

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].

Seperi 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 menyiapkan baris perintah baru dengan cara menekan tiga tombol bersamaan: [Shift]+[Ctrl]+[Enter].

Komentar (Teks Uraian)

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

```
- * Judul
- ** Sub-Judul
- latex: F (x) = \int_a^x f (t) \, dt
- mathjax: \frac{x^2-1}{x-1} = x + 1
- maxima: integrate(x^3,x) = integrate(x^3,x) + C
- http://www.euler-math-toolbox.de
- See: http://www.google.de | Google
- image: hati.png
- ---
```

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

Judul

Sub-Judul

$$F(x) = \int_a^x f(t) dt$$
$$\frac{x^2 - 1}{x - 1} = x + 1$$

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

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

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

image: hati.png

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

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

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

Baris Perintah

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

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

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

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

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

```
4.90873852123  
7.85398163397
```

Latihan untuk Anda

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

JAWABAN

NAMA : MUHAMMAD LUTFI RAMADHAN

KELAS : MATEMATIKA B

NIM : 23030630021

```
>sin(120)
```

0.580611184212

```
>(980-300+10)/15*3
```

138

```
>(100*15-5^5)^2
```

2640625

```
> 125/10-145/5
```

-16.5

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

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

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

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

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

1.25

Argumen atau input untuk fungsi ditulis di dalam tanda kurung.

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

0.707106781187
-1
0.5

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

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

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

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

```
1.61803398875  
2
```

Latihan untuk Anda

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

JAWABAN

NAMA : MUHAMMAD LUTFI RAMADHAN
KELAS : MATEMATIKA B
NIM : 23030630021

```
>[1,2,5;2,4,7]
```

1	2	5
2	4	7

```
>A=[1,2,3,4,5]
```

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

```
>format(4,0); intrandom(5,5,4), shortformat;
```

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

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

1	5	9
4	5	2
2	8	3
2		



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

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

1609.344

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

```
kilometer$:=1000;  
km$:=kilometer$;  
cm$:=0.01;  
mm$:=0.001;  
minute$:=60;  
min$:=minute$;  
minutes$:=minute$;  
hour$:=60*minute$;  
h$:=hour$;  
hours$:=hour$;  
day$:=24*hour$;  
days$:=day$;  
d$:=day$;  
year$:=365.2425*day$;  
years$:=year$;  
y$:=year$;  
inch$:=0.0254;  
in$:=inch$;
```

```
feet$:=12*inch$;
foot$:=feet$;
ft$:=feet$;
yard$:=3*feet$;
yards$:=yard$;
yd$:=yard$;
mile$:=1760*yard$;
miles$:=mile$;
kg$:=1;
sec$:=1;
ha$:=10000;
Ar$:=100;
Tagwerk$:=3408;
Acre$:=4046.8564224;
pt$:=0.376mm;
```

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

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

```
2.48548476895
101.6 mm
```

Format Tampilan Nilai

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

```
>pi
```

```
3.14159265359
```

```
>longest pi
```

```
3.141592653589793
```

```
>long pi
```

```
3.14159265359
```

```
>short pi
```

```
3.1416
```

```
>shortest pi
```

```
3.1
```

```
>fraction pi
```

```
312689/99532
```

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

```
1612.7
```

```
2.71828182846
```

```
3.141592653589793
```

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

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

```
>longestformat; pi, deformat; pi
```

```
3.141592653589793  
3.14159265359
```

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

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

```
0.392857142857  
11/28
```

Perintah Multibaris

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

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

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

```
-0.138444501319
-3.61155549868
```

Menampilkan Daftar Variabel

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

```
>listvar
```

r	1.25
a	4
b	15
c	2
D	1.73655549868123

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

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

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

```
>listvar m$
```

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

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

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

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

Menampilkan Panduan

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

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

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

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

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

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

```
>random(10)
```

```
[0.270906,  0.704419,  0.217693,  0.445363,  0.308411,  0.914541,  0.193585,
 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 dikerjakan oleh Maxima.

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

$$(b + a)^2$$

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

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

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

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

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

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

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

$$a^2 - b^2$$

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

3628800

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

3628800

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

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

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

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

$$\begin{matrix} 3 & 3 \\ 2 & 5 \end{matrix}$$

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

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

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

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

10

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

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

24

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

$$\cos(x) \cos(y) - \sin(x) \sin(y)$$

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

$$\cos(x) \sin(y) + \sin(x) \cos(y)$$

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

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

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)
```

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

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

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

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

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

```
>&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.

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

Selamat Belajar dan Berlatih!

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

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

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

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

JAWABAN

NAMA : MUHAMMAD LUTFI RAMADHAN
KELAS : MATEMATIKA B
NIM : 23030630021

```
>p1 &= (x^2+4)/(x+1)
```

$$\begin{array}{r} 2 \\ x + 4 \\ \hline x + 1 \end{array}$$

```
>&(a^3-b^3)/(a-b), &ratsimp(%); // ratsimp menyederhanakan bentuk pecahan
```

$$\frac{a^3 - b^3}{a - b}$$

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

Tugas Individu Pekan3-4_Muhammad Lutfi Ramadhan_23030630021

Nama: Muhammad Lutfi Ramadhan
NIM: 23030630021
Kelas: Matematika B 2023

EMT untuk Perhitungan Aljabar

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

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

Contoh pertama

Menyederhanakan bentuk aljabar:

$$5x^{-4}y^3 \times -8x^5y^{-6}$$

```
> $& 5*x^(-4)*y^3*-8*x^5*y^(-6)
```

$$-\frac{40x}{y^3}$$

Menyederhanakan fungsi :

```
> $& 5*y^2+3*x^6-7*y^2+2*x^2
```

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

Menjabarkan:

```
> $& showev('expand((5*x^(-4)+y^3)*(-8*x^5-y^(-6))))
```

$$\text{expand}\left(\left(y^3 + 5x^{-4}\right)\left(-y^{-6} - 8x^5\right)\right) = -y^{-6} + 3 - 5x^{-4}y^{-6} - 8x^5y^3 - 40x^{-4} + 5$$

Baris Perintah

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

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

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

37.6991118431

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

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

75.3982236862
150.796447372

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

```
>x := 4;  
>x := cos(x) // nilai cosinus (x dalam radian)
```

-0.653643620864

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

0.793873449226

Jika dua garis terhubung dengan "..." kedua garis akan selalu dieksekusi secara bersamaan.

```
>x := 3.5; ...
x := (x+2/x)/2, x := (x+2/x)/2, x := (x+2/x)/2,
```

2.03571428571
1.50908521303
1.41719571011

Ini juga merupakan cara yang baik untuk menyebarluaskan long command pada dua atau lebih baris. Anda dapat menekan Ctrl+Return untuk membagi garis menjadi dua pada posisi cursor saat ini, atau Ctrl+Back untuk menggabungkan garis.

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

```
>%+ x=4+5; ...
// This line will not be visible once the cursor is off the line
```

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

81

Euler Math Toolbox mendukung loop di baris perintah, selama mereka masuk ke dalam satu baris atau multi-baris. Dalam program, pembatasan ini tidak berlaku, tentu saja. Untuk informasi lebih lanjut lihat pengantar berikut.

```
>x=6; for i=1 to 10; x := (x+2/x)/2, end; // menghitung akar 2
```

3.16666666667
1.89912280702
1.47612029496
1.4155117098
1.41421415763
1.41421356237
1.41421356237
1.41421356237
1.41421356237
1.41421356237

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

```
>x := 2.5; // comments go here before the ...
repeat xnew:=(x+2/x)/2; until xnew~:=x; ...
x := xnew; ...
end; ...
x,
```

1.41421356237

Struktur bersyarat juga berfungsi.

```
>if E^pi>pi^E; then "Halo Rakyatku!", endif;
```

Halo Rakyatku!

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

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

```
>sqrt(sin(45°)/cos(60°))
```

1.189207115

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

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

```
>exp(log(5.7))
```

Sintaks Dasar

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

Untuk menyetel variabel, gunakan "=" atau ":=". Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir/terbaru. Spasi tidak menjadi masalah. Tetapi ruang antara perintah diharapkan untuk ada.

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

```
>g:=10.73; t:=4.2; 1/2*g*t^2
```

94.6386

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk mengetik

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

```
>E^2 * (1 / (9+5*log(0.7)) +4/9)
```

4.30791848586

Untuk menghitung ekspresi rumit seperti

$$\left(\frac{\frac{5}{8} + \frac{7}{6} + 3}{\frac{3}{7} + \frac{5}{9}} \right)^2 \pi$$

Anda harus memasukkannya dalam bentuk baris.

```
>((5/8 + 7/6 + 3) / (3/7 + 5/9))^2 * pi
```

74.4767625442

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

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

```
>2/3+5/7, fraction %
```

1.38095238095
29/21

Perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi terbuat dari operator dan fungsi. Jika diperlukan, hal tersebut harus berisi tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, memasang braket atau tanda kurung adalah ide yang bagus. Perhatikan bahwa EMT menunjukkan tanda kurung buka dan tutup saat mengedit baris perintah.

```
>(cos(pi/4)+2)^3*(sin(pi/4)+5)^2
```

646.172032434

Operator numerik Euler meliputi

```
+ unary atau operator plus
- unary atau operator minus
* operator perkalian
/ operator pecahan
. produk matriks
a^b daya untuk positif a atau bilangan bulat b (a**b juga berfungsi)
n! operator faktorial
```

dan masih banyak lagi.

Berikut adalah beberapa fungsi yang mungkin Anda butuhkan. Ada banyak lagi.

```
sin, cos, tan, atan, asin, acos, rad, deg
log, exp, log10, sqrt, logbase
bin, logbin, logfac, mod, lantai, ceil, bulat, abs, tanda
conj, re, im, arg, conj, nyata, kompleks
```

```
beta, betai, gamma, complexgamma, ellrf, ellf, ellrd, elle  
bitand, bitor, bitxor, bitnot
```

Beberapa perintah memiliki alias, mis. ln untuk log.

```
>ln(E^4), arctan(tan(0.75)), logbase(30,10)
```

```
4  
0.75  
1.47712125472
```

```
>sin(60°)
```

```
0.866025403784
```

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

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

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

Bilangan Asli

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

```
>longest(23/3)
```

```
7.666666666666667
```

Representasi ganda internal membutuhkan 8 byte.

Representasi ganda adalah format penyimpanan untuk floating-point yang menggunakan 64 bit(8 byte)

```
>printdual(23/3)
```

```
1.1110101010101010101010101010101010101010101010101010101011*2^2
```

```
>printhex(1/7)
```

```
2.4924924924924*16^-1
```

Perbedaan 'printdual' dan 'printhex' adalah 'printdual' yakni mencetak representasi internal dari sebuah bilangan floating-point dalam format presisi ganda (pendekatan yang sangat dekat dengan nilai aslinya tetapi tidak persis sama.) meskipun ia tergantung pada konteks bahasa pemrograman tertentu. sedangkan 'printhex' yakni representasi dari nilai floating-point dalam bentuk heksadesimal(basis 16), heksadesimal ini adalah cara yang lebih ringkas untuk menampilkan nilai biner karena setiap digit heksadesimal mempresentasikan empat digit biner.

String

Sebuah string dalam 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 itu.

```
>"Terjadi Gempa Mag pada hari Senin 26 Agustus 2024 dengan pusat gempa berada di laut " +95+ " km barat c
```

```
Terjadi Gempa Mag pada hari Senin 26 Agustus 2024 dengan pusat gempa berada di laut 95 km barat daya
```

Pada String fungsi print mengonversi angka menjadi string. Ini dapat mengambil sejumlah digit dan sejumlah tempat (0 untuk keluaran padat), dan secara optimal satu unit

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

```
Golden Ratio : 1.61803
```

Terdapat spesial string 'none', yang tidak dicetak.

```
>none
```

Untuk mengonversi string menjadi angka, cukup mengevaluasinya. Ini bekerja untuk ekspresi juga (lihat dibawah).

```
>"1234.5567"()
```

```
1234.5567
```

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

```
>v:= ["Indonesia","Malaysia","Brunei Darussalam"]
```

```
Indonesia  
Malaysia  
Brunei Darussalam
```

Vektor pada string kosong dilambangkan dengan [none]. Dan vektor string dapat digabungkan dengan |".

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

```
Indonesia  
Malaysia  
Brunei Darussalam  
Indonesia  
Malaysia  
Brunei Darussalam
```

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"&beta; = " + 90 + u"&deg; " // pdfLaTeX mungkin gagal menampilkan secara benar
```

```
β = 90°
```

Dalam komentar, entitas yang sama seperti alpha; beta; dll dapat digunakan untuk lateks.
Ada beberapa fungsi untuk membuat atau menganalisis string unicode.
Fungsi strtochr() 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]
```

Perintah ini menghasilkan array atau daftar angka berupa vektor angka yang mewakili karakter dalam string dalam bentuk kode Unicode.
Fungsi kebalikannya adalah chartoutf().

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

```
Ã is a German letter
```

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

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

```
We have α=β.
```

Memungkinkan juga untuk menggunakan entitas numerik.

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

```
Älphabet
```

Nilai Boolean

Nilai boolean direpresentasikan dengan 1=true atau 0=false dalam Euler. String dapat dibandingkan, seperti halnya angka.

```
>"saya">"aku", 6==3
```

```
1  
0
```

```
>5>1, "mobil"=="motor"
```

```
1  
0
```

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

```
>2<E || E<3
```

```
1
```

```
>6>E && E<2
```

```
0
```

Operator Boolean mematuhi aturan bahasa matriks

```
>(2:9)>3, nonzeros (%)
```

```
[0, 0, 1, 1, 1, 1, 1]  
[3, 4, 5, 6, 7, 8]
```

Kita dapat menggunakan fungsi bukan nol() untuk mengekstrak elemen tertentu dari vektor. Dalam contoh, menggunakan isprime bersyarat(n).

```
>N=5 | 7:2:50 // N berisi elemen 5 dan bilangan bilangan ganjil dari 7:50
```

```
[5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33,  
35, 37, 39, 41, 43, 45, 47, 49]
```

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

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

Output Formats

Default output formats EMT adalah 12 digit. Untuk memastikan yang kita lihat adalah bentuk default, maka perlu direset format.

```
>defformat; pi
```

```
3.14159265359
```

Secara internal, EMT menggunakan standar IEEE (Institute of Electrical and Electronics Engineers) untuk bilangan ganda dengan sekitar 16 digit desimal. Untuk melihat bentuk digit penuh, gunakan perintah "longestformat" atau gunakan operator "longest" untuk memunculkannya.

```
>longest pi
```

```
3.141592653589793
```

```
>longestformat; pi
```

```
3.141592653589793
```

Berikut ini adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhex(pi)
```

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

Heksadesimal adalah sistem bilangan yang menggunakan basis 16. Di mana angka 0 hingga 9 (untuk mewakili nilai 0 hingga 9) dan huruf A hingga F (untuk mewakili nilai 10 hingga 15).

Format standarnya adalah 12.

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

```
0.33333  
3.14159  
0.84147
```

```
>format(12); 1/7
```

```
0.1428571
```

Format output dapat diubah secara permanen dengan perintah format.

```
>format(12,5); 1/9, pi, cos(1)
```

```
0.11111  
3.14159  
0.54030
```

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 standar untuk skalar adalah 12, tetapi ini dapat diubah.

```
>setscalarformat(7); pi
```

```
3.14159
```

Begitu juga dengan fungsi "longestformat" mengatur format skalar.

```
>longestformat; pi
```

```
3.141592653589793
```

Notes: beberapa format output yang penting.

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

Akurasi internal EMT adalah sekitar 16 digit desimal mengikuti standar dari IEEE. Angka disimpan dalam format internal. Namun, format output EMT dapat diatur secara fleksibel.

```
>longestformat; pi,
```

```
3.141592653589793
```

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

```
3.14159
```

Standarnya adalah defformat().

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

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

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

```
4.934802200544679
```

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

```
>fraction 2+3/10+7/14+21/7
```

29/5

```
>fraction 5*7/2/3*2/5
```

7/3

Digunakan untuk menampilkan ke bentuk pecahan sederhana.

Karena format internal menggunakan cara biner untuk menyimpan angka, nilai 0,1 tidak akan direpresentasikan dengan tepat. Kesalahan bertambah sedikit, sep

```
>longest 0.2+0.25+0.3+0.3+0.25+0.2+0.5-2.6
```

-0.6000000000000001

Perintah ini menunjukkan presisi penuh dari operasi aritmetika yang melibatkan angka-angka kecil, dan bagaimana kesalahan akumulatif bisa muncul dalam perhitungan biner.

Expressions

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 menamakannya "fx" atau "fxy" dll. Ekspresi lebih diutamakan daripada fungsi. Variabel global dapat digunakan dalam evaluasi

```
>k:=5; fx:="pi*k"; fx()
```

15.70796326794897

Ekspresi akan selalu menggunakan global variable, bahkan jika ada variabel dalam fungsi dengan nama yang sama.

```
>fx:="a*cos(x)"; fx(10,a=0.8)
```

-0.671257223261162

Menggunakan parameter yang ditetapkan ke x, y, z, dll. Jika tidak, evaluasi ekspresi dalam fungsi dapat memberikan hasil yang membingungkan bagi pengguna yang memanggil fungsi tersebut.

```
>at:=6; function f(expr,x,at) := expr(x); ...
f("at*x^4",2,3) // computes 6*2^4 not 3*2^4
```

96

Menggunakan global variable pada fungsi, dimana "at" merupakan global variables. Jika ingin menggunakan nilai lain untuk "at" perlu menambahkan "at=value".

Untuk referensi, kami berkomentar bahwa koleksi panggilan (dibahas di tempat lain) dapat berisi ekspresi. Jadi kita bisa membuat contoh di atas sebagai berikut

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

128

Walaupun "ut" sebagai global variable sudah didefinisikan, tetapi didefinisikan kembali pada ekspresi fungsinya dimana "ut=p" sehingga nilainya berganti dari yang awalnya ut=5 menjadi ut=p=2.

```
>f &= x^2
```

$\frac{2}{x}$

```
>function f(x) := x^4
>f(2)
```

16

Ekspresi dalam x sering digunakan seperti fungsi.
Mendefinisikan fungsi dengan nama yang sama seperti ekspresi simbolik global ($f \&=$) menghapus nilai variabel sebelumnya untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

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

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

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

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

```
> "@(a,b) a^3+b^2", % (2,4)
```

$$\begin{matrix} @ (a,b) a^3+b^2 \\ 24 \end{matrix}$$

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 simbolis atau numerik. Jika variabel utama adalah x , ekspresi dapat dievaluasi seperti fungsi. Seperti yang Anda lihat dalam contoh berikut, variabel global terlihat selama evaluasi.

```
>fx &= 2*x-3*t; ...  
t=2.5; fx(0.8)
```

$$-5.9$$

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

```
>fx(1,t=1.5)
```

$$-2.5$$

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

Symbolic Mathematics

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

```
>$&35!
```

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

Mari kita hitung!

$$C(35,15) = \frac{35!}{20! \cdot 15!}$$

```
>$& 35!/(20!*15!) // nilai C(35,15)
```

$$3247943160$$

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

```
>$binomial(35,15) //menghitung C(35,15) menggunakan fungsi binomial()
```

$$3247943160$$

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

Anda akan belajar bahwa yang berikut ini juga berfungsi.

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

```
>$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=5 // substitusi x=5 ke C(x,3)}
```

10

Dengan begitu Anda dapat menggunakan solusi persamaan dalam persamaan lain. Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Alasan untuk ini adalah bendera simbolik khusus dalam string. Seperti yang akan Anda lihat pada contoh sebelumnya dan berikut, jika Anda telah menginstal LaTeX, Anda dapat mencetak ekspresi simbolis dengan Lateks. Jika tidak, perintah berikut akan mengeluarkan pesan kesalahan. Untuk mencetak ekspresi simbolis dengan LaTeX, gunakan \$ di depan & (atau Anda dapat menghilangkan &) sebelum perintah. Jangan menjalankan perintah Maxima dengan \$, jika Anda tidak menginstal LaTeX.

```
>$ (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 menyertakan ekspresi dalam "...". Untuk menggunakan lebih dari ekspresi sederhana adalah mungkin, tetapi sangat tidak disarankan

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

25

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

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

$$x^4 + 4x^3 + 6x^2 + 4x + 1 \\ 4(x+1)^3$$

Sekali lagi, % mengacu pada hasil sebelumnya.
Untuk mempermudah, kami menyimpan solusi ke variabel simbolik. Variabel simbolik didefinisikan dengan "&=".

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

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

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

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

2432902008176640000

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

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

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

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

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

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

2
g

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

$$\frac{x^3}{e}$$

$$x^3 e^x$$

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

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

$$\frac{125}{e^5}$$

$$125 e^5$$

$$18551.64488782208$$

```
>fx(5)
```

$$18551.6448878$$

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

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

$$\frac{1000}{e^{10}} - \frac{125}{e^5}$$

$$2.20079141499189 \times 10^7$$

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

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

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

```
>tex(fx)
```

$$x^3 e^x$$

Ekspresi simbolik dapat dievaluasi seperti ekspresi numerik.

```
>fx(0.5)
```

$$0.206090158838$$

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

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

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

Penugasan juga bisa bersifat simbolis.

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

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

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

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

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

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

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

1.56155281281

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

```
>sol &= solve(x^2+3*x=9,x); $&sol, sol(), $&float(sol)
```

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

[-4.8541, 1.8541]

$[x = -4.854101966249685, x = 1.854101966249685]$

Untuk mencetak ekspresi simbolis dengan LaTeX, gunakan \$ di depan & (atau dapat menghilangkan &) sebelum perintah. Jangan menjalankan perintah Maxima dengan \$, jika tidak menginstal LaTeX.

Untuk mendapatkan solusi simbolis tertentu, seseorang dapat menggunakan "with" dan index.

```
>$&solve(x^2+3*x=1,x), x2 &= x with %[2]; $&x2
```

$$\left[x = \frac{-\sqrt{13}-3}{2}, x = \frac{\sqrt{13}-3}{2} \right]$$

$$\frac{\sqrt{13}-3}{2}$$

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

```
>sol &= solve([x+y=5,x^2+y^2=15],[x,y]); $&sol, $&x*y with sol[1]
```

$$\left[\left[x = \frac{5-\sqrt{5}}{2}, y = \frac{\sqrt{5}+5}{2} \right], \left[x = \frac{\sqrt{5}+5}{2}, y = \frac{5-\sqrt{5}}{2} \right] \right]$$
$$\frac{(5-\sqrt{5})(\sqrt{5}+5)}{4}$$

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

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

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

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

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

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

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

Functions

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

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

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

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

```
>f(2)
```

4.472135955

Fungsi ini akan bekerja untuk vektor juga, dengan mematuhi bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi divektorkan.

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

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

Fungsi dapat diplot. Ailih-alih ekspresi, kita hanya perlu memberikan nama fungsi. Berbeda dengan ekspresi simbolik atau numerik, nama fungsi harus diberikan dalam string.

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

0.786151377757

Secara default, jika Anda perlu menimpa fungsi bawaan, Anda harus menambahkan kata kunci "overwrite". Overwriting fungsi bawaan berbahaya dan dapat menyebabkan masalah untuk fungsi lain tergantung pada fungsi tersebut.

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

```
>function overwrite sin(x) := _sin(x°) // tentukan kembali sinus dalam derajat  
>sin(45)
```

0.707106781187

Jika ingin menghapus definisi dari sin dan mendefinisikannya ulang, menggunakan perintah "forget"

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

0.707106781187

Default Parameters

Parameter default adalah fungsi parameter yang memiliki nilai awal.
Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

16

Menimpa default value.

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

80

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

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

16

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

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

24

Tetapi parameter yang ditetapkan menimpa global value.

Jika argumen tidak ada dalam daftar parameter yang telah ditentukan sebelumnya, argumen tersebut

harus dideklarasikan dengan ":="!

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

20

Fungsi simbolis didefinisikan dengan "\$&=". Fungsi simbolis didefinisikan dalam Euler dan maxima,dan bekerja keduanya. Ekspresi yang mendefinisikan dijalankan melalui Maxima sebelum di 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 //1. turunan pertama dari g(x), 2. memasukkan nilai x=4/3
```

$$\begin{aligned} &x e^{-x} - e^{-x} + 3 x^2 \\ &\frac{e^{-\frac{4}{3}}}{3} + \frac{16}{3} \end{aligned}$$

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

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

178.635099908

Itu juga dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik lainnya.

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

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

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

0.703467422498

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

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

0.703467422498

Dengan &= fungsinya simbolis, dan dapat digunakan dalam ekspresi simbolik lainnya. Contohnya dalam integral tak tentu sebagai berikut.

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

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

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

$$(x + 2)^n$$

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

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

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

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

625

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

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

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

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

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

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

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

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

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

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

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

```
>S&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^3-2; $&f(x)
```

$$x^3 - 2$$

Dengan &= maka fungsi adalah simbolik, dan dapat digunakan di ekspresi simbolik lainnya.

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

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

Dengan := fungsinya numerik. Contoh yang baik adalah integral tak tentu seperti

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

yang tidak dapat dinilai secara simbolis.

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

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

Sekarang fungsi dapat dipanggil dengan menggunakan suatu parameter "base" maupun tidak.

```
>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 elemen individual di tempat lain. Ini tepat terjadi dengan vektor parameter.

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

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

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

Fungsi simbolik seperti itu dapat digunakan untuk variabel simbolik. etapi 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 kedua
```

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

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

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

0

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

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

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

Ringkasam:

- &= mendefinisikan fungsi simbolis,
- := mendefinisikan fungsi numerik,
- &&= mendefinisikan fungsi simbolis murni

Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolis.

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

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

1.41421356237

Ini juga berfungsi untuk fungsi simbolik, perhatikan fungsi berikut ini.

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

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

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

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

```
>$&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])
```

$$\left[\left[x = \frac{bf - ce}{bd - ae}, y = \frac{cd - af}{bd - ae}\right]\right]$$

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.

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

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

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

0.966715594851
2

Memecahkan ekspresi simbolis dalam bentuk simbolis mengembalikan daftar 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 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

Memecahkan sistem persamaan secara simbolis dapat dilakukan dengan vektor persamaan dan solver simbolis solve(). Hadilnya dalam bentuk persamaan.

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

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

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

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 dengan nama fungsi dan parameternya (caranya lainnya adalah parameter titik koma).

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

```
-0.710242150858
```

Ini juga bekerja dengan ekspresi. Tapi daftar elemen yang ada harus digunakan.

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

```
0.0329709716756
```

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.

Eliminasi Fourier adalah analog dari eliminasi Gauss untuk linear (persamaan atau pertidaksamaan). Panggilan fungsi 'fourier_elim([eq1, eq2, ...], [var1, var2, ...])' melakukan eliminasi Fourier eliminasi pada pertidaksamaan linear '[eq1, eq2, ...]' dengan berkenaan dengan variabel '[var1, var2, ...]'; sebagai contoh

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

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

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

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

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

```
[1 < x] ∨ [x < -1]
```

```
>$&fourier_elim([x^2 - 4<0], [x])
```

$$[-2 < x, x < 2]$$

```
>$&fourier_elim([x^2 - 9# 0], [x])
```

$$[-3 < x, x < 3] \vee [3 < x] \vee [x < -3]$$

```
>$&fourier_elim([x # 10], [x])
```

$$[x < 10] \vee [10 < x]$$

Ketika himpunan penyelesaiannya adalah kosong maka 'emptyset', dan ketika himpunan penyelesaiannya adalah semua bilangan real, maka 'universalset'; sebagai contoh

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

universalset

```
>$&fourier_elim([x < 1, x > 1], [x])
```

emptyset

Untuk persamaan nonlinier, 'fourier_elim' mengembalikan sebuah daftar persamaan yang disederhanakan:

```
>$&fourier_elim([x^3 - 8 > 0], [x])
```

$$[2 < x, x^2 + 2x + 4 > 0] \vee [x < 2, -x^2 - 2x - 4 > 0]$$

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

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

Alih-alih sebuah daftar pertidaksamaan, 'fourier_elim' juga dapat berupa disjungsi atau konjungsi logika:

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

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

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

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

Fungsi 'fourier_elim' mendukung operator pertidaksamaan ' $<$, \leq , $>$, \geq ', '#', dan '='. Kode eliminasi Fourier memiliki sebuah preprocessor yang mengubah beberapa persamaan nonlinier yang melibatkan nilai absolut, minimum, dan fungsi maksimum menjadi linear dalam persamaan. Selain itu, preprocessor menangani beberapa ekspresi yang merupakan hasil kali atau hasil bagi dari suku-suku linier:

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

$$[6 < x, x < 8, y < -11] \vee [8 < x, y < -11] \vee [x < 8, 13 < y] \vee [x = y, 13 < y] \vee [8 < x, x < y, 13 < y] \vee [y < x, 13 < y]$$

```
>$&fourier_elim([(x+2)/(x-4) <= 2], [x])
```

$$[x = 10] \vee [10 < x] \vee [x < 4]$$

Bahasa Matriks

Dalam matematika, matriks adalah susunan[1] bilangan, simbol, atau ekspresi yang disusun dalam baris dan kolom sehingga membentuk suatu bangun persegi. Vektor dan matriks dimasukkan dengan tanda kurung siku, elemen dipisahkan dengan koma, baris dipisahkan dengan titik koma.

Matriks 1x2

```
>a=[1;2]
```

```
>b=[3,4;5,6]
```

$$\begin{matrix} 3 & 4 \\ 5 & 6 \end{matrix}$$

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

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

Transpose matriks adalah matriks baru yang diperoleh dengan cara menukar elemen-elemen baris menjadi elemen kolom atau sebaliknya.

```
>a'
```

$$\begin{bmatrix} 1, & 2 \end{bmatrix}$$

```
>b'
```

$$\begin{matrix} 3 & 5 \\ 4 & 6 \end{matrix}$$

```
>c'
```

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

Invers matriks adalah matriks baru yang merupakan kebalikan dari matriks asal

```
>inv(b)
```

$$\begin{matrix} -3 & 2 \\ 2.5 & -1.5 \end{matrix}$$

Perkalian matriks sendiri adalah proses mengalikan setiap elemen baris pada matriks pertama dengan elemen kolom pada matriks kedua.

```
>b.a
```

$$\begin{matrix} 11 \\ 17 \end{matrix}$$

Perkalian dari matriks dengan invers matriks itu sendiri akan menghasilkan matriks identitas

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

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

Perkalian matriks dan perpangkatan matriks

```
>b.b
```

$$\begin{matrix} 29 & 36 \\ 45 & 56 \end{matrix}$$

```
>b^2
```

$$\begin{matrix} 9 & 16 \\ 25 & 36 \end{matrix}$$

```
>b.b.b
```

$$\begin{matrix} 267 & 332 \\ 415 & 516 \end{matrix}$$

```
>power(b,3)
```

```
267      332  
415      516
```

Pembagian matriks

```
>a/a
```

```
1  
1
```

```
>a/b
```

```
0.333333  0.25  
0.4        0.333333
```

Perkalian invers matriks dengan matriks lainnya

```
>b\`a
```

```
1  
-0.5
```

```
>inv(b).a
```

```
1  
-0.5
```

Perkalian skalar

```
>b*2
```

```
6      8  
10     12
```

```
>2*b
```

```
6      8  
10     12
```

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

```
[2, 4]
```

Fungsi Matriks Lainnya

Untuk membangun matriks, kita dapat menumpuk satu matriks di atas yang lain. 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
```

```
>A=random(3,4)
```

```
0.525184  0.502255  0.168603  0.262253  
0.866587  0.536137  0.493453  0.601344  
0.659461  0.967468  0.193151  0.935921
```

```
>A|1
```

```
0.525184  0.502255  0.168603  0.262253  1  
0.866587  0.536137  0.493453  0.601344  1  
0.659461  0.967468  0.193151  0.935921  1
```

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

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

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

1	2	3
1	2	3

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

1	1
2	2
3	3

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

1	1
2	4
3	9

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

9

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

1	1
1	1

```
>zeros(2,2)
```

0	0
0	0

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

[6, 6, 6, 6, 6]

```
>random(1,2)
```

[0.0728753, 0.988966]

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

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

1	2	3
4	5	6
7	8	9

```
>function rep(v,n) := redim(dup(v,n),1,n*cols(v))  
>rep(1:3,5)
```

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

```
>multdup(1:3,3)
```

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

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

Keduanya tidak memiliki jumlah kolom yang sama, kolom yang lebih pendek akan diisi dengan 0.

```
>flipx(1:5)
```

[5, 4, 3, 2, 1]

```
>rotleft(1:5)
```

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

```
>rotright(1:5)
```

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

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

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

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

Ada beberapa fungsi khusus untuk mengatur diagonal atau untuk menghasilkan matriks diagonal. Kita mulai dengan matriks identitas

```
>A=id(3)
```

1	0	0
0	1	0
0	0	1

Vektorisasi

Hampir semua fungsi di Euler juga berfungsi untuk input matriks dan vektor, kapan pun ini masuk akal. Misalnya, fungsi sqrt() menghitung akar kuadrat dari semua elemen vektor atau matriks.

```
>sqrt(1:4)
```

```
[1, 1.41421, 1.73205, 2]
```

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

```
>x=3:0.05:6; y=log(x)
```

```
[1.09861, 1.11514, 1.1314, 1.1474, 1.16315, 1.17865, 1.19392,
1.20896, 1.22378, 1.23837, 1.25276, 1.26695, 1.28093, 1.29473,
1.30833, 1.32176, 1.335, 1.34807, 1.36098, 1.37372, 1.38629,
1.39872, 1.41099, 1.42311, 1.43508, 1.44692, 1.45862, 1.47018,
1.4816, 1.4929, 1.50408, 1.51513, 1.52606, 1.53687, 1.54756,
1.55814, 1.56862, 1.57898, 1.58924, 1.59939, 1.60944, 1.61939,
1.62924, 1.639, 1.64866, 1.65823, 1.66771, 1.6771, 1.6864,
1.69562, 1.70475, 1.7138, 1.72277, 1.73166, 1.74047, 1.7492,
1.75786, 1.76644, 1.77495, 1.78339, 1.79176]
```

Dengan ini dan operator titik dua a:delta:b, vektor nilai fungsi dapat dihasilkan dengan mudah. Pada contoh berikut, kita membangkitkan vektor nilai t[i] dengan spasi 0,1 dari -1 hingga 3. Kemudian kita membangkitkan vektor nilai fungsi.
lateks:s=t^3-t

```
>t=-1:0.1:3; 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, 0.231, 0.528, 0.897, 1.344, 1.875,
2.496, 3.213, 4.032, 4.959, 6, 7.161, 8.448, 9.867, 11.424,
13.125, 14.976, 16.983, 19.152, 21.489, 24]
```

```
>shortest (1:7)*(1:7)'
```

1	2	3	4	5	6	7
2	4	6	8	10	12	14
3	6	9	12	15	18	21
4	8	12	16	20	24	28
5	10	15	20	25	30	35
6	12	18	24	30	36	42
7	14	21	28	35	42	49

Perhatikan, bahwa ini sangat berbeda dari produk matriks. Produk matriks dilambangkan dengan titik "." di EMT.

```
>(1:7).(1:7)'
```

Secara default, vektor baris dicetak dalam format yang ringkas.

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

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

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

```
>v:=[2,3]; v.v', %^2
```

```
13  
169
```

Untuk mentranspos matriks kita menggunakan apostrof

```
>v=3:6; v'
```

```
3  
4  
5  
6
```

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

```
50
```

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

```
>v'.v
```

9	12	15	18
12	16	20	24
15	20	25	30
18	24	30	36

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

```
>v.v'
```

```
86
```

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

```
>norm(v)^3
```

```
797.531190613
```

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ringkasan aturannya.

- Fungsi yang diterapkan ke vektor atau matriks diterapkan ke setiap elemen.
 - Operator yang beroperasi pada dua matriks dengan ukuran yang sama diterapkan berpasangan ke elemen matriks.
 - jika kedua matriks memiliki dimensi yang berbeda, keduanya diperluas dengan cara yang masuk akal, sehingga memiliki ukuran yang sama.
- Misalnya, nilai skalar kali vektor mengalikan nilai dengan setiap elemen vektor. Atau matriks kali vektor (dengan *, bukan.) memperluas vektor ke ukuran matriks dengan menduplikasikan.
- Berikut ini adalah kasus sederhana dengan operator^.

```
>[2,3,6]^2
```

```
[4, 9, 36]
```

Berikut adalah kasus yang lebih rumit. Vektor baris dikalikan dengan vektor kolom mengembang keduanya dengan menduplikasi.

```
>v:=[2,3,6]; v*v'
```

4	6	12
6	9	18
12	18	36

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan *!

```
>v.v'
```

49

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

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

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

```
[4, 9, 36]  
49  
[4, 13, 49]
```

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

```
>3:9, 0:2:8
```

```
[3, 4, 5, 6, 7, 8, 9]  
[0, 2, 4, 6, 8]
```

```
>[1,2] | [3,4,5], [1,2]_5
```

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

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

```
>A:=[4,5,6,7;8,9,10,11]; A[2,2]
```

9

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

```
>v:=[1,3,5,7]; v[1], A[1]
```

```
1  
[4, 5, 6, 7]
```

Indeks juga bisa menjadi vektor baris dari indeks. : menunjukkan semua indeks.

```
>v[2:4], A[:,2]
```

```
[3, 5, 7]  
5  
9
```

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

```
>A[,2:4]
```

```
5 6 7  
9 10 11
```

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

```
>A{5}
```

8

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

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

```
[4, 5, 6, 7, 8, 9, 10, 11]
[4, 5, 6, 7, 8, 9, 10, 11]
```

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

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

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

Sekarang kita menambahkan kolom ke matriks.

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

```
0      0      1      0
45    0.785398  0.707107  0.0137074
90    1.5708    0          0.0274121
135   2.35619   -0.707107  0.0411118
180   3.14159   -1        0.0548037
225   3.92699   -0.707107  0.0684853
270   4.71239   0          0.082154
315   5.49779   0.707107  0.0958073
360   6.28319   1        0.109443
```

Dengan menggunakan bahasa matriks, kita dapat menghasilkan beberapa tabel dari beberapa fungsi sekali^ogus.

Dalam contoh berikut, kita menghitung t_{ij}^n untuk i dari 1 hingga n . Kami mendapatkan matriks, di mana setiap baris adalah tabel t_{ij}^n untuk satu i . Yaitu, matriks memiliki elemen lateks: $a_{\{i,j\}} = t_{ij}^n$, $\forall i \in 1 \text{ to } 10$, $\forall j \in 1 \text{ to } n$

Fungsi yang tidak berfungsi untuk input vektor harus "divektorkan". Ini dapat dicapai dengan kata kunci "peta" dalam definisi fungsi. Kemudian fungsi tersebut akan dievaluasi untuk setiap elemen dari parameter vektor.

Integrasi numerik terintegrasi() hanya berfungsi untuk batas interval skalar. Jadi kita perlu membuat vektor.

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

Kata kunci "peta" membuat vektor fungsi. Fungsinya sekarang akan bekerja untuk vektor bilangan.

```
>f([1:3])
```

```
[0, 2.05045, 13.7251]
```

Sub-Matriks dan Matriks-Elemen

Untuk mengakses elemen matriks, gunakan notasi braket.

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

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

6

Kita dapat mengakses satu baris matriks yang lengkap.

```
>A[1]
```

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

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

```
>v=1:8; v[2]
```

2

Untuk memastikan, Anda mendapatkan baris pertama untuk matriks $1 \times n$ dan $m \times n$, tentukan semua kolom menggunakan indeks kedua kosong.

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

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

Jika indeks adalah vektor indeks, Euler akan mengembalikan baris matriks yang sesuai. Di sini kita ingin baris pertama dan kedua dari A.

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

4	5	6
7	8	9

Kita bahkan dapat menyusun ulang A menggunakan vektor indeks. Tepatnya, kami tidak mengubah A disini, tetapi menghitung versi A yang disusun ulang.

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

7	8	9
4	5	6
1	2	3

Trik indeks bekerja dengan kolom juga.

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

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

2	3
5	6

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

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

2
5
8

Atau, biarkan indeks pertama kosong.

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

1	2	3
4	5	6
7	8	9

Kita juga bisa mendapatkan baris terakhir dari A.

```
>A[3]
```

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

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

```
>A[2,3]=9
```

1	2	3
4	5	9
7	8	9

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

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

4	5	3
6	7	9
7	8	9

Selain itu, beberapa jalan pintas diperbolehkan.

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

```
-1          -1          3
-1          -1          9
 7           8           9
```

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

```
>A[5]
```

```
Row index 5 out of bounds!
Error in:
A[5] ...
^
```

Menyortir dan Mengacak

Fungsi `sort()` mengurutkan vektor baris.

```
>sort([2,9,5,7,3,1])
```

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

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

```
>v=shuffle(1:8)
```

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

Indeks berisi urutan yang tepat dari v.

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

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

```
>s=["d","f","c","b","aa","g"]
```

```
d
f
c
b
aa
g
```

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

```
aa
b
c
d
f
g
```

```
>ind
```

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

Fungsi `unique` mengembalikan daftar elemen unik vektor yang diurutkan.

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

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

Ini bekerja untuk vektor string juga.

```
>intrandom(1,5,10), unique(%)
```

```
[3, 1, 4, 3, 10]
[1, 3, 4, 10]
```

```
>unique(s)
```

```
aa  
b  
c  
d  
f  
g
```

Aljabar Linier

EMT memiliki banyak fungsi untuk menyelesaikan sistem linier, sistem sparse, atau masalah regresi. Untuk sistem linier $Ax=b$, dengan A adalah matriks koefisien, x adalah vektor solusi yang ingin kita cari dan b adalah vektor hasil yang diberikan.

Anda dapat menggunakan algoritma Gauss, matriks invers atau kecocokan linier.

Operator $\backslash b$ menggunakan versi algoritma Gauss.

Operator backslash \ digunakan untuk menyelesaikan sistem persamaan linier ini. Ketika menulis $A\backslash b$,

perangkat lunak akan menghitung solusi yang memenuhi persamaan $Ax=b$.

Operator ini secara otomatis menggunakan algoritma eliminasi Gauss atau metode numerik serupa untuk menemukan solusi.

```
>A=[5,6;7,8]; b= [4;3]; A\b
```

```
-7  
6.5
```

Untuk contoh lain, kami membuat matriks 100×100 dan jumlah barisnya. Kemudian kita selesaikan $Ax=b$ menggunakan matriks invers. Kami mengukur kesalahan sebagai deviasi maksimal semua elemen dari 1, yang tentu saja merupakan solusi yang benar.

```
>A=normal(100,100); b=sum(A); longest totalmax(abs(inv(A).b-1))
```

```
2.873257187729905e-13
```

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

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

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

```
>det(A)
```

```
-34
```

Matriks Simbolik

Maxima memiliki matriks simbolis. Tentu saja, Maxima dapat digunakan untuk masalah aljabar linier sederhana seperti itu.

Kita dapat mendefinisikan matriks untuk Euler dan Maxima dengan &:=, dan kemudian menggunakannya dalam ekspresi simbolis. Bentuk [...] biasa untuk mendefinisikan matriks dapat digunakan di Euler untuk mendefinisikan matriks simbolik.

```
>A &= [a,1,1;1,a,1;1,1,a]; $A
```

$$\begin{pmatrix} a & 1 & 1 \\ 1 & a & 1 \\ 1 & 1 & a \end{pmatrix}$$

```
> $&det(A), $&factor(%)
```

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

```
>$&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)
```

$$(1 - x) (2 - x) - ab$$

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

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

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

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

Untuk mengekstrak vektor eigen tertentu perlu 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 seperti ekspresi simbolik lainnya.

```
>A(a=6,b=7)
```

$$\begin{matrix} 1 & 6 \\ 7 & 2 \end{matrix}$$

Dalam ekspresi simbolik, gunakan dengan.

```
>$&A with [a=6,b=7]
```

$$\begin{pmatrix} 1 & 6 \\ 7 & 2 \end{pmatrix}$$

Akses ke baris matriks simbolik bekerja seperti halnya dengan matriks numerik.

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

$$[1, a]$$

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

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

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

Ada fungsi simbolik di Maxima untuk membuat vektor dan matriks. Untuk ini, lihat dokumentasi Maxima atau tutorial tentang Maxima di EMT.

```
>v &= makelist(1/(i+j), i, 1, 3); $v
```

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

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

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

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

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

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

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

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

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

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

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

Misalnya, nilai eigen dari `A` dapat dihitung secara numerik.

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

$$[16.1168, -1.11684, 0]$$

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

```
>$&eigenvalues(@A)
```

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

Nilai Numerik dalam Ekspresi simbolis

Ekspresi simbolis hanyalah string yang berisi ekspresi. Jika kita ingin mendefinisikan nilai baik untuk ekspresi simbolik maupun 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 simbolik. Saat mentransfer matriks ke bentuk simbolis, pendekatan fraksional untuk real akan digunakan.

```
>$&A
```

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

Untuk menghindarinya, ada fungsi "`mxmset(variable)`".

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

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

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

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

$$1.4142135623730950488016887242097B \times 10^0$$

$$1.414213562373095$$

Ketepatan angka floating point besar dapat diubah

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

$$3.14159265358979323846264338327950288419716939937510582097494 \\ 4592307816406286208998628034825342117068b0$$

Variabel numerik dapat digunakan dalam ekspresi simbolis apa pun menggunakan "@var". Perhatikan bahwa ini hanya diperlukan, jika variabel telah didefinisikan dengan ":" atau "=" sebagai variabel numerik.

```
>B:=[1,pi;3,4]; $&det (@B)
```

```
-5.424777960769379
```

Demo - Suku Bunga

Di bawah ini, kami menggunakan Euler Math Toolbox (EMT) untuk perhitungan suku bunga. Kami melakukannya secara numerik dan simbolis untuk menunjukkan kepada Anda bagaimana Euler dapat digunakan untuk memecahkan masalah kehidupan nyata.

Asumsikan Anda memiliki modal awal 4000 (katakanlah dalam dolar).

```
>M=4000
```

```
4000
```

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

```
>M*1.03
```

```
4120
```

Euler akan memahami sintaks berikut juga.

```
>M+M*3%
```

```
4120
```

Tetapi lebih mudah menggunakan faktornya.

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

```
1.03  
4120
```

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

```
>M*q^10
```

```
5375.66551738
```

Untuk tujuan kita, kita dapat mengatur format menjadi 2 digit setelah titik desimal.

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

```
5375.67
```

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

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

```
Starting from 4000$ you get 5375.67$.
```

Bagaimana jika kita ingin mengetahui hasil antara dari tahun 1 sampai tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak harus menulis loop, tetapi cukup masukkan

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

```
Real 1 x 11 matrix
```

```
4000.00 4120.00 4243.60 4370.91 ...
```

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

```
>short 0:10
```

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

Kemudian semua operator dan fungsi dalam Euler dapat diterapkan pada elemen vektor untuk elemen.

```
>short q^(0:10)
```

```
[1, 1.03, 1.0609, 1.0927, 1.1255, 1.1593, 1.1941, 1.2299,  
1.2668, 1.3048, 1.3439]
```

adalah vektor faktor q^0 sampai q^{10} . Ini dikalikan dengan M, dan kami mendapatkan vektor nilai.

```
>VM=M*q^(0:10)
```

```
Real 1 x 11 matrix  
4000.00 4120.00 4243.60 4370.91 ...
```

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

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

Mari kita bandingkan dua hasil, dengan dan tanpa pembulatan.

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

```
1271.61  
1271.6071
```

Sekarang tidak ada rumus sederhana untuk tahun ke-n, dan kita harus mengulang selama bertahun-tahun.

Euler memberikan banyak solusi untuk ini.

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

```
>VMr=iterate("oneyear",4000,10)
```

```
Real 1 x 11 matrix  
4000.00 4120.00 4243.60 4370.91 ...
```

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

```
>VMr'
```

```
4000.00  
4120.00  
4243.60  
4370.91  
4502.04  
4637.10  
4776.21  
4919.50  
5067.09  
5219.10  
5375.67
```

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

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

```
4120.00  
4000.00 4120.00 4243.60
```

Anehnya, kita juga bisa menggunakan vektor indeks. Ingat bahwa 1:3 menghasilkan vektor [1,2,3]. Mari kita bandingkan elemen terakhir dari nilai yang dibulatkan dengan nilai penuh.

```
>VMr[-2], VM[-2]
```

```
5219.10  
5219.09
```

Perbedaannya sangat kecil.

Memecahkan Persamaan

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

```
>function onepay (M) := M*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",4000,10)
```

```
Real 1 x 11 matrix  
4000.00 4320.00 4649.60 4989.09 ...
```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

```
>R=-200; iterate("onepay",4000,10)
```

```
Real 1 x 11 matrix  
4000.00 3920.00 3837.60 3752.73 ...
```

Kami melihat bahwa uang berkurang. Jelas, jika kita hanya mendapatkan 150 bunga di tahun pertama,

tetapi menghapus 200, kita kehilangan uang setiap tahun.

Bagaimana kita bisa menentukan berapa tahun uang itu akan bertahan? Kita harus menulis loop untuk ini. Cara termudah adalah dengan iterasi cukup lama.

```
>VMR=iterate("onepay",4000,50)
```

```
Real 1 x 51 matrix  
4000.00 3920.00 3837.60 3752.73 ...
```

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

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

```
32.00
```

Alasan untuk ini adalah bahwa bukan `nol(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 31 tahun.

Fungsi `iterate()` memiliki satu trik lagi. Itu bisa mengambil kondisi akhir sebagai argumen. Kemudian akan mengembalikan nilai dan jumlah iterasi.

