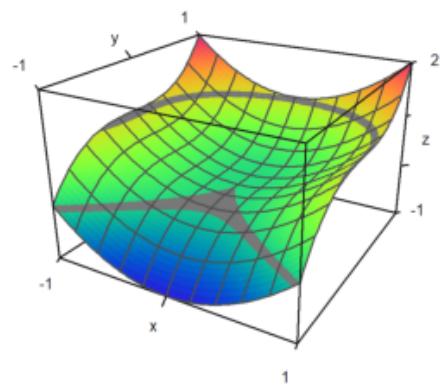
```
>plot3d("x^2-y^2",0,1,0,1,angle=220°,level=-1:0.2:1,color=redgreen):
```



Pada plot berikut, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas-batas level sebagai kolom.

Selain itu, kami menghamparkan kisi-kisi dengan 10 interval di setiap arah.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...
> >spectral,angle=30°,grid=10,contourcolor=gray):
```

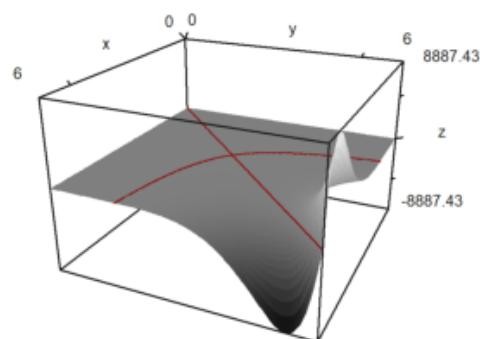


Pada contoh berikut, kami memplot himpunan, di mana

$$f(x, y) = x^y - y^x = 0$$

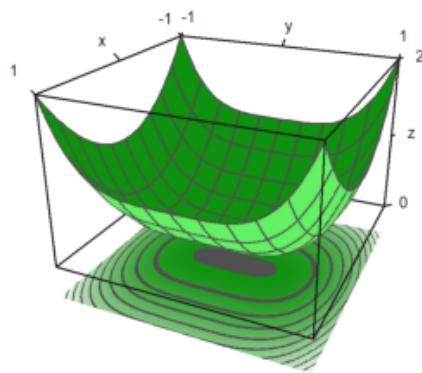
Kami menggunakan satu garis tipis untuk garis level.

```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```



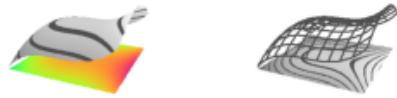
Dimungkinkan untuk menampilkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan

```
>plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2):
```



Berikut ini beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi-kisi.

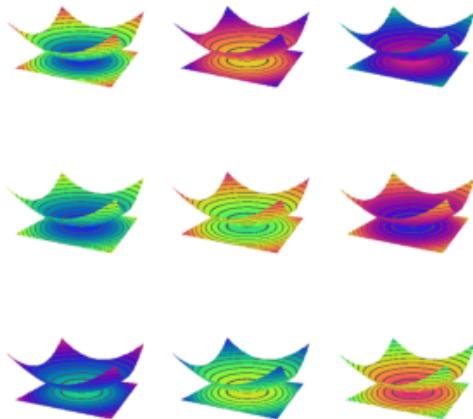
```
>figure(2,2); ...
>expr="y^3-x^2"; ...
>figure(1); ...
> plot3d(expr,<frame,>cp,cpcolor=spectral); ...
>figure(2); ...
> plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
>figure(3); ...
> plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
>figure(4); ...
> plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...
>figure(0):
```



Ada beberapa skema spektral lainnya, yang diberi nomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan color=value, di mana value

- spectral: untuk rentang dari biru ke merah
- white: untuk rentang yang lebih redup
- yellowblue,purplegreen,blueyellow,greenred
- blueyellow,greenpurple,yellowblue,redgreen

```
>figure(3,3); ...
>for i=1:9; ...
>  figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4);
>end; ...
>figure(0):
```

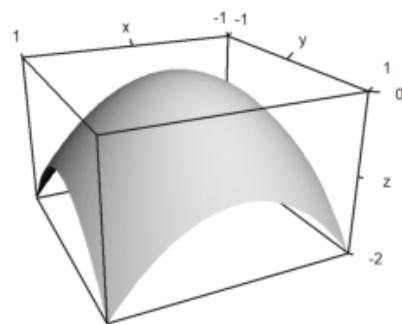


Sumber cahaya dapat diubah dengan 1 dan tombol cursor selama interaksi pengguna. Ini juga dapat ditetapkan dengan parameter.

- light: arah untuk cahaya
- amb: cahaya sekitar antara 0 dan 1

Perhatikan, bahwa program ini tidak membuat perbedaan di antara sisi-sisi plot. Tidak ada bayangan. Untuk ini, Anda memerlukan Povray

```
>plot3d("-x^2-y^2", ...
>  hue=true,light=[0,1,1],amb=0,user=true, ...
>  title="Press 1 and cursor keys (return to exit)":
```



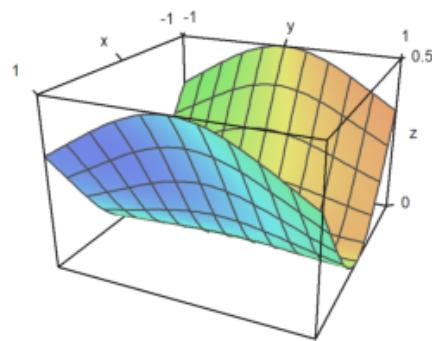
Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah

```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
> zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01):
```



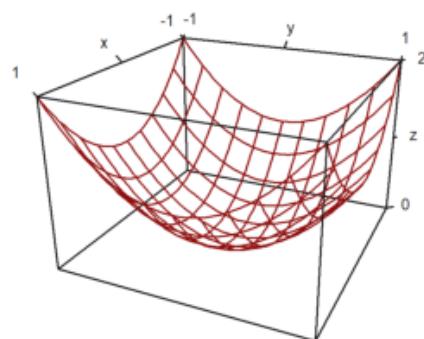
Warna 0 memberikan efek pelangi yang istimewa

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



Permukaannya juga bisa transparan.

```
>plot3d("x^2+y^2", >transparent, grid=10, wirecolor=red) :
```



Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari sebuah fungsi dalam tiga variabel. Solusi dari

$$f(x, y, z) = 0$$

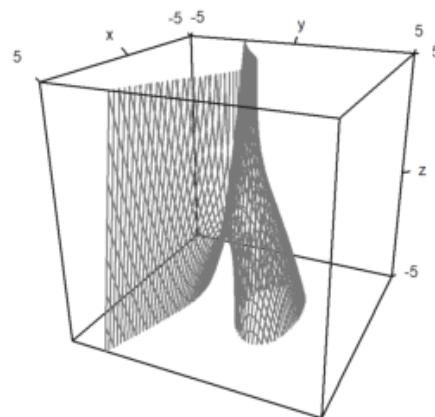
dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, bidang x-z dan bidang y-z.

- implisit = 1: potong sejajar dengan bidang y-z
- implicit=2: potong sejajar dengan bidang x-z
- implicit = 4: potong sejajar dengan bidang x-y

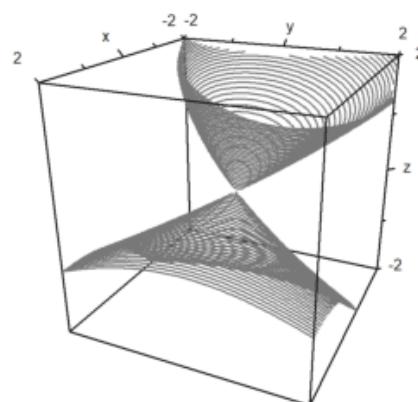
Tambahkan nilai-nilai ini, jika Anda mau. Dalam contoh, kami memplot

$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

```
>plot3d("x^2+y^3+z*y-1", r=5, implicit=3):
```



```
>plot3d("x^2+y^2+4*x*z+z^3", >implicit, r=2, zoom=2.5):
```



Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu menyediakan matriks nilai x, y, dan z, atau tiga fungsi atau ekspresi $fx(x,y)$, $fy(x,y)$, $fz(x,y)$.

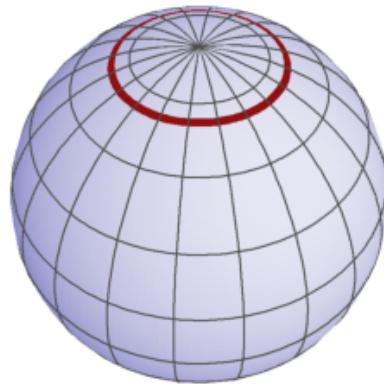
$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

Karena x, y, z adalah matriks, kita asumsikan bahwa (t, s) berjalan melalui kisi-kisi persegi. Hasilnya, Anda dapat memplot gambar persegi panjang dalam ruang.

Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

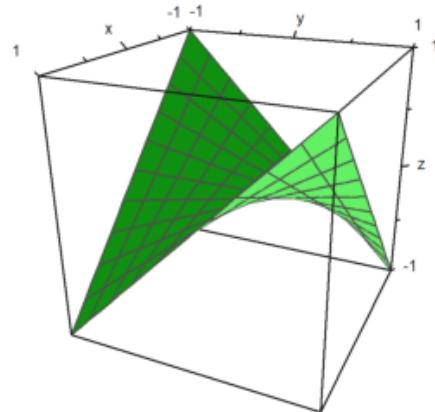
Pada contoh berikut, kita menggunakan vektor nilai t dan vektor kolom nilai s untuk memparameterkan permukaan bola. Pada gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...
>plot3d(x,y,z,>hue, ...
>color=blue,<frame,grid=[10,20], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.4,height=50°):
```



Berikut ini adalah contoh, yang merupakan grafik suatu fungsi.

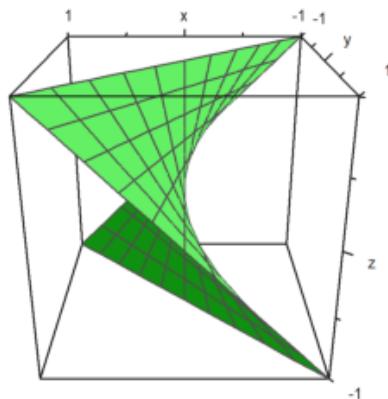
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



Namun demikian, kita bisa membuat segala macam permukaan. Berikut ini adalah permukaan yang sama dengan suatu fungsi

$$x = yz$$

```
>plot3d(t*s, t, s, angle=180°, grid=10) :
```



Dengan lebih banyak upaya, kita bisa menghasilkan banyak permukaan.

Dalam contoh berikut ini, kami membuat tampilan berbayang dari bola yang terdistorsi. Koordinat biasa untuk bola adalah

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

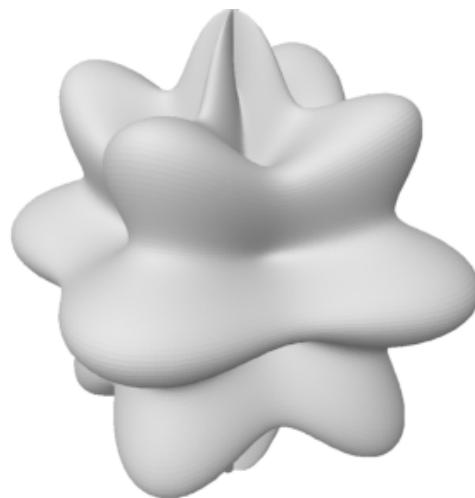
dengan

$$0 \leq t \leq 2\pi, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2}.$$

akan dikurangi dengan faktor

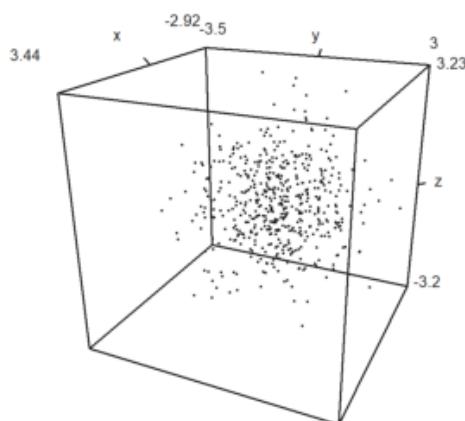
$$d(t, s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

```
>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):
```



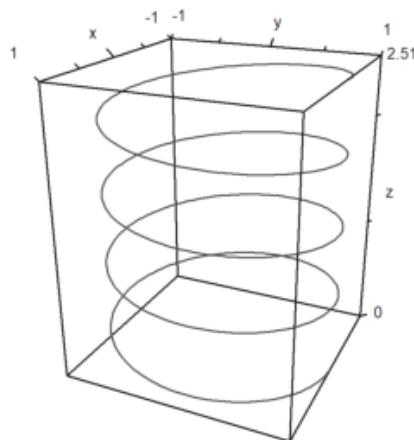
Tentu saja, sebuah point cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita memerlukan tiga vektor untuk koordinat titik. Gaya sama seperti di plot2d dengan poin=true;

```
>n=500; ...
> plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

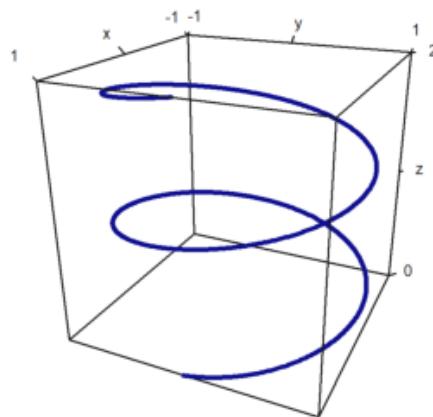


Anda juga dapat memplot kurva dalam bentuk 3D. Dalam hal ini, akan lebih mudah untuk menghitung titiktitik kurva. Untuk kurva pada bidang, kami menggunakan urutan koordinat dan parameter wire = true

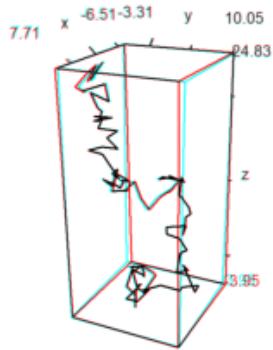
```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
>linewidth=3,wirecolor=blue):
```

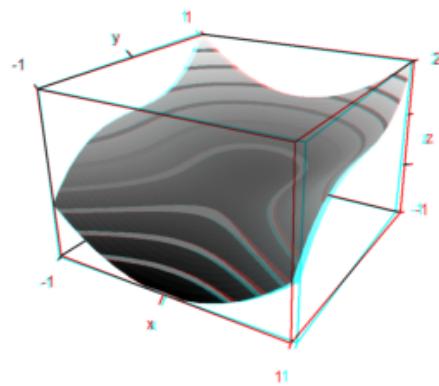


```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



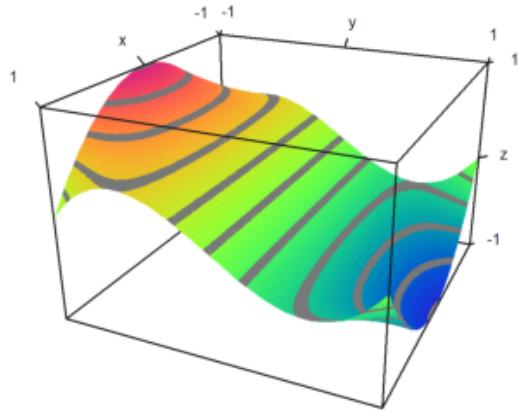
EMT juga dapat membuat plot dalam mode anaglyph. Untuk melihat plot semacam itu, Anda memerlukan kacamata merah/cyan.

```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°) :
```



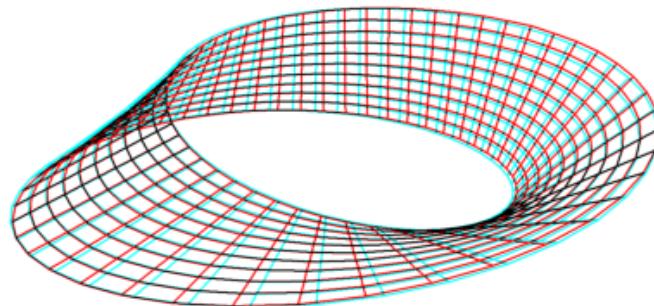
Sering kali, skema warna spektral digunakan untuk plot. Hal ini menekankan ketinggian fungsi

```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2) :
```



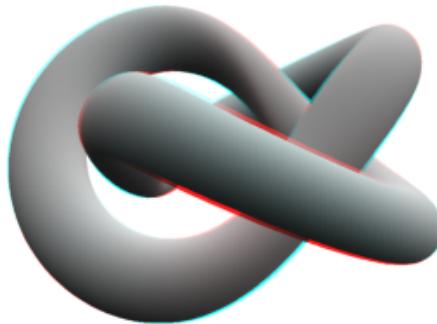
Euler juga dapat memplot permukaan yang diparameterkan, apabila parameternya adalah nilai x, y, dan z dari gambar kisi-kisi persegi panjang di dalam ruang. Untuk demo berikut ini, kita akan menyiapkan parameter u dan v, dan menghasilkan koordinat ruang dari parameter ini

```
>u=linspace (-1,1,10); v=linspace (0,2*pi,50)'; ...
>X=(3+u*cos (v/2 ))*cos (v); Y=(3+u*cos (v/2 ))*sin (v); Z=u*sin (v/2); ...
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```



Berikut ini contoh yang lebih rumit, yang tampak megah dengan kacamata merah/cyan.

```
>u:=linspace (-pi,pi,160); v:=linspace (-pi,pi,400)'; ...
>x:=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...
>y:=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...
>z=sin(u)+2*cos(3*v); ...
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```



Plot Statistik

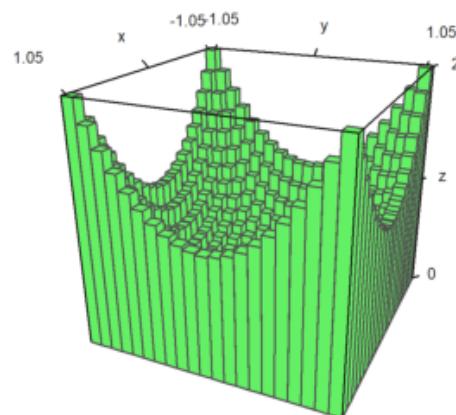
Petak batang juga dimungkinkan. Untuk ini, kita harus menyediakan

- x: vektor baris dengan n+1 elemen
- y: vektor kolom dengan n+1 elemen
- z: matriks nilai nxn.

z dapat lebih besar, tetapi hanya nilai nxn yang akan digunakan.

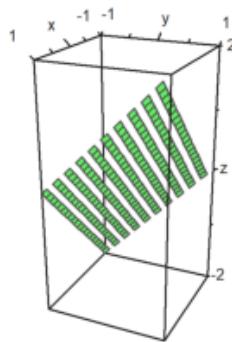
Dalam contoh, pertama-tama kita menghitung nilainya. Kemudian kita menyesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...
>plot3d(xa,ya,z,bar=true):
```



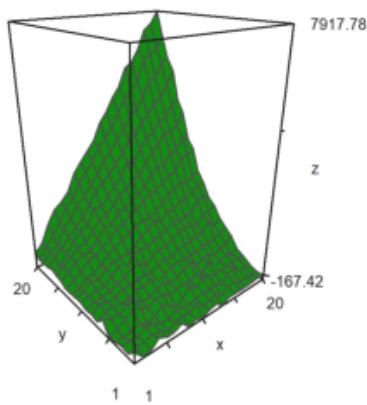
Hal ini memungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20);
```

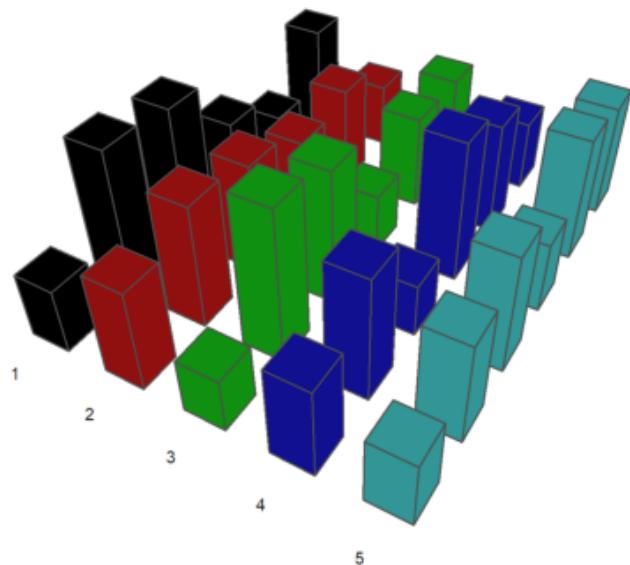


Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan scale(M), atau menskalakan matriks dengan >zscale. Hal ini dapat dikombinasikan dengan faktor penskalaan individual yang ditetapkan sebagai tambahan

```
>i=1:20; j=i'; ...
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8)
```

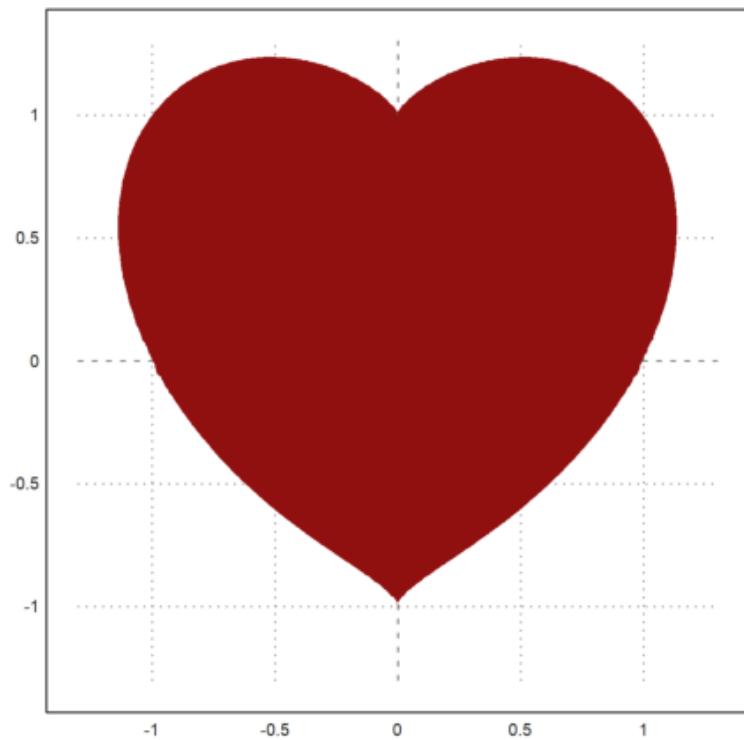


```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v', scols=1:5,ccols=[1:5]):
```



Permukaan Benda Putar

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



```
>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 jantung di sekitar sumbu y. Inilah ekspresi yang medefinisikan jantung:

$$f(x, y) = (x^2 + y^2 - 1)^3 - x^2 \cdot y^3.$$

Selanjutnya kita tetapkan

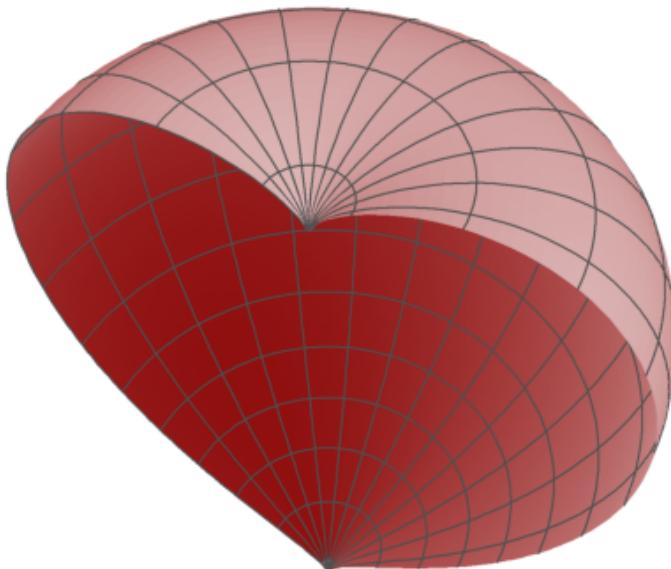
$$x = r \cdot \cos(a), \quad y = r \cdot \sin(a).$$

```
>function fr(r,a) &= ekspressi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $f
```

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2 \sin a) r^5}{16}$$

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang menyelesaikan untuk r , jika a diberikan. Dengan fungsi tersebut kita dapat memplotkan jantung yang diputar sebagai permukaan parametrik.

```
>function map f(a) := bisect("fr",0,2;a); ...
>t=linspace(-pi/2,pi/2,100); r=f(t); ...
>s=linspace(pi,2pi,100)'; ...
>plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```

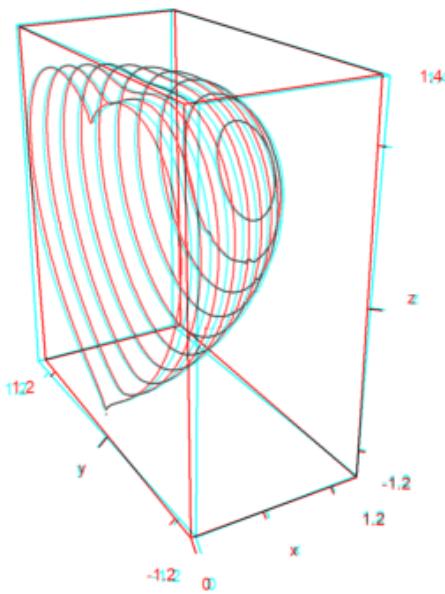


Berikut ini adalah plot 3D dari gambar di atas yang diputar mengelilingi sumbu-z. Kami mendefinisikan fungsi, yang menggambarkan objek.

```
>function f(x,y,z) ...
```

```
r=x^2+y^2;
return (r+z^2-1)^3-r*z^3;
endfunction
```

```
>plot3d("f(x,y,z)", ...
>xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...
>implicit=1,angle=-30°,zoom=2.5,n=[10,100,60],>anaglyph):
```



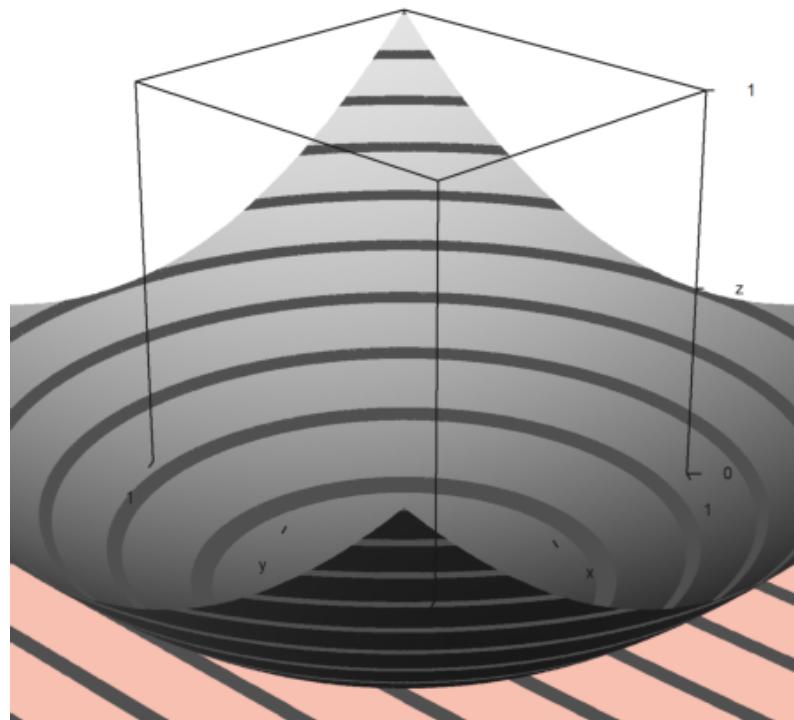
Plot 3D Khusus

Fungsi plot3d memang bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Di samping rutinitas yang lebih mendasar, Anda bisa mendapatkan plot berbingkai dari objek apa pun yang Anda sukai. Meskipun Euler bukan program 3D, namun dapat menggabungkan beberapa objek dasar. Kami mencoba memvisualisasikan parabola dan garis singgungnya.

```
>function myplot ...
y=-1:0.01:1; x=(-1:0.01:1)';
plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ...
    hues=0.5,>contour,color=orange);
h=holding(1);
plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
holding(h);
endfunction
```

Sekarang framedplot() menyediakan frame, dan mengatur tampilan.

```
>framedplot("myplot", [0.1,1,0,1,0,1],angle=-45°, ...
> center=[0,0,-0.7],zoom=6):
```

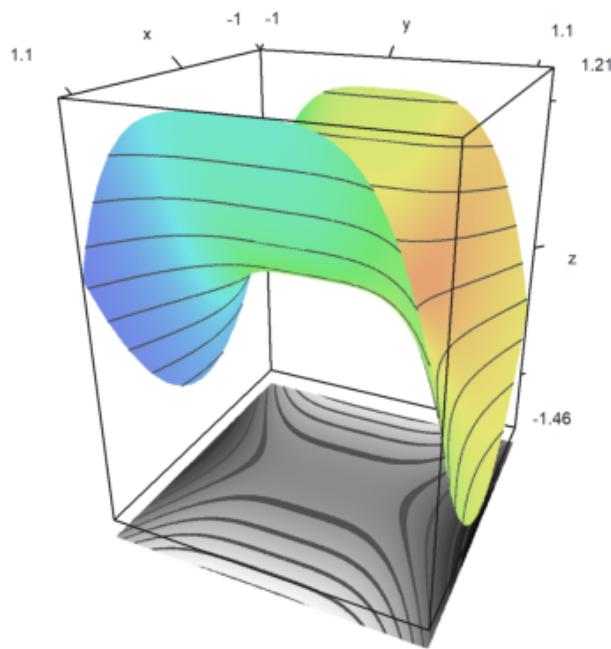


Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa `plot3d()` mengatur jendela ke `fullwindow()` secara default, namun `plotcontourplane()` mengasumsikannya.

```
>x=-1:0.02:1.1; y=x'; z=x^2-y^4;
>function myplot (x,y,z) ...
```

```
zoom(2);
wi=fullwindow();
plotcontourplane(x,y,z,level="auto",<scale);
plot3d(x,y,z,>hue,<scale,>add,color=white,level="thin");
window(wi);
reset();
endfunction
```

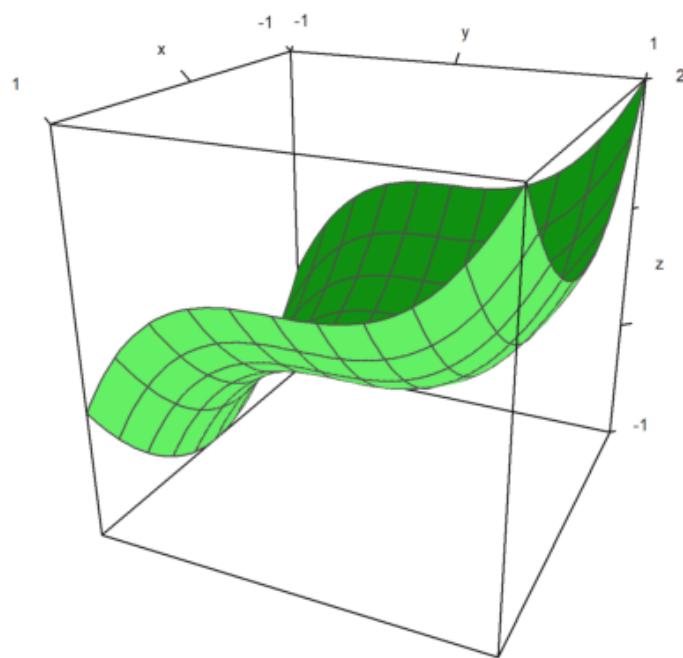
```
>myplot(x,y,z);
```



Animasi

Euler dapat menggunakan frame untuk melakukan pra-komputasi animasi. Salah satu fungsi yang memanfaatkan teknik ini adalah rotate. Fungsi ini dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi ini memanggil addpage() untuk setiap plot baru. Terakhir, fungsi ini menganimasikan plot-plot tersebut. Silakan pelajari sumber rotasi untuk mengetahui detail selengkapnya

```
>function testplot () := plot3d("x^2+y^3"); ...
>rotate("testplot"); testplot():
```



Menggambar Povray

Dengan bantuan file Euler povray.e, Euler dapat menghasilkan file Povray. Hasilnya sangat bagus untuk dilihat.

Anda perlu menginstal Povray (32bit atau 64bit) dari <http://www.povray.org/>, dan meletakkan subdirektori "bin" dari Povray ke dalam jalur lingkungan, atau mengatur variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe".

Antarmuka Povray dari Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah current.pov, dan direktori defaultnya adalah eulerhome(), biasanya c:\Users\Username\Euler. Povray menghasilkan sebuah file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan berkas-berkas ini, gunakan povclear().

Fungsi pov3d memiliki tujuan yang sama dengan plot3d. Fungsi ini dapat menghasilkan grafik fungsi $f(x,y)$, atau permukaan dengan koordinat X,Y,Z dalam matriks, termasuk garis level opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat adegan ke dalam notebook Euler.

Selain pov3d(), ada banyak fungsi yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string, yang berisi kode Povray untuk objek. Untuk menggunakan fungsi-fungsi ini, mulai file Povray dengan povstart(). Kemudian gunakan writeln(...) untuk menulis objek ke file scene. Terakhir, akhiri file dengan povend(). Secara default, raytracer akan dimulai, dan PNG akan dimasukkan ke dalam notebook Euler.

Fungsi objek memiliki parameter yang disebut "look", yang membutuhkan string dengan kode Povray untuk tekstur dan hasil akhir objek. Fungsi povlook() dapat digunakan untuk

menghasilkan string ini. Fungsi ini memiliki parameter untuk warna, transparansi, Phong Shading, dll.

Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi, Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk vertikal ke atas, dan sumbu x, y, z di tangan kanan.

Anda perlu memuat file povray

```
>load povray;
```

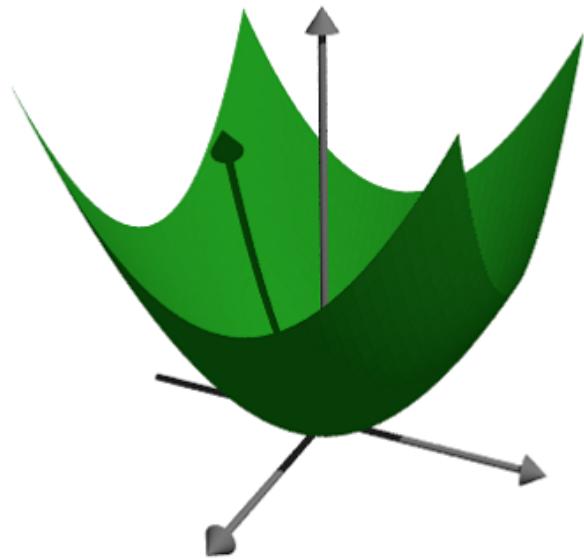
Pastikan direktori bin povray berada di dalam path. Jika tidak, edit variabel berikut sehingga berisi jalur ke povray yang dapat dieksekusi.

```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe

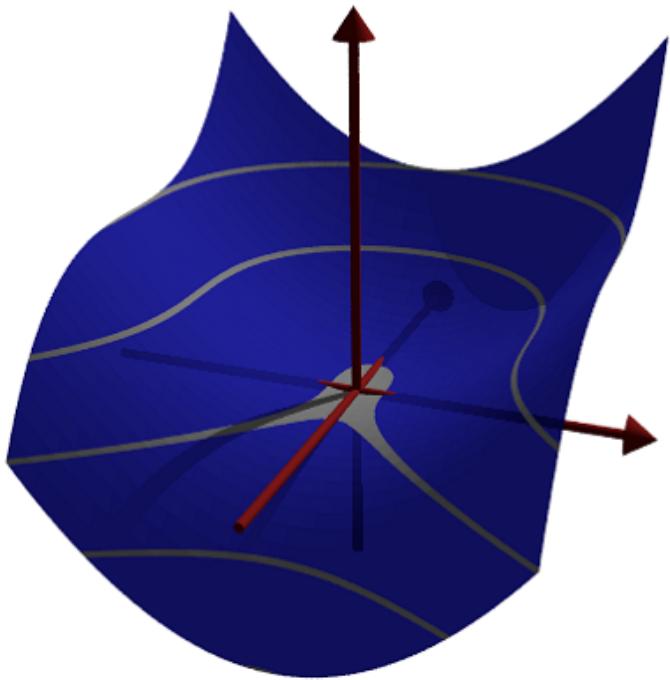
Sebagai kesan pertama, kita merencanakan sebuah fungsi sederhana. Perintah berikut ini menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk melacak sinar pada file ini. Jika Anda menjalankan perintah berikut, GUI Povray akan membuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya apakah Anda ingin mengizinkan file exe untuk dijalankan. Anda dapat menekan cancel untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK pada jendela Povray untuk mengetahui dialog awal Povray.

```
>pov3d("x^2+y^2", zoom=3);
```



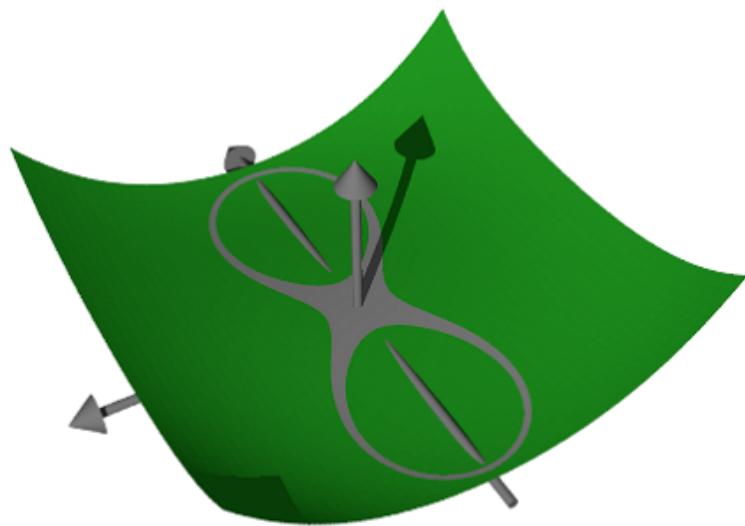
Kita dapat membuat fungsi menjadi transparan dan menambahkan hasil akhir lainnya. Kita juga dapat menambahkan garis level ke plot fungsi.

```
>pov3d("x^2+y^3",axiscolor=red,angle=20°, ...
> look=povlook(blue,0.2),level=-1:0.5:1,zoom=3.8);
```



Kadang-kadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan. Kami memplot kumpulan titik pada bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=1.5, ...
> angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=45°,n=50, ...
> <fscale,zoom=3.8);
```



Plotting dengan Koordinat

Selain menggunakan fungsi, kita dapat membuat plot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objek.

Pada contoh, kita memutar sebuah fungsi pada sumbu z

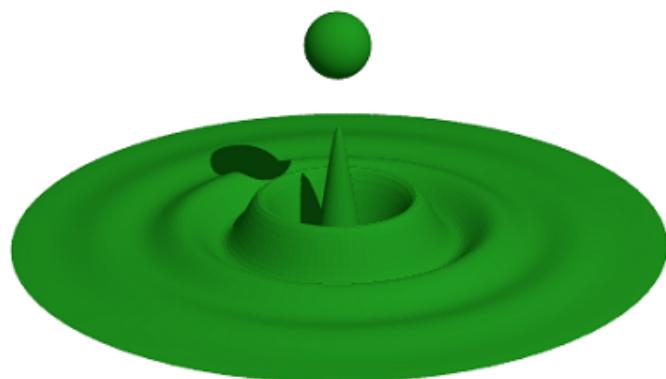
```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,8)';
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,height=20°,axis=0,zoom=4,light=[10,-5,5]);
```



Pada contoh berikut, kita memplot gelombang teredam. Kami menghasilkan gelombang dengan bahasa matriks Euler. Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut.

Perhatikan bahwa plot3d menskalakan plot, sehingga sesuai dengan kubus satuan.

```
>r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ...
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3; ...
>pov3d(x,y,z, zoom=5, axis=0, add=povsphere([0,0,0.5],0.1,povlook(green)), ...
>w=500,h=300);
```



Dengan metode bayangan canggih Povray, hanya sedikit titik yang bisa menghasilkan permukaan yang sangat halus. Hanya pada batas-batas dan bayangan, trik ini bisa terlihat jelas.

Untuk itu, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>Z &= x^2*y^3
```

$$\begin{matrix} 2 & 3 \\ x & y \end{matrix}$$

Persamaan permukaannya adalah $[x,y,Z]$. Kita menghitung dua turunan terhadap x dan y dari persamaan ini dan mengambil hasil perkalian silang sebagai normal.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

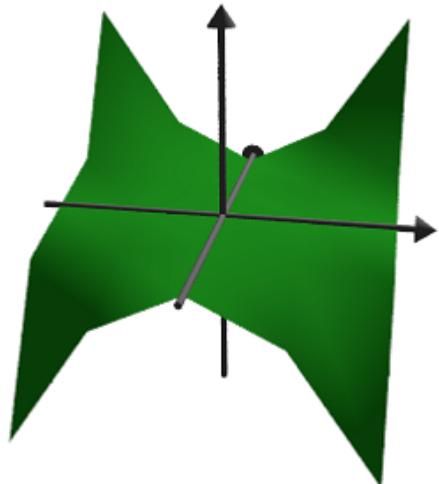
Kita mendefinisikan normal sebagai hasil kali silang dari turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

$$\begin{bmatrix} 3 & 2 & 2 \\ -2x & -3x & y & 1 \end{bmatrix}$$

Kita hanya menggunakan 25 poin

```
>x=-1:0.5:1; y=x';
>pov3d(x,y,Z(x,y),angle=10°, ...
> xv=NX(x,y), yv=NY(x,y), zv=NZ(x,y), <shadow>;
```



Berikut ini adalah simpul Trefoil yang dibuat oleh A. Busser di Povray. Ada versi yang lebih baik dari ini dalam contoh.

See: Examples\Trefoil Knot | Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normal untuk kami. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolis.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
>Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
>Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunan terhadap x dan y

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang yang normal, yang merupakan produk silang dari dua turunan.

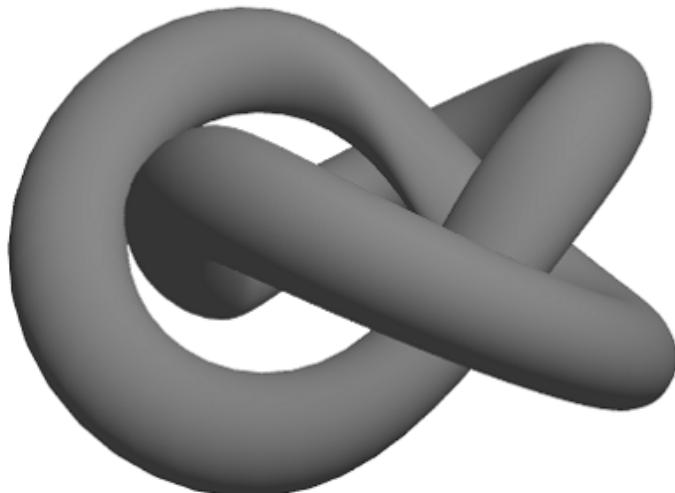
```
>dn &= crossproduct(dx,dy);
```

Kita sekarang mengevaluasi semua ini secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

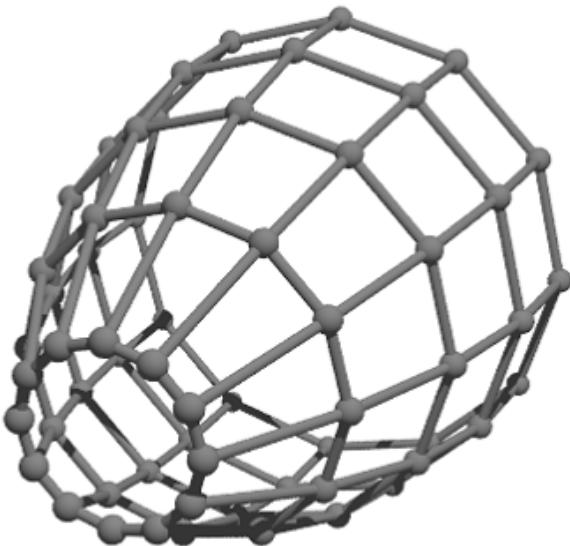
Vektor normal adalah evaluasi dari ekspresi simbolik $dn[i]$ untuk $i=1,2,3$. Sintaks untuk ini adalah &"expression"(parameter). Ini adalah sebuah alternatif dari metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),axis=0,zoom=5,w=450,h=350, ...
>    <shadow,look=povlook(gray), ...
>    xv=&"dn[1] "(x,y), yv=&"dn[2] "(x,y), zv=&"dn[3] "(x,y));
```



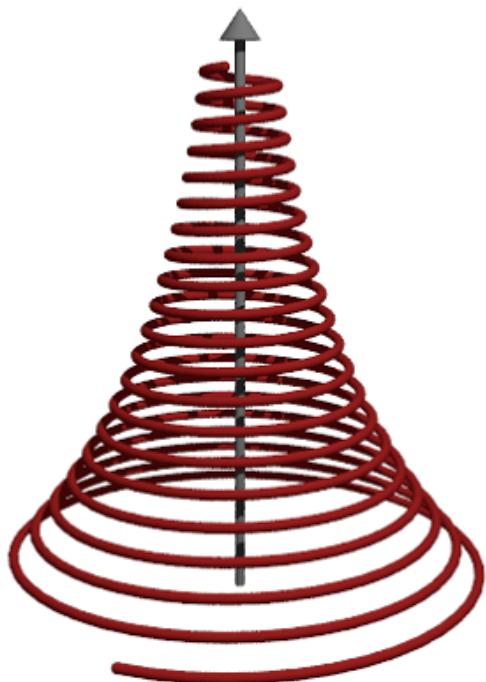
Kita juga dapat menghasilkan kisi-kisi dalam bentuk 3D

```
>povstart(zoom=4); ...
>x=-1:0.5:1; r=1-(x+1)^2/6; ...
>t=(0°:30°:360°)'; y=r*cos(t); z=r*sin(t); ...
>writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...
>povend();
```



Dengan povgrid(), kurva dapat dibuat

```
>povstart(center=[0,0,1],zoom=3.6); ...
>t=linspace(0,2,1000); r=exp(-t); ...
>x=cos(2*pi*10*t)*r; y=sin(2*pi*10*t)*r; z=t; ...
>writeln(povgrid(x,y,z,povlook(red))); ...
>writeAxis(0,2,axis=3); ...
>povend();
```



Objek Povray

Di atas, kami menggunakan pov3d untuk memplot permukaan. Antarmuka povray di Euler juga dapat menghasilkan objek Povray. Objek-objek ini disimpan sebagai string di Euler, dan perlu ditulis ke file Povray. Kita memulai output dengan povstart().

```
>povstart (zoom=4);
```

Pertama, kita mendefinisikan tiga silinder, dan menyimpannya dalam string di Euler. Fungsi povx() dll. hanya mengembalikan vektor [1,0,0], yang dapat digunakan sebagai gantinya.

```
>c1=povcylinder (-povx,povx,1,povlook(red)); ...
>c2=povcylinder (-povy,povy,1,povlook(yellow)); ...
>c3=povcylinder (-povz,povz,1,povlook(blue)); ...
```

String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

```
>c1
```

```
cylinder { <-1,0,0>, <1,0,0>, 1
    texture { pigment { color rgb <0.564706,0.0627451,0.0627451> } }
    finish { ambient 0.2 }
}
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda. Hal itu dilakukan dengan povlook(), yang mengembalikan sebuah string dengan kode Povray yang relevan. Kita dapat menggunakan warna default Euler, atau menentukan warna kita sendiri. Kita juga dapat menambahkan transparansi, atau mengubah cahaya sekitar

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

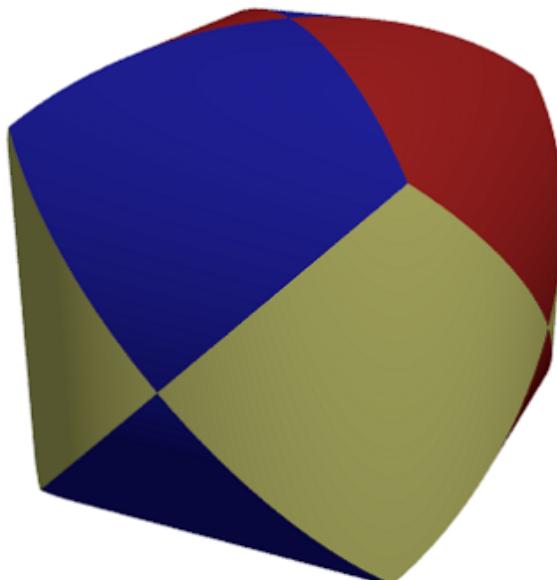
```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> } }
finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek perpotongan, dan menulis hasilnya ke file.

```
>writeln(povintersection([c1,c2,c3]));
```

Perpotongan tiga silinder sulit dibayangkan, jika Anda belum pernah melihatnya

```
>povend;
```



Fungsi-fungsi berikut ini menghasilkan fraktal secara rekursif.

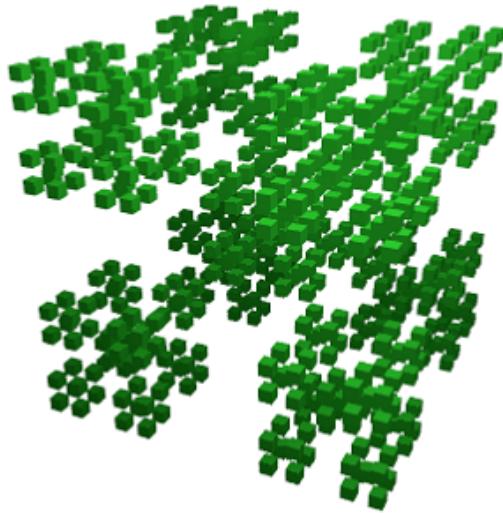
Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana.

Fungsi povbox() mengembalikan sebuah string, yang berisi koordinat kotak, tekstur dan hasil akhir.

```
>function onebox(x,y,z,d) := povbox([x,y,z],[x+d,y+d,z+d],povlook());  
>function fractal (x,y,z,h,n) ...
```

```
if n==1 then writeln(onebox(x,y,z,h));  
else  
    h=h/3;  
    fractal(x,y,z,h,n-1);  
    fractal(x+2*h,y,z,h,n-1);  
    fractal(x,y+2*h,z,h,n-1);  
    fractal(x,y,z+2*h,h,n-1);  
    fractal(x+2*h,y+2*h,z,h,n-1);  
    fractal(x+2*h,y,z+2*h,h,n-1);  
    fractal(x,y+2*h,z+2*h,h,n-1);  
    fractal(x+2*h,y+2*h,z+2*h,h,n-1);  
    fractal(x+h,y+h,z+h,h,n-1);  
endif;  
endfunction
```

```
>povstart(fade=10,<shadow>;  
>fractal(-1,-1,-1,2,4);  
>povend();
```



Perbedaan memungkinkan pemotongan satu objek dari objek lainnya. Seperti persimpangan, ada bagian dari objek CSG Povray.

```
>povstart (light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kita akan mendefinisikan sebuah objek di Povray, alih-alih menggunakan sebuah string di Euler. Definisi akan langsung dituliskan ke file. Koordinat kotak -1 berarti [-1,-1,-1]

```
>povdefine ("mycube",povbox (-1,1));
```

Kita dapat menggunakan objek ini dalam povobject(), yang mengembalikan sebuah string seperti biasa.

```
>c1=povobject ("mycube",povlook (red));
```

Kita menghasilkan kubus kedua, dan memutar serta menskalakannya sedikit.

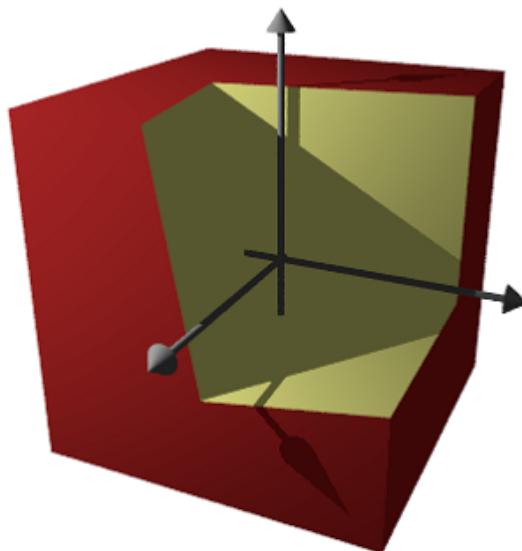
```
>c2=povobject ("mycube",povlook (yellow),translate=[1,1,1], ...
> rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita ambil selisih dari kedua objek tersebut.

```
>writeln(povdifference(c1,c2));
```

Sekarang tambahkan tiga sumbu

```
>writeAxis(-1.2,1.2,axis=1); ...
>writeAxis(-1.2,1.2,axis=2); ...
>writeAxis(-1.2,1.2,axis=4); ...
>povend();
```



Fungsi Implisit

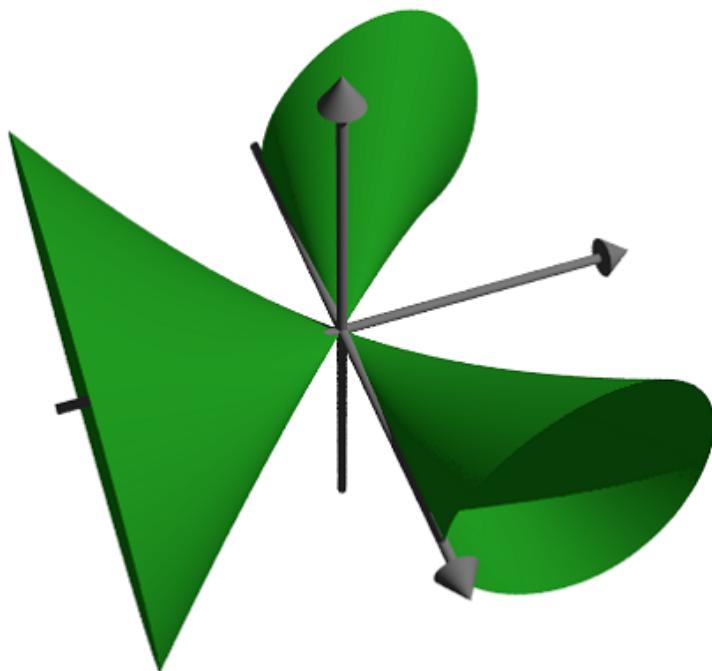
Povray dapat memplot himpunan di mana $f(x,y,z)=0$, seperti parameter implisit pada plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsi-fungsi tersebut sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

```
>povstart(angle=70°, height=50°, zoom=4);
```

Membuat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi.

```
>writeln(povsurface("pow(x, 2) *y-pow(y, 3)-pow(z, 2)", povlook(green))); ...
>writeAxes(); ...
>povend();
```



Objek Jaring

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan. Kami ingin memaksimalkan xy di bawah kondisi $x+y = 1$ dan mendemonstrasikan sentuhan tangensial dari garis level.

```
>povstart(angle=-10°, center=[0.5, 0.5, 0.5], zoom=7);
```

Kita tidak dapat menyimpan objek dalam sebuah string seperti sebelumnya, karena ukurannya terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan deklarasikan. Fungsi povtriangle() melakukan hal ini secara otomatis. Fungsi ini dapat menerima vektor normal seperti halnya pov3d().

Berikut ini mendefinisikan objek mesh, dan langsung menuliskannya ke dalam file

```
>x=0:0.02:1; y=x'; z=x*y; vx=-y; vy=-x; vz=1;
>mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kita tnetukan 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);
```

Tuliskan permukaan dikurangi kedua cakram

```
>writeln(povdifference(mesh,povunion([cl,l1]),povlook(green)));
```

Tuliskan kedua perpotongan tersebut

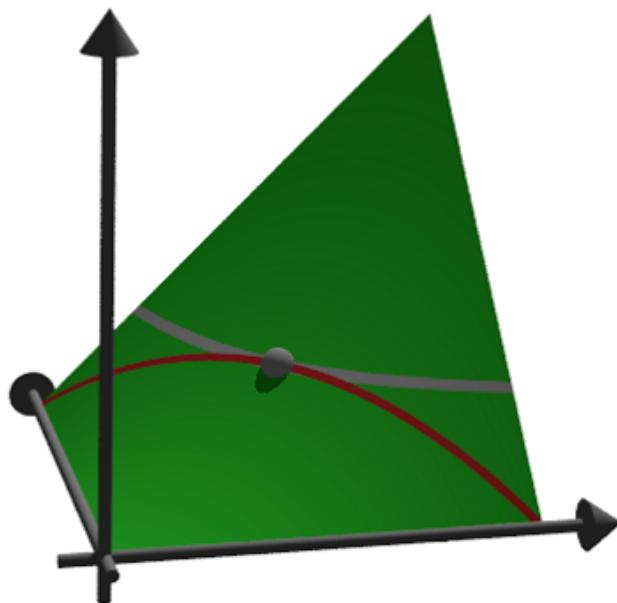
```
>writeln(povintersection([mesh,cl],povlook(red))); ...
>writeln(povintersection([mesh,l1],povlook(gray)));
```

Tulislah satu titik secara maksimal

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));
```

Tambahkan sumbu dan selesaikan

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...
>povend();
```

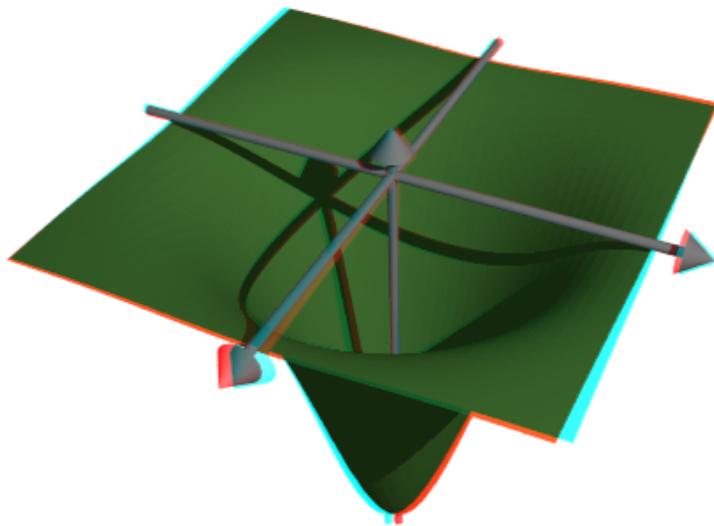


Anaglyph di Povray

Untuk menghasilkan anaglyph untuk kacamata merah/cyan, Povray harus dijalankan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph(). Tentu saja, Anda memerlukan kacamata merah/cyan untuk melihat contoh berikut ini dengan benar.

Fungsi pov3d() memiliki saklar sederhana untuk menghasilkan anaglyph.

```
>pov3d("-exp (-x^2-y^2) / 2", r=2, height=45°, >anaglyph, ...
>    center=[0, 0, 0.5], zoom=3.5);
```



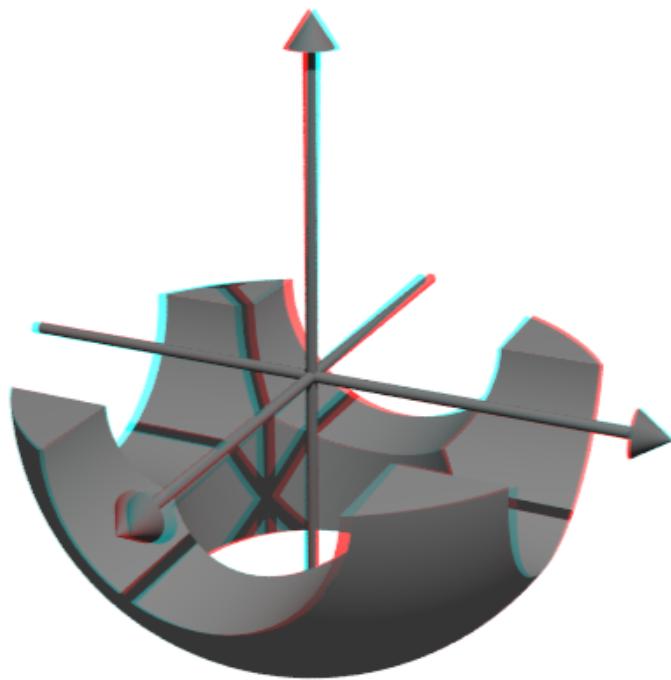
Jika Anda membuat scene dengan objek, Anda harus menempatkan pembuatan scene ke dalam fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph

```
>function myscene ...
```

```
s=povsphere(povc,1);
cl=povcylinder(-povz,povz,0.5);
clx=povobject(cl,rotate=xrotate(90°));
cly=povobject(cl,rotate=yrotate(90°));
c=povbox([-1,-1,0],1);
un=povunion([cl,clx,cly,c]);
obj=povdifference(s,un,povlook(red));
writeln(obj);
writeAxes();
endfunction
```

Fungsi povanaglyph() melakukan semua ini. Parameter-parameternya seperti pada povstart() dan povend() yang digabungkan

```
>povanaglyph ("myscene", zoom=4.5);
```



Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak sekali objek. Tetapi Anda tidak dibatasi pada objek-objek tersebut. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau objek yang benar-benar baru.

Kita mendemonstrasikan sebuah torus. Perintah Povray untuk ini adalah "torus". Jadi kita mengembalikan sebuah string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat di titik asal.

```
>function povdonat (r1,r2,look="") ...
```

```
    return "torus {" + r1 + "," + r2 + look + "}";
  endfunction
```

Inilah torus pertama kita

```
>t1=povdonat (0.8,0.2)
```

```
torus {0.8,0.2}
```

Mari kita gunakan objek ini untuk membuat torus kedua, diterjemahkan dan diputar

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
object { torus { 0.8,0.2 }
    rotate 90 *x
    translate <0.8,0,0>
}
```

Sekarang, kita tempatkan semua benda ini ke dalam suatu pemandangan. Untuk tampilannya, kami menggunakan Phong Shading.

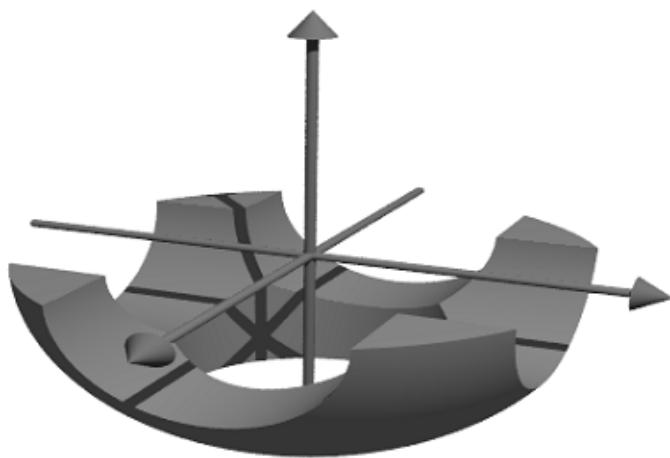
```
>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...
>writeln(povobject(t1,povlook(green,phong=1))); ...
>writeln(povobject(t2,povlook(green,phong=1))); ...
```

```
>povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, program ini tidak menampilkan kesalahan. Oleh karena itu, Anda harus menggunakan

`>povend(<exit>);` jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray terbuka.

```
>povend(h=320,w=480);
```



Berikut adalah contoh yang lebih rumit. Kita selesaikan

$$Ax \leq b, \quad x \geq 0, \quad c.x \rightarrow \text{Max.}$$

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
>b=[10,10,10,10]';
>c=[1,1,1];
```

pertama mari kita periksa, apakah contoh ini memiliki solusi atau tidak

```
>x=simplex(A,b,c,>max,>check)'
```

[0, 1, 0.5]

Yes, benar

Selanjutnya kita definisikan dua objek, yang pertama adalah pesawat

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look=""') ...
```

```
    return povplane(a,b,look)
endfunction
```

Kemudian kita mendefinisikan perpotongan semua setengah ruang dan kubus

```
>function adm (A, b, r, look=""') ...
```

```
ol=[];
loop 1 to rows(A); ol=ol|oneplane(A[#,b[#]]; end;
ol=ol|povbox([0,0,0],[r,r,r]);
return povintersection(ol,look);
endfunction
```

Sekarang kita bisa merencakan adegan tersebut

```
>povstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); ...
>writeln(adm(A,b,2,povlook(green,0.4))); ...
>writeAxes(0,1.3,0,1.6,0,1.5); ...
```

Berikut ini adalah lingkaran di sekeliling optimal

```
>writeln(povintersection([povsphere(x, 0.5), povplane(c, c.x')], ...
>  povlook(red, 0.9)));
```

```
intersection { sphere { <0,1,0.5>, 0.5 texture { pigment { color rgb <0.470588,0.470588,0.470588> } ...
  finish { ambient 0.2 }

}

plane { <1,1,1>, 0.866025403784
  texture { pigment { color rgb <0.470588,0.470588,0.470588> } ...
  finish { ambient 0.2 }

}

texture { pigment { color rgbf <0.564706,0.0627451,0.0627451,0.9> } ...
  finish { ambient 0.2 }

}
```

Dan kesalahan pada arah yang optimal

```
>writeln(povarrow(x, c*0.5, povlook(red)));
```

```
union { cylinder { <0,1,0.5>, <0.5,1.5,1>, 0.02
  texture { pigment { color rgb <0.470588,0.470588,0.470588> } ...
  finish { ambient 0.2 }

}

cone { <0.5,1.5,1>, 0.06 <0.557735,1.55774,1.05774>, 0 texture { pigment
  finish { ambient 0.2 }

}

texture { pigment { color rgb <0.564706,0.0627451,0.0627451> } ...
  finish { ambient 0.2 }

}
```

Kita menambahkan teks ke layar. Teks hanyalah sebuah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

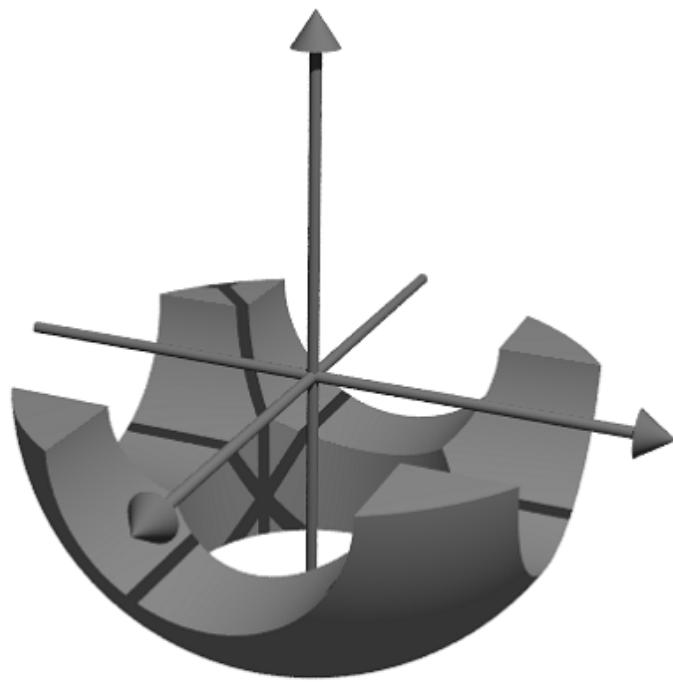
```
>writeln(povtext("Linear Problem", [0,0.2,1.3], size=0.05, rotate=5°)); ...
>povend();
```

```
object { text { ttf "timrom.ttf" "Linear Problem" 0.04 ,0  texture { pigment
  finish { ambient 0.2 }

}

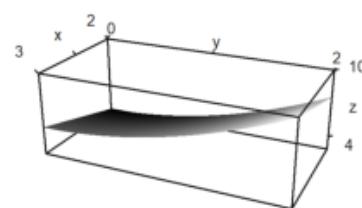
scale 0.05
rotate 90 *x  rotate 5 *z
translate <0,0.2,1.3>

}
```

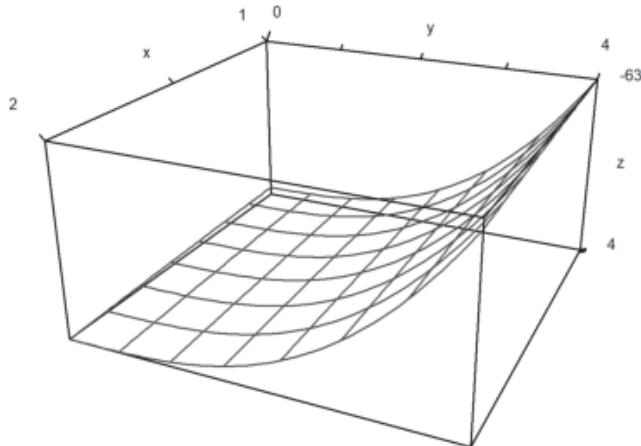


Contoh lain

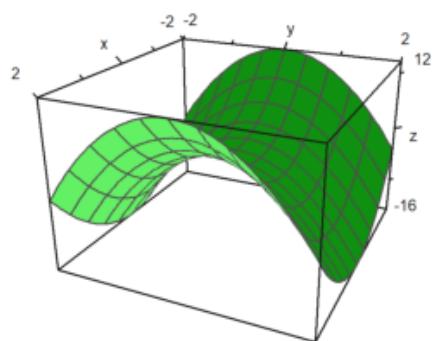
```
>plot3d("2*x+y^2", a=2, b=3, c=0, d=2, n=300, level=2, light=[2,1,3]):
```



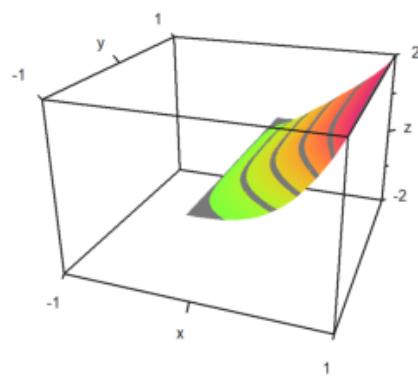
```
>plot3d("x^2-y^3", 2, 1, 0, 4, scale=[4,1,3], zoom=4, grid=8, >transparent):
```



```
>plot3d("3*x^2-4*y^2",r=2):
```



```
>plot3d("2*x^3*y^4", angle=20°,>contour,>spectral):
```



BAB 5

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

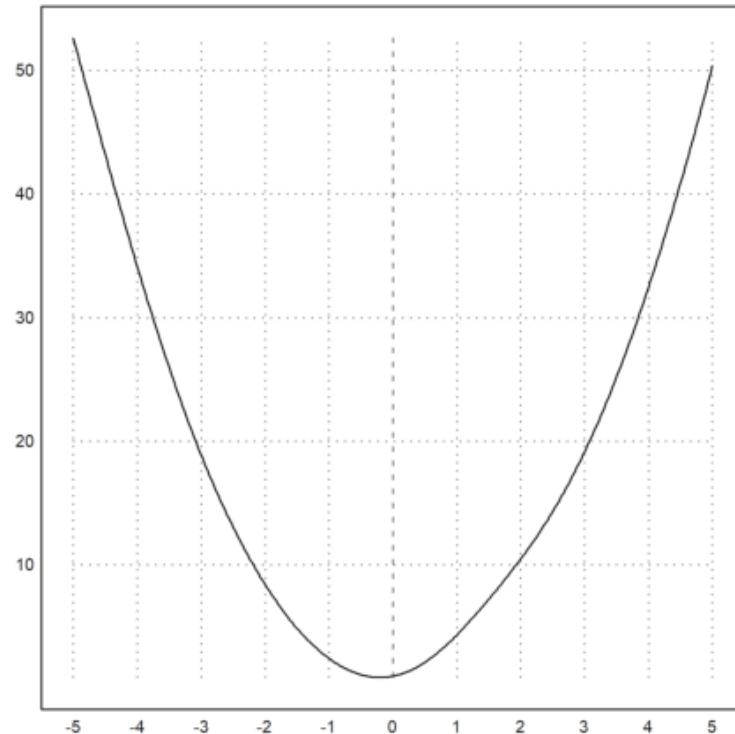
4.31977682472

20.7392088022

```
>f(a) // tidak dapat dihitung nilainya ...
```

Silakan Anda plot kurva fungsi di atas!

```
>plot2d("f(x)", -5, 5):
```



Berikutnya kita definisikan fungsi:

$$g(x) = \frac{\sqrt{x^2 - 3x}}{x + 1}.$$

```
>function g(x) := sqrt(x^2-3*x) / (x+1)
>g(3)
```

0

```
>g(0)
```

0

```
>g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik
```

Floating point error!

Error in sqrt

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

g:

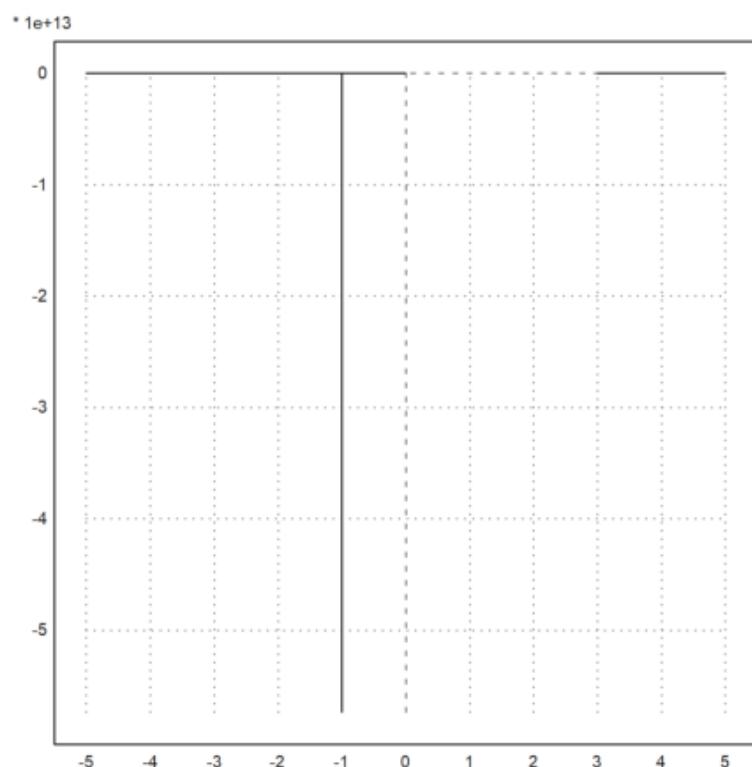
```
useglobal; return sqrt(x^2-3*x) / (x+1)
```

Error in:

```
g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik ...  
^
```

Silakan Anda plot kurva fungsi di atas!

```
>plot2d("g(x)", -5, 5):
```



```
>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:

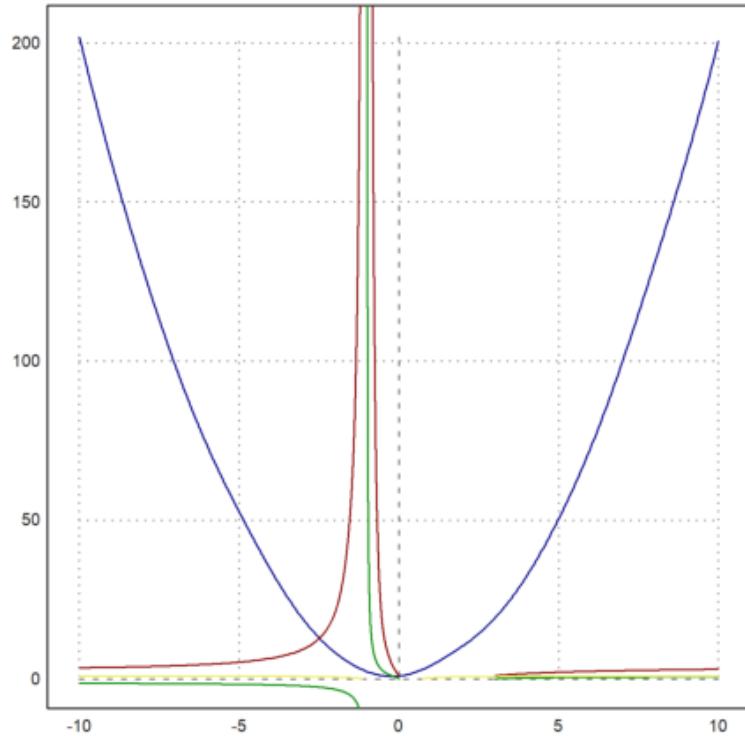
$$h(x) = f(g(x))$$

dan

$$u(x) = g(f(x))$$

bersama-sama kurva fungsi f dan g dalam satu bidang koordinat.

```
>function u(x) := g(f(x))
>plot2d("f(x)", -10, 10, color=blue);
>plot2d("g(x)", color=green, >add);
>plot2d("h(x)", color=red, >add);
>plot2d("u(x)", color=yellow, >add);
```



```
>f(0:10) // nilai-nilai f(0), f(1), f(2), ..., f(10)
```

```
[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562, 99.929,
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,
163.51, 200.58]
```

```
>gmap(200:210)
```

```
[0.987534, 0.987596, 0.987657, 0.987718, 0.987778, 0.987837, 0.987896]
```

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

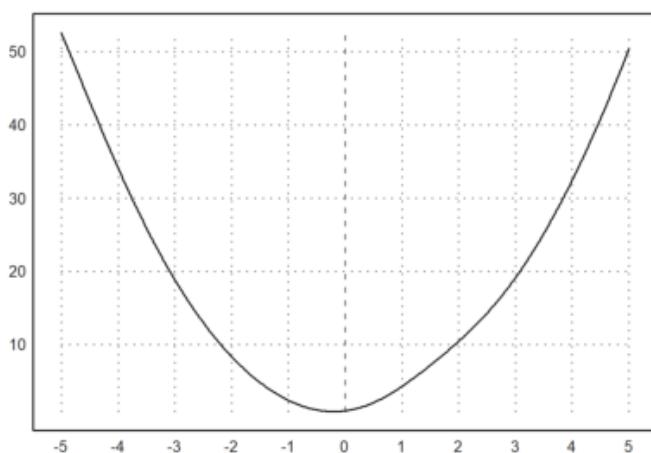
```
>f(-2)
```

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*x^x // fungsi simbolik
```

x

2 E

```
>$f(a) // nilai fungsi secara simbolik
```

$$f(a)$$

```
>f(E) // nilai fungsi berupa bilangan desimal
```

30.308524483

```
>$f(E), $float(%)
```

$$f(2.718281828459045)$$

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

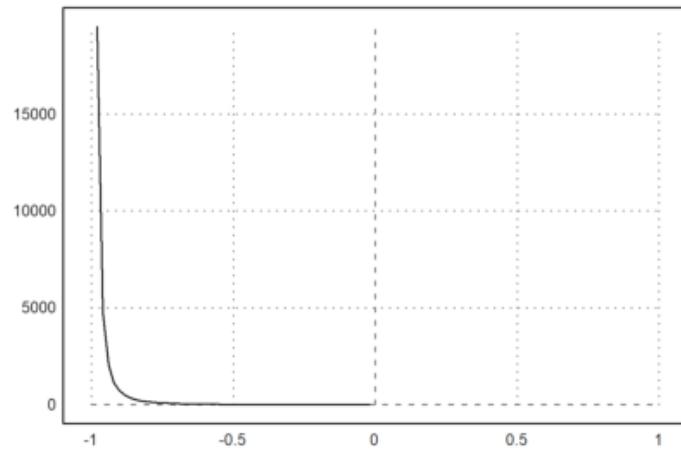
3 x + 1

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

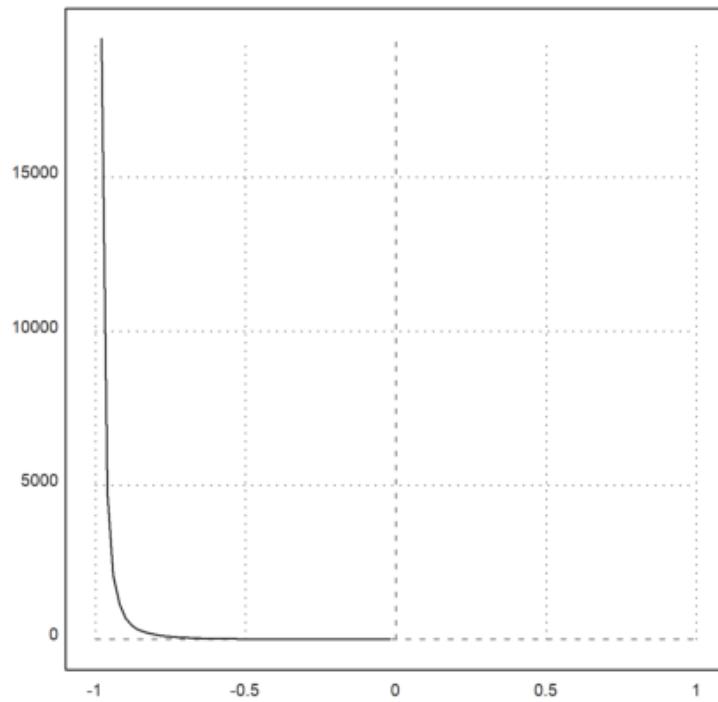
3 x + 1

2 E

```
>plot2d("h(x)", -1, 1):
```



```
>reset:
```



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.

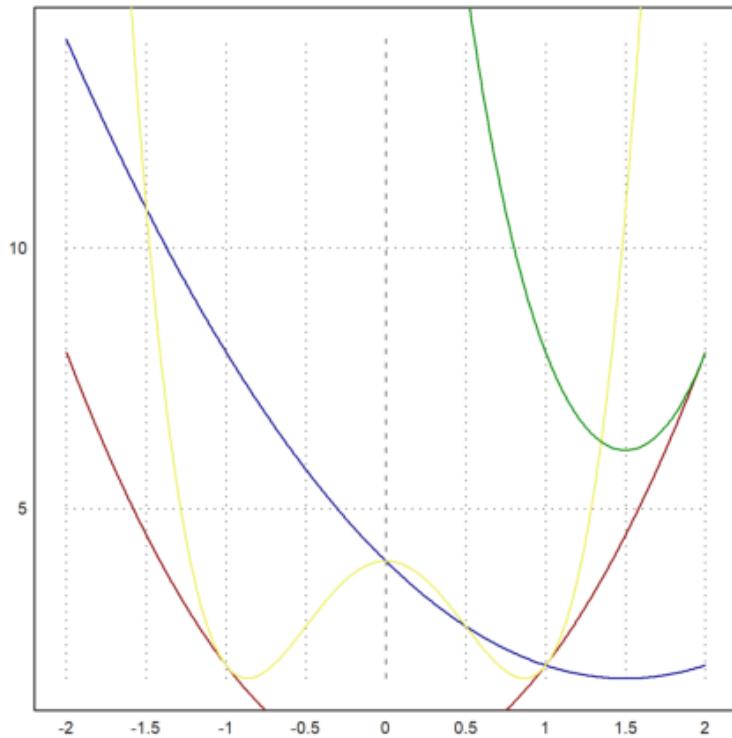
```
>function f(x) := x^2-3*x+4
>function g(x) := 2*x^2
>f(1), f(-1), f(2), g(-1), g(3)
```

2
8
2
2
18

```
>function h(x) := f(g(x))
>function j(x) := g(f(x))
>h(2), h(1), h(-1), j(2), j(3)
```

44
2
2
8
32

```
>plot2d("f(x)",color=blue), plot2d("g(x)",color=red,>add), plot2d("h(x)",co
```



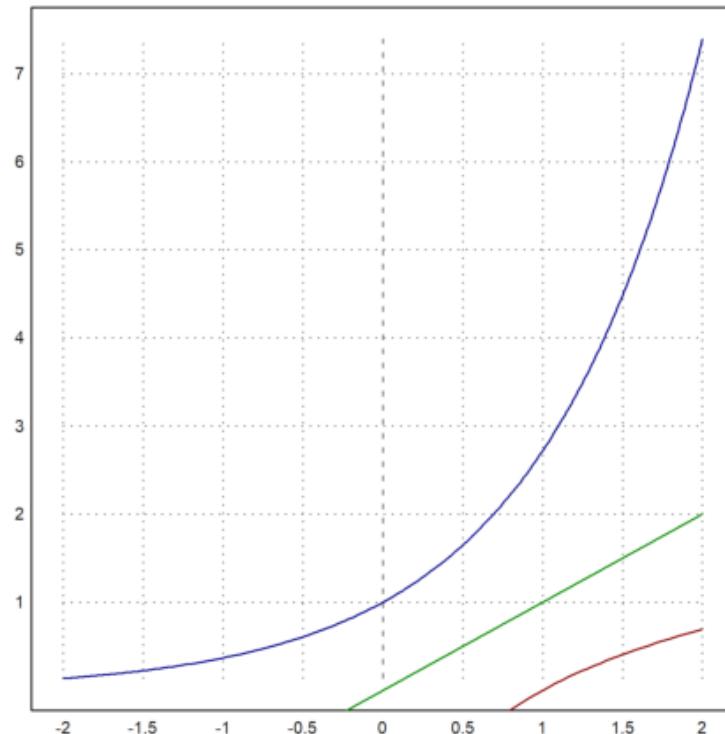
```
>function k(x) := E^x
>k(1), k(2), k(3)
```

2.71828182846
7.38905609893
20.0855369232

```
>function l(x) := log(x)
>l(1), l(2), l(3)
```

0
0.69314718056
1.09861228867

```
>function m(x) := k(l(x))
>function n(x) := l(k(x))
>plot2d("k(x)", color=blue), plot2d("l(x)", color=red, >add), plot2d("m(x)", co
```



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))
```

$$\lim_{x \rightarrow \infty} \frac{\sqrt{x^2 - 3x}}{x + 1} = 1$$

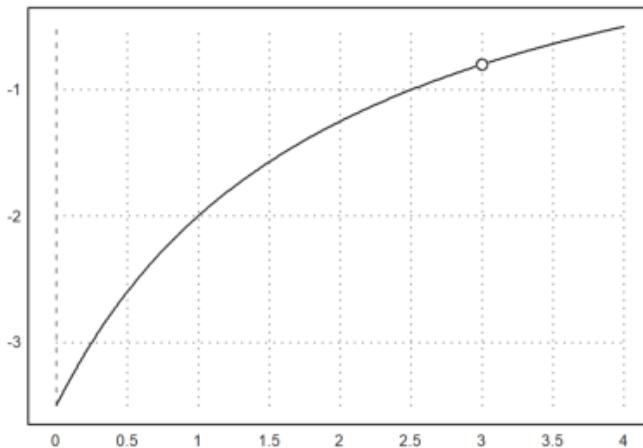
```
>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

$$-\frac{4}{5}$$

maxima: $\lim((x^3-13x^2+51x-63)/(x^3-4x^2-3x+18), x, 3) = \lim((x^3-13x^2+51x-63)/(x^3-4x^2-3x+18), x, 3)$

Fungsi tersebut diskontinu di titik $x=3$. Berikut adalah grafik fungsinya.

```
>aspect(1.5); plot2d("(x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18)", 0, 4); plot2d
```



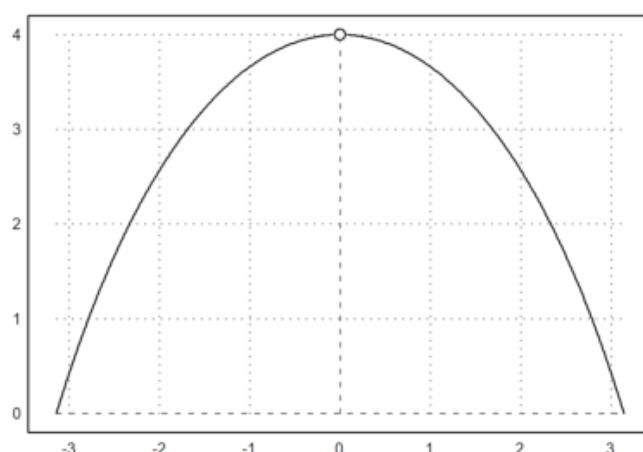
```
>$limit(2*x*sin(x)/(1-cos(x)), x, 0)
```

4

maxima: $\lim(2x\sin(x)/(1-\cos(x)), x, 0) = \lim(2x\sin(x)/(1-\cos(x)), x, 0)$

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
```



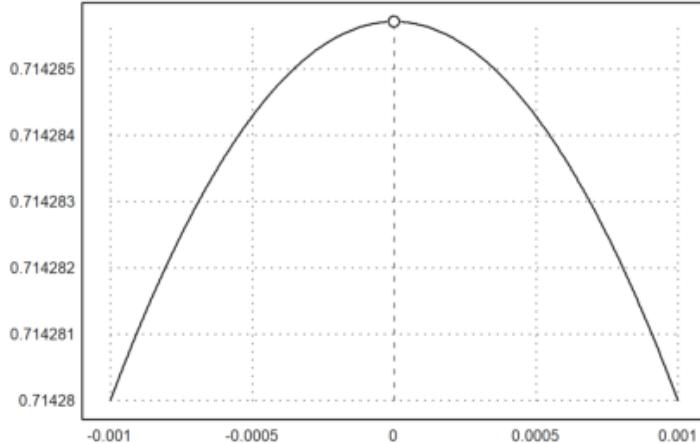
```
>$limit(cot(7*h)/cot(5*h), h, 0)
```

$$\frac{5}{7}$$

maxima: showev('limit(cot(7*h)/cot(5*h),h,0))

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di $x=0$. Berikut adalah grafiknya.

```
>plot2d("cot (7*x) /cot (5*x)", -0.001, 0.001); plot2d(0, 5/7, >points, style="ow",
```

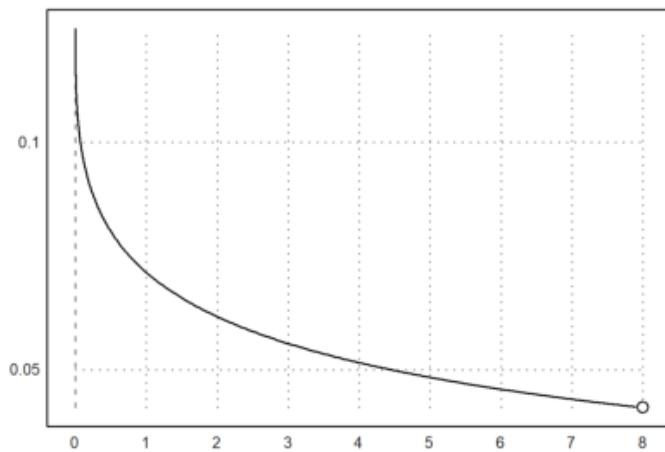


```
>$showev('limit(((x/8)^(1/3)-1)/(x-8), x, 8))
```

$$\lim_{x \rightarrow 8} \frac{\frac{x^{\frac{1}{3}}}{2} - 1}{x - 8} = \frac{1}{24}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x/8)^(1/3)-1)/(x-8)", 0, 8); plot2d(8, 1/24, >points, style="ow", >add
```



```
>$showev('limit(1/(2*x-1),x,0))
```

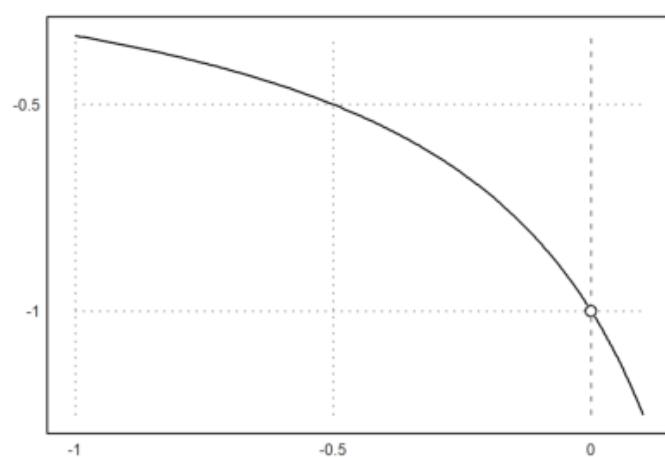
$$\lim_{x \rightarrow 0} \frac{1}{2x-1} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$limit(1/(2*x-1),x,0)
```

$$-1$$

```
>plot2d("1/(2*x-1)",-1,0.1); plot2d(0,-1,>points,style="ow",>add):
```



```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

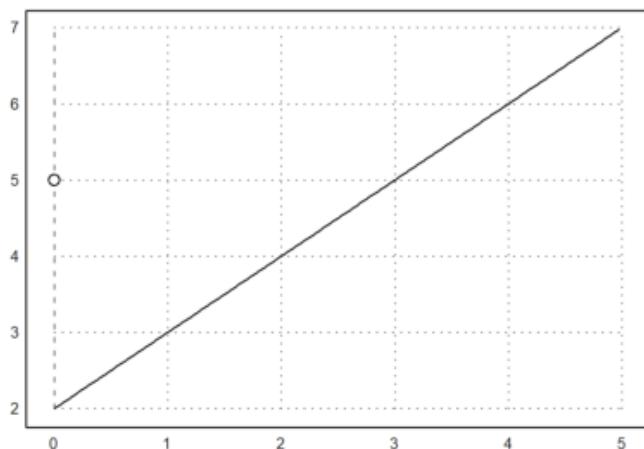
$$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$limit((x^2-3*x-10)/(x-5),x,5)
```

7

```
>plot2d("(x^2-3*x-10)/(x-5)",0,5); plot2d(0,5,>points,style="ow",>add):
```



```
>$showev('limit(sqrt(x^2+x)-x,x,inf))
```

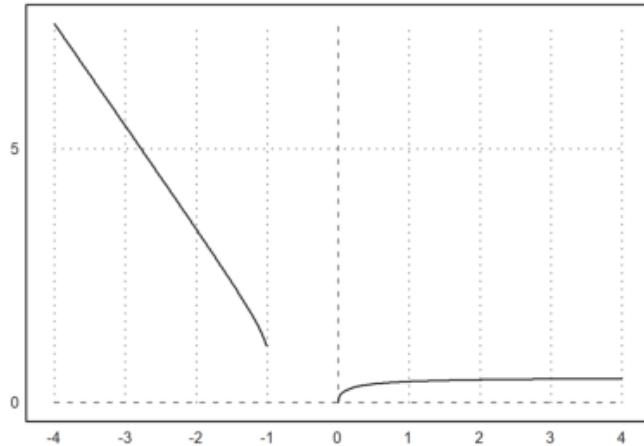
$$\lim_{x \rightarrow \infty} \sqrt{x^2 + x} - x = \frac{1}{2}$$

```
>$limit(sqrt(x^2+x)-x,x,inf)
```

$\frac{1}{2}$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("sqrt(x^2+x)-x",-4,4):
```



```
>$showev('limit(abs(x-1)/(x-1),x,1,minus))
```

$$\lim_{x \uparrow 1} \frac{|x-1|}{x-1} = -1$$

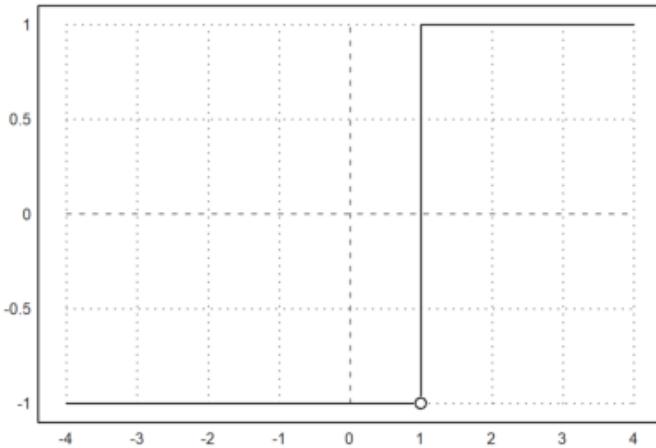
Hitung limit di atas untuk x menuju 1 dari kanan.

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$limit(abs(x-1)/(x-1),x,1,minus)
```

-1

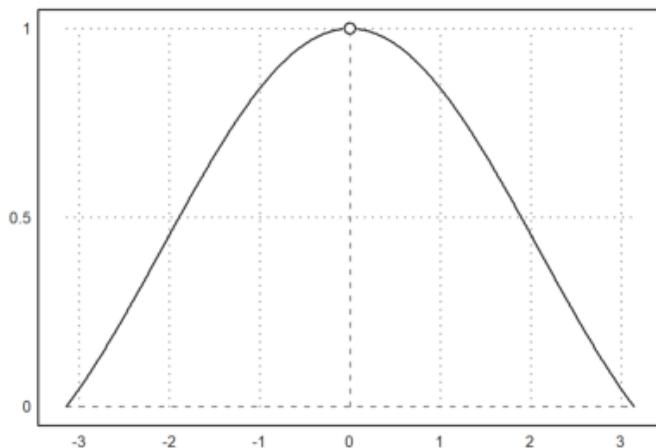
```
>plot2d("abs(x-1)/(x-1)",-4,4); plot2d(1,-1,>points,style="ow",>add):
```



```
>$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))
```

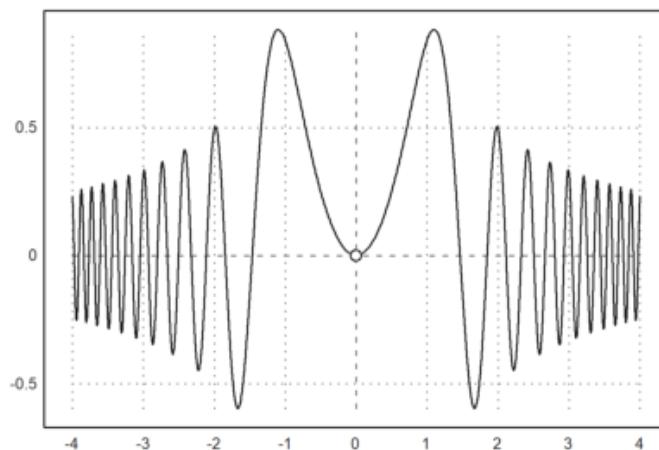
$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$limit(sin(x^3)/x,x,0)
```

0

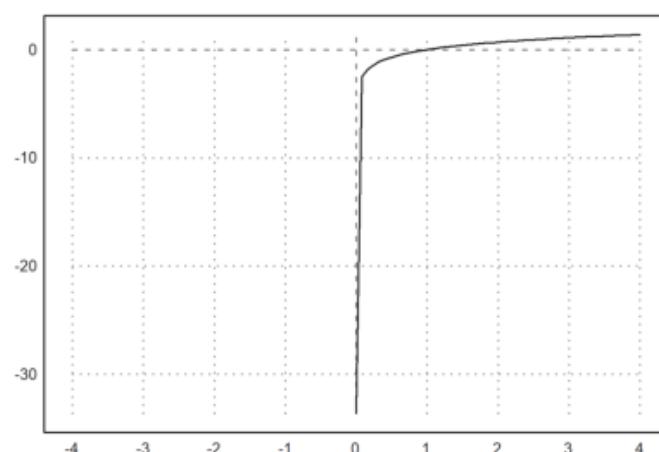
```
>plot2d("sin(x^3)/x", -4, 4); plot2d(0, 0, >points, style="ow", >add) :
```



```
>$showev('limit(log(x), x, minf))
```

$$\lim_{x \rightarrow -\infty} \log x = \text{infinity}$$

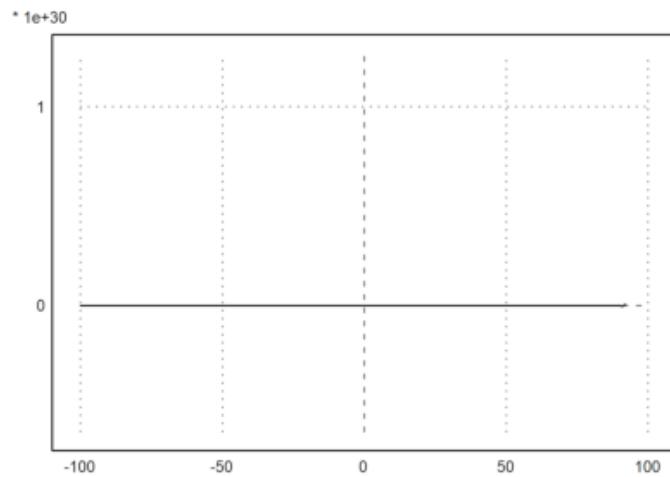
```
>plot2d("log(x)", -4, 4) :
```



```
>$showev('limit((-2)^x,x, inf))
```

$$\lim_{x \rightarrow \infty} (-2)^x = infinity$$

```
>plot2d("(-2)^x", -100, 100):
```



```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t \uparrow 2} t - \sqrt{2-t} = 2$$

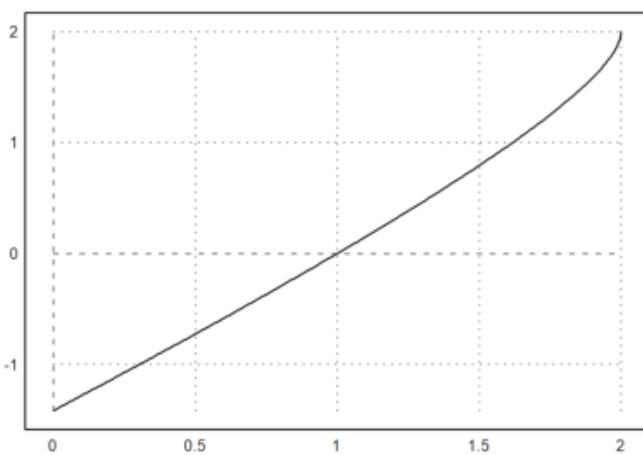
```
>$showev('limit(t-sqrt(2-t),t,2,plus))
```

$$\lim_{t \downarrow 2} t - \sqrt{2-t} = 2$$

```
>$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);
```

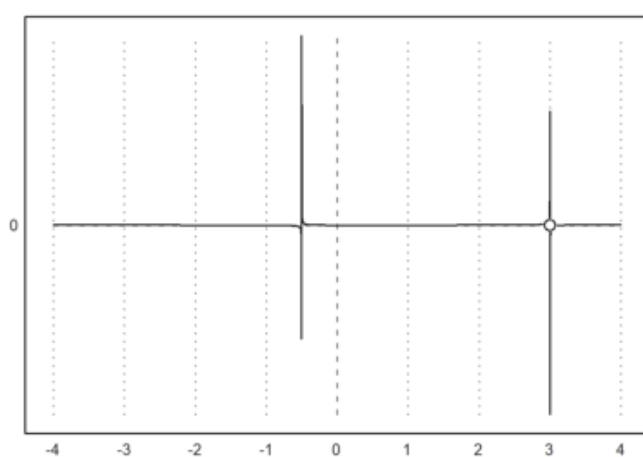


```
>$showev('limit((x^2-9)/(2*x^2-5*x-3), x, 3))
```

$$\lim_{x \rightarrow 3} \frac{x^2 - 9}{2x^2 - 5x - 3} = \frac{6}{7}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("x^2-9/(2*x^2-5*x-3)", -4, 4); plot2d(3, 6/7, >points, style="ow", >add);
```

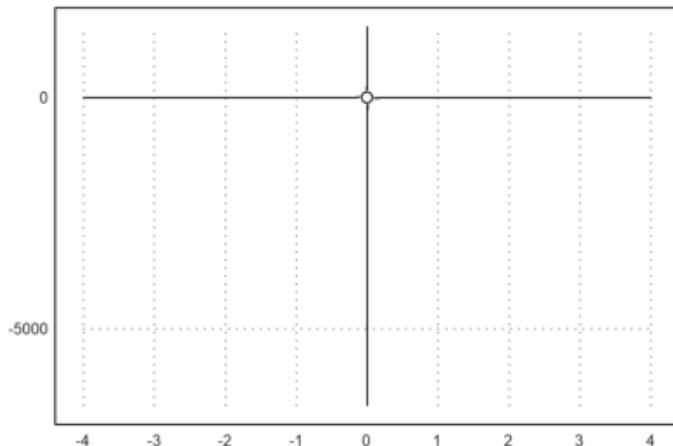


```
>showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("1-cos(x)/x", -4, 4); plot2d(0, 0, >points, style="ow", >add) :
```

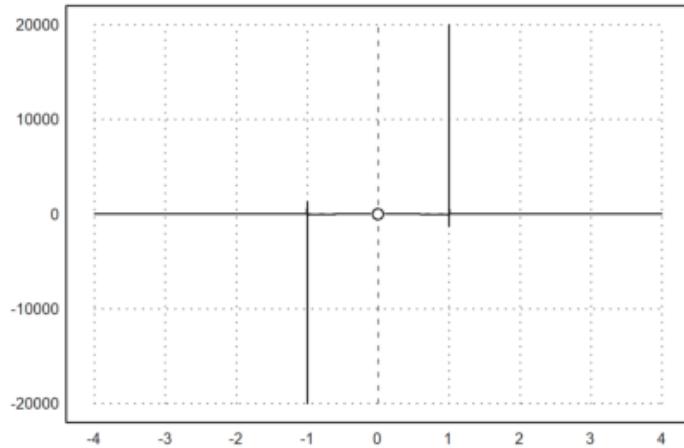


```
>showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

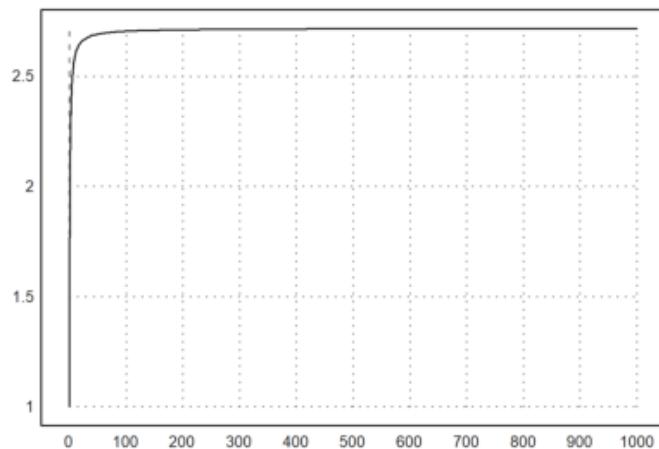
```
>plot2d("abs(x)+x^2/(x^2-abs(x))", -4, 4); plot2d(0, -1, >points, style="ow", >add)
```



```
>$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))
```

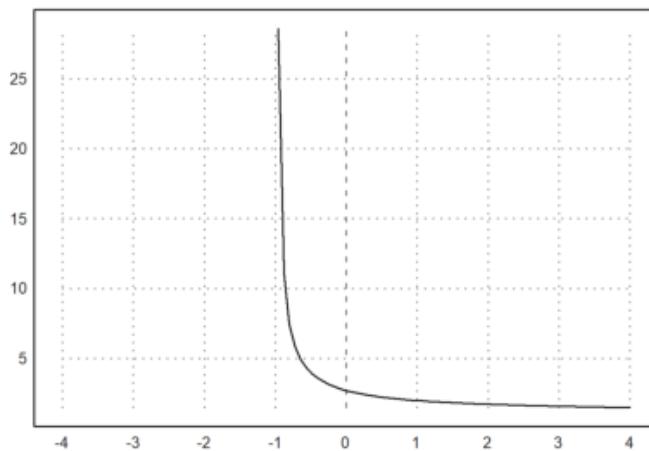
$$\lim_{x \rightarrow \infty} \left(\frac{k}{x} + 1 \right)^x = e^k$$

```
>$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)",-4,4):
```



```
>$showev('limit((x/(x+k))^x,x,inf))
```