```
>(x,n)=iterate("onepay",4000,till="x<0"); x, n
```

```
-0.21  
31.00
```

Mari kita coba menjawab pertanyaan yang lebih ambigu. Asumsikan kita tahu bahwa nilainya adalah 0 setelah 50 tahun. Apa yang akan menjadi tingkat bunga?

Ini adalah pertanyaan yang hanya bisa dijawab dengan angka. Di bawah ini, kita akan mendapatkan formula yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada formula yang mudah untuk tingkat bunga.

Tapi untuk saat ini, kami bertujuan untuk solusi numerik.

Langkah pertama adalah mendefinisikan fungsi yang melakukan iterasi sebanyak n kali. Kami menambahkan semua parameter ke fungsi ini.

```
>function f(M,R,P,n) := iterate("x*(1+P/100)+R",M,n;P,R)[-1]
```

Iterasinya sama seperti di atas.

Tapi kami tidak lagi menggunakan nilai global R dalam ekspresi kami. Fungsi seperti `iterate()` memiliki trik khusus di Euler. Anda dapat meneruskan nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R.

Selain itu, kami hanya tertarik pada nilai terakhir. Jadi kita ambil indeks [-1].

Mari kita coba tes.

```
>f(4000,-200,3,31)
```

```
-0.21
```

Sekarang kita bisa menyelesaikan masalah kita.

```
>solve("f(4000,-200,x,50)",3)
```

Rutin memecahkan memecahkan ekspresi=0 untuk variabel x. Jawabannya adalah 3,15% per tahun. Kami mengambil nilai awal 3% untuk algoritma. Fungsi solve() selalu membutuhkan nilai awal. Kita dapat menggunakan fungsi yang sama untuk menyelesaikan pertanyaan berikut: Berapa banyak yang dapat kita keluarkan per tahun sehingga modal awal habis setelah 20 tahun dengan asumsi tingkat bunga 3% per tahun.

```
>solve("f(4000,x,3,20)",-200)
```

-268.86

Perhatikan bahwa Anda tidak dapat memecahkan jumlah tahun, karena fungsi kami mengasumsikan n sebagai nilai integer.

Solusi Simbolik untuk Masalah Suku Bunga

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

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

$R + qM$

Kita sekarang dapat mengulangi ini.

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

$$\begin{aligned} &q(q(q(R+qM)+R)+R) \\ &q^3R + q^2R + qR + R + q^4M \end{aligned}$$

Kami melihat sebuah pola. Setelah n periode yang kita miliki

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

Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

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

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

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

Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

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

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

Fungsi tersebut melakukan hal yang sama seperti fungsi f kita sebelumnya. Tapi itu lebih efektif.

```
>longest f(4000,-200,3,31), longest fs(4000,-200,3,31)
```

-0.2142542009293322
-0.2142542009369208

Kita sekarang dapat menggunakan untuk menanyakan waktu n. Kapan modal kita habis? Dugaan awal kami adalah 30 tahun. fungsi untuk ini.

Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

```
>solve("fs(4000,-330,3,x)",30)
```

15.29

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

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

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

```
>equ &= fs(M,R,P,n)=Mn; $&equ
```

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

Biasanya rumus ini diberikan dalam bentuk

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

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

$$\frac{(i+1)^n - 1}{i} R + (i+1)^n M = Mn$$

Kita dapat memecahkan tingkat R secara simbolis.

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

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

Seperti yang Anda lihat dari rumus, fungsi ini mengembalikan kesalahan titik mengambang untuk $i=0$. Euler tetap merencanakannya.
Tentu saja, kami memiliki batas berikut.

```
>$&limit(R(4000,0,x,10),x,0)
```

$$\lim_{x \rightarrow 0} R(4000, 0, x, 10)$$

Jelas, tanpa bunga kita harus membayar kembali 10 tarif 500.
Persamaan juga dapat diselesaikan untuk n. Kelihatannya lebih bagus, jika kita menerapkan beberapa penyederhanaan untuk itu.

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

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

Penyelesaian Soal Aljabar

Nama: Muhammad Lutfi Ramadhan
NIM: 23030630021
Kelas: Matematika B 2023

R.2 Exercise Set

1. Sederhanakan bentuk eksponen berikut! (Soal nomor 49)

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

Penyelesaian:

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

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

2. Sederhanakan bentuk eksponen berikut! (Soal nomor 50)

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

Penyelesaian:

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

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

3. Hitunglah! (Soal nomor 90)

$$2^6 * 2^{-3} / 2^{10} / 2^{-8}$$

Penyelesaian:

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

4. Hitunglah! (Soal nomor 91)

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

Penyelesaian:

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

5.00

5. Hitunglah! (Soal nomor 92)

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

Penyelesaian:

```
> ((4*(8-6)^2+4)*(3-2*8)) / (2^2*(2^5+5))
```

-1.76

R.3 Exercise Set Perform the Indicated Operations

1. Lakukan operasi yang ditunjukkan! (Soal nomor 28)

$$(z+6)^2$$

Penyelesaian

```
>$&showev('expand((z+6)^2)')
```

$$\text{expand}\left((z+6)^2\right) = z^2 + 12z + 36$$

2. Lakukan operasi yang ditunjukkan! (Soal 13)

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

Penyelesaian:

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

$$\text{expand}\left(-21a^6\right) = -21a^6$$

3. Lakukan operasi yang ditunjukkan! (Soal nomor 21)

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

Penyelesaian:

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

$$\text{expand}\left((x+3)(x+6)\right) = x^2 + 9x + 18$$

4. Lakukan operasi yang ditunjukkan! (Soal nomor 40)

$$(2x-7)(2x+7)$$

Penyelesaian:

```
>$&showev('expand((2*x-7)*(2*x+7))')
```

$$\text{expand}\left((2x-7)(2x+7)\right) = 4x^2 - 49$$

5. Lakukan operasi yang ditunjukkan! (Soal nomor 29)

$$(y-5)^2$$

Penyelesaian:

```
>$&showev('expand((y-5)^2)')
```

$$\text{expand}\left((y-5)^2\right) = y^2 - 10y + 25$$

R.4 Exercise Set Faktor Trinomial

1. Faktorkan jumlah atau selisih kubus! (Soal nomor 69)

$$m^3 - 216$$

Penyelesaian:

```
>$&solve(m^3-216)
```

$$\left[m = 3^{\frac{3}{2}} i - 3, m = -3^{\frac{3}{2}} i - 3, m = 6 \right]$$

2. Faktorkan jumlah atau selisih kubus! (Soal nomor 70)

$$n^3 + 1$$

Penyelesaian:

```
>$&solve(n^3+1)
```

$$\left[n = \frac{1 - \sqrt{3}i}{2}, n = \frac{\sqrt{3}i + 1}{2}, n = -1 \right]$$

3. Faktorkan kuadrat dari sebuah binomial! (Soal nomor 58)

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

Penyelesaian:

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

$$[y = 3]$$

4. Faktorkan kuadrat dari sebuah binomial! (Soal nomor 59)

$$9z^2 - 12z + 4$$

Penyelesaian:

```
>$&solve(9*z^2-12*z+4)
```

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

5. Faktorkan sepenuhnya! (Soal nomor 110)

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

Penyelesaian:

```
>$&solve(x^3-x^2-x+1)
```

$$[x = -1, x = 1]$$

R.5 The Basics of Equation Solving

1. Selesaikan! (Soal nomor 37)

$$x^2 + 5x = 0$$

Penyelesaian:

```
>$&solve(x^2+5*x=0,x)
```

$$[x = -5, x = 0]$$

2. Selesaikan! (Soal nomor 48)

$$6x^2 - 7x = 10$$

Penyelesaian:

```
>$&solve(6*x^2-7*x=10,x)
```

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

3. Selesaikan! (Soal nomor 56)

$$t^2 = 25$$

Penyelesaian:

```
>$&solve(t^2=25,t)
```

$$[t = -5, t = 5]$$

4. Selesaikan! (Soal nomor 41)

$$x^2 + 100 = 20x$$

Penyelesaian:

```
>$&solve (x^2+100=20*x, x)
```

$$[x = 10]$$

5. Selesaikan! (Soal nomor 55)

$$z^2 = 144$$

Penyelesaian:

```
>$&solve (z^2=144)
```

$$[z = -12, z = 12]$$

Tugas Individu Pekan5-6_Muhammad Lutfi Ramadhan_23030630021

Nama : Muhammad Lutfi Ramadhan
Kelas : Matematika B 2023
NIM : 23030630021

Menggambar Grafik 2D dengan EMT

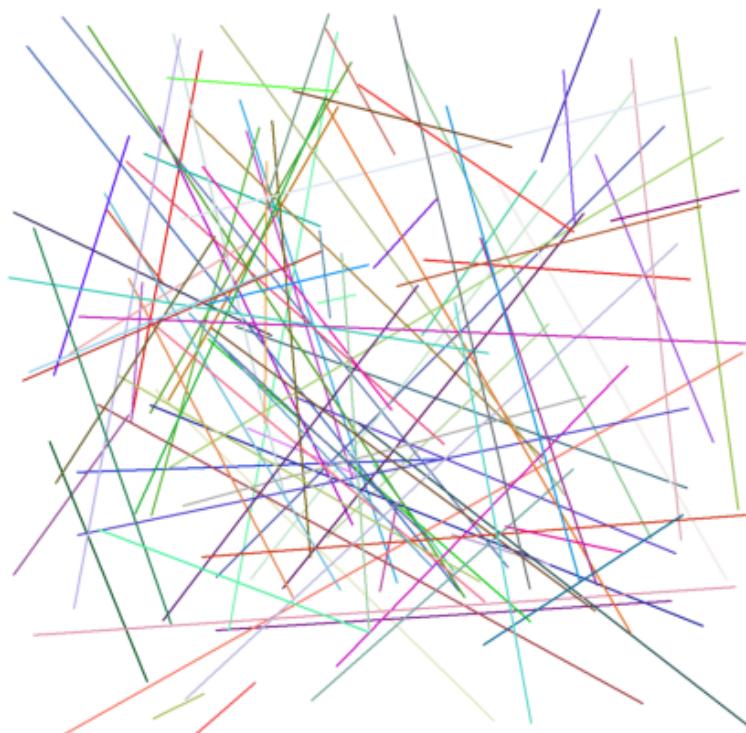
Notebook ini menjelaskan tentang cara menggambar berbagai kurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi plot2d() untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

Plot Dasar

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

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

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



```
>reset;
```

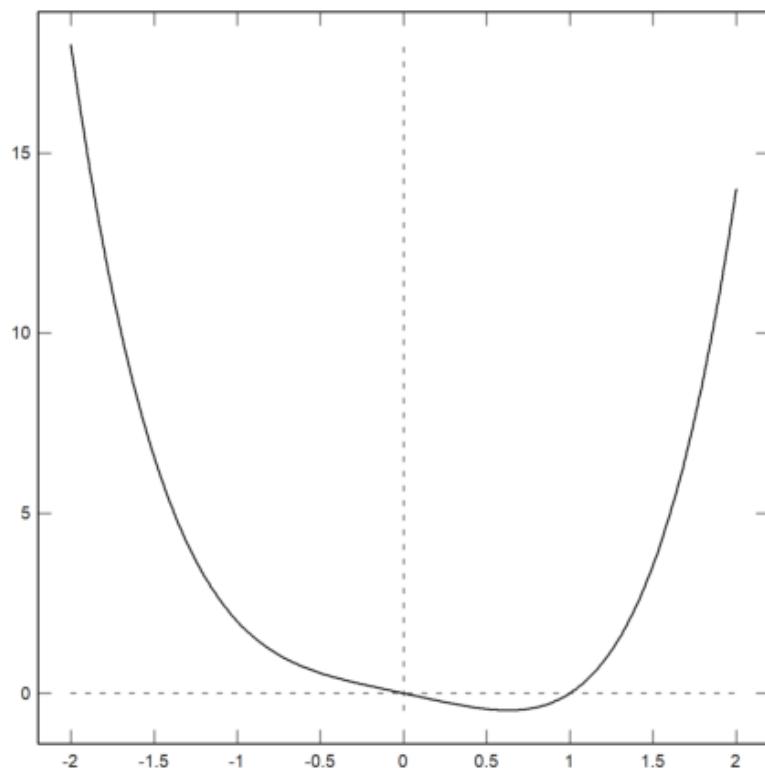
Ini penting untuk menahan grafik karena perintah plot() akan menghapus jendela plot.

Untuk menghapus semuanya kita menggunakan reset().

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

Sebagai contoh lain, kita menggambar sebuah plot sebagai inset (sisipan) di dalam plot lain. Ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perlu dicatat bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar jendela plot, sehingga kita perlu menambahkan margin sesuai kebutuhan. Kita menyimpan dan mengembalikan jendela penuh, serta menahan plot saat kita menggambar inset.

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



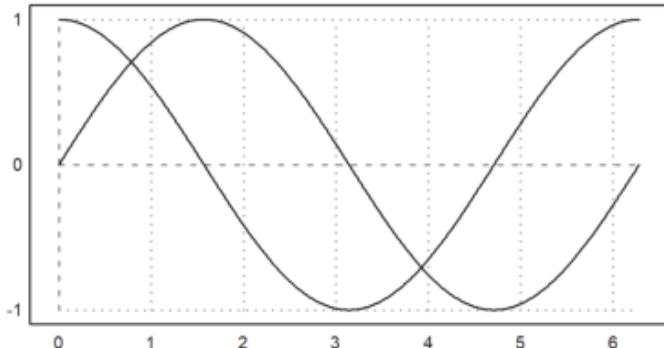
```
>hold off;
>>window(ow);
```

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

Aspek Plot

Plot default menggunakan jendela plot persegi. Kita dapat mengubah ini dengan fungsi aspect(). Jangan lupa untuk mengatur ulang aspek setelahnya. Kita juga dapat mengubah pengaturan ini dalam menu dengan "Set Aspect" ke rasio aspek tertentu atau ke ukuran grafik saat ini. Anda juga dapat mengubahnya untuk satu plot. Untuk ini, ukuran area plot saat ini diubah, dan jendela diatur agar label memiliki ruang yang cukup.

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



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

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

Plot 2D dalam Euler

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

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

Euler dapat memplot plot 2D dari:

- Ekspresi,
- Fungsi, variabel, atau kurva parameterisasi,
- Vektor nilai x-y,
- Sekumpulan titik di bidang,
- Kurva implisit dengan level atau area level,
- Fungsi kompleks,

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

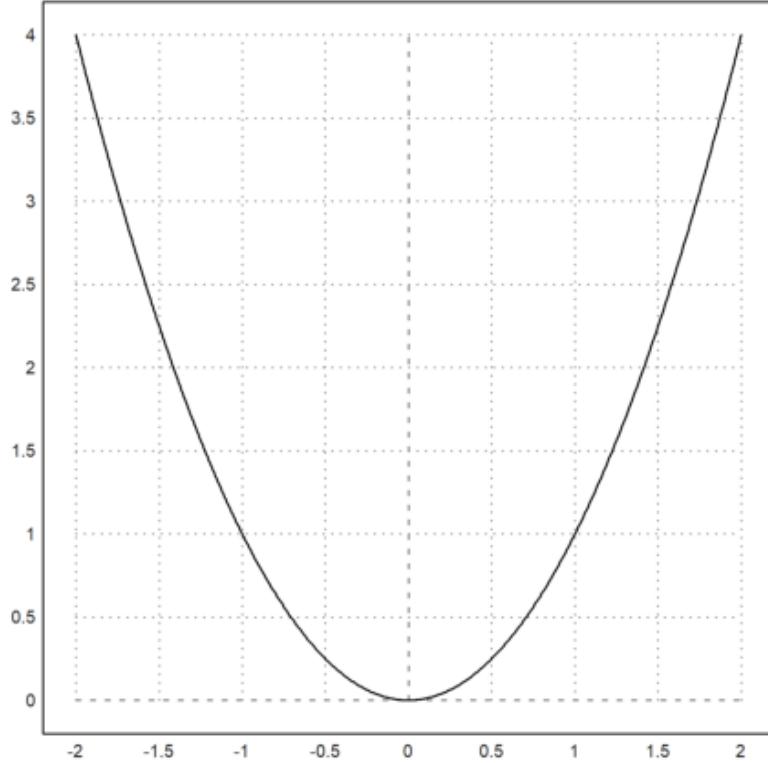
Plot Ekspresi atau Variabel

Sebuah ekspresi tunggal dalam "x" (misalnya "4*x^2") atau nama sebuah fungsi (misalnya "f") akan menghasilkan grafik fungsi tersebut.

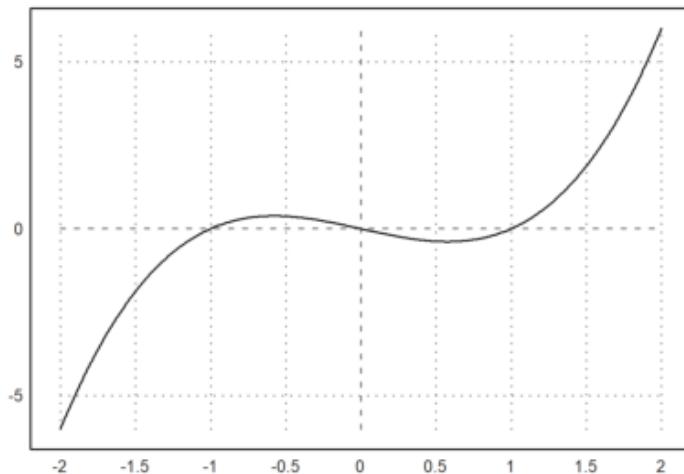
Berikut adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang sesuai untuk memuat plot fungsi tersebut.

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

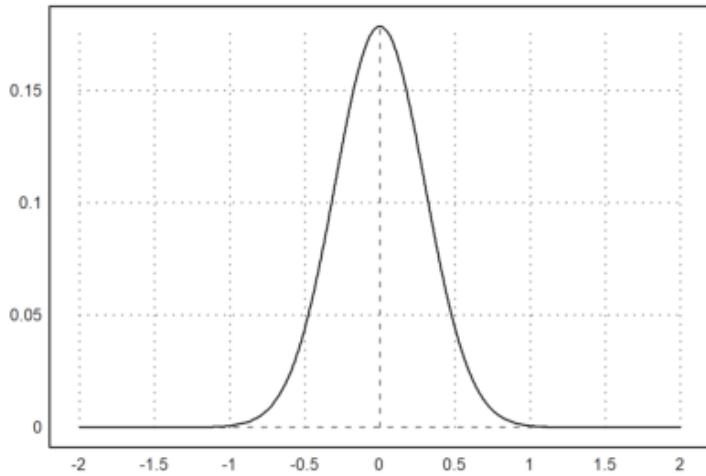
```
>plot2d("x^2":
```



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



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

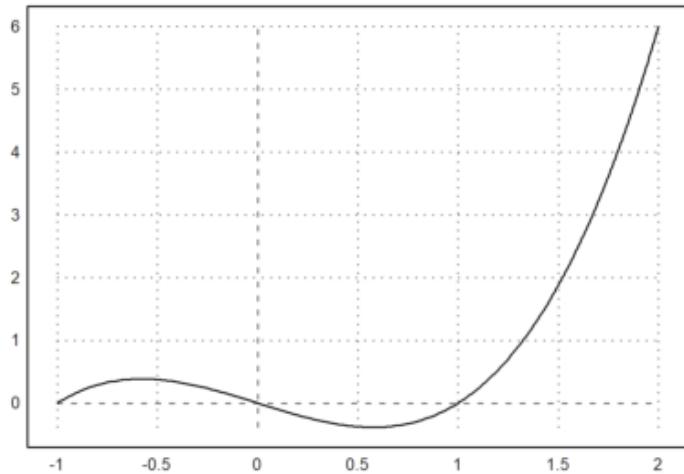


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

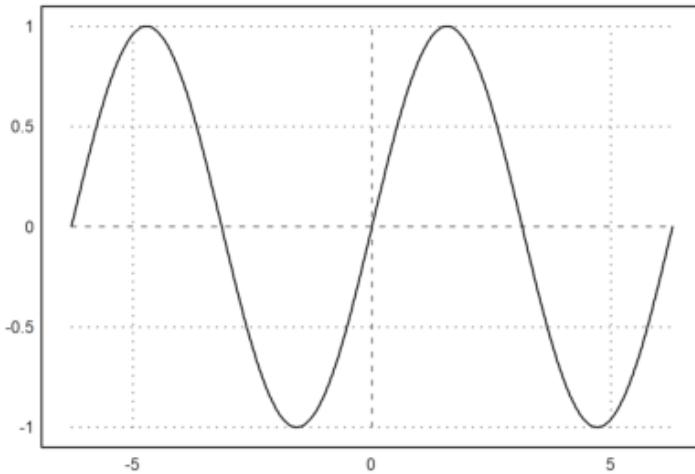
Rentang plot diatur dengan parameter yang ditetapkan berikut:

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

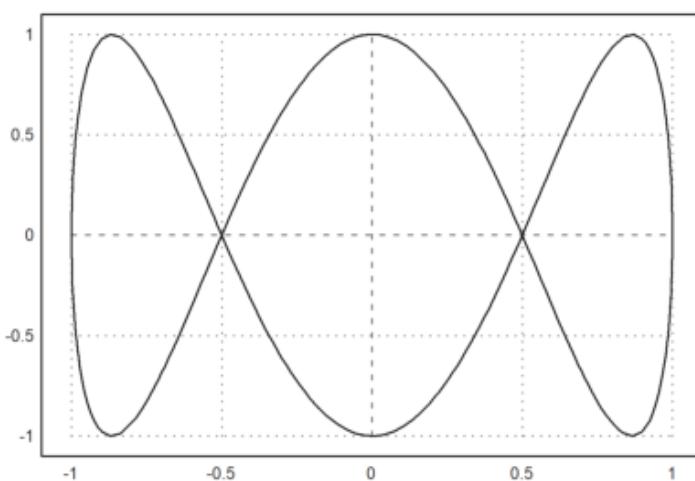
```
>plot2d("x^3-x",-1,2):
```



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



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



Alternatif untuk titik dua (:) adalah perintah insimg(lines), yang menyisipkan plot sesuai jumlah baris teks yang ditentukan.

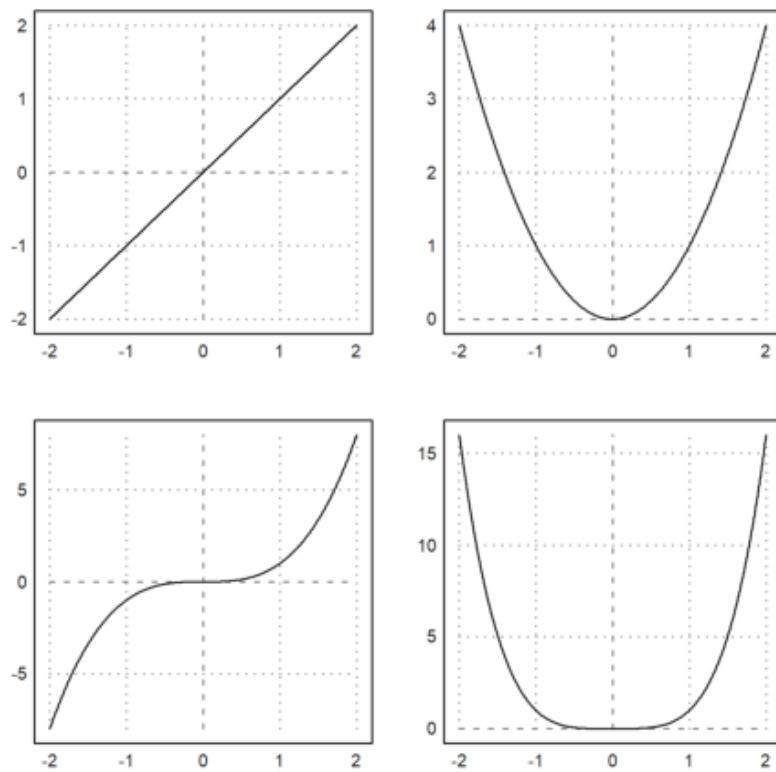
Dalam opsi lainnya, plot dapat diatur agar muncul:

- di jendela yang dapat diubah ukurannya
- di jendela notebook

Lebih banyak gaya dapat dicapai dengan perintah plot khusus.

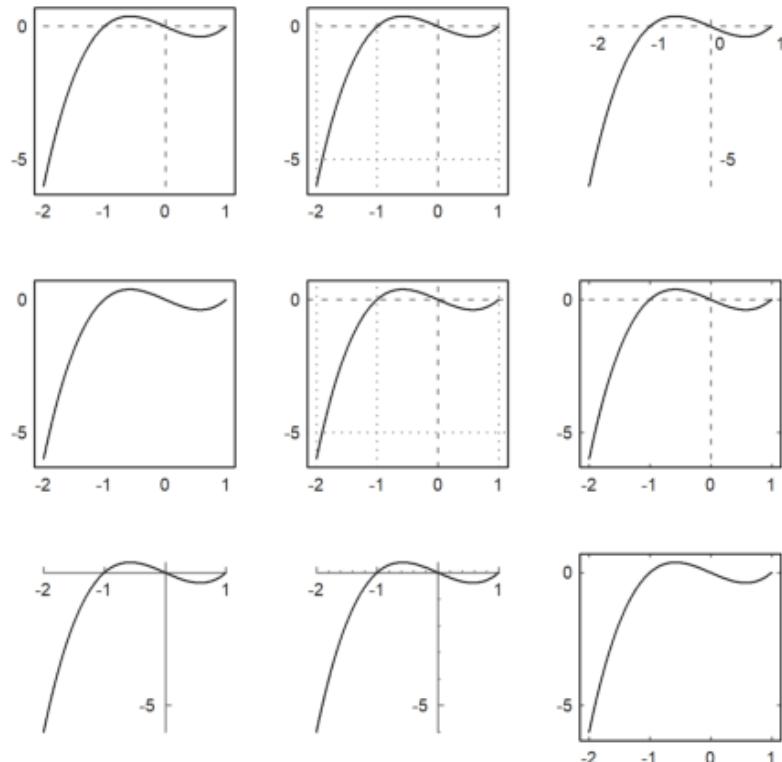
Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Dalam contoh berikut, kami memplotkan x^1 hingga x^4 ke dalam 4 bagian jendela. figure(0) mengatur ulang jendela ke default.

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



Dalam `plot2d()`, ada gaya alternatif yang tersedia dengan `grid=x`. Untuk gambaran, kami menunjukkan berbagai gaya grid dalam satu figur.

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

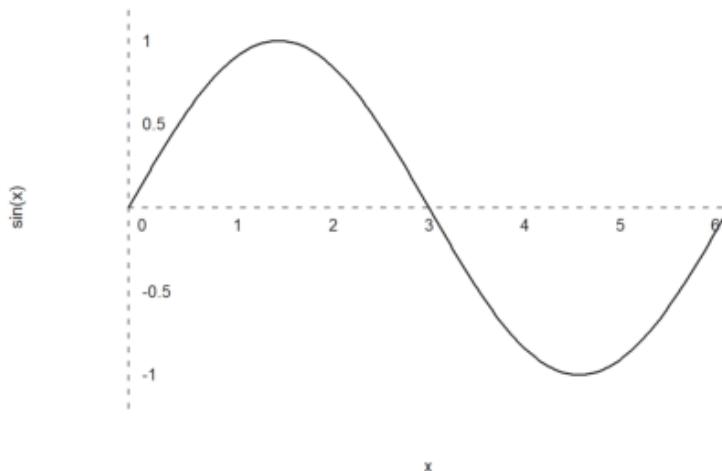


Jika argumen untuk `plot2d()` adalah ekspresi yang diikuti oleh empat angka, angka-angka tersebut adalah rentang x dan y untuk plot.

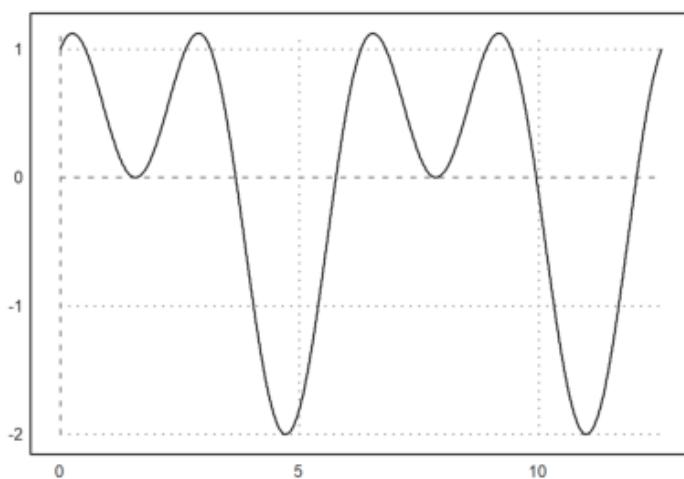
Sebagai alternatif, a, b, c, d dapat ditentukan sebagai parameter yang ditetapkan, misalnya `a=...` dll.

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

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



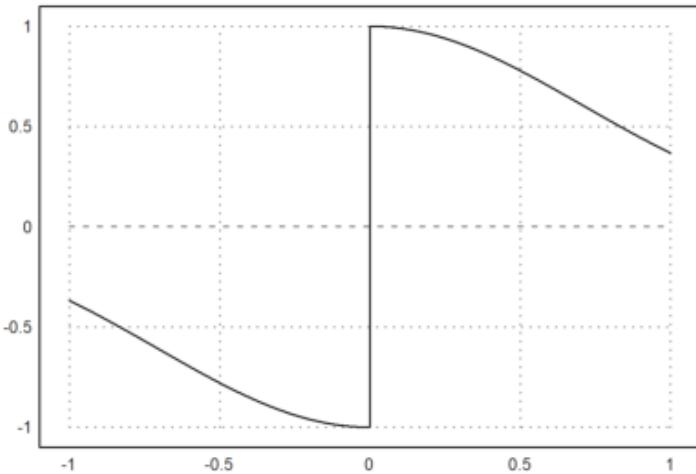
```
>plot2d("sin(x)+cos(2*x)",0,4pi):
```



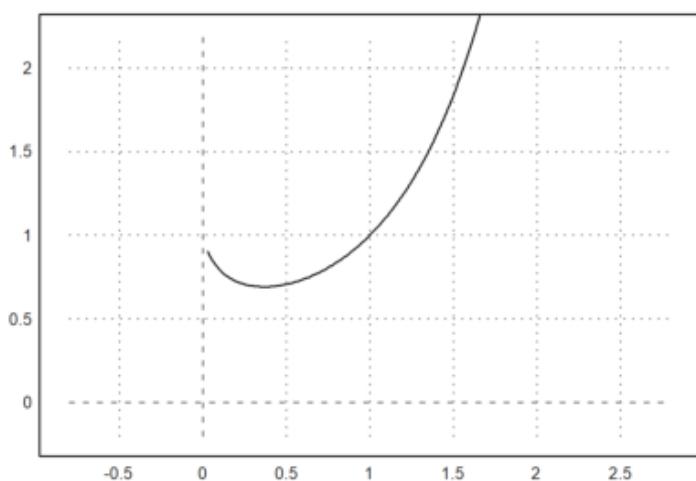
Gambar yang dihasilkan dari menyisipkan plot ke dalam jendela teks disimpan di direktori yang sama dengan notebook, secara default di subdirektori bernama "images". Gambar-gambar ini juga digunakan dalam ekspor HTML. Anda dapat menandai gambar apa pun dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengekspor grafik saat ini dengan fungsi di menu File.

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

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

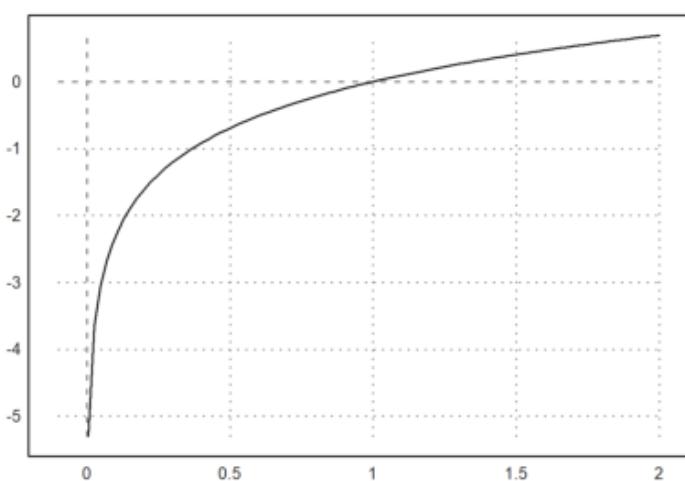


```
>plot2d("x^x", r=1.2, cx=1, cy=1):
```



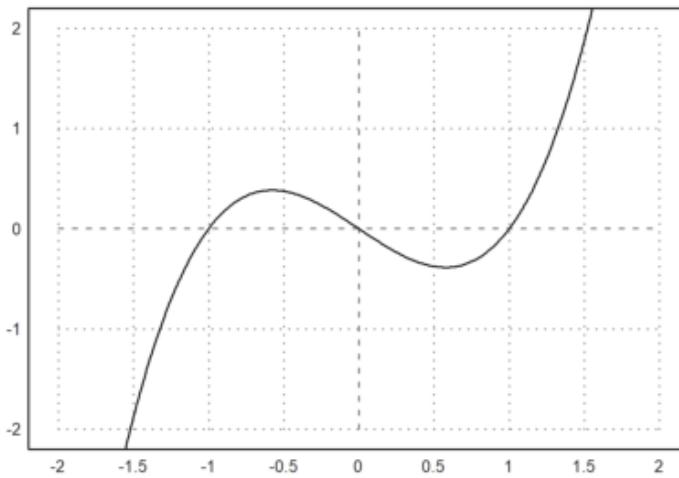
Perlu dicatat bahwa x^x tidak didefinisikan untuk $x \leq 0$. Fungsi plot2d menangkap kesalahan ini, dan mulai memplot begitu fungsi terdefinisi. Ini berlaku untuk semua fungsi yang mengembalikan NAN di luar rentang definisinya.

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

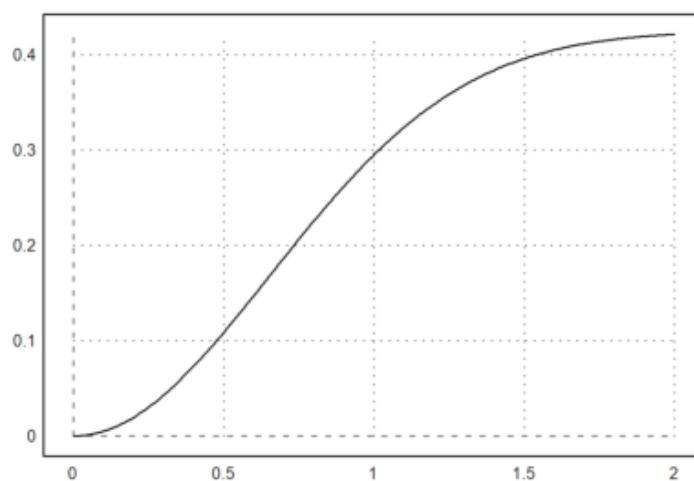


Parameter square=true (atau >square) memilih rentang y secara otomatis sehingga hasilnya adalah jendela plot persegi. Secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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



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

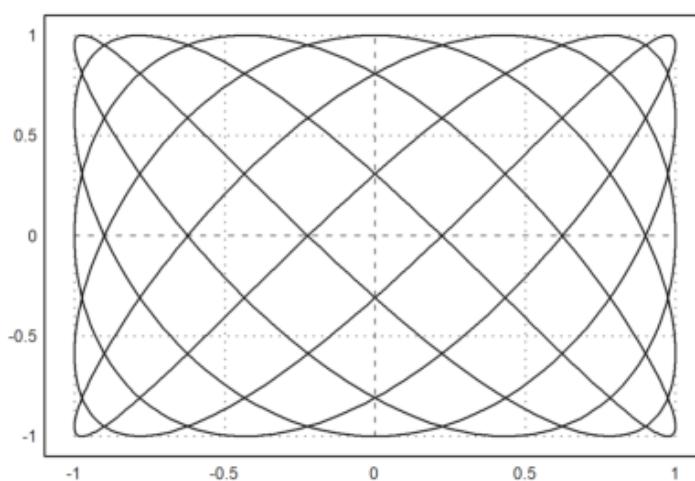


Jika Anda memerlukan lebih banyak ruang untuk label sumbu y, panggil shrinkwindow() dengan parameter yang lebih kecil, atau atur nilai positif untuk "smaller" di plot2d().

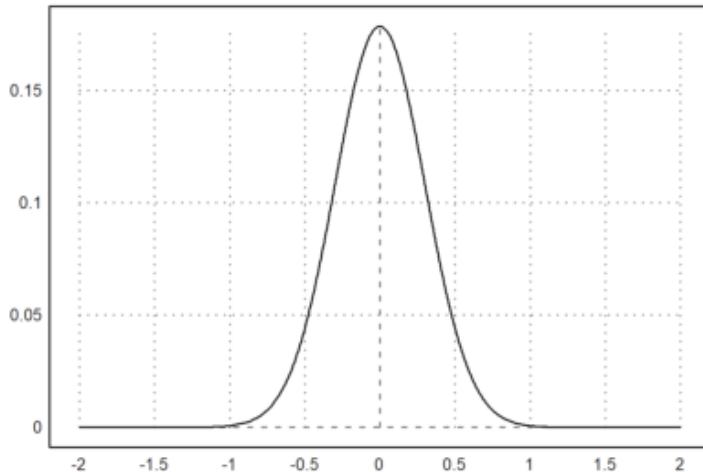
```
>plot2d("gamma(x)",1,10,yl="y-values",smaller=6,<vertical):
```

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

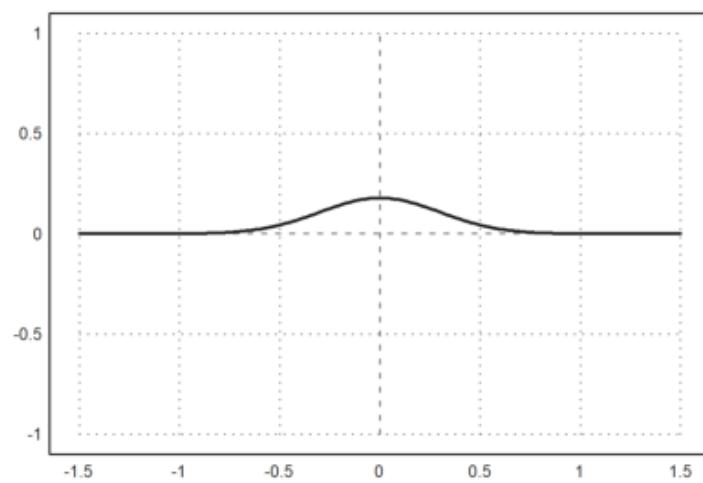
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



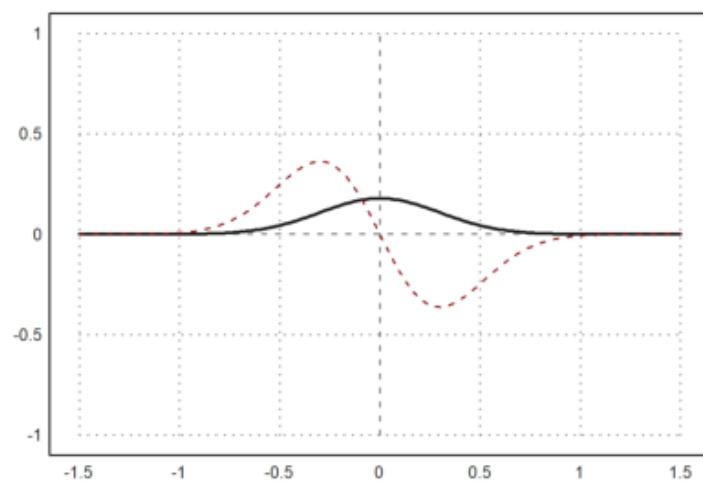
```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression  
>plot2d(expr,-2,2); // plot from -2 to 2
```



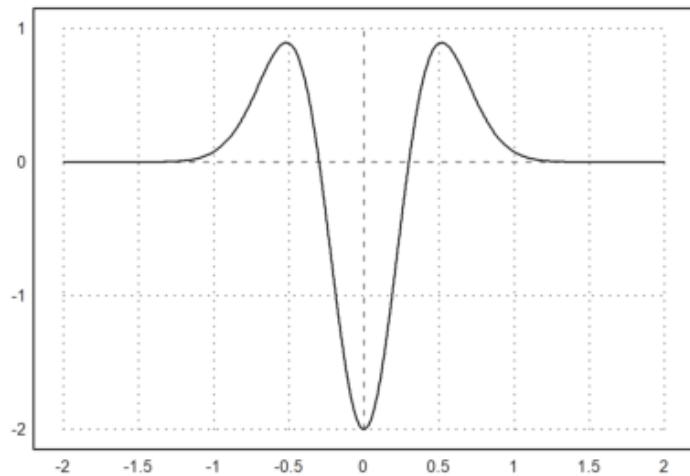
```
>plot2d(expr,r=1,thickness=2); // plot in a square around (0,0)
```



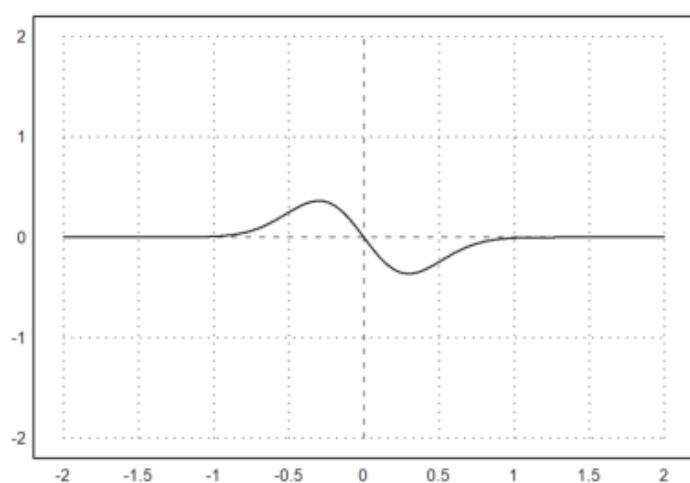
```
>plot2d(&diff(expr,x),>add,style="--",color=red); // add another plot
```



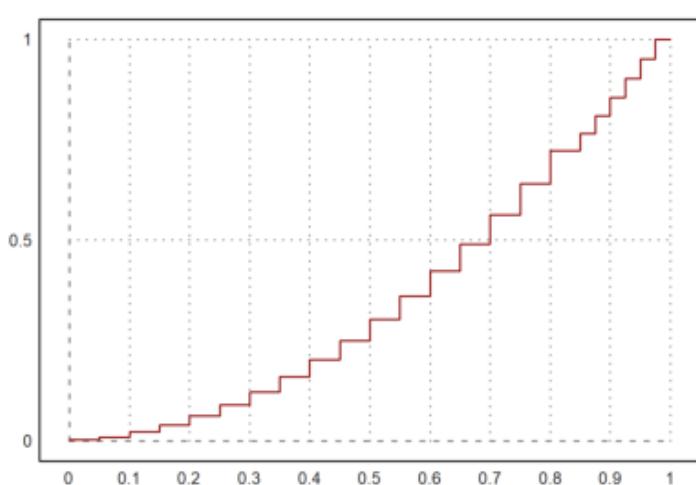
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1); // plot in rectangle
```



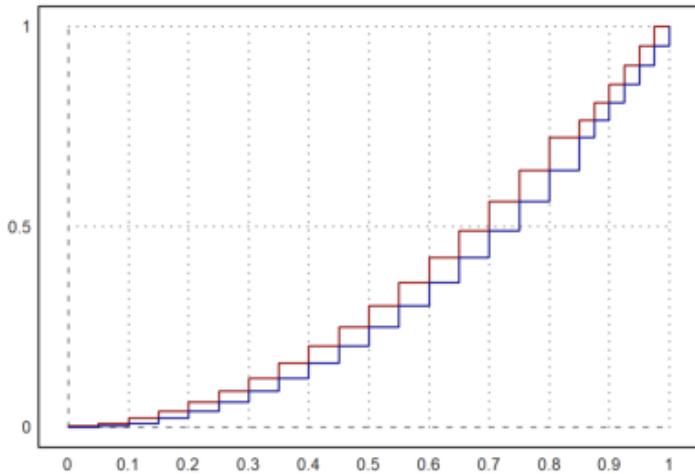
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```



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



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

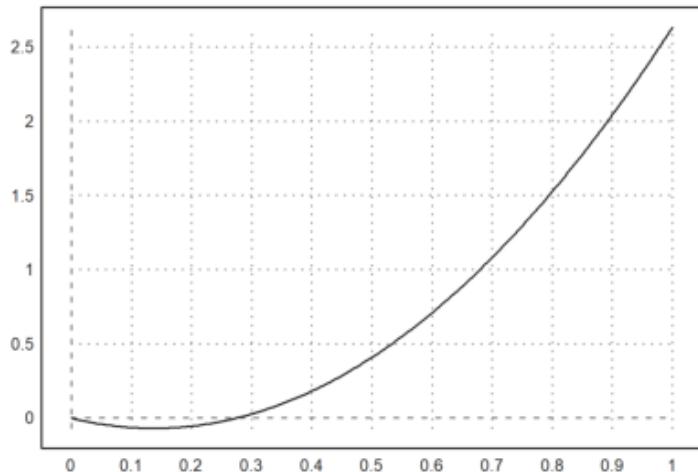


Fungsi dengan Satu Parameter

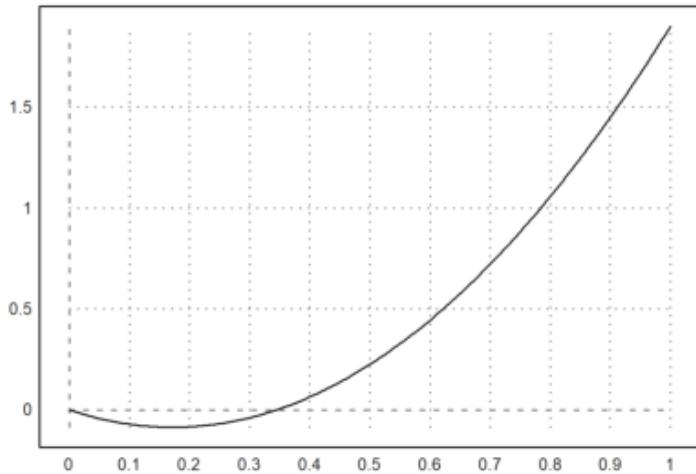
Fungsi plot paling penting untuk plot planar adalah `plot2d()`. Fungsi ini diimplementasikan dalam bahasa Euler di file "plot.e", yang dimuat saat program dimulai.

Berikut beberapa contoh menggunakan fungsi. Seperti biasa dalam EMT, fungsi yang bekerja untuk fungsi atau ekspresi lainnya, Anda dapat memberikan parameter tambahan (selain x) yang bukan variabel global kepada fungsi dengan parameter titik koma atau dengan koleksi panggilan.

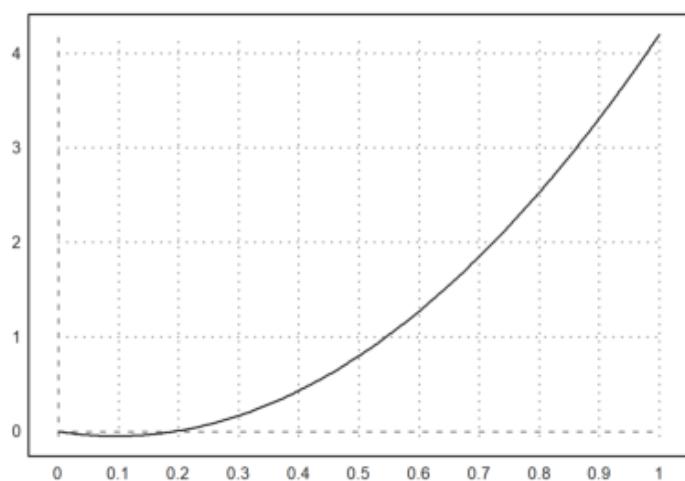
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a); // plot with a=0.3
```



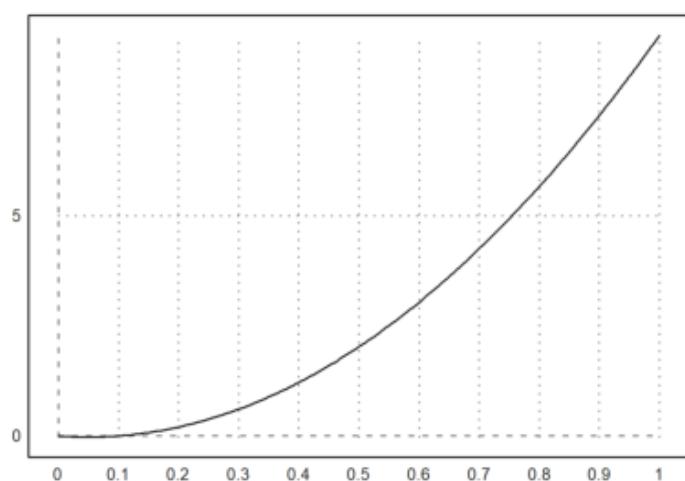
```
>plot2d("f",0,1;0.4); // plot with a=0.4
```



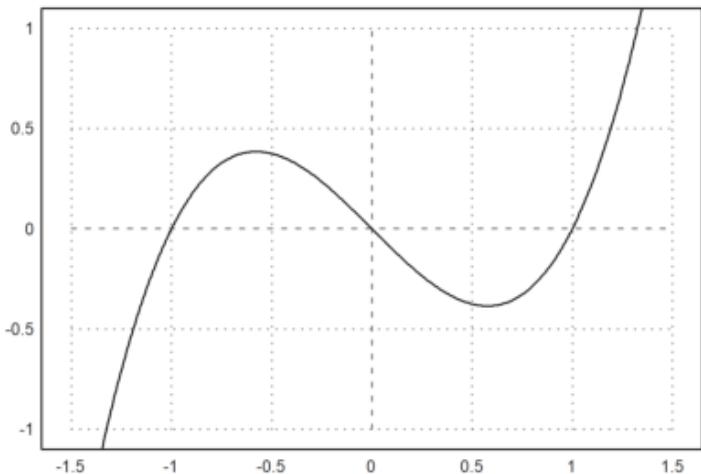
```
>plot2d({{"f",0.2}},0,1); // plot with a=0.2
```



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



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



Berikut ringkasan fungsi yang diterima:

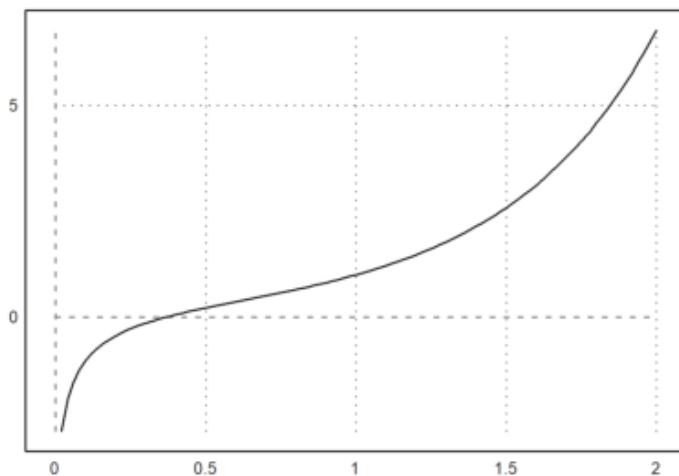
- Ekspresi atau ekspresi simbolik dalam x
- Fungsi atau fungsi simbolik berdasarkan nama seperti "f"
- Fungsi simbolik hanya dengan nama f

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

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

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

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

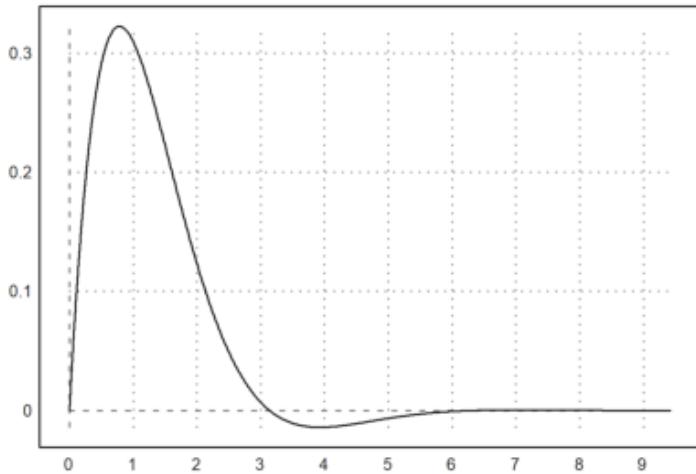


Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel saja cukup untuk memplotkannya.

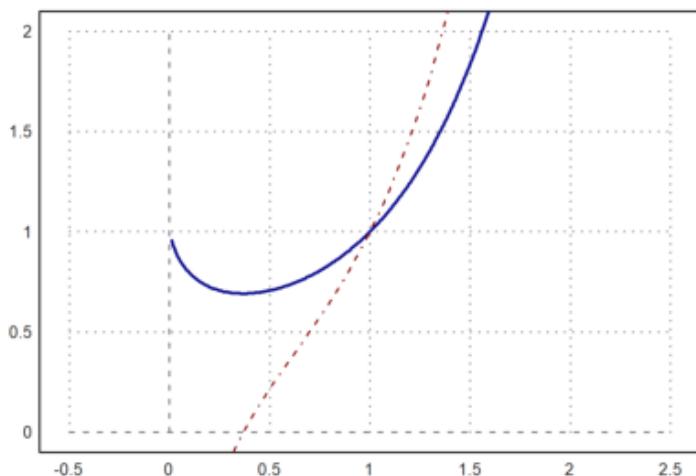
```
>expr &= sin(x)*exp(-x)
```

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

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



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



Untuk gaya garis, ada berbagai opsi:

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

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

- 0..15: indeks warna default
- constanta warna: putih, hitam, merah, hijau, biru, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, kuning
- rgb(red,green,blue): parameter adalah real dalam [0,1].

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

Berikut tampilan warna yang telah ditetapkan dalam EMT:

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

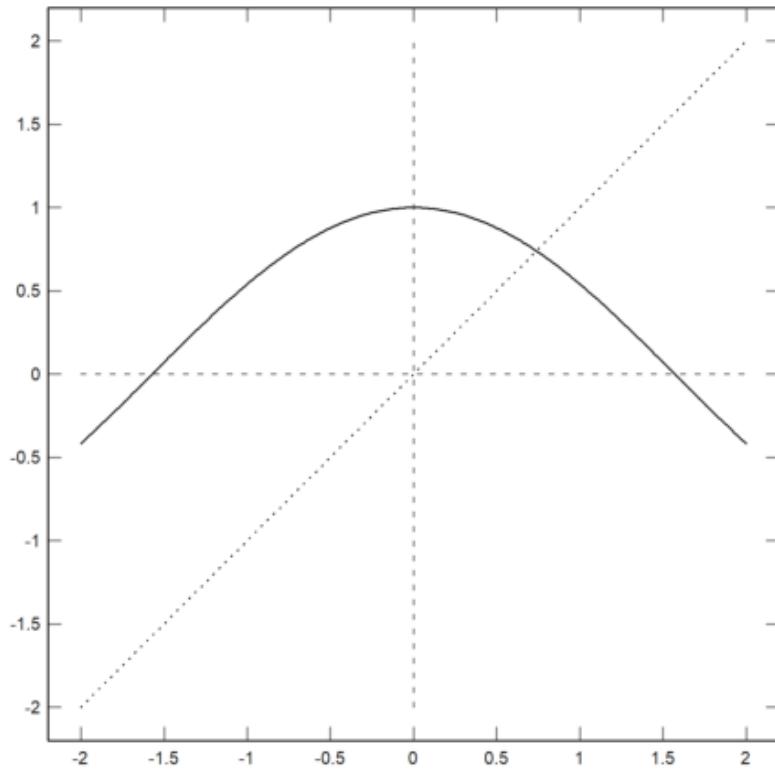
Namun, Anda dapat menggunakan warna apapun.

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

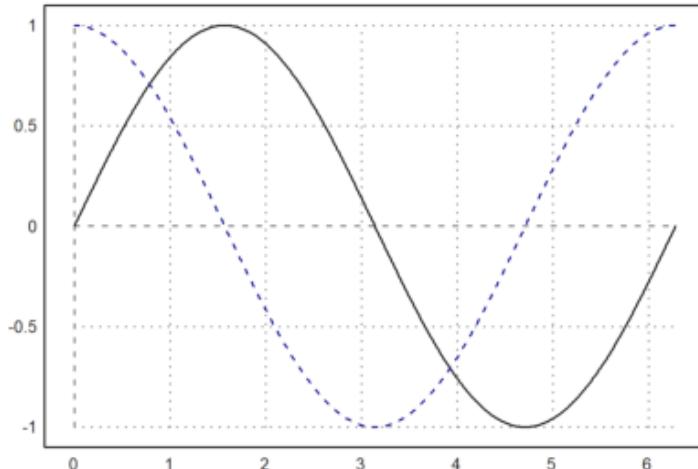
Menggambar Beberapa Kurva pada bidang koordinat yang sama

Memplot lebih dari satu fungsi ke dalam satu jendela dapat dilakukan dengan beberapa cara. Salah satu metodenya adalah menggunakan `>add` untuk beberapa panggilan ke `plot2d` di semua, kecuali panggilan pertama. Kami telah menggunakan fitur ini dalam contoh sebelumnya.

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

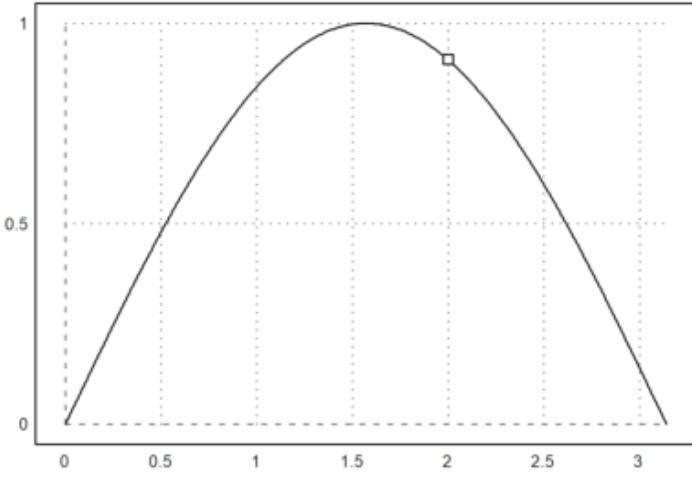


```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



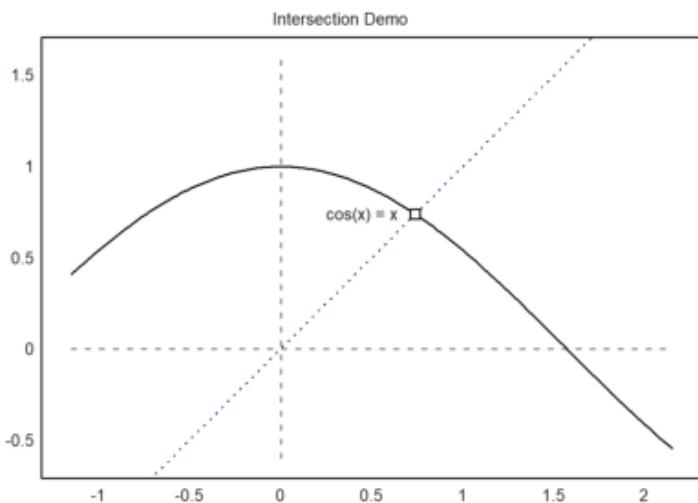
Salah satu kegunaan `>add` adalah untuk menambahkan titik pada kurva.

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



Kami menambahkan titik perpotongan dengan label (di posisi "cl" untuk tengah kiri), dan menyisipkan hasilnya ke dalam notebook. Kami juga menambahkan judul pada plot.

```
>plot2d(["cos(x)","x"],r=1.1,cx=0.5,cy=0.5, ...
    color=[black,blue],style=["-", "."], ...
    grid=1);
>x0=solve("cos(x)-x",1); ...
    plot2d(x0,x0,>points,>add,title="Intersection Demo"); ...
    label("cos(x) = x",x0,x0,pos="cl",offset=20):
```



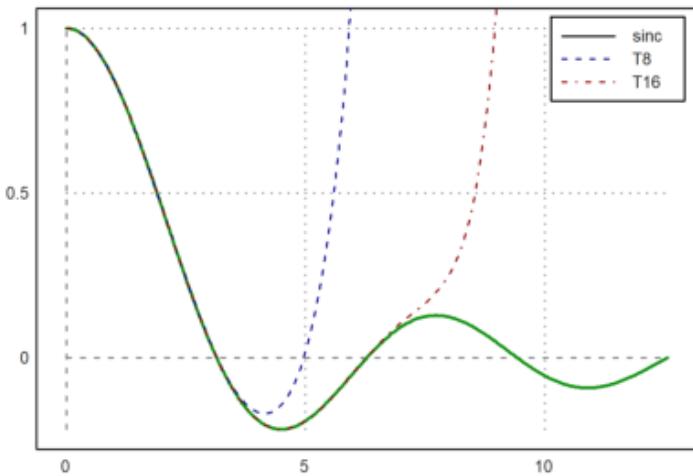
Dalam demo berikut, kami memplot fungsi $\text{sinc}(x)=\sin(x)/x$ dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolik. Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke `plot2d()`. Panggilan kedua dan ketiga memiliki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

Kami menambahkan kotak label untuk menjelaskan fungsi-fungsi tersebut.

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

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

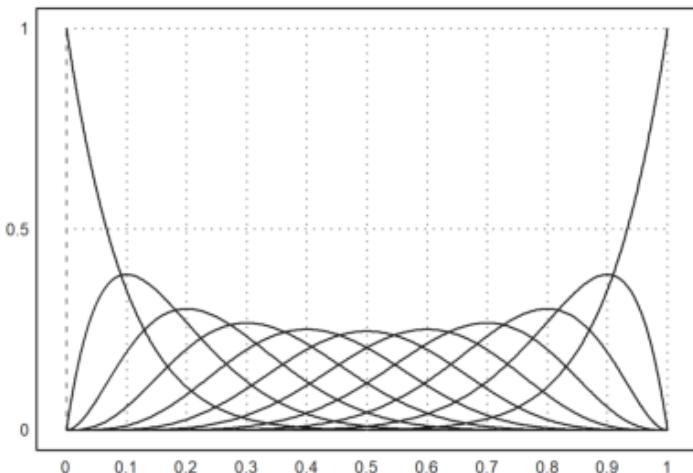
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
    plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
    plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
    labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
    colors=[black,blue,red]):
```



Dalam contoh berikut, kami menghasilkan polinomial Bernstein.

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

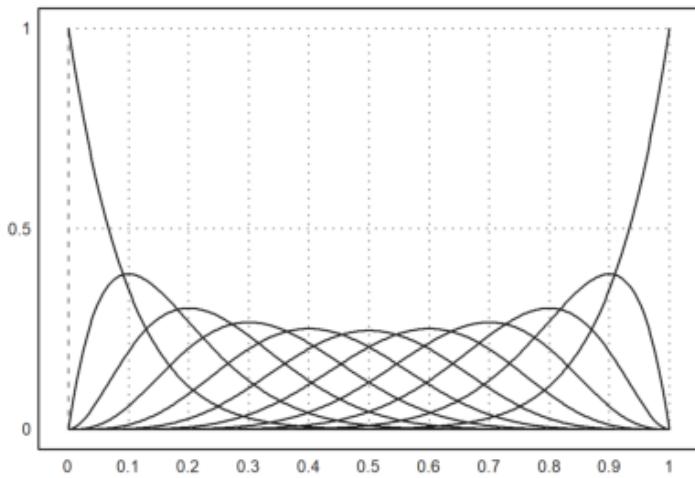
```
>plot2d("(1-x)^10",0,1); // plot first function
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```



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

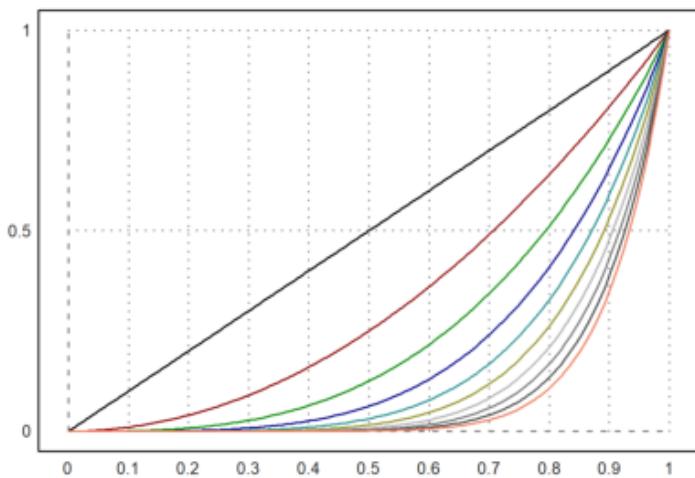
Kami menghasilkan matriks nilai dengan satu Polinomial Bernstein di setiap baris. Untuk ini, kami cukup menggunakan vektor kolom dari i. Lihat pengantar tentang bahasa matriks untuk mempelajari detail lebih lanjut.

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



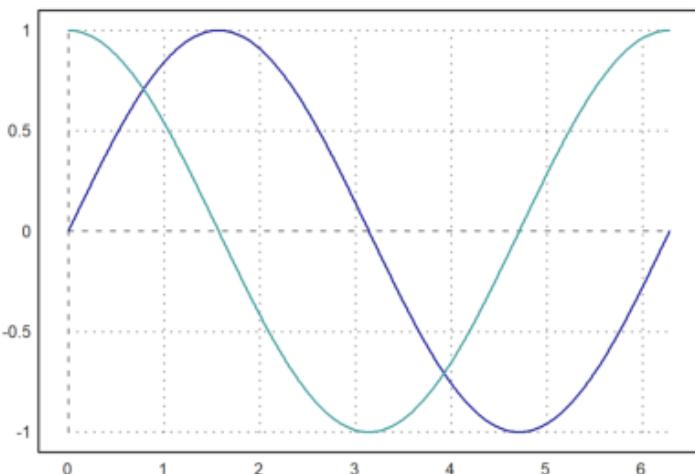
Perlu diperhatikan bahwa parameter `color` dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

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

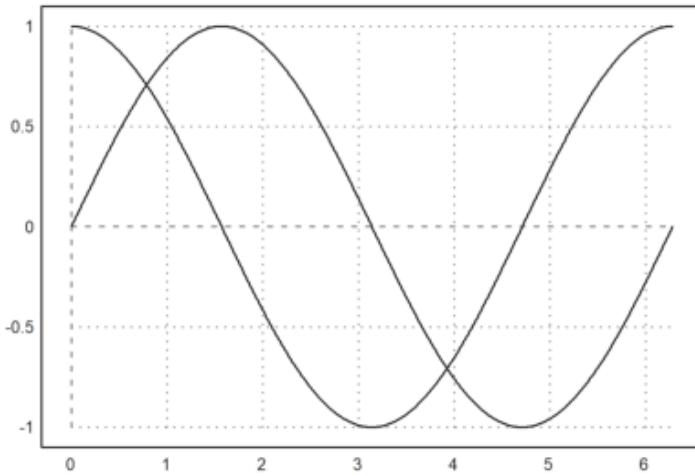


Metode lain adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan array warna, array gaya, dan array ketebalan dengan panjang yang sama.

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



```
>plot2d(["sin(x)","cos(x)"],0,2pi): // plot vector of expressions
```



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

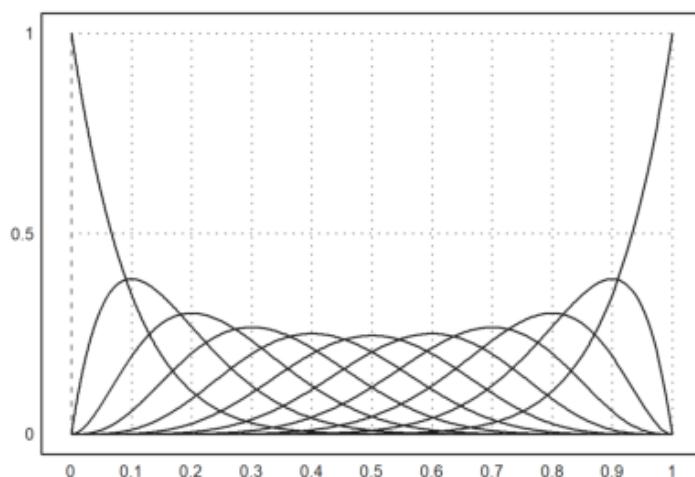
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

$$[(1-x)^{10}, 10(1-x)^9x, 45(1-x)^8x^2, 120(1-x)^7x^3, \\ 210(1-x)^6x^4, 252(1-x)^5x^5, 210(1-x)^4x^6, 120(1-x)^3x^7, \\ 45(1-x)^2x^8, 10(1-x)x^9, x^{10}]$$

```
>mxm2str(v) // get a vector of strings from the symbolic vector
```

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

```
>plot2d(mxm2str(v),0,1); // plot functions
```



Alternatif lain adalah menggunakan bahasa matriks dari 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 ditambahkan array warna, ini akan digunakan untuk setiap baris plot.

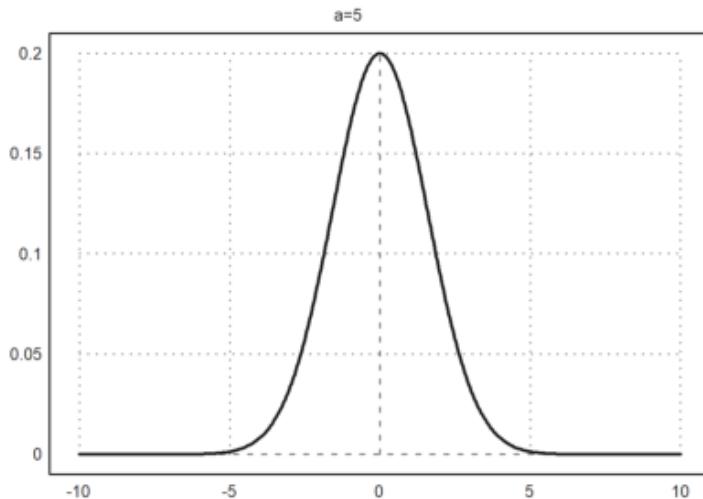
```
>n=(1:10)'; plot2d("x^n",0,1,color=1:10);
```

Ekspresi dan fungsi satu baris dapat melihat variabel global.

Jika Anda tidak bisa menggunakan variabel global, Anda perlu menggunakan fungsi dengan parameter tambahan, dan meneruskan parameter ini sebagai parameter titik koma. Pastikan semua parameter yang ditetapkan berada di akhir perintah plot2d.

Dalam contoh ini, kami meneruskan $a=5$ ke fungsi f , yang kami plot dari -10 hingga 10.

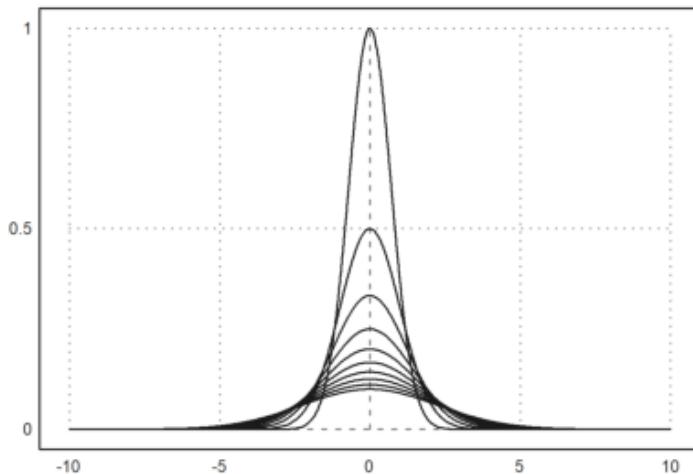
```
>function f(x,a) := 1/a*exp(-x^2/a); ...
plot2d("f",-10,10;5,thickness=2,title="a=5");
```



Sebagai alternatif, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut koleksi panggilan, dan ini adalah cara yang disukai untuk meneruskan argumen ke fungsi yang diteruskan sebagai argumen ke fungsi lain.

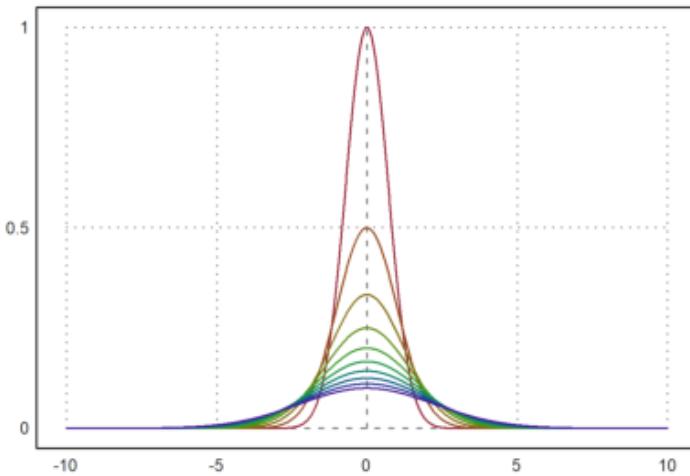
Dalam contoh berikut, kami menggunakan loop untuk memplot beberapa fungsi.

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



Hasil yang sama dapat dicapai dengan menggunakan bahasa matriks EMT. Setiap baris matriks $f(x,a)$ adalah satu fungsi. Selain itu, kita dapat mengatur warna untuk setiap baris matriks. Klik dua kali pada fungsi getspectral() untuk penjelasan.

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



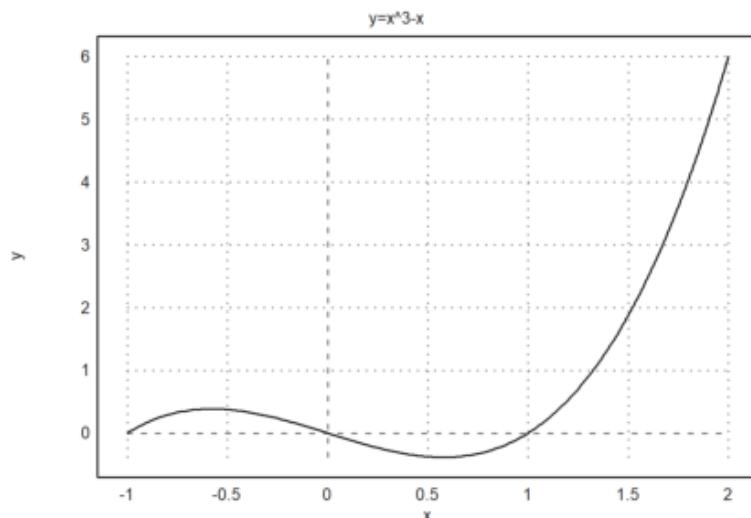
Label Teks

Dekorasi sederhana dapat berupa:

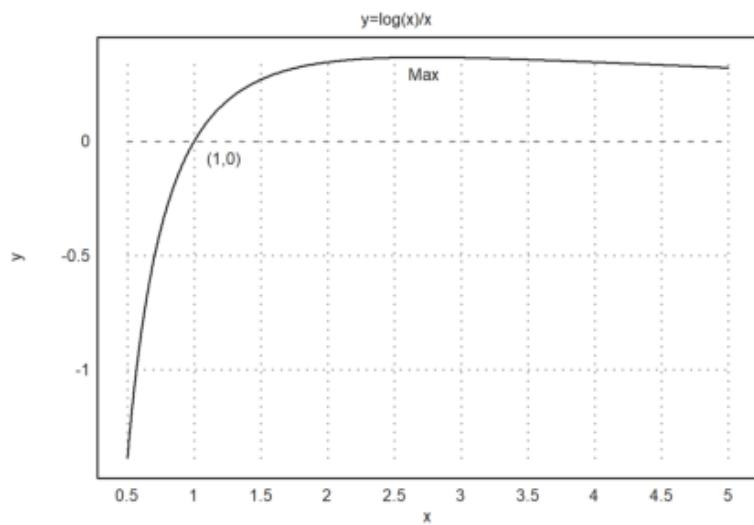
- judul dengan title="..."
- label x dan y dengan xl="...", yl="..."
- label teks lain dengan label("...",x,y)

Perintah label akan memplot ke plot saat ini di koordinat plot (x,y). Ia dapat mengambil argumen posisi.

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

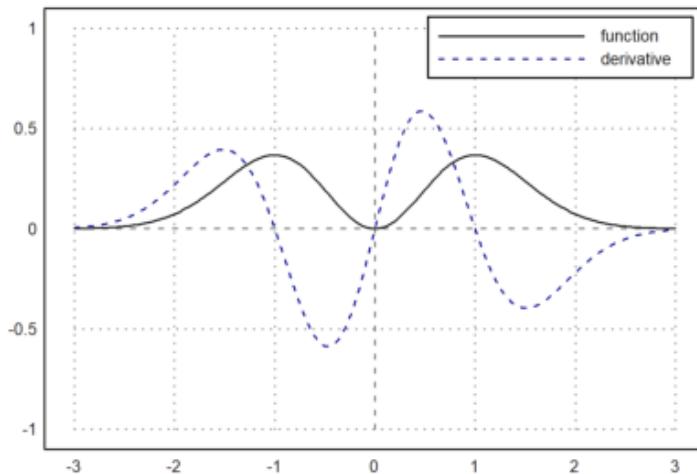


```
>expr := "log(x)/x"; ...
plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



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

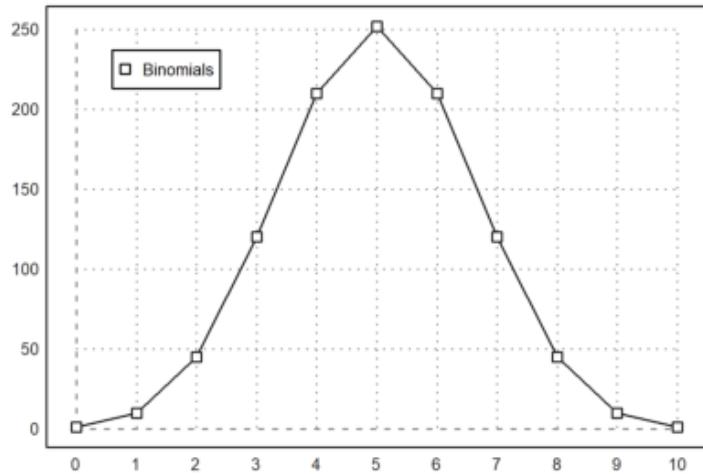
```
>function f(x) &= x^2*exp(-x^2); ...
plot2d(&f(x),a=-3,b=3,c=-1,d=1); ...
plot2d(&diff(f(x),x),>add,color=blue,style="--"); ...
labelbox(["function","derivative"],styles=[["-", "--"], ...
colors=[black,blue],w=0.4):
```



Kotak ini diposisikan di sudut kanan atas secara default, tetapi `>left` akan memposisikannya di sudut kiri atas. Anda dapat memindahkannya ke tempat mana saja yang Anda inginkan. Posisi jangkar adalah sudut kanan atas kotak, dan angka-angka adalah pecahan dari ukuran jendela grafik. Lebarnya otomatis.

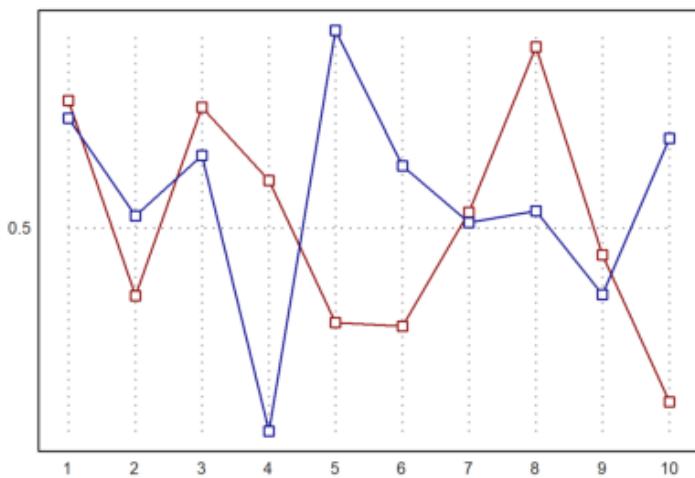
Untuk plot titik, kotak label juga berfungsi. Tambahkan parameter `>points`, atau vektor flag, satu untuk setiap label. Dalam contoh berikut, hanya ada satu fungsi. Jadi kita bisa menggunakan string alih-alih vektor string. Kami menetapkan warna teks ke hitam untuk contoh ini.

```
>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 dalam statplot(). Seperti dalam plot2d(), warna dapat diatur untuk setiap baris plot. Ada lebih banyak plot khusus untuk tujuan statistik (lihat tutorial tentang statistik).

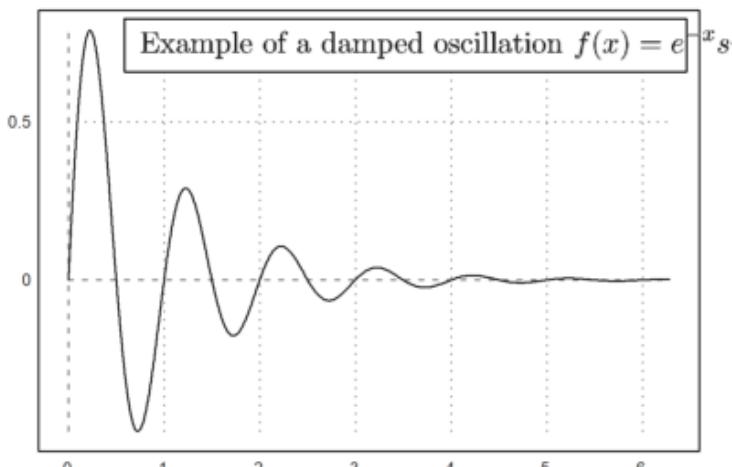
```
>statplot(1:10,random(2,10),color=[red,blue]):
```



Fitur serupa adalah fungsi textbox().

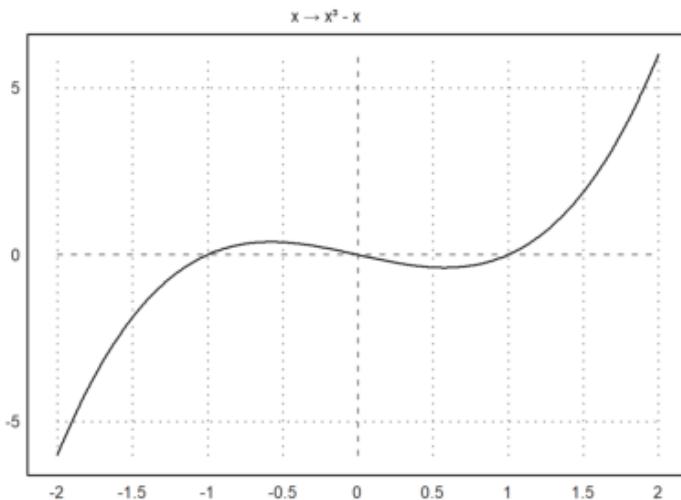
Lebarnya secara default adalah lebar maksimal dari garis teks, tetapi bisa diatur oleh pengguna juga.

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



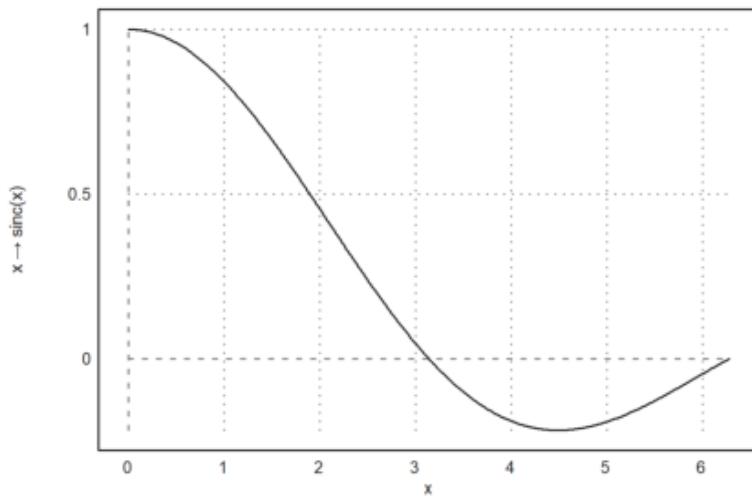
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk lebih lanjut tentang string Unicode).

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



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

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



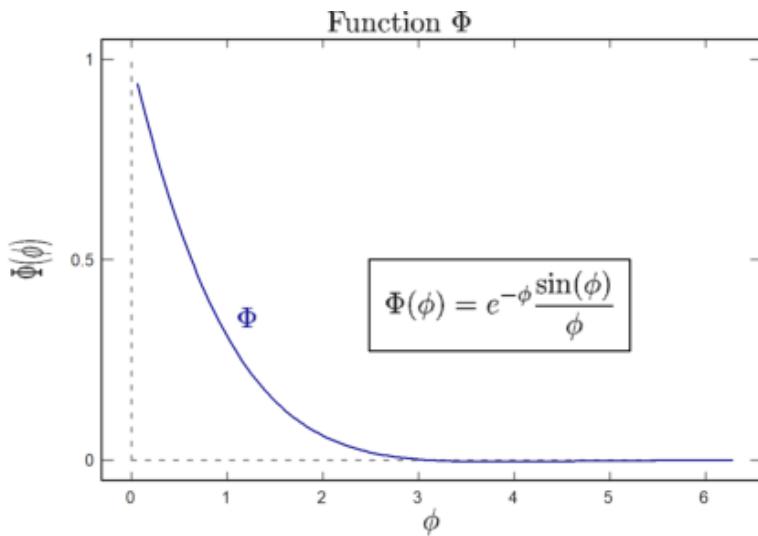
LaTeX

Anda juga dapat memplotkan formula LaTeX jika Anda sudah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Path ke biner "latex" dan "dvipng" harus ada dalam path sistem, atau Anda harus menyiapkan LaTeX di menu opsi.

Perlu dicatat, parsing LaTeX lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, sebaiknya panggil `latex()` sebelum loop sekali dan gunakan hasilnya (sebuah gambar dalam matriks RGB).

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

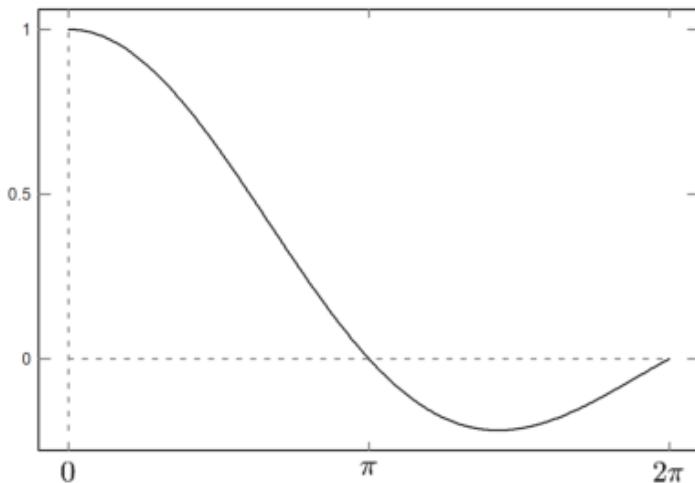
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
  title=latex("\text{Function } \Phi\$"), ...
  xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
  textbox( ...
    latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
  label(latex("\Phi"),color=blue),1,0.4):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



Sering kali, kita menginginkan jarak non-konformal dan label teks pada sumbu x. Kita dapat menggunakan xaxis() dan yaxis() seperti yang akan kami tunjukkan nanti.

Cara termudah adalah membuat plot kosong dengan bingkai menggunakan grid=4, lalu tambahkan grid dengan ygrid() dan xgrid(). Dalam contoh berikut, kami menggunakan tiga string LaTeX untuk label pada sumbu x dengan xtick().

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



Tentu saja, fungsi juga bisa digunakan.

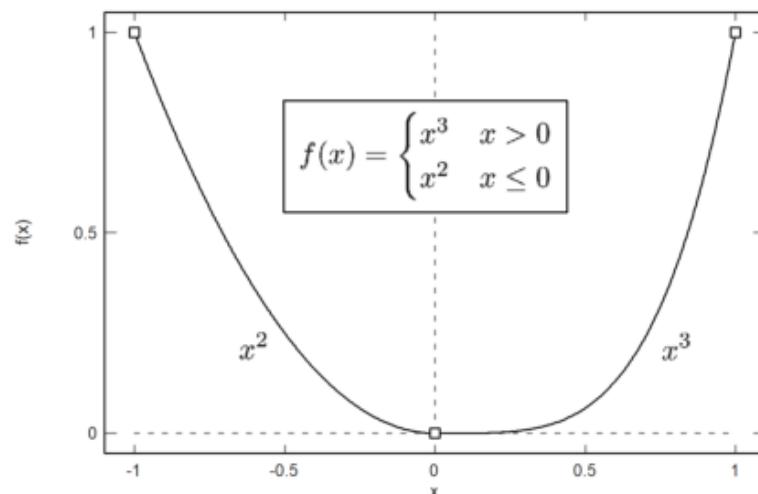
```
>function map f(x) ...
if x>0 then return x^4
else return x^2
endif
endfunction
```

Parameter "map" membantu menggunakan fungsi untuk vektor. Untuk plot, ini sebenarnya tidak perlu. Namun, untuk menunjukkan bahwa vektorisasi berguna, kita tambahkan beberapa titik kunci pada plot di $x=-1$, $x=0$, dan $x=1$.

Dalam plot berikut, kita juga memasukkan beberapa kode LaTeX. Kita menggunakanannya untuk dua label dan sebuah kotak teks. Tentu saja, Anda hanya dapat menggunakan LaTeX jika sudah menginstalnya 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

Saat memplot sebuah fungsi atau ekspresi, parameter `>user` memungkinkan pengguna memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat:

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

Tombol spasi akan mengatur ulang plot ke jendela plot asli. Saat memplot data, flag `>user` akan menunggu penekanan tombol.

```
>plot2d({{"x^3-a*x",a=1}},>user,title="Press any key!":
```

```
>plot2d("exp(x)*sin(x)",user=true, ...  
title="+/- or cursor keys (return to exit)":
```

Berikut ini adalah contoh cara interaksi pengguna yang lebih maju (lihat tutorial tentang pemrograman untuk detail).

Fungsi built-in `mousedrag()` menunggu peristiwa mouse atau keyboard. Fungsi `dragpoints()` memanfaatkan ini dan memungkinkan pengguna untuk menyeret titik mana pun di dalam plot. Pertama, kita membutuhkan fungsi `plotf`.

Sebagai contoh, kita menginterpolasi 5 titik dengan polinomial. Fungsi ini harus memplot ke 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
```

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

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

```
>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-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-nama untuk parameter, nilai awal, dan matriks nx2 dari rentang, opsional dengan garis heading. Ada slider interaktif, yang bisa mengatur nilai-nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

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

Dimungkinkan untuk membatasi nilai-nilai yang diseret menjadi bilangan bulat. Sebagai contoh, kita menulis fungsi plot yang memplot polinomial Taylor dari derajat n untuk fungsi cosinus.

```
>function plotf(n) ...
    plot2d("cos(x)",0,2pi,>square,grid=6);
    plot2d(&"taylor(cos(x),x,0,@n)",color=blue,>add);
    textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",>left);
endfunction
```

Sekarang kita memungkinkan derajat n bervariasi dari 0 hingga 20 dalam 20 langkah. Hasil dari dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk memasukkan plot ke dalam notebook.

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

Berikut ini adalah demonstrasi sederhana dari fungsi ini. Pengguna dapat menggambar di atas jendela plot, meninggalkan jejak titik-titik.

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

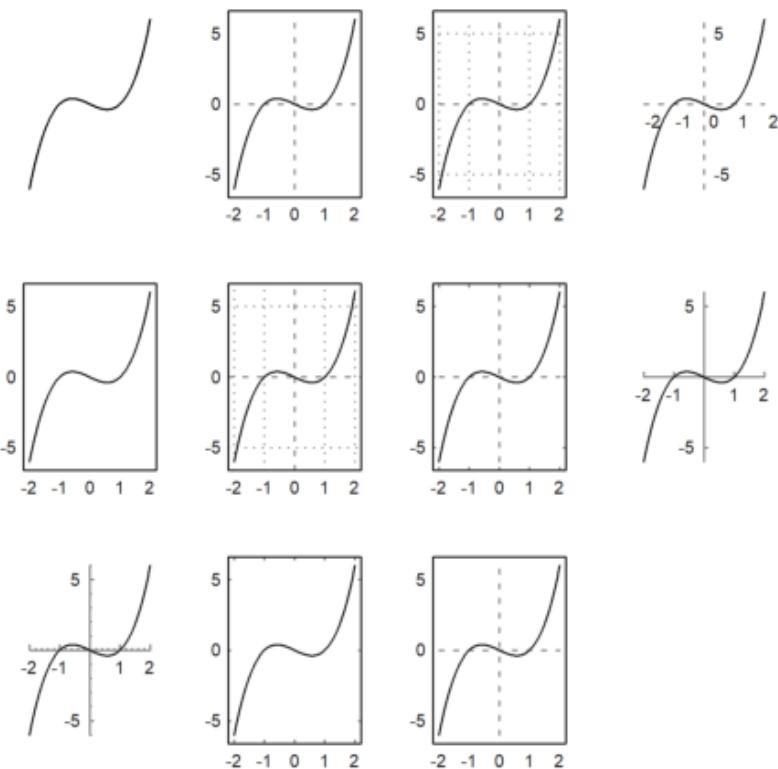
```
>dragtest // lihat hasilnya dan cobalah lakukan!
```

Gaya Plot 2D

Secara default, EMT menghitung ticks sumbu otomatis dan menambahkan label ke setiap tick. Ini bisa diubah dengan parameter grid. Gaya default dari sumbu dan label bisa dimodifikasi. Selain itu, label dan judul bisa ditambahkan secara manual. Untuk mengatur ulang ke gaya default, gunakan reset().

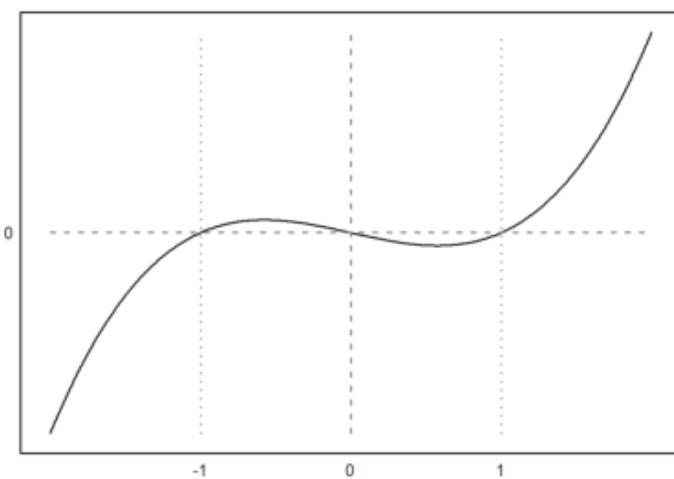
```
>aspect();
>figure(3,4); ...
    figure(1); plot2d("x^3-x",grid=0); ... // tanpa grid, bingkai, atau sumbu
> figure(2); plot2d("x^3-x",grid=1); ... // sumbu x-y
> figure(3); plot2d("x^3-x",grid=2); ... // centang default
> figure(4); plot2d("x^3-x",grid=3); ... // sumbu x-y dengan label di dalam
> figure(5); plot2d("x^3-x",grid=4); ... // tanpa centang, hanya label
> figure(6); plot2d("x^3-x",grid=5); ... // default, tanpa margin
> figure(7); plot2d("x^3-x",grid=6); ... // hanya sumbu
> figure(8); plot2d("x^3-x",grid=7); ... // hanya sumbu, centang di sumbu
> figure(9); plot2d("x^3-x",grid=8); ... // hanya sumbu, centang lebih halus di sumbu
> figure(10); plot2d("x^3-x",grid=9); ... // default, centang kecil di dalam
```

```
> figure(11); plot2d("x^3-x",grid=10); // tanpa centang, hanya sumbu
> figure(0):
```



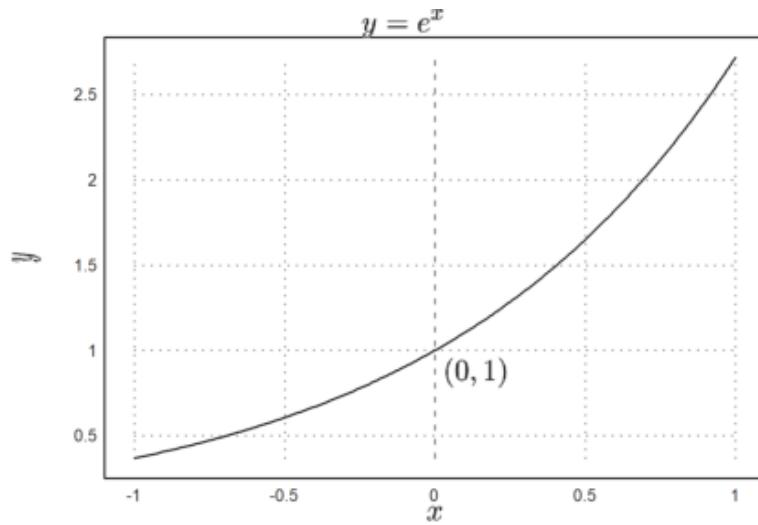
Parameter `<frame` mematikan bingkai, dan `framecolor=blue` mengatur bingkai ke warna biru. Jika Anda ingin menambahkan centang sendiri, Anda bisa menggunakan `style=0`, dan tambahkan semuanya kemudian.