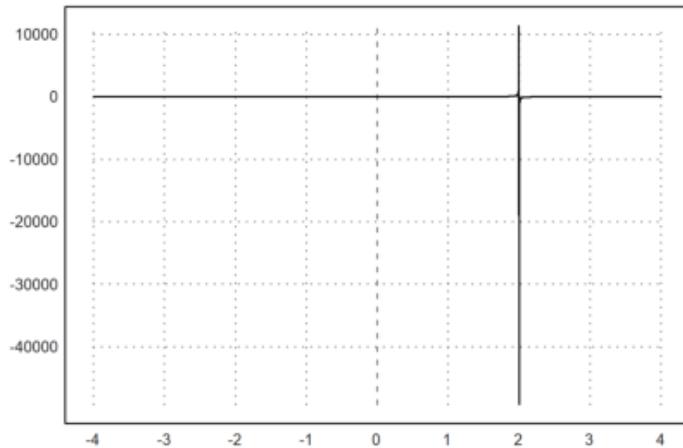
$$\lim_{x \rightarrow \infty} \left(\frac{x}{x+k} \right)^x = e^{-k}$$

```
>$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)", -4, 4):
```



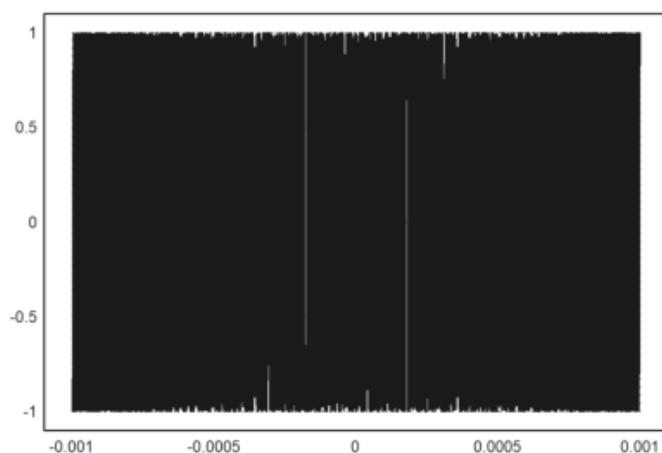
```
>$showev('limit(sin(1/x), x, 0))
```

$$\lim_{x \rightarrow 0} \sin\left(\frac{1}{x}\right) = \text{ind}$$

```
>$showev('limit(sin(1/x), x, inf))
```

$$\lim_{x \rightarrow \infty} \sin\left(\frac{1}{x}\right) = 0$$

```
>plot2d("sin(1/x)", -0.001, 0.001):
```

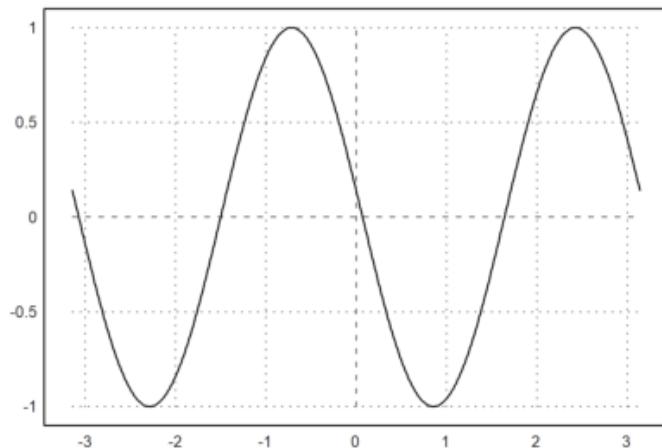


Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

```
>$showev('limit(sin(2*x+3),x,pi/3))
```

$$\lim_{x \rightarrow \frac{\pi}{3}} \sin(2x + 3) = \sin\left(\frac{2\pi + 9}{3}\right)$$

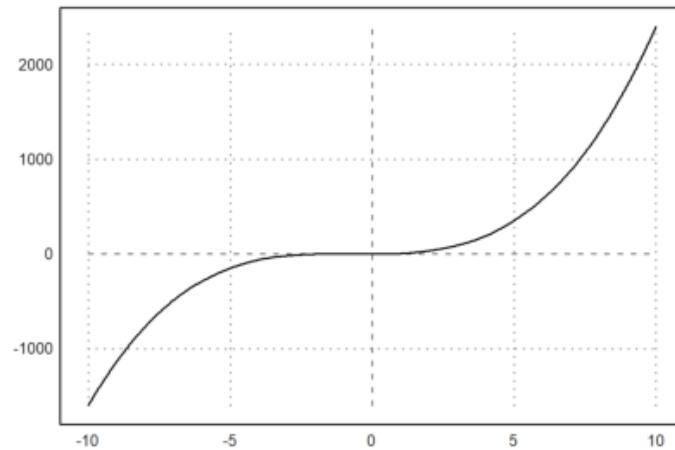
```
>plot2d("sin(2*x+3)",-pi,pi):
```



```
>$showev('limit(2*x^3+4*x^2-2,x,6))
```

$$\lim_{x \rightarrow 6} 2x^3 + 4x^2 - 2 = 574$$

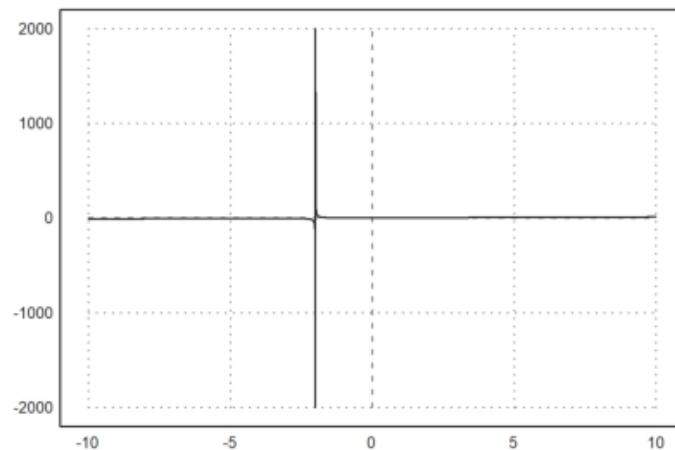
```
>plot2d("2*x^3+4*x^2-2sin(2*x)",-10,10):
```



```
>$showev('limit((x^2+3*x+4)/(x+2), x, 5))
```

$$\lim_{x \rightarrow 5} \frac{x^2 + 3x + 4}{x + 2} = \frac{44}{7}$$

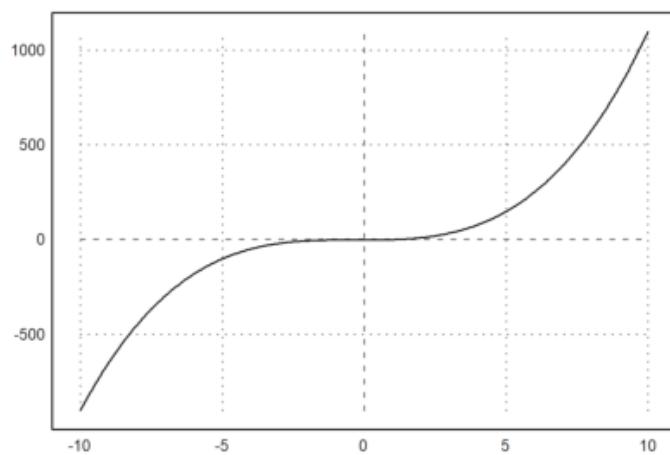
```
>plot2d("(x^2+3*x+4)/(x+2)", -10, 10):
```



```
>$showev('limit(x^3+x^2-3, x, 5))
```

$$\lim_{x \rightarrow 5} x^3 + x^2 - 3 = 147$$

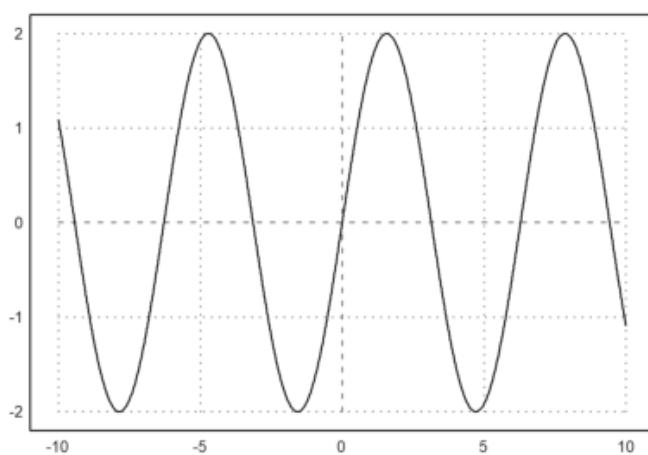
```
>plot2d("x^3+x^2-3", -10, 10):
```



```
>$showev('limit(sin(2*x)/cos(x), x, pi/6))
```

$$\lim_{x \rightarrow \frac{\pi}{6}} \frac{\sin(2x)}{\cos x} = 1$$

```
>plot2d("sin(2*x)/cos(x)", -10, 10):
```



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 disederha
```

$$2hx + h^2$$

```
>q &= ratsimp(p/h); $q // ekspresi yang akan dihitung limitnya disederhanaka
```

$$2x + h$$

```
>$limit(q,h,0) // nilai limit sebagai turunan
```

$$2x$$

```
>$showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} = n x^{n-1}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan $(x+h)^n$ dengan menggunakan teorema binomial.

```
>$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.

```
>$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.

```
>$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x
```

$$\lim_{h \rightarrow 0} \frac{\frac{1}{x+h} - \frac{1}{x}}{h} = -\frac{1}{x^2}$$

```
>$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x
```

Answering "Is x an integer?" with "integer"

Maxima is asking

294

BAB 5. KALKULUS DENGAN EMT

Acceptable answers are: yes, y, Y, no, n, N, unknown, uk

Is x an integer?

Use assume!

Error in:

```
$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x ...
```

 \wedge

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
```

 x x

```
>$showev('limit(f(x),x,0))
```

$$\lim_{x \rightarrow 0} x^x = 1$$

Silakan Anda gambar kurva

$$y = x^x.$$

```
> $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = infinity$$

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.

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

$$\sinh(x)$$

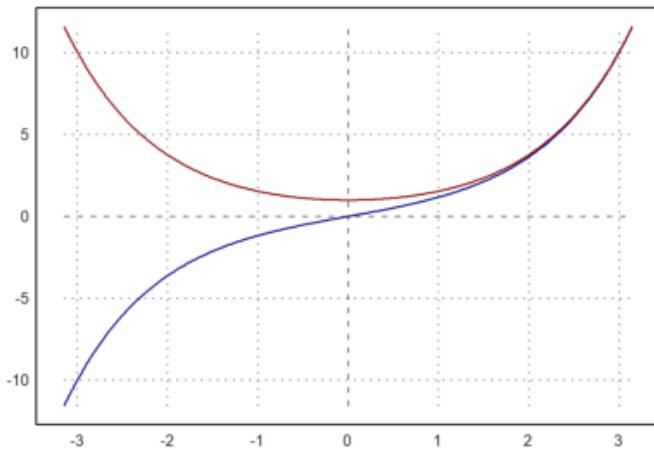
```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

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

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?

```
>$showev('diff(f(x), x))
```

$$\frac{d}{dx} \sin^2(3x^5 + 7) = 30x^4 \cos(3x^5 + 7) \sin(3x^5 + 7)$$

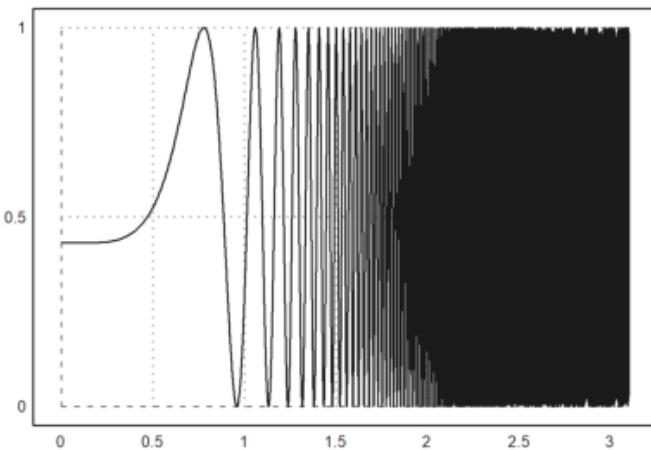
```
>$% with x=3
```

$$\%at \left(\frac{d}{dx} \sin^2(3x^5 + 7), x = 3 \right) = 2430 \cos 736 \sin 736$$

```
>${float(%)}
```

$$\% \text{at} \left(\frac{d^{1.0}}{dx^{1.0}} \sin^2 (3.0 x^5 + 7.0), x = 3.0 \right) = 1198.728637211748$$

```
>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)}
```

$$-0.2410081230863468$$

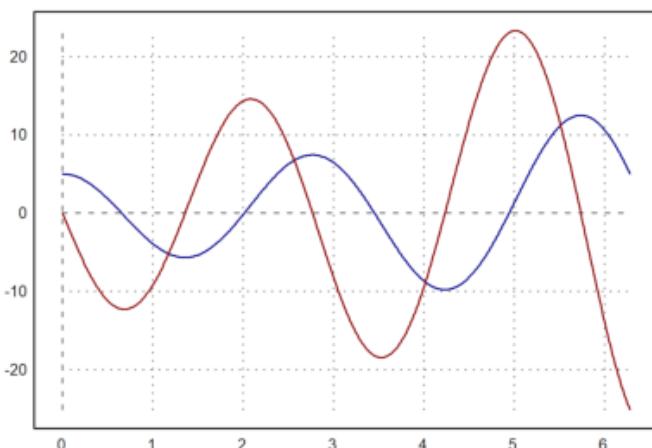
```
>xp=solve("df(x)",1,2,0) // solusi  $f'(x)=0$  pada interval [1, 2]
```

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.

```
>function f(x) &= sin(x^2-3*x)^2
```

$$\sin^2(x^2 - 3x)$$

```
>$showev('limit(f(x),x,0))
```

$$\lim_{x \rightarrow 0} \sin^2(x^2 - 3x) = 0$$

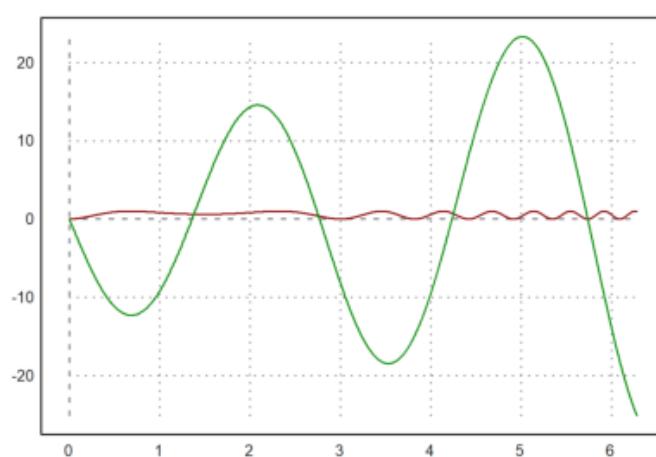
```
>$showev('limit((f(x+h)-f(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{\sin^2((x+h)^2 - 3(x+h)) - \sin^2(x^2 - 3x)}{h} = 0$$

```
>$dfx
```

$$dfx$$

```
>plot2d(["f(x)", "df(x)", 0, 2*pi, color=[red, green]):
```



```
>function f(x) &= 2*sin(2*x)+3*cos(x+1)
```

$$3 \cos(x + 1) + 2 \sin(2x)$$

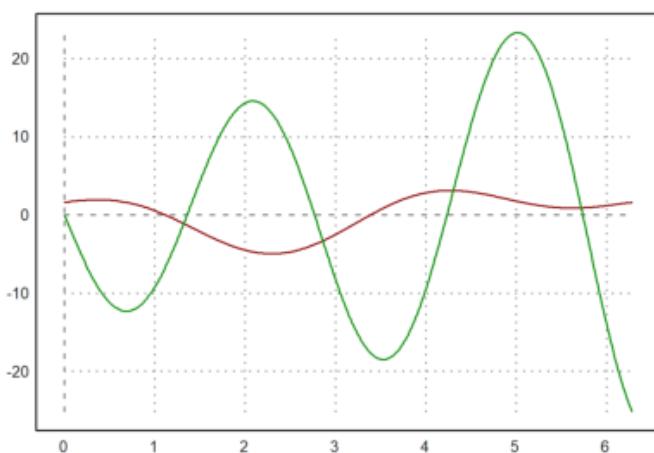
```
>$showev('limit(f(x), x, 0))
```

$$\lim_{x \rightarrow 0} 3 \cos(x + 1) + 2 \sin(2x) = 3 \cos 1$$

```
>$showev('limit((f(x+h)-f(x))/h, h, 0))
```

$$\lim_{h \rightarrow 0} \frac{3 \cos(x + h + 1) + 2 \sin(2(x + h)) - 3 \cos(x + 1) - 2 \sin(2x)}{h} = 4 \cos(2x) - 3 \sin(x + 1)$$

```
>plot2d(["f(x)", "df(x)", 0, 2*pi, color=[red, green]):
```



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

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

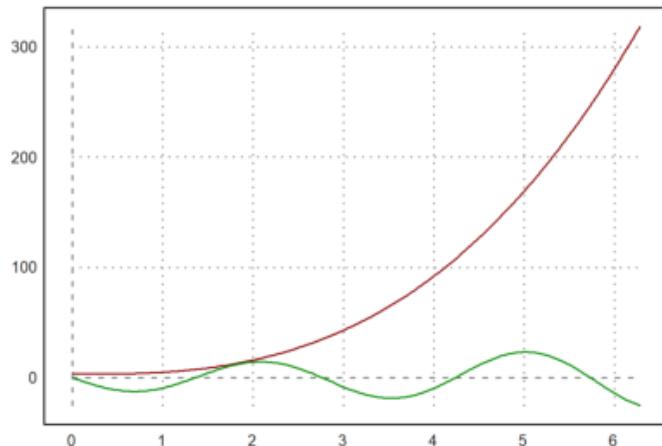
```
>$showev('limit(f(x),x,0))
```

$$\lim_{x \rightarrow 0} x^3 + 2x^2 - 2x + 4 = 4$$

```
>$showev('limit((f(x+h)-f(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^3 + 2(x+h)^2 - x^3 - 2x^2 - 2(x+h) + 2x}{h} = 3x^2 + 4x - 2$$

```
>plot2d(["f(x)", "df(x")], 0, 2*pi, color=[red, green]):
```



```
>function f(x) &= (3*x^3-2*x^2+x+5)/(x+2)
```

$$\begin{array}{r} 3 \quad \quad \quad 2 \\ 3x^3 - 2x^2 + x + 5 \\ \hline x + 2 \end{array}$$

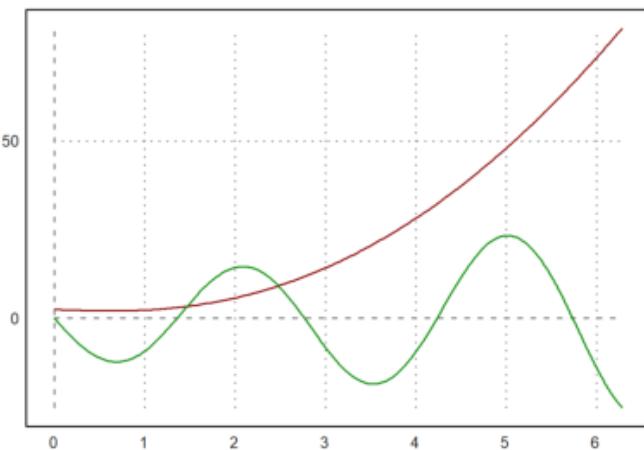
```
>$showev('limit(f(x),x,0))
```

$$\lim_{x \rightarrow 0} \frac{3x^3 - 2x^2 + x + 5}{x + 2} = \frac{5}{2}$$

```
>\$showev('limit((f(x+h)-f(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{\frac{3(x+h)^3 - 2(x+h)^2 + x+h+5}{x+h+2} - \frac{3x^3 - 2x^2 + x + 5}{x+2}}{h} = \frac{6x^3 + 16x^2 - 8x - 3}{x^2 + 4x + 4}$$

```
>plot2d(["f(x)", "df(x)"], 0, 2*pi, color=[red, green]):
```



```
>function f(x) &= (2*x+1)/(4-x)
```

$$\frac{2x + 1}{4 - x}$$

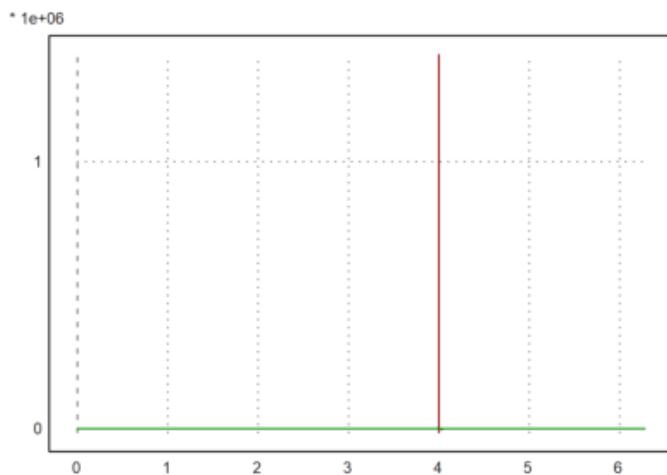
```
>\$showev('limit(f(x),x,0))
```

$$\lim_{x \rightarrow 0} \frac{2x + 1}{4 - x} = \frac{1}{4}$$

```
>$showev('limit((f(x+h)-f(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{\frac{2(x+h)+1}{-x-h+4} - \frac{2x+1}{4-x}}{h} = \frac{9}{x^2 - 8x + 16}$$

```
>plot2d(["f(x)", "df(x")], 0, 2*pi, color=[red, green]):
```



Integral

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).

Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

$$\int_a^b f(x) dx = F(b) - F(a), \quad \text{dengan } F'(x) = f(x).$$

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak integrabel, kita tidak akan menggunakan metode ini).

```
>$showev('integrate(x^n, x))
```

Answering "Is n equal to -1?" with "no"

$$\int x^n dx = \frac{x^{n+1}}{n+1}$$

```
>$showev('integrate(1/(1+x),x))
```

$$\int \frac{1}{x+1} dx = \log(x+1)$$

```
>$showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2+1} dx = \arctan x$$

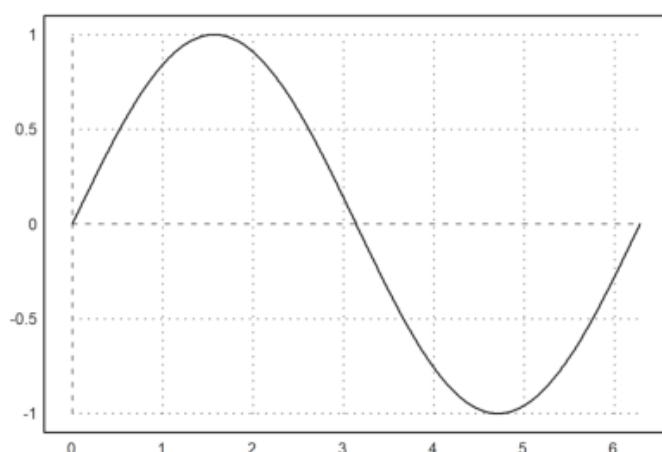
```
>$showev('integrate(1/sqrt(1-x^2),x))
```

$$\int \frac{1}{\sqrt{1-x^2}} dx = \arcsin x$$

```
>$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))
```

Answering "Is n positive, negative or zero?" with "positive"

$$\int_a^b x^n \, dx = \frac{b^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
```

$$\int x^2 \sqrt{2x+1} \, dx = \frac{(2x+1)^{\frac{7}{2}}}{28} - \frac{(2x+1)^{\frac{5}{2}}}{10} + \frac{(2x+1)^{\frac{3}{2}}}{12}$$

```
>$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.

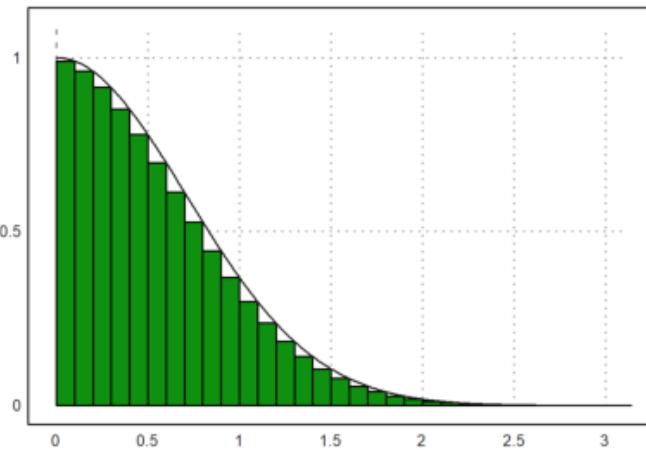
$$\operatorname{erf}(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}} dx.$$

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:

maxima: 'integrate(f(x),x,0,pi)

```
>x=0:0.1:pi-0.1; plot2d(x, f(x+0.1), >bar); plot2d("f(x)", 0, pi, >add) :
```



Integral tentu

maxima: 'integrate(f(x),x,0,pi)

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:

maxima: 'integrate(f(x),x,0,pi) = 0.1*sum(fx[i],i,1,length(fx))

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!

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

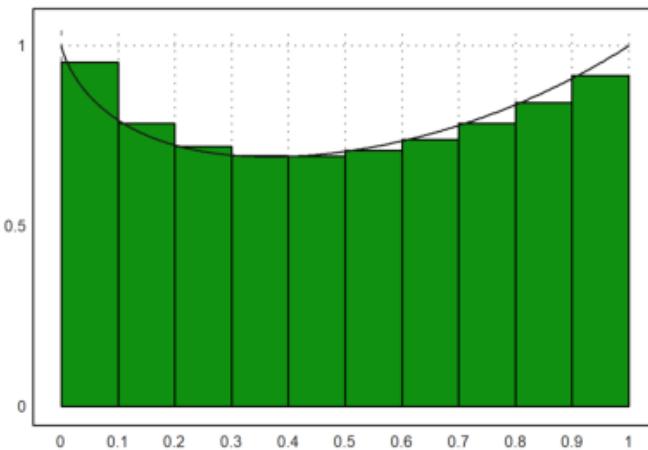
```
>function f(x) &= x^x
```

```
x  
x
```

```
>$showev('integrate(f(x),x,0,1))
```

$$\int_0^1 x^x dx = \int_0^1 x^x dx$$

```
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```



Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
```

maxima: 'integrate(f(x),x,0,1) = 0.01*sum(fx[i],i,1,length(fx))

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
>function f(x) &= sin(3*x^5+7)^2
```

$$\sin^2(3x^5 + 7)$$

```
>integrate(f,0,1)
```

0.542581176074

```
>&showev('integrate(f(x),x,0,1))
```

```

1                               1           pi
/                               gamma(-) sin(14) sin(--)
[   2      5                   5           10
I   sin (3 x  + 7) dx = -----
]                               1/5
/
0
4/5           1           4/5           1
- ((6 gamma_incomplete(-, 6 I) + 6 gamma_incomplete(-, - 6 I))
5                           5
4/5           1
sin(14) + (6 I gamma_incomplete(-, 6 I)
5
4/5           1           pi
- 6 I gamma_incomplete(-, - 6 I)) cos(14)) sin(--)- 60)/120
5                           10

```

```
>&float(%)
```

```

1.0
/
[      2      5
I sin (3.0 x  + 7.0) dx =
]
/
0.0
0.09820784258795788 - 0.00833333333333333
(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)
```

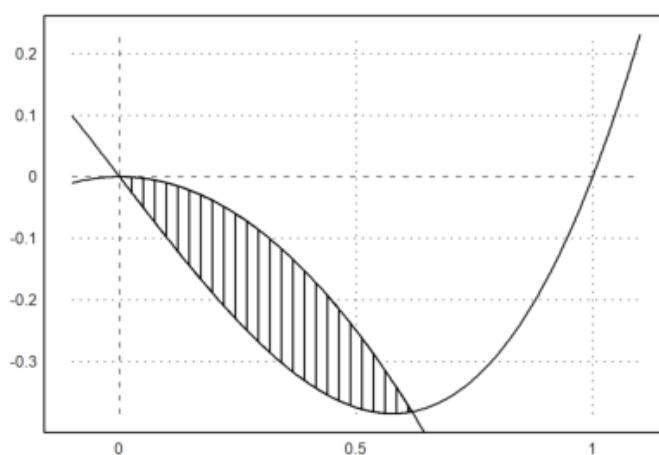
```
>$showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)
```

$$\int_0^1 x e^{-x} dx = 1 - 2e^{-1}$$

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 daera
```



```
>a=solve("x^3-x+x^2", 0), b=solve("x^3-x+x^2", 1) // absis titik-titik potong
```

0
0.61803398875

```
>integrate("(-x^2)-(x^3-x)", a, b) // luas daerah yang diarsir
```

0.0758191713542

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```
>a &= solve( (-x^2)-(x^3-x), x); $a // menentukan absis titik potong kedua ku
```

$$\left[x = \frac{-\sqrt{5}-1}{2}, x = \frac{\sqrt{5}-1}{2}, x = 0 \right]$$

```
>$showev('integrate(-x^2-x^3+x, x, 0, (sqrt(5)-1)/2)) // Nilai integral secara
```

$$\int_0^{\frac{\sqrt{5}-1}{2}} -x^3 - x^2 + x \, dx = \frac{13 - 5^{\frac{3}{2}}}{24}$$

```
>$float (%)
```

$$\int_{0.0}^{0.6180339887498949} -1.0 x^3 - 1.0 x^2 + x \, dx = 0.07581917135421037$$

Panjang Kurva

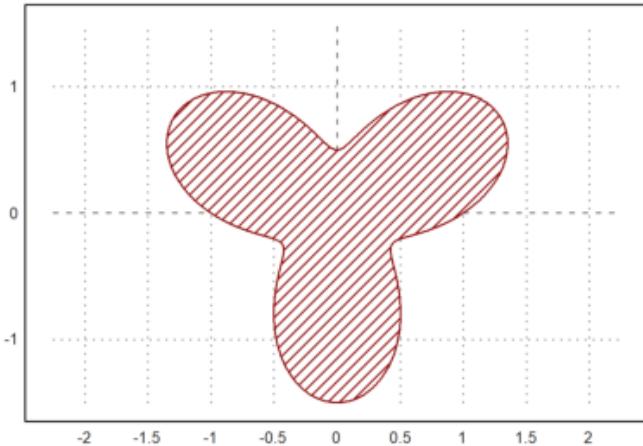
Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

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

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}, \quad 0 \leq t \leq 2\pi.$$

```
>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
```



```
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
```

$r([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21])$

```
>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
```

$fx([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21])$

```
>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
```

$fy([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21])$

```
>function ds(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2)))
```

Maxima said:

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

Error in:

```
... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $' ds(t)=ds(t ...  
^
```

```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
```

$$\int_0^{2\pi} ds(x) \, dx$$

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara umerik dengan perintah EMT.

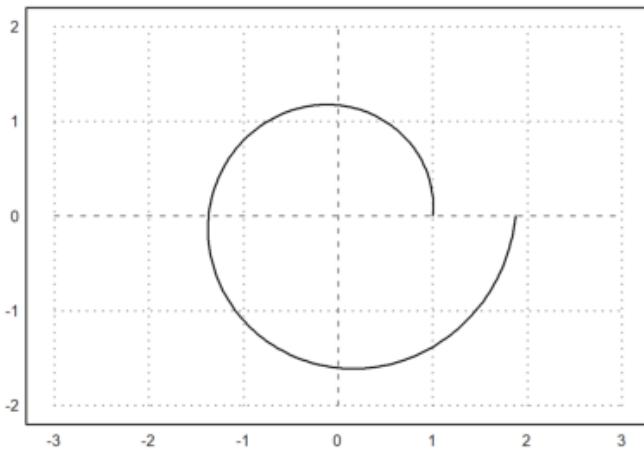
```
>integrate("ds(x)",0,2*pi)
```

```
Function ds not found.  
Try list ... to find functions!  
Error in expression: ds(x)  
%mapexpression1:  
    return expr(x,args());  
Error in map.  
%evalexpression:  
    if maps then return %mapexpression1(x,f$;args());  
gauss:  
    if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());  
adaptivegauss:  
    t1=gauss(f$,c,c+h;args(),=maps);  
Try "trace errors" to inspect local variables after errors.  
integrate:  
    return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
```

Spiral Logaritmik

$$x = e^{ax} \cos x, \quad y = e^{ax} \sin x.$$

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi):
```



```
>&kill(a) // hapus expresi a
```

done

```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
```

$fx([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2,$

```
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
```

$fy([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2,$

```
>function df(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2)))
```

Maxima said:

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

Error in:

```
... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t ...  
^
```

```
>S &=integrate(df(t),t,0,2*pi); $S // panjang kurva (spiral)
```

Maxima said:

```
defint: variable of integration cannot be a constant; found errexp1
-- an error. To debug this try: debugmode(true);
```

Error in:

```
S &=integrate(df(t),t,0,2*pi); $S // panjang kurva (spiral) ...  
^
```

```
>S(a=0.1) // Panjang kurva untuk a=0.1
```

Function S not found.

Try list ... to find functions!

Error in:

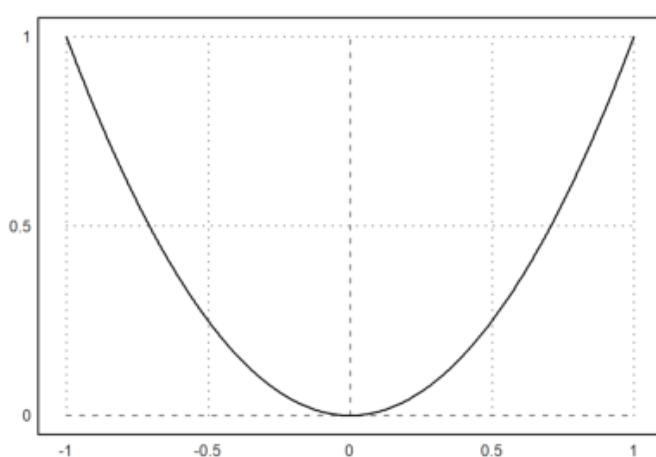
```
S(a=0.1) // Panjang kurva untuk a=0.1 ...  
^
```

Soal:

Tunjukkan bahwa keliling lingkaran dengan jari-jari r adalah $K=2\pi r$.

Berikut adalah contoh menghitung panjang parabola.

```
>plot2d("x^2", xmin=-1, xmax=1):
```



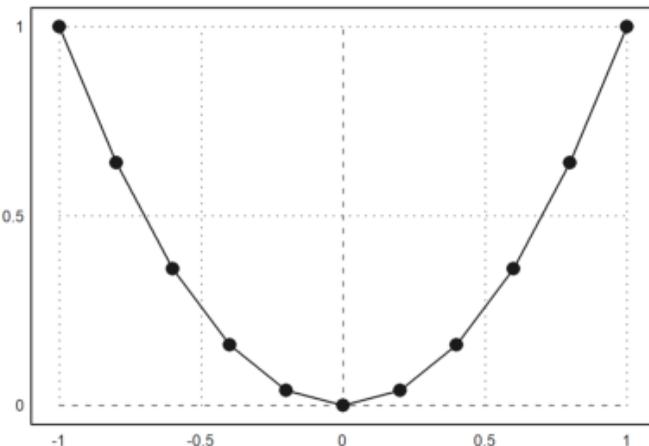
```
>$showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))
```

$$\int_{-1}^1 \sqrt{4x^2 + 1} dx = \frac{\operatorname{asinh} 2 + 2\sqrt{5}}{2}$$

```
> $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.

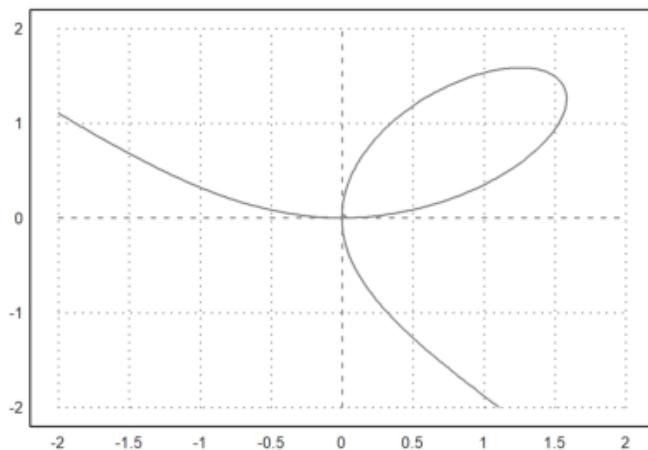
Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

$$x^3 + y^3 - 3xy = 0.$$

```
>z &= x^3+y^3-3*x*y; $z
```

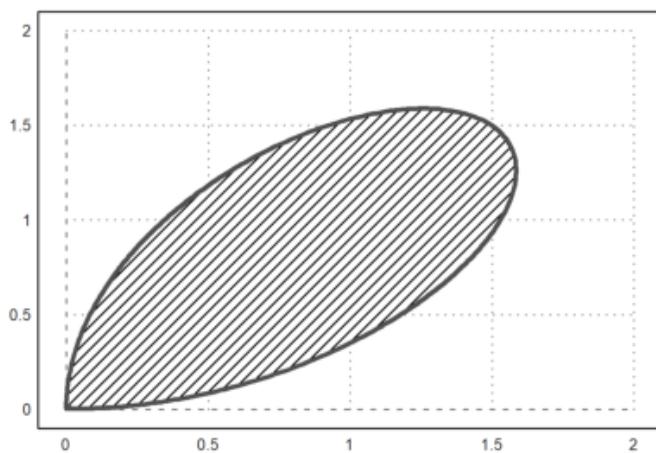
$$y^3 - 3xy + x^3$$

```
>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
```

$$\left[x = \frac{3l}{l^3 + 1}, x = 0 \right]$$

$$\left[x = \frac{3l}{l^3 + 1}, x = 0 \right]$$

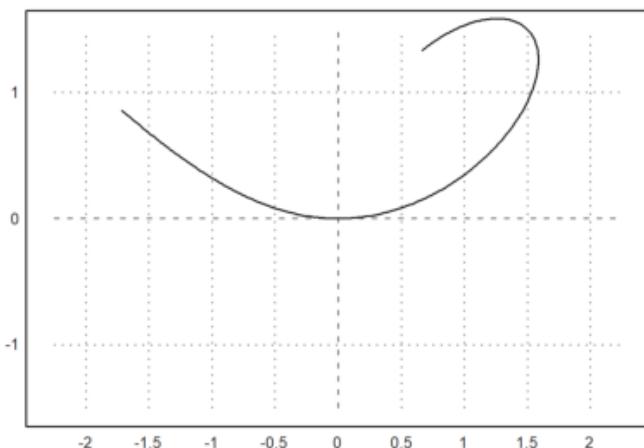
Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); $' f(l)=f(l)
```

$$f(l) = \frac{3l}{l^3 + 1}$$

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)=
```

$$ds(l) = \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}}$$

```
>$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
```

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 parabol
```

2.95788571509

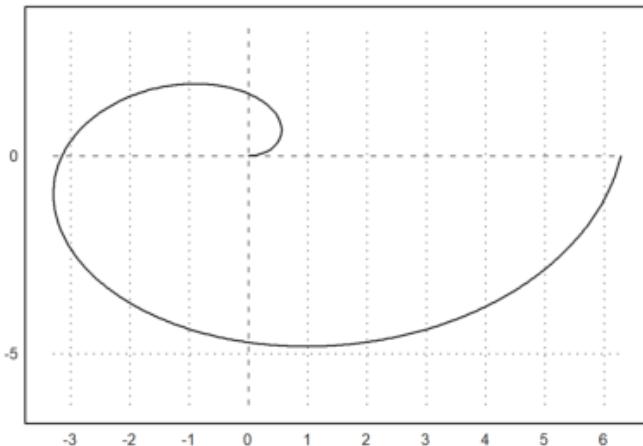
Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)",mxm("x*f(x)",0,1)) // cek contoh terakhir, band
```

4.91748872168

Kita hitung panjang spiral 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)
```

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

```
>sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk menghitung panjan
```

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

```
>$sol | trigreduce | expand, $integrate(% ,x,0,2*pi), %()
```

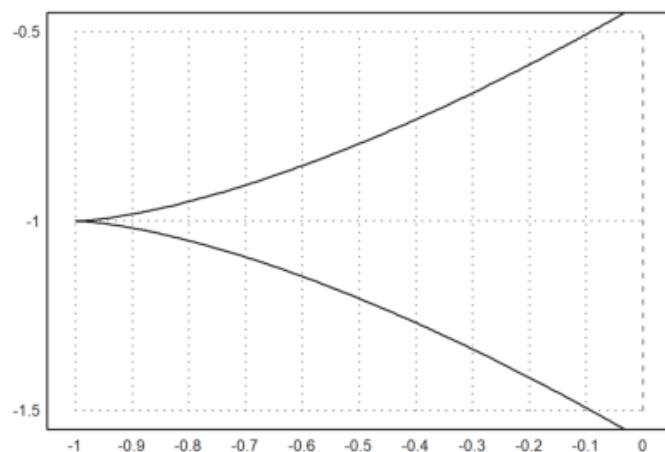
$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$

$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$

21.2562941482

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.
Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
>plot2d("3*x^2-1", "3*x^3-1", xmin=-1/sqrt(3), xmax=1/sqrt(3), square=1):
```



```
>sol &= radcan(ds(3*x^2-1, 3*x^3-1)); $sol
```

$$3x\sqrt{9x^2 + 4}$$

```
>$showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%); // panjang kurva di at
```

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0x^2 + 4.0} dx = 2.337835372767141$$

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0x^2 + 4.0} dx = 2.337835372767141$$

Sikloid

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar ke kanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah r . Posisi titik pusat lingkaran pada saat t adalah:

$$(rt, r).$$

Misalkan posisi titik pada lingkaran tersebut mula-mula $(0,0)$ dan posisinya pada saat t adalah:

$$(r(t - \sin(t)), r(1 - \cos(t))).$$

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika $t=0, t=\pi/2, t=r*\pi$.

```
>x &= r*(t-sin(t))
```

```
[0, 1.66665833335744e-7 r, 1.33330666692022e-6 r,
4.499797504338432e-6 r, 1.066581336583994e-5 r,
2.083072932167196e-5 r, 3.599352055540239e-5 r,
5.71526624672386e-5 r, 8.530603082730626e-5 r,
1.214508019889565e-4 r, 1.665833531718508e-4 r,
2.216991628251896e-4 r, 2.877927110806339e-4 r,
3.658573803051457e-4 r, 4.568853557635201e-4 r,
5.618675264007778e-4 r, 6.817933857540259e-4 r,
8.176509330039827e-4 r, 9.704265741758145e-4 r,
```

0.001141105023499428 r, 0.001330669204938795 r,
 0.001540100153900437 r, 0.001770376919130678 r,
 0.002022476464811601 r, 0.002297373572865413 r,
 0.002596040745477063 r, 0.002919448107844891 r,
 0.00326856331168871 r, 0.003644351435886262 r,
 0.004047774895164447 r, 0.004479793338660443 r, 0.0049413635565565 r,
 0.005433439383882244 r, 0.005956971605131645 r,
 0.006512907859185624 r, 0.007102192544548636 r,
 0.007725766724910044 r, 0.00838456803503801 r,
 0.009079530587017326 r, 0.009811584876838586 r, 0.0105816576913495 r,
 0.01139067201557714 r, 0.01223954694042984 r, 0.01312919757078923 r,
 0.01406053493400045 r, 0.01503446588876983 r, 0.01605189303448024 r,
 0.01711371462093175 r, 0.01822082445851714 r, 0.01937411182884202 r,
 0.02057446139579705 r, 0.02182275311709253 r, 0.02311986215626333 r,
 0.02446665879515308 r, 0.02586400834688696 r, 0.02731277106934082 r,
 0.02881380207911666 r, 0.03036795126603076 r, 0.03197606320812652 r,
 0.0336389770872163 r, 0.03535752660496472 r, 0.03713253989951881 r,
 0.03896483946269502 r, 0.0408552420577305 r, 0.04280455863760801 r,
 0.04481359426396048 r, 0.04688314802656623 r, 0.04901401296344043 r,
 0.05120697598153157 r, 0.05346281777803219 r, 0.05578231276230905 r,
 0.05816622897846346 r, 0.06061532802852698 r, 0.0631303649963022 r,
 0.06571208837185505 r, 0.06836123997666599 r, 0.07107855488944881 r,
 0.07386476137264342 r, 0.07672058079958999 r, 0.07964672758239233 r,
 0.08264390910047736 r, 0.0857128256298576 r, 0.08885417027310427 r,
 0.09206862889003742 r, 0.09535688002914089 r, 0.0987195948597075 r,
 0.1021574371047232 r, 0.1056710629744951 r, 0.1092611211010309 r,
 0.1129282524731764 r, 0.1166730903725168 r, 0.1204962603100498 r,
 0.1243983799636342 r, 0.1283800591162231 r, 0.1324418995948859 r,
 0.1365844952106265 r, 0.140808431699002 r, 0.1451142866615502 r,
 0.1495026295080298 r, 0.1539740213994798 r]

```
>y &= r*(1-cos(t))
```

[0, 4.999958333473664e-5 r, 1.999933334222437e-4 r,
 4.499662510124569e-4 r, 7.998933390220841e-4 r,
 0.001249739605033717 r, 0.00179946006479581 r,
 0.002448999746720415 r, 0.003198293697380561 r,
 0.00404726698005727 r, 0.004995834721974179 r,
 0.006043902043303184 r, 0.00719136414613375 r, 0.00843810628521191 r,
 0.009784003787362772 r, 0.01122892206395776 r, 0.01277271662437307 r,
 0.01441523309043924 r, 0.01615630721187855 r, 0.01799576488272969 r,
 0.01993342215875837 r, 0.02196908527585173 r, 0.02410255066939448 r,
 0.02633360499462523 r, 0.02866202514797045 r, 0.03108757828935527 r]

0.03361002186548678 r, 0.03622910363410947 r, 0.03894456168922911 r,
 0.04175612448730281 r, 0.04466351087439402 r, 0.04766643011428662 r,
 0.05076458191755917 r, 0.0539576564716131 r, 0.05724533447165381 r,
 0.06062728715262111 r, 0.06410317632206519 r, 0.06767265439396564 r,
 0.07133536442348987 r, 0.07509094014268702 r, 0.07893900599711501 r,
 0.08287917718339499 r, 0.08691105968769186 r, 0.09103425032511492 r,
 0.09524833678003664 r, 0.09955289764732322 r, 0.1039475024744748 r,
 0.1084317118046711 r, 0.113005077220716 r, 0.1176671413898787 r,
 0.1224174381096274 r, 0.1272554923542488 r, 0.1321808203223502 r,
 0.1371929294852391 r, 0.1422913186361759 r, 0.1474754779404944 r,
 0.152744888986584 r, 0.1580990248377314 r, 0.1635373500848132 r,
 0.1690593208998367 r, 0.1746643850903219 r, 0.1803519821545206 r,
 0.1861215433374662 r, 0.1919724916878484 r, 0.1979042421157076 r,
 0.2039162014509444 r, 0.2100077685026351 r, 0.216178334119151 r,
 0.2224272812490723 r, 0.2287539850028937 r, 0.2351578127155118 r,
 0.2416381240094921 r, 0.2481942708591053 r, 0.2548255976551299 r,
 0.2615314412704124 r, 0.2683111311261794 r, 0.2751639892590951 r,
 0.2820893303890569 r, 0.2890864619877229 r, 0.2961546843477643 r,
 0.3032932906528349 r, 0.3105015670482534 r, 0.3177787927123868 r,
 0.3251242399287333 r, 0.3325371741586922 r, 0.3400168541150183 r,
 0.3475625318359485 r, 0.3551734527599992 r, 0.3628488558014202 r,
 0.3705879734263036 r, 0.3783900317293359 r, 0.3862542505111889 r,
 0.3941798433565377 r, 0.4021660177127022 r, 0.4102119749689023 r,
 0.418316910536117 r, 0.4264800139275439 r, 0.4347004688396462 r,
 0.4429774532337832 r, 0.451310139418413 r]

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):
```

Error : [0,1.6666583335744e-7*r-sin(1.6666583335744e-7*r),1.33330666692

```
dw/n, dw/n^2, dw/n; args());
```

Berikut dihitung panjang lintasan untuk 1 putaran penuh. (Jangan salah menduga bahwa panjang lintasan 1 putaran penuh sama dengan keliling lingkaran!)

```
>ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds) // elemen
```

Maxima said:

```
diff: second argument must be a variable; found errexpl
-- an error. To debug this try: debugmode(true);
```

Error in:

```
ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds ...
^
```

```
>ds &= trigsimp(ds); $ds
```

```
>$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu putaran pe
```

Maxima said:

```
defint: variable of integration must be a simple or subscripted variable.
```

```
defint: found errexpl
```

```
#0: showev(f='integrate(ds,[0,1.6666583335744e-7*r,1.33330666692022e-6*r
-- an error. To debug this try: debugmode(true);
```

Error in:

```
$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid sat ...
^
```

```
>integrate(mxm("ds"),0,2*pi) // hitung secara numerik
```

Illegal function result in map.

```
%evalexpression:
```

```
if maps then return %mapexpression1(x,f$;args());
```

```
gauss:
```

```
if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());
```

```
adaptivegauss:
```

```
t1=gauss(f$,c,c+h;args(),=maps);
```

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

```
integrate:
```

```
return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
```

```
>romberg(mxm("ds"), 0, 2*pi) // cara lain hitung secara numerik
```

Wrong argument!

Cannot combine a symbolic expression here.
Did you want to create a symbolic expression?
Then start with &.

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

```
if cols(y)==1 then return y*(b-a); endif;  
Error in:  
romberg(mxm("ds"), 0, 2*pi) // cara lain hitung secara numerik ...  
^
```

Kurvatur (Kelengkungan) Kurva

image: Osculating.png

Aslinya, kelengkungan kurva diferensiabel (yakni, kurva mulus yang tidak lancip) di titik P didefinisikan melalui lingkaran oskulasi (yaitu, lingkaran yang melalui titik P dan terbaik memperkirakan, paling banyak menyinggung kurva di sekitar P). Pusat dan radius kelengkungan kurva di P adalah pusat dan radius lingkaran oskulasi. Kelengkungan adalah kebalikan dari radius kelengkungan:

$$\kappa = \frac{1}{R}$$

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar 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

$$x = r \cos t, y = r \sin t.$$

```
>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
```

Maxima said:

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

Error in:

```
... =integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s // elemen ... ^
```

```
>&kill(s); fx &= r*cos(s/r); fy &=r*sin(s/r); // definisi ulang persamaan p
>k &= trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur l
```

$$\frac{1}{r}$$

Untuk representasi parametrik umum, misalkan

$$x = x(t), \quad y = y(t)$$

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)
```

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) //
```

Maxima said:

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

Error in:

```
... (x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $' k(x)=k(x) ...  
^
```

```
>function f(x) &= x^2+x+1; $y=f(x) // akan kita plot kelengkungan parabola
```

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $' k(x)=k(x) //
```

Maxima said:

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

Error in:

```
... (x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $' k(x)=k(x) ...  
^
```

Berikut kita gambar parabola tersebut beserta kurva kelengkungan, kurva jari-jari kelengkungan dan salah satu lingkaran oskulasi di titik puncak parabola. Perhatikan, puncak parabola dan jari-jari lingkaran oskulasi di puncak parabola adalah

$$(-1/2, 3/4), 1/k(2) = 1/2,$$

sehingga pusat lingkaran oskulasi adalah $(-1/2, 5/4)$.