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



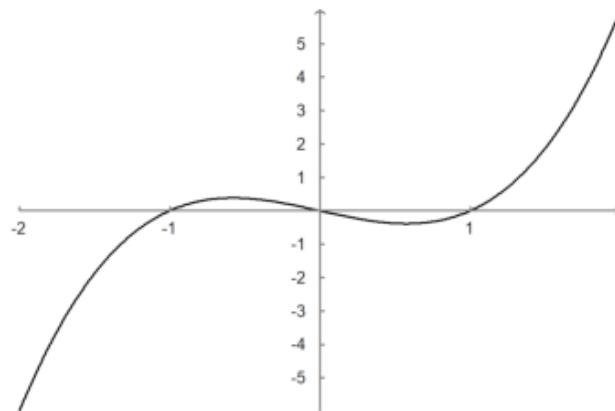
Untuk judul plot dan label sumbu, lihat contoh berikut.

```
>plot2d("exp(x)",-1,1);
>textcolor(black); // set the text color to black
>title(latex("y=e^x")); // title above the plot
>xlabel(latex("x")); // "x" for x-axis
>ylabel(latex("y"),>vertical); // vertical "y" for y-axis
>label(latex("(0,1")),0,1,color=blue); // label a point
```



Sumbu dapat digambar secara terpisah dengan `xaxis()` dan `yaxis()`.

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

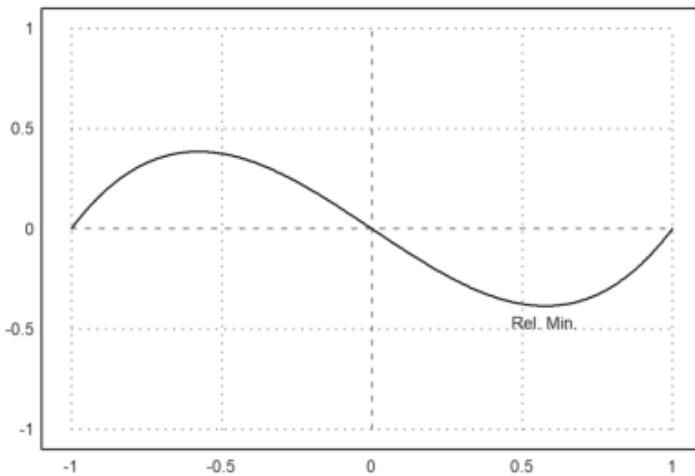


Teks pada plot dapat diatur dengan `label()`. Dalam contoh berikut, "lc" berarti lower center. Ini mengatur posisi label relatif terhadap koordinat plot.

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

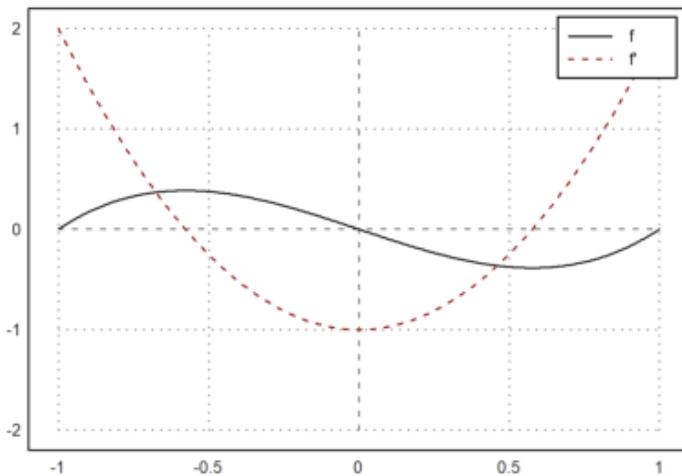
$$x^3 - x$$

```
>plot2d(f,-1,1,>square);
>x0=fmin(f,0,1); // compute point of minimum
>label("Rel. Min.",x0,f(x0),pos="lc"): // add a label there
```

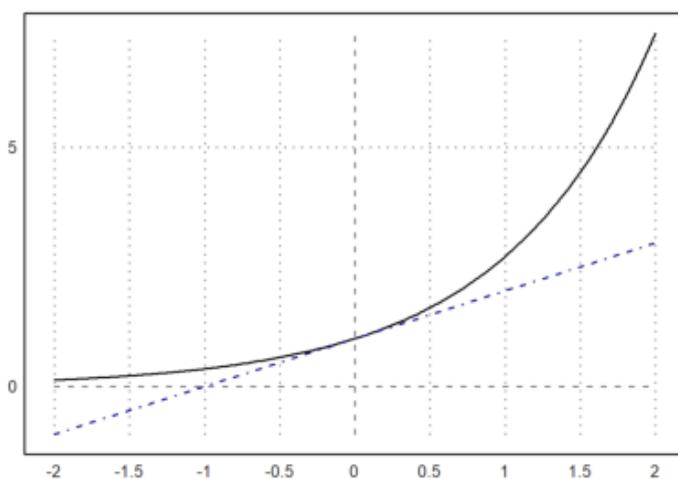


Ada juga kotak teks (text boxes).

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

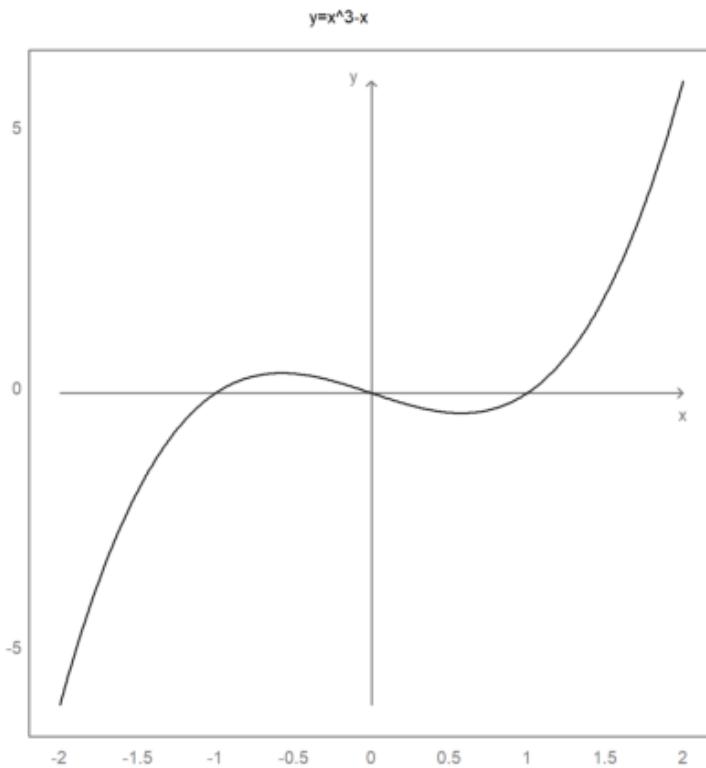


```
>plot2d(["exp(x)","1+x"],color=[black,blue],style=["-","-.-"]):
```



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

```
label("y",0,6,pos="cl",color=gray); ...
reset():
```



Untuk kontrol yang lebih mendetail, sumbu x dan sumbu y dapat dilakukan secara manual. Perintah `fullwindow()` memperluas jendela plot karena kita tidak lagi memerlukan tempat untuk label di luar jendela plot. Gunakan `shrinkwindow()` atau `reset()` untuk 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 contoh lainnya, di mana digunakan string Unicode 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",u"2\u03c0"],style="-",>ticks,>zero); ...
xgrid((0:0.5:2)*pi,<ticks); ...
yaxis(-0.1*pi,-1:0.2:1,style="-",>zero,>grid); ...
labelbox(["sin","cos"],colors=[red,green],x=0.5,y=0.2,>left); ...
xlabel(u"\u03c6"); ylabel(u" f(\u03c6) "):
```

Plotting Data 2D

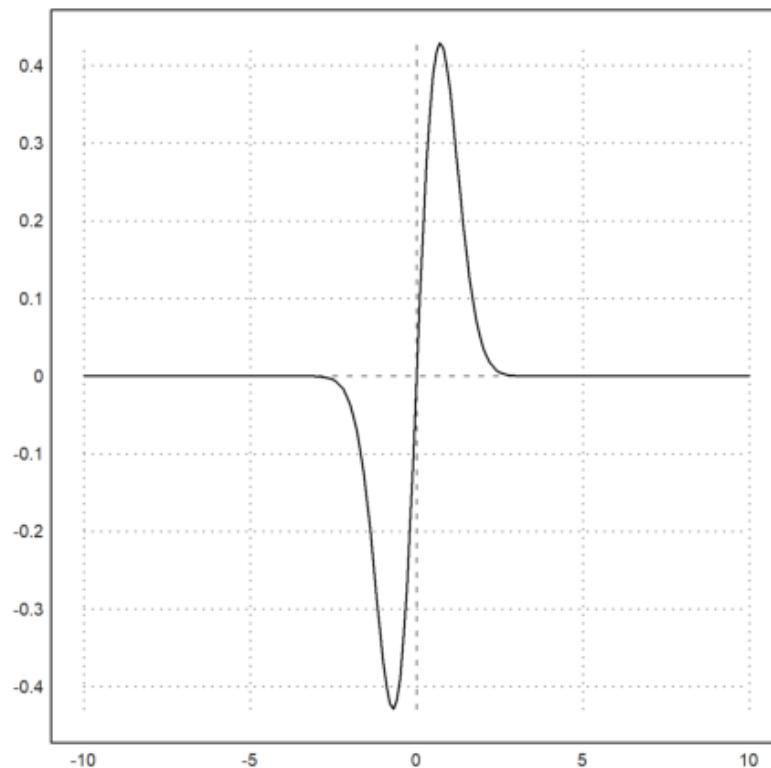
Jika x dan y adalah vektor data, data ini akan digunakan sebagai koordinat x dan y dari sebuah kurva. Dalam hal ini, a, b, c, dan d, atau radius r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Alternatifnya, square dapat diatur untuk menjaga rasio aspek persegi.

Memplot ekspresi hanyalah singkatan untuk 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 dipetakan, secara default dengan evaluasi adaptif terhadap fungsi tersebut. Untuk plot titik gunakan "`>points`", untuk garis dan titik campuran gunakan "`>addpoints`".

Namun, Anda dapat memasukkan data secara langsung.
- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y akan dipetakan 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):
```



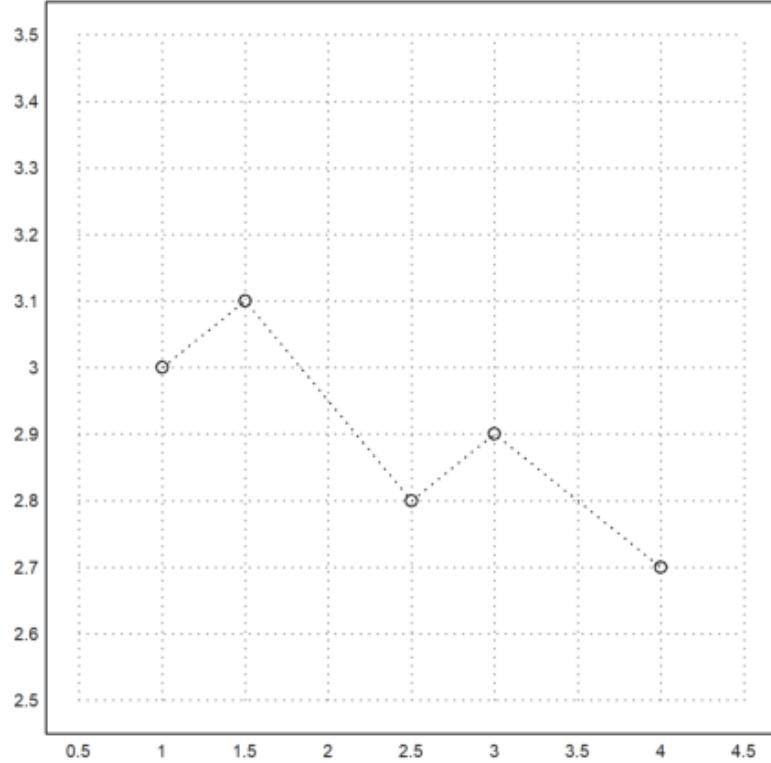
Data juga dapat dipetakan sebagai titik-titik. Gunakan points=true untuk ini. Plot bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya.

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

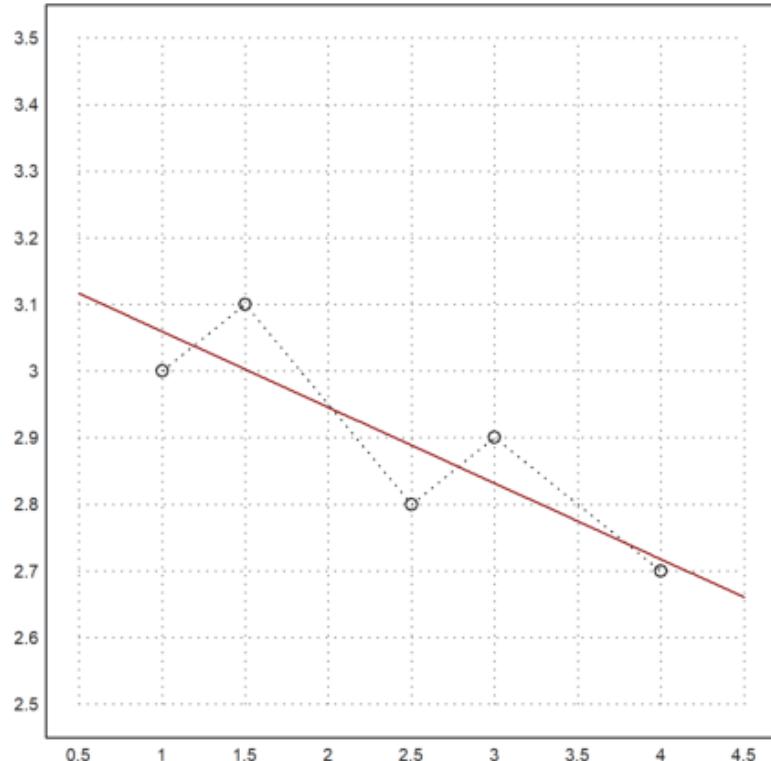
Untuk memetakan kumpulan titik gunakan points. Jika warna adalah vektor warna, setiap titik akan memiliki warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna berlaku untuk baris-baris matriks.

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="."); // lines
>plot2d(xdata,ydata,>points,>add,style="o"); // add points
```



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



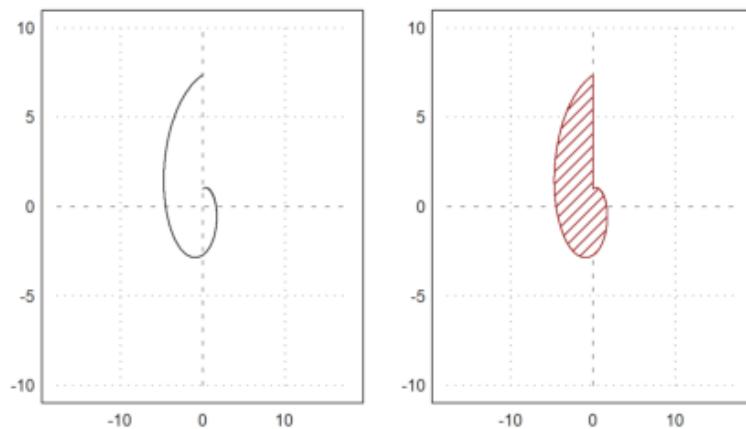
Menggambar Daerah Yang Dibatasi Kurva

Plot data sebenarnya adalah poligon. Kita juga bisa memplot kurva atau kurva terisi (filled curves).

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

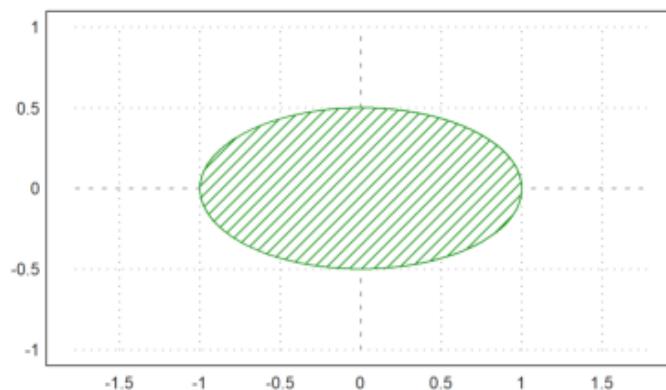
Warna pengisian ditentukan oleh argumen "fillcolor", dan opsional <outline mencegah menggambar batas untuk semua gaya kecuali gaya default.

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

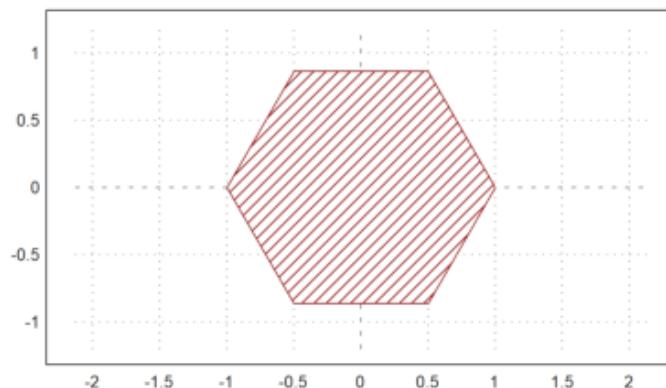


Dalam contoh berikut, kita memplot sebuah elips yang terisi dan dua heksagon terisi menggunakan kurva tertutup dengan 6 titik dan gaya pengisian yang berbeda.

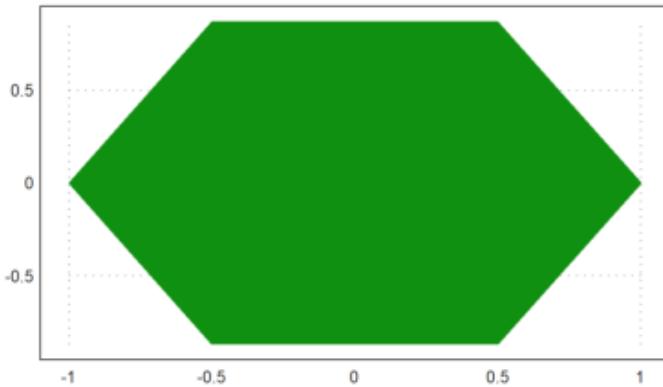
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```



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



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

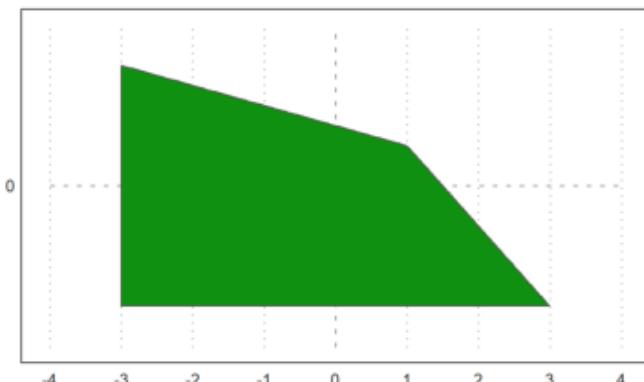


Contoh berikut adalah sebuah septagon, yang kita buat dengan 7 titik pada lingkaran satuan.

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

Berikut adalah set nilai maksimal dari empat kondisi linear yang kurang dari atau sama dengan 3. Ini adalah $A[k].v \leq 3$ untuk semua baris dari A . Untuk mendapatkan 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):
```

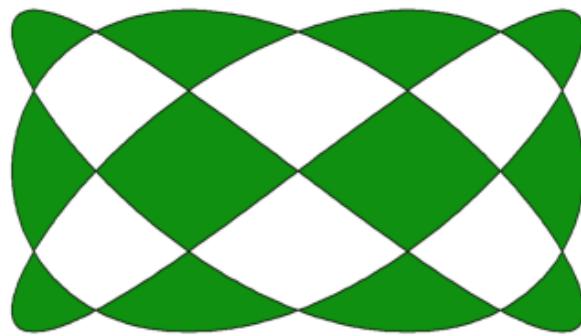


Poin utama dari bahasa matriks adalah memungkinkan kita untuk menghasilkan tabel fungsi dengan mudah.

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

Sekarang kita memiliki vektor x dan y dari nilai. Fungsi `plot2d()` dapat menggambar nilai-nilai ini sebagai kurva yang menghubungkan titik-titik tersebut. Plot dapat diisi. Dalam hal ini, hasilnya menjadi indah karena aturan pengisian, yang digunakan untuk mengisi.

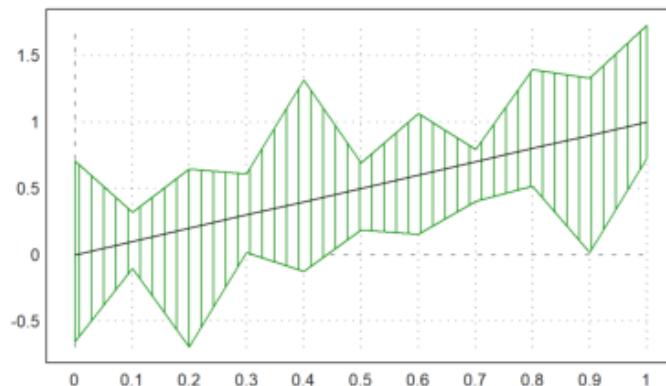
```
>plot2d(x,y,<grid,<frame,>filled):
```



Sebuah vektor interval diplotkan terhadap nilai x sebagai daerah terisi antara nilai bawah dan atas dari interval.

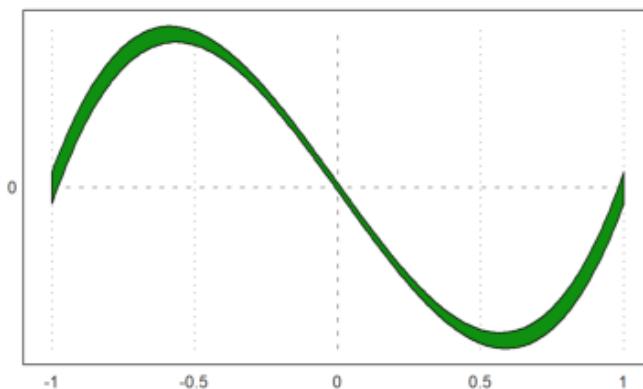
Ini berguna untuk menggambarkan kesalahan perhitungan. Namun, ini juga dapat digunakan untuk memplot kesalahan statistik.

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



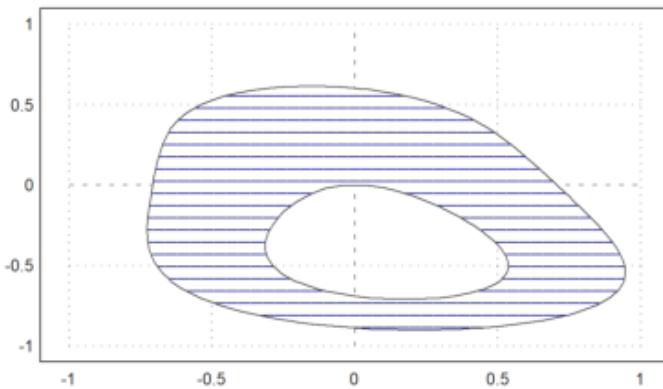
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d() akan memplot rentang interval yang terisi di dalam bidang. Gaya pengisian sama dengan gaya poligon.

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



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

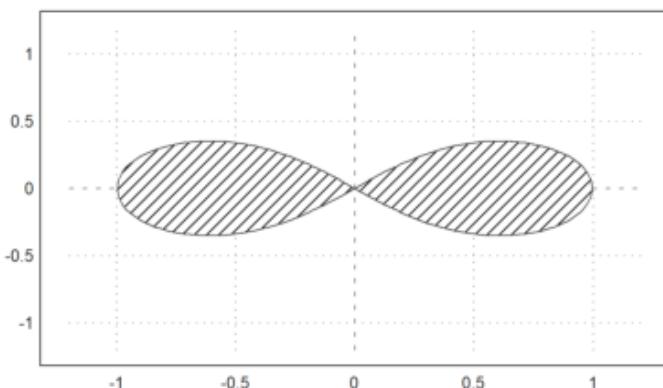
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue); // 0 <= f(x,y) <= 1
```



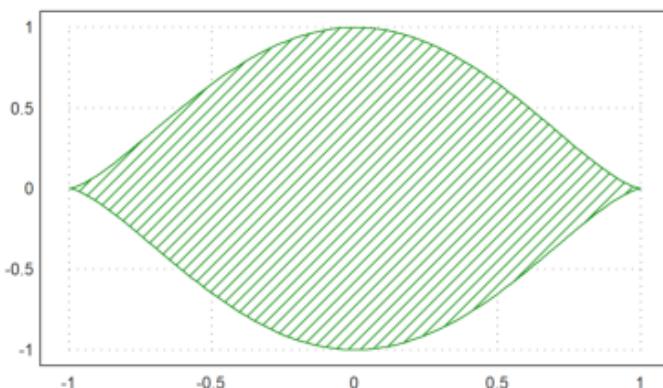
Kita juga dapat mengisi rentang nilai seperti:

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

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



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



Grafik Fungsi Parametrik

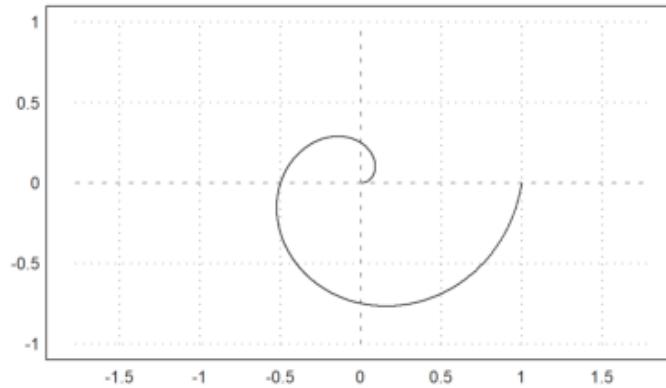
Nilai x tidak perlu diurutkan. (x, y) hanya menggambarkan sebuah kurva. Jika x diurutkan, maka kurva tersebut adalah grafik suatu fungsi.

Pada contoh berikut, kita menggambar spiral.

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

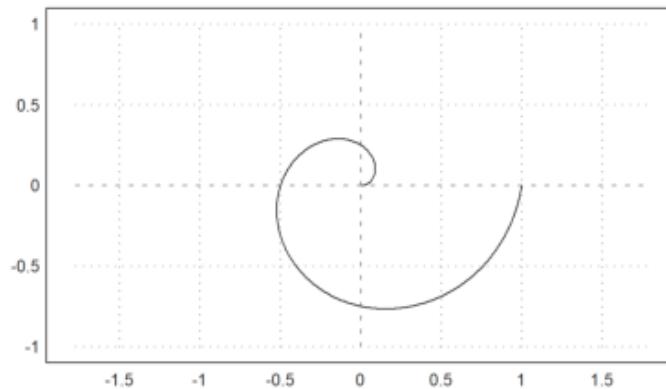
Kita perlu menggunakan banyak titik untuk tampilan yang halus atau menggunakan fungsi adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk lebih detail).

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

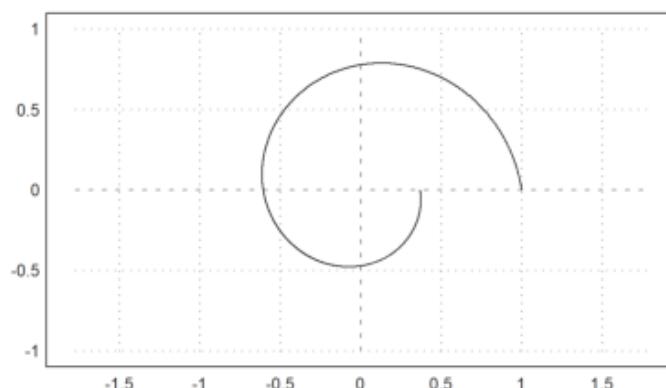


Alternatifnya, dimungkinkan menggunakan dua ekspresi untuk kurva. Berikut ini adalah plot yang sama seperti di atas.

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



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



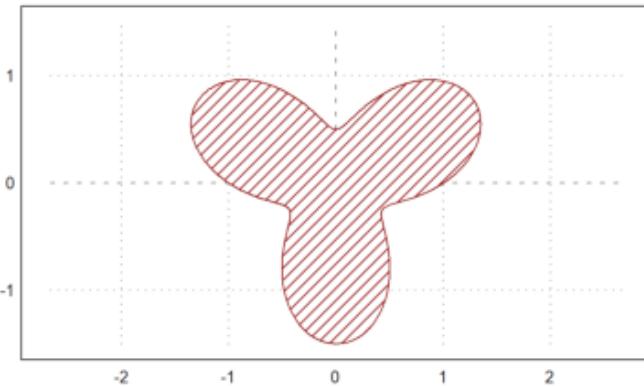
Pada contoh berikut, kita menggambar kurva:

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

dengan

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

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

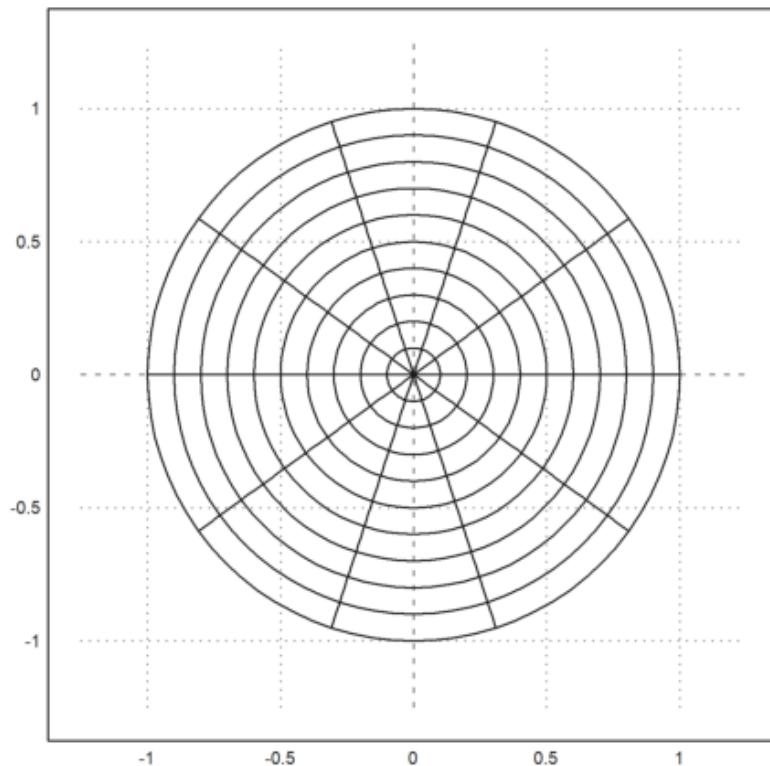


Menggambar Grafik Bilangan Kompleks

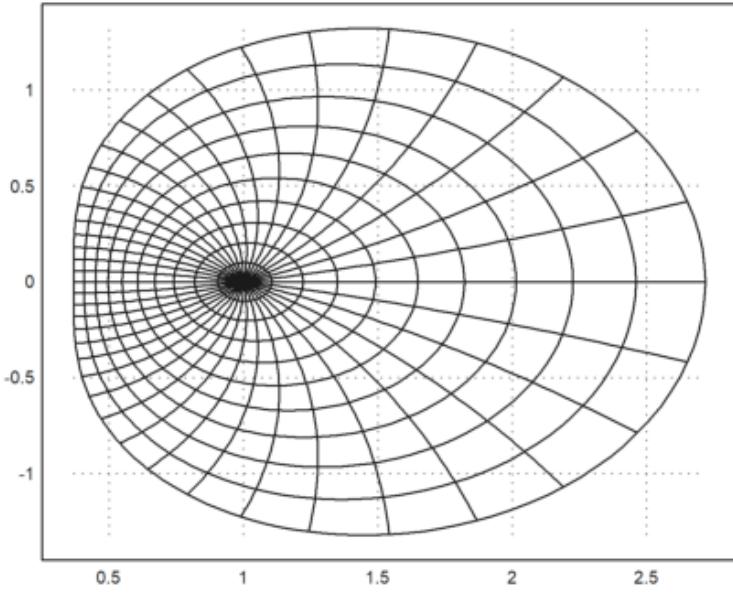
Sebuah array dari bilangan kompleks juga dapat diplotkan. Titik-titik grid akan terhubung. Jika sejumlah garis grid ditentukan (atau vektor 1x2 dari garis grid) dalam argumen cgrid, hanya garis-garis grid tersebut yang akan terlihat. Matriks bilangan kompleks secara otomatis akan diplot sebagai grid di bidang kompleks.

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

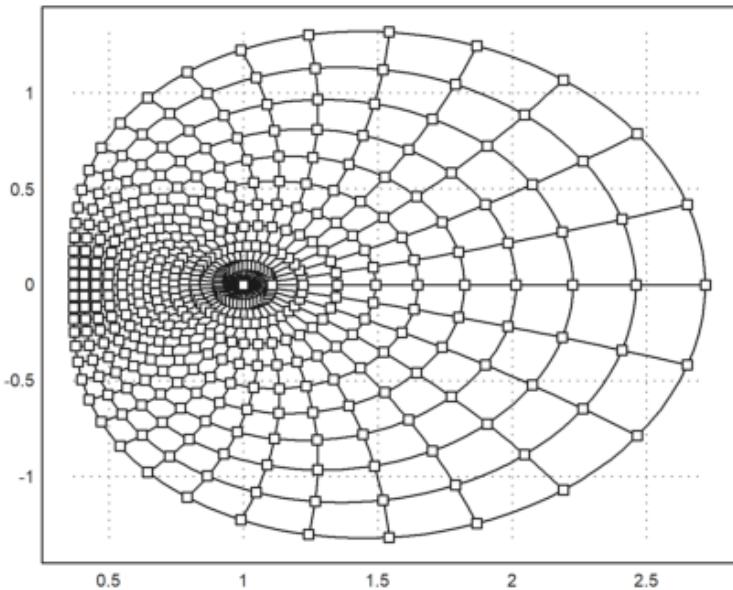
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```



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



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



Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva di bidang kompleks dengan bagian nyata dan bagian imajiner.

Pada contoh ini, kita menggambar lingkaran satuan dengan

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

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

Plot Statistik

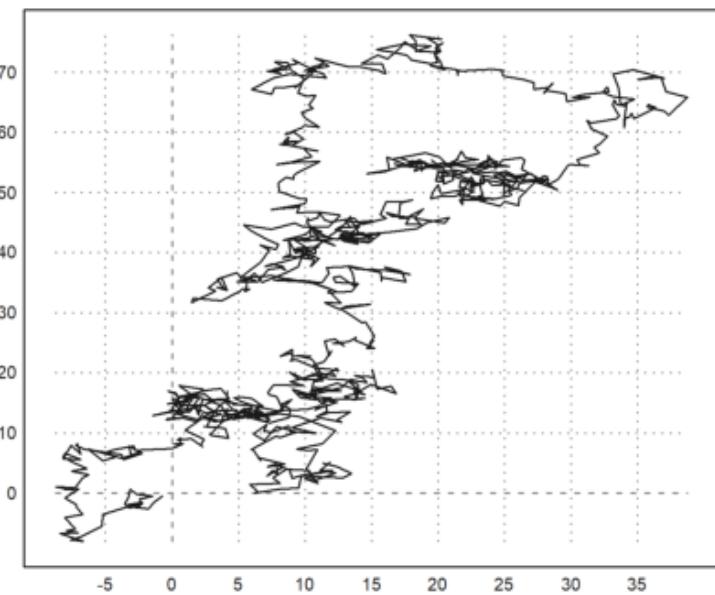
Ada banyak fungsi yang dikhususkan untuk plot statistik. Salah satu plot yang sering digunakan adalah plot kolom. Penjumlahan kumulatif dari nilai yang terdistribusi normal 0-1 menghasilkan gerak acak.

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

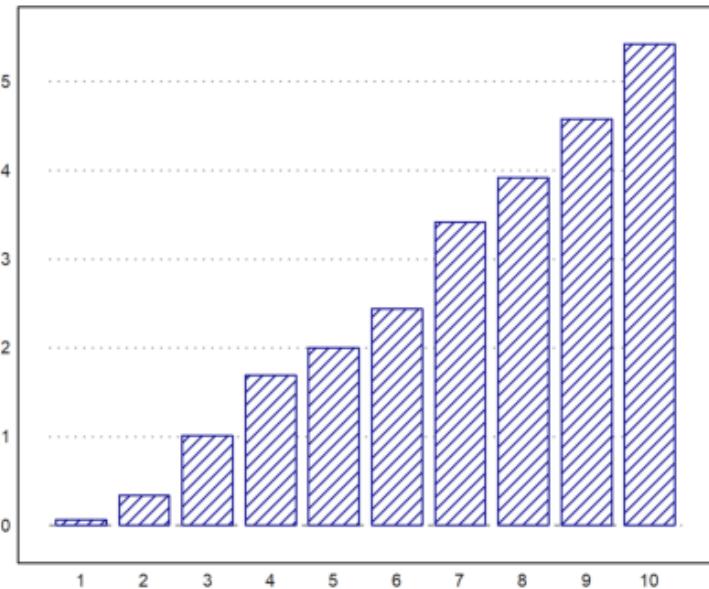


Dengan menggunakan dua baris, kita menunjukkan gerak acak dalam dua dimensi.

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

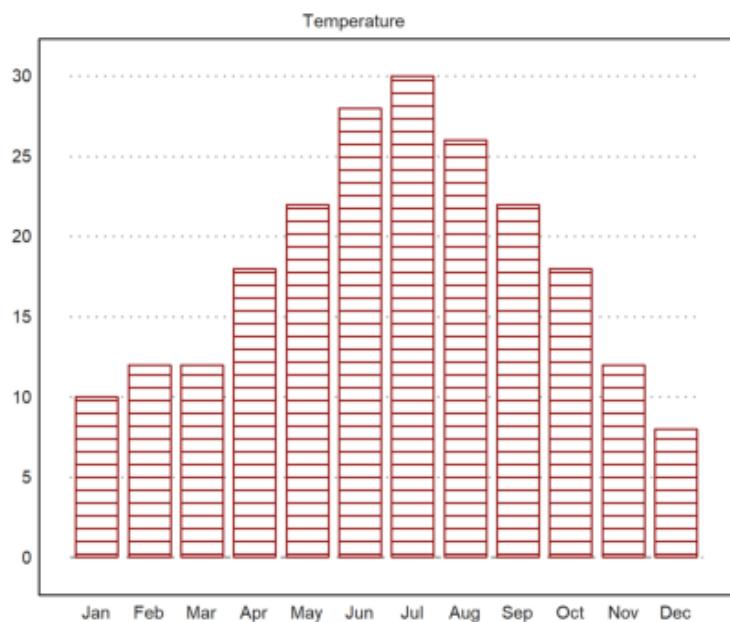


```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

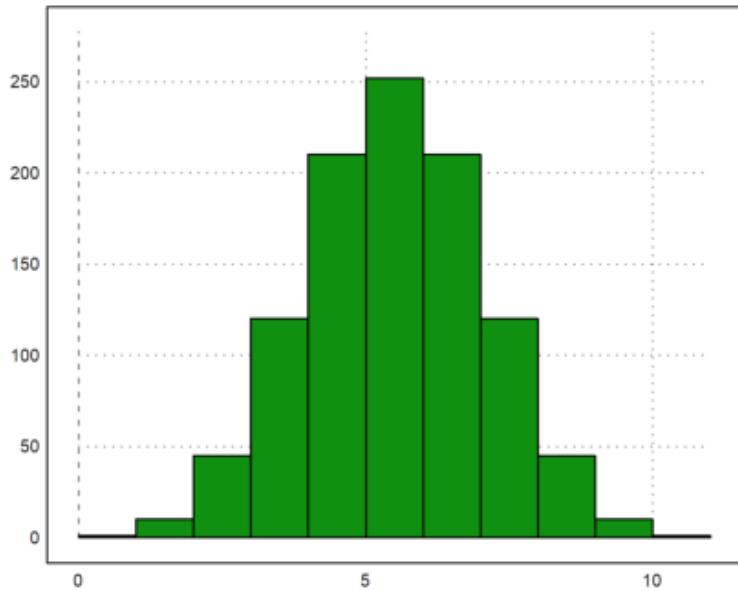


Plot ini juga dapat menampilkan string sebagai label.

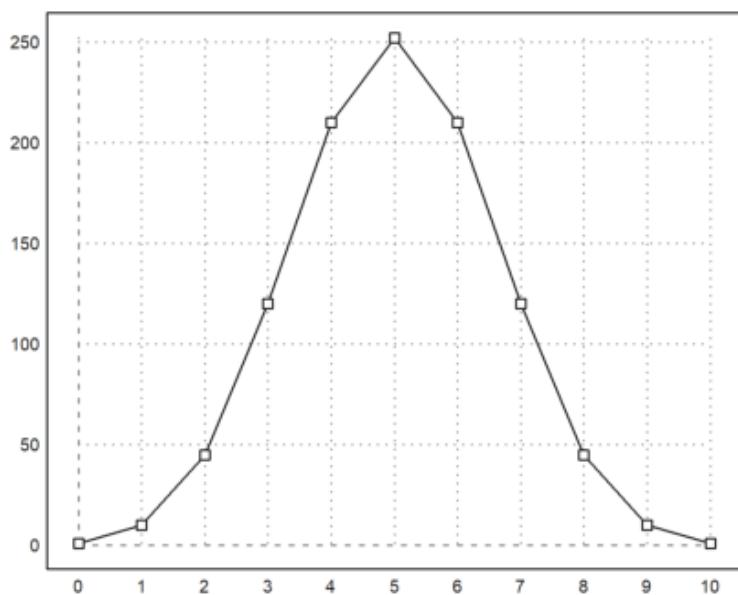
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
"Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnspplot(values,lab=months,color=red,style="-");
>title("Temperature");
```



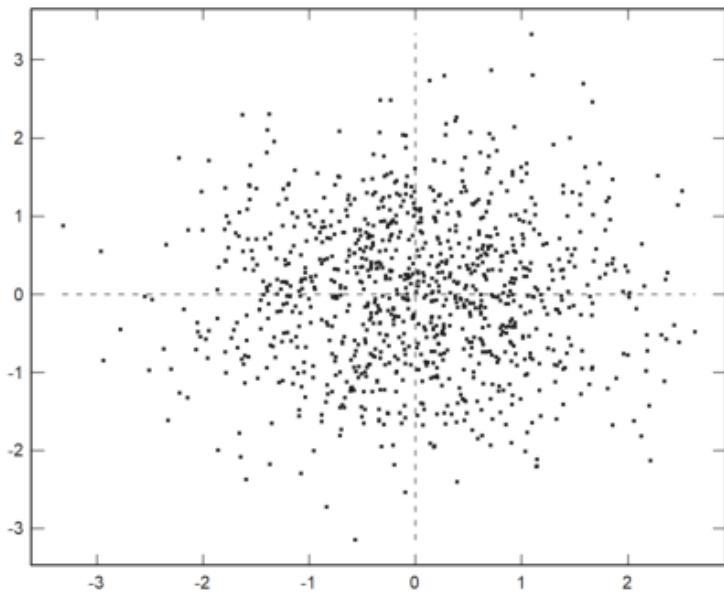
```
>k=0:10;
>plot2d(k,bin(10,k),>bar);
```



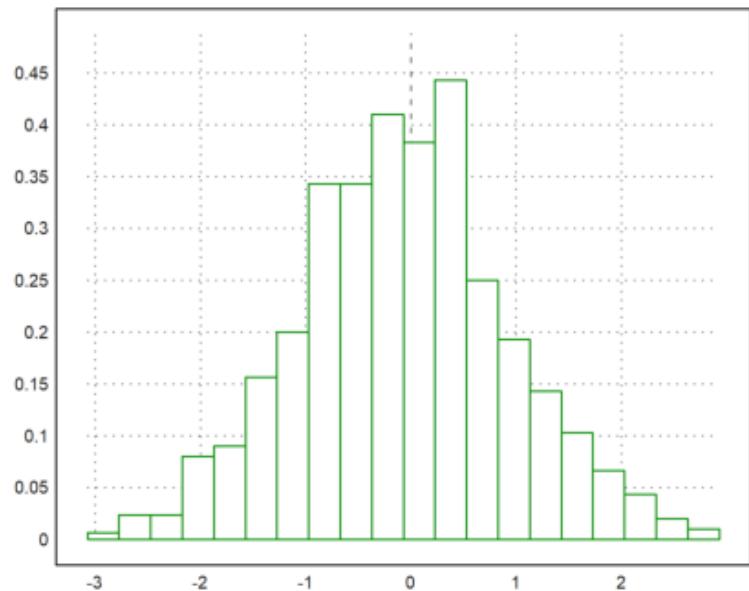
```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



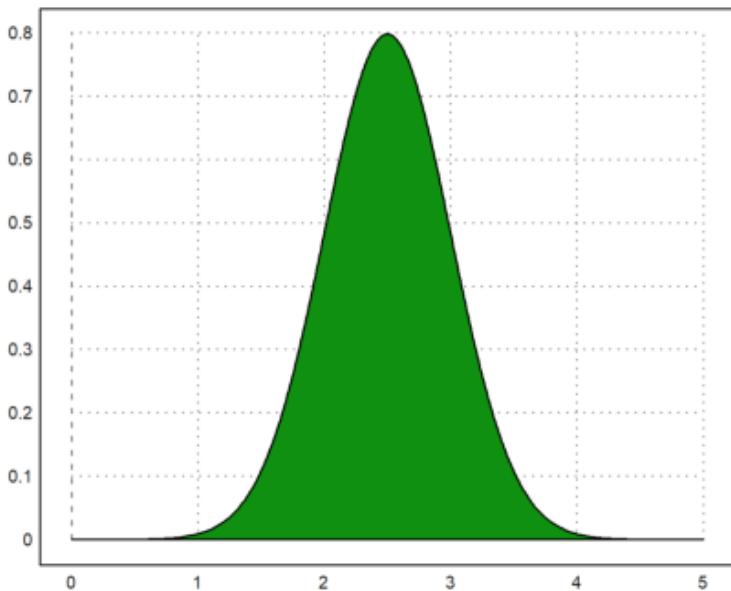
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style="."):
```



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

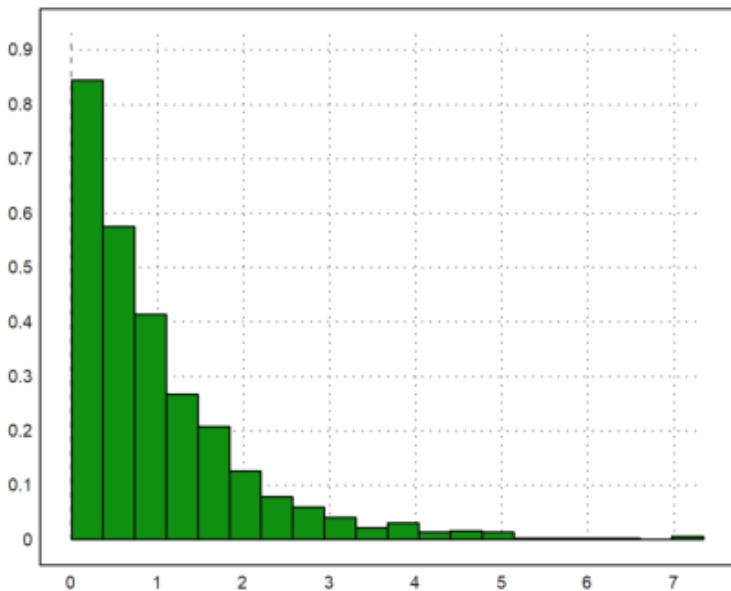


```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



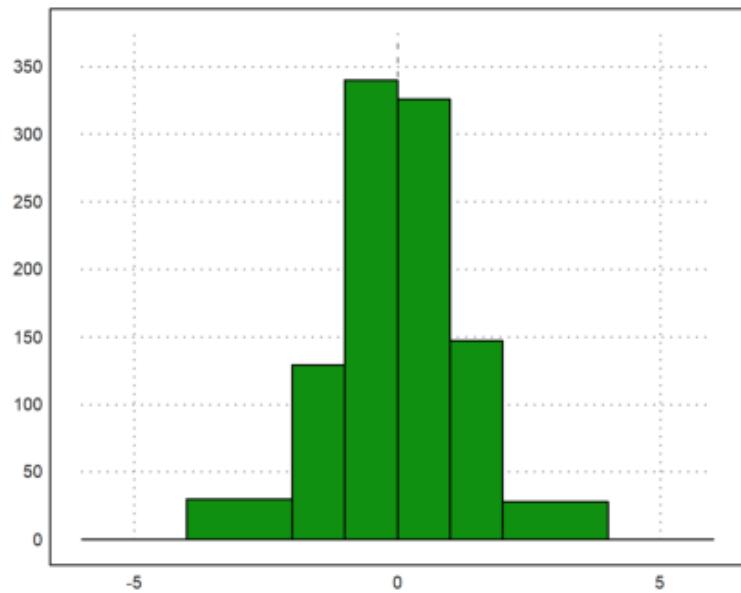
Distribusi eksperimen statistik dapat menggunakan `distribution=n` dengan `plot2d`.