```
>plot2d(["f(x)", "k(x)"], -2, 1, color=[blue, red]); plot2d("1/k(x)", -1.5, 1, co
>plot2d("-1/2+1/k(-1/2)*cos(x)", "5/4+1/k(-1/2)*sin(x)", xmin=0, xmax=2pi, >add
```

Error : $f(x)$ does not produce a real or column vector

Error generated by error() command

```
%ploteval:
error(f$|" does not produce a real or column vector");
adaptiveevalone:
s=%ploteval(g$,t;args());
```

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

plot2d:

```
dw/n, dw/n^2, dw/n, auto; args());
```

Untuk kurva yang dinyatakan dengan fungsi implisit

$$F(x, y) = 0$$

dengan turunan-turunan parsial

$$F_x = \frac{\partial F}{\partial x}, \quad F_y = \frac{\partial F}{\partial y}, \quad F_{xy} = \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial x} \right), \quad F_{xx} = \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial x} \right), \quad F_{yy} = \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial y} \right),$$

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)
```

```
>Fx &= diff(F(x, y), x), Fxx &=diff(F(x, y), x, 2), Fy &=diff(F(x, y), y), Fxy &=d
```

Maxima said:

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

Error in:

```
Fx &= diff(F(x,y),x), Fxx &=diff(F(x,y),x,2), Fy &=diff(F(x,y)) . . .
^
```

```
>function k(x) &= (Fy^2*Fxx-2*Fx*Fy*Fxy+Fx^2*Fyy)/(Fx^2+Fy^2)^(3/2); $'k(x)
```

$$k([0, 1.66665833335744 \times 10^{-7} r, 1.33330666692022 \times 10^{-6} r, 4.499797504338432 \times 10^{-6} r, 1.066581336583])$$

Hasilnya sama dengan sebelumnya yang menggunakan persamaan parabola biasa. **Lati-han**

- 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 oskulasinya.

```
>$showev('integrate(2*x^2-3,x,3,9))
```

Maxima said:

```
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([-3,5.5555000002996e-14*r^2-3,3.555413336107812e-
-- an error. To debug this try: debugmode(true);
```

Error in:

```
$showev('integrate(2*x^2-3,x,3,9)) ...  
^
```

```
>$showev('integrate(5*x^3-2*x^2-x+3,x,4,8))
```

Maxima said:

```
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([3,2.314780092866648e-20*r^3-5.5555000002996e-14*-
-- an error. To debug this try: debugmode(true);
```

Error in:

```
$showev('integrate(5*x^3-2*x^2-x+3,x,4,8)) ...  
^
```

```
>$showev('integrate(x^4+2*x^2+x-2,x,4,8))
```

Maxima said:

```
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([-2,7.715895063332219e-28*r^4+5.5555000002996e-14
-- an error. To debug this try: debugmode(true);
```

Error in:

```
$showev('integrate(x^4+2*x^2+x-2,x,4,8)) ...
^
```

```
>$showev('integrate(4*x^3-6*x^2-x-1,x,5,10))
```

Maxima said:

```
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([-1,1.851824074293318e-20*r^3-1.66665000008988e-14
-- an error. To debug this try: debugmode(true);
```

Error in:

```
$showev('integrate(4*x^3-6*x^2-x-1,x,5,10)) ...
^
```

```
>$showev('integrate(2*sin(x)-cos(x-2),x,3*pi))
```

Maxima said:

```
integrate: wrong number of arguments.
-- an error. To debug this try: debugmode(true);
```

Error in:

```
$showev('integrate(2*sin(x)-cos(x-2),x,3*pi)) ...
^
```

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 beraturan (menggunakan titik dua ":");
- menggunakan perintah "sequence" dan rumus barisan (suku ke -n);

- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create_list" atau "makelist" untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana
```

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

```
>1:2:30
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
```

Iterasi dan Barisan

EMT menyediakan fungsi iterate("g(x)", x0, n) untuk melakukan iterasi

$$x_{k+1} = g(x_k), \quad x_0 = x_0, \quad k = 1, 2, 3, \dots, n.$$

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.

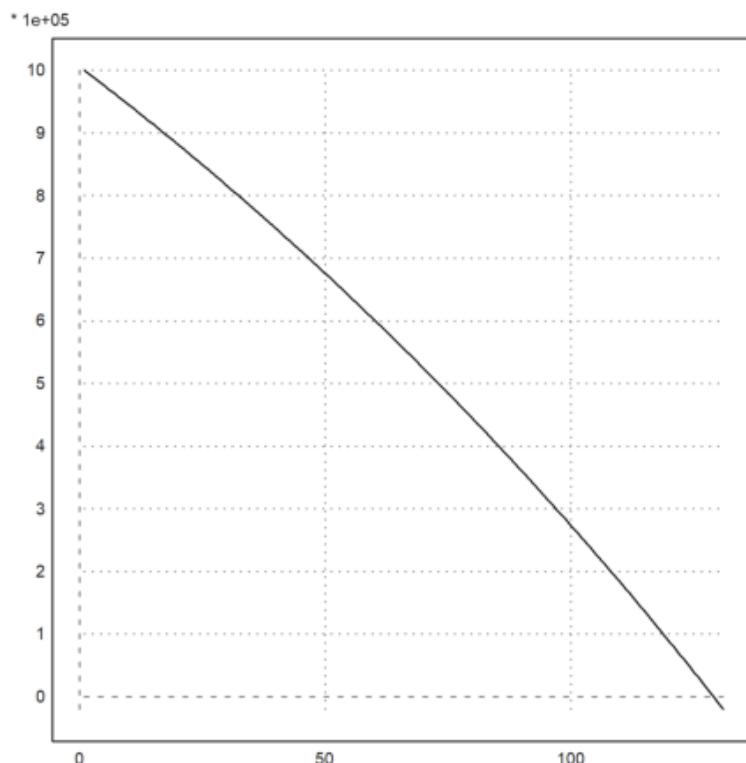
```
>q=1.05; iterate("x*q", 1000, n=10)'
```

```
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]
```

$$[A = 2500.0]$$

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)
```

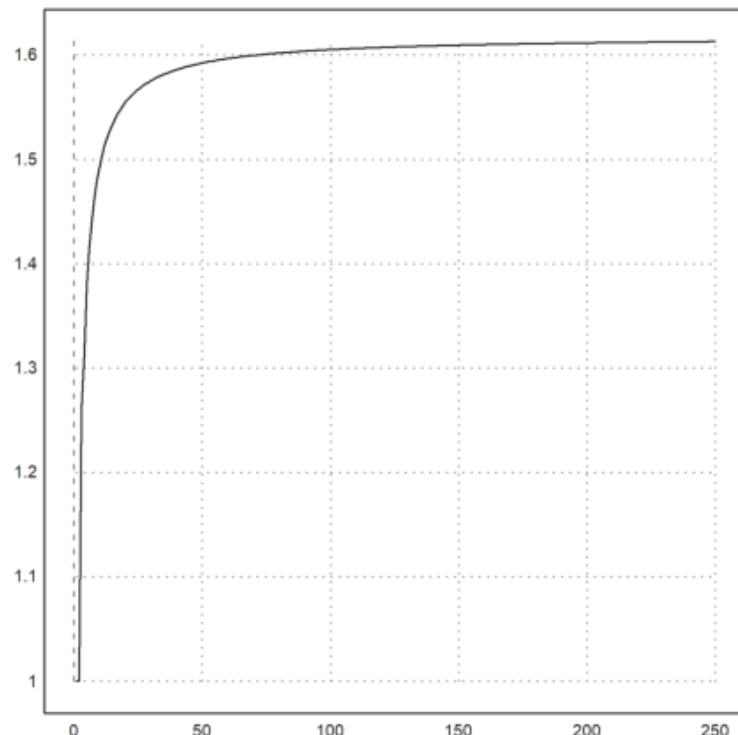
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]

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]

```
>A=[1,1;1,2]; function suku(x,n) := A.x[,n-1]
>sequence("suku", [1;1], 6)
```

Real 2 x 6 matrix

1	2	5	13	...
1	3	8	21	...

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "matrixpower()". Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak $\log_2(n)$.

$$x_n = A \cdot x_{n-1} = A^2 \cdot x_{n-2} = A^3 \cdot x_{n-3} = \dots = A^{n-1} \cdot x_1.$$

```
>sequence("matrixpower(A,n).[1;1]", 1, 6)
```

Real 2 x 6 matrix

1	5	13	34	...
1	8	21	55	...

Spiral Theodorus

image: Spiral_of_Theodorus.png

Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

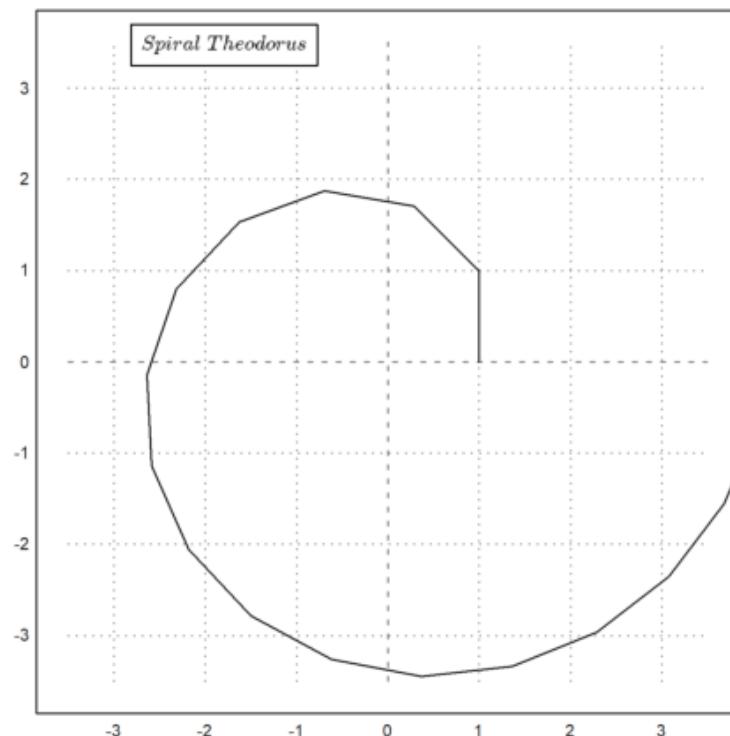
$$x_n = \left(1 + \frac{i}{\sqrt{n-1}}\right) x_{n-1}, \quad x_1 = 1,$$

yang menghasilkan barisan bilangan kompleks.

```
>function g(n) := 1+I/sqrt(n)
```

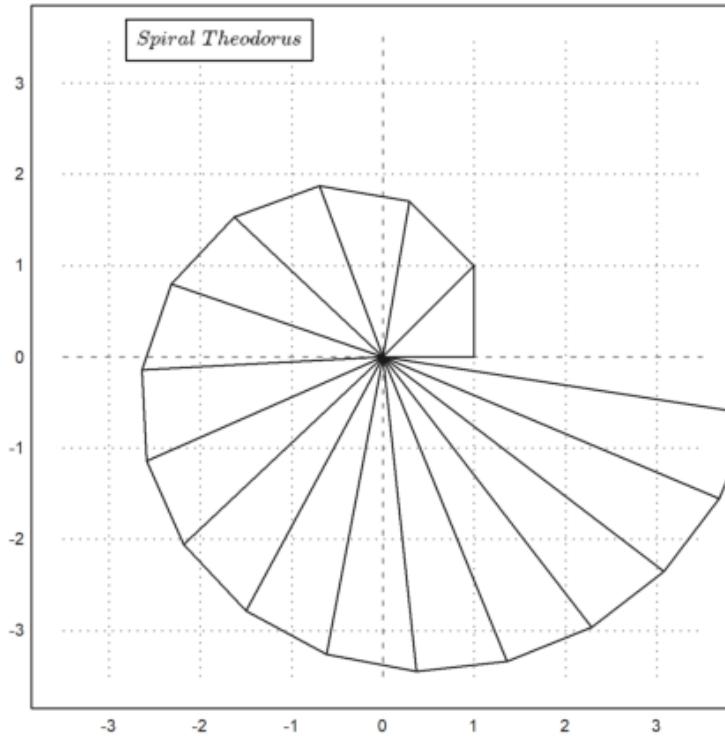
Rekursinya dapat dijalankan sebanyak 17 untuk menghasilkan barisan 17 bilangan kompleks, kemudian digambar bilangan-bilangan kompleksnya.

```
>x=sequence ("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox(latex("Spiral\\"
```



Selanjutnya dihubungkan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end:
```



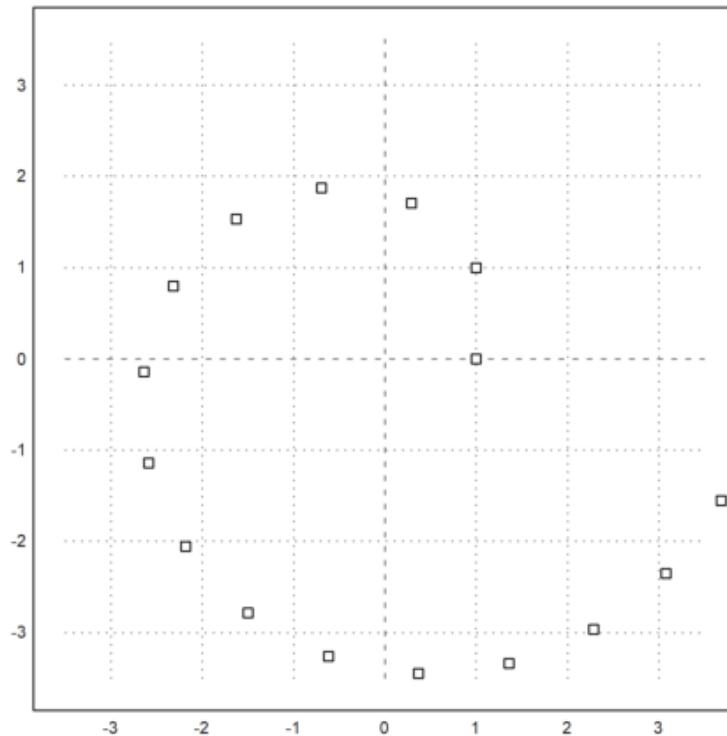
Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memerlukan indeks dan bilangan kompleks. Dalam hal ini diigunakan vektor kolom pada bidang.

```
>function gstep (v) ...
```

```
w=[-v[2];v[1]];
return v+w/norm(w);
endfunction
```

Jika dilakukan iterasi 16 kali dimulai dari [1;0] akan didapatkan matriks yang memuat vektor-vektor dari setiap iterasi.

```
>x=iterate("gstep", [1;0],16); plot2d(x[1],x[2],r=3.5,>points):
```



Kekonvergenan

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```
>iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1.
```

0.739085133216

Iterasi tersebut konvergen ke penyelesaian persamaan

$$x = \cos(x).$$

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.

```
>hasil := iterate("cos(x)",~1,2~) //iterasi x(n+1)=cos(x(n)), dengan interval
```

~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
```

$\sim 0.73908513309, 0.73908513333\sim$

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

$\sim 0.99999999999999778, 1.0000000000000022 \sim \dots$
 $\sim 4.999999999999911, 5.0000000000000089 \sim \dots$

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

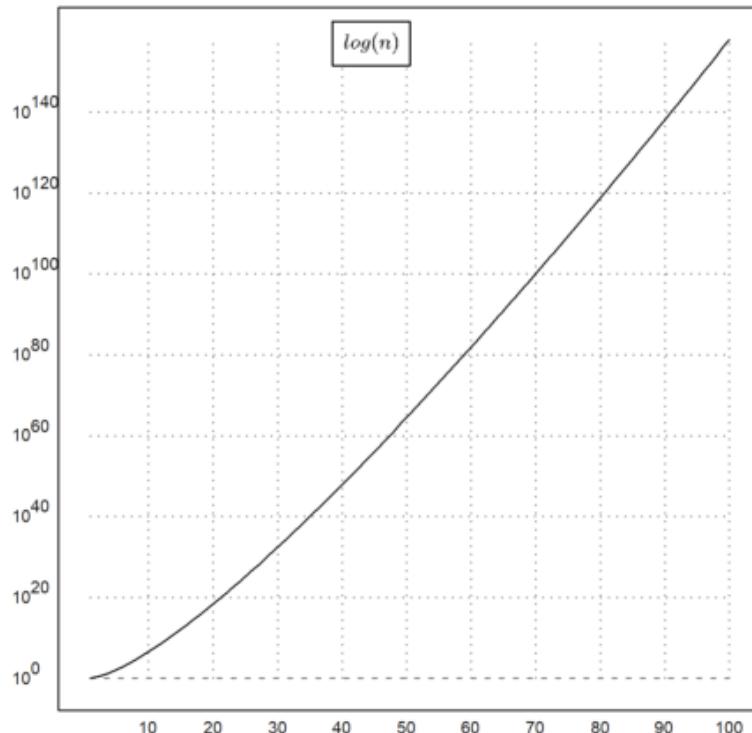
Penggabungan matriks menggunakan tanda " $|$ " dapat digunakan untuk menyimpan semua hasil iterasi.

```
>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)
```

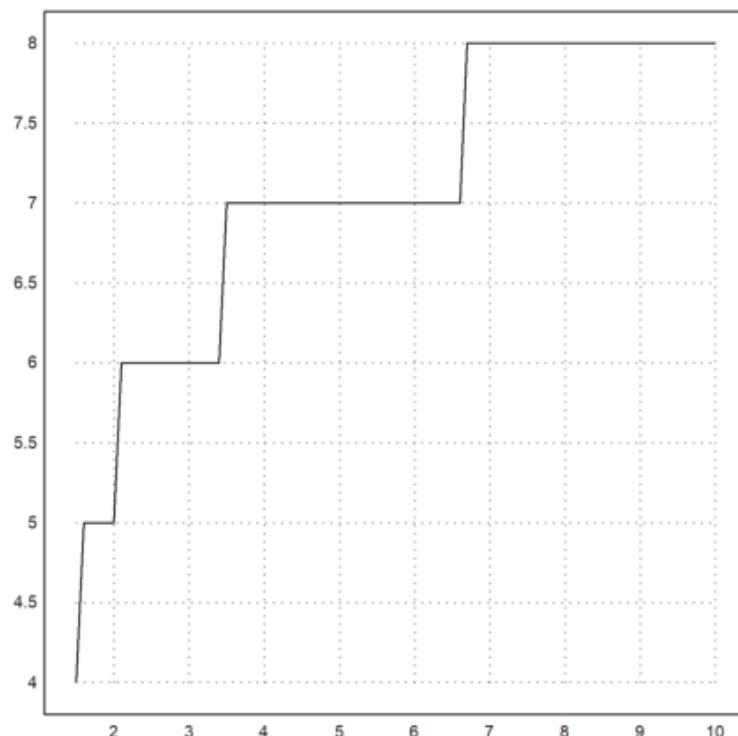
5

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]
```

```
>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
```

Maxima said:

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

Error in:

```
p &= x^3-1; newton &= x-p/diff(p,x); $newton ...  
^
```

Selanjutnya didefinisikan fungsi untuk melakukan iterasi (aslanya 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);
```

Function newton needs at least 3 arguments!

Use: newton (f\$: call, df\$: call, x: scalar complex {, y: number, eps: no

Error in:

```
... x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z); ...  
^
```

Berikut adalah akar-akar polinomial di atas.

```
>z=&solve(p) ()
```

```
Maxima said:  
solve: more equations than unknowns.  
Unknowns given :  
[r]  
Equations given:  
errexpl  
-- an error. To debug this try: debugmode(true);  
  
Error in:  
z=&solve(p) () ...  
^
```

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):
```

```
Variable W not found!  
Error in:  
C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1)) ...  
^
```

Iterasi Simbolik

Seperti sudah dibahas sebelumnya, untuk menghasilkan barisan ekspresi simbolik dengan Maxima dapat digunakan fungsi makelist().

```
>&powerdisp:true // untuk menampilkan deret pangkat mulai dari suku berpang
```

true

```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Tay
```

Maxima said:

```
taylor: 0.1539740213994798*r cannot be a variable.  
-- an error. To debug this try: debugmode(true);
```

Error in:

```
deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // baris ...  
^
```

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 hampi
```

Maxima said:

```
length: argument cannot be a symbol; found deret  
-- an error. To debug this try: debugmode(true);
```

mxmeval:

```
    return evaluate(mxm(s));
```

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

mxm2str:

```
n=mxmeval("length(VVV)");
```

Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

```
>$deret[3]
```

*deret*₃

```
>plot2d(["exp(x)",&deret[1],&deret[2],&deret[3]],0,3,color=1:4):
```

deret is not a variable!

Error in expression: deret[1]

%plotheval:

```
y0=f$(x[1],args());
```

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

plot2d:

```
u=u_(%ploteval(xx[#,t;args()));
```

```
>$sum(sin(k*x)/k,k,1,5)
```



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):
```

Maxima output too long!

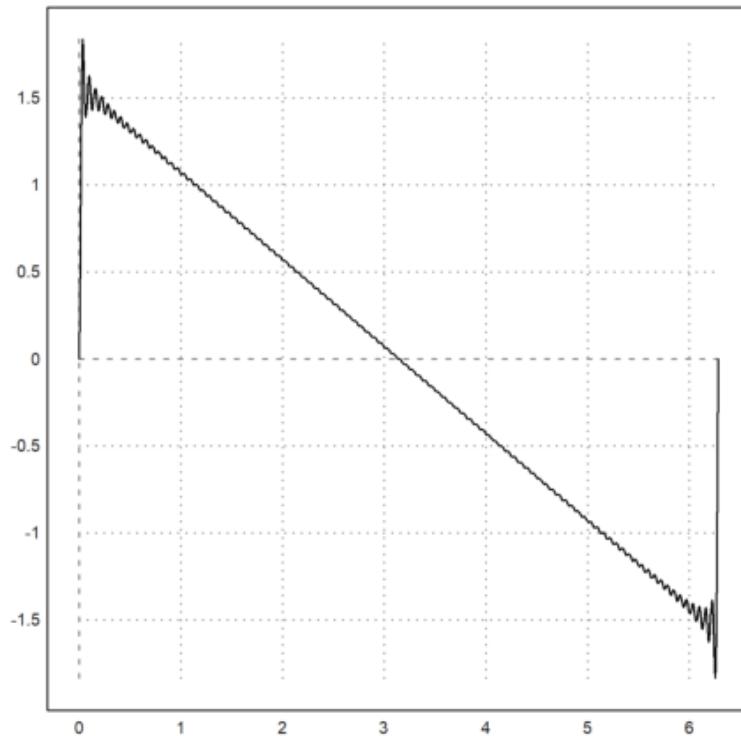
Error in:

```
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);
```

Maxima said:

```
diff: second argument must be a variable; found errexp1
#0: diffat(expr=[0,1.66665833335744e-7*r,1.33330666692022e-6*r,4.49979750
-- an error. To debug this try: debugmode(true);
```

%mxmevtable:

```
    return mxm("@expr,@var=@value")();
```

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

mxmtable:

```
y[#,1]=%mxmevtable(expr,var,x[#]);
```

```
>$' sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:me
```

$$\sum_{k=1}^n k = \frac{n(1+n)}{2}$$

```
> $' sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf), simpsum=
```

$$\sum_{k=0}^{\infty} \frac{1}{k+3^k} = \sum_{k=0}^{\infty} \frac{1}{k+3^k}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
> $' sum(1/x^2, x, 1, inf) = ev(sum(1/x^2, x, 1, inf), simpsum=true) // ev: men
```

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

```
> $' sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf), si
```

$$\sum_{k=1}^{\infty} \frac{(-1)^{-1+k}}{k} = - \sum_{x=1}^{\infty} \frac{(-1)^x}{x}$$

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)
```

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{-1+2k} = \sum_{k=1}^{\infty} \frac{(-1)^k}{-1+2k}$$

```
> $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;
```

Answering "Is $-94914474571+15819\pi r$ positive, negative or zero?" with "positive" Maxima said:

```
sum: sum is divergent.  
-- an error. To debug this try: debugmode(true);
```

```
Error in:  
... k, 0, inf)=ev(sum(a*x^k, k, 0, inf), simpsum=true), &forget(abs ...  
^
```

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)
```

$$\left[0, \sum_{k=0}^{\infty} \frac{(1.66665833335744 \times 10^{-7})^k r^k}{k!}, \sum_{k=0}^{\infty} \frac{(1.33330666692022 \times 10^{-6})^k r^k}{k!}, \sum_{k=0}^{\infty} \frac{(4.499797504338432 \times 10^{-1})^k r^k}{k!} \right]$$

```
>$limit(sum(x^k/k!, k, 0, n), n, inf)
```

$$\left[0, \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{(1.66665833335744 \times 10^{-7})^k r^k}{k!}, \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{(1.33330666692022 \times 10^{-6})^k r^k}{k!}, \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{(4.499797504338432 \times 10^{-1})^k r^k}{k!} \right]$$

```
>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 su
```

Maxima said:

```
taylor: 0.1539740213994798*r cannot be a variable.  
-- an error. To debug this try: debugmode(true);
```

Error in:

```
$' e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x= ...  
^
```

```
> $' log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1
```

Maxima said:

```
log: encountered log(0).  
-- an error. To debug this try: debugmode(true);
```

Error in:

```
$' log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1 ...  
^
```

BAB 6

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
```

```

plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d
plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"
plotLabel (label, P, V, d): menuliskan label pada posisi P

```

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

```

turn(v, phi): memutar vektor v sejauh phi
turnLeft(v): memutar vektor v ke kiri
turnRight(v): memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang vektorv dan w.
lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh.

```

$ax+by=c$.

```

lineWithDirection(A,v): garis melalui A searah vektor v
getLineDirection(g): vektor arah (gradien) garis g
getNormal(g): vektor normal (tegak lurus) garis g
getPointOnLine(g): titik pada garis g
perpendicular(A, g): garis melalui A tegak lurus garis g
parallel (A, g): garis melalui A sejajar garis g
lineIntersection(g, h): titik potong garis g dan h
projectToLine(A, g): proyeksi titik A pada garis g
distance(A, B): jarak titik A dan B
distanceSquared(A, B): kuadrat jarak A dan B
quadrance(A, B): kuadrat jarak A dan B
areaTriangle(A, B, C): luas segitiga ABC
computeAngle(A, B, C): besar sudut <ABC
angleBisector(A, B, C): garis bagi sudut <ABC
circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r
getCircleCenter(c): pusat lingkaran c
getCircleRadius(c): jari-jari lingkaran c
circleThrough(A,B,C): lingkaran melalui A, B, C
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan lingkran c
circleCircleIntersections (c1, c2): titik potong lingkaran c1 dan

```

c2

```
planeThrough(A, B, C): bidang melalui titik A, B, C
```

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

`getLineEquation (g, x, y)`: persamaan garis g dinyatakan dalam x dan y
`getHesseForm (g, x, y, A)`: bentuk Hesse garis g dinyatakan dalam x dan

y 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 membentuk

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, tetapkan tiga titik dan plotkan.

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
>B=[0,1]; plotPoint(B,"B");
>C=[2,2]; plotPoint(C,"C");
```

Kemudian tiga segmen.

```
>plotSegment(A,B,"c"); // c=AB
>plotSegment(B,C,"a"); // a=BC
>plotSegment(A,C,"b"); // b=AC
```

Fungsi geometri mencakup fungsi untuk membuat garis dan lingkaran. Format untuk garis adalah [a,b,c], yang merepresentasikan garis dengan persamaan $ax+by=c$.

```
>lineThrough(B,C) // garis yang melalui B dan C
```

$[-1, 2, 2]$

Hitung garis tegak lurus yang melalui A pada BC

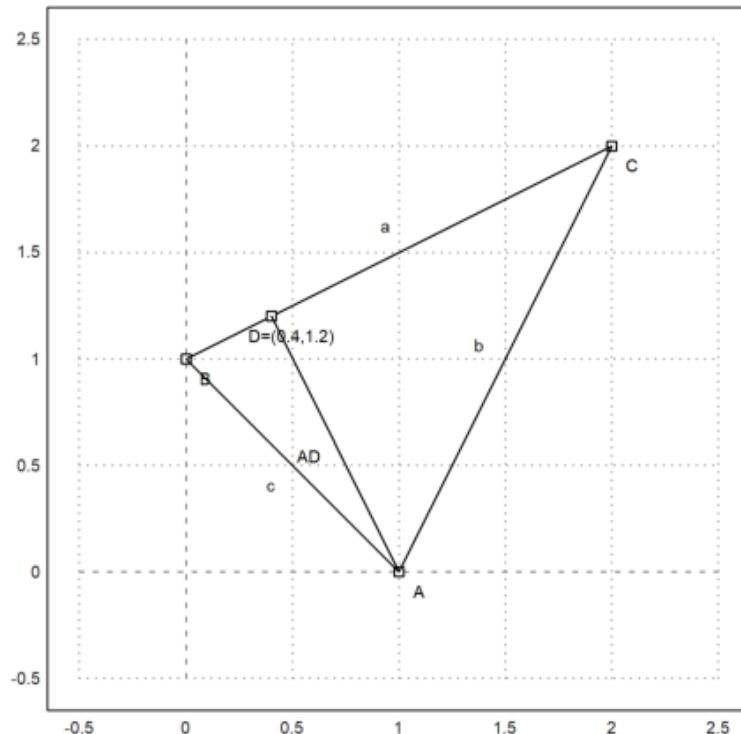
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan persimpangannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Plotkan itu

```
>plotPoint(D,value=1); // koordinat D ditampilkan
>aspect(1); plotSegment(A,D); // tampilkan semua gambar hasil plot...()
```



Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2} AD \cdot BC.$$

```
>norm(A-D) *norm(B-C) /2 // AD=norm(A-D) , BC=norm(B-C)
```

1.5

Bandingkan dengan rumus determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langsung dengan fungsi
```

1.5

Cara lain menghitung luas segitiga ABC:

```
>distance(A,D) *distance (B,C) /2
```

1.5

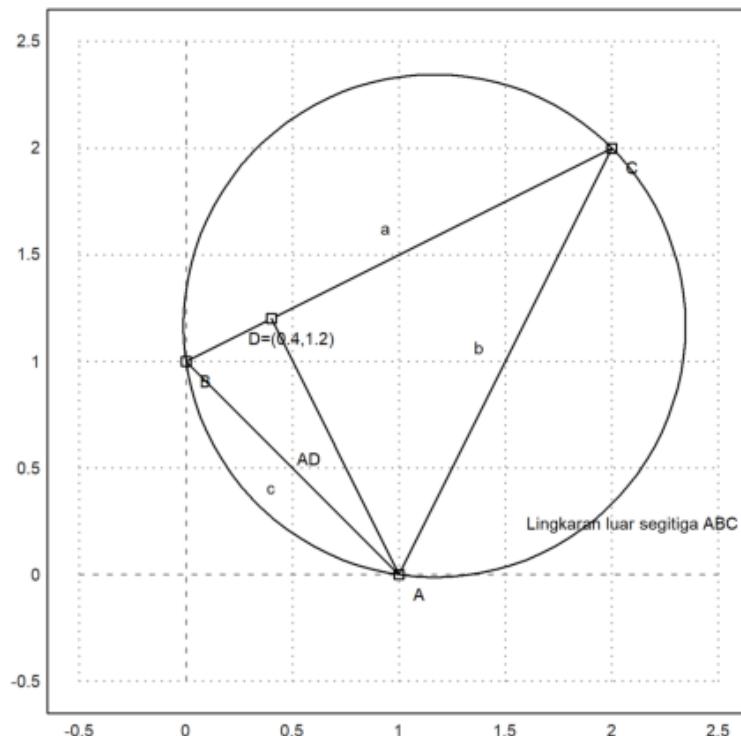
Sudut pada C

```
>degprint(computeAngle(B,C,A))
```

$36^\circ 52' 11.63''$

Sekarang, lingkarilah segitiga tersebut.

```
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
>R=getCircleRadius(c); // jari2 lingkaran luar
>O=getCircleCenter(c); // titik pusat lingkaran c plotPoint(O,"O"); // gambarkan titik pusat
>plotCircle(c,"Lingkaran luar segitiga ABC");
```



Menampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>O, R
```

[1.16667, 1.16667]
1.17851130198

Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

```
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

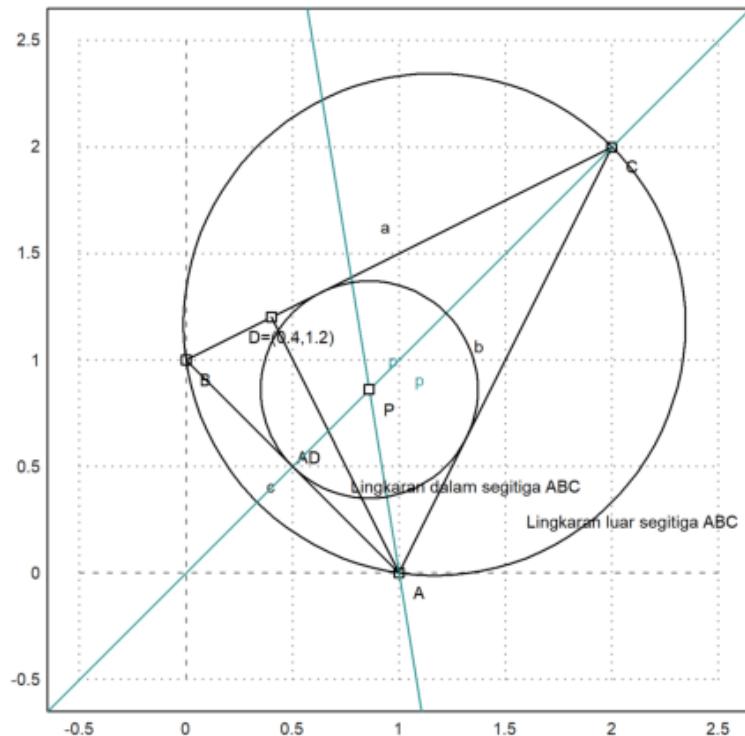
[0.86038, 0.86038]

Tambahkan semuanya ke plot.

```
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi s
>plotPoint(P, "P"); // gambar titik potongnya
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

0.509653732104

```
>plotCircle(circleWithCenter(P,r), "Lingkaran dalam segitiga ABC"); // gambar
```



1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

```
>reset
```

0

```
>setPlotRange(0,3,0,3);
>A=[1.5,0]; plotPoint(A,"A");
>B=[0,1.5]; plotPoint(B,"B");
>C=[2.5,2]; plotPoint(C,"C");
```

2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.

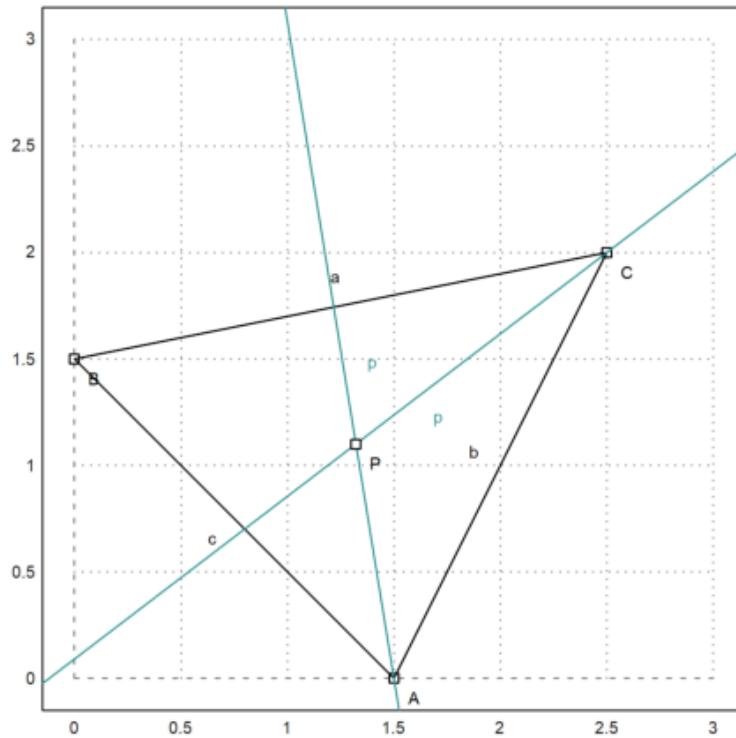
```
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
>plotSegment(A,C,"b");
```

3. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

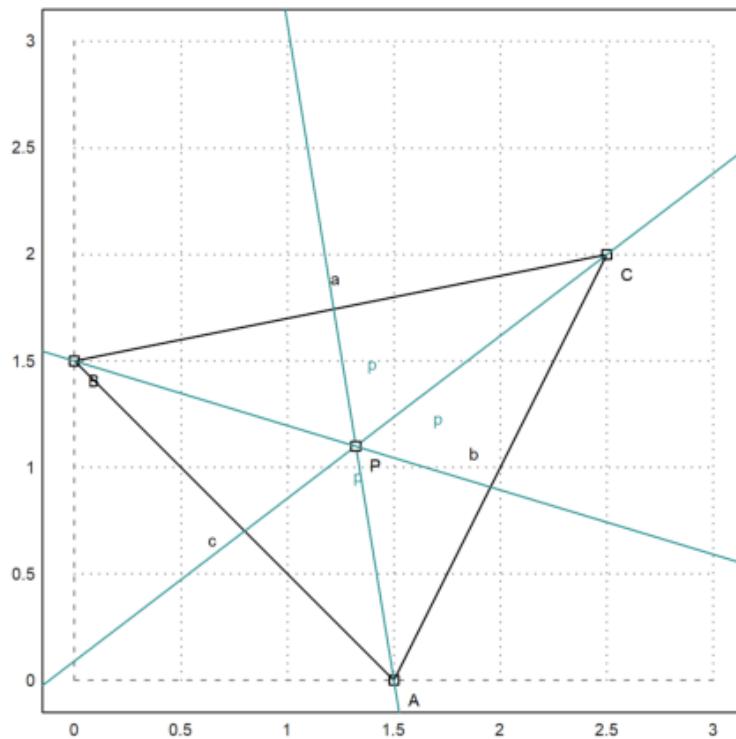
```
>g=angleBisector(C,A,B);
>l=angleBisector(A,C,B);
>P=lineIntersection(l,g)
```

[1.32151, 1.09988]

```
>color(5); plotLine(l); plotLine(g); color(1);
>plotPoint(P,"P");
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC");
```



```
>j=angleBisector(A,B,C);
>color(5); plotLine(j);
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):
```



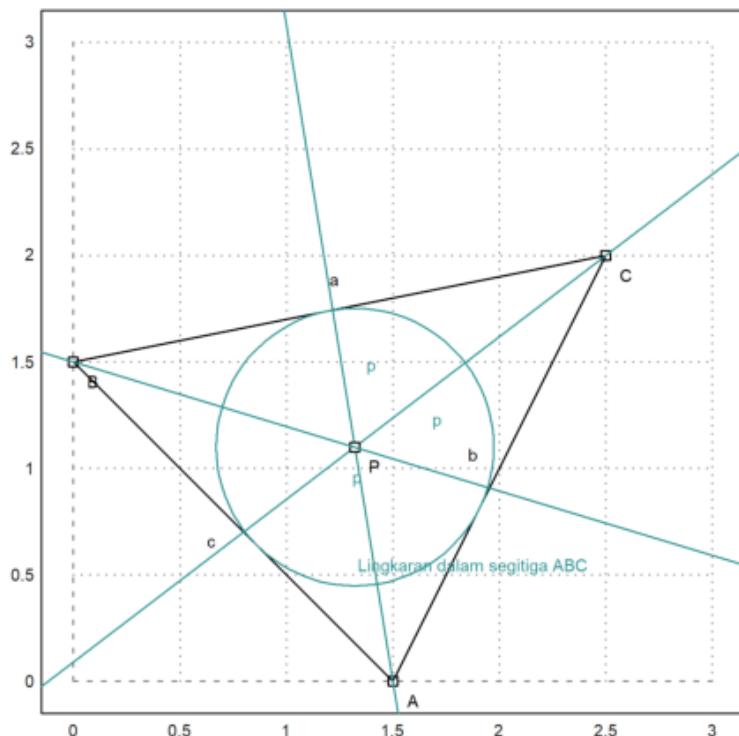
Jadi, terbukti bahwa garis bagi sudut ketiga juga melalui titik pusat lingkaran

4. Gambar jari-jari lingkaran dalam.

```
>r=norm(P-projectToLine(P, lineThrough(A, B)) )
```

0.651522571967

```
>plotCircle(circleWithCenter(P, r), "Lingkaran dalam segitiga ABC") :
```



Contoh 2: Geometri Smbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometri.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, kita dapat menggunakan komputasi simbolik sekarang.

```
>A &= [1, 0]; B &= [0, 1]; C &= [2, 2]; // menentukan tiga titik A, B, C
```

Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi menyediakan komputasi simbolis

```
>c &= lineThrough(B,C) // c=BC
```

[- 1, 2, 2]

Kita bisa mendapatkan persamaan untuk sebuah garis dengan mudah.

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$\left[y = \frac{x}{2} + 1 \right]$$

$$\left[y = \frac{x}{2} + 1 \right]$$

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A
```

$$(x_1 - 1) y - x y_1 = -y_1$$

```
>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 m
```

$$\left[\frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}} \right]$$

```
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

$$1.178511301977579$$

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // ti
```

$$\left[\frac{\sqrt{2}\sqrt{5} + 2}{6}, \frac{\sqrt{2}\sqrt{5} + 2}{6} \right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

[0.86038, 0.86038]

Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga bisa memotong garis dengan lingkaran, dan lingkaran dengan lingkaran.

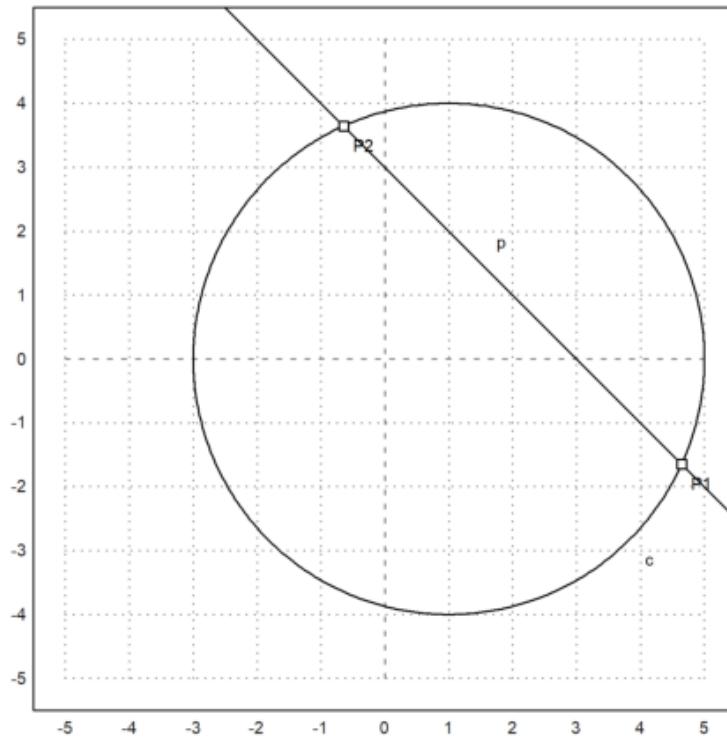
```
>A &:= [1,0]; c=circleWithCenter(A,4);
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l);
```

Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik perpotongan.

```
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2, f
```

[4.64575, -1.64575]
[-0.645751, 3.64575]
2

```
>plotPoint(P1); plotPoint(P2);
```



Hal yang sama pada Maxima.

```
>c &= circleWithCenter(A, 4) // lingkaran dengan pusat A jari-jari 4
```

```
[1, 0, 4]
```

```
>l &= lineThrough(B,C) // garis l melalui B dan C
```

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

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan ga
```

$$\left[\left[\sqrt{7} + 2, 1 - \sqrt{7} \right], \left[2 - \sqrt{7}, \sqrt{7} + 1 \right] \right]$$

Akan ditunjukkan bahwa sudut-sudut yang menghadap bsumur yang sama adalah sama besar.

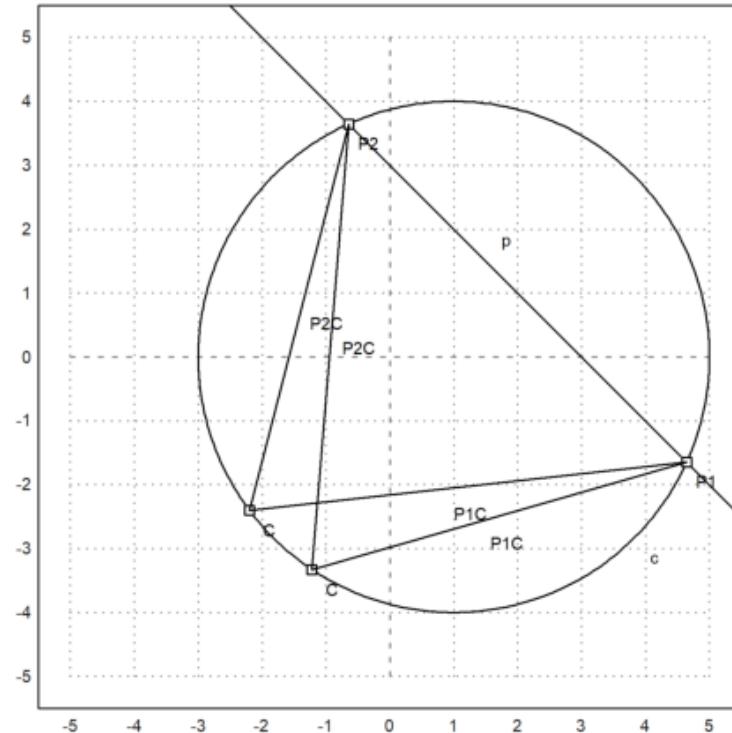
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>deprint(computeAngle(P1,C,P2))
```

$69^\circ 17' 42.68''$

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>deprint(computeAngle(P1,C,P2))
```

$69^\circ 17' 42.68''$

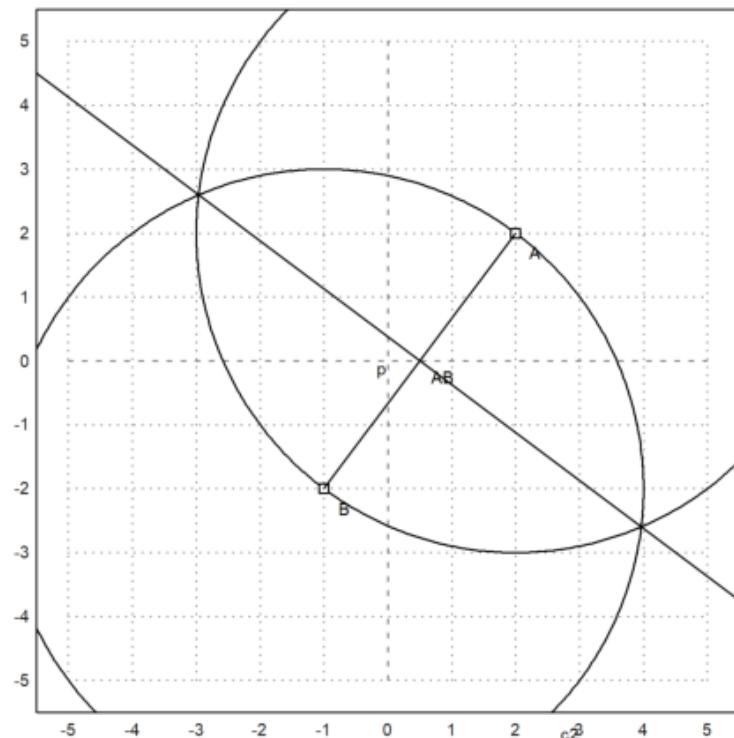
```
>insimg;
```



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, kami melakukan hal yang sama di Maxima dengan koordinat umum.

```
>A &= [a1,a2]; B &= [b1,b2];
>c1 &= circleWithCenter(A,distance(A,B));
>c2 &= circleWithCenter(B,distance(A,B));
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];
```

Persamaan untuk persimpangan cukup rumit. Tetapi kita dapat menyederhanakannya, jika kita menyelesaikan untuk y.

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)
```

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

Ini memang sama dengan tegak lurus tengah, 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)
```

$$\left[y = \frac{(b_2 - a_2)x - a_1b_2 + a_2b_1}{b_1 - a_1} \right]$$

Perhatikan hasil kali gradien garis g dan h adalah:

$$\frac{-(b_1 - a_1)}{(b_2 - a_2)} \times \frac{(b_2 - a_2)}{(b_1 - a_1)} = -1.$$

Artinya kedua garis tegak lurus.

Contoh 3: Rumus Heron

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a, b dan c adalah:

$$L = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{dengan } s = (a+b+c)/2,$$

atau bisa ditulis dalam bentuk lain:

$$L = \frac{1}{4} \sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}$$

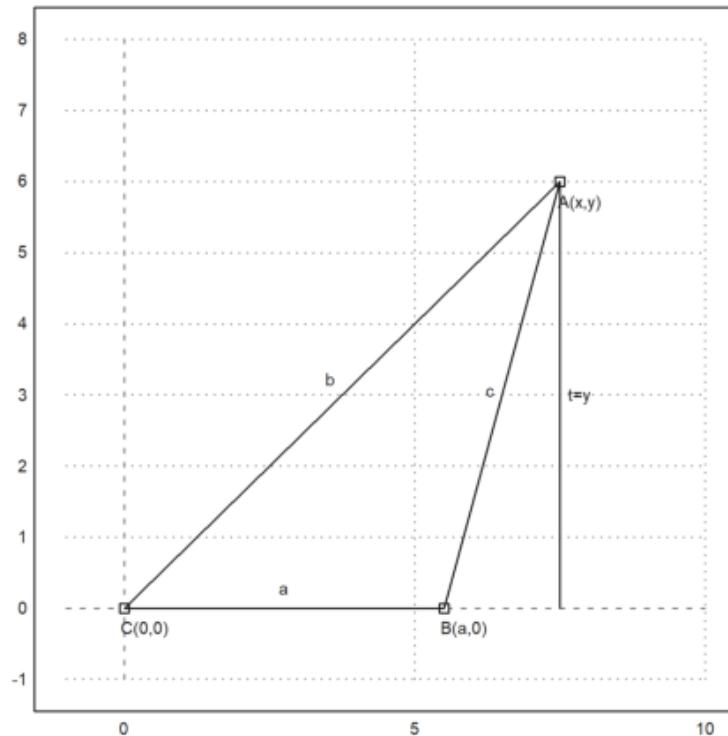
Untuk membuktikan hal ini kita misalkan $C(0,0)$, $B(a,0)$ dan $A(x,y)$, $b=AC$, $c=AB$. Luas segitiga ABC adalah

$$L_{\triangle ABC} = \frac{1}{2}a \times y.$$

Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2 + y^2 = b^2, \quad (x - a)^2 + y^2 = c^2.$$

```
>setPlotRange (-1,10,-1,8); plotPoint([0,0], "C(0,0)"); plotPoint([5.5,0], "  
> plotPoint([7.5,6], "A(x,y)");  
>plotSegment([0,0],[5.5,0], "a",25); plotSegment([5.5,0],[7.5,6],"c",15);  
>plotSegment([0,0],[7.5,6],"b",25);  
>plotSegment([7.5,6],[7.5,0],"t=y",25):
```



```
>&assume(a>0); sol &= solve([x^2+y^2=b^2, (x-a)^2+y^2=c^2], [x,y])
```

[]

Ekstrak solusi y

```
>ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2))
```

Maxima said:

part: invalid index of list or matrix.
-- an error. To debug this try: debugmode(true);

Error in:

```
ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2)) ...  
^
```

Kita dapatkan formula 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
```

$$Luas = H(2, 5, 6)$$

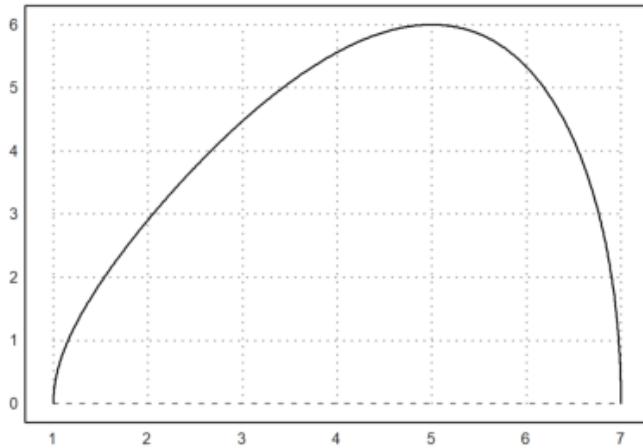
Tentu saja, setiap segitiga panjang adalah kasus yang umum.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

6

Dan juga jelas, bahwa ini adalah segitiga dengan luas maksimal dan dua sisi 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7); // Kurva luas segitiga sengan panjang
```



Kasus umum juga bisa digunakan.

```
> $solve (diff (H(a,b,c)^2, c) = 0, c)
```

$$\left[c = -\sqrt{b^2 + a^2}, c = \sqrt{b^2 + a^2}, c = 0 \right]$$

Sekarang mari kita cari himpunan semua titik di mana $b+c=d$ untuk suatu 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 hal ini.

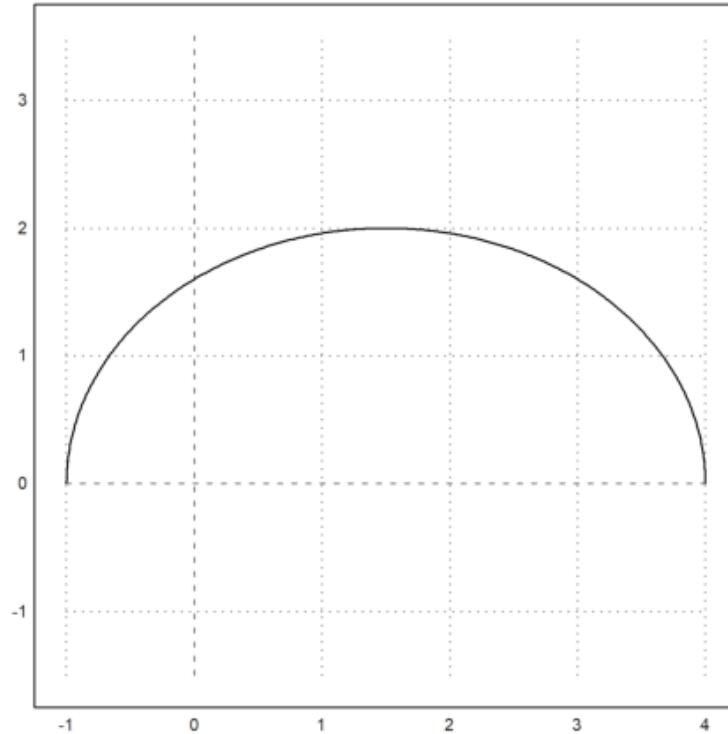
```
> function fx (a, c, d) &= rhs (s1[1]); $fx(a, c, d), function fy (a, c, d) &= rhs (s1[2]); $fy(a, c, d)
```

$$\frac{\sqrt{-(d-c)^4 + 2c^2(d-c)^2 + 2a^2(d-c)^2 - c^4 + 2a^2c^2 - a^4}}{2a}$$

$$\frac{\sqrt{-(d-c)^4 + 2c^2(d-c)^2 + 2a^2(d-c)^2 - c^4 + 2a^2c^2 - a^4}}{2a}$$

Sekarang kita dapat menggambar himpunan tersebut. Sisi b bervariasi dari 1 hingga 4. Sudah diketahui bahwa kita 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 pusat, dan u serta v adalah setengah sumbu.

```
>$ratsimp((fx(a,c,d)-a/2)^2/u^2+fy(a,c,d)^2/v^2 with [u=d/2,v=sqrt(d^2-a^2)]
```

1

Kita melihat bahwa tinggi dan luas segitiga adalah maksimal untuk $x=0$. Dengan demikian, luas segitiga dengan $a+b+c=d$ adalah maksimal, jika segitiga tersebut sama sisi. Kita ingin membuktikannya 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 minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusi $a = b = c = d / 3$

```
>$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 $H(a,b,c)^2$ sehubungan dengan $a+b+d=d$

```
>&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 bisa membuat plot situasi, pertama-tama tetapkan titik-titik di Maxima.