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



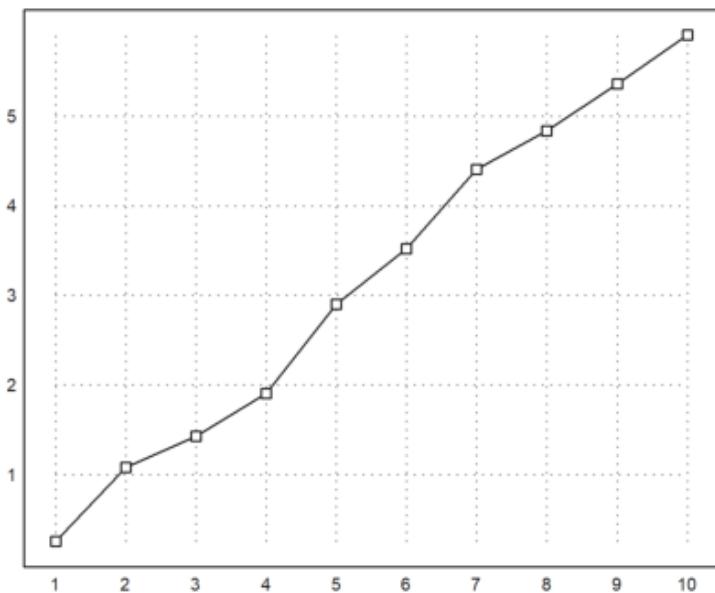
Atau kita bisa menghitung distribusi dari data dan memplot hasilnya dengan menggunakan `bar` dalam `plot2d` atau `column plot`.

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

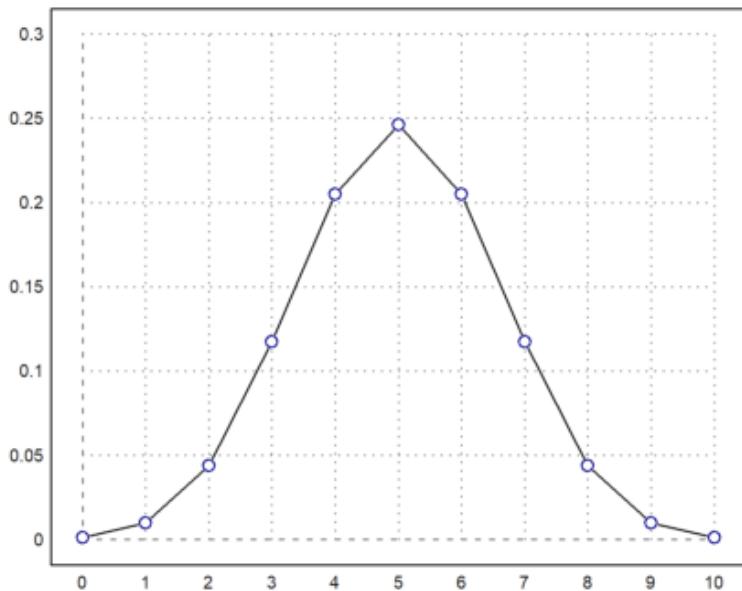


Fungsi statplot() menetapkan gaya dengan string sederhana.

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

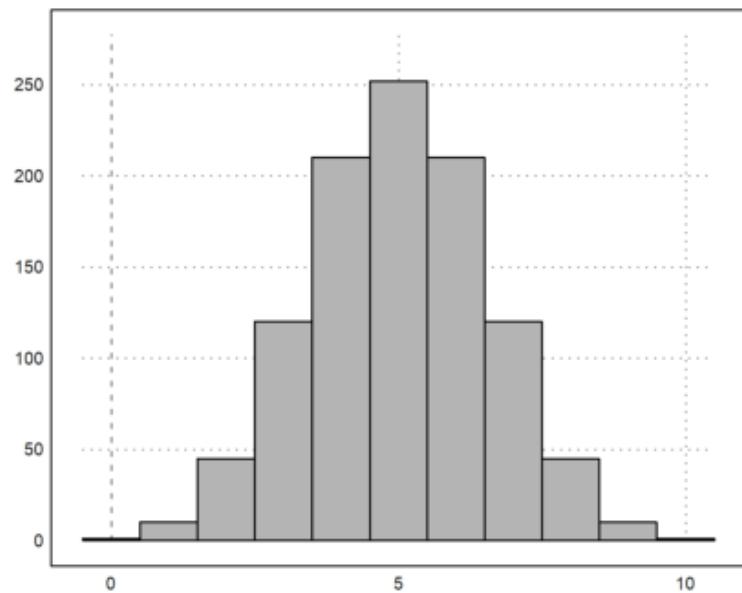


```
>n=10; i=0:n; ...
plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



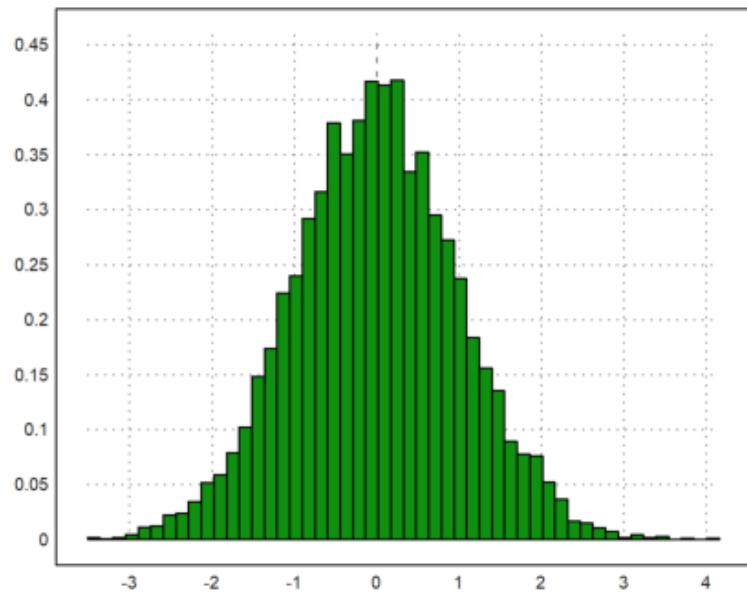
Selain itu, data dapat diplot sebagai batang (bars). Dalam hal ini, x harus diurutkan dan satu elemen lebih panjang dari y. Batang akan diperpanjang dari $x[i]$ hingga $x[i+1]$ dengan nilai $y[i]$. Jika x memiliki ukuran yang sama dengan y, maka akan diperpanjang dengan satu elemen dengan jarak terakhir.

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

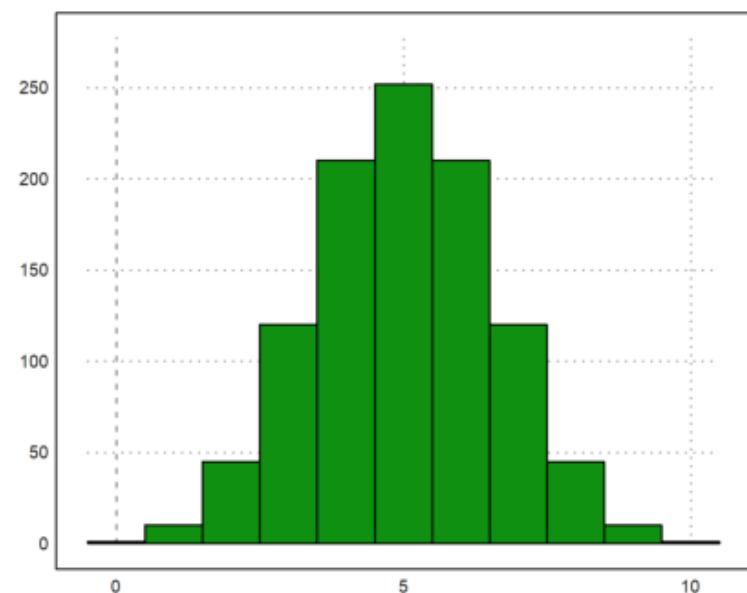


Data untuk plot batang (bar=1) dan histogram (histogram=1) dapat diberikan secara eksplisit dalam xv dan yy, atau dihitung dari distribusi empiris dalam xv dengan distribution=n. Histogram dari nilai-nilai xv akan dihitung secara otomatis dengan histogram. Jika even=true ditentukan, nilai-nilai xv akan dihitung dalam interval integer.

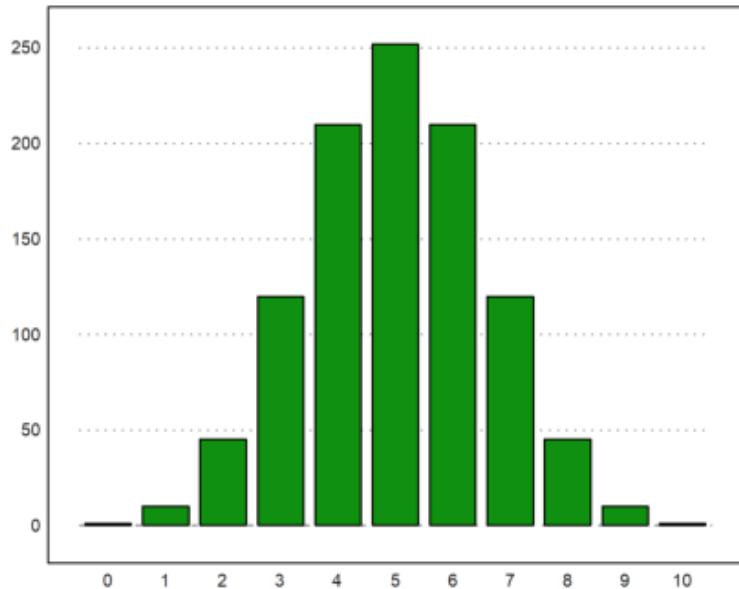
```
>plot2d(normal(10000),distribution=50) :
```



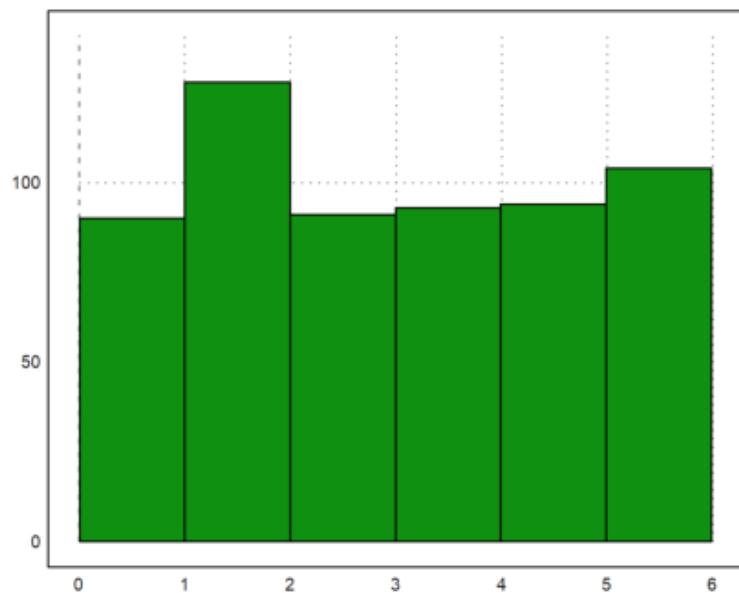
```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```



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

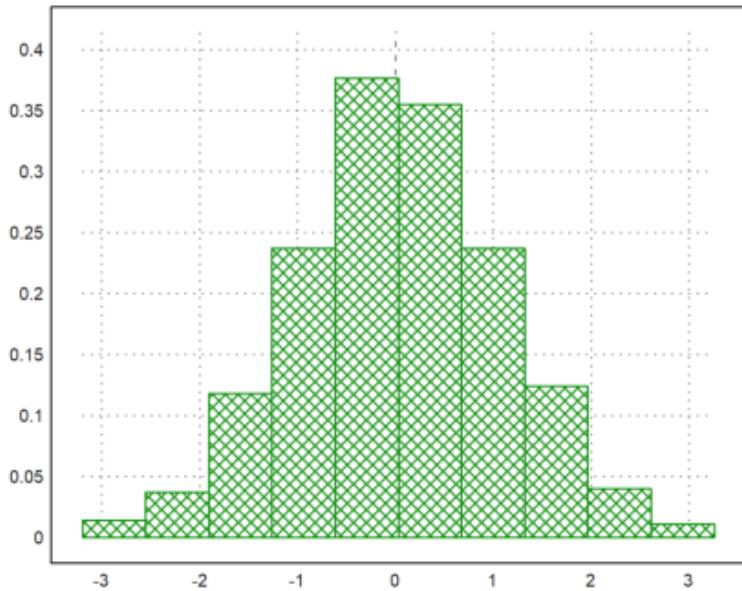


```
>plot2d(random(600)*6,histogram=6):
```



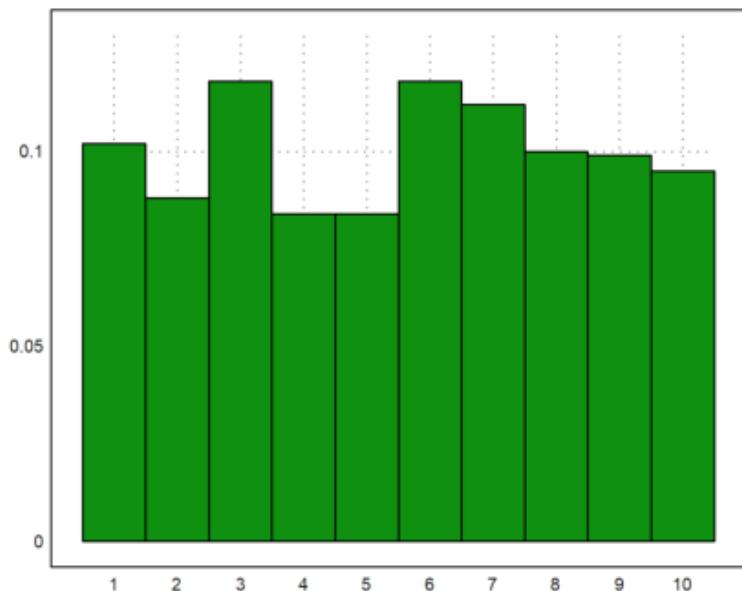
Untuk distribusi, terdapat parameter `distribution=n`, yang secara otomatis menghitung nilai dan mencetak distribusi relatif dengan n sub-interval.

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



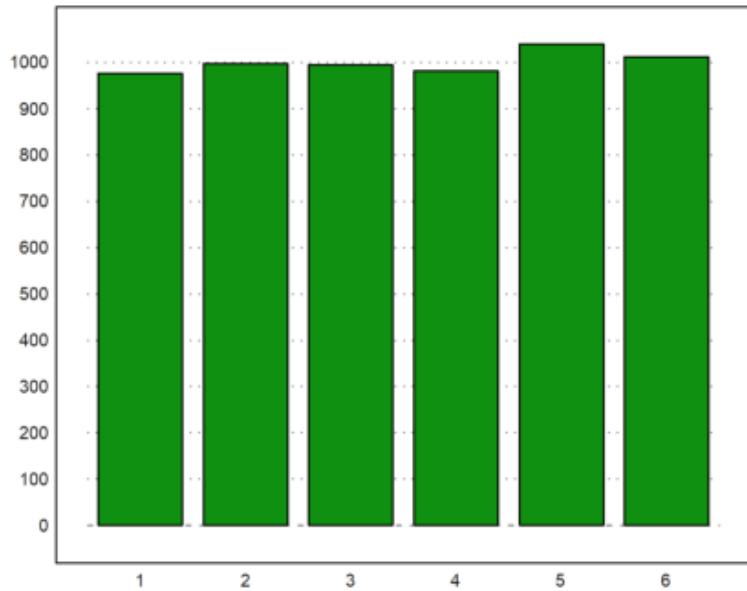
Dengan parameter even=true, ini akan menggunakan interval bilangan bulat.

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

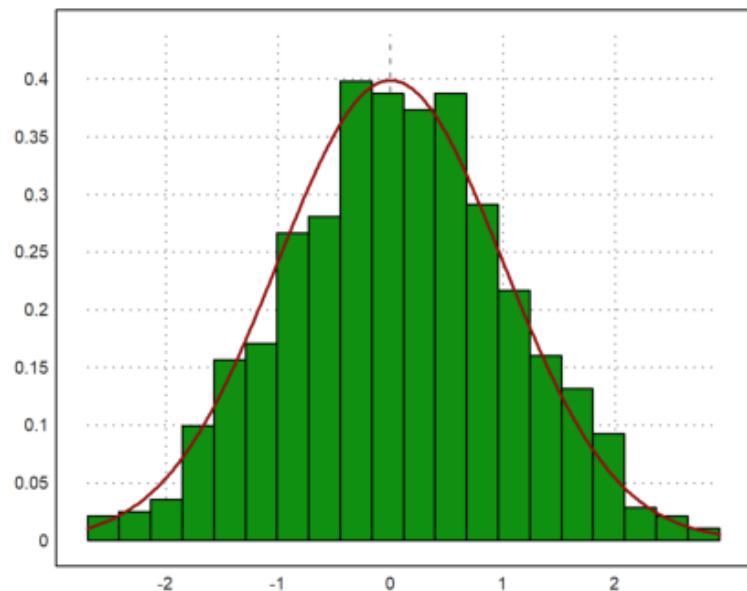


Perlu diingat bahwa ada banyak plot statistik yang mungkin berguna. Lihatlah tutorial tentang statistik.

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

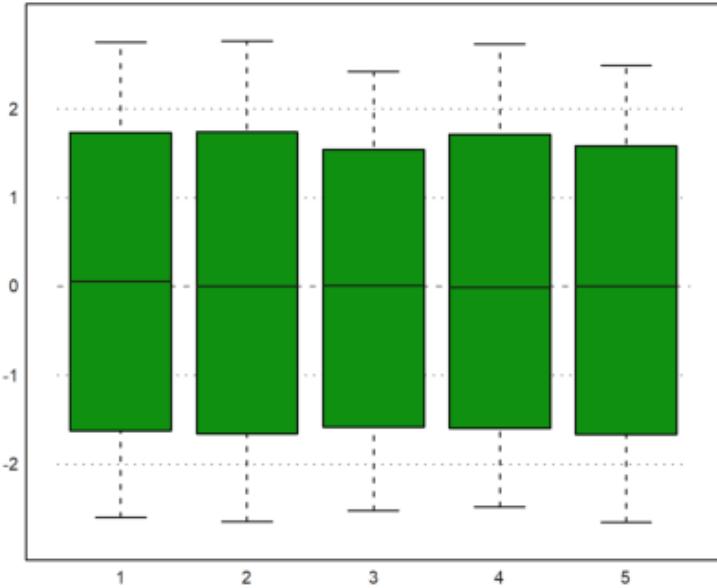


```
>plot2d(normal(1,1000),>distribution); ...
  plot2d("qnormal(x)",color=red,thickness=2,>add) :
```



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

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



Fungsi Implisit

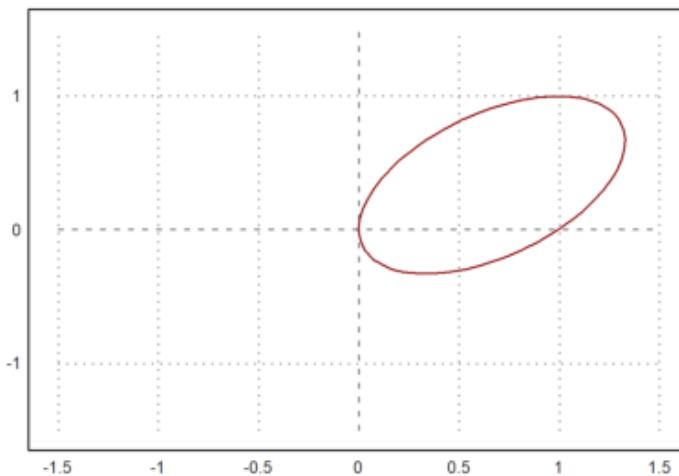
Plot implisit menampilkan garis level yang menyelesaikan persamaan $f(x,y)=\text{level}$, dimana "level" dapat berupa nilai tunggal atau vektor nilai. Jika level="auto", akan ada nc garis level, yang akan tersebar merata antara nilai minimum dan maksimum fungsi. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan hue untuk menunjukkan nilai fungsi. Untuk fungsi implisit, xv harus berupa fungsi atau ekspresi dari parameter x dan y, atau xv bisa berupa matriks nilai.

Euler dapat menandai garis level dari fungsi apapun seperti

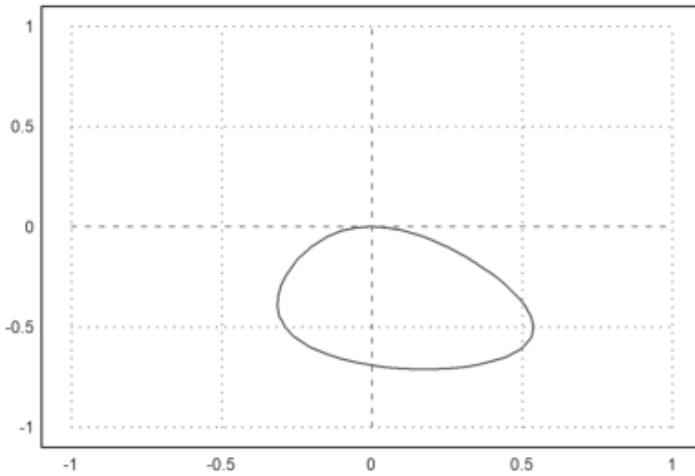
$$f(x, y) = c$$

Untuk satu atau lebih konstanta c , Anda bisa menggunakan `plot2d()` dengan plot implisitnya di bidang. Parameter untuk c adalah `level=c`, di mana c bisa menjadi vektor dari garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi untuk setiap titik dalam plot. Parameter n menentukan tingkat ketelitian plot.

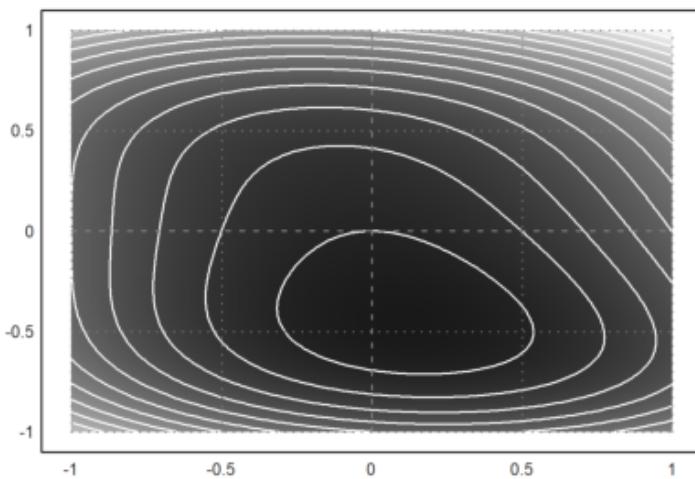
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



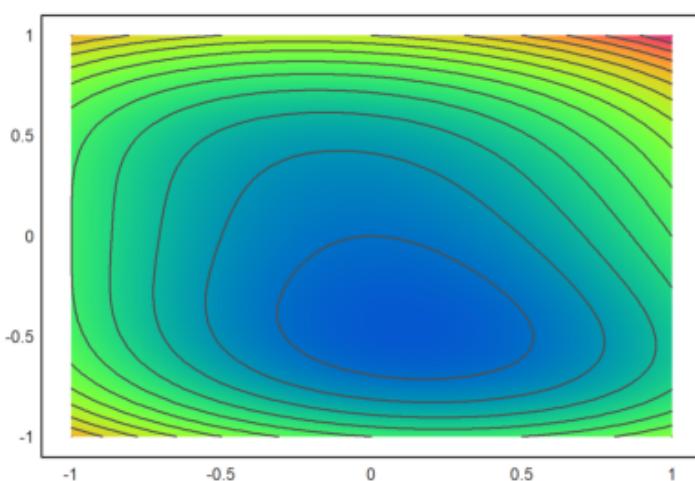
```
>expr := "2*x^2+x*y+3*y^4+y"; // definisikan ekspresi f(x,y)
>plot2d(expr,level=0); // solusi dari f(x,y)=0
```



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

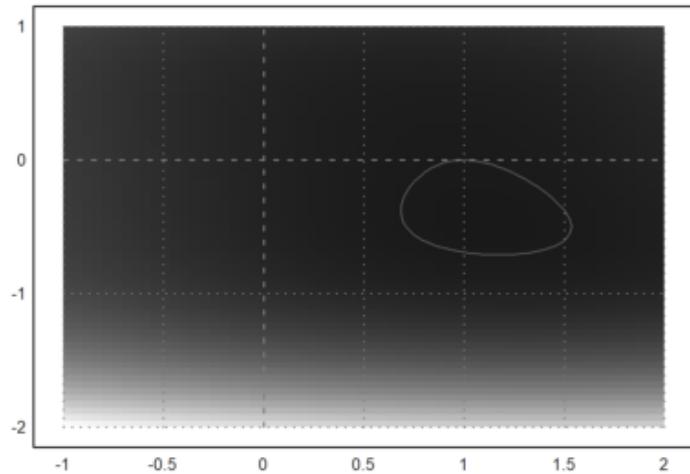


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // tampilan yang lebih baik
```

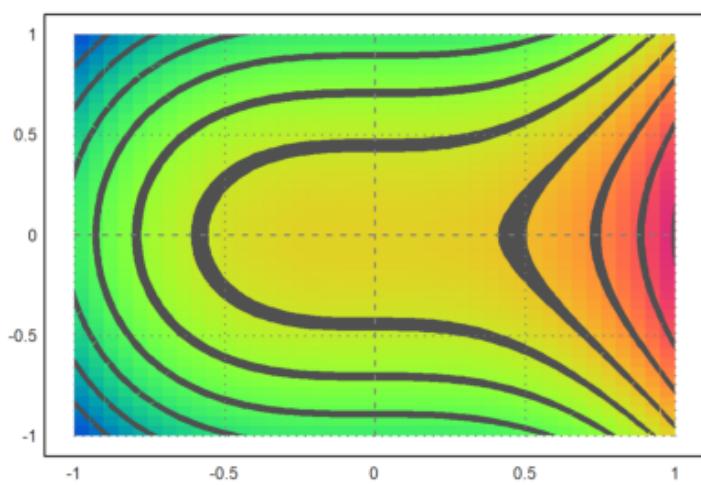


Ini juga bekerja untuk plot data. Namun, Anda harus menentukan rentang untuk label sumbu.

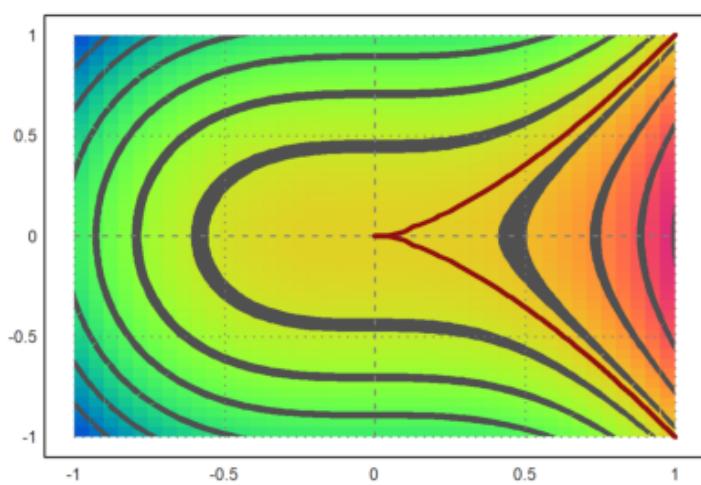
```
>x=-2:0.05:1; y=x'; z=expr(x,y);  
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



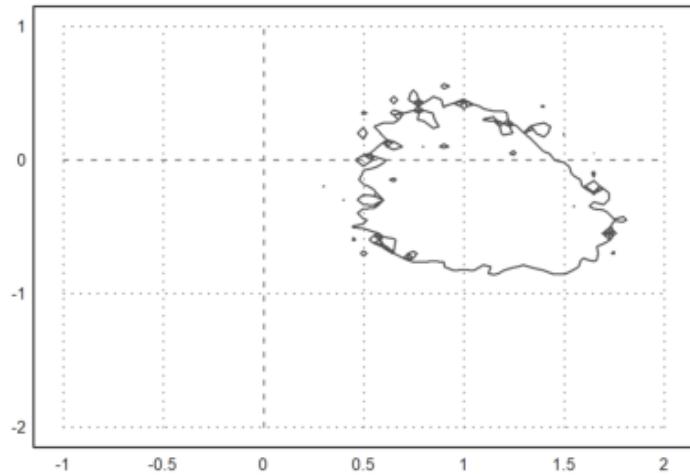
```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



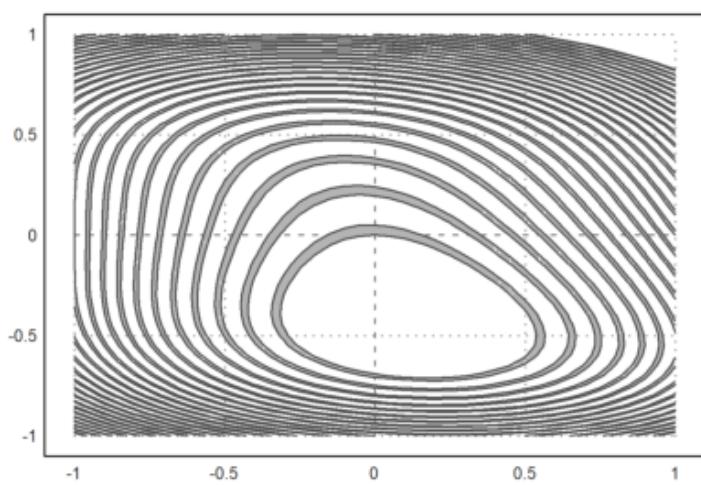
```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



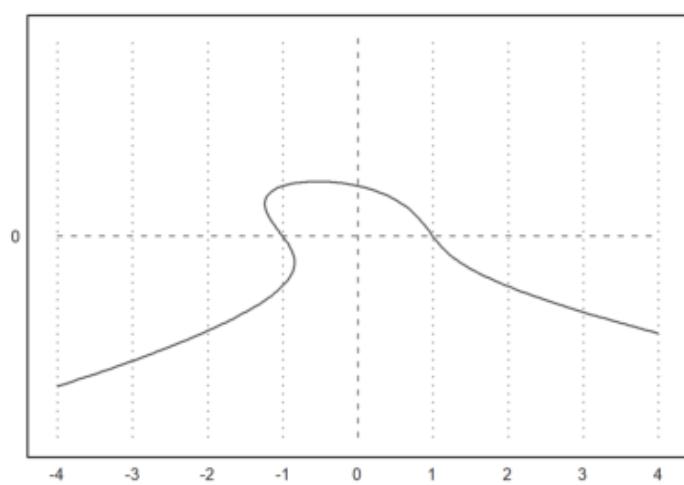
```
>z=z+normal(size(z))*0.2;  
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



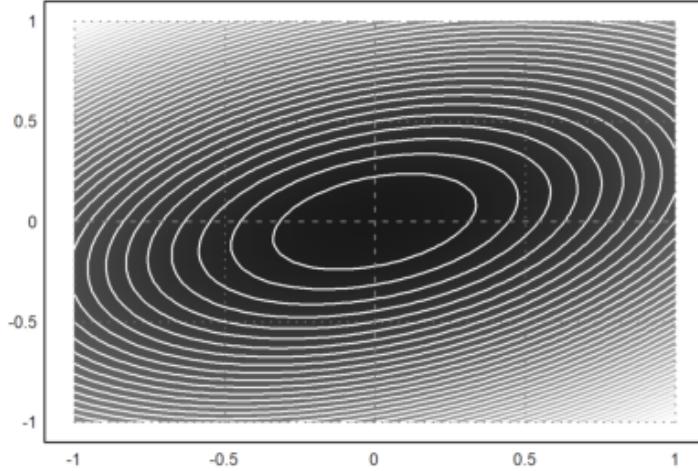
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



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



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



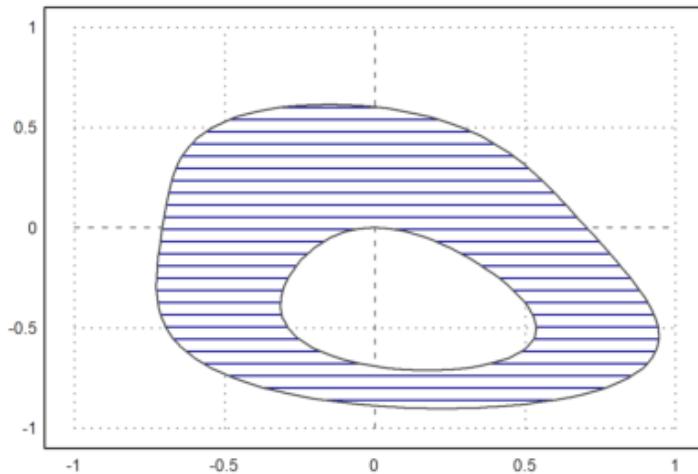
Dimungkinkan juga untuk mengisi himpunan

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

dengan rentang level

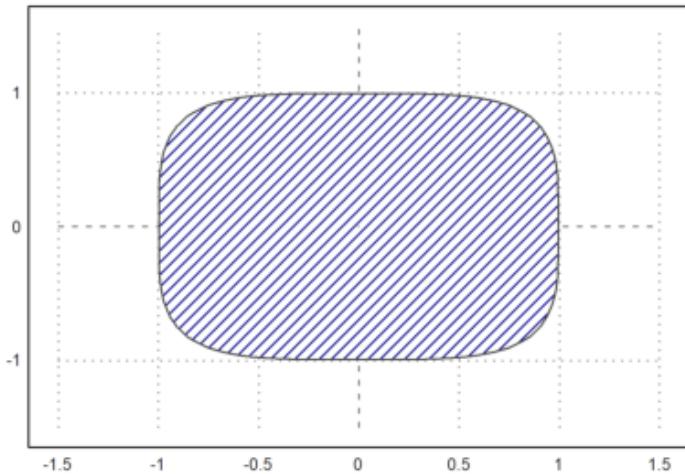
Dimungkinkan untuk mengisi area nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

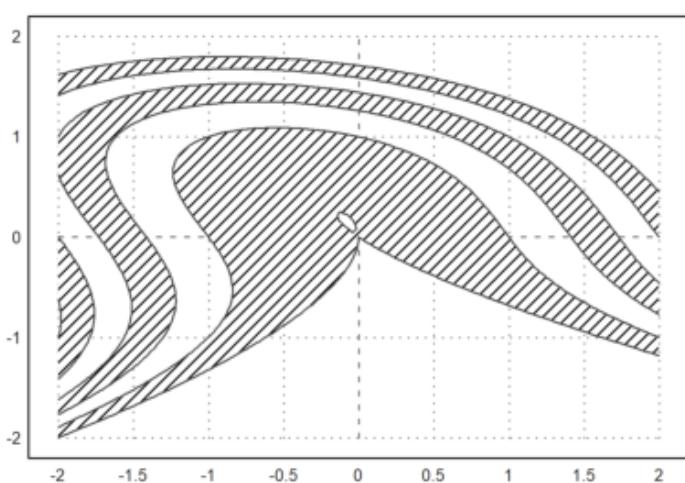


Plot implisit juga dapat menunjukkan rentang level. Maka level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua berisi akhir dari setiap interval. Sebagai alternatif, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level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

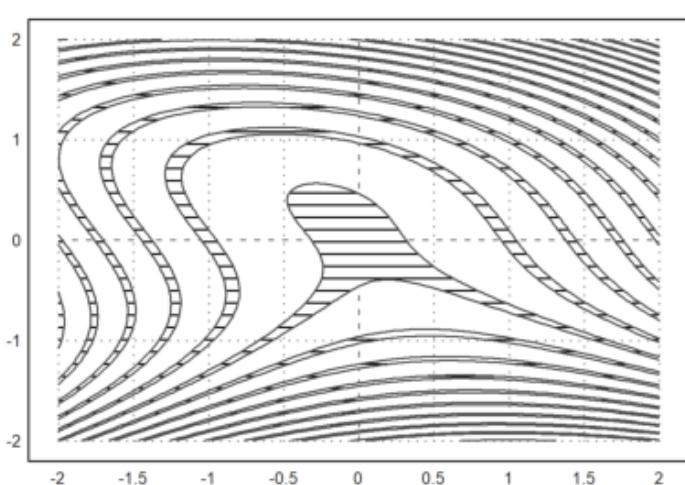
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



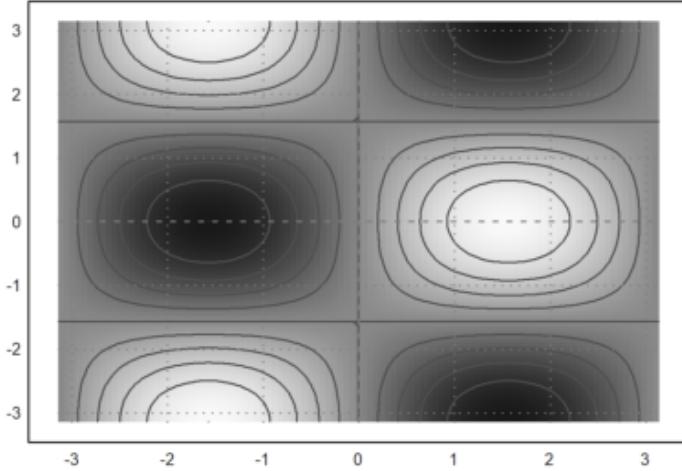
```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```



```
>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):
```

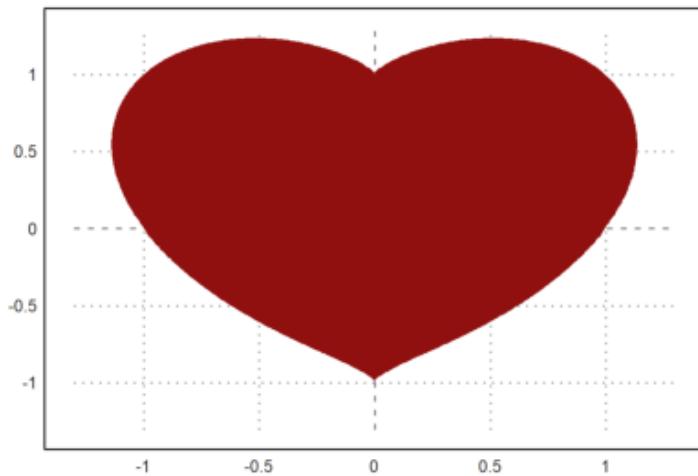


Dimungkinkan juga untuk menandai area

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

Ini dilakukan dengan menambahkan level dengan dua baris.

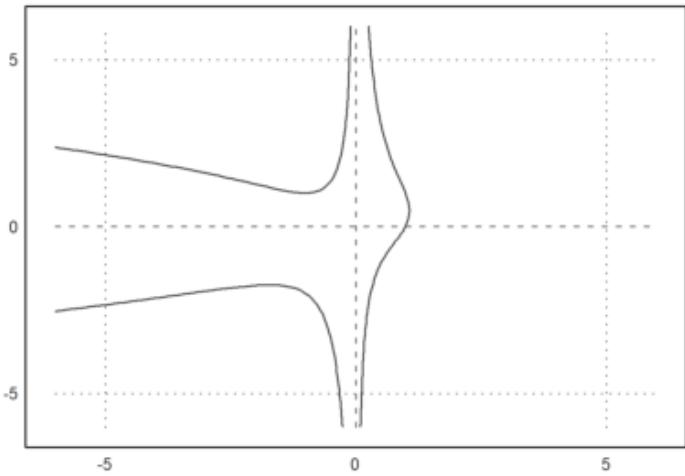
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
  style="#",color=red,<outline, ...
  level=[-2;0],n=100):
```



Dimungkinkan untuk menentukan level tertentu. Misalnya, kita dapat memplot solusi dari persamaan seperti:

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

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

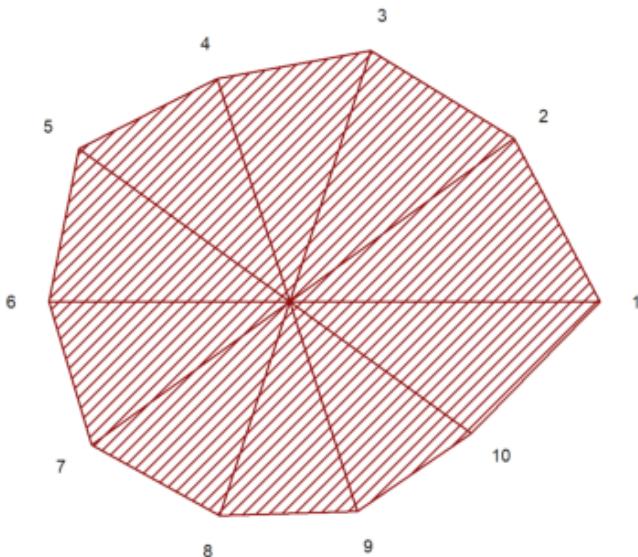


```
>function starplot1 (v, style="/", color=green, lab=none) ...
  if !holding() then clg; endif;
  w=window(); window(0,0,1024,1024);
  h=holding(1);
  r=max(abs(v))*1.2;
  setplot(-r,r,-r,r);
  n=cols(v); t=linspace(0,2pi,n);
  v=v|v[1]; c=v*cos(t); s=v*sin(t);
  cl=barcolor(color); st=barstyle(style);
  loop 1 to n
    polygon([0,c[#],c[#+1]],[0,s[#],s[#+1]],1);
    if lab!=none then
      rlab=v[#]+r*0.1;
      {col,row}=toscreen(cos(t[#])*rlab,sin(t[#])*rlab);
      ctext(""+lab#[],col,row-textheight()/2);
    endif;
  end;
  barcolor(cl); barstyle(st);
  holding(h);
  window(w);
endfunction
```

Tidak ada grid atau tanda sumbu di sini. Selain itu, kita menggunakan seluruh jendela untuk plot.

Kita memanggil reset sebelum menguji plot ini untuk mengembalikan default grafis. Ini tidak diperlukan jika Anda yakin plot Anda berfungsi.

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



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

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

```
>function logimpulseplot1 (x,y) ...
{  
    {x0,y0}=makeimpulse(x,log(y)/log(10));
    plot2d(x0,y0,>bar,grid=0);
    h=holding(1);
    frame();
    xgrid(ticks(x));
    p=plot();
    for i=-10 to 10;
        if i<=p[4] and i>=p[3] then
            ygrid(i,yt="10^"+i);
        endif;
    end;
    holding(h);
endfunction
```

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

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

Mari kita animasikan kurva 2D 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 muncul di jendela grafis. Jika tidak, penggambaran ulang terjadi dalam interval waktu yang jarang.

```
>function animliss (n,m) ...
t=linspace(0,2pi,500);
f=0;
c=framecolor(0);
l=lineWidth(2);
setplot(-1,1,-1,1);
repeat
    clg;
    plot(sin(n*t),cos(m*t+f));
    wait(0);
    if testkey() then break; endif;
```

```

f=f+0.02;
end;
framecolor(c);
linewidth(l);
endfunction

```

Tekan tombol apa saja untuk menghentikan animasi ini.

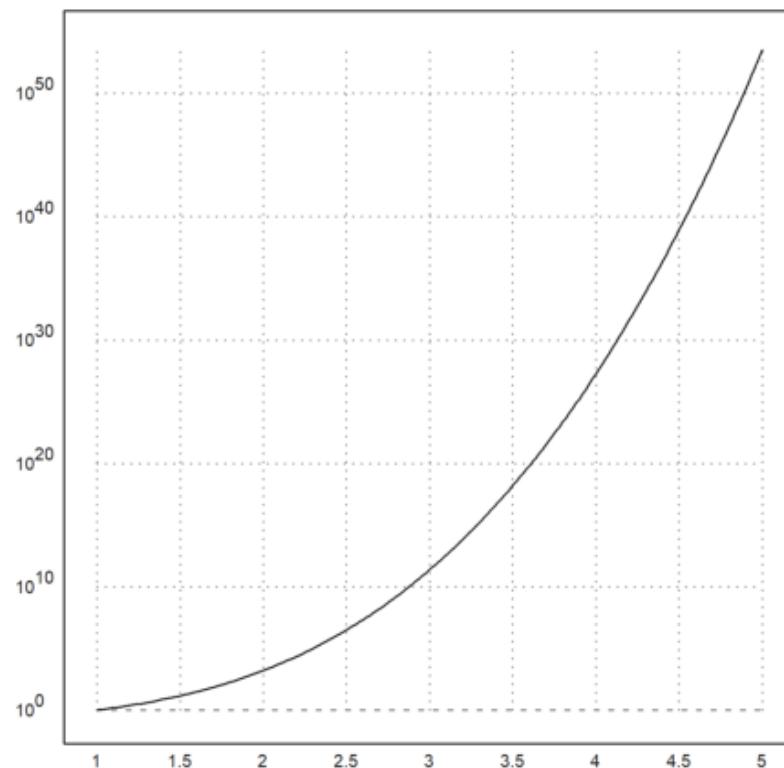
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Grafik Logaritmik

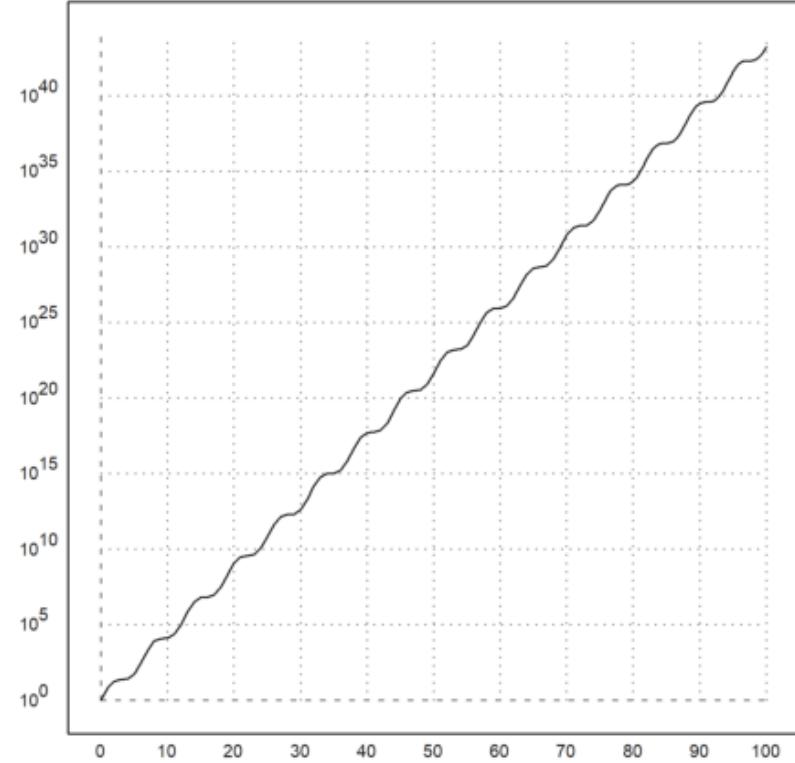
EMT menggunakan parameter logplot untuk skala logaritmik. Plot logaritmik dapat dibuat dengan menggunakan skala logaritmik pada sumbu y dengan logplot=1, atau menggunakan skala logaritmik pada sumbu x dan y dengan logplot=2, atau hanya pada sumbu x dengan logplot=3.

- logplot=1: skala logaritmik pada sumbu y
- logplot=2: skala logaritmik pada sumbu x dan y
- logplot=3: skala logaritmik pada sumbu x

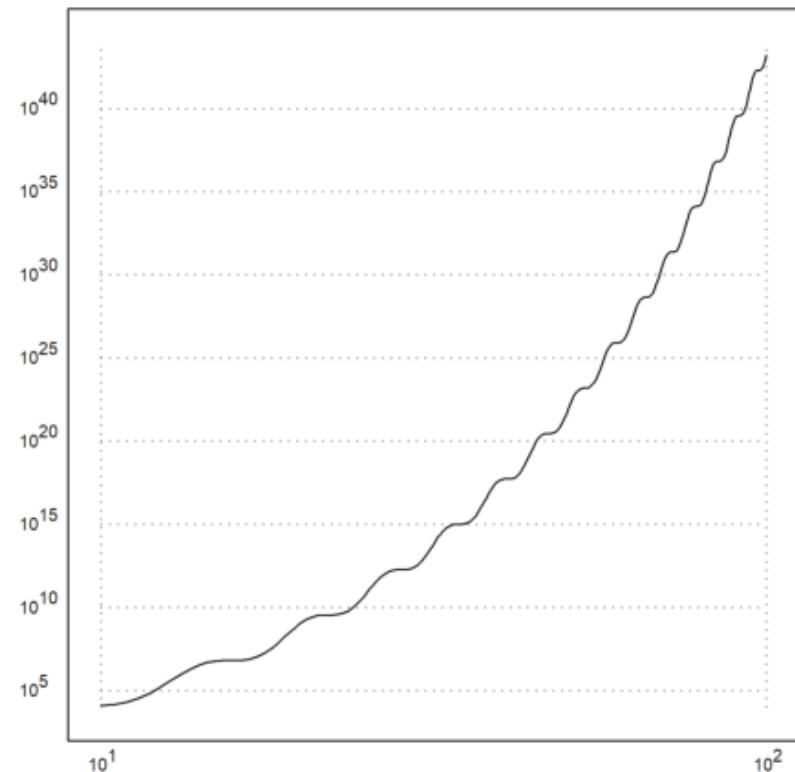
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



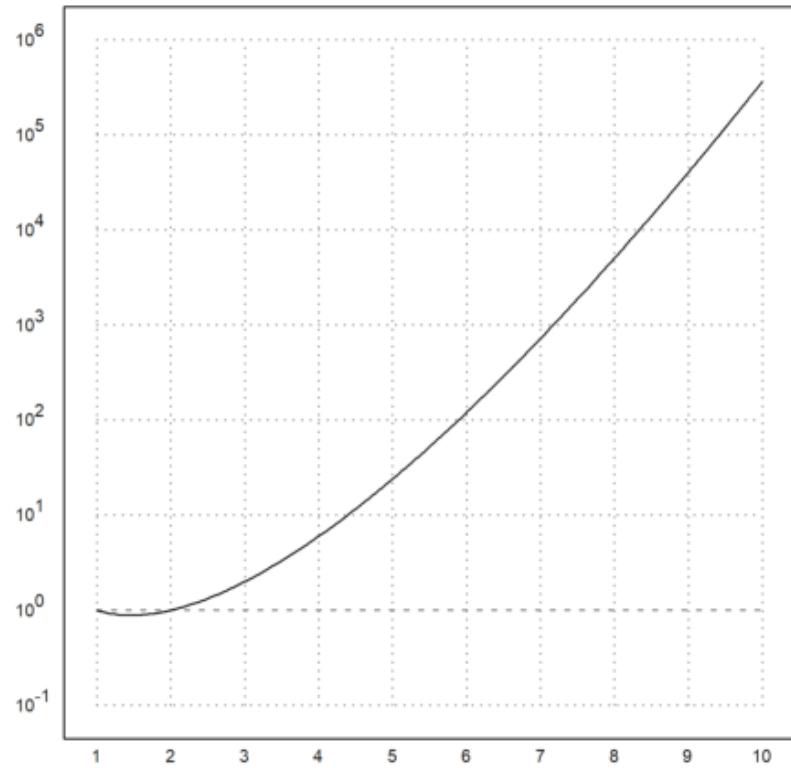
```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



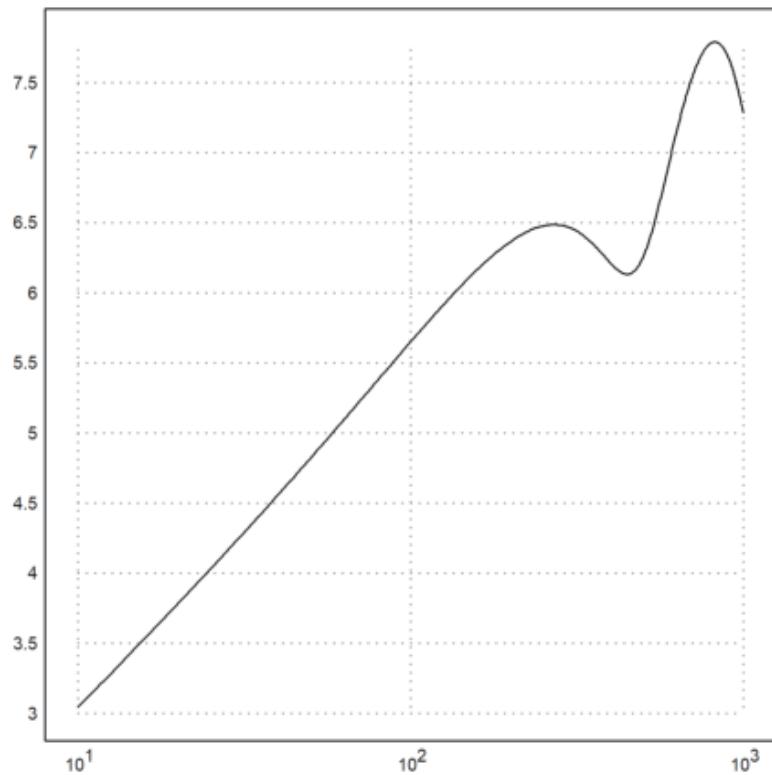
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```



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

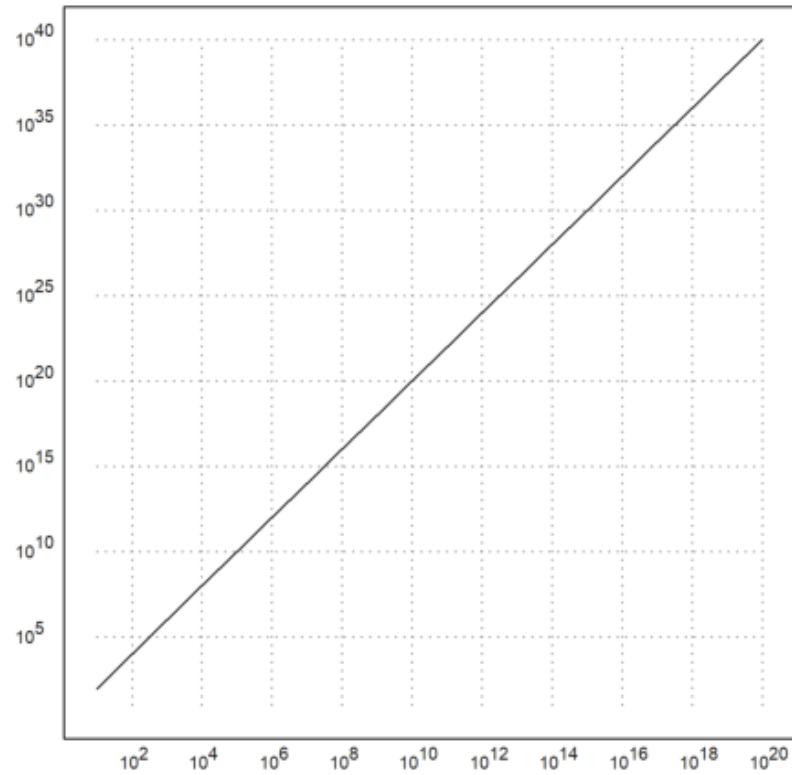


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Ini juga bekerja pada grafik data.

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



Latihan Soal

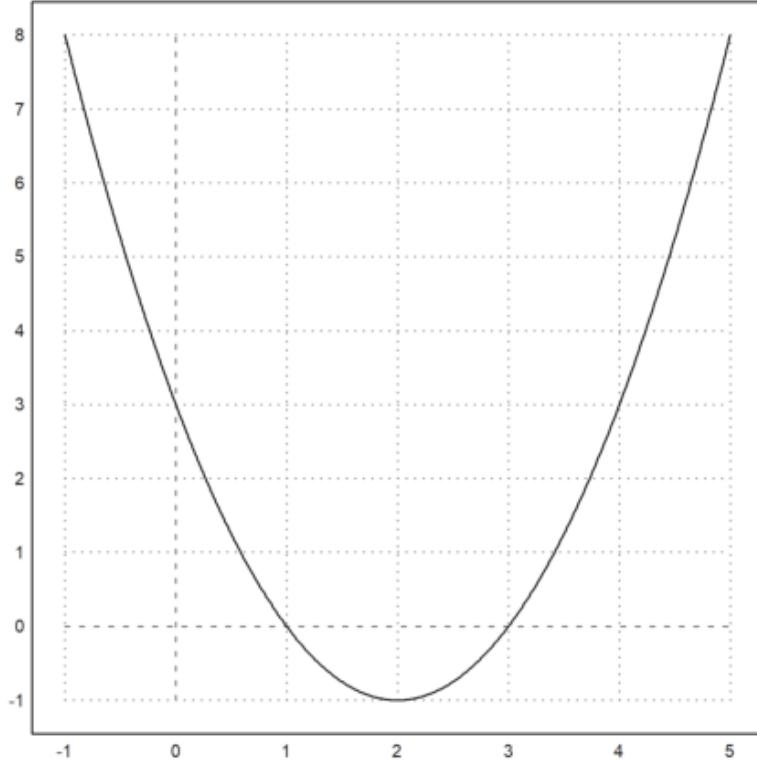
Nama : Muhammad Lutfi Ramadhan
Kelas : Matematika B 2023
NIM : 23030630021

1. Gambarkan grafik fungsi berikut!

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

Penyelesaian:

```
>plot2d("x^2-4*x+3",-1,5):
```

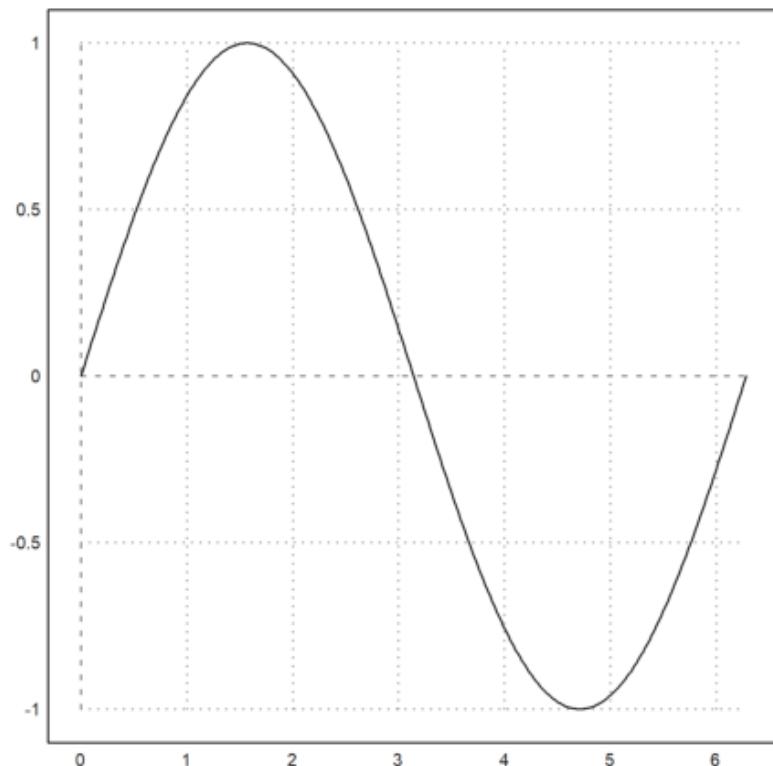


2. Buatkan grafik fungsi berikut pada interval 0 sampai 2π !

$$f(x) = \sin(x)$$

Penyelesaian:

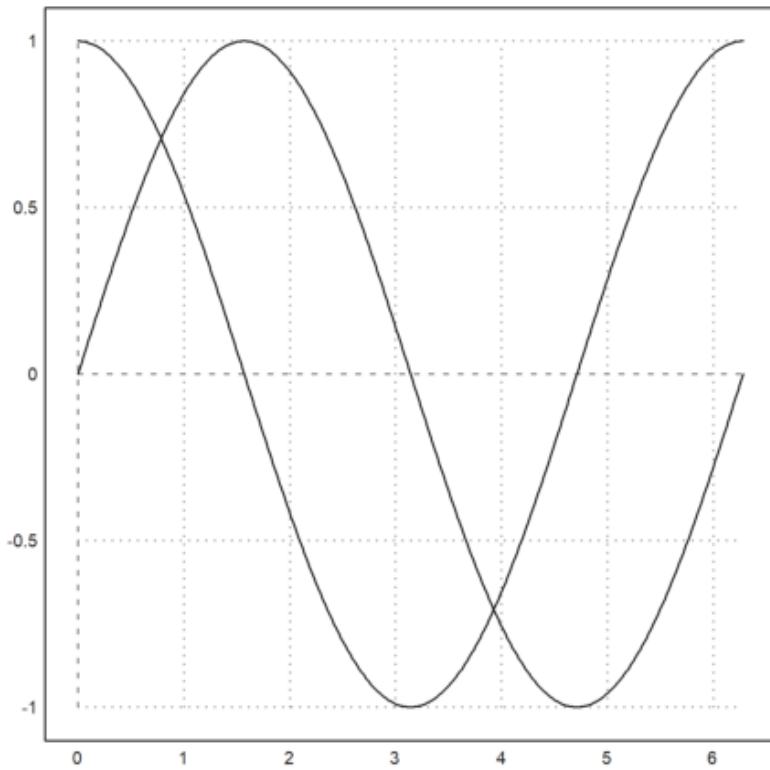
```
>plot2d("sin(x)", 0, 2*pi);
```



3. Gambarkan grafik fungsi $f(x)=\sin(x)$ dan $g(x)=\cos(x)$ secara bersamaan pada interval 0 sampai 2π !

Penyelesaian

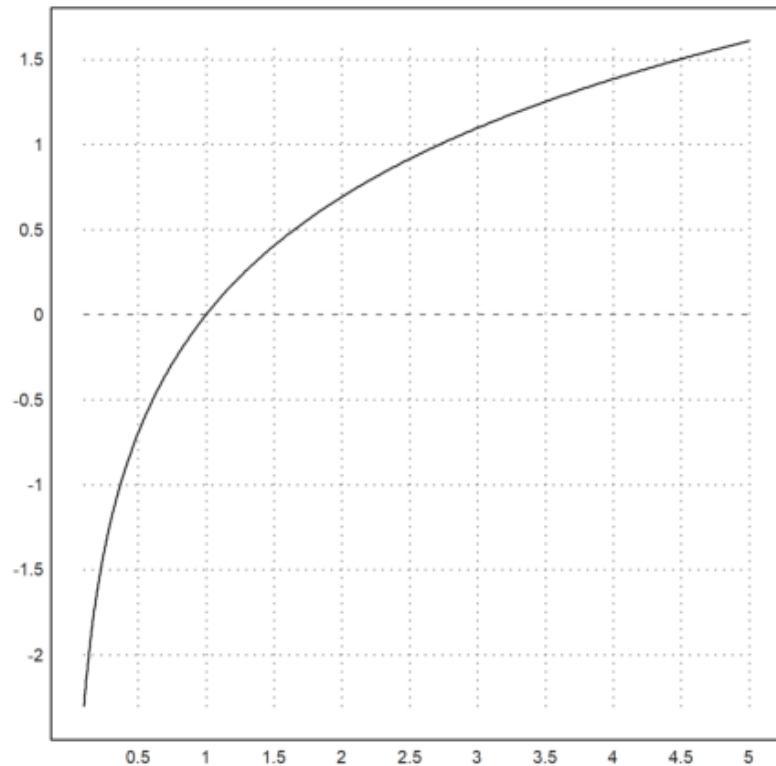
```
>plot2d(["sin(x)", "cos(x)", 0, 2*pi):
```



4. Gambarkan grafik fungsi $f(x)=\log(x)$ pada interval $[0.1,5]$!

Penyelesaian:

```
>plot2d("log(x)", 0.1, 5):
```

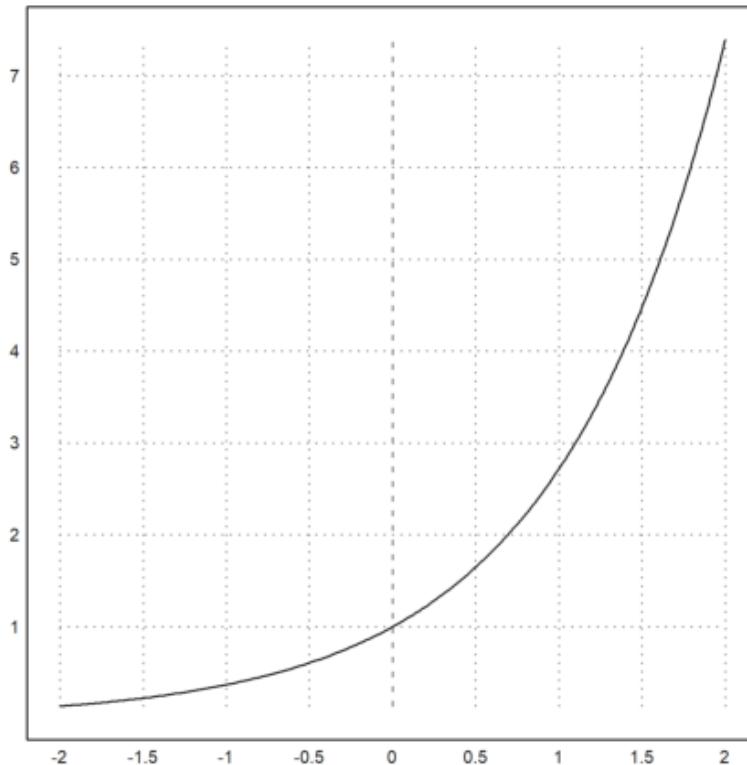


5. Gambarkan fungsi eksponen berikut pada interval $[-2,2]$!

$$f(x) = e^x$$

Penyelesaian:

```
>plot2d("exp(x)", -2, 2):
```



```
>
```

Rujukan Lengkap Fungsi plot2d()

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

Multipurpose plot function for plots in the plane (2D plots). This function can do plots of functions of one variables, data plots, curves in the plane, bar plots, grids of complex numbers, and implicit plots of functions of two variables.

Parameters

x,y : equations, functions or data vectors
a,b,c,d : Plot area (default a=-2,b=2)
r : if r is set, then a=cx-r, b=cx+r, c=cy-r, d=cy+r
r can be a vector [rx,ry] or a vector [rx1,rx2,ry1,ry2].
xmin,xmax : range of the parameter for curves
auto : Determine y-range automatically (default)
square : if true, try to keep square x-y-ranges
n : number of intervals (default is adaptive)
grid : 0 = no grid and labels,
1 = axis only,
2 = normal grid (see below for the number of grid lines)
3 = inside axis
4 = no grid
5 = full grid including margin
6 = ticks at the frame
7 = axis only
8 = axis only, sub-ticks
frame : 0 = no frame

framecolor: color of the frame and the grid
 margin : number between 0 and 0.4 for the margin around the plot
 color : Color of curves. If this is a vector of colors,
 it will be used for each row of a matrix of plots. In the case of
 point plots, it should be a column vector. If a row vector or a
 full matrix of colors is used for point plots, it will be used for
 each data point.
 thickness : line thickness for curves
 This value can be smaller than 1 for very thin lines.
 style : Plot style for lines, markers, and fills.
 For points use
 "[]", "<>", ".", "..", "...",
 "*"*, "+", "|", "-", "o"
 "[]#", "<>#", "o#" (filled shapes)
 "[]w", "<>w", "ow" (non-transparent)
 For lines use
 "-", "--", "-.", ".-", "-.-", "->"
 For filled polygons or bar plots use
 "#", "#O", "O", "/", "\", "\\", "\\\\",
 "+", "!", "-", "t"
 points : plot single points instead of line segments
 addpoints : if true, plots line segments and points
 add : add the plot to the existing plot
 user : enable user interaction for functions
 delta : step size for user interaction
 bar : bar plot (x are the interval bounds, y the interval values)
 histogram : plots the frequencies of x in n subintervals
 distribution=n : plots the distribution of x with n subintervals
 even : use inter values for automatic histograms.
 steps : plots the function as a step function (steps=1,2)
 adaptive : use adaptive plots (n is the minimal number of steps)
 level : plot level lines of an implicit function of two variables
 outline : draws boundary of level ranges.

If the level value is a 2xn matrix, ranges of levels will be drawn in the color using the given fill style. If outline is true, it will be drawn in the contour color. Using this feature, regions of f(x,y) between limits can be marked.

hue : add hue color to the level plot to indicate the function
 value
 contour : Use level plot with automatic levels
 nc : number of automatic level lines
 title : plot title (default "")
 xl, yl : labels for the x- and y-axis
 smaller : if >0, there will be more space to the left for labels.
 vertical :
 Turns vertical labels on or off. This changes the global variable
 verticallabels locally for one plot. The value 1 sets only vertical
 text, the value 2 uses vertical numerical labels on the y axis.
 filled : fill the plot of a curve
 fillcolor : fill color for bar and filled curves
 outline : boundary for filled polygons
 logplot : set logarithmic plots
 1 = logplot in y,
 2 = logplot in xy,
 3 = logplot in x
 own :
 A string, which points to an own plot routine. With >user, you get
 the same user interaction as in plot2d. The range will be set
 before each call to your function.
 maps : map expressions (0 is faster), functions are always mapped.
 contourcolor : color of contour lines
 contourwidth : width of contour lines
 clipping : toggles the clipping (default is true)
 title :
 This can be used to describe the plot. The title will appear above
 the plot. Moreover, a label for the x and y axis can be added with
 xl="string" or yl="string". Other labels can be added with the
 functions label() or labelbox(). The title can be a unicode
 string or an image of a Latex formula.
 cgrid :
 Determines the number of grid lines for plots of complex grids.

Should be a divisor of the the matrix size minus 1 (number of subintervals). cgrid can be a vector [cx,cy].

Overview

The function can plot

- expressions, call collections or functions of one variable,
- parametric curves,
- x data against y data,
- implicit functions,
- bar plots,
- complex grids,
- polygons.

If a function or expression for xv is given, plot2d() will compute values in the given range using the function or expression. The expression must be an expression in the variable x. The range must be defined in the parameters a and b unless the default range [-2,2] should be used. The y-range will be computed automatically, unless c and d are specified, or a radius r, which yields the range [-r,r] for x and y. For plots of functions, plot2d will use an adaptive evaluation of the function by default. To speed up the plot for complicated functions, switch this off with <adaptive, and optionally decrease the number of intervals n. Moreover, plot2d() will by default use mapping. I.e., it will compute the plot element for element. If your expression or your functions can handle a vector x, you can switch that off with <maps for faster evaluation.

Note that adaptive plots are always computed element for element. If functions or expressions for both xv and for yv are specified, plot2d() will compute a curve with the xv values as x-coordinates and the yv values as y-coordinates. In this case, a range should be defined for the parameter using xmin, xmax. Expressions contained in strings must always be expressions in the parameter variable x.

Tugas Individu Pekan7-8_Muhammad Lutfi Ramadhan_23030630021

Nama : Muhammad Lutfi Ramadhan
Kelas : Matematika B 2023
NIM : 23030630021

Menggambar Plot 3D dengan EMT

Ini adalah pengenalan untuk plot 3D di Euler. Kita membutuhkan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

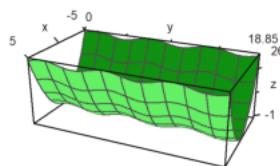
Euler menggambar fungsi seperti itu menggunakan algoritma pengurutan untuk menyembunyikan bagian yang ada di latar belakang. Secara umum, Euler menggunakan proyeksi sentral. Defaultnya adalah dari kuadran positif x-y menuju ke asal $x=y=z=0$, tetapi sudut=0° melihat dari arah sumbu y. Sudut pandang dan ketinggian dapat diubah.

Euler dapat memplot:

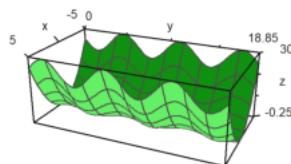
- permukaan dengan bayangan dan garis kontur atau rentang kontur,
- awan titik,
- kurva parametrik,
- permukaan implisit.

Plot 3D dari sebuah fungsi menggunakan plot3d. Cara termudah adalah memplot ekspresi dalam x dan y. Parameter r mengatur rentang plot di sekitar (0,0).

```
>aspect(1.5); plot3d("x^2+sin(y)", -5, 5, 0, 6*pi);
```



```
>plot3d("x^2+x*sin(y)", -5, 5, 0, 6*pi);
```



Silakan lakukan modifikasi agar gambar "talang bergelombang" tersebut tidak lurus melainkan melengkung/melingkar, baik melingkar secara mendatar maupun melingkar turun/naik (seperti papan peluncur pada kolam renang). Temukan rumusnya.

```
>aspect(1.5); plot3d("x^2+sin(y)", r=pi);
```

```
Closing bracket missing in function call!
Error in:
aspect(1.5); plot3d("x^2+sin(y)", r=pi: ...
```

Fungsi Dua Variabel

Untuk grafik sebuah fungsi, gunakan:

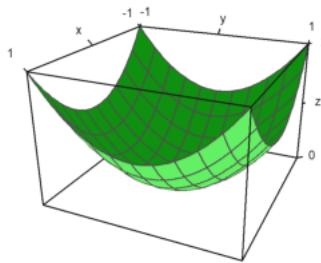
- ekspresi sederhana dalam x dan y,
- nama fungsi dari dua variabel,
- atau matriks data.

Defaultnya adalah grid kawat berisi dengan warna berbeda di kedua sisinya. Perhatikan bahwa jumlah interval grid default adalah 10, tetapi plot menggunakan jumlah 40x40 persegi panjang untuk membangun permukaan. Ini dapat diubah.

- n=40, n=[40,40]: jumlah garis grid di setiap arah
- grid=10, grid=[10,10]: jumlah garis grid di setiap arah

Kita menggunakan default n=40 dan grid=10.

```
>plot3d("x^2+y^2"):
```

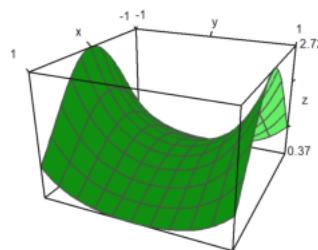


Interaksi pengguna dimungkinkan dengan parameter >user. Pengguna dapat menekan tombol berikut:

- kiri, kanan, atas, bawah: mengubah sudut pandang
- +, -: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l: beralih sumber cahaya
- spasi: mengatur ulang ke default
- enter: mengakhiri interaksi

```
>plot3d("exp(-x^2+y^2)",>user, ...
         title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



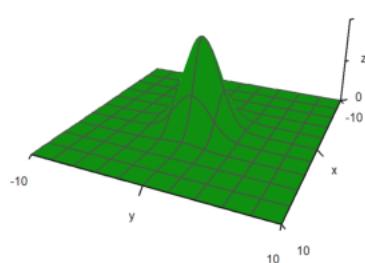
Rentang plot untuk fungsi dapat ditentukan dengan:

- a,b: rentang x
- c,d: rentang y
- r: bujur sangkar simetris di sekitar (0,0)
- n: jumlah subinterval untuk plot

Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik:

fscale: menskalakan nilai fungsi (default adalah <fscale>).
cale: angka atau vektor 1x2 untuk menskalakan ke arah x dan y.
rame: tipe bingkai (default 1).

```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):
```



Pandangan bisa diubah dalam banyak cara:

- distance: jarak pandang ke plot,
- zoom: nilai pembesaran,
- angle: sudut terhadap sumbu y negatif dalam radian,
- height: ketinggian pandangan dalam radian.

ilai default dapat diperiksa atau diubah dengan fungsi view(). Fungsi ini mengembalikan parameter dalam urutan di atas.

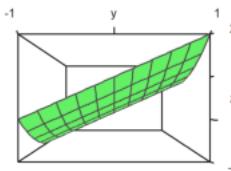
```
>view
```

```
[5, 2.6, 2, 0.4]
```

Jarak yang lebih dekat memerlukan pembesaran yang lebih sedikit. Efeknya lebih mirip dengan lensa sudut lebar.

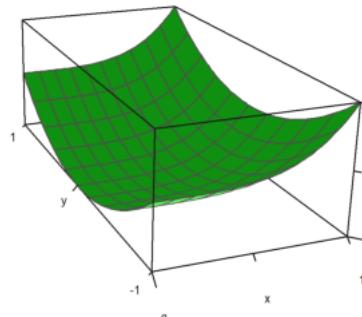
Dalam contoh berikut, angle=0 dan height=0 melihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```



Plot selalu melihat ke pusat kubus plot. Kamu dapat memindahkan pusat dengan parameter center.

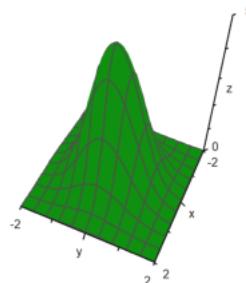
```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
center=[0.4,0,0],zoom=5):
```



Plot diskalakan agar sesuai dengan kubus unit untuk tampilan. Jadi tidak perlu mengubah jarak atau zoom tergantung pada ukuran plot. Namun, label tetap merujuk pada ukuran yang sebenarnya.

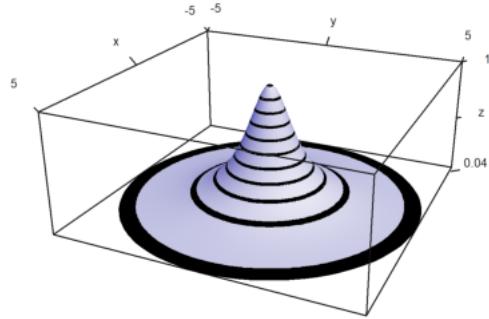
Jika kamu mematikan ini dengan scale=false, kamu harus memastikan bahwa plot tetap sesuai dengan jendela pemetaan, dengan mengubah jarak pandang atau zoom, serta menggerakkan pusat.

```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
center=[0,0,-2],frame=3):
```

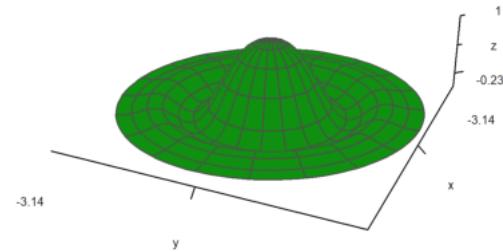


Plot polar juga tersedia. Parameter polar=true menggambar plot polar. Fungsi tetap harus berupa fungsi dari x dan y. Parameter 'fscale' menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi akan diskalakan agar sesuai dengan sebuah kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
fscale=2,>hue,n=100,zoom=4,>contour,color=blue):
```



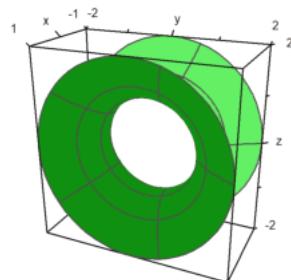
```
>function f(r) := exp(-r/2)*cos(r); ...
plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```



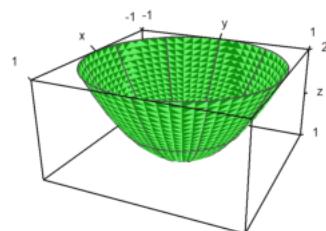
Parameter `rotate` memutar sebuah fungsi dalam `x` di sekitar sumbu `x`.

- `rotate=1`: Menggunakan sumbu `x`
- `rotate=2`: Menggunakan sumbu `z`

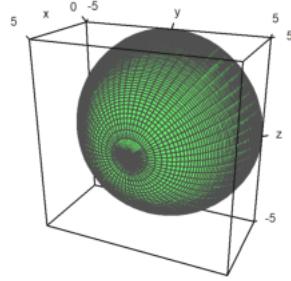
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



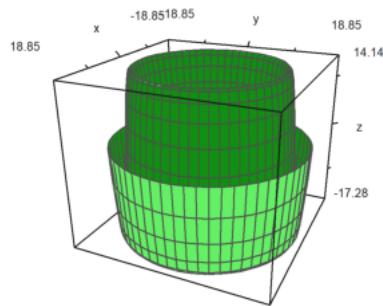
```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```



```
>plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1):
```

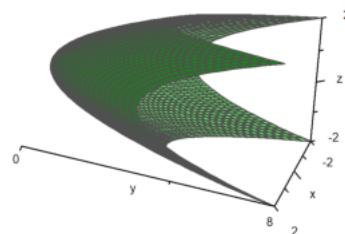


```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```



Ini adalah fungsi dengan 3 variabel.

```
>plot3d("x","x^2+y^2","y",r=2,zoom=3.5,frame=3):
```



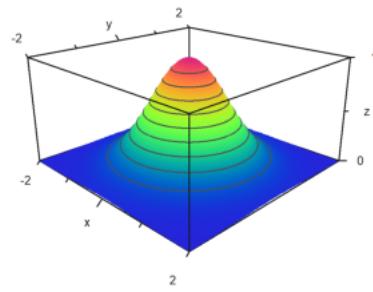
Plot Pola Kontur

Untuk plot, Euler menambahkan garis grid. Sebagai gantinya, bisa menggunakan garis level dan satu warna atau spektrum berwarna. Euler bisa menggambar ketinggian fungsi pada sebuah plot dengan bayangan. Dalam semua plot 3D, Euler bisa menghasilkan anaglyph merah/sian.

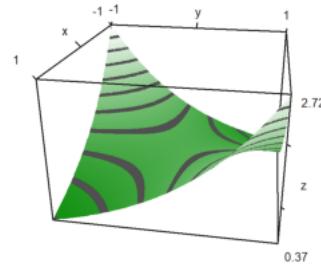
>hue: Mengaktifkan bayangan cahaya alih-alih kawat,
>contour: Memplot garis kontur otomatis pada plot,
level=... (atau levels): Sebuah vektor nilai untuk garis kontur.

Default-nya adalah level="auto", yang secara otomatis menghitung beberapa garis level. Seperti yang kamu lihat dalam plot, level-level ini sebenarnya adalah rentang level. Gaya default bisa diubah. Untuk plot kontur berikut, kita menggunakan grid yang lebih halus yaitu 100x100 titik, menskalakan fungsi dan plot, serta menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...
>contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):
```



```
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```



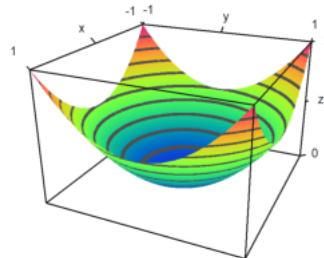
Bayangan default menggunakan warna abu-abu, tetapi skema warna spektral juga tersedia.

>spectral: Menggunakan skema spektral default.

color=...: Menggunakan warna khusus atau skema spektral.

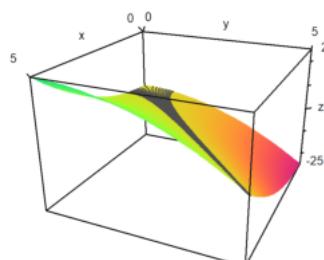
Untuk plot berikut, kita menggunakan skema spektral default dan meningkatkan jumlah titik untuk mendapatkan tampilan yang sangat halus.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```



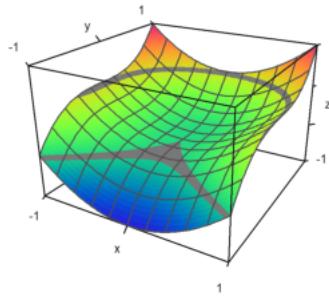
Ailih-alih garis level otomatis, kita juga bisa menetapkan nilai-nilai garis level. Ini akan menghasilkan garis level tipis alih-alih rentang level.

```
>plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=redgreen):
```



Dalam plot berikut, kita menggunakan dua pita level yang sangat lebar dari -0.1 hingga 1, dan dari 0.9 hingga 1. Ini dimasukkan sebagai matriks dengan batas level sebagai kolom. Kita juga menambahkan grid dengan 10 interval di setiap arah.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...
>spectral,angle=30°,grid=10,contourcolor=gray):
```

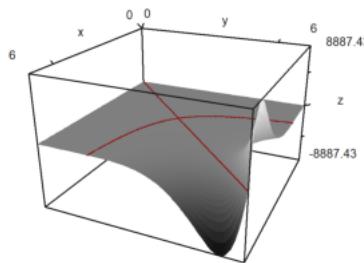


Dalam plot berikut, kita memplot set di mana:

$$f(x,y) = x^y - y^x = 0$$

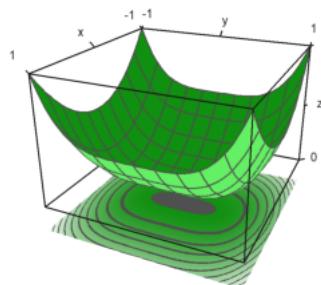
Kita menggunakan satu garis tipis untuk garis level.

```
>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 dari plot dapat ditentukan.

```
>plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2):
```



Berikut adalah beberapa gaya lainnya. Kita selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan grid.

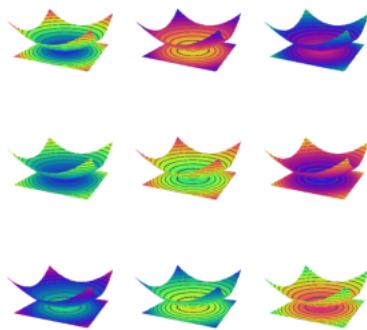
```
>figure(2,2); ...
expr="y^3-x^2"; ...
figure(1); ...
plot3d(expr,<frame,>cp,cpcolor=spectral); ...
figure(2); ...
plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
figure(3); ...
plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
figure(4); ...
plot3d(expr,<frame,>hue,grid=10,>cp,cpcolor=gray); ...
figure(0):
```



Ada beberapa skema spektral lainnya, bernomor dari 1 hingga 9. Tetapi kamu juga dapat menggunakan color=value, di mana value adalah:

- spectral: untuk rentang dari biru ke merah,
- white: untuk rentang yang lebih lembut,
- 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);
```



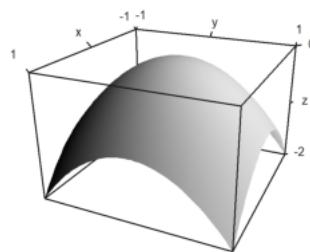
Sumber cahaya dapat diubah dengan menekan l dan menggunakan tombol panah selama interaksi pengguna. Parameter sumber cahaya dapat diatur dengan:

- light: arah cahaya,
- amb: cahaya ambient antara 0 dan 1.

Perhatikan bahwa program ini tidak membedakan antara sisi-sisi plot. Tidak ada bayangan. Untuk ini, kamu memerlukan Povray.

```
>plot3d("-x^2-y^2", ...
hue=true,light=[0,1,1],amb=0,user=true, ...
title="Press l and cursor keys (return to exit)":
```

Press l and cursor keys (return to exit)



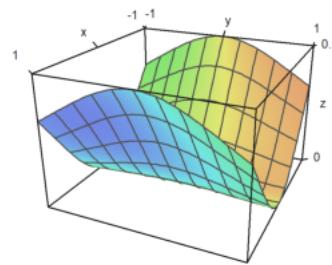
Parameter color mengubah warna permukaan. Warna garis level juga dapat diubah.

```
>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,d1=0.01):
```



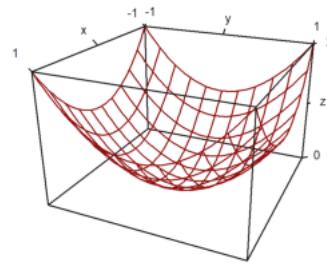
Warna 0 memberikan efek pelangi khusus.

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



Permukaan 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 ini menunjukkan set nol dari sebuah fungsi dalam tiga variabel.

Solusi dari:

$$f(x, y, z) = 0$$

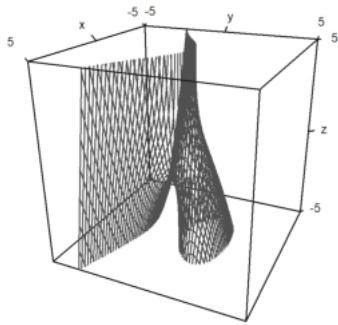
dapat divisualisasikan dalam potongan-potongan yang sejajar dengan bidang x-y, x-z, dan y-z.

- implicit=1: potongan sejajar dengan bidang y-z,
- implicit=2: potongan sejajar dengan bidang x-z,
- implicit=4: potongan sejajar dengan bidang x-y.

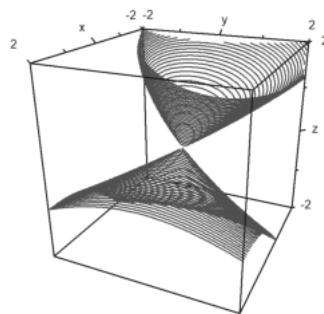
Tambahkan nilai-nilai ini jika kamu suka. Dalam contoh berikut, kita memplot:

$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```



```
>c=1; d=1;
>plot3d("((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)-d", r=2, frame=true;
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5);
```



Plot Data 3D

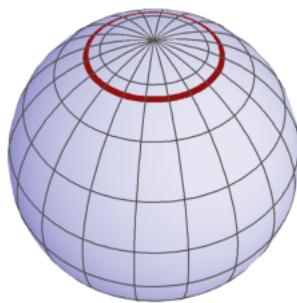
Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, kamu perlu menyediakan matriks nilai x, y, dan z, atau tiga fungsi atau ekspresi $fx(x,y)$, $fy(x,y)$, $fz(x,y)$.

$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

Karena x, y, z adalah matriks, kita mengasumsikan bahwa (t, s) berjalan melalui grid bujur sangkar. Hasilnya, kamu bisa memplot gambar persegi panjang di ruang angkasa. Kamu bisa menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

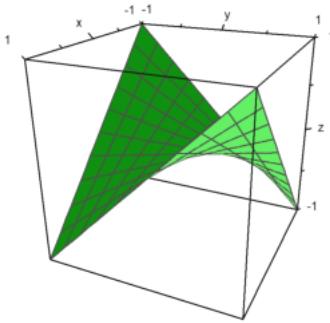
Dalam contoh berikut, kita menggunakan vektor nilai t dan vektor kolom nilai s untuk memparametrikkan permukaan bola. Dalam gambar tersebut, kita dapat menandai wilayah, dalam kasus ini wilayah kutub.

```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)';
x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s);
plot3d(x,y,z,>hue, ...
color=blue,<frame,grid=[10,20], ...
values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
scale=1.4,height=50°);
```



Berikut adalah contoh, yang merupakan grafik dari sebuah fungsi.

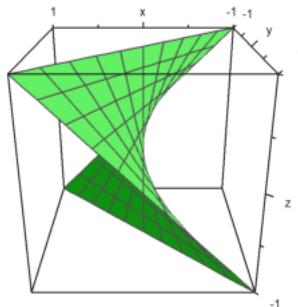
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10);
```



Namun, kita bisa membuat berbagai jenis permukaan. Berikut adalah permukaan yang sama sebagai fungsi

$$x = yz$$

```
>plot3d(t*s,t,s,angle=180°,grid=10):
```



Dengan lebih banyak usaha, kita bisa menghasilkan banyak permukaan.

Dalam contoh berikut, kita membuat pandangan bayangan dari bola yang terdistorsi. Koordinat yang biasa untuk bola adalah:

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

dengan:

$$0 \leq t \leq 2\pi, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2}.$$

Kita mengubahnya dengan faktor:

$$d(t, s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

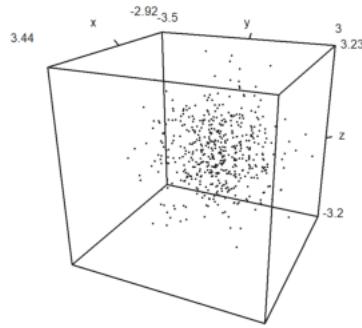
```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
d=1+0.2*(cos(4*t)+cos(8*s)); ...
plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
light=[1,0,1],frame=0,zoom=5):
```



Tentu saja, awan titik juga dimungkinkan. Untuk memplot data titik di ruang, kita memerlukan tiga vektor untuk koordinat titik-titik tersebut.

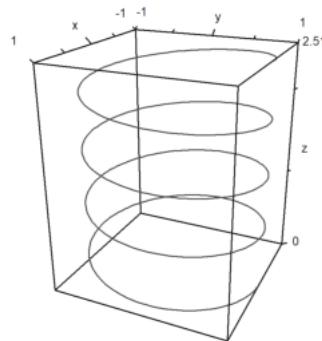
Gaya visualisasi adalah sama seperti di plot2d dengan points=true.

```
>n=500; ...
plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

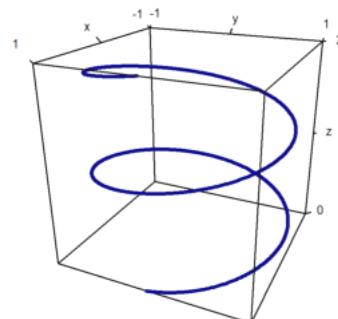


Dimungkinkan juga untuk memplot kurva dalam 3D. Dalam kasus ini, lebih mudah untuk menghitung titik-titik kurva sebelumnya. Untuk kurva dalam bidang, kita menggunakan urutan koordinat dan parameter `wire=true`.

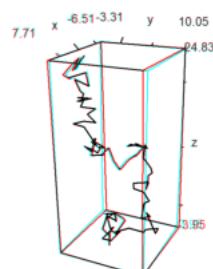
```
>t=linspace(0,8pi,500); ...
plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
linewidth=3, wirecolor=blue):
```

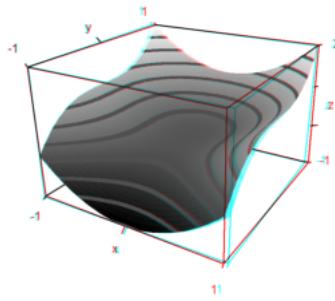


```
>X=cumsum(normal(3,100)); ...
plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



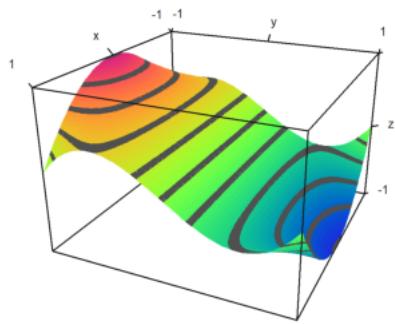
EMT juga dapat membuat plot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda memerlukan kacamata merah/sian.

```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```



Seringkali, skema warna spektral digunakan untuk plot. Ini menekankan ketinggian dari fungsi tersebut.

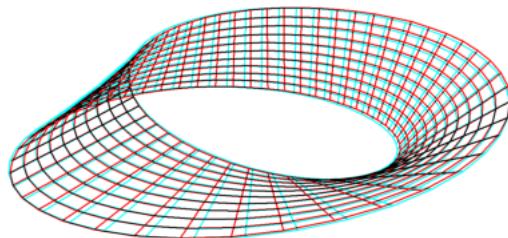
```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```



Euler juga dapat membuat plot permukaan parameterisasi, di mana parameter-parameter tersebut adalah nilai x, y, dan z dari gambar grid persegi panjang di ruang.

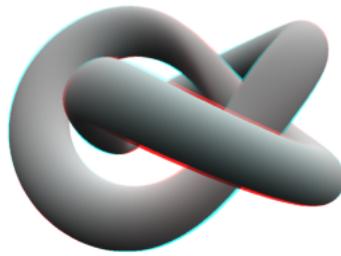
Untuk demo berikut, kita menetapkan parameter u dan v, dan menghasilkan koordinat ruang dari parameter-parameter ini.

```
>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 adalah contoh yang lebih rumit, yang terlihat megah dengan kacamata merah/sian.

```
>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

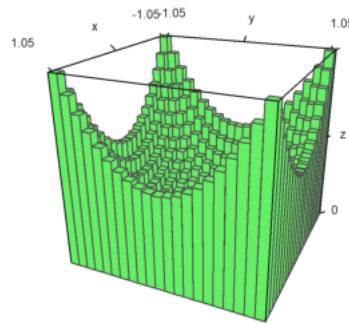
Plot batang juga dimungkinkan. Untuk ini, kita harus menyediakan:

- x: vektor baris dengan n+1 elemen,
- y: vektor kolom dengan n+1 elemen,
- z: matriks nxn nilai.

z bisa lebih besar, tetapi hanya nilai nxn yang akan digunakan. Dalam contoh ini, kita pertama-tama menghitung nilai-nilainya.

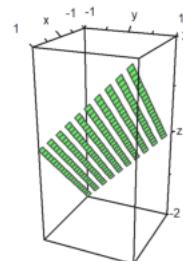
Lalu kita menyesuaikan x dan y, sehingga vektor-vektor tersebut terpusat pada nilai yang digunakan.

```
>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);
```



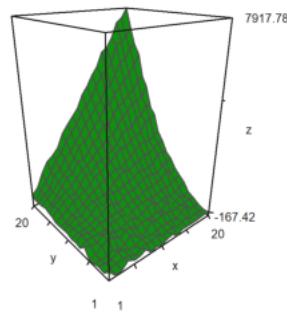
Dimungkinkan untuk membagi plot permukaan menjadi dua atau lebih bagian.

```
>x=-1:0.1:1; y=x'; z=x*y; d=zeros(size(x)); ...
plot3d(x,y,z,disconnect=2:2:20);
```

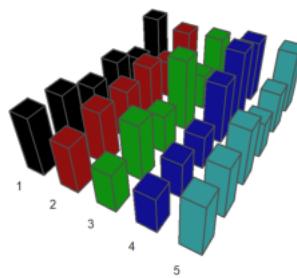


Jika kamu memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, kamu bisa menyesuaikan matriks tersebut menjadi [-1,1] dengan scale(M), atau menyesuaikan matriks dengan >zscale. Ini bisa dikombinasikan dengan faktor skala individu yang diterapkan secara tambahan.

```
>i=1:20; j=i'; ...
plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8);
```

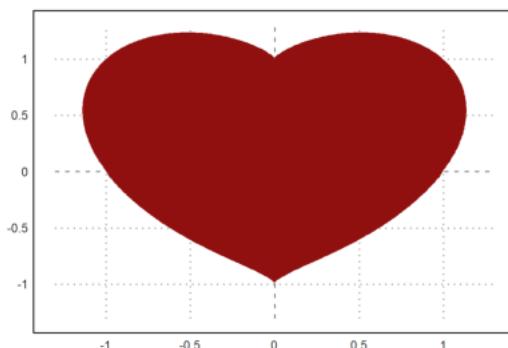


```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
columnsplot3d(v',scols=1:5,ccols=[1:5]):
```



Permukaan Benda Putar

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



```
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

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

Ekspresi ini menghasilkan bentuk hati tiga dimensi, yang kemudian kita putar di sekitar sumbu y. Ini adalah ekspresi yang menggambarkan hati:

$$f(x, y) = (x^2 + y^2 - 1)^3 - x^2 y^3.$$

Selanjutnya kita set:

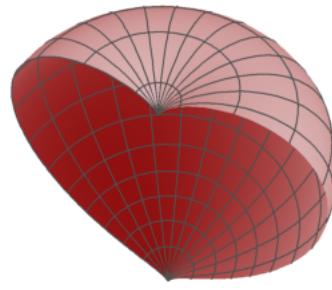
$$x = r \cos(a), \quad y = r \sin(a).$$

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2 \sin a) r^5}{16}$$

Ini memungkinkan kita untuk mendefinisikan fungsi numerik yang menyelesaikan r, jika a diberikan. Dengan fungsi ini, kita dapat memplot bentuk hati yang diputar sebagai permukaan parametrik.