```
>A &= at([x,y],sol[2]); $A
```

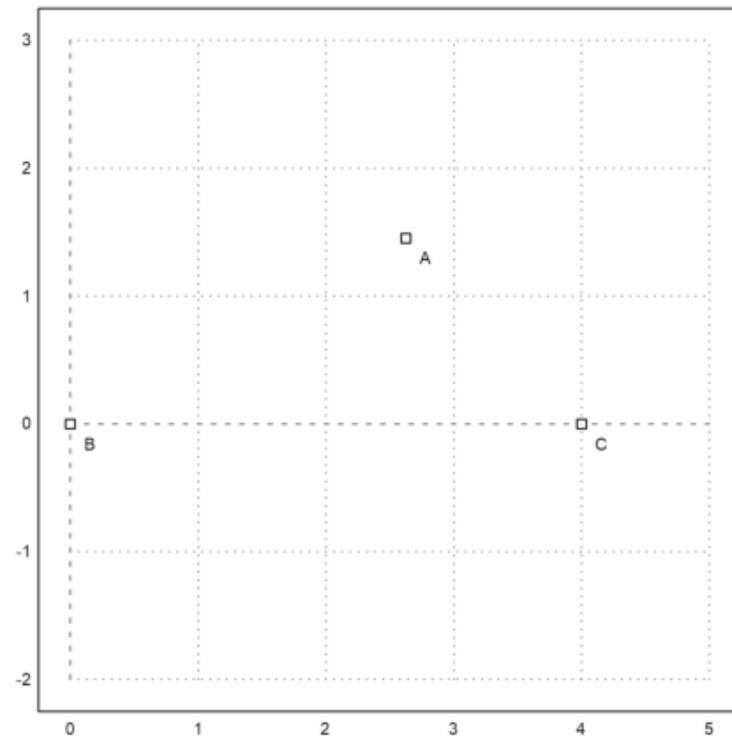
$$\left[\frac{-c^2 + b^2 + a^2}{2a}, \frac{\sqrt{-c^4 + 2b^2c^2 + 2a^2c^2 - b^4 + 2a^2b^2 - a^4}}{2a} \right]$$

```
>B &= [0,0]; $B, C &= [a,0]; $C
```

$[a, 0]$

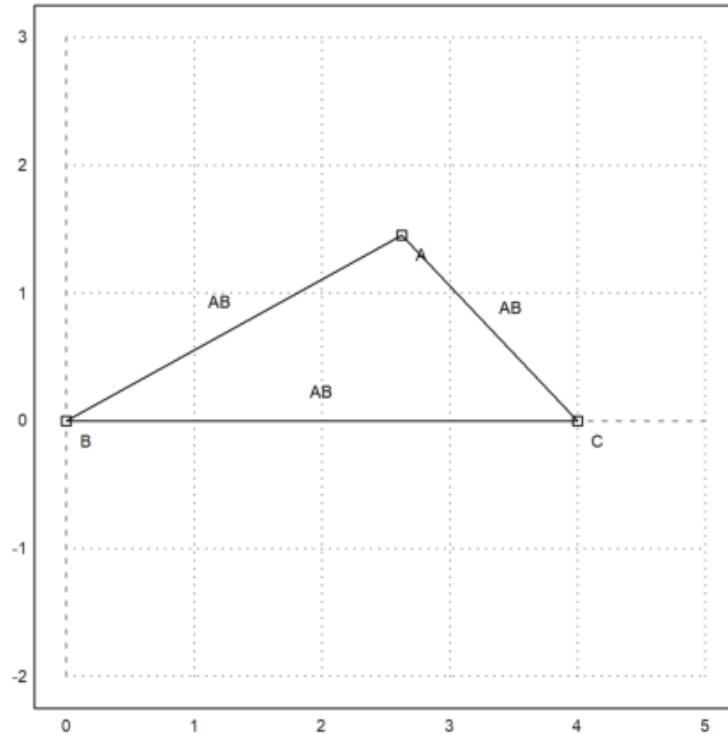
Kemudian, tetapkan kisaran plot, dan plot titik-titiknya.

```
>setPlotRange(0,5,-2,3); ...
>a=4; b=3; c=2; ...
>plotPoint(mxmeval("B"), "B"); plotPoint(mxmeval("C"), "C"); ...
>plotPoint(mxmeval("A"), "A");
```



Plot segmen-segmen tersebut.

```
>plotSegment(mxmeval("A"), mxmeval("C")); ...
>plotSegment(mxmeval("B"), mxmeval("C")); ...
>plotSegment(mxmeval("B"), mxmeval("A"));
```



Hitung garis tegak lurus tengah dalam Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan bagian tengah lingkar.

```
>U &= lineIntersection(h,g);
```

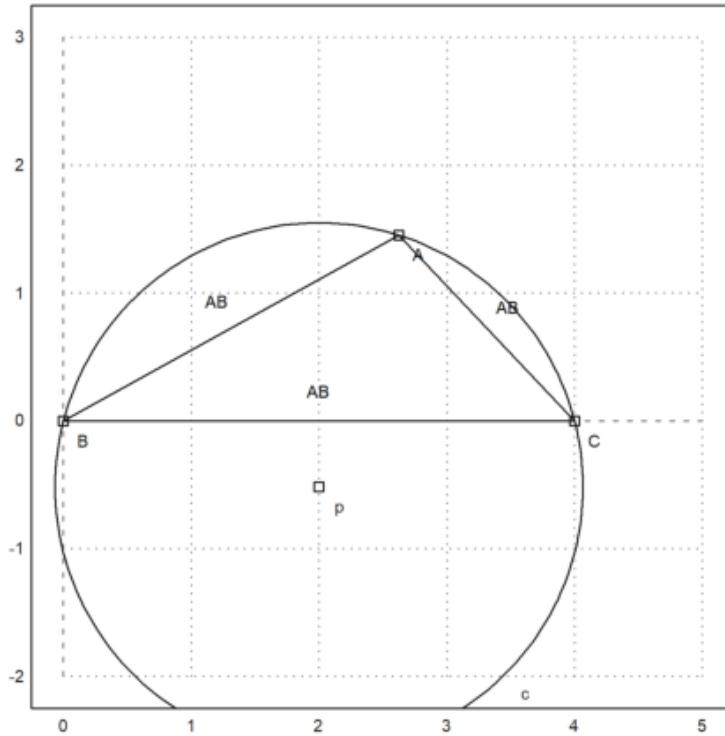
Kita mendapatkan rumus untuk jari-jari lingkaran.

```
>&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 ke plot

```
>plotPoint(U()); ...
>plotCircle(circleWithCenter(mxmeval("U")),mxmeval("distance(U,C)"));
```



Dengan menggunakan geometri, kami memperoleh rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk radius. Kita bisa mengecek, apakah hal ini benar dengan Maxima. Maxima akan memperhitungkannya 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. Garis ini merupakan garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk ortosentrum, circumcenter, centroid, titik Exeter, dan pusat lingkaran sembilan titik segitiga.

Sebagai demonstrasi, kami menghitung dan memplot garis Euler dalam sebuah segitiga. Pertama, kita mendefinisikan sudut-sudut segitiga dalam Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolis.

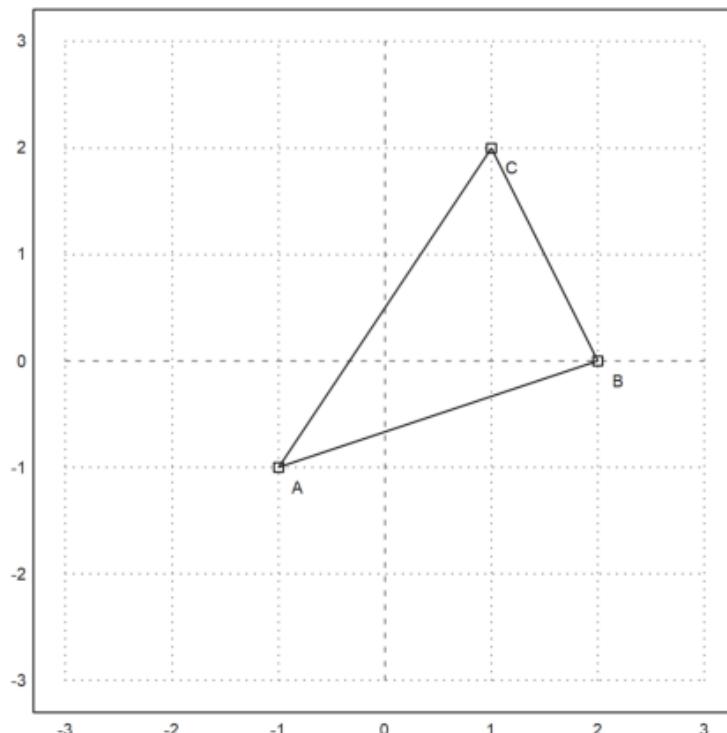
```
>A:::=[-1,-1]; B:::[2,0]; C:::[1,2];
```

Untuk memplot objek geometris, kita menyiapkan area plot, dan menambahkan titik-titiknya. Semua plot objek geometris ditambahkan ke plot saat ini.

```
>setPlotRange(3); plotPoint(A, "A"); plotPoint(B, "B"); plotPoint(C, "C");
```

Kita juga bisa menambahkan sisi-sisi segitiga

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,"");
```



Berikut ini adalah luas area segitiga, dengan 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 formula 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 akan menghasilkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$$\frac{7}{\sqrt{10}}$$

$$\frac{7}{\sqrt{10}}$$

Sekarang kita hitung lingkaran ABC

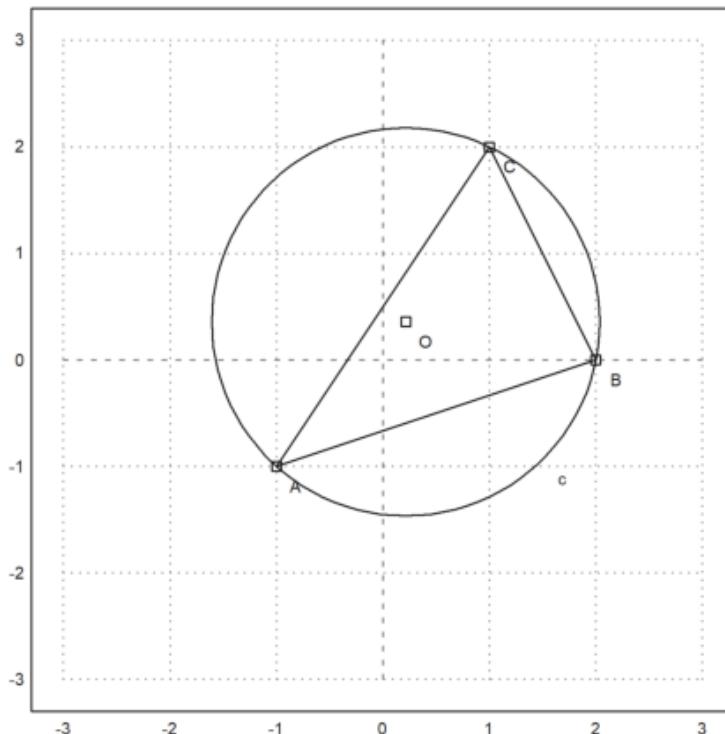
```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

$$\left(y - \frac{5}{14}\right)^2 + \left(x - \frac{3}{14}\right)^2 = \frac{325}{98}$$

```
>O &= getCircleCenter(LL); $O
```

$$\left[\frac{3}{14}, \frac{5}{14}\right]$$

```
>plotCircle(LL()); plotPoint(O(), "O");
```



Kita dapat menghitung perpotongan ketinggian di ABC (pusat ortosentrum) secara numerik dengan perintah berikut ini.

```
>H &= lineIntersection(perpendicular(A, lineThrough(C, B)), ...  
>    perpendicular(B, lineThrough(A, C))); $H
```

$$\left[\frac{11}{7}, \frac{2}{7} \right]$$

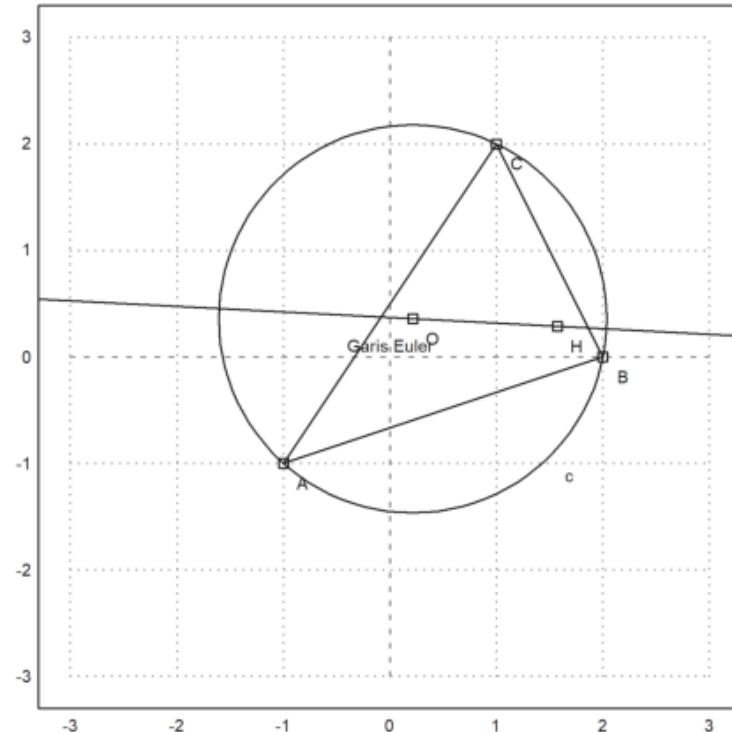
Sekarang kita hitung garis Euler dari segitiga

```
>el &= lineThrough(H, O); $getLineEquation(el, x, y)
```

$$-\frac{19y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan ke plot

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```

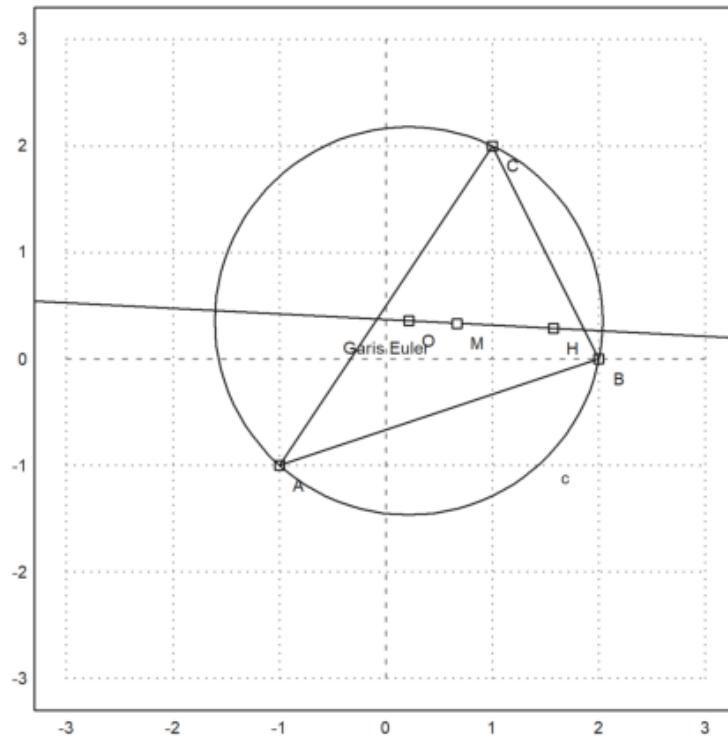


Pusat gravitasi harus berada pada garis ini.

```
>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 mengatakan bahwa $MH = 2 \cdot MO$. Kita perlu menyederhanakan dengan radcan untuk mencapai hal ini.

```
>$distance(M, H) / distance(M, O) | radcan
```

2

Fungsi ini juga termasuk fungsi fungsi untuk sudut

```
>$computeAngle(A, C, B), degprint(%())
```

$$\arccos\left(\frac{4}{\sqrt{5}\sqrt{13}}\right)$$

$60^\circ 15' 18.43''$

Persamaan untuk bagian tengah lingkaran tidak terlalu bagus.

```
>Q &= lineIntersection(angleBisector(A, C, B), angleBisector(C, B, A)) | radcan; $
```

$$\left[\frac{\left(2^{\frac{3}{2}} + 1\right) \sqrt{5} \sqrt{13} - 15 \sqrt{2} + 3}{14}, \frac{(\sqrt{2} - 3) \sqrt{5} \sqrt{13} + 5 2^{\frac{3}{2}} + 5}{14} \right]$$

Mari kita hitung juga ekspresi untuk jari-jari lingkaran yang tertulis.

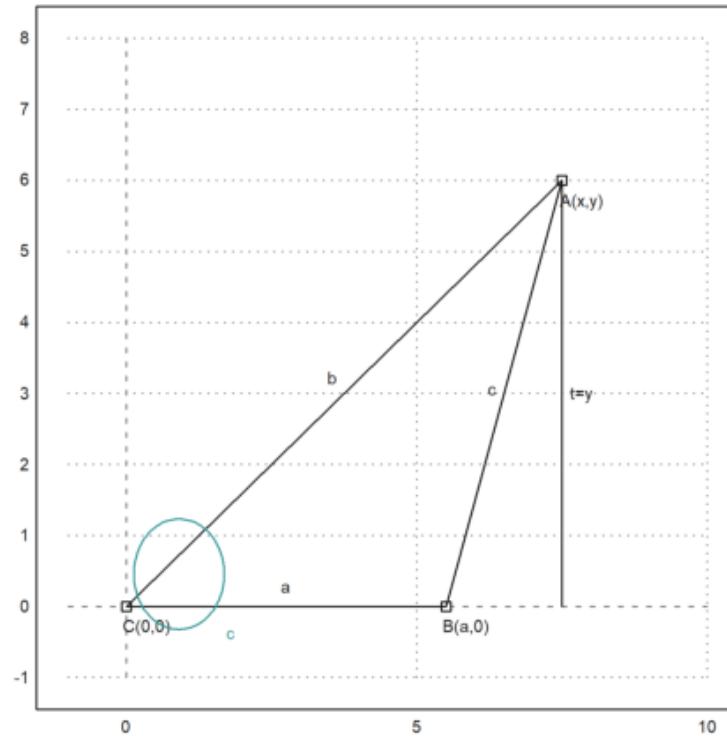
```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{(-41\sqrt{2} - 31)\sqrt{5}\sqrt{13} + 115\sqrt{2} + 614}}{7\sqrt{2}}$$

```
>LD &= circleWithCenter(Q,r); // Lingkaran dalam
```

tambahkan ke plot

```
>color(5); plotCircle(LD());
```



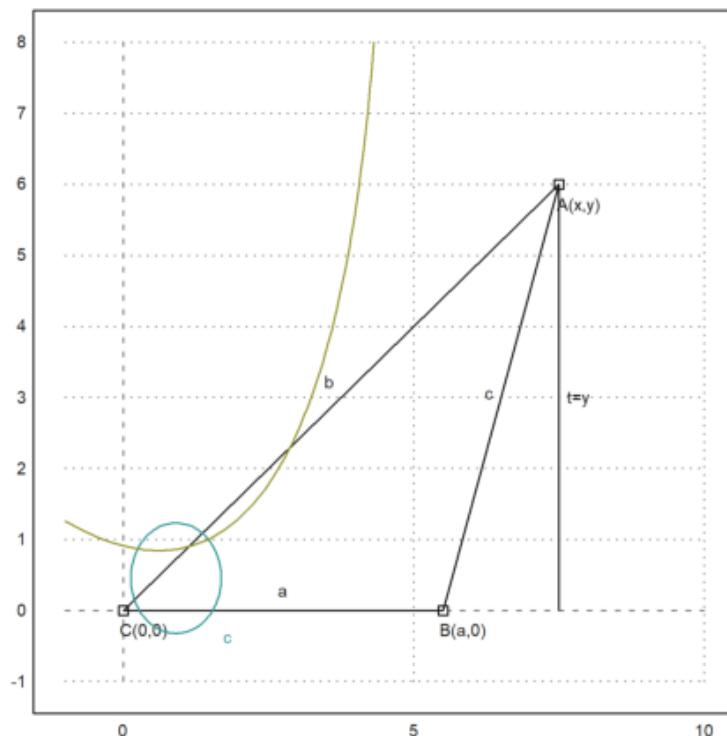
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>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 merupakan suatu fungsi, tetapi solver default Maxima hanya bisa menemukan solusinya 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)
```

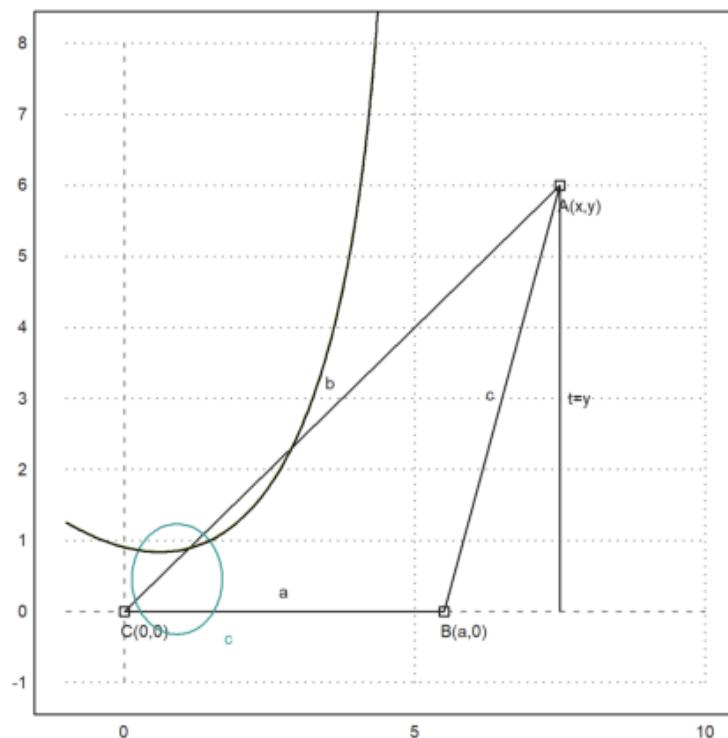
$$\begin{aligned} y &= -3x - \sqrt{70} \sqrt{9 - 2x} + 26, \\ y &= -3x + \sqrt{70} \sqrt{9 - 2x} + 26 \end{aligned}$$

Solusi pertama adalah

$$y = -3x - \sqrt{70} \sqrt{9 - 2x} + 26$$

Menambahkan solusi pertama ke plot menunjukkan bahwa itu memang yang kita cari. Teori menunjukkan bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]), add=1):
```



```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan k
```

$$g(x) = -3x - \sqrt{70} \sqrt{9 - 2x} + 26$$

```
>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(%) // jarak 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

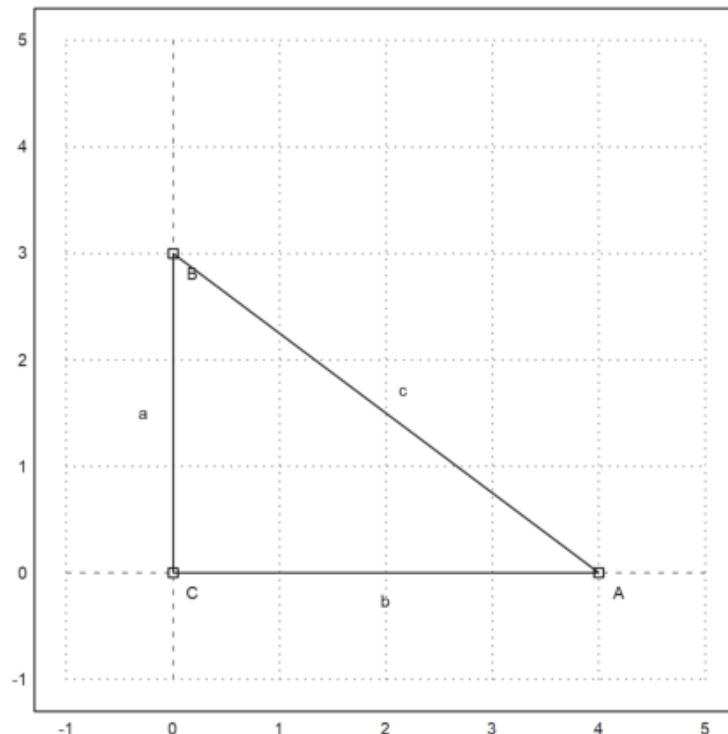
Hal ini terinspirasi dari sebuah ceramah N.J. Wildberger. Dalam bukunya "Proporsi Ilahi", Wildberger mengusulkan untuk mengganti gagasan klasik tentang jarak dan sudut dengan kuadransi dan penyebaran. Dengan menggunakan ini, memang memungkinkan untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

Berikut ini, saya akan memperkenalkan konsep-konsepnya, dan memecahkan beberapa masalah. Saya menggunakan komputasi simbolis Maxima di sini, yang menyembunyikan keuntungan utama dari trigonometri rasional yang komputasinya dapat dilakukan dengan kertas dan pensil saja. Anda dipersilakan untuk memeriksa hasilnya tanpa komputer. Intinya adalah bahwa komputasi rasional simbolis sering kali memberikan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang dievaluasi dengan perkiraan numerik saja

```
>load geometry;
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
```

Untuk pengenalan pertama, kita menggunakan segitiga persegipanjang dengan proporsi Mesir yang terkenal 3, 4, dan 5. Perintah berikut ini adalah perintah Euler untuk memplot geometri bidang yang terdapat pada file Euler "geometry.e".

```
>plotSegment (B,A,"c"); plotSegment (A,C,"b"); plotSegment (C,B,"a"); ...
>insimg(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 kebalikan dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak kira-kira.

```
>wa := arcsin(3/5); degprint(wa)
```

$36^\circ 52' 11.63''$

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadran, yang menggantikan jarak. Sebenarnya, ini hanyalah jarak yang dikuadratkan. Berikut ini, a, b, dan c menunjukkan kuadran sisi-sisinya.

Teorema Pythagoras menjadi $a+b=c$. Maka, teorema Pythagoras menjadi $a+b=c$.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

$$25 = 25$$

Gagasan kedua dari trigonometri rasional adalah penyebaran. Penyebaran mengukur bukaan di antara garis-garis. Ini adalah 0, jika garis-garisnya sejajar, dan 1, jika garis-garisnya persegi panjang. Ini adalah kuadran dari sinus sudut antara kedua garis tersebut.

Penyebaran garis AB dan AC pada gambar di atas didefinisikan sebagai

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

di mana a dan c adalah kuadran dari segitiga persegi panjang dengan satu sudut di A.

```
>sa &= a/c; $sa
```

$$\frac{9}{25}$$

Tentu saja, hal ini lebih mudah dihitung daripada sudut. Tetapi Anda kehilangan sifat bahwa sudut dapat ditambahkan dengan mudah. Tentu saja, kita bisa mengonversi nilai perkiraan kita untuk sudut wa ke sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

$$9/25$$

Hukum kosinus trigonometri klasik diterjemahkan ke dalam "hukum silang" berikut ini.

$$(c + b - a)^2 = 4bc(1 - s_a)$$

Di sini, a, b, dan c adalah kuadran dari sisi-sisi segitiga, dan sa adalah penyebaran di sudut A. Sisi a, seperti biasa, berlawanan dengan sudut A. Hukum-hukum ini diimplementasikan dalam file geometry.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 dapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024 = 1024$$

Mari kita gunakan crosslaw ini untuk mencari sebaran di A. Untuk melakukannya, kita buat crosslaw untuk kuadran a, b, dan c, dan selesaikan untuk sebaran sa yang tidak diketahui. Anda bisa melakukan ini dengan tangan dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasil yang sudah kami dapatkan.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

$$\left[x = \frac{9}{25} \right]$$

$$\left[x = \frac{9}{25} \right]$$

Kita sudah mengetahui hal ini. Definisi spread adalah kasus khusus dari crosslaw. Kita juga dapat menyelesaiannya untuk a, b, c secara umum. Hasilnya adalah sebuah rumus yang menghitung penyebaran sudut segitiga dengan kuadran ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[x = \frac{-cc^2 - (-2 bb - 2 aa) cc - bb^2 + 2 aa bb - aa^2}{4 bb cc} \right]$$

Kita dapat membuat sebuah fungsi dari hasil tersebut. Fungsi seperti itu sudah didefinisikan dalam file geometry.e dari Eule

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Sebagai contoh, kita dapat menggunakan untuk menghitung sudut segitiga dengan sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak mudah didapat jika kita menggunakan trigonometri klasik

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat

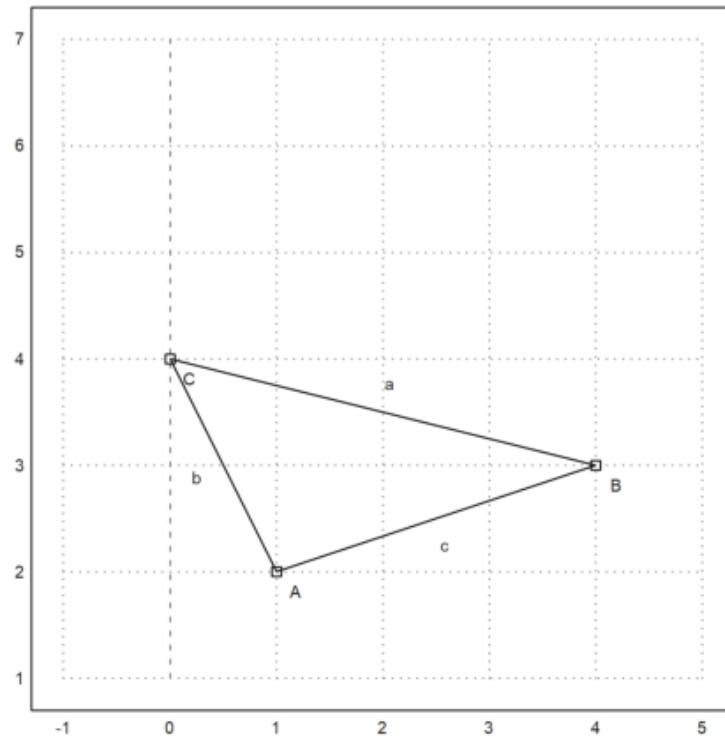
```
>degprint(arcsin(sqrt(6/7)))
```

$$67^\circ 47' 32.44''$$

contoh lain

sekarang, mari kita coba contoh yang lebih lanjut. kami 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;
```



Dengan menggunakan Pythagoras, mudah untuk menghitung jarak antara dua titik. Pertama-tama saya menggunakan jarak fungsi dari file Euler untuk geometri. Jarak fungsi menggunakan geometri klasik.

```
>$distance(A, B)
```

$$\sqrt{10}$$

Euler juga memiliki fungsi untuk kuadransi antara dua titik. Pada contoh berikut, karena $c+b$ bukan a , maka segitiga tersebut tidak berbentuk persegi panjang.

```
>c &= quad(A, B); $c, b &= quad(A, C); $b, a &= quad(B, C); $a,
```

Pertama, mari kita menghitung sudut tradisional. Fungsi computeAngle menggunakan metode yang biasa berdasarkan hasil kali titik dari dua vektor. Hasilnya adalah beberapa perkiraan titik mengambang.

$$A = \langle 1, 2 \rangle \quad B = \langle 4, 3 \rangle, \quad C = \langle 0, 4 \rangle$$

$$\mathbf{a} = C - B = \langle -4, 1 \rangle, \quad \mathbf{c} = A - B = \langle -3, -1 \rangle, \quad \beta = \angle ABC$$

$$\mathbf{a} \cdot \mathbf{c} = |\mathbf{a}| \cdot |\mathbf{c}| \cos \beta$$

$$\cos \angle ABC = \cos \beta = \frac{\mathbf{a} \cdot \mathbf{c}}{|\mathbf{a}| \cdot |\mathbf{c}|} = \frac{12 - 1}{\sqrt{17} \sqrt{10}} = \frac{11}{\sqrt{17} \sqrt{10}}$$

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

$$\arccos \left(\frac{11}{\sqrt{10} \sqrt{17}} \right)$$

32.4711922908

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kita masukkan kuadran a, b, dan c ke dalam hukum silang dan selesaikan 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 dengan menggunakan trigonometri biasa, jika kita memaksakannya. Ia menyelesaikan suku $\sin(\arccos(...))$ menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{49}{170}$$

Setelah kita memiliki penyebaran di B, kita dapat menghitung tinggi ha di sisi a.

$$s_b = \frac{h_a}{c}$$

Ingatlah bahwa menurut definisi

```
>ha &= c*sb; $ha
```

$$\frac{49}{17}$$

Gambar berikut ini telah dibuat dengan program geometri C.a.R., yang dapat menggambar kuadran dan penyebaran.

e: (20) Rational_Geometry_CaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadrannya.

```
>$sqrt(ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita berurusan dengan kuadran!

```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{7}{2}$$

Rumus penentu yang biasa menghasilkan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

$$\frac{7}{2}$$

Formula Heron

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue(a,b,c,sb,ha);
```

Pertama-tama kita menghitung luas di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat ("quadrea"?), memfaktorkannya dengan Maxima, dan kita mendapatkan rumus Heron yang terkenal.

Memang, hal ini sulit dilakukan dengan pensil dan kertas

```
>$spread(b^2,c^2,a^2), $factor(%*c^2*a^2/4)
```

$$\frac{(-c + b + a) (c - b + a) (c + b - a) (c + b + a)}{16}$$

$$\frac{(-c + b + a) (c - b + a) (c + b - a) (c + b + a)}{16}$$

Aturan Triple Spread

Kerugian dari spread adalah bahwa mereka tidak lagi hanya menambahkan sudut seperti. Namun, tiga spread 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 yang berjumlah 180° .

$$\alpha + \beta + \gamma = \pi$$

Karena spread dari

$$\alpha, \pi - \alpha$$

adalah sama, aturan triple spread juga benar, jika

$$\alpha, \pi - \alpha$$

Karena penyebaran sudut negatif adalah sama, aturan penyebaran tiga kali lipat juga berlaku, jika

$$\alpha + \beta = \gamma$$

Contohnya, kita bisa menghitung penyebaran sudut 60° . Hasilnya adalah $3/4$. Namun, persamaan ini memiliki solusi kedua, di mana semua penyebarannya adalah 0.

```
>$solve(triplespread(x, x, x), x)
```

$$\left[x = \frac{3}{4}, x = 0 \right]$$

Penyebaran 90° jelas adalah 1. Jika dua sudut ditambahkan ke 90° , penyebarannya akan menyelesaikan persamaan penyebaran tiga dengan $a, b, 1$. Dengan perhitungan berikut, kita mendapatkan $a + b = 1$.

```
>$triplespread(x, y, 1), $solve(%, x)
```

$$[x = 1 - y]$$

Karena penyebaran $180^\circ - t$ sama dengan penyebaran t , rumus penyebaran tiga juga berlaku, jika salah satu sudut adalah jumlah atau selisih dari dua sudut lainnya. Jadi, kita dapat menemukan penyebaran sudut dua kali lipat. Perhatikan bahwa ada dua solusi lagi. Kita jadikan ini sebuah fungsi.

```
>$solve(triplespread(a, a, x), x), function doublespread(a) &= factor(rhs(%[1])
```

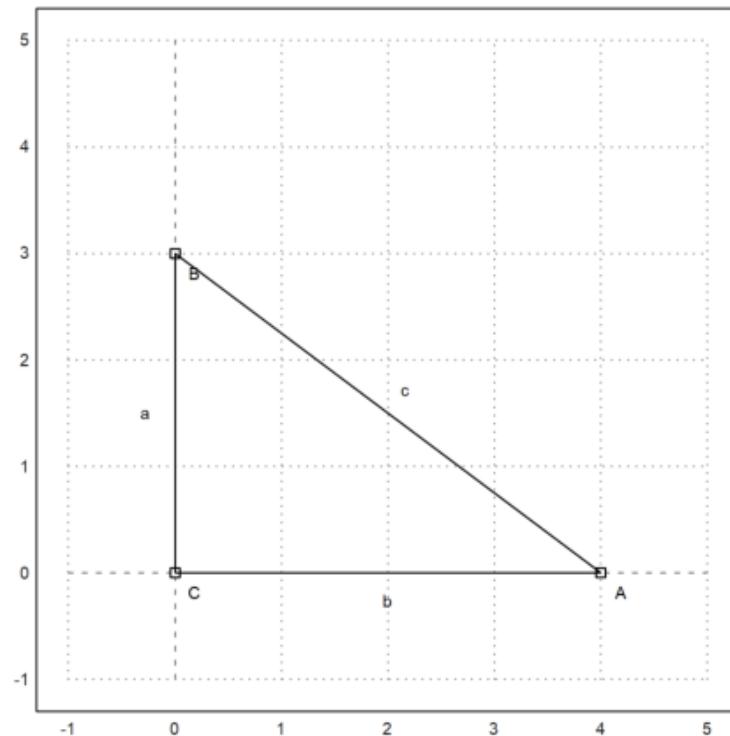
$$[x = 4a - 4a^2, x = 0]$$

$$- 4(a - 1) a$$

Pembagi Sudut

Ini adalah situasi yang 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 garis bagi sudut di A. Tetapi kita ingin menyelesaikannya untuk a, b, c secara umum.

```
>&remvalue (a,b,c);
```

Jadi, pertama-tama kita menghitung penyebaran sudut yang dibelah dua di A, dengan menggunakan rumus penyebaran tiga. Masalah dengan rumus ini muncul lagi. Rumus ini memiliki dua solusi. Kita harus memilih salah satu yang benar. Solusi lainnya mengacu pada sudut yang dibagi dua 180° -wa.

```
>$triplespread(x,x,a/(a+b)), $solve(% ,x), sa2 &= rhs(%[1]); $sa2
```

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

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

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

Mari kita periksa persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

Kita bisa mencetak sudut dalam Euler, setelah mentransfer penyebaran ke radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

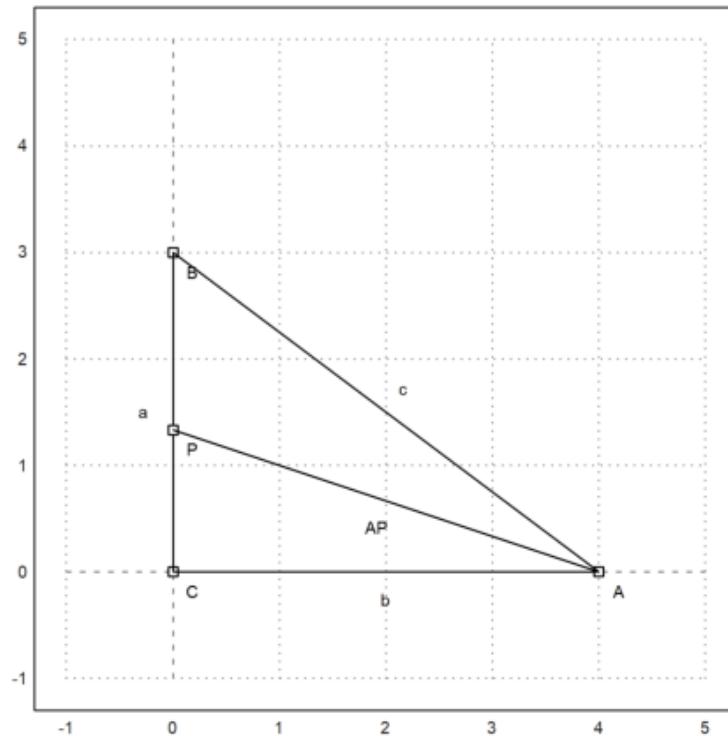
$$18^\circ 26' 5.82''$$

Titik P adalah perpotongan garis bagi sudut dengan sumbu y.

```
>P := [0, tan(wa2)*4]
```

$$[0, 1.33333]$$

```
>plotPoint(P, "P"); plotSegment(A,P):
```



Mari kita periksa sudut-sudutnya dalam contoh spesifik kita

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

0.321750554397

0.321750554397

Sekarang kita menghitung panjang garis bagi AP. Kami menggunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

berlaku dalam segitiga apa pun. Kuadratkan, ini diterjemahkan ke dalam apa yang disebut "hukum penyebaran"

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

di mana a, b, c menunjukkan qudrat. $a = b = c$ s s sa b b Karena spread CPA adalah 1-sa2, kita bisa mendapatkan bisa /1 = b/(1-sa2) dan bisa menghitung bisa (kuadran dari pembagi sudut)

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$\frac{2b(b+a)}{\sqrt{b(b+a)} + b + a}$$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa, [a=3^2, b=4^2])")), distance(A,P)
```

4.21637021356
4.21637021356

Kita juga dapat menghitung P menggunakan rumus penyebaran.

```
>py&=factor(ratsimp(sa2*bisa)); $py
```

$$-\frac{b \left(\sqrt{b(b+a)} - b - a \right)}{\sqrt{b(b+a)} + b + a}$$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri

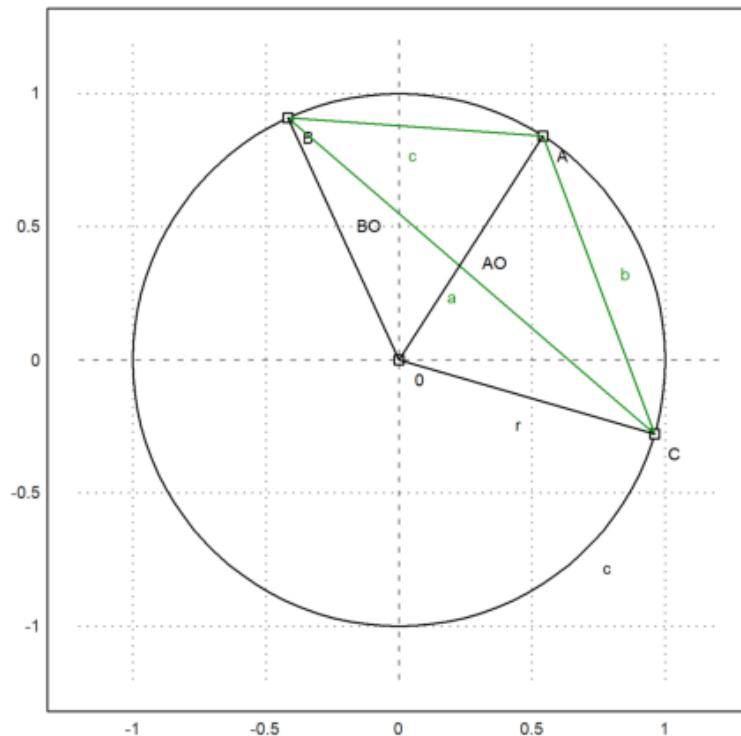
```
>sqrt(mxmeval("at(py, [a=3^2, b=4^2])"))
```

1.33333333333

```
>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")
```

Lihatlah situasi berikut ini.

```
>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 rumus penyebaran tiga kali lipat untuk sudut-sudut di pusat O untuk r . Dengan demikian kita mendapatkan rumus untuk jari-jari kuadrat dari pericircle dalam hal kuadran sisisisinya. Kali ini, Maxima menghasilkan beberapa angka nol yang rumit, yang kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),
```

$$-\frac{abc}{c^2 - 2bc + a(-2c - 2b) + b^2 + a^2}$$

Kita dapat menjadikannya sebuah fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasilnya untuk poin A, B, C.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Radiusnya memang 1.

```
>periradius(a,b,c)
```

1

Faktanya adalah, bahwa penyebaran CBA hanya bergantung pada b dan c. Ini adalah teorema sudut akor

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Pada kenyataannya, penyebarannya adalah $b/(4r)$, dan kita melihat bahwa sudut chord b adalah setengah dari sudut tengah.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

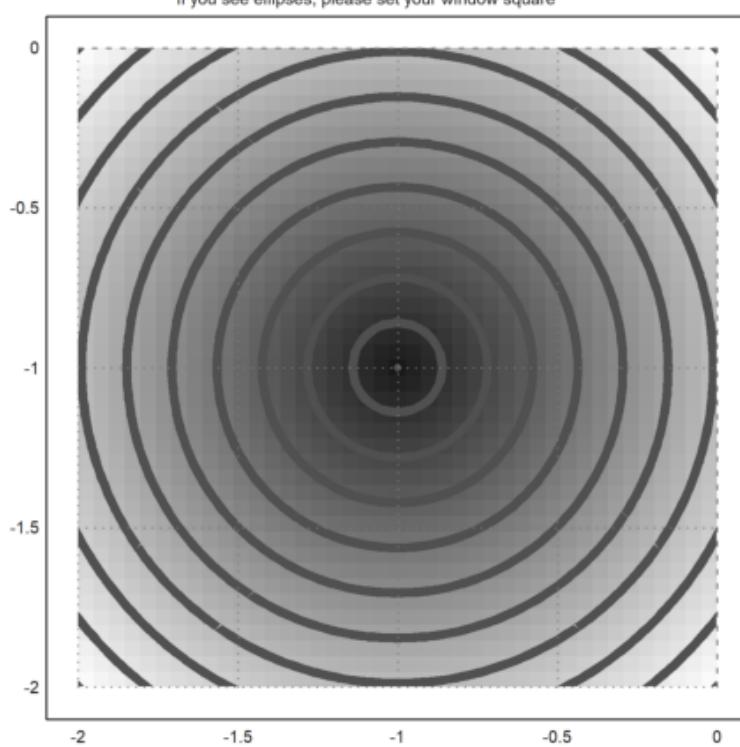
0

Contoh 6: Jarak Minimal pada Bidang

Komentar awal

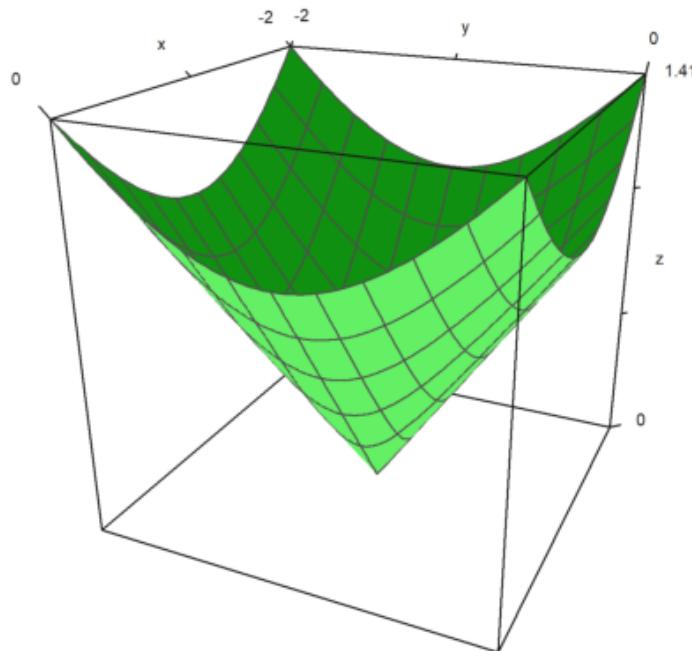
Fungsi yang, pada sebuah titik M pada bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis-garis tingkat yang cukup sederhana: lingkaran yang berpusat di A.

```
>&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) :
```

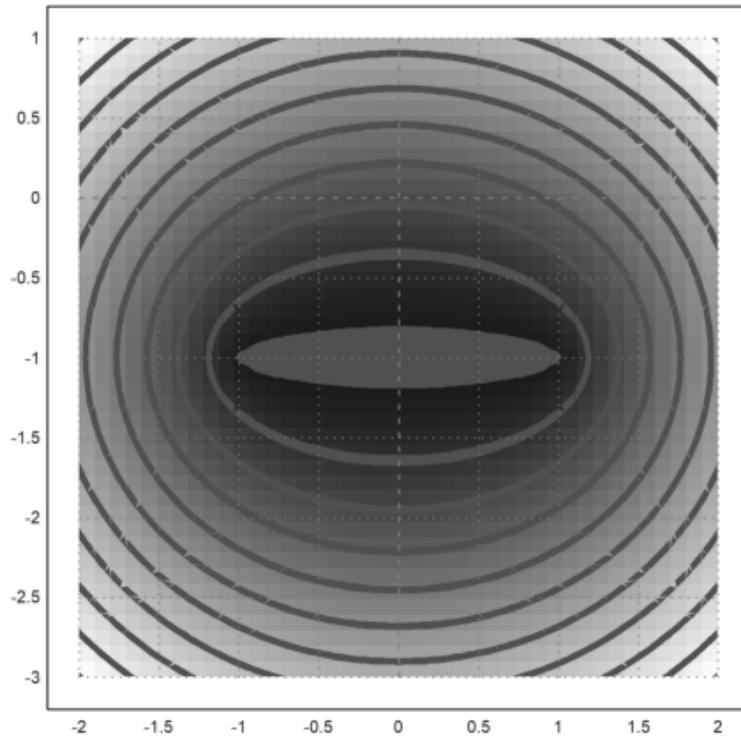


Tentu saja, nilai minimum 0 diperoleh di A.

Dua poin

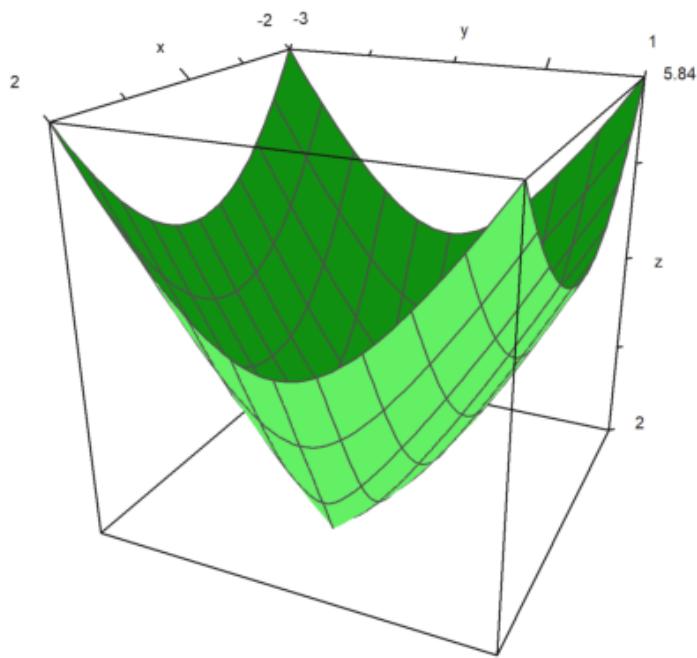
Sekarang kita lihat fungsi $MA+MB$ di mana A dan B adalah dua titik (tetap). Ini adalah "fakta yang terkenal" bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali AB minimum yang konstan pada segmen [AB]:

```
>B=[1, -1];
>function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2+(y-B[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



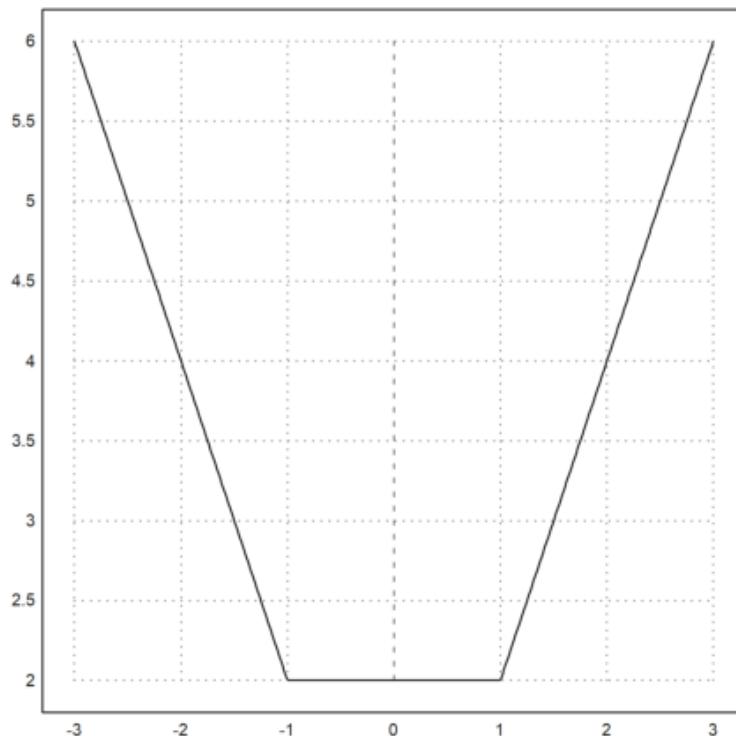
Grafiknya lebih menarik

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Pembatasan pada garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)", xmin=-3, xmax=3) :
```

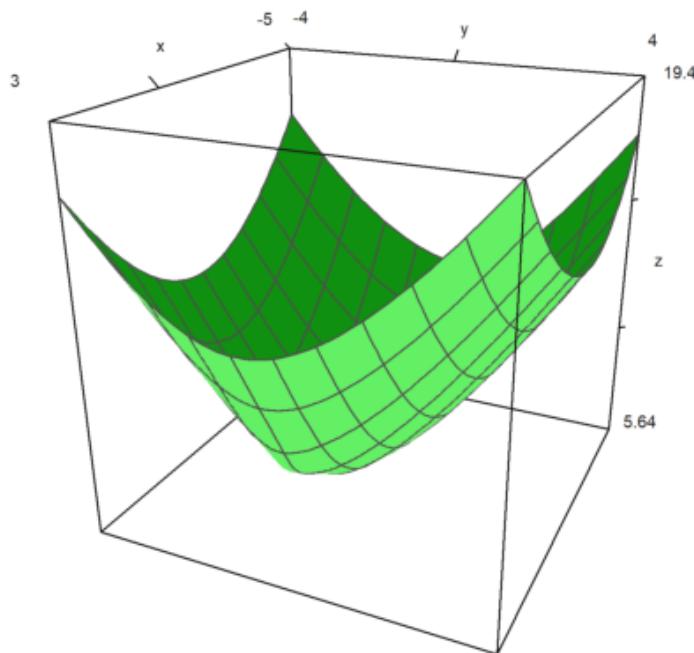


Sekarang, hal-hal menjadi kurang sederhana: Hal ini sedikit kurang dikenal bahwa $MA+MB+MC$ mencapai titik minimum pada satu titik bidang, tetapi untuk menentukannya tidak sesederhana itu:

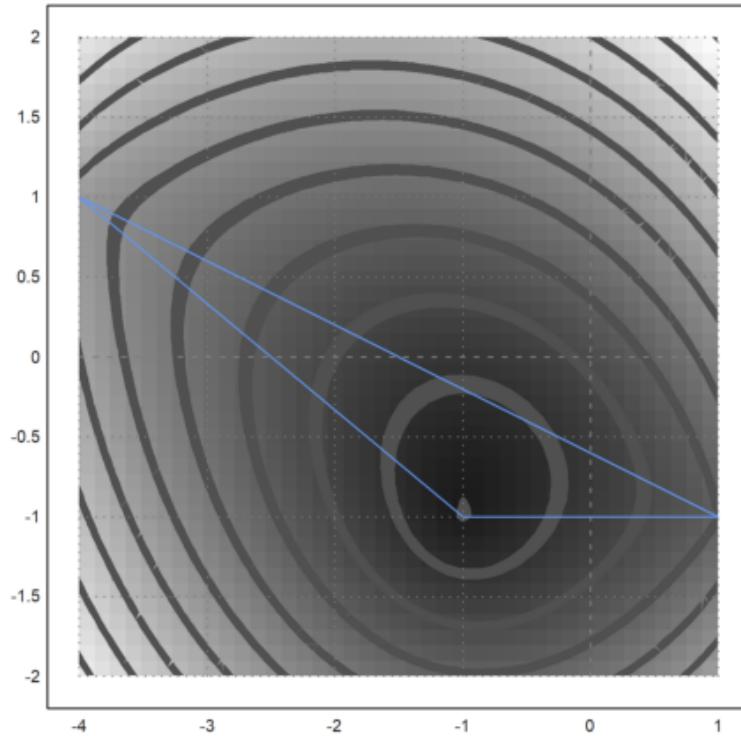
- 1) Jika salah satu sudut segitiga ABC lebih dari 120° (katakanlah di A), maka sudut minimum dicapai pada titik ini (katakanlah AB+AC).

Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
>insimg;
```

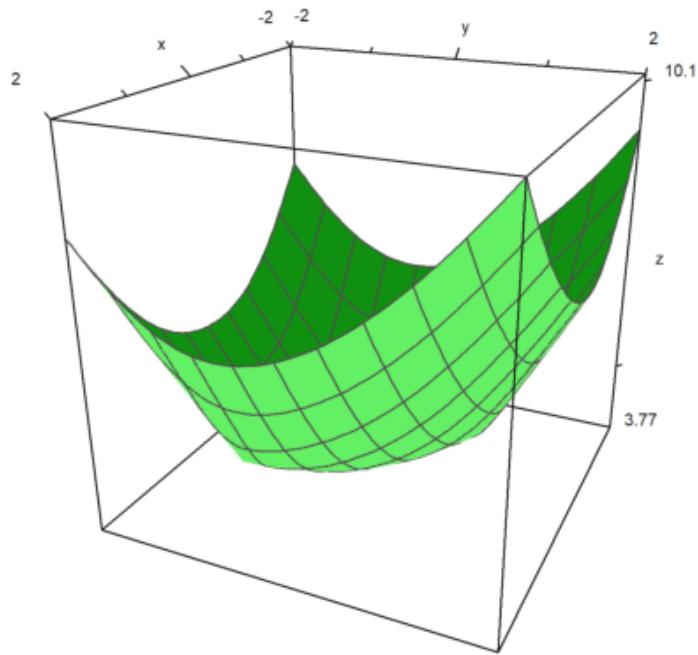


```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

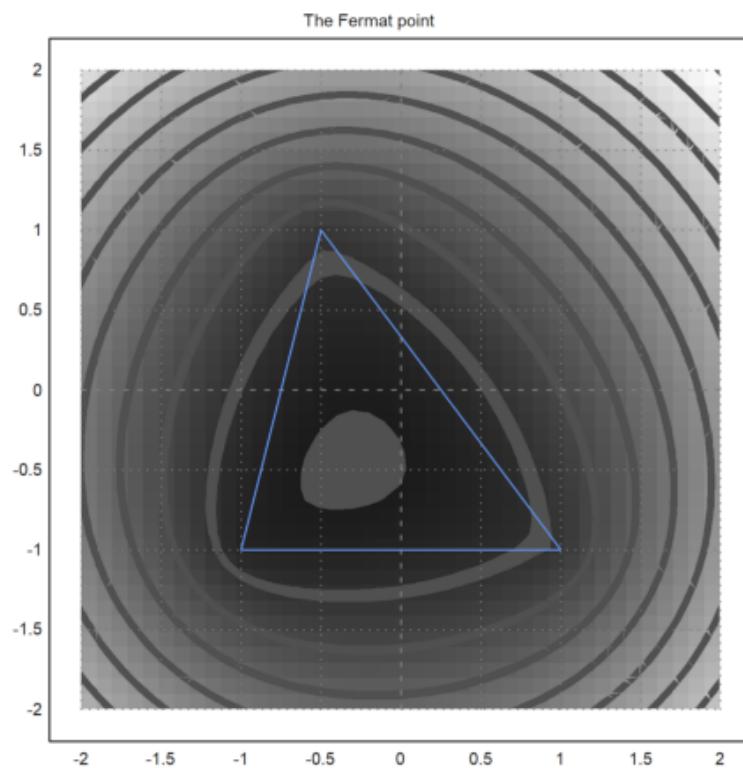


- 2) Tetapi jika semua sudut segitiga ABC kurang dari 120° , minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (masing-masing 120°):

```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):
```



```
>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point"
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```



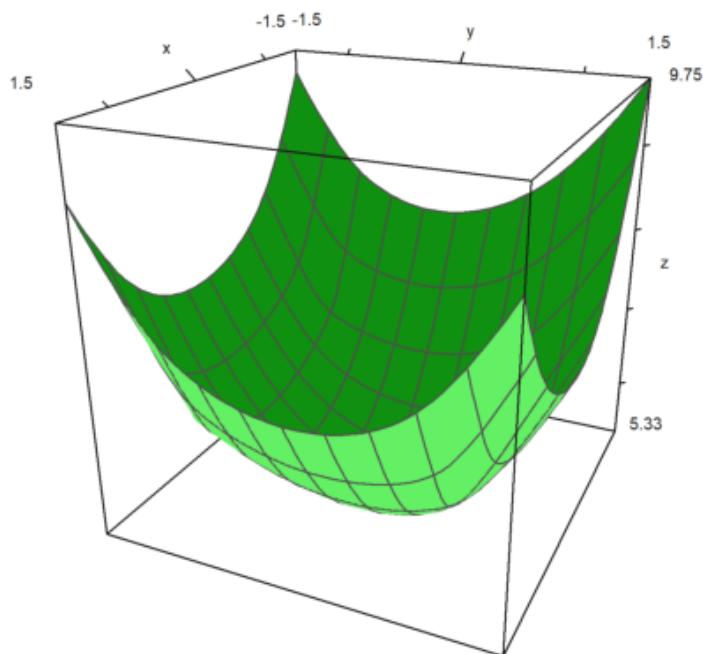
Merupakan kegiatan yang menarik untuk merealisasikan gambar di atas dengan perangkat lunak geometri; sebagai contoh, saya tahu sebuah perangkat lunak yang ditulis dalam bahasa Java yang memiliki instruksi "garis kontur"...

Semua hal di atas telah ditemukan oleh seorang hakim Perancis bernama Pierre de Fermat; ia menulis surat kepada para ahli dilet lainnya seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di bagian pajak pendapatan. Jadi titik unik F sedemikian rupa sehingga $FA+FB+FC$ minimal, disebut titik Fermat dari segitiga. Namun tampaknya beberapa tahun sebelumnya, Torricelli dari Italia telah menemukan titik ini sebelum Fermat menemukannya! Tradisi yang berlaku adalah mencatat titik F ini...

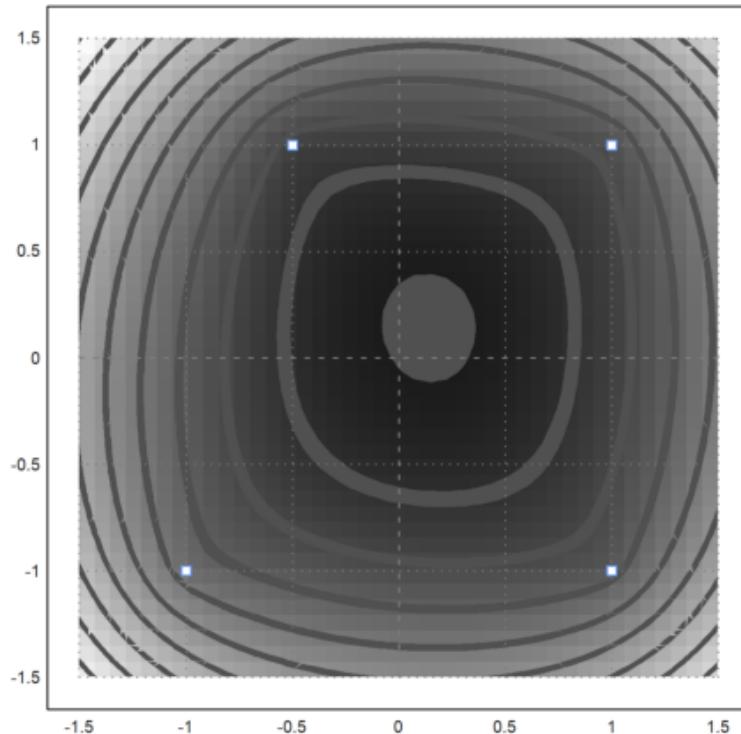
Empat poin

Langkah selanjutnya adalah menambahkan titik ke-4 D dan mencoba meminimumkan $MA+MB+MC+MD$, misalkan Anda adalah operator TV kabel dan ingin menemukan di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
```



```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
>insimg;
```



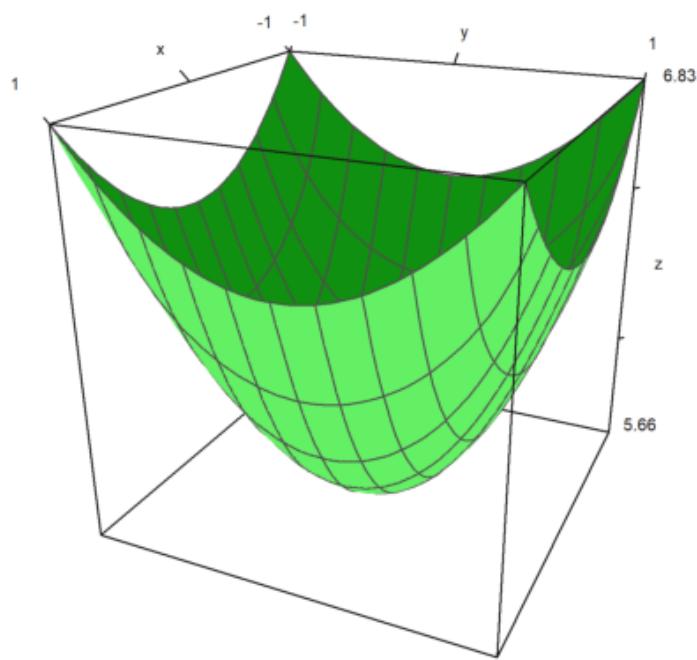
Masih ada nilai minimum dan tidak ada simpul A, B, C atau D yang tercapai:

```
>function f(x):=d4(x[1],x[2])
>neldermin("f",[0.2,0.2])
```

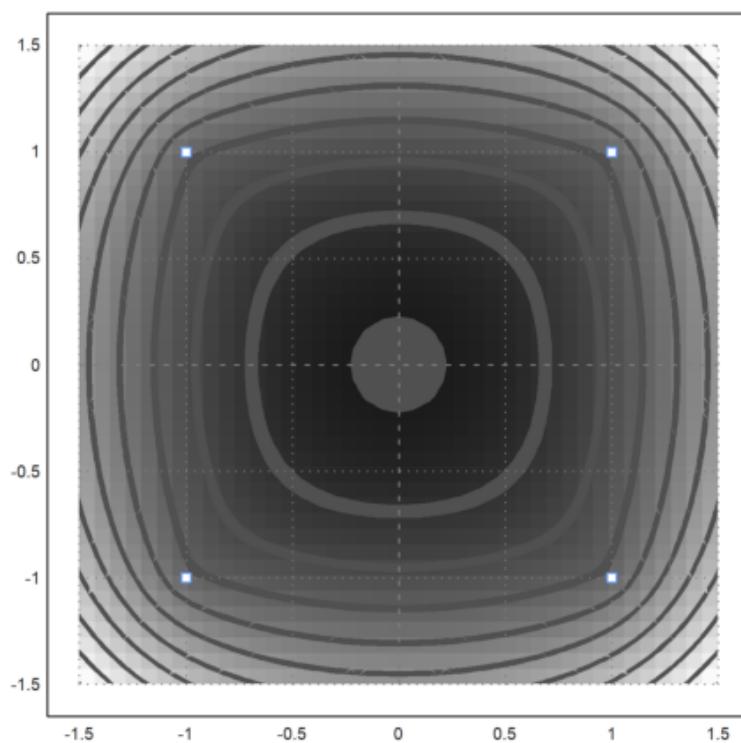
[0.142858, 0.142857]

Tampaknya dalam kasus ini, koordinat titik optimal adalah rasional atau mendekati rasional... Karena ABCD adalah sebuah bujur sangkar, kita berharap bahwa titik optimalnya adalah pusat dari 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;
```



Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan pvenGINE.exe pada jalur program.

Pertama, kita menghitung jari-jari bola. Jika Anda melihat gambar di bawah ini, Anda dapat melihat bahwa kita membutuhkan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file 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])
```

$[- a, - 1, 0]$

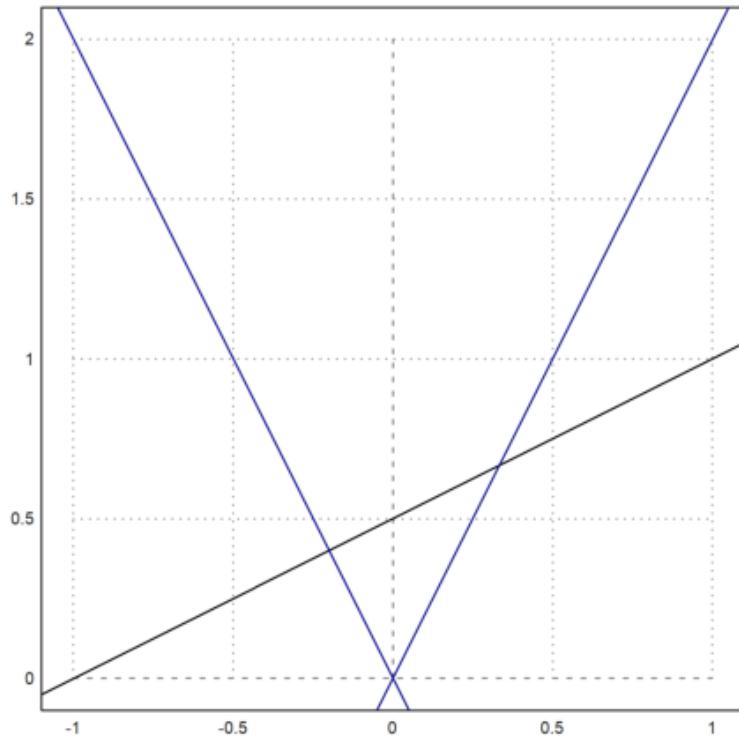
Kemudian baris ketiga.

```
>g &= lineThrough([-1,0],[1,1])
```

$[- 1, 2, 1]$

Kami merencanakan semuanya sejauh ini.

```
>setPlotRange(-1,1,0,2);
>color(black); plotLine(g(), "")
>a:=2; color(blue); plotLine(g1(), ""), plotLine(g2(), ""):
```



Sekarang, kita ambil titik umum pada sumbu y.

```
>P &= [0,u]
```

[0, u]

Hitung jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

$$\sqrt{\left(\frac{a^2 u}{a^2 + 1} - u\right)^2 + \frac{a^2 u^2}{(a^2 + 1)^2}}$$

Hitung jarak ke g

```
>d &= distance(P,projectToLine(P,g)); $d
```

$$\sqrt{\left(\frac{u+2}{5} - u\right)^2 + \frac{(2u-1)^2}{25}}$$

Dan temukan pusat kedua lingkaran, yang jaraknya sam

```
>sol &= solve(d1^2=d^2,u); $sol
```

$$\left[u = \frac{-\sqrt{5}\sqrt{a^2+1} + 2a^2 + 2}{4a^2 - 1}, u = \frac{\sqrt{5}\sqrt{a^2+1} + 2a^2 + 2}{4a^2 - 1} \right]$$

Ada dua solusi. Kami mengevaluasi solusi simbolis, dan menemukan kedua pusat, dan kedua jarak

```
>u := sol()
```

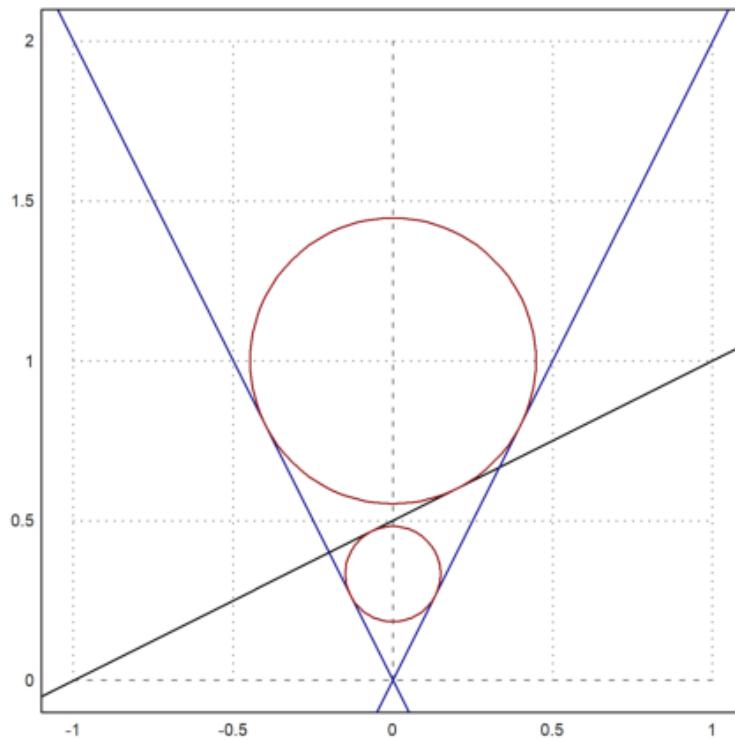
[0.333333, 1]

```
>dd := d()
```

[0.149071, 0.447214]

Plot lingkaran ke dalam gambar.

```
>color(red);
>plotCircle(circleWithCenter([0,u[1]],dd[1]), "");
>plotCircle(circleWithCenter([0,u[2]],dd[2]), "");
>insimg;
```



Plot dengan Povray

Selanjutnya kita plot semuanya dengan Povray. Perhatikan bahwa Anda mengubah perintah apa pun dalam urutan perintah Povray berikut ini, dan jalankan kembali semua perintah dengan Shift-Return.

Pertama, kita memuat fungsi povray.

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe

Kami menyiapkan pemandangan dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

Dan kerucutnya, transparan.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

Kami menghasilkan bidang 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));
```

Sekarang kita menghasilkan dua titik pada lingkaran, di mana bola menyentuh kerucut.

```
>function turnz(v) := return [-v[2],v[1],v[3]]
>P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
>writeln(povpoint(P1,povlook(yellow)));
>P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
>writeln(povpoint(P2,povlook(yellow)));
```

Kemudian, kita menghasilkan dua titik di mana bola-bola tersebut menyentuh bidang. Ini adalah fokus 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)));
```

Kami menghubungkan titik-titik dengan segmen garis.

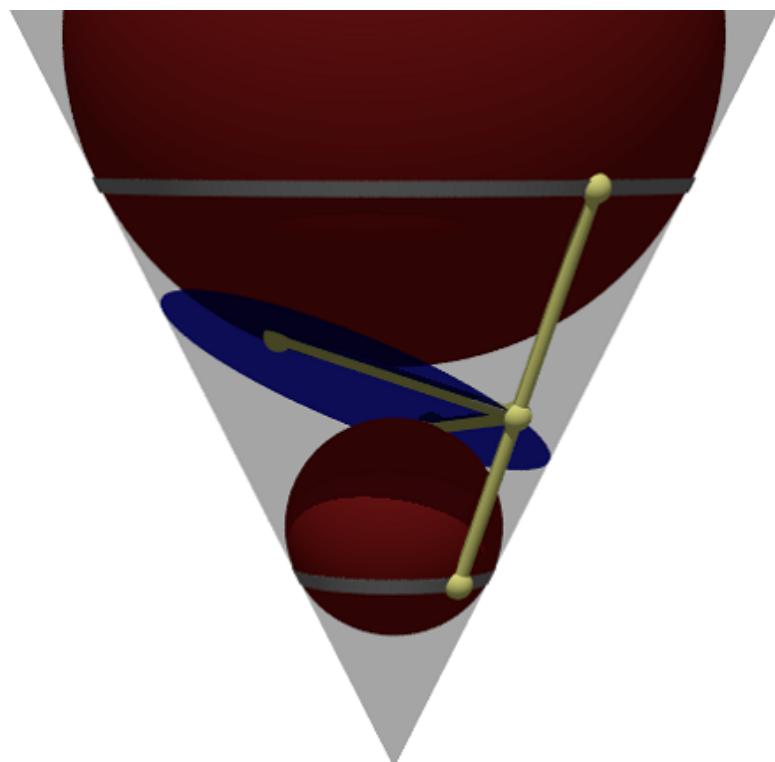
```
>writeln(povsegment(P1,P2,povlook(yellow)));
>writeln(povsegment(P5,P3,povlook(yellow)));
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Sekarang, kita menghasilkan pita abu-abu, di mana bola-bola menyentuh kerucut.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);
>pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1.01);
>writeln(povintersection([pcw,pc1],povlook(gray)));
>pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1.01);
>writeln(povintersection([pcw,pc2],povlook(gray))');
```

Mulai program Povray.

```
>povend();
```



Untuk mendapatkan Anaglyph ini, kita perlu memasukkan semuanya ke dalam fungsi `scene`. Fungsi ini akan digunakan dua kali nanti.

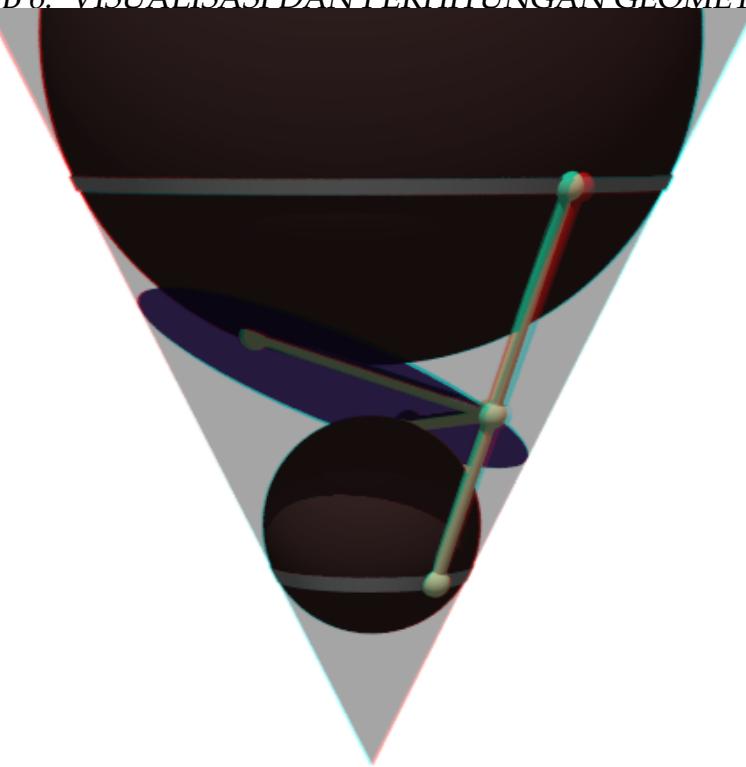
```
>function scene () ...
```

```
global a,u,dd,g,g1,defaultpointsize;
writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

```
writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
gp=g();
pc=povcone([0,0,0],0,[0,0,a],1,"");
vp=[gp[1],0,gp[2]]; dp=gp[3];
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow)));
writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow)));
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],0.01);
writeln(povintersection([pcw,pc1],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],0.01);
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction
```

You need red/cyan glasses to appreciate the following effect.

```
>povanaglyph("scene", zoom=11, center=[0,0,0.5], height=10°, angle=140°);
```



Contoh 8: Geometri Bumi

Dalam buku catatan ini, kita ingin melakukan beberapa komputasi bola. Fungsi-fungsi tersebut terdapat pada file "spherical.e" di dalam folder examples. Kita perlu memuat file tersebut terlebih dahulu

```
>load "spherical.e";
```

Untuk memasukkan posisi geografis, kita menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut ini 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 sposprint (cetak posisi bola

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
S 7°46.467' E 110°23.050'
```

Mari kita tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,2
```

```
S 7°34.333' E 110°49.683'  
S 6°59.050' E 110°24.533'
```

Pertama, kita menghitung vektor dari satu titik ke titik lainnya pada bola ideal. Vektor ini adalah [heading, jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang 7

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan
```

```
65°20'26.60''  
53.8945384608
```

Ini adalah perkiraan yang baik. Rutinitas berikut ini menggunakan perkiraan yang lebih baik lagi. Pada jarak yang begitu pendek, hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)-> " km" // perkiraan jarak FMIPA-Semarang
```

```
Commands must be separated by semicolon or comma!  
Found: // perkiraan jarak FMIPA-Semarang (character 32)  
You can disable this in the Options menu.  
Error in:  
esdist(FMIPA,Semarang)-> " km" // perkiraan jarak FMIPA-Semara ...  
^
```

Ada fungsi untuk judul, dengan mempertimbangkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

```
65.34°
```

Sudut segitiga melebihi 180° pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,
```

```
180°0'10.77''
```

Ini bisa digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, cara ini tidak akurat karena kesalahan pengurangan dalam asum-pi.

```
> (asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FMIPA-Solo-Sem
```

Commands must be separated by semicolon or comma!

Found: // perkiraan luas segitiga FMIPA-Solo-Semarang (character 32)

You can disable this in the Options menu.

Error in:

```
(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FM ...  
^
```

Ada sebuah fungsi untuk hal ini, yang menggunakan garis lintang rata-rata segitiga untuk menghitung radius bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi
```

2123.64310526 km²

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi arah dan jarak, keduanya dalam radian. Untuk mendapatkan sebuah vektor, kita menggunakan svector. Untuk menambahkan sebuah vektor ke sebuah 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 yang ideal. Hal yang sama di bumi

```
>sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(So
```

S 7°34.333' E 110°49.683'
S 7°34.333' E 110°49.683'

Mari kita beralih ke contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya)

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];  
>sposprint(Tugu), sposprint(Monas)
```

S 7°46.998' E 110°21.966'
S 6°10.500' E 106°48.717'

Menurut Google Earth, jaraknya adalah 429,66 km. Kami mendapatkan perkiraan yang bagus.

```
>esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Monas Jakarta
```

Commands must be separated by semicolon or comma!

Found: // perkiraan jarak Tugu Jogja - Monas Jakarta (character 32)

You can disable this in the Options menu.

Error in:

```
esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Mona ...  
^
```

Judulnya sama dengan yang dihitung di Google Earth

```
>degprint(esdir(Tugu,Monas))
```

$294^\circ 17' 2.85''$

Namun demikian, kita tidak lagi mendapatkan posisi target yang tepat, jika kita menambahkan arah dan jarak ke posisi semula. Hal ini terjadi, karena kita tidak menghitung fungsi inversi secara tepat, tetapi mengambil perkiraan radius bumi di sepanjang jalur

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

$S\ 6^\circ 10.500'\ E\ 106^\circ 48.717'$

Namun demikian, kesalahannya tidak besar.

```
>sposprint(Monas),
```

$S\ 6^\circ 10.500'\ E\ 106^\circ 48.717'$

Tentu saja, kita tidak bisa berlayar dengan arah yang sama dari satu tujuan ke tujuan lainnya, jika kita ingin mengambil jalur terpendek. Bayangkan, Anda terbang ke arah NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti arah yang konstan!

Perhitungan berikut ini menunjukkan bahwa kita melenceng jauh dari tujuan yang benar, jika kita menggunakan arah yang sama selama perjalanan.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh dari jarak tersebut, dengan menggunakan arah ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya jauh berbeda

```
>sposprint(p), skmpoint(esdist(p,Monas))
```

S $6^{\circ}11.250'$ E $106^{\circ}48.372'$
1.529km

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran lintang 30° , tetapi jalur yang lebih pendek yang dimulai 10° lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

79.69°

Namun, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi, kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kita sesuaikan pada 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); e
```

79.69°
 81.67°
 83.71°
 85.78°
 87.89°
 90.00°
 92.12°
 94.22°
 96.29°
 98.33°

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti judul yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

0.203km

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan arah setiap 1/100 dari total jarak dari Tugu ke Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...
> loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;
>skmprint(esdist(p,Monas))
```

0.000km

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS di sepanjang Bundaran HI menuju 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'

Kami menulis sebuah fungsi, yang memetakan bumi, dua posisi, dan posisi di antaranya

```
>function testplot ...
```

```
useglobal;
plotearth;
plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");
plotposline(v);
endfunction
```

Now plot everything.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4) :
```

tau 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.

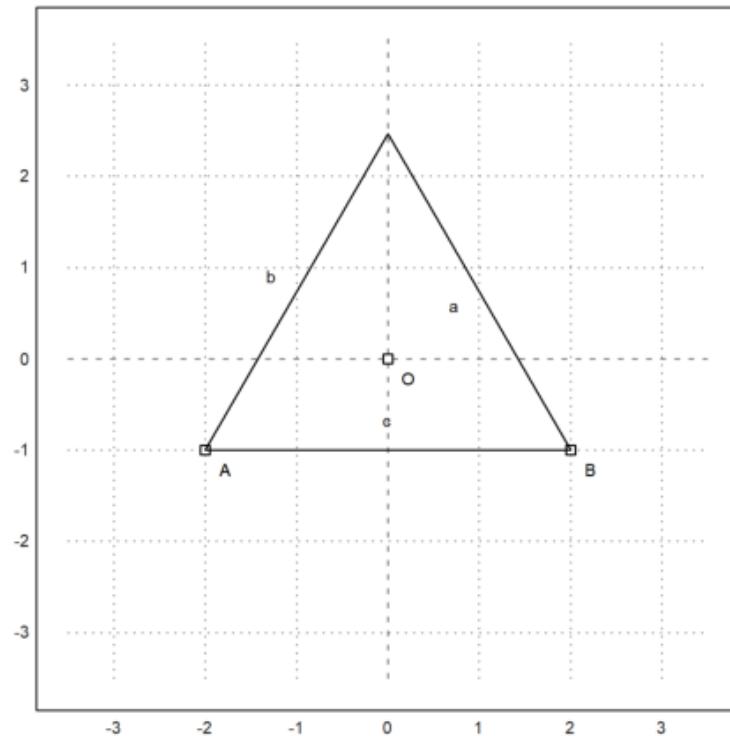
Petunjuk:

- 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.

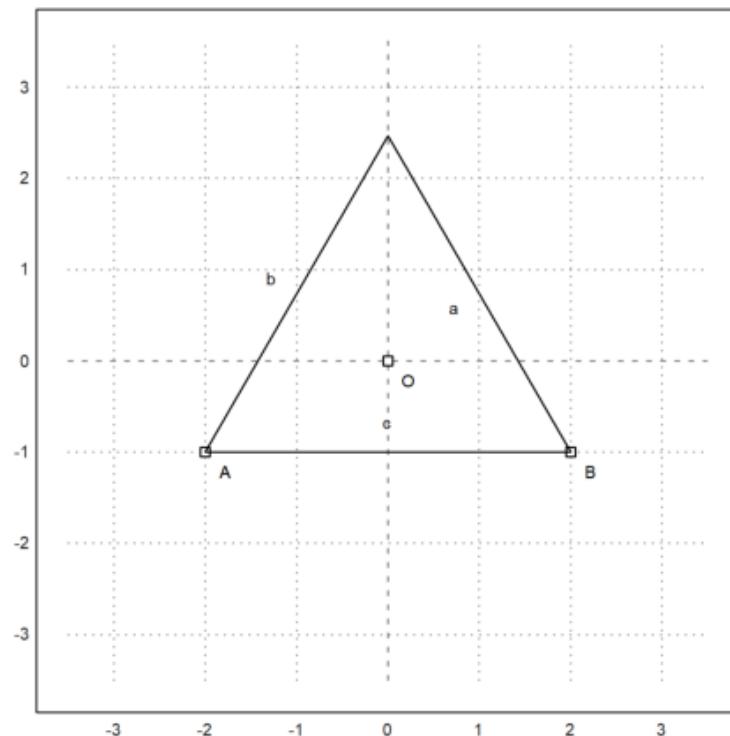
```
>load geometry
```

Numerical and symbolic geometry.

```
>setPlotRange(-3.5,3.5,-3.5,3.5);
>O=[0,0]; plotPoint(O,"O");
>A=[-2,-1]; plotPoint(A,"A");
>B=[2,-1]; plotPoint(B,"B");
>C=[0,2*3^0.5-1]; plotPoint(C,"C");
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
>plotSegment(C,A,"b");
>aspect(1);
```



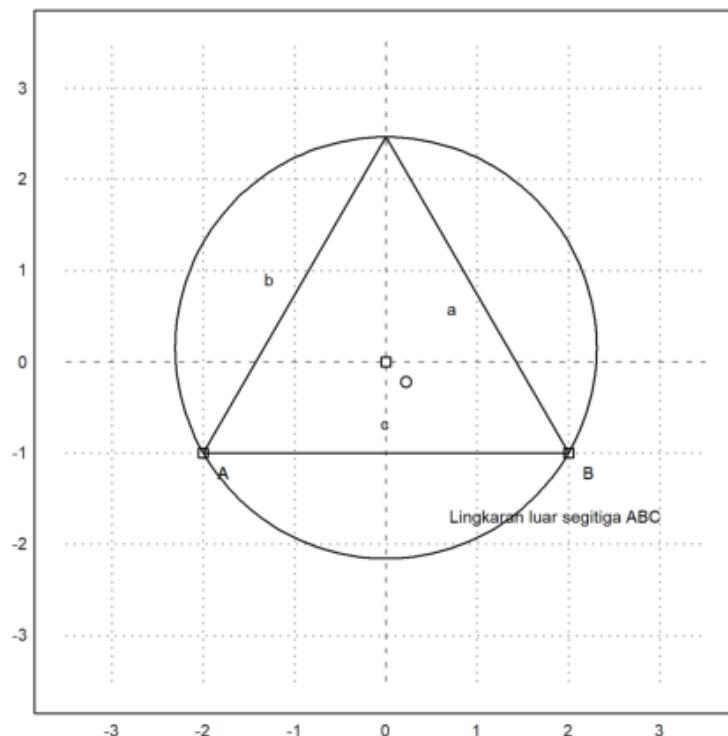
```
>c=circleThrough(A,B,C) :
```



```
>R=getCircleRadius(c);
>O=getCircleCenter(c)
```

[0, 0.154701]

```
>plotCircle(c,"Lingkaran luar segitiga ABC");
```

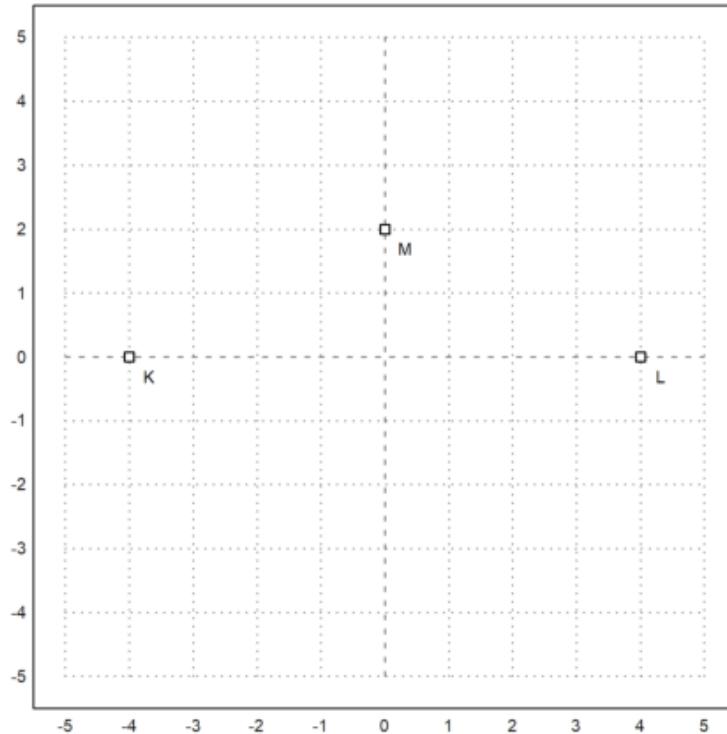


2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

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.

```
>setPlotRange(5);
>K=[-4,0]; L=[4,0] ; M=[0,2];
>plotPoint(K,"K"); plotPoint(L,"L"); plotPoint(M,"M") :
```



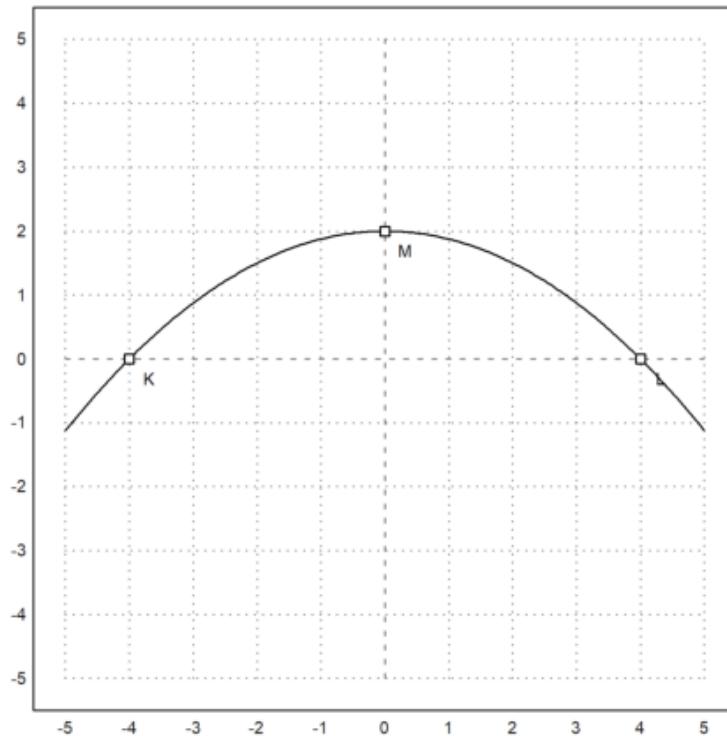
```
>sol &= solve([16*a+8*b=-c, 16*a-8*b=-c, c=2], [a, b, c])
```

$$\left[\left[a = -\frac{1}{8}, b = 0, c = 2 \right] \right]$$

```
>function y&=-1/8*(x^2)-0*x+2
```

$$\frac{x^2}{8} - \frac{1}{8}$$

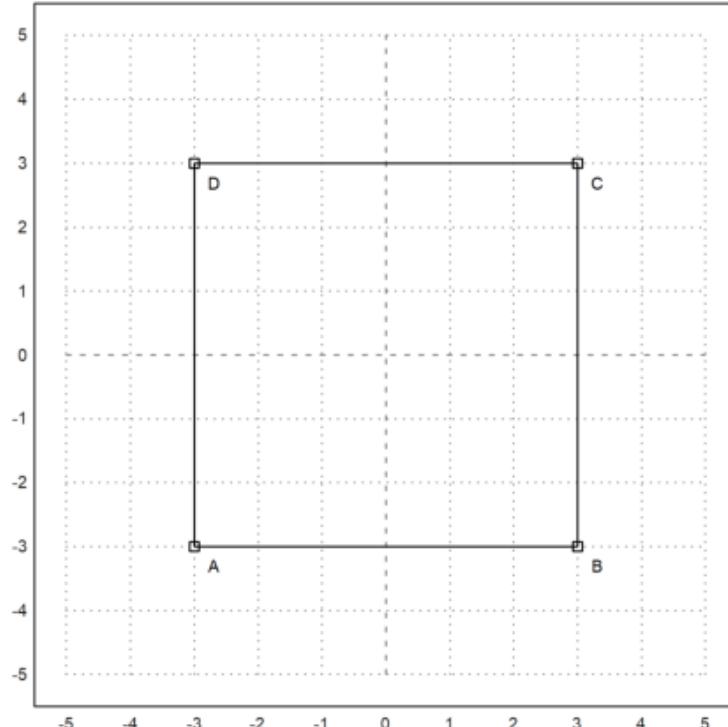
```
>plot2d("-1/8*(x^2)-0*x+2", -5, 5, -5, 5); ...
>plotPoint(K, "K"); plotPoint(L, "L"); plotPoint(M, "M");
```



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.

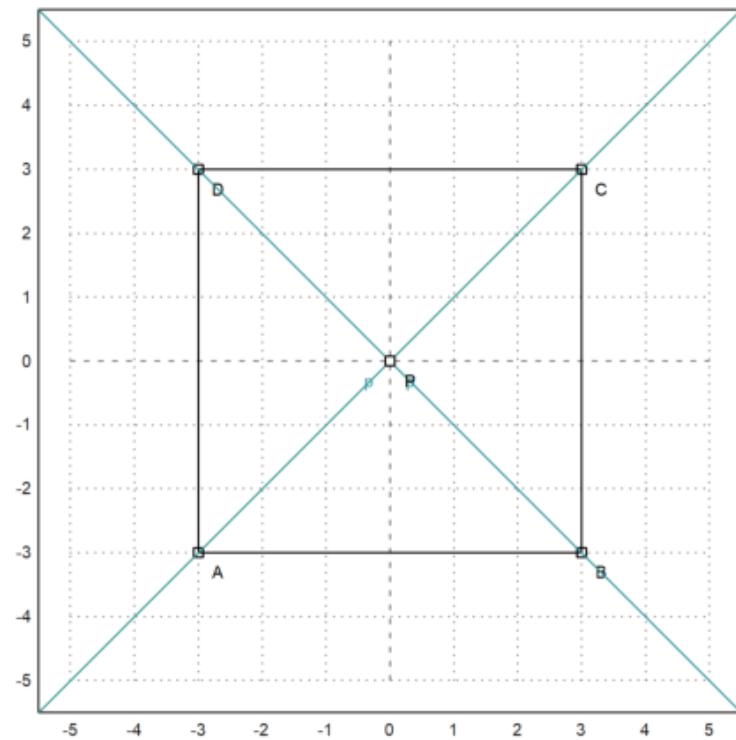
```
>setPlotRange(5);
>A=[-3,-3]; B=[3,-3] ; C=[3,3]; D=[-3,3];
>plotPoint(A, "A"); plotPoint(B, "B"); plotPoint(C, "C"); plotPoint(D, "D");
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,D,""); plotSegment
```



```
>l=angleBisector(A,B,C);
>g=angleBisector(B,C,D);
>P=lineIntersection(l,g)
```

[0, 0]

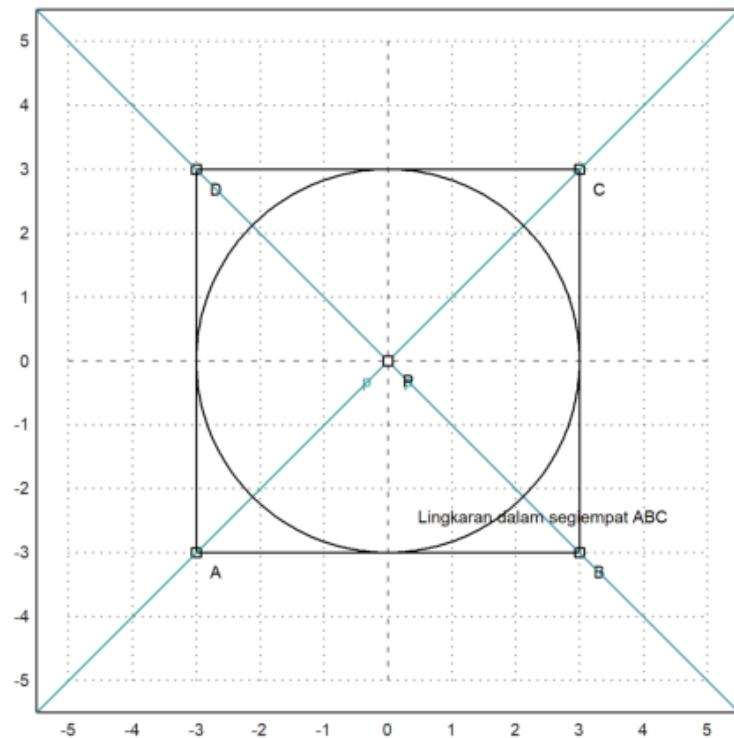
```
>color(5); plotLine(l); plotLine(g); color(1);
>plotPoint(P, "P"):
```



```
>r=norm(P-projectToLine(P, lineThrough(A,B)) )
```

3

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat ABC"):
```



Dari gambar di atas, terlihat bahwa sisi-sisinya merupakan garis singgung lingkaran yang sama yaitu lingkaran dalam segiempat.

Kemudian akan ditunjukkan bahwa hasil kali panjang sisi-sisi yang berhadapan sama.

```
>AB=norm(A-B) // panjang sisi AB
```

6

```
>BC=norm(B-C) // panjang sisi ABC
```

6

```
>CD=norm(C-D) // panjang sisi CD
```

6

```
>DA=norm(D-A) // panjang sisi DA
```

6

```
>AB . CD
```

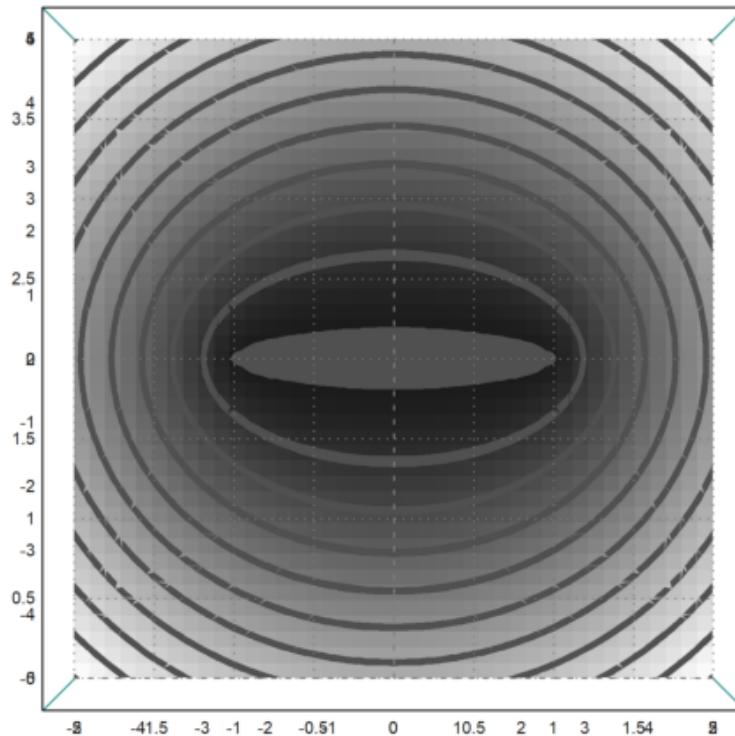
36

```
>DA . BC
```

36

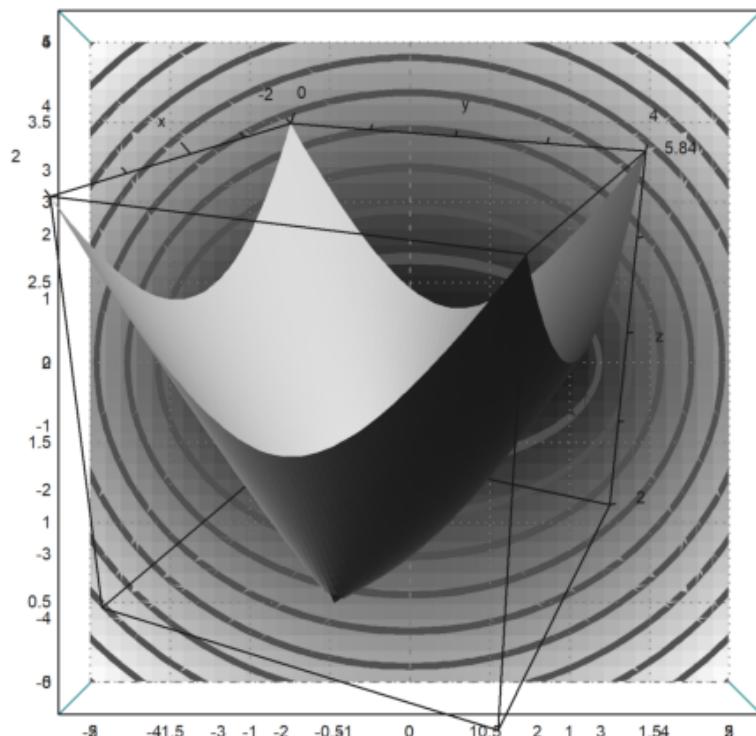
5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

```
>P=[-1,2]; Q=[1,2];
>function d1(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)
>function d2(x,y):=d1(x,y)+sqrt((x-Q[1])^2+(y-Q[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=0,ymax=4,hue=1):
```



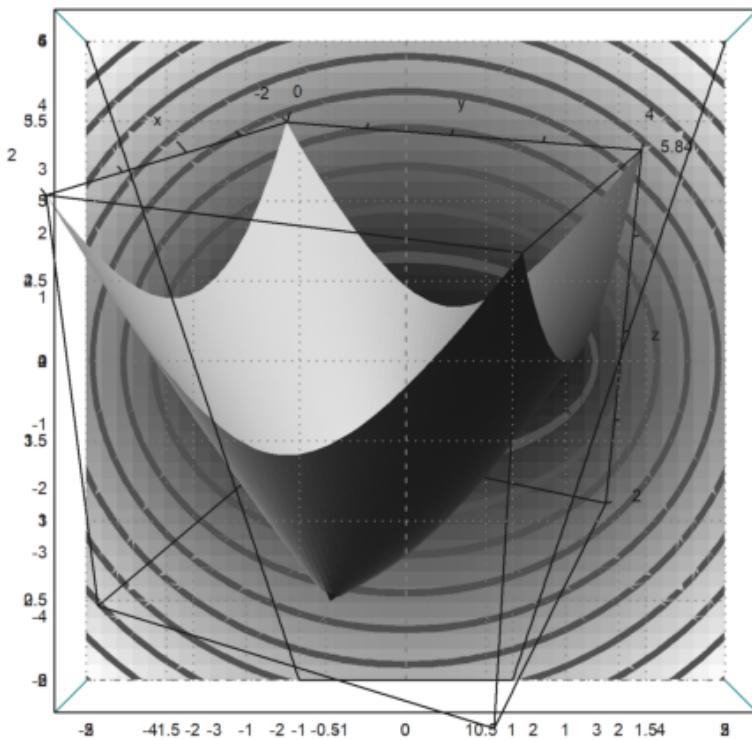
Grafik yang lebih menarik

```
>plot3d("d2", xmin=-2, xmax=2, ymin=0, ymax=4, hue=1) :
```



Batasan garis PQ

```
>plot2d("abs(x+1)+abs(x-1)", xmin=-3, xmax=3) :
```



BAB 7

EMT untuk Statistika

Dalam buku catatan ini, kami mendemonstrasikan plot statistik utama, tes dan distribusi dalam Euler.

Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar statistik. Jadi, Anda mungkin memerlukan latar belakang untuk memahami detailnya.

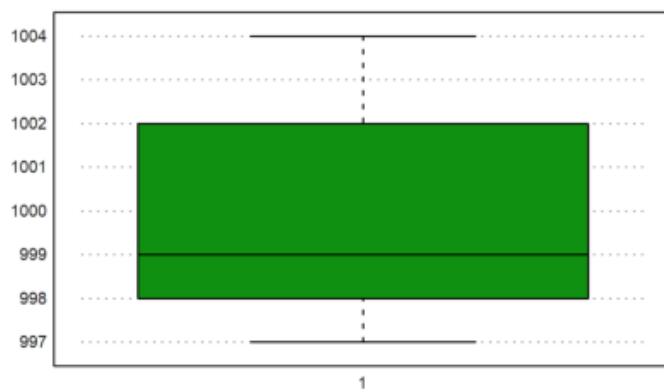
Asumsikan pengukuran berikut ini. Kita ingin menghitung nilai rata-rata dan deviasi standar yang diukur

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...  
>median(M), mean(M), dev(M),
```

```
999  
999.9  
2.72641400622
```

Kita dapat memplot plot kotak dan kumis untuk data tersebut. Dalam kasus kami, tidak ada pencilan

```
>aspect(1.75); boxplot(M):
```



Kami menghitung probabilitas bahwa suatu nilai lebih besar dari 1005, dengan mengasumsikan nilai yang diukur dan distribusi normal.

Semua fungsi untuk distribusi di Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

$$\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} dt.$$

Kami mencetak hasilnya dalam % dengan akurasi 2 digit menggunakan fungsi cetak.

```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
```

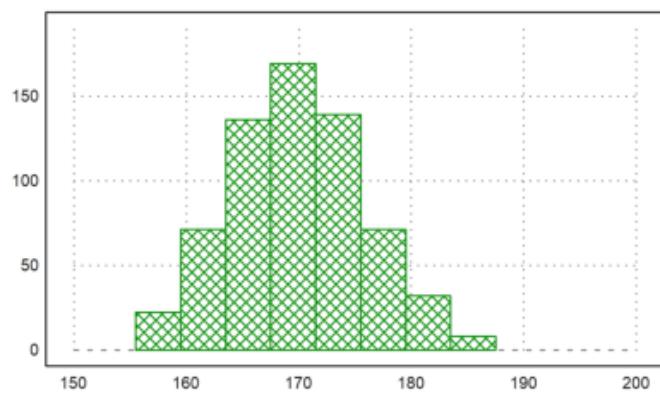
3.07 %

Untuk contoh berikutnya, kami mengasumsikan jumlah pria berikut dalam rentang ukuran tertentu.

```
>r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Berikut ini adalah plot distribusi

```
>plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="/"):
```



Kita dapat memasukkan data mentah tersebut ke dalam tabel. Tabel adalah sebuah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal rentang, akhir rentang, jumlah pria dalam rentang. Tabel dapat dicetak dengan header. Kami menggunakan vektor string untuk mengatur header.