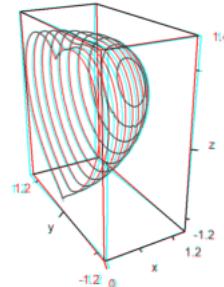
```
>function map f(a) := bisect("fr",0,2;a); ...
t=linspace(-pi/2,pi/2,100); r=f(t); ...
s=linspace(pi,2pi,100)'; ...
plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```



Berikut adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kita mendefinisikan fungsi yang menggambarkan objek tersebut.

```
>function f(x,y,z) ...
r=x^2+y^2;
return (r+z^2-1)^3-r*z^3;
endfunction

>plot3d("f(x,y,z)", ...
xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...
implicit=1,angle=-30°,zoom=2.5,n=[10,100,60],>anaglyph):
```



Plot 3D Khusus

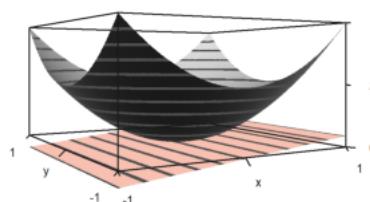
Fungsi plot3d memang berguna, tetapi tidak memuaskan semua kebutuhan. Selain dari rutinitas dasar, dimungkinkan untuk mendapatkan plot yang dibingkai dari objek apa pun yang kamu inginkan.

Meskipun Euler bukan program 3D, program ini dapat menggabungkan beberapa objek dasar. Kita mencoba memvisualisasikan paraboloid 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 bingkai-bingkai, dan mengatur pandangan.

```
>framedplot("myplot",[-1,1,-1,1,0,1],height=0,angle=-30°, ...
center=[0,0,-0.7],zoom=3):
```



Dengan cara yang sama, kamu dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d() mengatur jendela menjadi fullwindow() secara default, tetapi plotcontourplane() mengasumsikan itu.

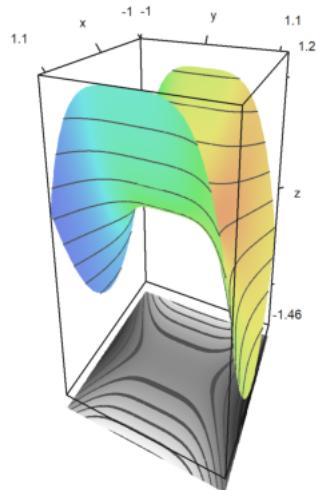
```

>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 bingkai untuk menghitung animasi terlebih dahulu.

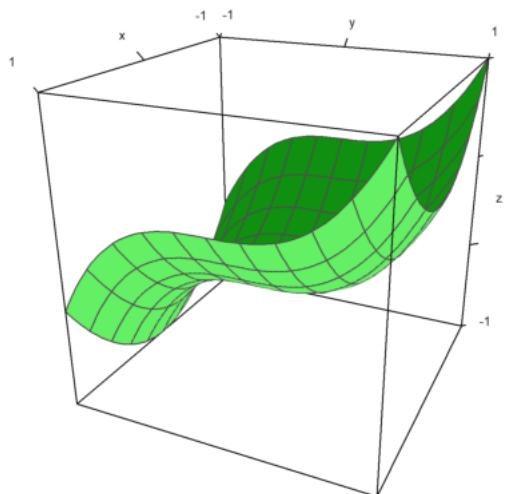
Salah satu fungsi yang menggunakan teknik ini adalah `rotate`. Fungsi ini dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi ini memanggil `adddpage()` untuk setiap plot baru. Akhirnya, animasi dari plot tersebut ditampilkan.

Silakan pelajari sumber dari `rotate` untuk melihat lebih detail.

```

>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. Kamu perlu menginstal Povray (32bit atau 64bit) dari <http://www.povray.org>, dan menambahkan sub-direktori "bin" dari Povray ke jalur lingkungan, atau mengatur variabel defaultpovray dengan jalur penuh yang menunjuk ke "pvengine.exe".

Antarmuka Povray dari Euler menghasilkan file-file Povray di direktori rumah pengguna, dan memanggil Povray untuk mengurai file-file tersebut. Nama file default adalah `current.pov`, dan direktori default adalah `eulerhome()`, biasanya `c:\Users\Username\Euler`. Povray menghasilkan file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan file-file ini, gunakan `povclear()`.

Fungsi pov3d serupa dengan plot3d. Fungsi ini dapat menghasilkan grafik sebuah fungsi $f(x,y)$, atau permukaan dengan koordinat X, Y, Z dalam matriks, termasuk garis level opsional. Fungsi ini secara otomatis memulai raytracer, dan memuat pemandangan ke dalam notebook Euler.

Selain pov3d(), terdapat banyak fungsi yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string yang berisi kode Povray untuk objek-objek tersebut. Untuk menggunakan fungsi-fungsi ini, mulai file Povray dengan povstart(). Kemudian gunakan writeln(..) untuk menulis objek ke file pemandangan. Terakhir, akhir file tersebut dengan povend(). Secara default, raytracer akan memulai, dan PNG akan dimasukkan ke dalam notebook Euler.

Fungsi objek memiliki parameter yang disebut look, yang memerlukan 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, dan Phong Shading, dll.

erlu diperhatikan bahwa alam semesta Povray memiliki sistem koordinat yang berbeda. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi kamu bisa terus berpikir dalam sistem koordinat Euler dengan z yang mengarah vertikal ke atas, serta sumbu x, y, z mengikuti aturan tangan kanan. Kamu perlu memuat file povray.

```
>load povray;
```

Pastikan, direktori bin Povray ada di jalur. Jika tidak, edit variabel berikut sehingga berisi jalur ke executable Povray.

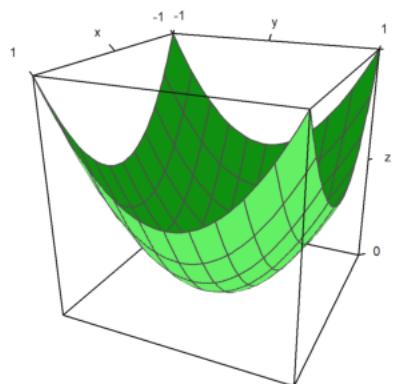
```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Untuk kesan pertama, kita memplot sebuah fungsi sederhana. Perintah berikut menghasilkan sebuah file povray di direktori pengguna, dan menjalankan Povray untuk melakukan pelacakan sinar pada file ini.

Jika kamu memulai perintah berikut, GUI Povray akan terbuka, menjalankan file tersebut, dan menutup secara otomatis. Karena alasan keamanan, kamu akan diminta apakah ingin mengizinkan file exe untuk dijalankan. Kamu bisa menekan cancel untuk menghentikan pertanyaan lebih lanjut. Mungkin kamu harus menekan OK di jendela Povray untuk mengakui dialog start-up Povray.

```
>plot3d("x^2+y^2",zoom=2) :
```



```
>pov3d("x^2+y^2",zoom=3) ;
```

Kita dapat membuat fungsi transparan dan menambahkan hasil akhir lainnya. Kita juga dapat menambahkan garis level ke plot fungsi.

Kadang-kadang, perlu untuk mencegah penskalaan fungsi, dan menskalakannya secara manual.

Kita memplot set titik-titik dalam bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=2, ...
angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=10°,n=50, ...
<fscale,zoom=3.8) ;
```

Plot dengan Koordinat

Alih-alih fungsi, kita bisa memplot dengan koordinat. Seperti pada plot3d, kita memerlukan tiga matriks untuk mendefinisikan objek. Dalam contoh ini, kita memutar sebuah fungsi di sekitar sumbu z.

```
>load povray;
>function f(x) := x^3-x+1; ...
x=-1:0.01:1; t=linspace(0,2pi,50)'; ...
Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light=[10,5,15]):
```

```
Command was not allowed!
exec:
    return _exec(program,param,dir,print,hidden,wait);
```

```

povray:
  exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
pov3d:
  if povray then povray(currentfile,w,h,w/h); endif;

```

Dalam contoh berikut, kita memplot gelombang teredam. Kita menghasilkan gelombang dengan bahasa matriks Euler. Kita juga menunjukkan bagaimana sebuah objek tambahan bisa ditambahkan ke pemandangan pov3d. Untuk menghasilkan objek, lihat contoh-contoh berikut ini. Perhatikan bahwa plot3d menskalakan plot, sehingga sesuai dengan kubus unit.

```

>r=linspace(0,1,80); phi=linspace(0,2pi,80)';
x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3; ...
pov3d(x,y,z,zoom=6,axis=0,height=30°,add=povsphere([0.5,0,0.25],0.15,povlook(red)), ...
w=500,h=300);

```

```

Function povlook not found.
Try list ... to find functions!
Error in:
... =30°,add=povsphere([0.5,0,0.25],0.15,povlook(red)), w=500,h= ...

```

Dengan metode shading canggih dari Povray, sangat sedikit titik yang bisa menghasilkan permukaan yang sangat halus. Hanya di batas dan bayangan trik ini mungkin menjadi jelas. Untuk ini, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>z &= x^2*y^3
```

$$\begin{matrix} 2 & 3 \\ x & y \end{matrix}$$

Persamaan permukaan adalah $[x,y,Z]$. Kita menghitung dua turunan terhadap x dan y dari ini dan mengambil hasil silang sebagai normal.

```
>dx &= diff([x,y,z],x); dy &= diff([x,y,z],y);
```

Kita mendefinisikan normal sebagai hasil silang dari kedua turunan tersebut, dan mendefinisikan fungsi koordinat.

```
>n &= crossproduct(dx,dy); NX &= n[1]; NY &= n[2]; NZ &= n[3]; N,
```

$$[-2x^3y, -3x^2y^2, 1]$$

Kita hanya menggunakan 25 titik.

```
>x=-1:0.5:1; y=x'
```

$$\begin{matrix} -1 \\ -0.5 \\ 0 \\ 0.5 \\ 1 \end{matrix}$$

```
>pov3d(x,y,z(x,y),angle=10°, ...
xv=Nx(x,y),yv=Ny(x,y),zv=Nz(x,y),<shadow>:
```

```

Function pov3d not found.
Try list ... to find functions!
Error in:
... =10°, xv=Nx(x,y),yv=Ny(x,y),zv=Nz(x,y),<shadow>: ...

```

Persamaan simpul Trefoil dikerjakan oleh A. Busser dalam Povray. Ada versi yang lebih baik dari ini di contoh-contohnya. Simak di:

Trefoil Knot

Untuk hasil yang baik dengan tidak terlalu banyak titik, kita menambahkan vektor normal di sini. Kita menggunakan Maxima untuk menghitung normal bagi kita. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolik:

```

>x &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
z &= sin(x)+2*cos(3*y);

```

Kemudian vektor turunan terhadap x dan y .

```
>dx &= diff([x,y,z],x); dy &= diff([x,y,z],y);
```

Sekarang normal, yang merupakan hasil silang dari kedua turunan tersebut.

```
>dn &= crossproduct(dx,dy);
```

Sekarang kita menghitung semuanya secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

Vektor normal adalah hasil evaluasi dari ekspresi simbolik $dn[i]$ untuk $i=1,2,3$. Sintaks untuk ini adalah &"expression"(parameters). Ini merupakan alternatif dari metode di contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),>anaglyph,axis=0,zoom=5,w=450,h=350, ...
<shadow,look=povlook(blue), ...
xv=&"dn[1]"(x,y), yv=&"dn[2]"(x,y), zv=&"dn[3]"(x,y));
```

```
Function povlook not found.
Try list ... to find functions!
Error in:
... ,zoom=5,w=450,h=350,    <shadow,look=povlook(blue),    xv=&"dn[1] ...
```

Kita juga bisa menghasilkan grid dalam 3D.

```
>povstart(zoom=4); ...
x=-1:0.5:1; r=1-(x+1)^2/6; ...
t=(0°:30°:360°); y=r*cos(t); z=r*sin(t); ...
writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...
povend();
```

```
Function povstart not found.
Try list ... to find functions!
Error in:
povstart(zoom=4); x=-1:0.5:1; r=1-(x+1)^2/6; t=(0°:30°:360°); ...
^
```

Dengan povgrid(), kurva-kurva dimungkinkan.

```
>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();
```

```
Function povstart not found.
Try list ... to find functions!
Error in:
povstart(center=[0,0,1],zoom=3.6); t=linspace(0,2,1000); r=exp ...
```

Objek Povray

Di atas, kita menggunakan pov3d untuk memplot permukaan. Antarmuka Povray di Euler juga bisa menghasilkan objek Povray. Objek-objek ini disimpan sebagai string di Euler, dan perlu ditulis ke file Povray. Kita memulai output dengan povstart().

```
>load povray;
>/>defaultpovray="pvengine.exe"list
>povstart(zoom=4):
```

Pertama kita mendefinisikan tiga silinder, dan menyimpannya sebagai string di Euler. Fungsi povx() dan seterusnya, hanya mengembalikan vektor [1.0.0], yang juga bisa digunakan.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...
c2=povcylinder(-povy,povy,1,povlook(yellow)); ...
c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

String-string tersebut berisi kode Povray, yang tidak perlu dipahami pada titik ini.

```
>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 kamu lihat, kita menambahkan tekstur pada objek dengan tiga warna berbeda. Ini dilakukan oleh `povlook()`, yang mengembalikan string dengan kode Povray yang relevan. Kita bisa menggunakan warna Euler default, atau mendefinisikan warna sendiri. Kita juga bisa menambahkan transparansi, atau mengubah cahaya ambient.

```
>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 persimpangan, dan menulis hasilnya ke file.

```
>writeln(povintersection([c1,c2,c3]));
```

Persimpangan dari tiga silinder sulit divisualisasikan jika kamu belum pernah melihatnya sebelumnya.

```
>povend();
```

```
Command was not allowed!
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Fungsi-fungsi berikut menghasilkan fraktal secara rekursif. Fungsi pertama menunjukkan bagaimana Euler menanganai objek Povray sederhana. Fungsi `povbox()` mengembalikan string, 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();
```

```
Command was not allowed!
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Objek bisa dikurangi dengan menggunakan perbedaan. Seperti persimpangan, ini adalah bagian dari objek CSG dari Povray.

```
>povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kita mendefinisikan objek di Povray, bukan menggunakan string di Euler. Definisi ditulis langsung ke file.

```
>povdefine("mycube",povbox(-1,1));
```

Kita bisa menggunakan objek ini di `povobject()`, yang mengembalikan string seperti biasa.

```
>c1=povobject("mycube",povlook(red));
```

Kita menghasilkan kubus kedua, memutarnya sedikit, dan menskalakannya.

```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...
rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita mengambil perbedaan dari kedua objek.

```
>writeln(povdifference(c1,c2));
```

Sekarang kita 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();

union {
    cylinder { <-1.3,0,0>,<1.3,0,0>,0.02 }
    cone {
        <1.42,0,0>,0
        <1.3,0,0>,0.08
    }
    texture { pigment { color rgb <0.470588,0.470588,0.470588> } }
}
union {
    cylinder { <-1.3,0,0>,<1.3,0,0>,0.02 }
    cone {
        <1.42,0,0>,0
        <1.3,0,0>,0.08
    }
    rotate 90*z
    texture { pigment { color rgb <0.470588,0.470588,0.470588> } }
}
union {
    cylinder { <-1.3,0,0>,<1.3,0,0>,0.02 }
    cone {
        <1.42,0,0>,0
        <1.3,0,0>,0.08
    }
    rotate -90*y
    texture { pigment { color rgb <0.470588,0.470588,0.470588> } }
}
Command was not allowed!
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Fungsi Implisit

Povray bisa memplot set di mana $f(x,y,z)=0$, sama seperti parameter implisit di plot3d. Hasilnya tampak jauh lebih baik, bagaimanapun. Sintaks untuk fungsi ini sedikit berbeda. Kamu tidak bisa menggunakan output dari ekspresi Maxima atau Euler.

$$((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2) = d$$

```
>povstart(angle=70°,height=50°,zoom=4);
>c=0.1; d=0.1; ...
writeln(povsurface("(pow(pow(x,2)+pow(y,2)-pow(c,2),2)+pow(pow(z,2)-1,2))*(pow(pow(y,2)+pow(z,2)-pow(c,2),2)+
writeAxes();...
povend(exit);
```

```
Variable or function exit not found.
Error in:
povend(exit); ...
```

```
>povstart(angle=25°,height=10°);...
writeln(povsurface("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,2,"")));...
writeAxes(); ...
povend();
```

```
Command was not allowed!
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

```
>povstart(angle=70°,height=50°,zoom=4);
```

Create the implicit surface. Note the different syntax in the expression.

```
>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...
writeAxes(); ...
povend(exit);
```

```

object {
isosurface {
function { pow(x,2)*y-pow(y,3)-pow(z,2) }
max_gradient 5
open
contained_by { box { <-1,-1,-1>, <1,1,1>
} }
texture { pigment { color rgb <0.0627451,0.564706,0.0627451> } }
finish { ambient 0.2 }
}
union {
cylinder { <-1.1,0,0>,<1.1,0,0>,0.02 }
cone {
<1.22,0,0>,0
<1.1,0,0>,0.08
}
texture { pigment { color rgb <0.470588,0.470588,0.470588> } }
}
union {
cylinder { <-1.1,0,0>,<1.1,0,0>,0.02 }
cone {
<1.22,0,0>,0
<1.1,0,0>,0.08
}
rotate 90*z
texture { pigment { color rgb <0.470588,0.470588,0.470588> } }
}
union {
cylinder { <-1.1,0,0>,<1.1,0,0>,0.02 }
cone {
<1.22,0,0>,0
<1.1,0,0>,0.08
}
rotate -90*y
texture { pigment { color rgb <0.470588,0.470588,0.470588> } }
}
Variable or function exit not found.
Error in:
... (z,2)",povlook(green))); writeAxes(); povend(exit); ...

```

Objek Mesh

Dalam contoh ini, kita menunjukkan bagaimana cara membuat objek mesh, dan menggambarnya dengan informasi tambahan. Kita ingin memaksimalkan xy di bawah kondisi $x+y=1$ dan menunjukkan sentuhan tangensial dari garis level.

```
>povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kita tidak dapat menyimpan objek ini sebagai string seperti sebelumnya, karena terlalu besar. Jadi kita mendefinisikan objek di file Povray menggunakan declare. Fungsi povtriangle() melakukan ini secara otomatis. Fungsi ini bisa menerima vektor normal seperti pov3d(). Berikut definisi objek mesh, dan langsung menulisnya ke 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);
```

Kita sekarang mendefinisikan dua cakram yang akan dipotong dengan permukaan.

```
>cl=povdisc([0.5,0.5,0],[1,1,0],2); ...
ll=povdisc([0,0,1/4],[0,0,1],2);
```

Tulis permukaan minus dua cakram ini.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tulis dua persimpangan ini.

```
>writeln(povintersection([mesh,cl],povlook(red))); ...
writeln(povintersection([mesh,ll],povlook(gray)));
```

Tambahkan sebuah titik di maksimum.

```
>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(exit);
```

```

union {
cylinder { <-0.1,0,0>,<1.1,0,0>,0.015 }
cone {
<1.19,0,0>,0
<1.1,0,0>,0.06
}
texture { pigment { color rgb <0.470588,0.470588,0.470588> } }
}
union {
cylinder { <-0.1,0,0>,<1.1,0,0>,0.015 }
cone {
<1.19,0,0>,0
<1.1,0,0>,0.06
}
}
```

```

    rotate 90*z
    texture { pigment { color rgb <0.470588,0.470588,0.470588> } }
}
union {
    cylinder { <-0.1,0,0>,<1.1,0,0>,0.015 }
    cone {
        <1.19,0,0>,0
        <1.1,0,0>,0.06
    }
    rotate -90*y
    texture { pigment { color rgb <0.470588,0.470588,0.470588> } }
}
Variable or function exit not found.
Error in:
writeAxes(0,1,0,1,0,1,d=0.015); povend(exit); ...

```

Anaglyphs di Povray

Untuk menghasilkan anaglyph untuk dilihat dengan kacamata merah/sian, 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, kamu memerlukan kacamata merah/sian untuk melihat contoh berikut dengan benar. Fungsi pov3d() memiliki saklar sederhana untuk menghasilkan anaglyph.

```
>pov3d("-exp(-x^2-y^2)/2",r=2,height=45°,>anaglyph, ...
center=[0,0,0.5],zoom=3.5);
```

```

Command was not allowed!
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
pov3d:
    if povray then povray(currentfile,w,h,w/h); endif;

```

Jika kamu membuat sebuah pemandangan dengan objek-objek, kamu perlu menempatkan pembuatan pemandangan tersebut 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 secara otomatis. Parameter-parameternya mirip dengan gabungan antara povstart() dan povend()

```
>povanaglyph ("myscene",zoom=4.5);
```

```

Command was not allowed!
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povanaglyph:
    povray(currentfile,w,h,aspect,exit);

```

Mendefinisikan Objek Sendiri

Antarmuka Povray di Euler berisi banyak objek. Namun kamu tidak terbatas pada objek-objek ini. Kamu dapat membuat objek sendiri, yang menggabungkan objek-objek lain, atau objek baru sepenuhnya. Berikut adalah contoh pembuatan torus. Perintah Povray untuk ini adalah "torus". Jadi kita mengembalikan string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berada di tengah titik asal.

```
>function povdonut (r1,r2,look "") ...
    return "torus {" + r1 + "," + r2 + look + "}";
endfunction
```

Berikut adalah torus pertama kita.

```
>t1=povdonut(0.8,0.2)
```

```
torus {0.8,0.2}
```

Sekarang kita menggunakan objek ini untuk membuat torus kedua, diterjemahkan dan diputar.

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
object { torus {0.8,0.2}
rotate 90 *x
```

```
    translate <0.8,0,0>
}
```

Sekarang kita tempatkan objek-objek ini ke dalam sebuah pemandangan. Untuk tampilannya, kita menggunakan Phong Shading.

```
>povstart(centers=[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();
```

Povray memanggil program tersebut. Namun, jika terjadi kesalahan, program tidak akan menampilkan error-nya. Kamu harus menggunakan perintah berikut:

```
>povend(<exit>);
```

Jika ada yang tidak berfungsi. Ini akan membuat jendela Povray tetap terbuka.

```
>povend(h=320,w=480);
```

```
Function povstart not found.
Try list ... to find functions!
Error in:
... rt(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); writeln(pov ...  
^
```

Berikut adalah contoh yang lebih rumit. Kita menyelesaikan persamaan:

$$Ax \leq b, \quad x \geq 0, \quad c \cdot x \rightarrow \text{Max.}$$

dan menunjukkan titik-titik yang layak serta optimum dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
>b=[10,10,10,10]';
>c=[1,1,1];
```

Pertama, kita cek apakah contoh ini memiliki solusi.

```
>x=simplex(A,b,c,>max,>check) '
```

```
[0, 1, 0.5]
```

Ya, memiliki solusi. Selanjutnya kita mendefinisikan dua objek. Yang pertama adalah bidang:

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look="")
  return povplane(a,b,look)
endfunction
```

Kemudian kita mendefinisikan persimpangan semua setengah ruang dan sebuah kubus. Berikut adalah kode di dalam fungsi:

```
>function adm (A, b, r, look="")
  ol=[];
  loop 1 to rows(A);
    ol=ol|oneplane(A[#,b[#]],r,r,r);
  end;
  ol=ol|povbox([0,0,0],[r,r,r]);
  return povintersection(ol,look);
endfunction
```

```
>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); ...
```

Sekarang kita bisa memplot pemandangan.

```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
  povlook(red,0.9)));
```

```
Function povstart not found.
Try list ... to find functions!
Error in:
... ovstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); writeln(adm ...  
^
```

Dan sebuah panah ke arah optimum.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

```
Function povlook not found.
Try list ... to find functions!
Error in:
writeln(povarrow(x,c*0.5,povlook(red))); ...  
^
```

Kita tambahkan teks ke layar. Teks adalah 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();
```

```
Function povtext not found.  
Try list ... to find functions!  
Error in:  
... "Linear Problem", [0,0.2,1.3],size=0.05,rotate=5°)); povend(); ...  
^
```

More Examples

You can find some more examples for Povray in Euler in the following files.

[Examples/Dandelin Spheres](#)
[Examples/Donut Math](#)
[Examples/Trefoil Knot](#)
[Examples/Optimization by Affine Scaling](#)

Soal-soal

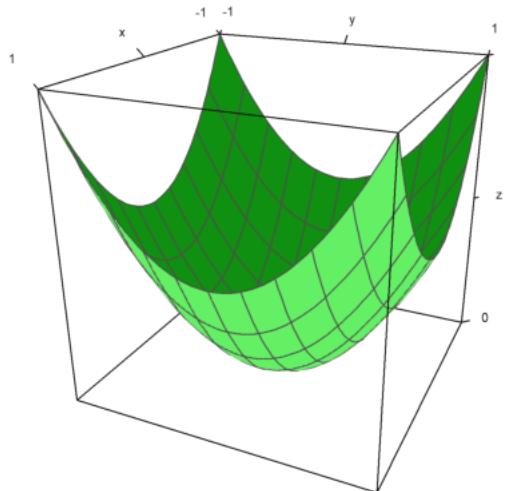
Nama : Muhammad Lutfi Ramadhan
Kelas : Matematika B 2023
NIM : 23030630021

1. Gambarkan permukaan paraboloid yang diberikan oleh fungsi berikut.

$$z = x^2 + y^2$$

Penyelesaian:

```
>plot3d("x^2 + y^2":
```

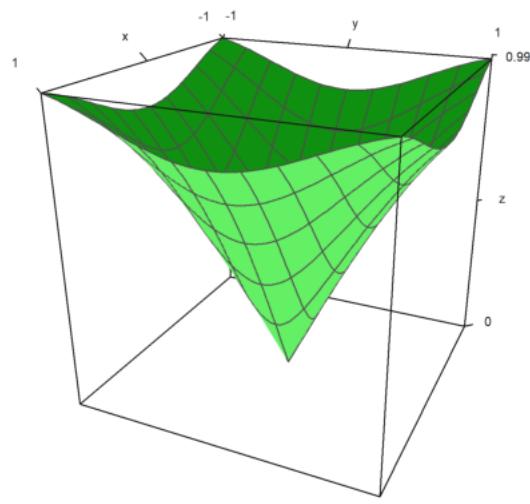


2. Gambarkan kurva heliks yang didefinisikan oleh parameter berikut

$$z = \sin\sqrt{x^2 + y^2}$$

Penyelesaian:

```
>plot3d("sin(sqrt(x^2 + y^2))":
```

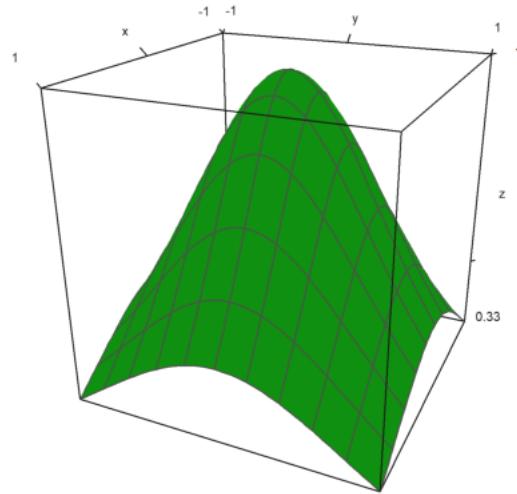


3. Gambarkan permukaan hyperboloid yang didefinisikan oleh fungsi.

$$z = \frac{1}{x^2 + y^2 + 1}$$

Penyelesaian:

```
>plot3d("1/(x^2 + y^2 + 1)":
```

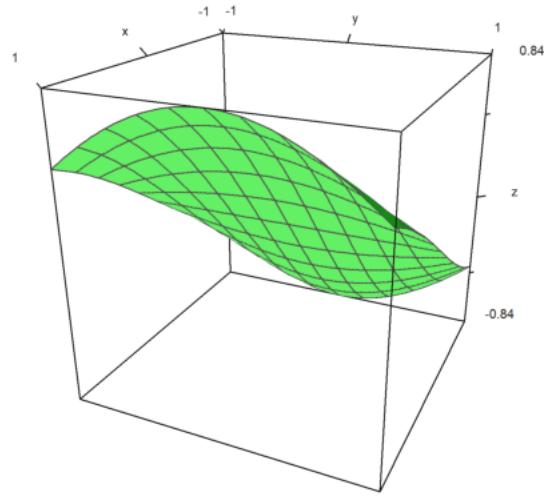


4. Gambarkan permukaan yang didefinisikan oleh fungsi berikut.

$$z = \sin(x) \cdot \cos(y)$$

Penyelesaian:

```
>plot3d("sin(x) * cos(y)":
```

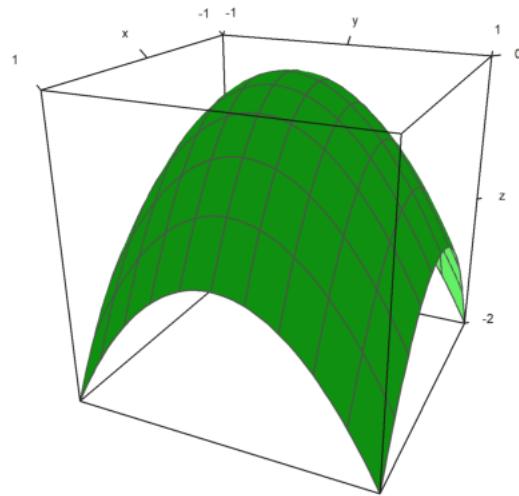


5. Gambarkan permukaan paraboloid terbalik yang didefinisikan oleh fungsi berikut.

$$z = -x^2 - y^2$$

Penyelesaian:

```
>plot3d("-x^2 - y^2") :
```



Tugas Individu Pekan9-10_Muhammad Lutfi Ramadhan_23030630021

Nama : Muhammad Lutfi Ramadhan
Kelas : Matematika B 2023
NIM : 23030630021

Kalkulus dengan EMT

Materi Kalkulus mencakup di antaranya:

- Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi)
- Limit Fungsi,
- Turunan Fungsi,
- Integral Tak Tentu,
- Integral Tentu dan Aplikasinya,
- Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

Mendefinisikan Fungsi

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:

- Menggunakan format `nama_fungsi := rumus fungsi` (untuk fungsi numerik),
- Menggunakan format `nama_fungsi &= rumus fungsi` (untuk fungsi simbolik, namun dapat dihitung secara numerik),
- Menggunakan format `nama_fungsi &&= rumus fungsi` (untuk fungsi simbolik murni, tidak dapat dihitung langsung),
- Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah `function` (bukan sebagai ekspresi).

Berikut adalah beberapa contoh cara mendefinisikan fungsi:

$$f(x) = 2x^2 + e^{\sin(x)}.$$

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
```

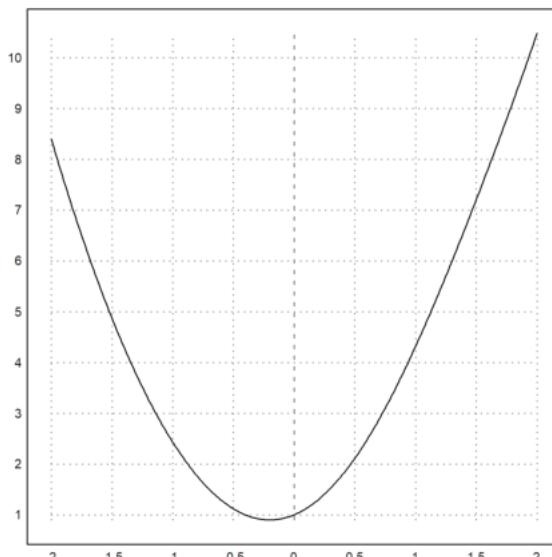
```
1
4.31977682472
20.7392088022
```

```
>f(a) // tidak dapat dihitung nilainya
```

```
Variable or function a not found.
Error in:
f(a) // tidak dapat dihitung nilainya ...
^
```

Silakan Anda plot kurva fungsi di atas!

```
>plot2d("2*x^2+exp(sin(x))":
```



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(4)

0.4

>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!

```
>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:

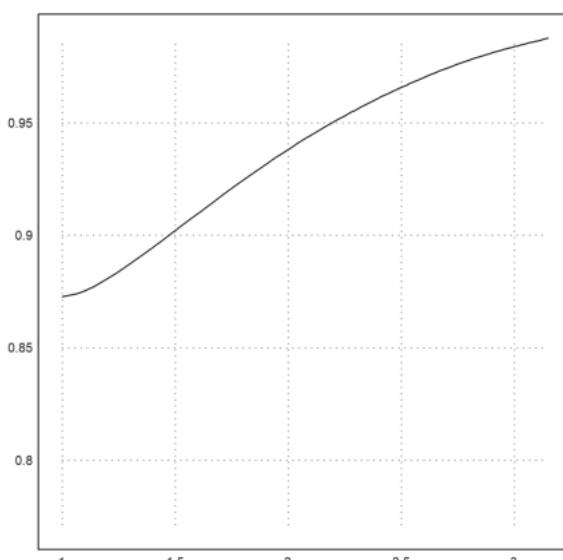
$$h(x) = f(g(x))$$

dan

$$u(x) = g(f(x))$$

bersama-sama kurva fungsi f dan g dalam satu bidang koordinat.

```
>function h(x) := f(g(x))
>function u(x) := g(f(x))
>plot2d("h(x)", "u(x)", -10, 10):
```



```
>f(0:10) // nilai-nilai f(0), f(1), f(2), ..., f(10)
```

```
[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,
99.929, 130.69, 163.51, 200.58]

>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi

[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,
99.929, 130.69, 163.51, 200.58]

>gmap(200:210)

[0.987534, 0.987596, 0.987657, 0.987718, 0.987778, 0.987837,
0.987896, 0.987954, 0.988012, 0.988069, 0.988126]
```

Misalkan kita akan mendefinisikan fungsi

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

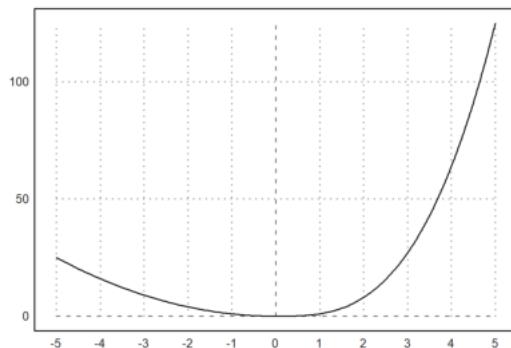
>f(-2)

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 // fungsi simbolik


$$2^x$$


>$f(a) // nilai fungsi secara simbolik


$$2e^a$$


>f(E) // nilai fungsi berupa bilangan desimal

30.308524483

>$f(E), $float(%)


$$2e^E$$


30.30852448295852
```

```

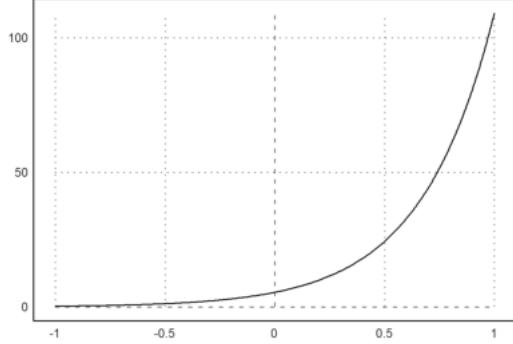
>function g(x) &= 3*x+1
3 x + 1
2 E

>function h(x) &= f(g(x)) // komposisi fungsi

3 x + 1
2 E

>plot2d("h(x)",-1,1):

```



Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan fungsi-fungsi tersebut dan komposisinya di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik fungsi-fungsi tersebut dan komposisi-komposisi 2 fungsi.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

```

>function f(x) := x^3-2*x
>function g(x) := sqrt(x)
>function h(x,y) := sin(x)*cos(y)
>function u(x,y) := x^2+y^2
>function p(x) := x^2
>function a(x) := g(f(x))
>function b(x) := p(f(x))
>f(3)

```

21

```
>g(10)
```

3.16227766017

```
>h(1,1)
```

0.454648713413

```
>u(2,7)
```

53

```
>p(-4)
```

16

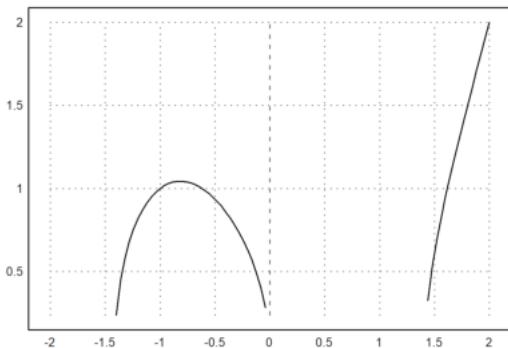
```
>a(2)
```

2

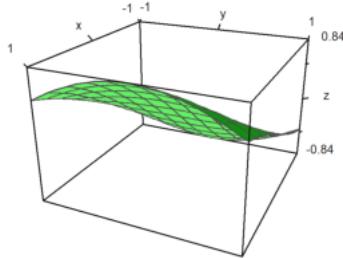
```
>b(6)
```

41616

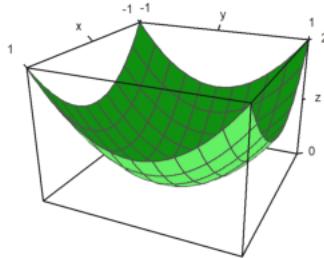
```
>plot2d("a(x)":
```



```
>plot3d("h(x, y) ":
```



```
>plot3d("u(x, y) ":
```



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))
```

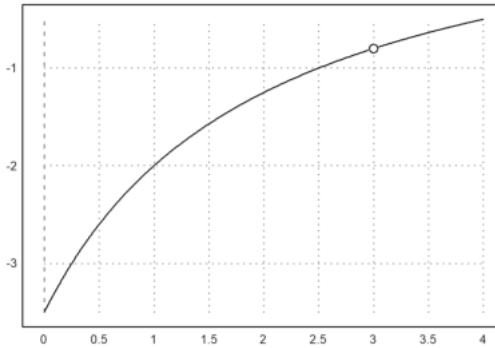
$$\lim_{x \rightarrow \infty} \frac{\sqrt{x^2 - 3x}}{x + 1} = 1$$

```
>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

$$\lim_{x \rightarrow 3} \frac{x^3 - 13x^2 + 51x - 63}{x^3 - 4x^2 - 3x + 18} = -\frac{4}{5}$$

Fungsi tersebut diskontinu di titik $x=3$. Berikut adalah grafik fungsinya.

```
>aspect(1.5); plot2d("(x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18)",0,4); plot2d(3,-4/5,>points,style="ow",>add):
```

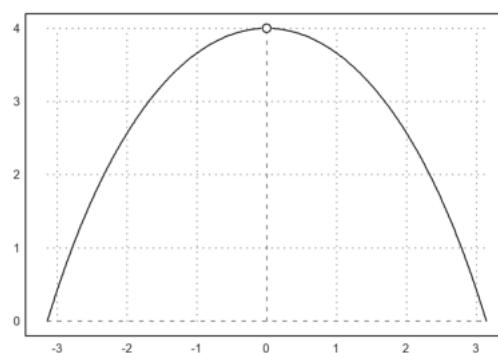


```
>$limit(2*x*sin(x)/(1-cos(x)),x,0)
```

$$2 \left(\lim_{x \rightarrow 0} \frac{x \sin x}{1 - \cos x} \right) = 4$$

Fungsi tersebut diskontinu di titik $x=0$. Berikut adalah grafik fungsinya.

```
>plot2d("2*x*sin(x)/(1-cos(x))",-pi,pi); plot2d(0,4,>points,style="ow",>add):
```



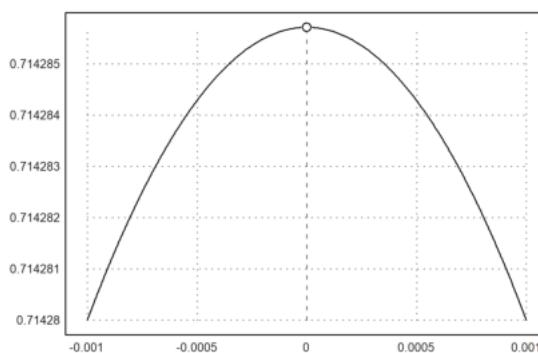
```
>$limit(cot(7*h)/cot(5*h),h,0)
```

$$\frac{5}{7}$$

$$\lim_{h \rightarrow 0} \frac{\cot(7h)}{\cot(5h)} = \frac{5}{7}$$

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di $x=0$. Berikut adalah grafiknya.

```
>plot2d("cot(7*x)/cot(5*x)",-0.001,0.001); plot2d(0,5/7,>points,style="ow",>add):
```

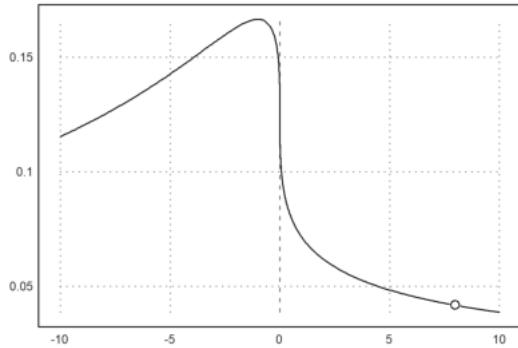


```
>$showev('limit(((x/8)^(1/3)-1)/(x-8),x,8))
```

$$\lim_{x \rightarrow 8} \frac{\frac{x^{\frac{1}{3}}}{2} - 1}{x - 8} = \frac{1}{24}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x/8)^(1/3)-1)/(x-8)",-10,10); plot2d(8,1/24,>points,style="ow",>add):
```

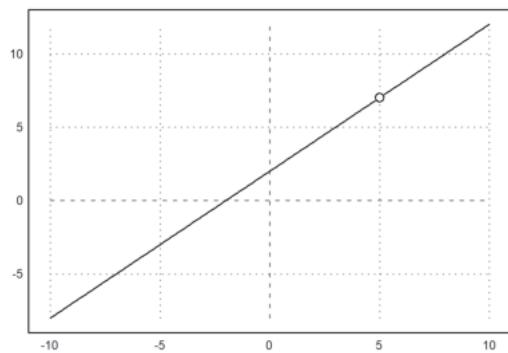


```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

$$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(x^2-3*x-10)/(x-5)", -10,10); plot2d(5,7,>points,style="ow",>add):
```

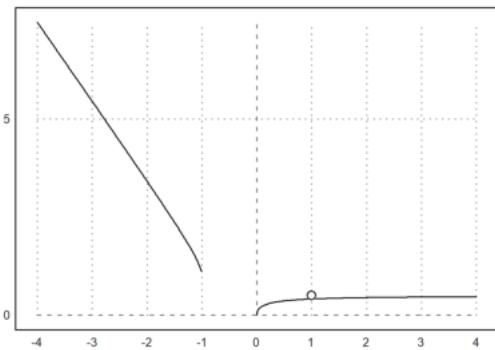


```
>$showev('limit(sqrt(x^2+x)-x,x,inf))
```

$$\lim_{x \rightarrow \infty} \sqrt{x^2 + x} - x = \frac{1}{2}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("sqrt(x^2+x)-x",-4,4); plot2d(1,1/2,>points,style="ow",>add):
```



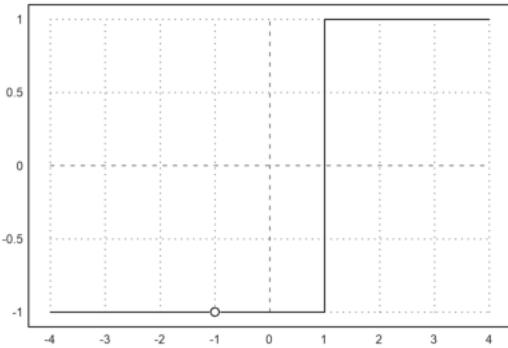
```
>$showev('limit(abs(x-1)/(x-1),x,1,minus))
```

$$\lim_{x \uparrow 1} \frac{|x - 1|}{x - 1} = -1$$

Hitung limit di atas untuk x menuju 1 dari kanan.