```
>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 membutuhkan nilai rata-rata dan statistik lain dari ukuran, kita perlu menghitung titik tengah rentang. 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 header 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 akan lebih mudah, untuk melipat 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 sampel dengan frekuensi yang diberikan

```
>{m,d}=meandev(M,v); m, d,
```

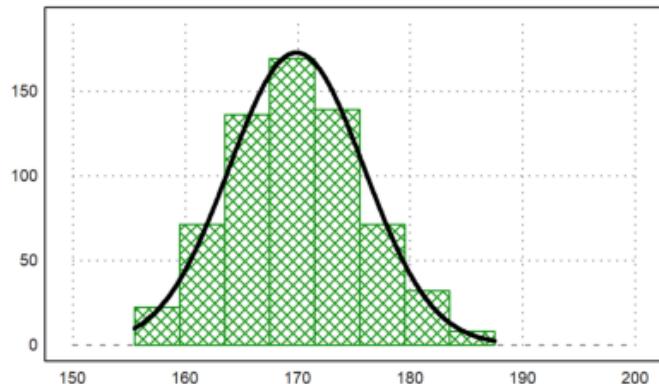
```
169.901234568
5.98912964449
```

Mari kita tambahkan distribusi normal dari nilai-nilai tersebut ke dalam diagram batang di atas. Rumus untuk distribusi normal dengan rata-rata m dan deviasi standar d adalah:

$$y = \frac{1}{d\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2d^2}}.$$

Karena nilainya antara 0 dan 1, untuk memplotnya pada plot batang, maka harus dikalikan dengan 4 kali jumlah data.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...
> xmin=min(r),xmax=max(r),thickness=3,add=1) :
```



Tabel

Dalam direktori buku catatan ini, Anda akan menemukan file dengan tabel. Data tersebut merupakan hasil survei. Berikut adalah empat baris pertama dari file tersebut. Data berasal dari sebuah buku online berbahasa 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 berisi 7 kolom angka atau token (string). Kita ingin membaca tabel tersebut dari file. Pertama, kita menggunakan terjemahan kita sendiri untuk token-token tersebut. Untuk itu, kita mendefinisikan set token. Fungsi `strtokens()` mendapatkan vektor string token dari string yang diberikan.

```
>mf := ["m", "f"]; yn := ["y", "n"]; ev := strtokens("g vg m b vb");
```

Sekarang kita membaca tabel dengan terjemahan ini. Argumen tok2, tok4, dan lain-lain adalah terjemahan dari kolom-kolom tabel. Argumen-argumen ini tidak ada dalam daftar parameter readtable(), jadi Anda harus menyediakannya dengan ":=".

```
>{MT, hd} = readtable("table.dat", tok2 := mf, tok4 := yn, tok5 := ev, tok7 := yn);
>load over statistics;
```

Untuk mencetak, kita perlu menentukan set token yang sama. Kami mencetak empat baris pertama saja

```
>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

Titik-titik "." mewakili nilai yang tidak tersedia. Jika kita tidak ingin menentukan token untuk terjemahan sebelumnya, kita hanya perlu menentukan kolom mana yang berisi token dan bukan angka.

```
>ctok = [2, 4, 5, 7]; {MT, hd, tok} = readtable("table.dat", ctok = ctok);
```

Fungsi readtable() sekarang mengembalikan satu set token.

```
>tok
```

```
m
n
f
y
g
vg
```

Tabel berisi entri dari file dengan token yang diterjemahkan ke angka. String khusus NA="." ditafsirkan sebagai "Not Available", dan mendapatkan 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 ini adalah isi tabel dengan angka yang tidak 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 kenyamanan, Anda dapat meletakkan output dari readtable() ke dalam sebuah dafatar.

```
>Table={{readtable("table.dat", ctok=ctok)} };
```

Dengan menggunakan kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dll. atau menggunakan daftar Tabel.

```
>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	.	n
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 kolom dari tabel, melewatkkan setiap baris dengan nilai NAN ("." dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

```
>{c,i}=tablecol(MT,[5,6]);
```

Kita dapat menggunakan ini untuk mengekstrak kolom dari tabel untuk 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 Tabel dalam kasus ini

```
>MT=Table[1];
```

Tentu saja, kita juga dapat menggunakannya untuk menentukan nilai rata-rata kolom atau nilai statistik lainnya.

```
>mean(tablecol(MT, 6))
```

2.175

Fungsi getstatistics() mengembalikan elemen-elemen dalam sebuah vektor, dan jumlahnya. Kita menerapkannya pada nilai "m" dan "f" pada kolom kedua tabel kita.

```
>{xu, count}=getstatistics(tablecol(MT, 2)); xu, count,
```

```
[1, 3]
[12, 13]
```

Kita bisa mencetak hasilnya dalam tabel baru

```
>writetable(count', labr=tok[xu])
```

m	12
f	13

Fungsi `selectable()` mengembalikan sebuah tabel baru dengan nilai dalam satu kolom yang dipilih dari vektor indeks. Pertama, kita mencari indeks dari dua nilai kita dalam tabel `token`.

```
>v:=indexof(tok, ["g", "vg"])
```

```
[5, 6]
```

Sekarang kita dapat memilih baris dari tabel, yang memiliki salah satu nilai dalam `v` di baris ke-5.

```
>MT1:=MT [selectrows(MT, 5, v)]; i:=sortedrows(MT1, 5);
```

Sekarang kita dapat mencetak tabel, dengan nilai yang diekstrak dan diurutkan di kolom ke-5.

```
>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, kita ingin menghubungkan dua kolom tabel. Jadi kita mengekstrak kolom 2 dan 4 dan mengurutkan tabel.

```
>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
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
f	Y

```
>MT24=tablecol(MT, [2,4]); ...
```

Dengan getstatistics(), kita juga dapat menghubungkan hitungan dalam dua kolom tabel satu sama lain.

```
>{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 sebuah file.

```
>filename="test.dat"; ...
>writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
>{MT2,hd,tok2,hdr}=readtable(filename,>clabs,>rlabs); ...
```

Kemudian kita dapat membaca tabel dari file tersebut.

```
>writetable(MT2, labr=hdr, labc=hd)
```

	n	Y
m	7	5
f	1	12

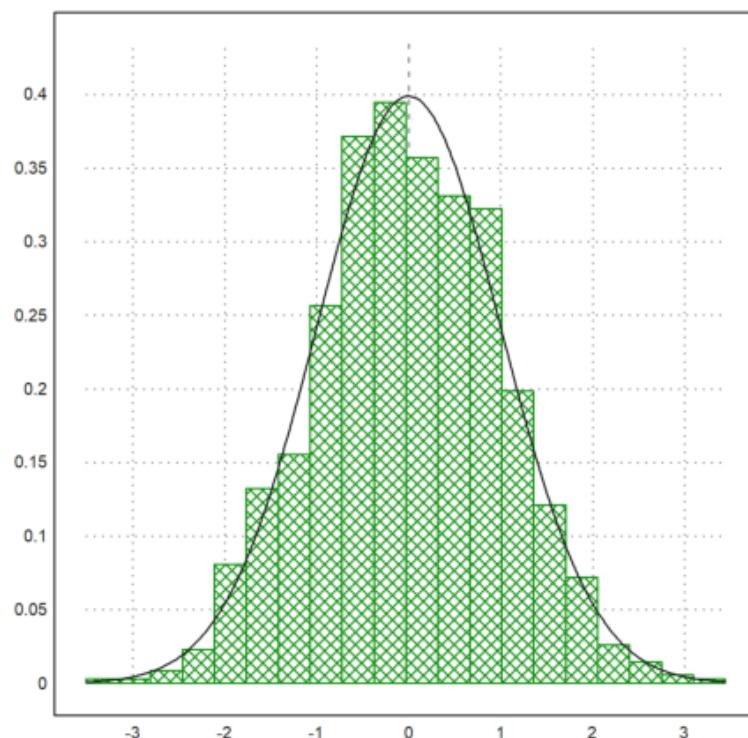
Dan hapus file tersebut.

```
>fileremove(filename);
```

Distribusi

Dengan plot2d, ada metode yang sangat mudah untuk memplot distribusi data eksperimen.

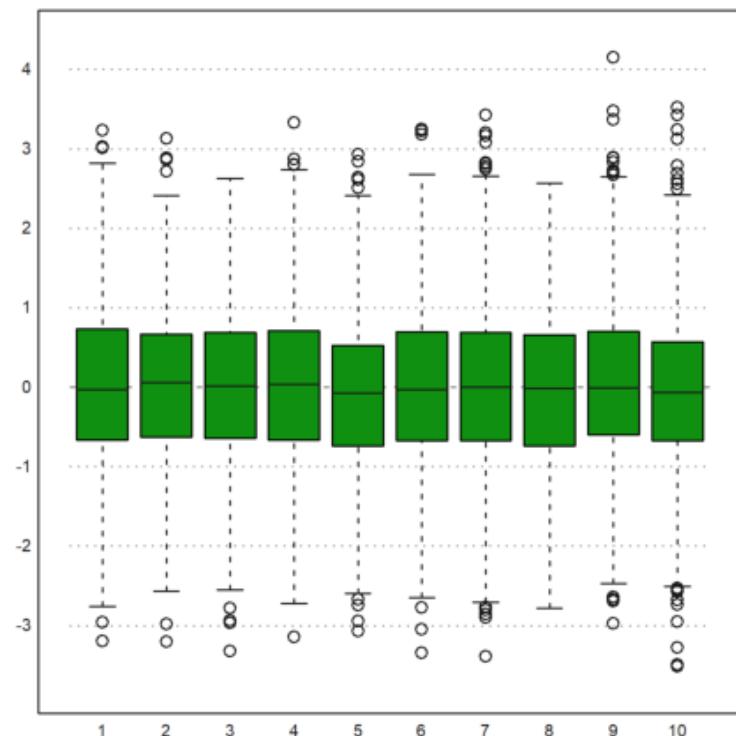
```
>p=normal(1,1000); //1000 random normal-distributed sample p
>plot2d(p,distribution=20,style="\\""); // plot the random sample p
>plot2d("qnormal(x,0,1)",add=1): // add the standard normal distribution pl
```



Perhatikan perbedaan antara plot batang (sampel) dan kurva normal (distribusi sesungguhnya). Masukkan kembali ketiga perintah tersebut untuk melihat hasil pengambilan sampel yang lain.

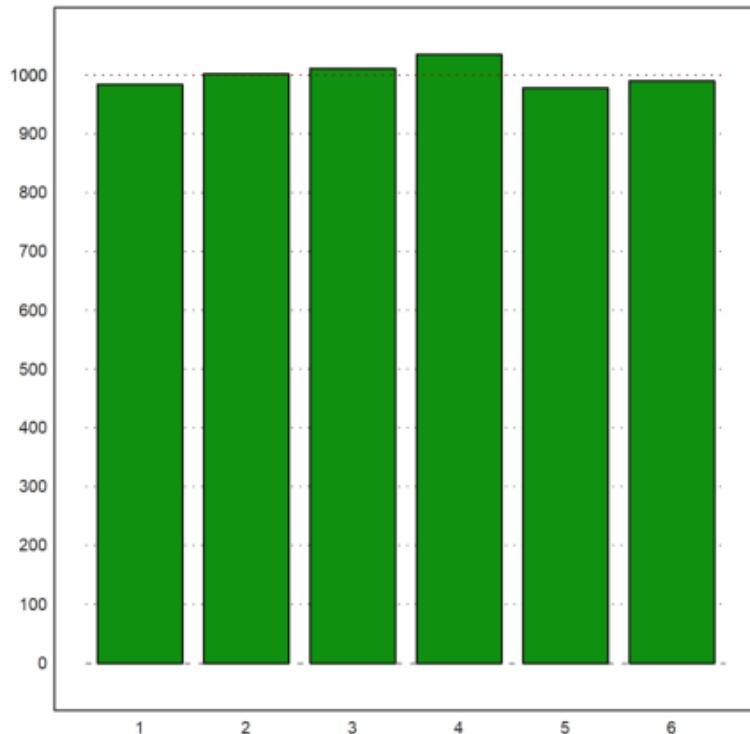
Berikut ini adalah perbandingan 10 simulasi dari 1000 nilai terdistribusi normal dengan menggunakan apa yang disebut plot kotak. Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, serta pencilan.

```
>p=normal(10,1000); boxplot(p):
```



Untuk menghasilkan bilangan bulat acak, Euler memiliki intrandom. Mari kita simulasikan pelemparan dadu dan plot distribusinya. Kita menggunakan fungsi `getmultiplicities(v,x)`, yang menghitung seberapa sering elemen-elemen `v` muncul di dalam `x`. Kemudian kita memplot hasilnya menggunakan `columnsplot()`.

```
>k=intrandom(1,6000,6); ...
>columnsplot(getmultiplicities(1:6,k)); ...
>ygrid(1000,color=red):
```



Walaupun `intrandom(n,m,k)` mengembalikan bilangan bulat yang terdistribusi secara seragam dari 1 sampai k, adalah mungkin untuk menggunakan distribusi bilangan bulat yang lain 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]
```

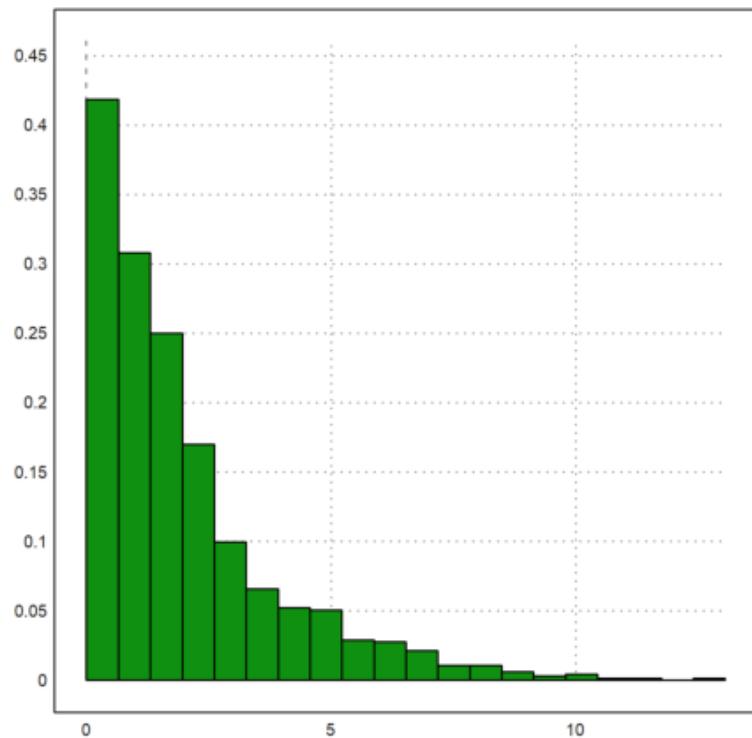
Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Lihatlah referensi. Sebagai contoh, kita mencoba distribusi eksponensial. Variabel acak kontinu X dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

dengan parameter

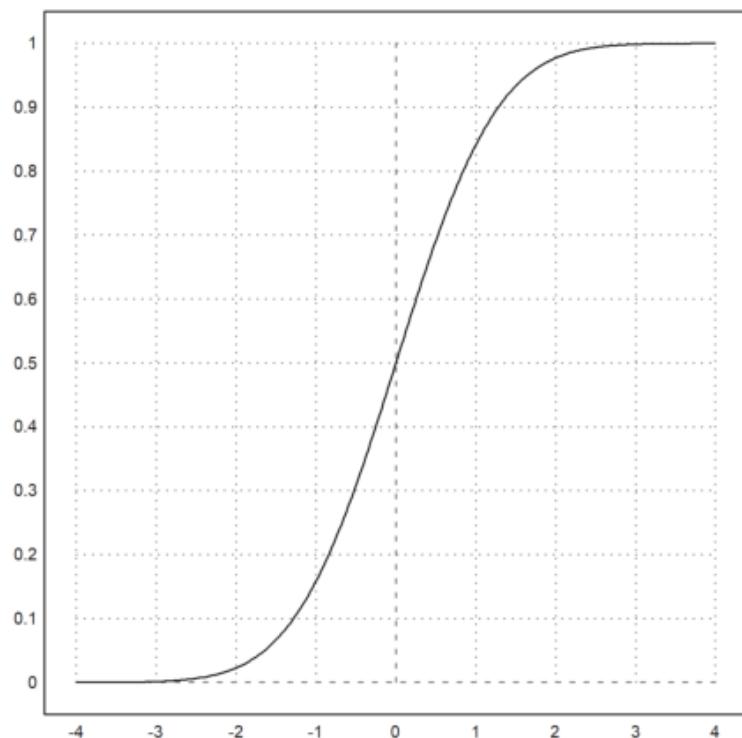
$$\lambda = \frac{1}{\mu}, \quad \mu \text{ adalah rata rata, dan dilambangkan dengan } X \sim \text{Eksponensial}(\lambda).$$

```
>plot2d(randexponential(1,1000,2),>distribution):
```



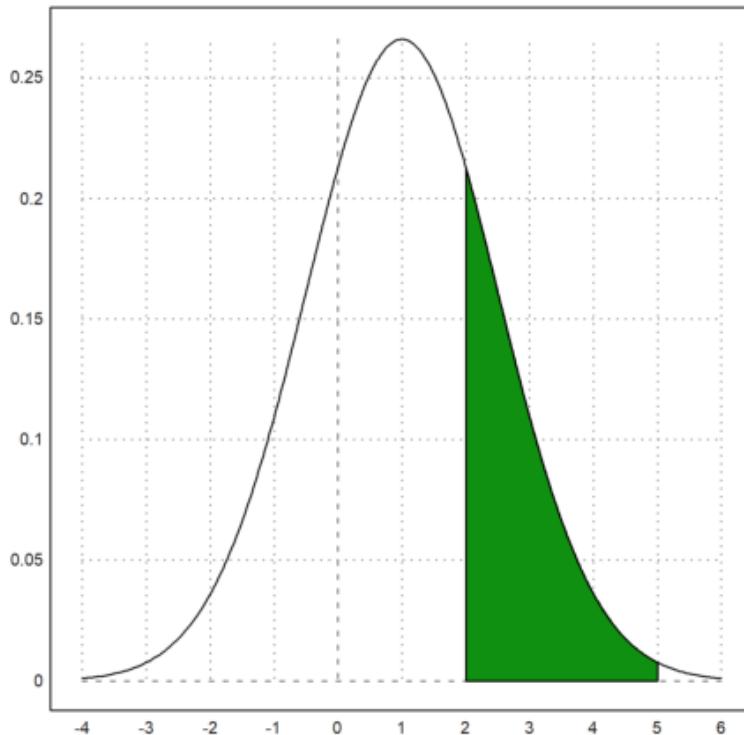
Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan kebalikannya

```
>plot2d("normaldis", -4, 4):
```



Berikut ini adalah salah satu cara untuk memplot 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 untuk berada di area hijau adalah sebagai berikut

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

0.248662156979

Hal ini dapat dihitung secara numerik dengan integral berikut ini

$$\int_2^5 \frac{1}{1.5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-1}{1.5})^2} dx.$$

```
>5gauss("qnormal(x,1,1.5)",2,5)
```

0.248662156979

Mari kita bandingkan distribusi binomial dengan distribusi normal dengan rata-rata dan deviasi yang sama. Fungsi invbindis() menyelesaikan interpolasi linier antara nilai bilangan bulat.

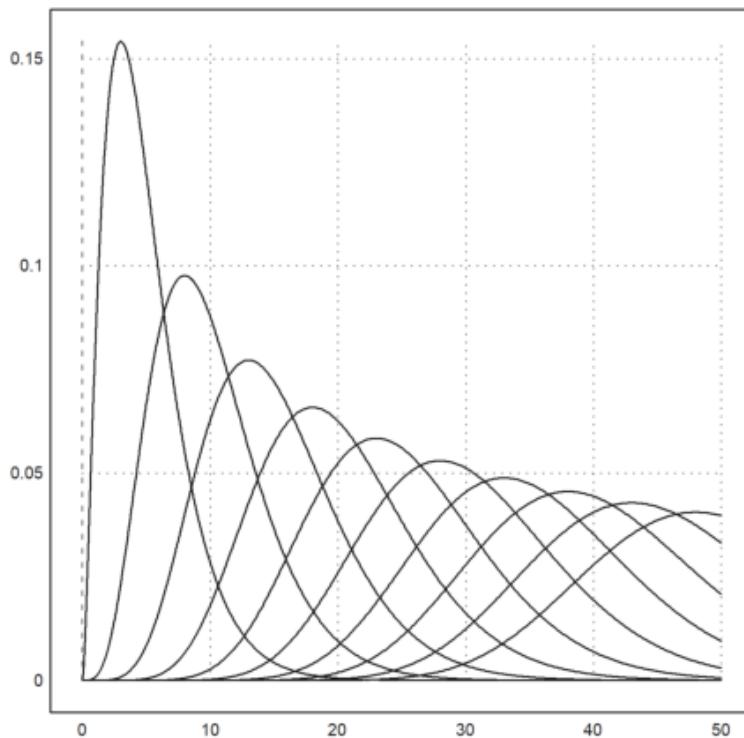
```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

525.516721219

526.007419394

Fungsi qdis() adalah densitas dari distribusi chi-square. Seperti biasa, Euler memetakan vektor ke fungsi ini. Dengan demikian kita mendapatkan plot semua distribusi chi-kuadrat dengan derajat 5 hingga 30 dengan mudah dengan cara berikut.

```
>plot2d("qchidis(x,(5:5:50)')",0,50):
```



Euler memiliki fungsi-fungsi yang akurat untuk mengevaluasi distribusi-distribusi. Mari kita periksa chidis() dengan sebuah integral. Penamaannya diusahakan untuk konsisten. Contohnya,

- distribusi chi-square adalah chidis(),
- fungsi kebalikannya adalah invchidis(),
- densitasnya adalah qchidis().

Pelengkap distribusi (ekor atas) adalah chicdis().

```
>chidis(1.5,2), integrate("qchidis(x,2)",0,1.5)
```

```
0.527633447259
0.527633447259
```

Distribusi Diskrit

Untuk menentukan distribusi diskrit Anda sendiri, Anda dapat menggunakan metode berikut. Pertama, kita tetapkan fungsi distribusinya.

```
>wd = 0 | ((1:6)+[-0.01,0.01,0,0,0,0])/6
```

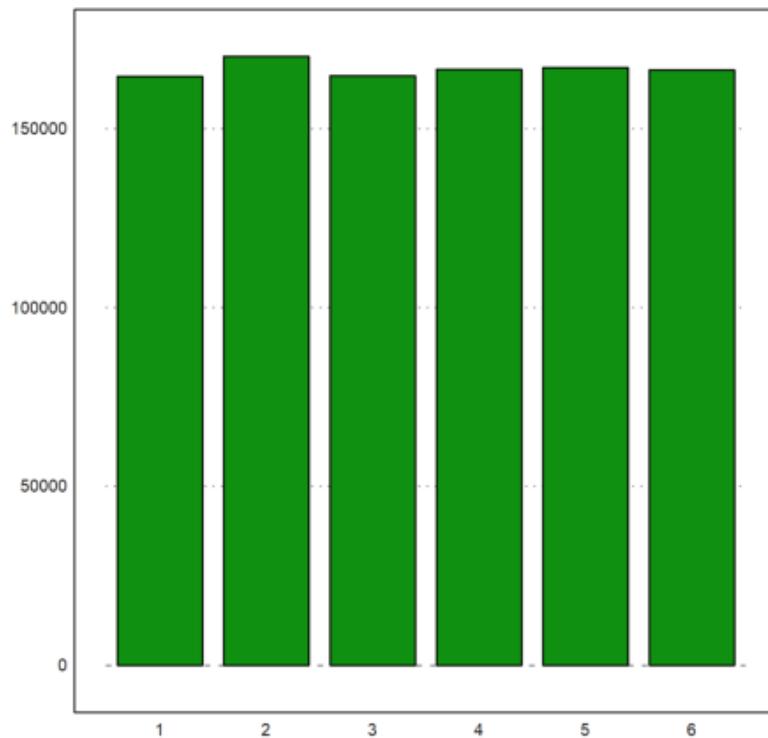
```
[0, 0.165, 0.335, 0.5, 0.666667, 0.833333, 1]
```

Artinya, dengan probabilitas $wd[i+1]-wd[i]$ kita menghasilkan nilai acak i. Ini hampir merupakan distribusi yang seragam. Mari kita definisikan sebuah generator bilangan acak untuk ini. Fungsi $find(v,x)$ menemukan nilai x dalam vektor v. Fungsi ini juga dapat digunakan untuk vektor x.

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahan ini sangat halus sehingga kita hanya bisa melihatnya setelah melakukan iterasi yang sangat banyak.

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```



Berikut ini adalah fungsi sederhana untuk memeriksa distribusi seragam dari 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 ini menolak distribusi seragam.

```
>checkrandom(wrongdice(1,1000000))
```

0

Dan ini menerima generator acak bawaan

```
>checkrandom(intrandom(1,1000000,6))
```

1

Kita dapat menghitung distribusi binomial. Pertama, ada `binomials()`, yang mengembalikan probabilitas i atau kurang dari n percobaan.

```
>bindis(410,1000,0.4)
```

0.751401349654

Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p. Tingkat defaultnya adalah alpha. Arti dari interval ini adalah bahwa jika p berada di luar interval, hasil yang diamati dari 410 dalam 1000 jarang terjadi.

```
>clopperpearson(410,1000)
```

[0.37932, 0.441212]

Perintah berikut ini 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

Dan juga, `invbinsum()` menghitung kebalikan dari `binomials()`

```
>invbindis(0.75,1000,0.4)
```

409.932733047

Dalam Bridge, kita mengasumsikan 5 kartu yang terbuka (dari 52 kartu) di 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

Merencanakan Data

Untuk memplot data, kami mencoba hasil pemilihan umum Jerman sejak tahun 1990, yang 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 setiap pihak, kami menggunakan serangkaian nama

```
>P := ["CDU/CSU", "SPD", "FDP", "Gr", "Li"];
```

Mari kita cetak persentase dengan baik. Pertama, kita mengekstrak kolom-kolom yang diperlukan. Kolom 3 sampai 7 adalah kursi masing-masing partai, dan kolom 2 adalah jumlah total kursi. kolom adalah tahun pemilihan.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Kemudian kita mencetak statistik dalam bentuk tabel. Kita menggunakan nama sebagai judul kolom, dan tahun sebagai judul baris. Lebar default untuk kolom adalah $wc = 10$, tetapi kami lebih suka output yang lebih padat. Kolom-kolom akan diperluas untuk label-label kolom, jika perlu.

```
>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

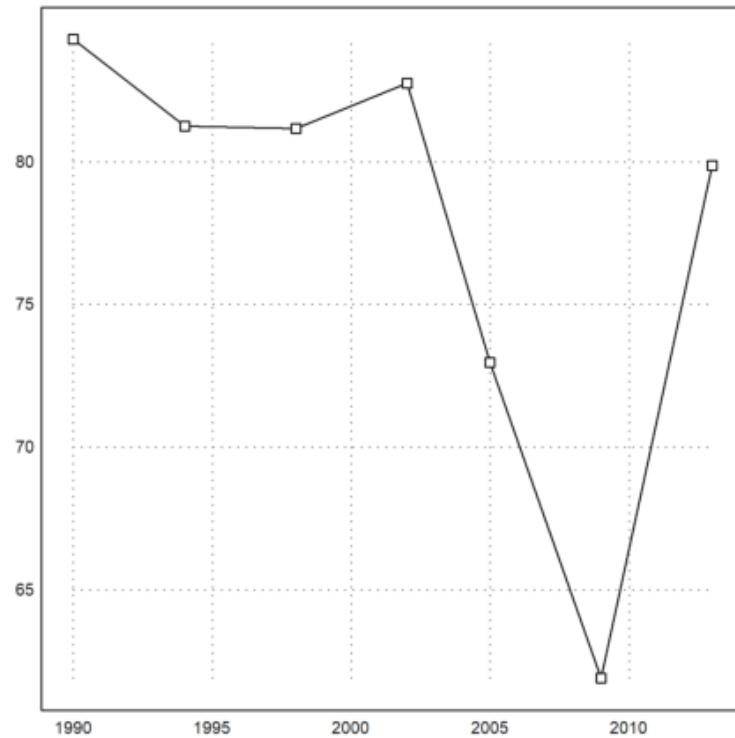
Perkalian matriks berikut ini mengekstrak jumlah persentase dua partai besar yang menunjukkan bahwa partai-partai kecil telah memperoleh suara di parlemen hingga tahun 2009.

```
>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 menggunakannya untuk menampilkan garis dan titik secara bersamaan. Alternatif lainnya adalah memanggil plot2d dua kali dengan $>\text{add}$.

```
>statplot(YT,BT1,"b"):
```

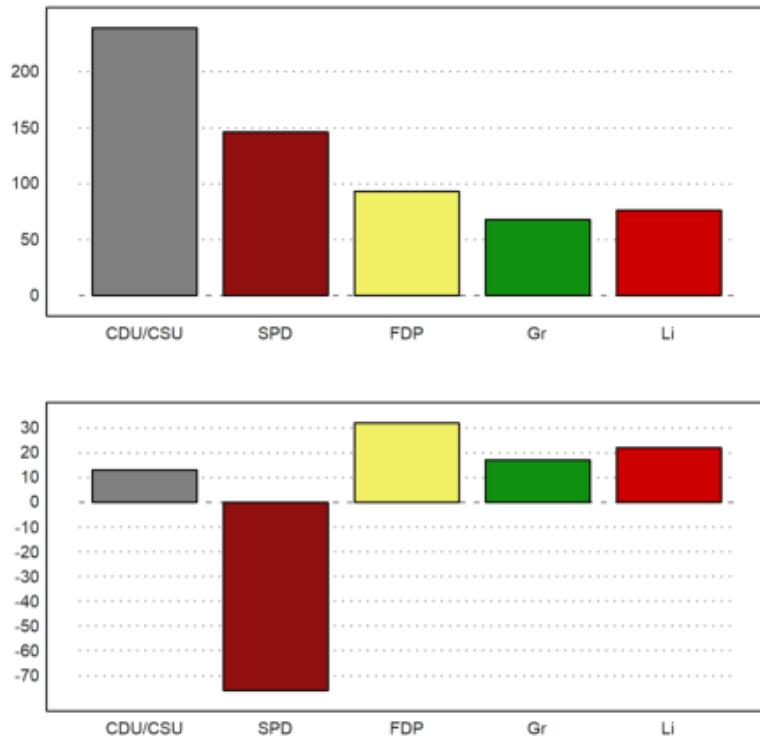


Tentukan beberapa warna untuk masing-masing pihak.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

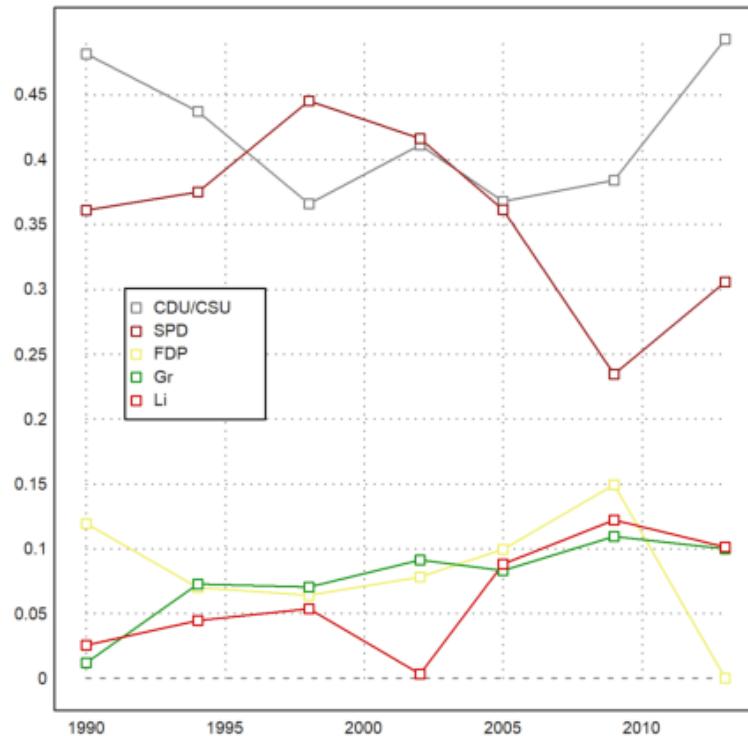
Sekarang kita dapat memplot hasil pemilu 2009 dan perubahannya ke dalam satu plot menggunakan figure. Kita dapat menambahkan vektor kolom pada setiap plot.

```
>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):
```



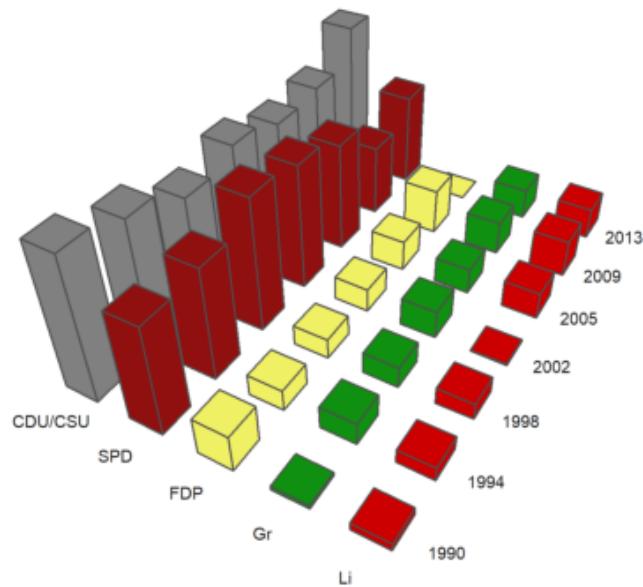
Plot data menggabungkan baris data statistik dalam satu plot

```
>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):
```



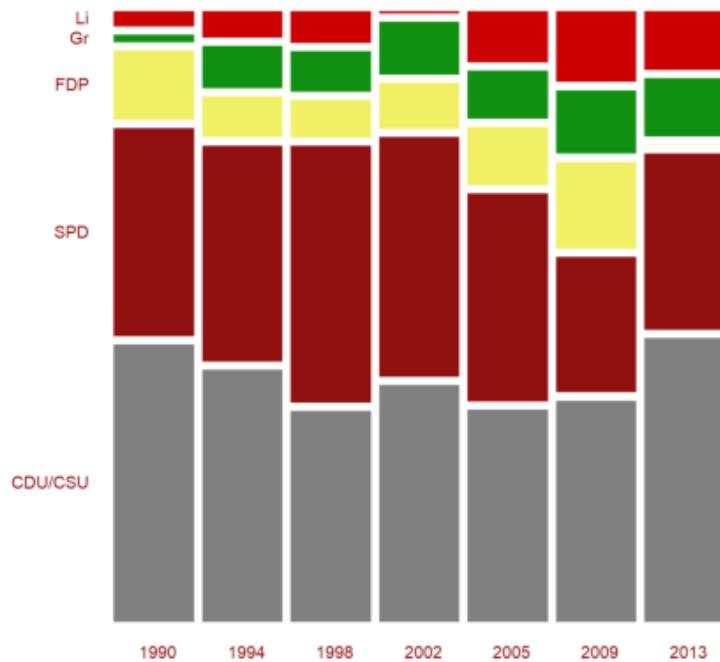
Plot kolom 3D menunjukkan deretan data statistik dalam bentuk kolom. Kami menyediakan label untuk baris dan kolom. angle adalah sudut pandang.

```
>columnplot3d(BT, scols=P, srows=YT, ...
>  angle=30°, ccols=CP) :
```



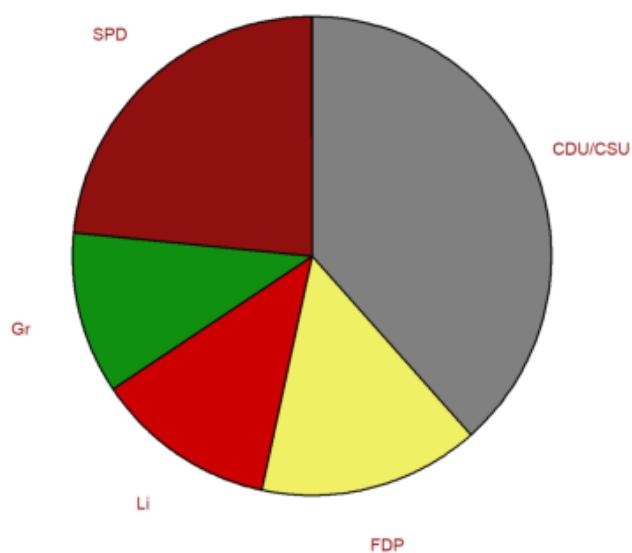
Representasi lainnya adalah plot mosaik. Perhatikan bahwa kolom-kolom pada plot mewakili kolom-kolom pada matriks di sini. Karena panjangnya label CDU/CSU, kita mengambil jendela yang lebih kecil dari biasanya.

```
>shrinkwindow(>smaller); ...
>mosaicplot(BT', srows=YT, scols=P, color=CP, style="#" ); ...
>shrinkwindow():
```



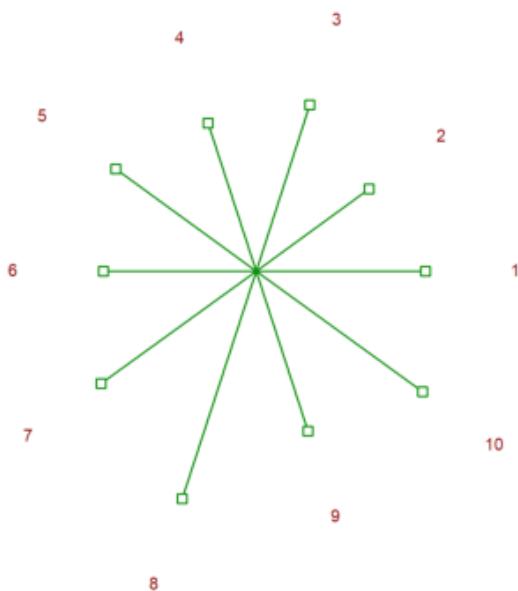
Kita juga bisa membuat diagram lingkaran. Karena warna hitam dan kuning membentuk koalisi, kita menyusun ulang elemen-elemennya.

```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```



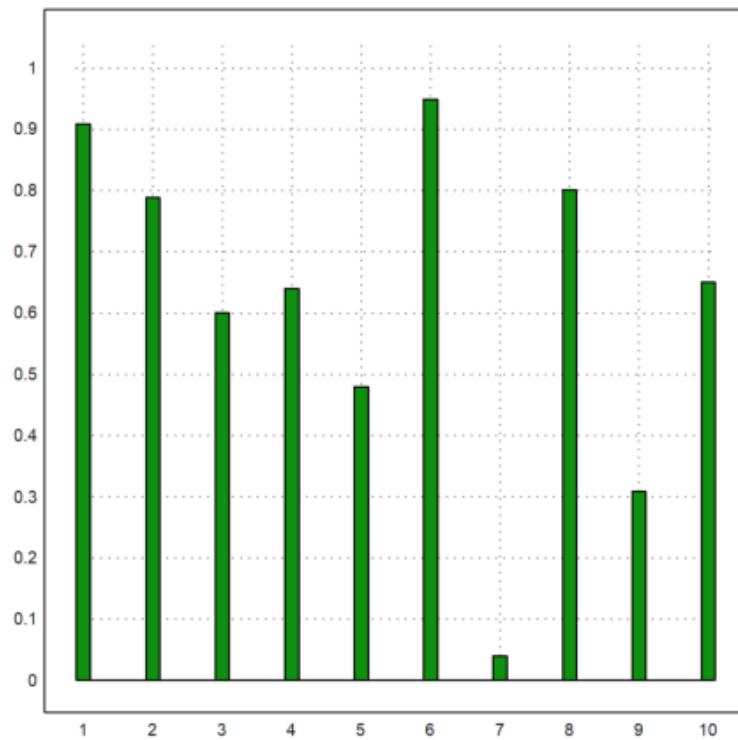
Berikut ini jenis plot yang lain

```
>starplot(normal(1,10)+4,lab=1:10,>rays) :
```



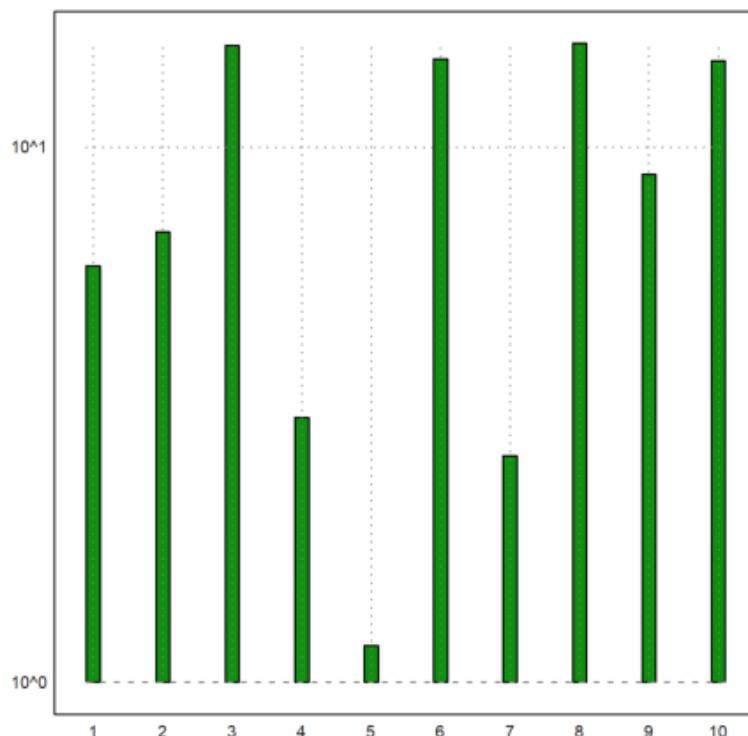
Beberapa plot di plot2d bagus untuk statika. Berikut ini adalah plot impuls dari data acak, yang terdistribusi secara seragam dalam $[0,1]$.

```
>plot2d(makeimpulse(1:10,random(1,10)),>bar) :
```



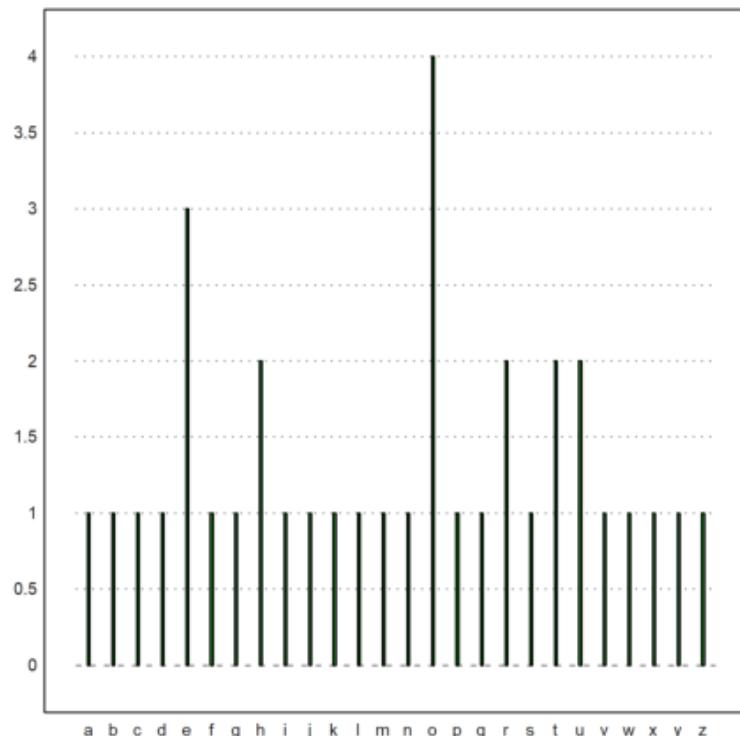
Tetapi untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

```
>logimpulseplot(1:10,-log(random(1,10))*10):
```



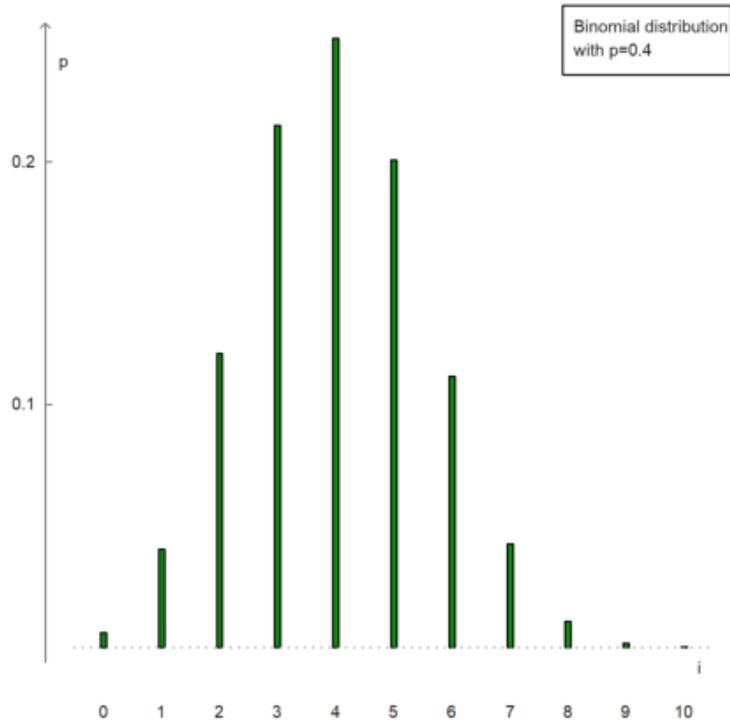
Fungsi `columnsplot()` lebih mudah digunakan, karena hanya membutuhkan sebuah vektor nilai. Selain itu, fungsi ini dapat mengatur labelnya menjadi apa pun yang kita inginkan, kita telah mendemonstrasikan hal ini dalam tutorial ini. Berikut ini adalah aplikasi lain, di mana kami menghitung karakter dalam kalimat dan membuat statistik

```
>v=strtochar("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):
```



Anda juga dapat menetapkan sumbu secara manual.

```
>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 ini adalah cara untuk memplot frekuensi angka dalam vektor. Kami membuat vektor angka acak bilangan bulat 1 hingga 6.

```
>v:=intrandom(1,10,10)
```

```
[8, 5, 8, 8, 6, 8, 8, 3, 5, 5]
```

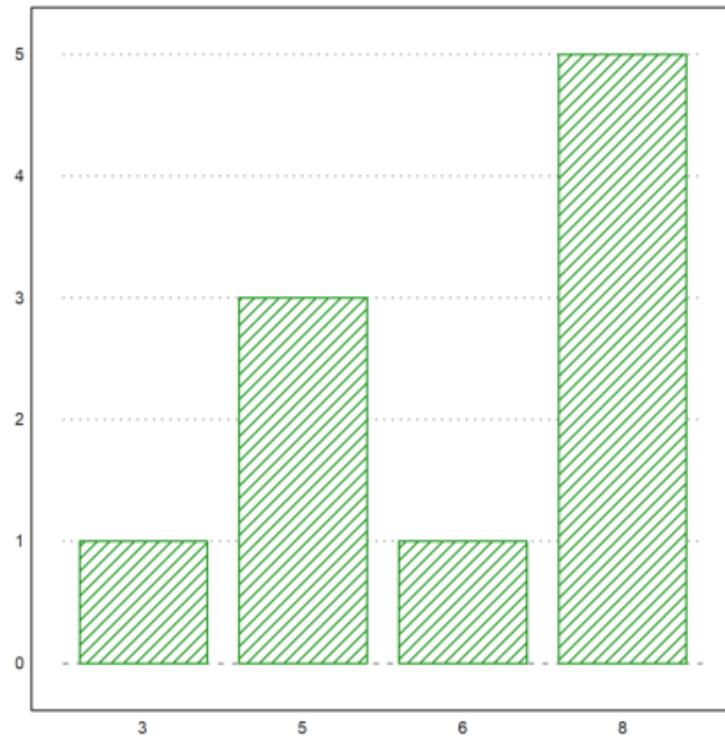
Kemudian ekstrak nomor unik dalam v.

```
>vu:=unique(v)
```

```
[3, 5, 6, 8]
```

Dan memplot frekuensi dalam plot kolom.

```
>columnsplot(getmultiplicities(vu,v),lab=vu,style="/"):
```



Kami ingin mendemonstrasikan fungsi untuk distribusi nilai empiris.

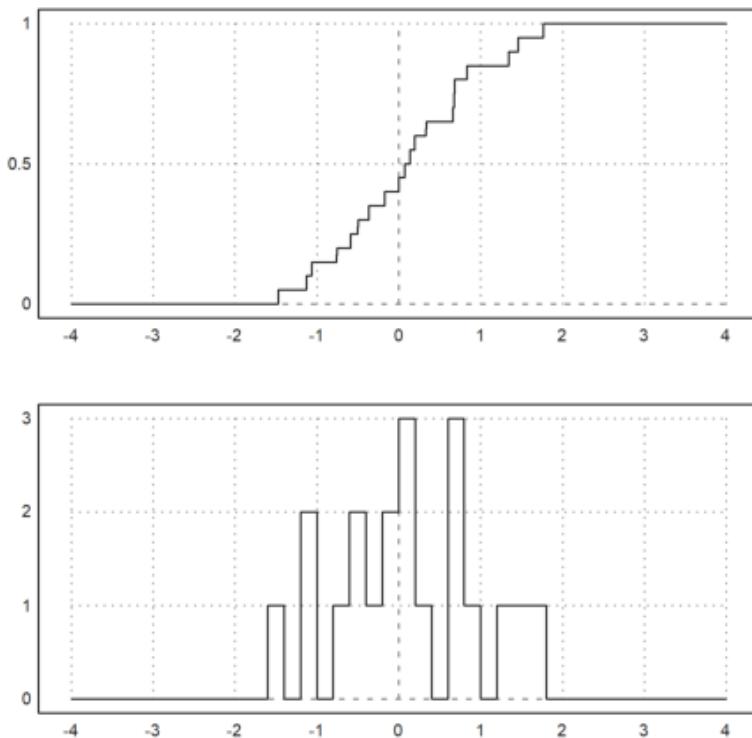
```
>x=normal(1,20);
```

Fungsi empdist(x,vs) membutuhkan larik nilai yang telah diurutkan. Jadi kita harus mengurutkan x sebelum dapat menggunakannya

```
>xs=sort(x);
```

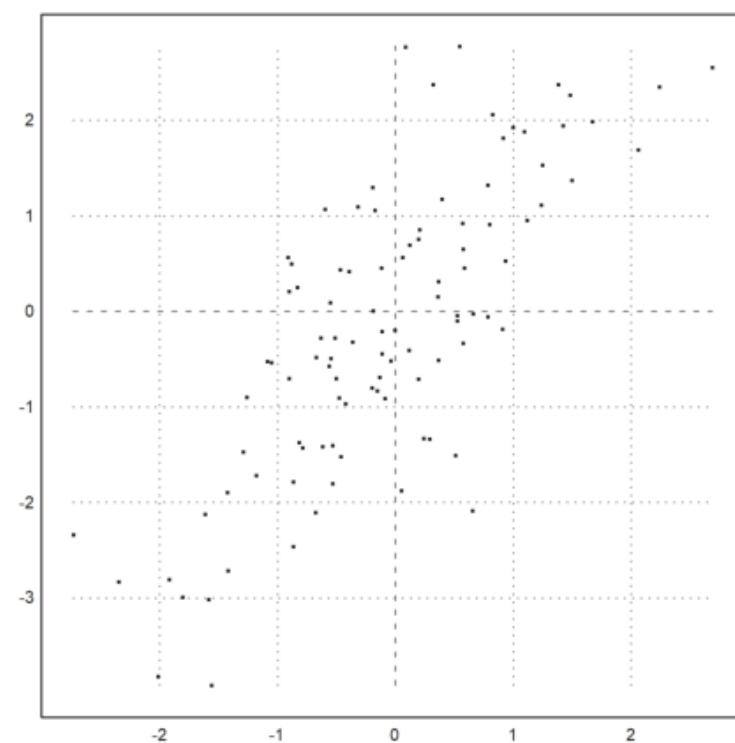
Kemudian kita memplot distribusi empiris dan beberapa batang kepadatan ke dalam satu plot. Alih-alih plot batang untuk distribusi, kali ini kami menggunakan plot gigi gergaji.

```
>figure(2,1); ...
>figure(1); plot2d("empdist",-4,4;xs); ...
>figure(2); plot2d(histo(x,v=-4:0.2:4,<bar>)); ...
>figure(0):
```



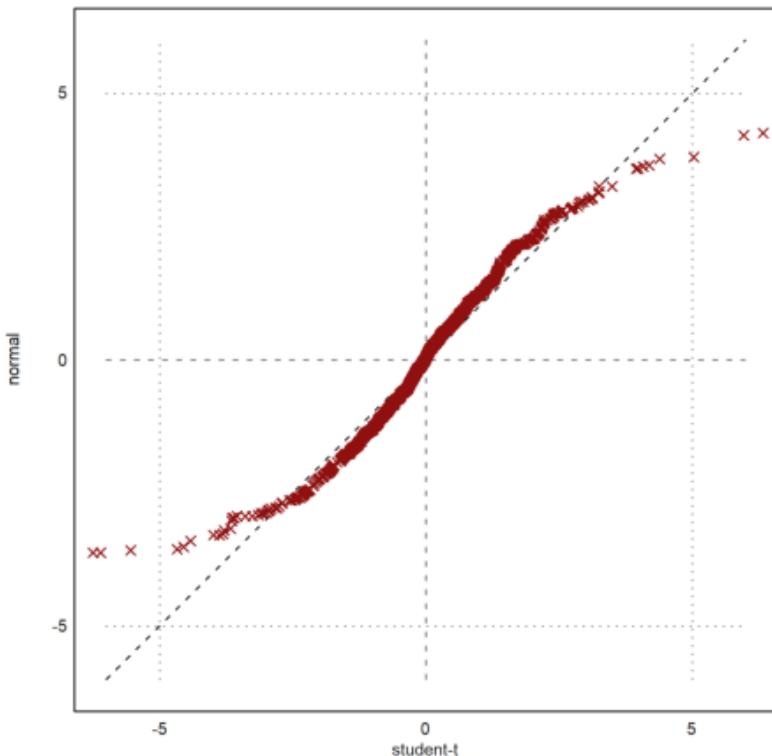
Plot sebaran mudah dilakukan di Euler dengan plot titik biasa. Grafik berikut ini menunjukkan bahwa X dan $X+Y$ berkorelasi positif secara jelas

```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style=".."):
```



Sering kali, kita ingin membandingkan dua sampel dari distribusi yang berbeda. Hal ini dapat dilakukan dengan plot kuantil-kuantil. Untuk pengujian, kami mencoba distribusi student-t 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):
```

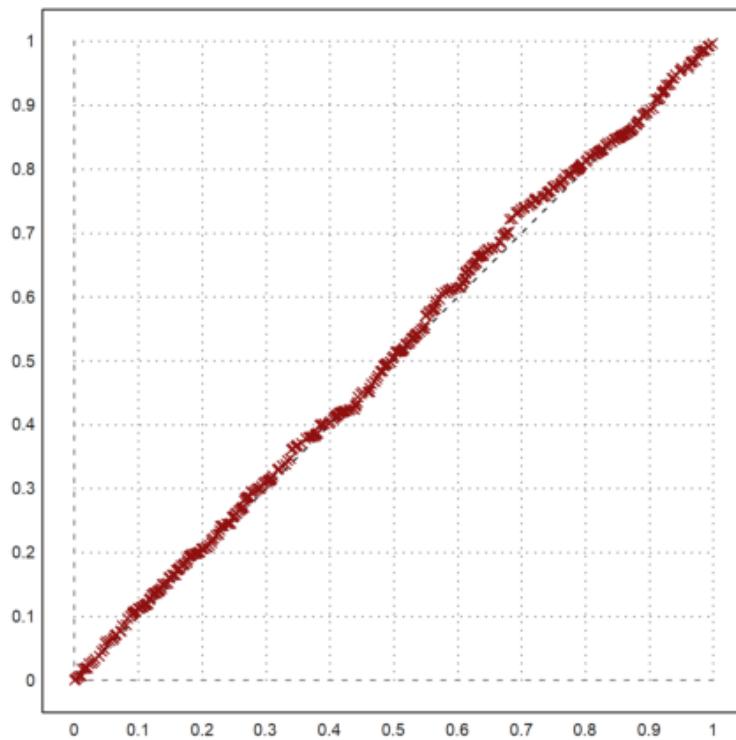


Plot tersebut dengan jelas menunjukkan bahwa nilai yang terdistribusi normal cenderung lebih kecil pada ujung yang ekstrem. Jika kita memiliki dua distribusi dengan ukuran yang berbeda, kita dapat memperluas yang lebih kecil atau memperkecil yang lebih besar. Fungsi berikut ini bagus untuk keduanya. Fungsi ini mengambil nilai median dengan persentase antara 0 dan 1.

```
>function medianexpand (x,n) := median(x,p=linspace(0,1,n-1));
```

Mari kita bandingkan 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)
```



Regresi dan Korelasi

Regresi linier dapat dilakukan dengan fungsi polyfit() atau berbagai fungsi fit. Sebagai perintah, 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

Kami ingin membandingkan kecocokan tanpa bobot dan dengan bobot. Pertama, koefisien dari kecocokan linier

```
>p=polyfit(x,y,1)
```

[0.733333, 0.812121]

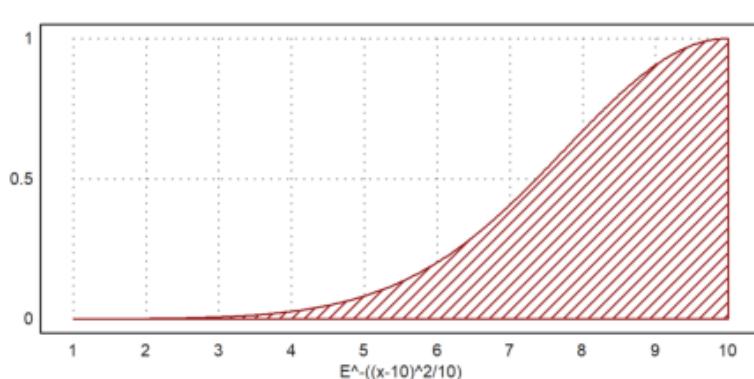
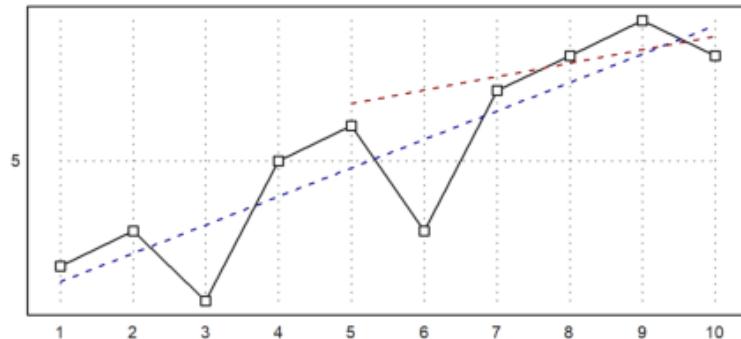
Sekarang, koefisien dengan bobot yang menekankan nilai terakhir

```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

[4.71566, 0.38319]

Kami menempatkan semuanya ke dalam satu plot untuk titik-titik dan garis regresi, dan 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):
```



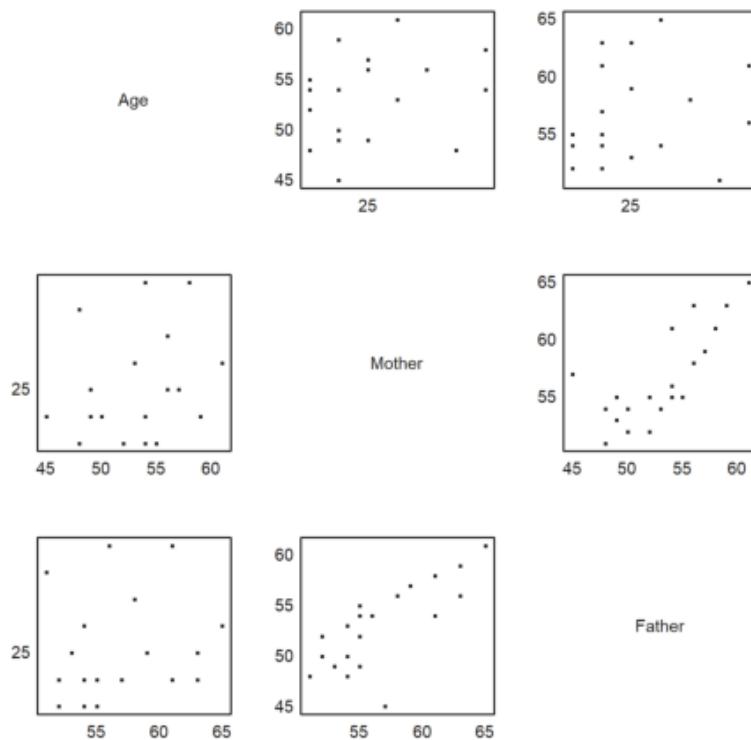
Sebagai contoh lain, kami membaca survei tentang siswa, usia mereka, usia orang tua mereka, dan jumlah saudara kandung dari sebuah file. Tabel ini berisi "m" dan "f" pada kolom kedua. Kita menggunakan variabel tok2 untuk mengatur terjemahan yang tepat alih-alih membiarkan readtable() mengumpulkan terjemahan.

```
>{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 saling bergantung satu sama lain? Kesan pertama datang dari scatterplot berpasangan

```
>scatterplots(tablecol(MS, 3:5),hd[3:5]):
```



Jelas bahwa usia ayah dan ibu saling bergantung satu sama lain. Mari kita tentukan dan plot garis regresinya.

```
>cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1)
```

```
[17.3789, 0.740964]
```

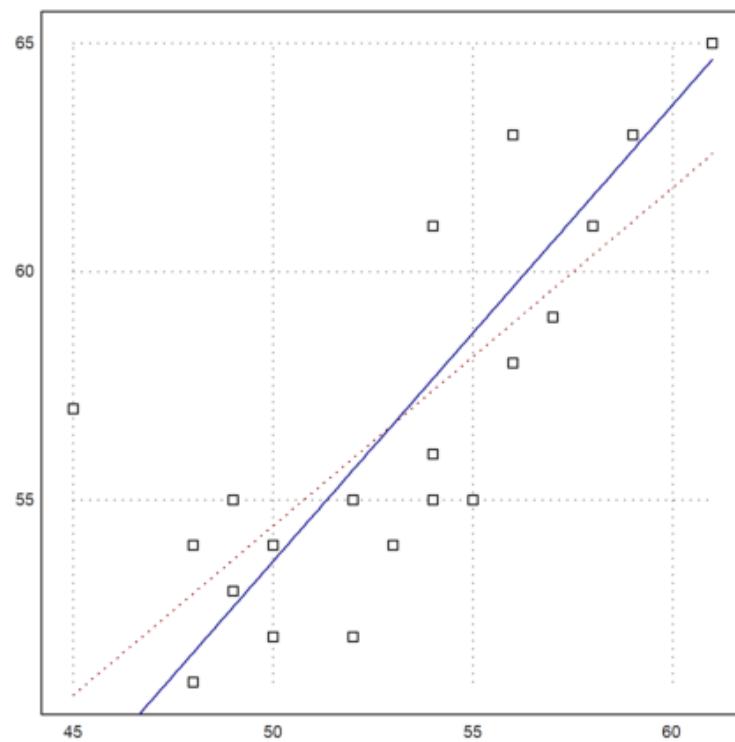
Ini jelas merupakan model yang salah. Garis regresinya adalah $s = 17 + 0,74t$, di mana t adalah usia ibu dan s adalah usia ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak terlalu banyak. Sebaliknya, kita menduga sebuah fungsi seperti $s = a + t$. Maka a adalah rata-rata dari s-t. Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

```
>da:=mean(cs[2]-cs[1])
```

```
3.65
```

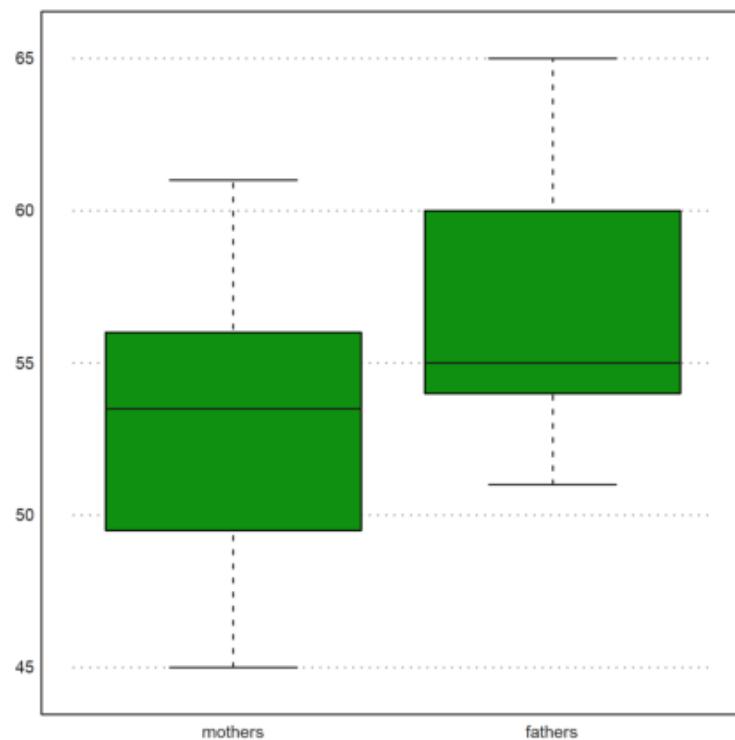
Mari kita plotkan ini ke dalam satu scatter plot.

```
>plot2d(cs[1],cs[2],>points); ...
>plot2d("evalpoly(x,ps)",color=red,style=".",>add); ...
>plot2d("x+da",color=blue,>add):
```



Berikut ini adalah plot kotak dari kedua usia tersebut. Ini hanya menunjukkan, bahwa usia keduanya berbeda.

```
>boxplot(cs, ["mothers", "fathers"]):
```



Sangat menarik bahwa perbedaan dalam median tidak sebesar perbedaan dalam 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. Korelasi 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.

$$\text{sk}(x) = \frac{\sqrt{n} \sum_i (x_i - m)^3}{(\sum_i (x_i - m)^2)^{3/2}}$$

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 ini adalah fungsi lain, yang disebut koefisien kemencengan Pearson.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)
>skew1(data)
```

-0.0801873249135

Simulasi Monte Carlo

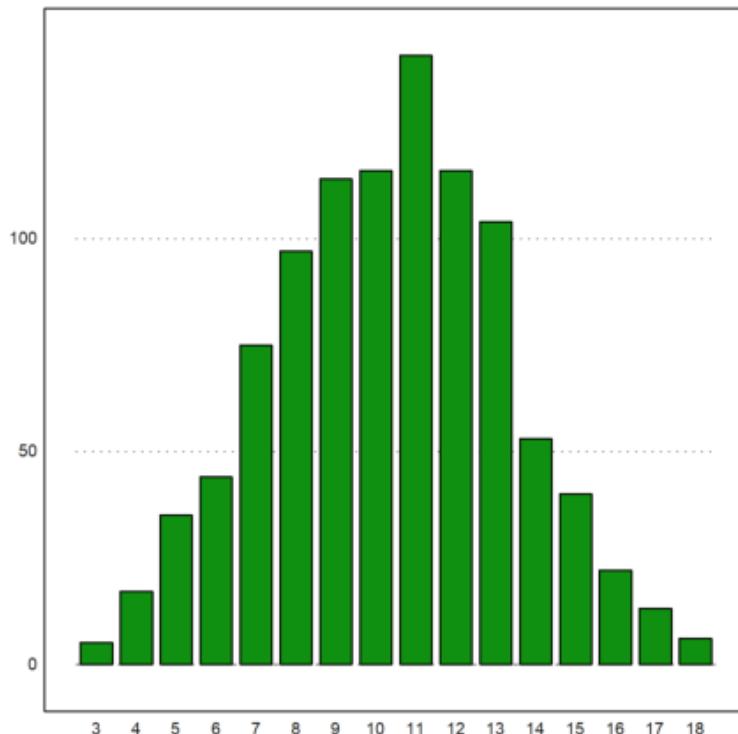
Euler dapat digunakan untuk mensimulasikan kejadian acak. Kita telah melihat contoh sederhana di atas. Berikut ini adalah contoh lainnya, yang mensimulasikan 1000 kali pelemparan 3 dadu, dan menanyakan distribusi dari jumlah tersebut

```
>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]

Kita bisa merencanakan ini sekarang

```
>columnsplot(fs,lab=3:18):
```



Untuk menentukan distribusi yang diharapkan tidaklah mudah. Kami menggunakan rekursi tingkat lanjut untuk hal ini. Fungsi berikut ini menghitung jumlah cara angka k dapat direpresentasikan sebagai jumlah n angka dalam rentang 1 hingga m. Fungsi 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
```

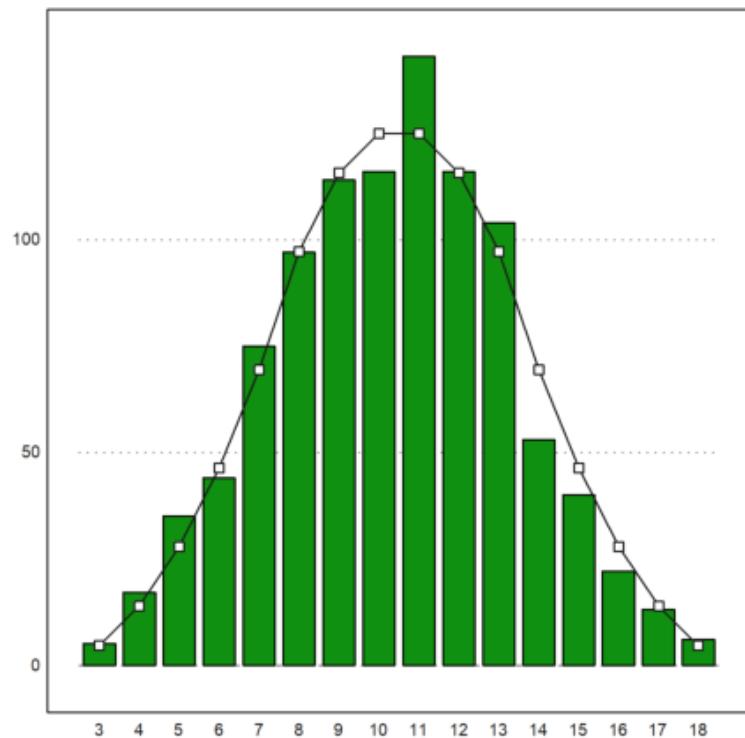
Berikut ini adalah hasil dari tiga lemparan dadu.

```
>cw=countways(3:18,3,6)
```

```
[1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3,
1]
```

Kami menambahkan nilai yang diharapkan ke plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```



Untuk simulasi lainnya, deviasi nilai rata-rata dari n variabel acak berdistribusi normal 0-1 adalah $1/\sqrt{n}$

```
>longformat; 1/sqrt(10)
```

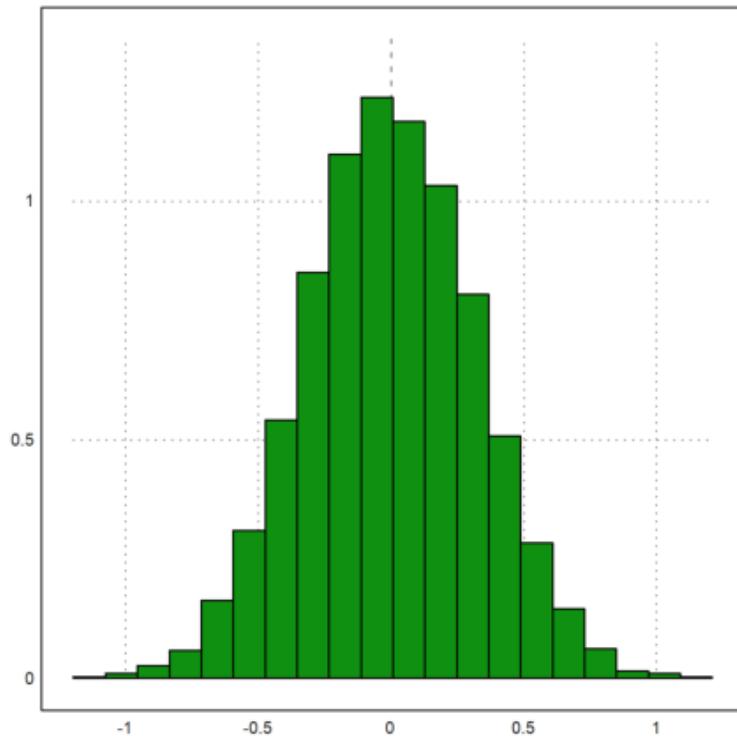
0.316227766017

Mari kita periksa dengan sebuah simulasi. Kami menghasilkan 10.000 kali 10 vektor acak

```
>M=normal(10000,10); dev(mean(M)')
```

0.319493614817

```
>plot2d(mean(M)',>distribution):
```



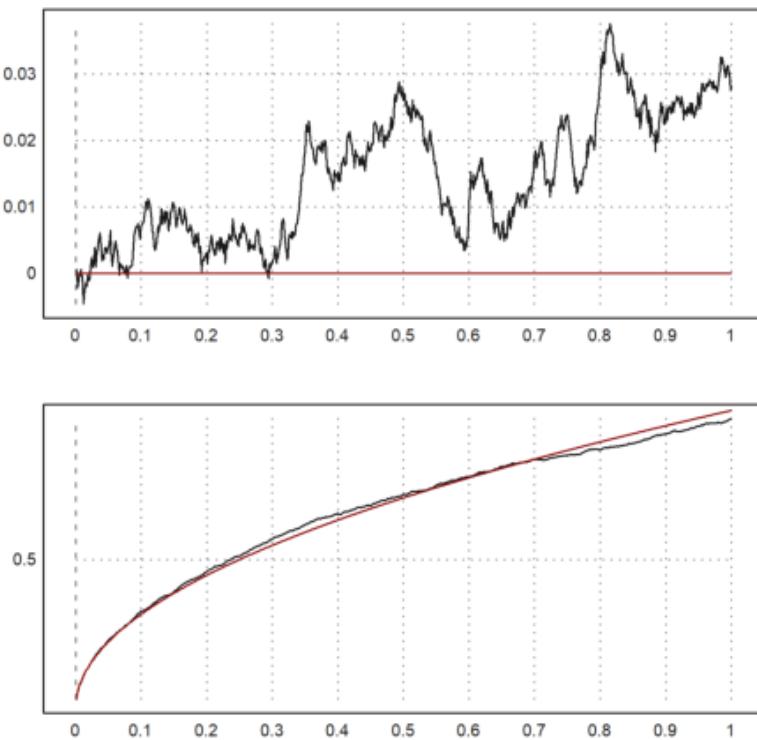
Median dari 10 angka acak berdistribusi 0-1 yang terdistribusi normal memiliki deviasi yang lebih besar

```
>dev(median(M)')
```

0.374460271535

Karena kita dapat dengan mudah menghasilkan jalan acak, kita dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami kemudian memplot deviasi standar dan rata-rata dari 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):
```



Uji

Uji adalah alat yang penting dalam statistik. Dalam Euler, banyak uji yang diterapkan. Semua uji ini mengembalikan kesalahan yang kita terima jika kita menolak hipotesis nol. Sebagai contoh, kami menguji lemparan dadu untuk mengetahui distribusi yang seragam. Pada 600 lemparan, kami mendapatkan nilai berikut, yang kami 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 mengejuti statistik. Hasilnya seharusnya hampir sama. Parameter `>p` menginterpretasikan vektor `y` sebagai vektor probabilitas

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

0.526

Kesalahan ini terlalu besar. Jadi kita tidak bisa menolak distribusi seragam. Ini tidak membuktikan bahwa dadu kita adil. Namun kita tidak bisa menolak hipotesis kita. Selanjutnya kami 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 kita uji 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 untuk diuji. Sekarang mari kita periksa dua pengukuran untuk mean yang sama. Kita tolak hipotesis bahwa kedua pengukuran tersebut memiliki nilai r a t a - r a t a yang sama, jika hasilnya < 0,05.

```
>tcomparedata(normal(1,10),normal(1,10))
```

0.38722000942

Jika kita menambahkan bias pada satu distribusi, kita akan mendapatkan lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

5.60009101758e-07

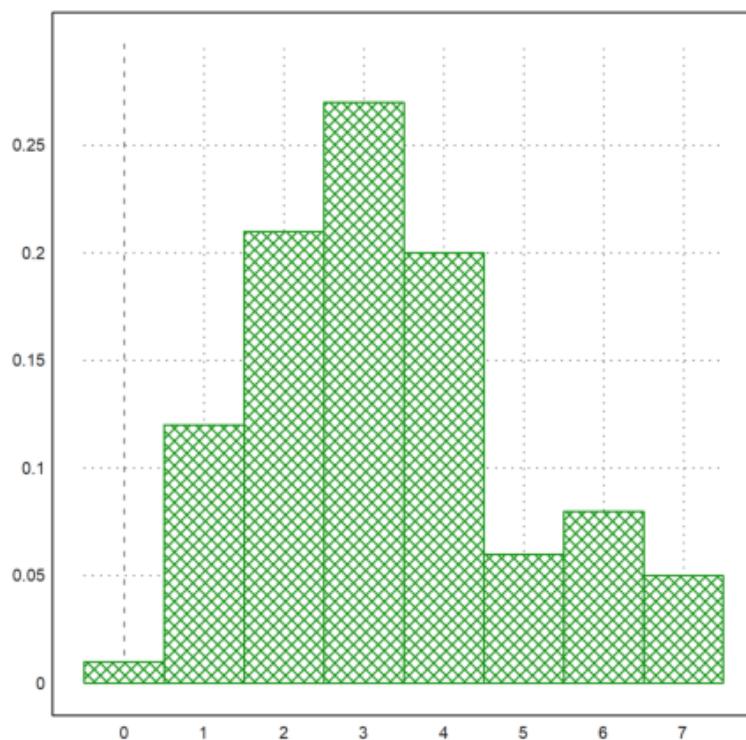
Pada contoh berikut, kita membuat 20 lemparan dadu secara acak sebanyak 100 kali dan menghitung jumlah dadu yang muncul. Rata-rata harus ada $20/6 = 3,3$ mata dadu.

```
>R=random(100,20); R=sum(R*6<=1)'; mean(R)
```

3.28

Sekarang kita bandingkan jumlah satu dengan distribusi binomial. Pertama, kita memplot distribusi angka satu

```
>plot2d(R,distribution=max(R)+1,even=1,style="\\\"/\\"):
```



```
>t=count(R,21);
```

Kemudian kami menghitung nilai yang diharapkan

```
>n=0:20; b=bin(20,n)*(1/6)^n*(5/6)^(20-n)*100;
```

Kami 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 berikut ini berisi hasil dari dua kelompok orang (laki-laki dan perempuan, katakanlah) yang memberikan suara untuk 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

Kami ingin menguji independensi suara dari jenis kelamin. Uji tabel chi² melakukan hal ini. Hasilnya terlalu besar untuk menolak independensi. Jadi kita tidak dapat mengatakan, jika pemungutan suara tergantung pada jenis kelamin dari data ini.

```
>tabletest(A)
```

0.990701632326

Berikut ini adalah tabel yang diharapkan, jika kita mengasumsikan frekuensi pemungutan suara 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 telah dikoreksi. Karena koefisien ini sangat dekat dengan 0, kami menyimpulkan bahwa pemungutan suara tidak bergantung pada jenis kelamin

```
>contingency(A)
```

0.0427225484717

Beberapa Tes Lainnya

Selanjutnya kita menggunakan analisis varians (uji F) untuk menguji tiga sampel data yang terdistribusi secara normal untuk nilai rata-rata yang sama. Metode ini disebut ANOVA (analisis varians). 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, kami menolak hipotesis nilai rata-rata yang sama. Kami melakukan ini dengan probabilitas kesalahan sebesar 1,3%. Ada juga uji median, yang menolak sampel data dengan distribusi rata-rata yang berbeda dengan menguji median dari sampel gabungan.

```
>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. Uji ini jauh lebih tajam daripada uji median

```
>ranktest(a,b)
```

0.00199969612469

Dalam contoh berikut, kedua distribusi memiliki rata-rata yang sama

```
>ranktest(random(1,100),random(1,50)*3-1)
```

0.129608141484

Sekarang mari kita coba mensimulasikan dua perawatan 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 signum memutuskan, apakah a lebih baik daripada b

```
>signtest(a,b)
```

0.0546875

Ini adalah kesalahan yang terlalu besar. Kita tidak dapat menolak bahwa a sama baiknya dengan b. Uji Wilcoxon lebih tajam daripada uji ini, tetapi bergantung pada nilai kuantitatif dari perbedaan

```
>wilcoxon(a,b)
```

0.0296680599405

Mari kita coba dua pengujian lagi dengan menggunakan rangkaian yang dihasilkan.

```
>>wilcoxon(normal(1,20),normal(1,20)-1)
```

0.0068706451766

```
>>wilcoxon(normal(1,20),normal(1,20))
```

0.275145971064

Angka Acak

Berikut ini adalah tes untuk generator angka acak. Euler menggunakan generator yang sangat baik, jadi kita tidak perlu mengharapkan masalah. Pertama, kita akan membangkitkan sepuluh juta angka acak dalam [0,1].

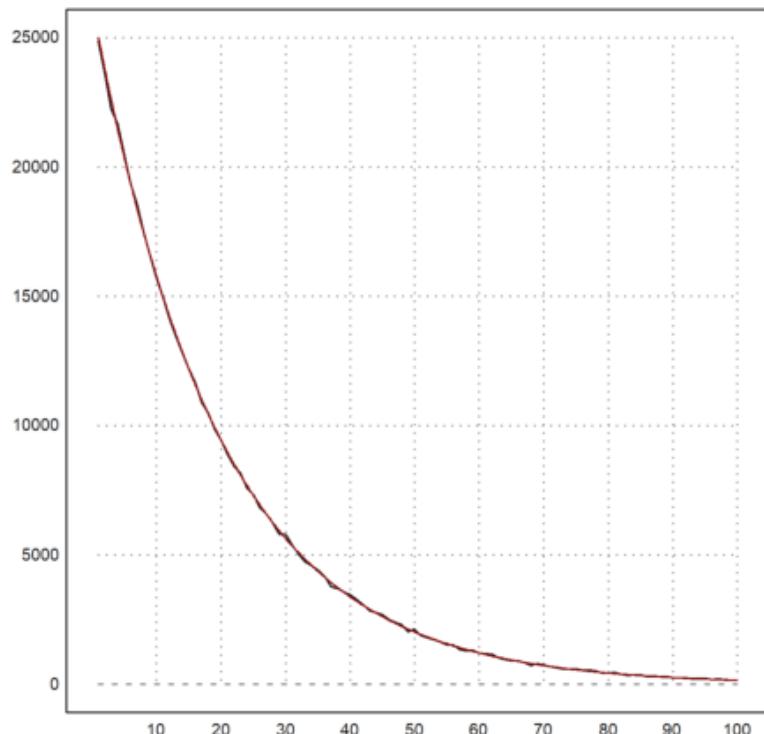
```
>n:=10000000; r:=random(1,n);
```

Selanjutnya, kami menghitung jarak antara dua angka yang kurang dari 0,05.

```
>a:=0.05; d:=differences(nonzeros(r<a));
```

Terakhir, kami memplot berapa kali, setiap jarak yang 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):
```



Menghapus data.

```
>remvalue n;
```

Jelas, EMT tidak bersaing dengan R sebagai paket statistik. Namun, ada banyak prosedur dan fungsi statistik yang tersedia di EMT juga. Jadi, EMT dapat memenuhi kebutuhan dasar. Bagaimanapun, EMT hadir dengan paket numerik dan sistem aljabar komputer. Buku catatan ini cocok untuk Anda jika Anda sudah terbiasa dengan R, tetapi perlu mengetahui perbedaan sintaks EMT dan R. Kami mencoba memberikan gambaran umum tentang hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui. Selain itu, kami juga akan membahas caracara untuk bertukar data di antara kedua sistem. Perhatikan bahwa ini adalah pekerjaan yang sedang berlangsung

Sintaksis Dasar

Hal pertama yang Anda pelajari dalam R adalah membuat sebuah vektor. Dalam EMT, perbedaan utamanya adalah operator : dapat mengambil ukuran langkah. Selain itu, operator ini memiliki daya ikat yang 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. Dimungkinkan untuk menggunakan vektor untuk menggabungkan berbagai hal. Contoh berikut ini, seperti banyak contoh lainnya, berasal dari "Introduction to R" yang disertakan dengan proyek R. Jika Anda membaca PDF ini, Anda akan menemukan bahwa saya mengikuti alurnya 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 dituliskan 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 unit dalam EMT.

```
>125km -> " miles"
```

77.6713990297 miles

Operator "<- " untuk penugasan juga menyesatkan, dan bukan ide yang baik untuk R. Berikut ini akan membandingkan a dan -4 dalam EMT

```
>a=2; a<-4
```

0

Dalam R, "a<-4<3" bisa digunakan, tetapi "a<-4<-3" tidak. Saya juga mengalami ambiguitas yang sama dalam EMT, tetapi mencoba untuk menghilangkannya. EMT dan R memiliki vektor dengan tipe boolean. Namun dalam EMT, angka 0 dan 1 digunakan untuk merepresentasikan salah dan benar. Dalam R, nilai benar dan salah tetap dapat digunakan dalam aritmatika biasa seperti dalam EMT.

```
>x<5, %*%
```

[0, 0, 1, 0, 0]
[0, 0, 3.1, 0, 0]

EMT melempar kesalahan atau menghasilkan NAN tergantung pada flag "kesalahan"

```
>errors off; 0/0, isNaN(sqrt(-1)), errors on;
```

NAN

1

String sama saja dalam R dan EMT. Keduanya berada dalam lokal saat ini, bukan dalam Unicode. Dalam R terdapat paket-paket untuk Unicode. Dalam EMT, sebuah string dapat berupa string Unicode. Sebuah string unicode dapat diterjemahkan ke pengkodean lokal dan sebaliknya. Selain itu, u "..." dapat berisi entitas HTML.

```
>u"     ; Ren   Grothmann"
```

Berikut ini mungkin atau mungkin tidak ditampilkan dengan benar pada sistem Anda sebagai A dengan titik dan tanda hubung di atasnya. Hal ini tergantung pada jenis huruf yang Anda gunakan.

```
>chartoutf([480])
```

Penggabungan string dilakukan dengan "+" atau "|". Ini dapat menyertakan angka, yang akan dicetak dalam format saat ini.

```
>"pi = "+pi
```

```
pi = 3.14159265359
```

Pengindeksan

Sebagian besar waktu, ini akan bekerja seperti dalam R. Tetapi EMT akan menginterpretasikan indeks negatif dari bagian belakang vektor, sedangkan 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 dalam EMT dengan drop()

```
>drop(x, 2)
```

```
[10.4, 3.1, 6.4, 21.7]
```

Vektor logika tidak diperlakukan secara berbeda dengan indeks di EMT, berbeda dengan R. Anda perlu mengekstrak elemen yang bukan 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 dalam R, vektor indeks dapat berisi pengulangan

```
>x[ [1,2,2,1] ]
```

```
[10.4, 5.6, 5.6, 10.4]
```

Namun pemberian nama untuk indeks tidak dimungkinkan dalam EMT. Untuk paket statistik, hal ini mungkin 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)]; ... ...
^
```

Tipe Data

EMT memiliki lebih banyak tipe data yang tetap dibandingkan R. Jelas, dalam R ada vektor yang berkembang. Anda bisa mengatur sebuah vektor numerik kosong v dan memberikan sebuah nilai pada elemen v[17]. Hal ini tidak mungkin dilakukan dalam EMT. Berikut ini agak tidak efisien

```
>v=[]; for i=1 to 10000; v=v|i; end;
```

EMT sekarang akan membuat vektor dengan v dan i yang ditambahkan pada tumpukan dan menyalin vektor tersebut kembali ke variabel global v. Semakin efisien mendefinisikan vektor

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Untuk mengubah jenis tanggal di EMT, Anda dapat menggunakan fungsi seperti complex().

```
>complex(1:4)
```

```
[ 1+0i , 2+0i , 3+0i , 4+0i ]
```

Konversi ke string hanya dapat dilakukan untuk tipe data dasar. Format saat ini digunakan untuk penggabungan string sederhana. Tetapi ada fungsi-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 komunikasi dengan Maxima, ada fungsi convertm xm(), 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

Dalam pengantar R ada sebuah contoh dengan apa yang disebut faktor. Berikut ini 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"];
```

Asumsikan, 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 tersebut. Sebagai sebuah program statistik, R memiliki fungsi factor() dan tapply() untuk hal ini. EMT dapat melakukan hal ini dengan menemukan indeks wilayah dalam daftar wilayah yang unik.

```
>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, 7, 1]
```

Pada titik ini, kita dapat menulis fungsi perulangan kita sendiri untuk melakukan berbagai hal untuk satu faktor saja. Atau kita dapat meniru fungsi tapply() dengan cara berikut.

```
>function map_tapp1 (i; f$call, cat, x) ...
```

```
u=sort(unique(cat));
f=indexof(u,cat);
return f$(x[nonzeros(f==indexof(u,i))]);
endfunction
```

Ini sedikit tidak efisien, karena menghitung wilayah unik untuk setiap i, tetapi berfungsi.

```
>tapp1(auterr,"mean",austates,incomes)
```

```
[44.5, 57.3333333333, 55.5, 53.6, 55, 60.5, 56, 52.25]
```

Perhatikan bahwa ini bekerja untuk setiap vektor wilayah.

```
>tapp1(["act","nsw"],"mean",austates,incomes)
```

```
[44.5, 57.3333333333]
```

Sekarang, paket statistik EMT mendefinisikan tabel seperti halnya di R. Fungsi readtable() dan writetable() dapat digunakan untuk input dan output. Jadi kami dapat mencetak pendapatan negara rata-rata di wilayah tersebut dengan cara yang bersahabat.

```
>writetable(tapply(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 tersebut harus disimpan dengan jelas dalam sebuah koleksi dengan jenis dan kategorinya (negara bagian dan teritori 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 ini 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 tapply() dengan cara berikut. Ini akan mengembalikan sebuah tabel sebagai kumpulan 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 adalah kategori (faktor) yang tidak memiliki data. Tetapi kita harus memeriksa panjang t yang benar dan ketepatan koleksi tf. Tabel ini dapat dicetak sebagai sebuah 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

Susunan

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Akan lebih mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau pustaka C untuk hal ini. R memiliki lebih dari dua dimensi. Dalam R, larik adalah vektor dengan bidang berdimensi. Dalam EMT, sebuah vektor adalah sebuah matriks dengan satu baris. Ini bisa dibuat menjadi sebuah matriks dengan 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, sama seperti di R

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

2	3
7	8
12	13
17	18

Namun, dalam R dimungkinkan untuk mengatur daftar indeks tertentu dari vektor ke suatu nilai. Hal yang sama juga dapat dilakukan dalam EMT hanya dengan sebuah perulangan

```
>function setmatrixvalue (M, i, j, v) ...
```

```
loop 1 to max(length(i),length(j),length(v))
  M[i{#},j{#}] = v{#};
end;
endfunction
```

Kami mendemonstrasikan hal ini untuk menunjukkan bahwa matriks diteruskan dengan referensi dalam EMT. Jika Anda tidak ingin mengubah matriks asli M, Anda perlu menyalininya dalam fungsi.

```
>setmatrixvalue(X,1:3,3:-1:1,0); X,
```

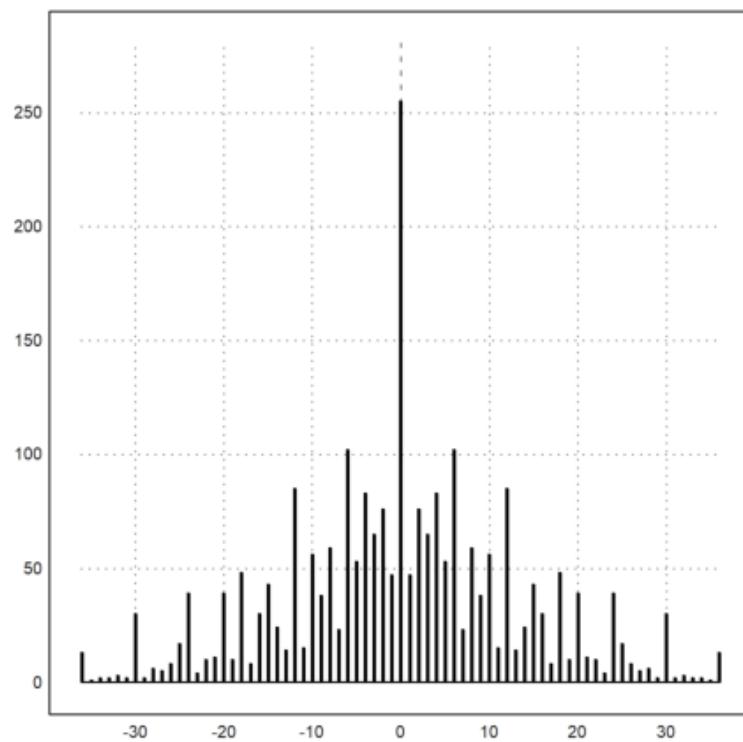
1	2	0	4	5
6	0	8	9	10
0	12	13	14	15
16	17	18	19	20

```
>(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 pengantar PDF untuk R ada sebuah contoh, yang menghitung distribusi ab-cd untuk a, b, c, d yang dipilih dari 0 sampai n secara acak. Solusinya dalam R adalah membentuk sebuah matriks 4 dimensi dan menjalankan tabel() di atasnya. Tentu saja, hal ini bisa dicapai dengan perulangan. Tetapi perulangan tidak efektif dalam EMT atau R. Dalam EMT, kita dapat menulis perulangan dalam C dan itu akan menjadi solusi tercepat. Tetapi kita ingin meniru perilaku R. Untuk itu, kita perlu meratakan perkalian ab dan membuat matriks abcd.

```
>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"):
```



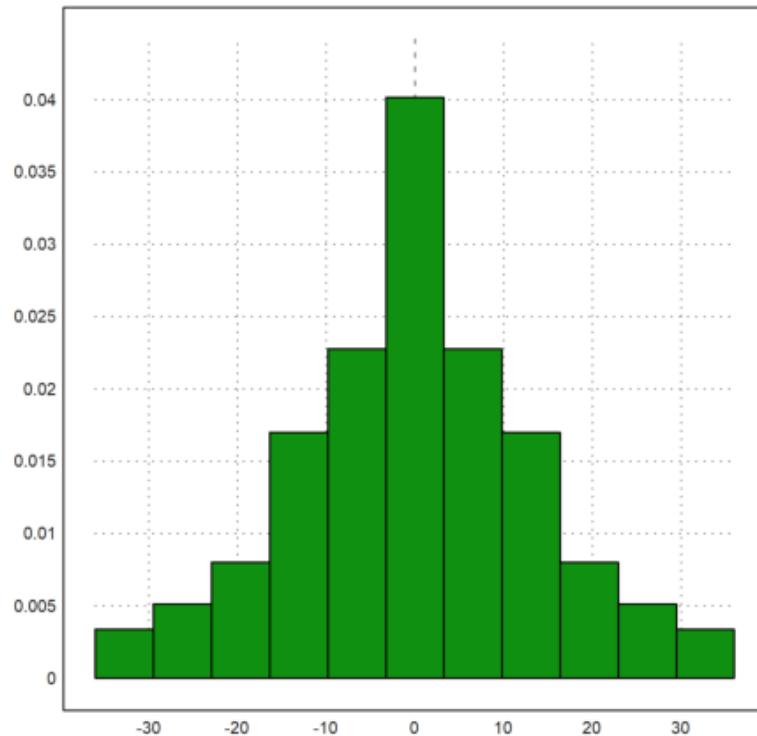
Selain kelipatan yang tepat, EMT dapat menghitung frekuensi dalam vektor

```
>getfrequencies(q, -50:10:50)
```

```
[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]
```

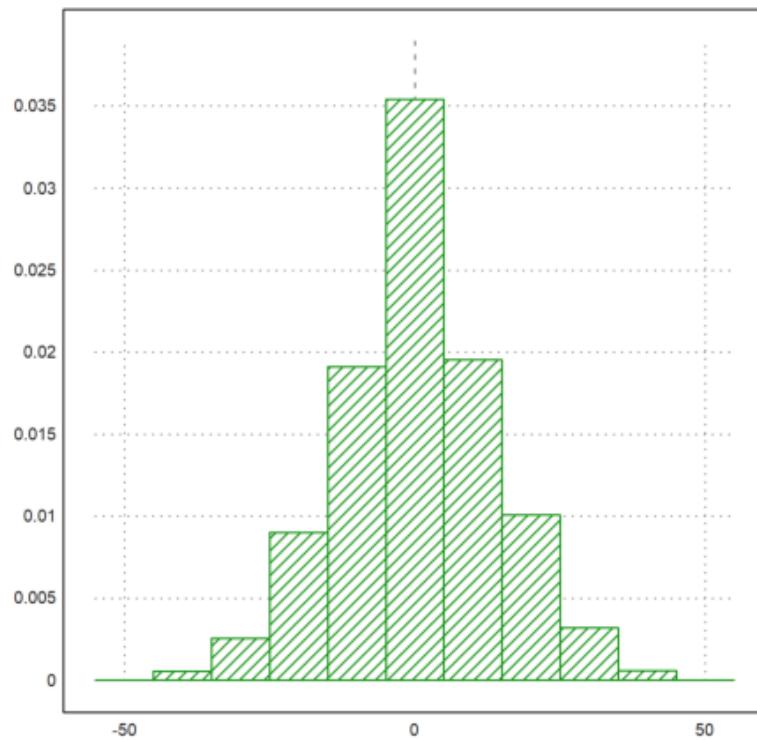
Cara yang paling mudah untuk memplot ini sebagai distribusi adalah sebagai berikut.

```
>plot2d(q,distribution=11):
```



Tetapi juga memungkinkan untuk menghitung jumlah dalam interval yang dipilih sebelumnya. Tentu saja, berikut ini menggunakan `getfrequencies()` secara internal. Karena fungsi `histo()` mengembalikan frekuensi, kita perlu menskalakannya sehingga integral di bawah grafik batang adalah 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. Yang pertama adalah daftar global yang dapat diubah, dan yang kedua adalah jenis daftar yang tidak dapat diubah. Kita tidak akan membahas daftar global di sini. Tipe daftar yang tidak dapat diubah disebut koleksi dalam EMT. Tipe ini berperilaku seperti struktur di C, tetapi elemen-elemennya hanya diberi nomor dan tidak diberi nama.

```
>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. Elemen-elemen tersebut diakses dengan angka.

```
>(L[4])[2]
```

```
1992
```

Input dan Output File (Membaca dan Menulis Data)

Anda mungkin sering ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberitahu Anda tentang banyak cara untuk mencapai hal ini. Fungsi yang sederhana adalah writematrix() dan readmatrix(). Mari kita tunjukkan cara membaca dan menulis vektor real ke sebuah file

```
>a=random(1,100); mean(a), dev(a),
```

```
0.49815  
0.28037
```

Untuk menulis data ke sebuah file, kita menggunakan fungsi writematrix(). Karena pengenalan ini kemungkinan besar berada di direktori, di mana pengguna tidak memiliki akses tulis, kami menulis data ke direktori home pengguna. Untuk notebook sendiri, hal ini tidak perlu dilakukan, karena file data akan ditulis ke direktori yang sam

```
>filename="test.dat";
```

Sekarang kita tuliskan vektor kolom a' ke dalam file. Hal ini akan menghasilkan satu angka pada 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 dikonfigurasikan untuk bahasa lain. Misalnya, jika Anda memiliki sistem Indonesia (titik desimal dengan koma), Excel Anda membutuhkan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan dengan koma). File "test.csv" berikut ini akan muncul di folder cuurent Anda.

```
>filename="test.csv"; ...
>writematrix(random(5,3),file=filename,separator=",");
```

Anda sekarang dapat membuka file ini dengan Excel Indonesia secara langsung.

```
>fileremove(filename);
```

Terkadang kita memiliki string dengan token seperti berikut ini.

```
>s1:="f m m f m m m f f f m m f"; ...
>s2:="f f f m m f f";
```

Untuk menandai ini, kita mendefinisikan vektor token

```
>tok:=[ "f", "m" ]
```

f

m

Kemudian kita dapat menghitung berapa kali setiap token muncul dalam string, dan memasukkan hasilnya ke dalam tipe

```
>M:=getmultiplicities(tok,strtokens(s1))_ ...
>  getmultiplicities(tok,strtokens(s2));
```

Tulis tabel dengan tajuk token.

```
>writetable(M,labc=tok,labr=1:2,wc=8)
```

	f	m
1	6	7
2	5	2

Untuk statika, 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 bentuknya yang paling sederhana dapat membaca ini dan mengembalikan kumpulan nilai dan baris judul.

```
>L=readtable(file,>list);
```

Koleksi ini dapat dicetak dengan `writetable()` ke buku catatan, atau ke sebuah 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()` dalam EMT menghitung nilai rata-rata dari baris-baris matriks.

```
>mean (L[1])
```

```
0.40472
0.37102
0.75547
```

File CSV

Pertama, mari kita tulis sebuah matriks ke dalam sebuah file. Untuk keluarannya, kita buat sebuah file di direktori kerja saat ini.

```
>file="test.csv"; ...
>M=random(3,3); writematrix(M,file);
```

Berikut ini adalah isi file ini.

```
>printfile(file)
```

```
0.8221197733097619, 0.821531098722547, 0.7771240608094004
0.8482947121863489, 0.3237767724883862, 0.6501422353377985
0.1482301827518109, 0.3297459716109594, 0.6261901074210923
```

CSV ini dapat dibuka di sistem bahasa Inggris ke Excel dengan klik dua kali. Jika Anda mendapatkan file seperti itu di sistem Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal. Tetapi titik desimal juga merupakan format default untuk EMT. Anda dapat membaca sebuah matriks dari sebuah file dengan readmatrix().

```
>readmatrix(file)
```

```
0.82212  0.82153  0.77712
0.84829  0.32378  0.65014
0.14823  0.32975  0.62619
```

Dimungkinkan untuk menulis beberapa matriks ke dalam satu file. Perintah open() dapat membuka file untuk menulis dengan parameter "w". Standarnya adalah "r" untuk membaca

```
>open(file, "w"); writematrix(M); writematrix(M'); close();
```

Matriks-matriks tersebut dipisahkan oleh sebuah baris kosong. Untuk membaca matriks, 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 matriks sebagai CSV (nilai yang dipisahkan dengan koma). Pada Excel 2007, gunakan "simpan sebagai" dan "format lain", lalu pilih "CSV". Pastikan, tabel saat ini hanya berisi data yang ingin Anda ekspor. Berikut ini 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
```

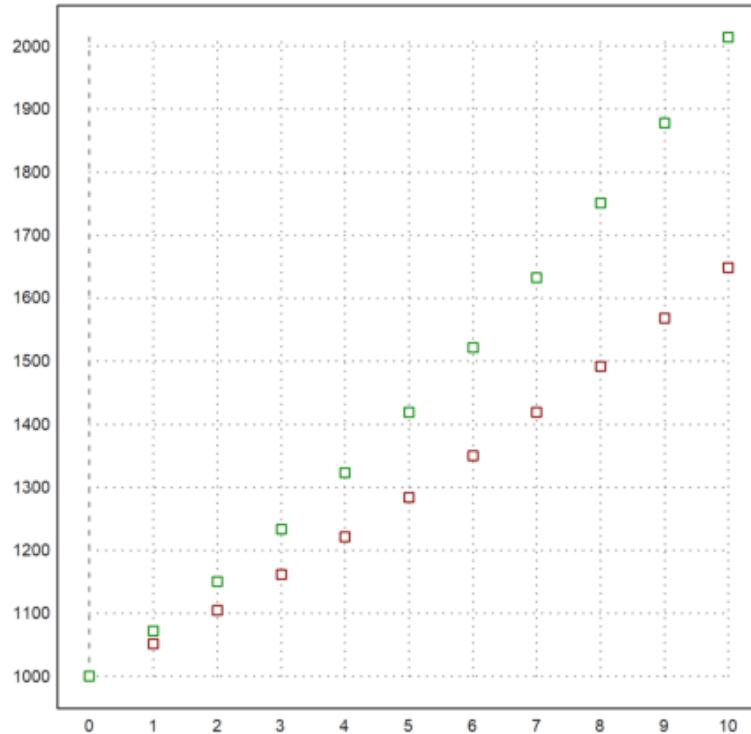
Seperti yang Anda lihat, sistem Jerman saya menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubahnya di pengaturan sistem atau di Excel, tetapi tidak diperlukan untuk membaca matriks ke dalam EMT. Cara termudah untuk membaca ini ke dalam Euler adalah `readmatrix()`. Semua koma digantikan oleh titik-titik dengan parameter `> koma`. Untuk CSV bahasa Inggris, cukup hilangkan 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 rencanakan ini

```
>plot2d(M' [1],M' [2:3],>points,color=[red,green]):
```



Ada beberapa cara yang lebih mendasar untuk membaca data dari file. Anda dapat membuka file dan membaca angka baris demi baris. Fungsi getvectorline() akan membaca angka dari sebuah baris data. Secara default, fungsi ini mengharapkan sebuah titik desimal. Tetapi fungsi ini juga dapat menggunakan koma desimal, jika Anda memanggil setdecimaldot(",") sebelum menggunakan fungsi ini. Fungsi berikut ini adalah contohnya. Fungsi ini akan berhenti di 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
```

Anda juga dapat membaca semua angka dalam file tersebut dengan 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
```

Dengan demikian, sangat mudah untuk menyimpan vektor nilai, satu nilai di 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 jumlah baris dan kolom ke 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 Excel. Untuk membaca berkas 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 Garis

Anda bahkan dapat mengevaluasi setiap baris dengan tangan. Misalkan, kita memiliki baris dengan format berikut ini.

```
>line="2020-11-03,Tue,1'114.05"
```

2020-11-03, Tue, 1'114.05

Pertama, kita dapat memberi tanda pada garis

```
>vt=strtoks(line)
```

2020-11-03
Tue
1'114.05

Kemudian, kita dapat mengevaluasi setiap elemen garis dengan 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, Anda dapat mengekstrak hampir semua informasi dari sebuah baris data. Anggaplah kita memiliki baris dokumen HTML berikut ini.

```
>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, kami menggunakan ekspresi reguler, yang mencari

- tanda kurung penutup >,
- string apa pun yang tidak mengandung tanda kurung dengan sub-cocokan "...",
- braket pembuka dan penutup menggunakan solusi terpendek,
- lagi semua string yang tidak mengandung tanda kurung,
- dan braket pembuka <

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-cocokan.

```
>for k=1:length(vt); vt[k](), end;
```

```
1145.5  
5.6
```

Berikut ini adalah 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
```

Situs web atau file dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris. Dalam contoh, kita membaca versi saat ini dari situs EMT. Kami menggunakan ekspresi reguler untuk memindai "Versi ..." dalam sebuah 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

Masukan dan Keluaran Variabel

Anda dapat menulis variabel dalam bentuk definisi Euler ke file atau ke baris perintah

```
>writevar(pi, "mypi");
```

```
mypi = 3.141592653589793;
```

Untuk pengujian, kami membuat 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
```

Sebagai catatan, jika writevar() digunakan pada sebuah variabel, maka ia 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, kami menambahkan ke file yang telah dibuat 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, gunakan fileremove().

```
>fileremove(file);
```

Sebuah vektor baris dalam sebuah file tidak membutuhkan koma, jika setiap angka berada dalam baris baru. Mari kita buat file seperti itu, menulis setiap baris satu per satu dengan 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]
```