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

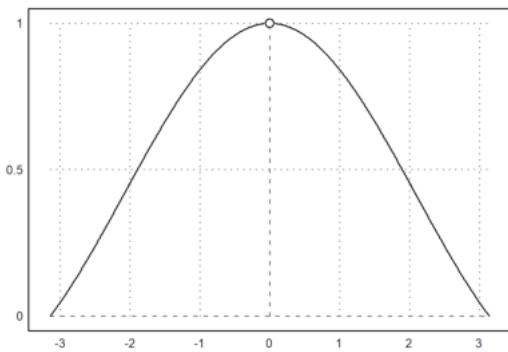
```
>plot2d("abs(x-1)/(x-1)",-4,4); plot2d(-1,-1,>points,style="ow",>add):
```



```
>$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):
```

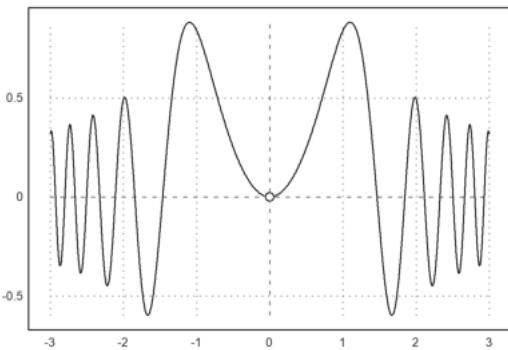


```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("sin(x^3)/x",-3,3); plot2d(0,0,>points,style="ow",>add):
```



```
>$showev('limit(log(x), x, minf))
```

$$\lim_{x \rightarrow -\infty} \log x = \text{infinity}$$

```
>$showev('limit((-2)^x,x, inf))
```

$$\lim_{x \rightarrow \infty} (-2)^x = \text{infinity}$$

```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t \uparrow 2} t - \sqrt{2-t} = 2$$

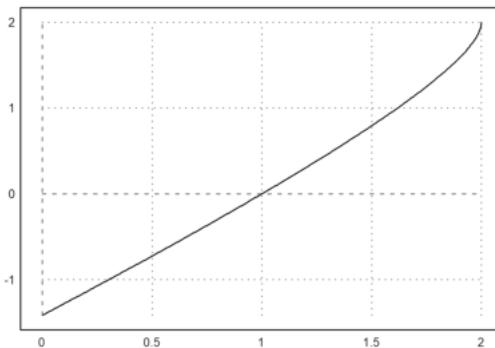
```
>$showev('limit(t-sqrt(2-t),t,2,plus))
```

$$\lim_{t \downarrow 2} t - \sqrt{2-t} = 2$$

```
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
```

$$\lim_{t \downarrow 5} t - \sqrt{2-t} = 5 - \sqrt{3} i$$

```
>plot2d("x-sqrt(2-x)",0,2):
```

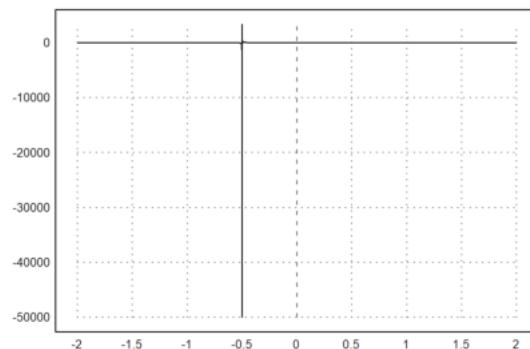


```
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
```

$$\lim_{x \rightarrow 3} \frac{x^2 - 9}{2x^2 - 5x - 3} = \frac{6}{7}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(x^2-9)/(2*x^2-5*x-3)",-2,2); plot2d(3,6/7,>points,style="ow",>add):
```

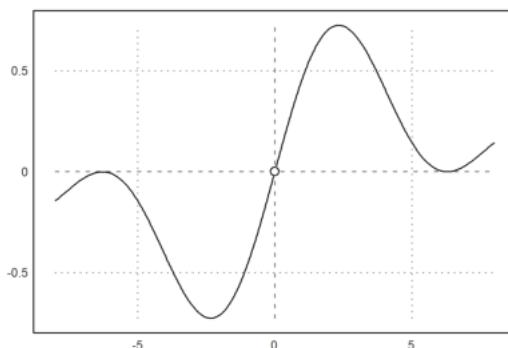


```
>$showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1-cos(x))/x",-8,8); plot2d(0,0,>points,style="ow",>add):
```

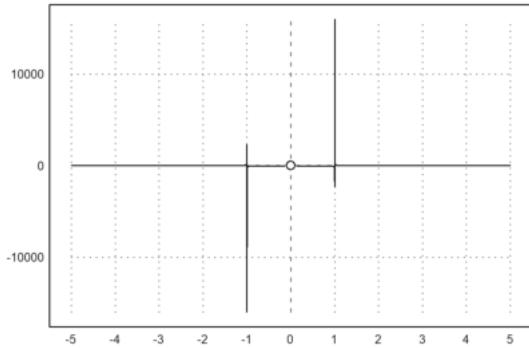


```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

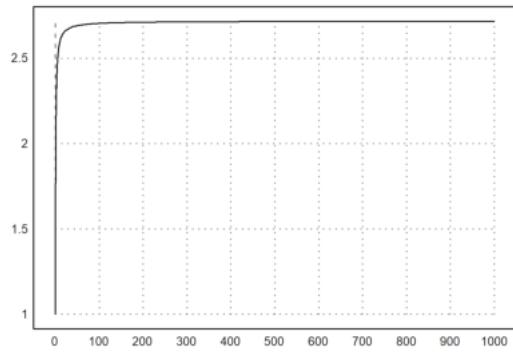
```
>plot2d("(x^2+abs(x))/(x^2-abs(x))",-5,5); plot2d(0,-1,>points,style="ow",>add):
```



```
>$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))
```

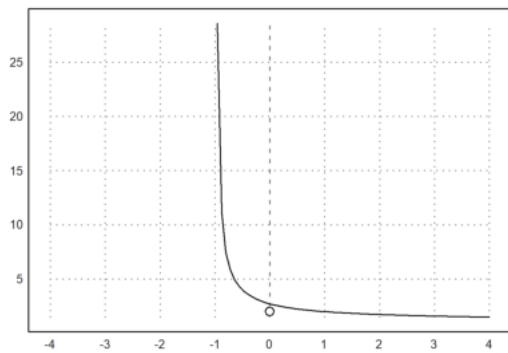
$$\lim_{x \rightarrow \infty} \left(\frac{k}{x} + 1 \right)^x = e^k$$

```
>$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); plot2d(0,2,>points,style="ow",>add):
```



```
>$showev('limit((x/(x+k))^x,x,inf))
```

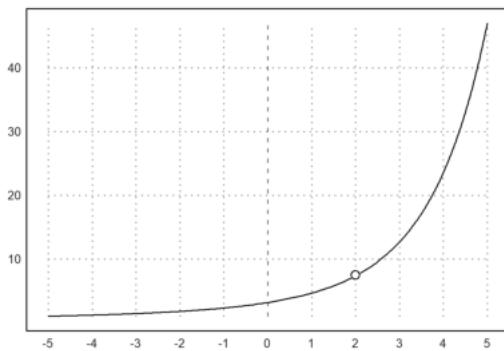
$$\lim_{x \rightarrow \infty} \left(\frac{x}{x+k} \right)^x = e^{-k}$$

```
>$showev('limit((E^x-E^2)/(x-2),x,2))
```

$$\lim_{x \rightarrow 2} \frac{e^x - e^2}{x - 2} = e^2$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(E^x-E^2)/(x-2)",-5,5); plot2d(2,7.5,>points,style="ow",>add):
```



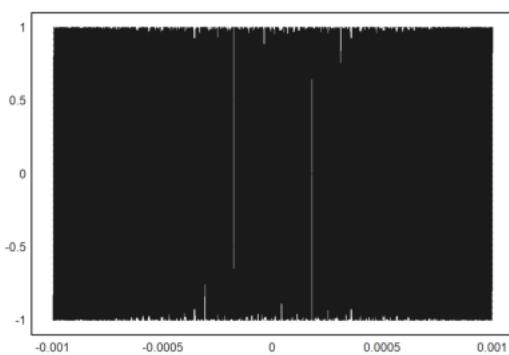
```
>$showev('limit(sin(1/x),x,0))
```

$$\lim_{x \rightarrow 0} \sin\left(\frac{1}{x}\right) = \text{ind}$$

```
>$showev('limit(sin(1/x),x,inf))
```

$$\lim_{x \rightarrow \infty} \sin\left(\frac{1}{x}\right) = 0$$

```
>plot2d("sin(1/x)",-0.001,0.001):
```



Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

```
>function f(x) := (1+1/x)^x
>function g(x) := x/sqrt(x^2+1)
>function h(x) := (exp(x)-1)/x
>function u(x) := sin(x)/x
>function p(x) := (sqrt(x+1)-sqrt(x))/(x+1-x)
>$showev('limit(f(x),x,infinity))
```

$$2 \left(\lim_{x \rightarrow \infty} e^x \right) = 2 \text{ und}$$

```
>$showev('limit(g(x),x,infinity))
```

$$\lim_{x \rightarrow \infty} 3x + 1 = \text{infinity}$$

```
>$showev('limit(h(x),x,0))
```

$$2 \left(\lim_{x \rightarrow 0} e^{3x+1} \right) = 2e$$

```
>$showev('limit(u(x),x,1))
```

$$\lim_{x \rightarrow 1} u(x) = \lim_{x \rightarrow 1} u(x)$$

```
>$showev('limit(p(x),x,infinity))
```

$$\lim_{x \rightarrow \infty} p(x) = \lim_{x \rightarrow \infty} p(x)$$

Turunan Fungsi

Definisi turunan:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>$showev('limit(((x+h)^2-x^2)/h,h,0)) // turunan x^2
lim (x + h)^2 - x^2
      h
      2 x + h^2

>p &= expand((x+h)^2-x^2)|simplify; $p // pembilang dijabarkan dan disederhanakan
2 x + h

>q &=ratsimp(p/h); $q // ekspresi yang akan dihitung limitnya disederhanakan
2 x

>$limit(q,h,0) // nilai limit sebagai turunan
2 x

>$showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
lim (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.

$$\text{Akan ditunjukkan bahwa } f'(x) = \lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} = nx^{n-1}$$

Pertama, ekspansikan $(x+h)^n$, yakni:

$$\begin{aligned} (x+h)^n &= \sum_{k=0}^n \binom{n}{k} x^{n-k} h^k \\ \Leftrightarrow (x+h)^n &= \binom{n}{0} x^n + \binom{n}{1} x^{n-1} h + \binom{n}{2} x^{n-2} h^2 + \dots + \binom{n}{n} h^n \\ \Leftrightarrow (x+h)^n &= x^n + nx^{n-1} h + \binom{n}{2} x^{n-2} h^2 + \binom{n}{3} x^{n-3} h^3 + \dots + h^n \end{aligned}$$

$$\begin{aligned} \text{Sehingga, } f'(x) \text{ menjadi: } f'(x) &= \lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} \\ \Leftrightarrow f'(x) &= \lim_{h \rightarrow 0} \frac{x^n + nx^{n-1} h + \binom{n}{2} x^{n-2} h^2 + \binom{n}{3} x^{n-3} h^3 + \dots + h^n - x^n}{h} \\ \Leftrightarrow f'(x) &= \lim_{h \rightarrow 0} nx^{n-1} + \binom{n}{2} x^{n-2} h + \binom{n}{3} x^{n-3} h^2 + \dots + h^{n-1} \\ \Leftrightarrow f'(x) &= nx^{n-1}. \text{ Terbukti.} \end{aligned}$$

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

$$\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h} = \cos x$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan $\sin(x+h)$ dengan menggunakan rumus jumlah dua sudut.

$$\text{Akan ditunjukkan bahwa } \lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h} = \cos x$$

Diketahui bahwa:

$$1). \sin(x+h) = \sin x \cos h + \cos x \sin h$$

$$2). \lim_{h \rightarrow 0} \frac{1 - \cos h}{h} = 0$$

$$3). \lim_{h \rightarrow 0} \frac{\sin h}{h} = 1$$

$$\begin{aligned} &\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h} \\ &= \lim_{h \rightarrow 0} \frac{\sin x \cos h + \cos x \sin h - \sin x}{h} \end{aligned}$$

$$\begin{aligned}
&= \lim_{h \rightarrow 0} \left[-\sin x \cdot \frac{1 - \cos h}{h} + \cos x \cdot \frac{\sin h}{h} \right] \\
&= (-\sin x) \left[\lim_{h \rightarrow 0} \frac{1 - \cos h}{h} + (\cos x) \lim_{h \rightarrow 0} \frac{\sin h}{h} \right] \\
&= (-\sin x)(0) + (\cos x)(1) = \cos x. \text{ Terbukti.}
\end{aligned}$$

```
>$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.

Ambil $f(x) =^a \log x$.

$$\begin{aligned}
&\lim_{h \rightarrow 0} \frac{^a \log(x+h) - ^a \log x}{h} \\
&= \lim_{h \rightarrow 0} \frac{^a \log \frac{(x+h)}{x}}{h} \\
&= \lim_{h \rightarrow 0} \frac{^a \log(1 + \frac{h}{x})}{h} \\
&= \lim_{h \rightarrow 0} \frac{^a \log(1 + \frac{h}{x})}{\frac{h}{x} \cdot x} \\
&= \lim_{h \rightarrow 0} \frac{\frac{x}{h} \cdot ^a \log(1 + \frac{h}{x})}{x} \\
&= \lim_{h \rightarrow 0} \frac{^a \log(1 + \frac{h}{x})^{\frac{x}{h}}}{x} \\
&= \frac{\lim_{h \rightarrow 0} ^a \log(1 + \frac{h}{x})^{\frac{x}{h}}}{\lim_{h \rightarrow 0} x} \\
&= \frac{1}{x \cdot {}^e \log a} \\
&= \frac{1}{x \cdot \ln a}
\end{aligned}$$

Menggunakan hasil di atas, maka:

$$\frac{d \ln x}{dx} = \frac{d {}^e \log x}{dx} = \frac{1}{x \cdot \ln e} = \frac{1}{x}. \text{ Terbukti.}$$

```
>$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
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is x an integer?

```

```

Use assume!
Error in:
$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x ...
^
```

Maxima bermasalah dengan limit:

$$\lim_{h \rightarrow 0} \frac{e^{x+h} - e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{e^h - 1}{h} = 1$$

```
>$showev('factor(E^(x+h)-E^x))
```

$$factor(e^{x+h} - e^x) = (e^h - 1) e^x$$

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

$$\left(\lim_{h \rightarrow 0} \frac{e^h - 1}{h} \right) e^x = e^x$$

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

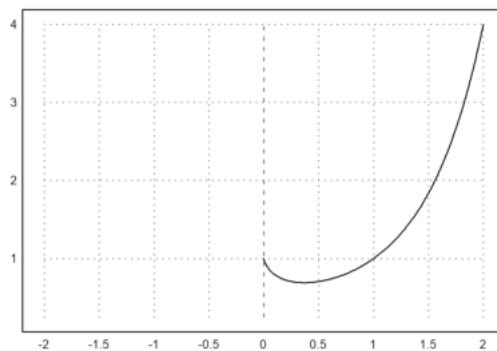
$$\begin{matrix} x \\ x \end{matrix}$$

```
>$showev('limit(f(x),x,0))
```

$$\lim_{x \rightarrow 0} x^x = 1$$

Silakan Anda gambar kurva

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



```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = \text{infinity}$$

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)

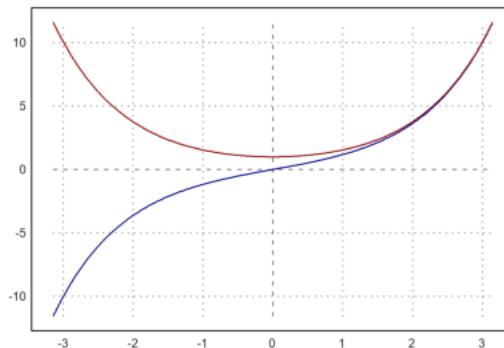
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)


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


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)')


$$\frac{d}{dx} \sin^2(3x^5 + 7) = 30x^4 \cos(3x^5 + 7) \sin(3x^5 + 7)$$

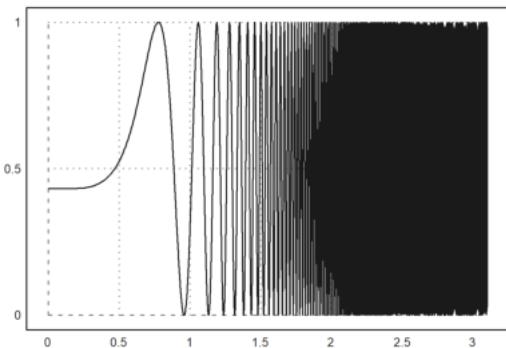

>% with x=3

%at  $\left(\frac{d}{dx} \sin^2(3x^5 + 7), x = 3\right) = 2430 \cos 736 \sin 736$ 

>$float(%)

%at  $\left(\frac{d^{1.0}}{dx^{1.0}} \sin^2(3.0x^5 + 7.0), x = 3.0\right) = 1198.728637211748$ 

>plot2d(f,0,3.1):
```



```
>function f(x) &=5*cos(2*x)-2*x*sin(2*x) // mendefinisikan fungsi f
5 cos(2 x) - 2 x sin(2 x)
```

```
>function df(x) &=diff(f(x),x) // fd(x) = f'(x)
- 12 sin(2 x) - 4 x cos(2 x)
```

```
>$'f(1)=f(1), $float(f(1)), '$f(2)=f(2), $float(f(2)) // nilai f(1) dan f(2)
```

$$f(1) = 5 \cos 2 - 2 \sin 2$$

$$-3.899329036387075$$

$$f(2) = 5 \cos 4 - 4 \sin 4$$

$$-0.2410081230863468$$

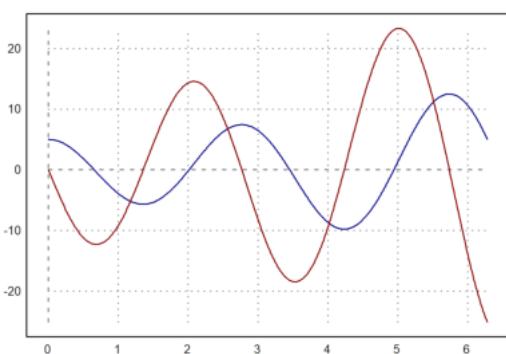
```
>xp=solve("df(x)",1,2,0) // solusi f'(x)=0 pada interval [1, 2]
```

$$1.35822987384$$

```
>df(xp), f(xp) // cek bahwa f'(xp)=0 dan nilai ekstrim di titik tersebut
```

$$\begin{matrix} 0 \\ -5.67530133759 \end{matrix}$$

```
>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.

1.

```
>function f(x) := x^2
>$showev('limit(((x+h)^2-x^2)/h),h,0) // turunan x^2
```

```

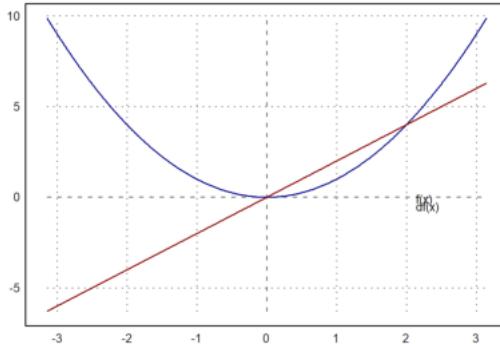

$$\lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2}{h} = 2x$$


>function df(x) &= limit(((x+h)^2-x^2)/h,h,0); $df(x)// df(x) = f'(x)

2x

>plot2d(["f(x)","df(x")],-pi,pi,color=[blue,red]), label("f(x)",2,0.6), label("df(x)",2,0.17):

```



2.

```

>function f(x) := sin(x)*cos(x)
>$showev('limit((sin(x+h)*cos(x+h))-sin(x)*cos(x))/h,h,0)) // turunan sin(x)*cos(x)

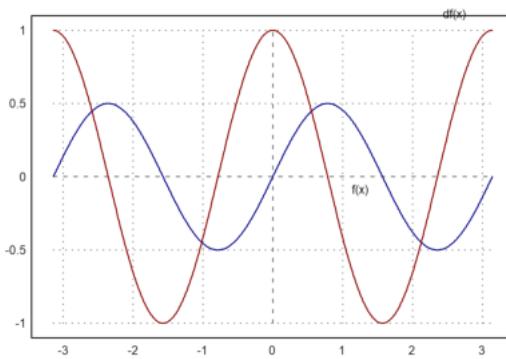

$$\lim_{h \rightarrow 0} \frac{\cos(x+h)\sin(x+h) - \cos x \sin x}{h} = \cos^2 x - \sin^2 x$$


>function df(x) &= limit((sin(x+h)*cos(x+h))-sin(x)*cos(x)/h,h,0); $df(x)// df(x) = f'(x)

\cos^2 x - \sin^2 x

>plot2d(["f(x)","df(x")],-pi,pi,color=[blue,red]), label("f(x)",1,0), label("df(x)",2.3,1.2):

```



3.

```

>function f(x) := (log(x))^5
>$showev('limit((log(x+h))^5-(log(x))^5)/h,h,0)) // turunan (log(x))^5

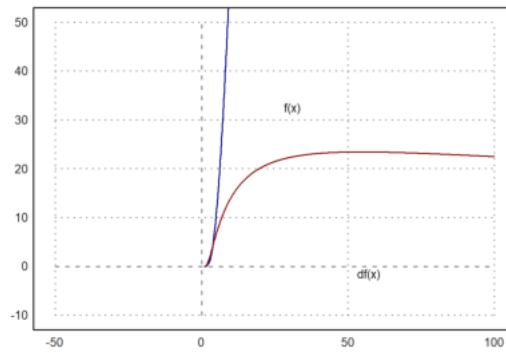

$$\lim_{h \rightarrow 0} \frac{\log^5(x+h) - \log^5 x}{h} = \frac{5 \log^4 x}{x}$$


>function df(x) &= limit((log(x+h))^5-(log(x))^5)/h,h,0); $df(x)// df(x) = f'(x)

\frac{5 \log^4 x}{x}

>plot2d(["f(x)","df(x")],-50,100,-10,50,color=[blue,red]), label("f(x)",25,35), label("df(x)",50,1):

```



4.

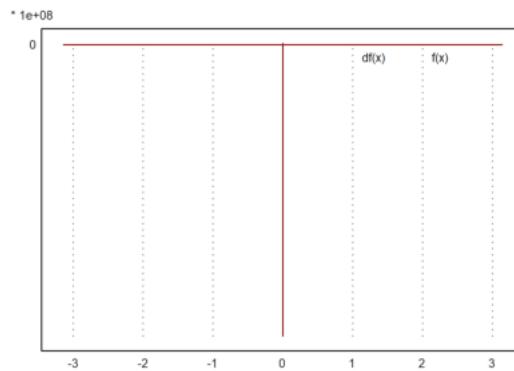
```
>function f(x) := cos(1/x)
>$showev('limit((cos(1/(x+h))-cos(1/x))/h,h,0)) // turunan cos(1/x)


$$\lim_{h \rightarrow 0} \frac{\cos\left(\frac{1}{x+h}\right) - \cos\left(\frac{1}{x}\right)}{h} = \frac{\sin\left(\frac{1}{x}\right)}{x^2}$$


>function df(x) &= limit((cos(1/(x+h))-cos(1/x))/h,h,0); $df(x)// df(x) = f'(x)


$$\frac{\sin\left(\frac{1}{x}\right)}{x^2}$$


>plot2d(["f(x)","df(x")],-pi,pi,color=[blue,red]), label("f(x"),2,0.4), label("df(x"),1,-0.5):
```



5.

```
>function f(x) := sqrt(tan(x))
>$showev('limit((sqrt(tan(x+h))-sqrt(tan(x)))/h,h,0)) // turunan exp(x)*cos(x)


$$\lim_{h \rightarrow 0} \frac{\sqrt{\tan(x+h)} - \sqrt{\tan x}}{h} = \frac{1}{2 \cos^2 x \sqrt{\tan x}}$$


>function df(x) &= limit((sqrt(tan(x+h))-sqrt(tan(x)))/h,h,0); $df(x)// df(x) = f'(x)


$$\frac{1}{2 \cos^2 x \sqrt{\tan x}}$$


>plot2d(["f(x)","df(x")],-10,10,-10,10,color=[blue,red]), label("f(x"),4.5,0), label("df(x"),5.5,5)
```

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):



```

```
>function map f(x) &= E^(-x^2)
```

$$\frac{e^{-x^2}}{\sqrt{\pi}}$$

```
>$showev('integrate(f(x),x))
```

$$\int e^{-x^2} dx = \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

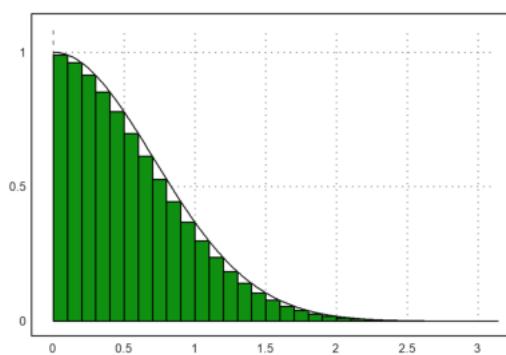
$$\operatorname{erf}(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}} dx.$$

Kita tidak dapat menggunakan teorema Dasar Kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

$$\int_0^\pi e^{-x^2} dx$$

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```



Integral tentu

$$\int_0^\pi e^{-x^2} dx$$

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva $y=f(x)$ tersebut. Langkah-langkahnya adalah sebagai berikut.

```
>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-nilai x
>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)
/// jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!
```

Hasilnya adalah:

$$\int_0^\pi e^{-x^2} dx = 0.8362196102528469$$

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai $f(x)$ untuk $x = 0.1, 0.2, 0.3, \dots, 3.2$.

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

0.836219610253

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervallnya 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
```

```

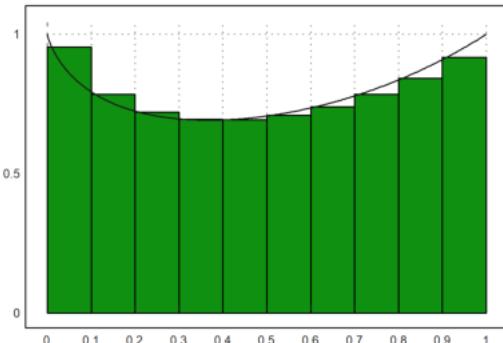
x
x

>$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))


$$\int_0^1 \sin^2(3x^5 + 7) \, dx = \frac{\gamma(-\frac{1}{5}) \sin(\frac{14}{5}\pi)}{10} - \frac{((6\gamma_{incomplete}(-\frac{1}{6}, \frac{1}{5}) + 6\gamma_{incomplete}(-\frac{1}{6}, -\frac{1}{6})) \sin(\frac{14}{5}) + (6\gamma_{incomplete}(-\frac{1}{6}, \frac{1}{5}) \sin(\frac{14}{5}) - 6\gamma_{incomplete}(-\frac{1}{6}, -\frac{1}{6}) \cos(\frac{14}{5}) \sin(\frac{\pi}{10})) - 60)/120$$


>&float(%)


$$\int_0^{1.0} \sin^2(3.0x^5 + 7.0) \, dx =$$

0.09820784258795788 - 0.00833333333333333
(0.3090169943749474 (0.1367372182078336
(4.192962712629476 I gamma_incomplete(0.2, 6.0 I)
- 4.192962712629476 I gamma_incomplete(0.2, - 6.0 I))
+ 0.9906073556948704 (4.192962712629476 gamma_incomplete(0.2, 6.0 I)
+ 4.192962712629476 gamma_incomplete(0.2, - 6.0 I))) - 60.0

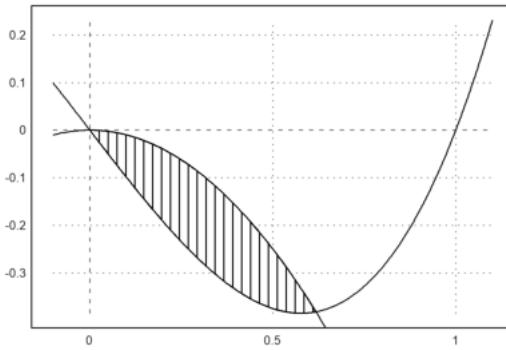
```

```
>$showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)
```

$$\int_0^1 x e^{-x} dx = 1 - 2e^{-1}$$

Aplikasi Integral Tentu

```
>plot2d("x^3-x",-0.1,1.1); plot2d("-x^2",>add); ...
b=solve("x^3-x+x^2",0.5); x=linspace(0,b,200); xi=flipx(x); ...
plot2d(x|xi,x^3-x|-xi^2,>filled,style="|",fillcolor=1,>add); // Plot daerah antara 2 kurva
```



```
>a=solve("x^3-x+x^2",0), b=solve("x^3-x+x^2",1) // absis titik-titik potong kedua kurva
```

$$\begin{array}{l} 0 \\ 0.61803398875 \end{array}$$

```
>integrate("(-x^2)-(x^3-x)",a,b) // luas daerah yang diarsir
```

$$0.0758191713542$$

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```
>a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik potong kedua kurva secara eksak
```

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

```
>$showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara eksak
```

$$\int_0^{\frac{\sqrt{5}-1}{2}} -x^3 - x^2 + x \, dx = \frac{13 - 5^{\frac{3}{2}}}{24}$$

```
>$float(%)
```

$$\int_{0.0}^{0.6180339887498949} -1.0 x^3 - 1.0 x^2 + x \, dx = 0.07581917135421037$$

Panjang Kurva

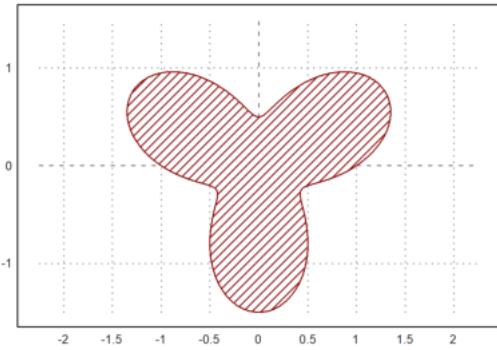
Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

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

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 terlebih dahulu
```



```
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
```

Tugas Individu Pekan9-10_Muhammad Lutfi Ramadhan_23030630021

```
>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
```

Tugas Individu Pekan9-10_Muhammad Lutfi Ramadhan_23030630021

```
>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
```

Tugas Individu Pekan9-10_Muhammad Lutfi Ramadhan_23030630021

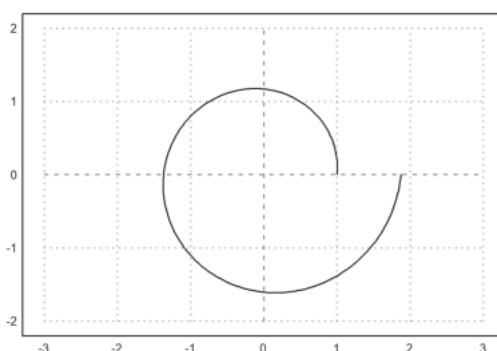
```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
```

$$\int_0^{2\pi} ds(x) \, dx$$

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara numerik dengan perintah EMT.

```
>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

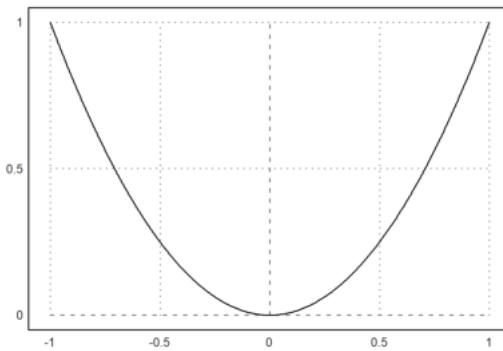
```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
```

Tugas Individu Pekan9-10_Muhammad Lutfi Ramadhan_23030630021

```
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
```

Tugas Individu Pekan9-10_Muhammad Lutfi Ramadhan_23030630021

```
>plot2d("x^2",xmin=-1,xmax=1):
```



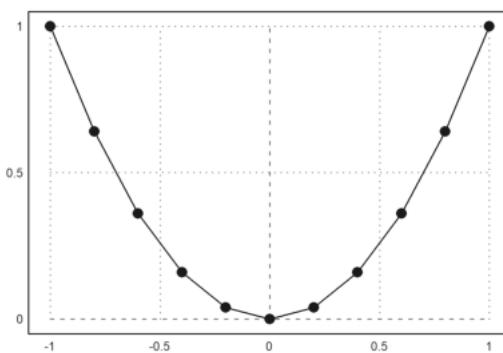
```
>showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))
```

$$\int_{-1}^1 \sqrt{4x^2 + 1} dx = \frac{\operatorname{asinh} 2 + 2\sqrt{5}}{2}$$

```
>float(%)
```

$$\int_{-1.0}^{1.0} \sqrt{4.0x^2 + 1.0} dx = 2.957885715089195$$

```
>x=-1:0.2:1; y=x^2; plot2d(x,y); ...
plot2d(x,y,points=1,style="o#",add=1):
```



Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```
>i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])^2+(y[i+1]-y[i])^2))
```

2.95191957027

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

Koordinat Kartesius

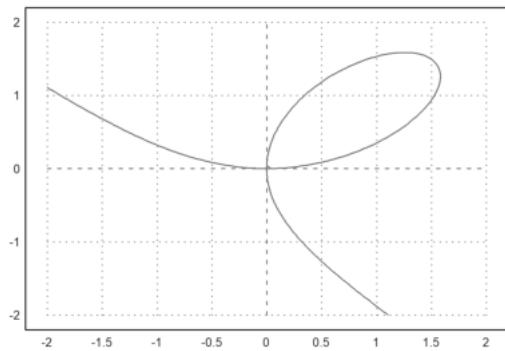
Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

$$x^3 + y^3 - 3xy = 0.$$

```
>z &= x^3+y^3-3*x*y; $z
```

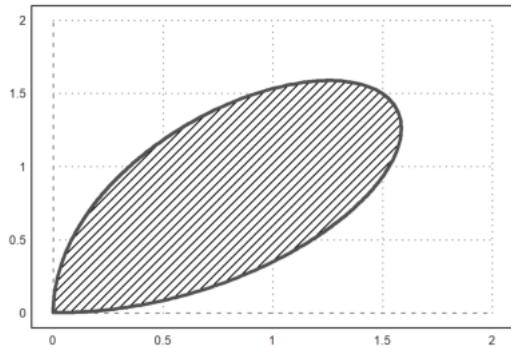
$$y^3 - 3xy + x^3$$

```
>plot2d(z,r=2,level=0,n=100):
```



Kita tertarik pada kurva di kuadran pertama.

```
>plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="/"):
```



Kita selesaikan persamaannya untuk x.

```
>$z with y=l*x, sol &= solve(% ,x); $sol
```

$$\begin{aligned} l^3 x^3 + x^3 - 3 l x^2 \\ \left[x = \frac{3 l}{l^3 + 1}, x = 0 \right] \end{aligned}$$

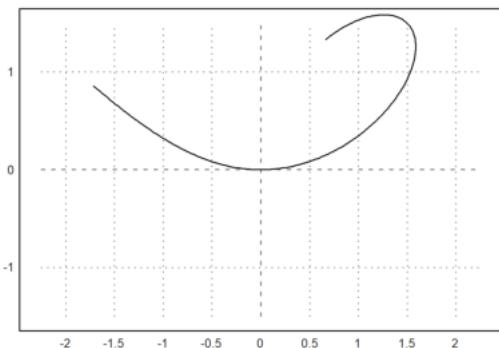
Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); $'f(l)=f(l)
```

$$f(l) = \frac{3l}{l^3 + 1}$$

Fungsi tersebut juga dapat digunakan untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai y=l*x, yakni x=f(l) dan y=l*f(l).

```
>plot2d(&f(x),&x*f(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):
```



Elemen panjang kurva adalah:

$$ds = \sqrt{f'(l)^2 + (lf'(l) + f(l))^2}.$$

```
>function ds(l) &= ratsimp(sqrt(diff(f(l),l)^2+diff(l*f(l),l)^2)); $'ds(l)=ds(l)
```

$$ds(l) = \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}}$$

```
>$integrate(ds(l),l,0,1)
```

$$\int_0^1 \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}} dl$$

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral tersebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
>2*integrate("ds(x)",0,1)

4.91748872168

>2*romberg(&ds(x),0,1) // perintah Euler lain untuk menghitung nilai hampiran integral yang sama

4.91748872168

Perhitungan di atas dapat dilakukan untuk sebarang fungsi x dan y dengan mendefinisikan fungsi EMT,
misalnya kita beri nama panjangkurva. Fungsi ini selalu memanggil Maxima untuk menurunkan fungsi
yang diberikan.

>function panjangkurva(fx,fy,a,b) ...
ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");
return romberg(ds,a,b);
endfunction

>panjangkurva("x","x^2",-1,1) // cek untuk menghitung panjang kurva parabola sebelumnya

2.95788571509
```

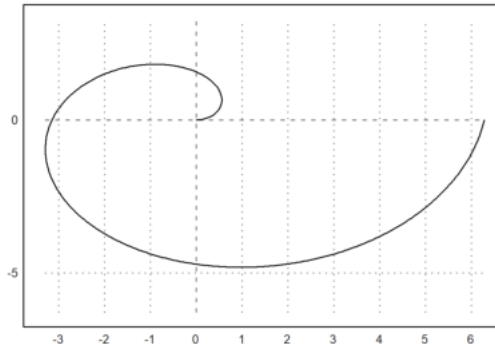
Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)"),mxm("x*f(x)"),0,1) // cek contoh terakhir, bandingkan hasilnya!

4.91748872168
```

Kita hitung panjang spiral Archimedes berikut ini dengan fungsi tersebut.

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



```
>panjangkurva("x*cos(x)","x*sin(x)",0,2*pi)
```

21.2562941482

Berikut kita definisikan fungsi yang sama namun dengan Maxima, untuk perhitungan eksak.

```
>&kill(ds,x,fx,fy)

done

>function ds(fx,fy) &=& sqrt(diff(fx,x)^2+diff(fy,x)^2)
          ^2           ^2
          sqrt(diff (fy, x) + diff (fx, x))

>sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk menghitung panjang kurva terakhir di atas
          ^2
          sqrt((cos x - x sin x)^2 + (sin x + x cos x)^2)

>$sol | trigreduce | expand, $integrate(% ,x,0,2*pi), %()
```

$$\sqrt{x^2 + 1}$$

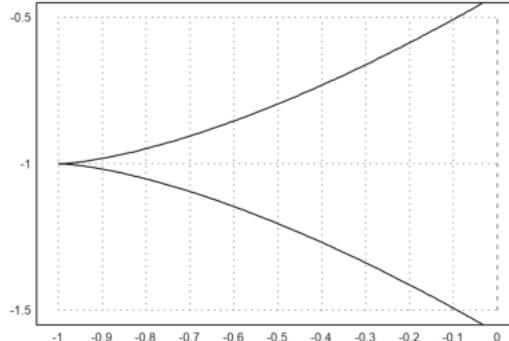
$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$

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
3 x \sqrt{9 x^2 + 4}

>$showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%)
// panjang kurva di atas

3 \int_0^{\frac{1}{\sqrt{3}}} x \sqrt{9 x^2 + 4} dx = 3 \left( \frac{7^{\frac{3}{2}}}{27} - \frac{8}{27} \right)

6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0 x^2 + 4.0} dx = 2.337835372767141
```

Sikloid

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar ke kanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah r . Posisi titik pusat lingkaran pada saat t adalah:

$$(rt, r).$$

Misalkan posisi titik pada lingkaran tersebut mula-mula $(0,0)$ dan posisinya pada saat t adalah:

$$(r(t - \sin(t)), r(1 - \cos(t))).$$

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika $t=0, t=\pi/2, t=r\pi$.

```
>x &= r*(t-sin(t))

[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,
 0.00326856311168871 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,
 0.0140605349340045 r, 0.0150344658876983 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,
0.004047266988005727 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, 0.028866202514797045 r, 0.031087578228935527 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,
0.3941798433563377 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([x,ex(2)],[y,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 : [0,1.66665833335744e-7*r-sin(1.66665833335744e-7*r),1.33330666692022e-6*r-sin(1.33330666692022e-6*r)
Error generated by error() command

adaptiveeval:
error(f$|" does not produce a real or column vector");
Try "trace errors" to inspect local variables after errors.
plot2d:
dw/n,dw/n^2,dw/n,args());

```

Berikut dihitung panjang lintasan untuk 1 putaran penuh. (Jangan salah menduga bahwa panjang lintasan 1 putaran penuh sama dengan keliling lingkaran!)

```
>ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds) // elemen panjang kurva sikloid
```

```

Maxima said:
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);

```

```

Error in:
ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds ...
^

```

```

>ds &= trigsimp(ds); $ds
>$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu putaran penuh
>integrate(mxm("ds"),0,2*pi) // hitung secara numerik
>romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik

```

Perhatikan, seperti terlihat pada gambar, panjang sikloid lebih besar daripada keliling lingkarannya, yakni:

2π .

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 menyimpung kurva di sekitar P). Pusat dan radius kelengkungan kurva di P adalah pusat dan radius lingkaran oskulasi. Kelengkungan adalah kebalikan dari radius kelengkungan:

$$\kappa = \frac{1}{R}$$

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar 2π sejauh $2\pi R$.) Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekivalen.

Definisi Kurvatur dengan Fungsi Parametrik Panjang Kurva

Setiap kurva diferensiabel dapat dinyatakan dengan persamaan parametrik terhadap panjang kurva s:

$$\gamma(s) = (x(s), y(s)),$$

dengan x dan y adalah fungsi riil yang diferensiabel, yang memenuhi:

$$\|\gamma'(s)\| = \sqrt{x'(s)^2 + y'(s)^2} = 1.$$

Ini berarti bahwa vektor singgung

$$\mathbf{T}(s) = (x'(s), y'(s))$$

memiliki norm 1 dan merupakan vektor singgung satuan.

Apabila kurvanya memiliki turunan kedua, artinya turunan kedua x dan y ada, maka $\mathbf{T}'(s)$ ada. Vektor ini merupakan normal kurva yang arahnya menuju pusat kurvatur, norm-nya merupakan nilai kurvatur (kelengkungan):

$$\begin{aligned}\mathbf{T}(s) &= \gamma'(s), \\ \mathbf{T}^2(s) &= 1 \text{ (konstanta)} \Rightarrow \mathbf{T}'(s) \cdot \mathbf{T}(s) = 0 \\ \kappa(s) &= \|\mathbf{T}'(s)\| = \|\gamma''(s)\| = \sqrt{x''(s)^2 + y''(s)^2}.\end{aligned}$$

Nilai

$$R(s) = \frac{1}{\kappa(s)}$$

disebut jari-jari (radius) kelengkungan kurva.

Bilangan riil

$$k(s) = \pm \kappa(s)$$

disebut nilai kelengkungan bertanda.

Contoh:

Akan ditentukan kurvatur lingkaran

$$x = r \cos t, \quad y = r \sin t.$$

```
>fx &= r*cos(t); fy &= r*sin(t);
>&assume(t>0,r>0); s &=integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s // elemen panjang kurva, panjang bu
```

```
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 parametrik terhadap s dengan substit
>k &= trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur lingkaran dengan menggunakan definis
```

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) // kelengkungan parabola
```

```
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 untuk a=b=c=1  
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) // kelengkungan parabola
```

Berikut kita gambar parabola tersebut beserta kurva kelengkungan, kurva jari-jari kelengkungan dan salah satu lingkaran oskulasi di titik puncak parabola. Perhatikan, puncak parabola dan jari-jari lingkaran oskulasi di puncak parabola adalah

$$(-1/2, 3/4), \quad 1/k(2) = 1/2,$$

sehingga pusat lingkaran oskulasi adalah (-1/2, 5/4).

```
>plot2d(["f(x)", "k(x)"], -2, 1, color=[blue, red]); plot2d("1/k(x)", -1.5, 1, color=green, >add); ...  
plot2d("-1/2+1/k(-1/2)*cos(x)", "5/4+1/k(-1/2)*sin(x)", xmin=0, xmax=2pi, >add, color=blue);
```

Untuk kurva yang dinyatakan dengan fungsi implisit

$$F(x, y) = 0$$

dengan turunan-turunan parsial

$$F_x = \frac{\partial F}{\partial x}, \quad F_y = \frac{\partial F}{\partial y}, \quad F_{xy} = \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial x} \right), \quad F_{xx} = \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial x} \right), \quad F_{yy} = \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial y} \right),$$

berlaku

$$F_x dx + F_y dy = 0 \quad \text{atau} \quad \frac{dy}{dx} = -\frac{F_x}{F_y},$$

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 &=diff(diff(F(x,y),x),y), Fyy &=diff(F
```

```
Maxima said:  
diff: second argument must be a variable; found errexpl  
-- 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(x) // kurvatur parabola tersebut
```

Hasilnya sama dengan sebelumnya yang menggunakan persamaan parabola biasa.

```
>
```

Barisan dan Deret

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen berurutan (menggunakan titik dua ":");
- menggunakan perintah "sequence" dan rumus barisan (suku ke $-n$);
- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create_list" atau "makelist" untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturut-turut

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana
```

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

```
>1:2:30
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
```

Iterasi dan Barisan

EMT menyediakan fungsi iterate("g(x)", x0, n) untuk melakukan iterasi

$$x_{k+1} = g(x_k), x_0 = x_0, k = 1, 2, 3, \dots, n.$$

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
```

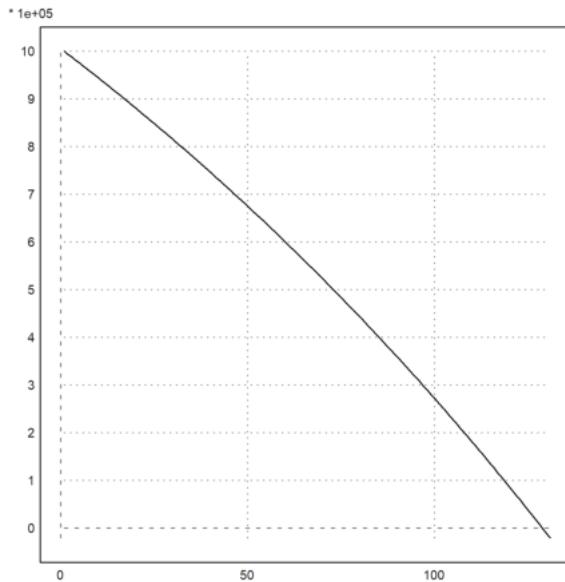
```

1407.1
1477.46
1551.33
1628.89

```

Contoh berikutnya memperlihatkan bahwa 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 tahun akan habis diambil oleh bank!

```
>r=0.005; plot2d(iterate("(1+0.8*r)*x-10000",1000000,n=130)):
```



Silakan Anda coba-coba, dengan tabungan minimal berapa agar tidak akan habis diambil oleh bank dengan ketentuan bunga dan biaya administrasi seperti di atas.

Berikut adalah perhitungan minimal tabungan agar aman di bank dengan bunga sebesar r dan biaya administrasi a , pajak bunga 20%.

```
>$solve(0.8*r*A-a,A), $% with [r=0.005, a=10]
```

$$\left[A = \frac{5a}{4r} \right]$$

$$[A = 2500.0]$$

Berikut didefinisikan fungsi untuk menghitung saldo tabungan, kemudian dilakukan iterasi.

```

>function saldo(x,r,a) := round((1+0.8*r)*x-a,2);
>iterate({{"saldo",0.005,10}},1000,n=6)

[1000, 994, 987.98, 981.93, 975.86, 969.76, 963.64]

>iterate({{"saldo",0.005,10}},2000,n=6)

[2000, 1998, 1995.99, 1993.97, 1991.95, 1989.92, 1987.88]

>iterate({{"saldo",0.005,10}},2500,n=6)

[2500, 2500, 2500, 2500, 2500, 2500, 2500]

```

Tabungan senilai 2,5 juta akan aman dan tidak akan berubah nilai (jika tidak ada penarikan), sedangkan jika tabungan awal kurang dari 2,5 juta, lama kelamaan akan berkurang meskipun tidak pernah dilakukan penarikan uang tabungan.

```
>iterate({{"saldo",0.005,10}},3000,n=6)

[3000, 3002, 3004.01, 3006.03, 3008.05, 3010.08, 3012.12]
```

Tabungan yang lebih dari 2,5 juta baru akan bertambah jika tidak ada penarikan.

Untuk barisan yang lebih kompleks dapat digunakan fungsi "sequence()". Fungsi ini menghitung nilai-nilai $x[n]$ dari semua nilai sebelumnya, $x[1], \dots, x[n-1]$ yang diketahui.
Berikut adalah contoh barisan Fibonacci.

$$x_n = x_{n-1} + x_{n-2}, \quad x_1 = 1, \quad x_2 = 1$$

```
>sequence("x[n-1]+x[n-2]",[1,1],15)

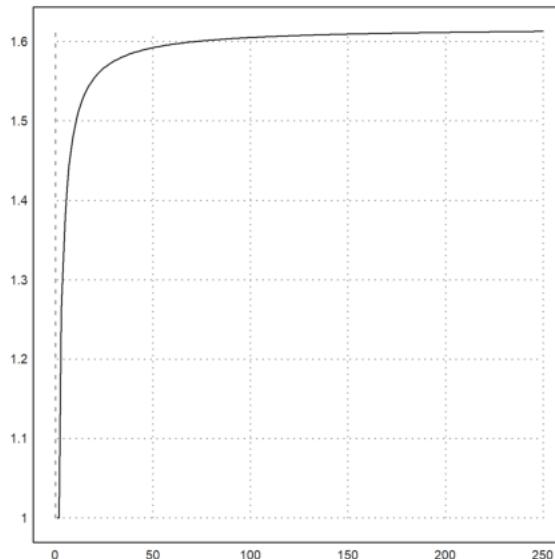
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]
```

Barisan Fibonacci memiliki banyak sifat menarik, salah satunya adalah akar pangkat ke-n suku ke-n akan konvergen ke pecahan emas:

```
>$' (1+sqrt(5))/2=float((1+sqrt(5))/2)
```

$$\frac{\sqrt{5} + 1}{2} = 1.618033988749895$$

```
>plot2d(sequence("x[n-1]+x[n-2]", [1,1], 250)^(1/(1:250))):
```



Barisan yang sama juga dapat dihasilkan dengan menggunakan loop.

```
>x=ones(500); for k=3 to 500; x[k]=x[k-1]+x[k-2]; end;
```

Rekursi dapat dilakukan dengan menggunakan rumus yang tergantung pada semua elemen sebelumnya. Pada contoh berikut, elemen ke-n merupakan jumlah (n-1) elemen sebelumnya, dimulai dengan 1 (elemen ke-1). Jelas, nilai elemen ke-n adalah 2^{n-2} , untuk $n=2, 4, 5, \dots$

```
>sequence("sum(x)",1,10)
```

```
[1, 1, 2, 4, 8, 16, 32, 64, 128, 256]
```

Selain menggunakan ekspresi dalam x dan n, kita juga dapat menggunakan fungsi.

Pada contoh berikut, digunakan iterasi

dengan A suatu matriks 2x2, dan setiap x[n] merupakan matriks/vektor 2x1.

```
>A=[1,1;1,2]; function suku(x,n) := A.x[,n-1]
>sequence("suku",[1;1],6)
```

Real 2 x 6 matrix

1	2	5	13	...
1	3	8	21	...

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "matrixpower()". Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak $\log_2(n)$.

```
>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:

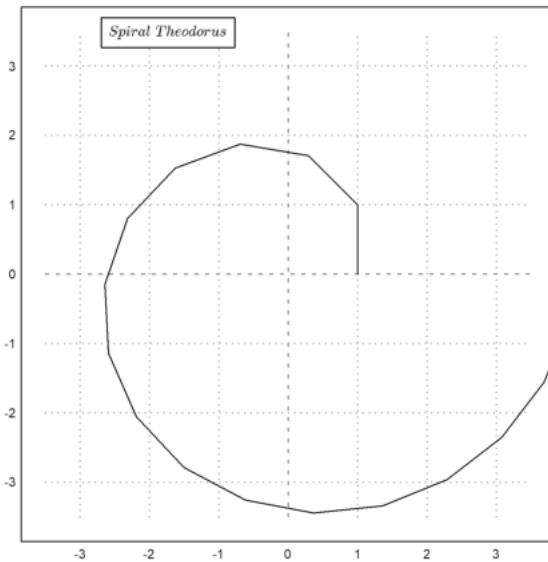
$$x_n = \left(1 + \frac{i}{\sqrt{n-1}}\right) x_{n-1}, \quad x_1 = 1,$$

yang menghasilkan barisan bilangan kompleks.

```
>function g(n) := 1+I/sqrt(n)
```

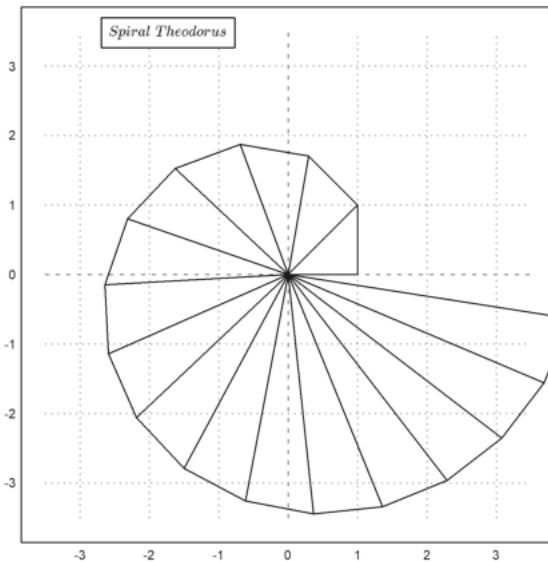
Rekursinya dapat dijalankan sebanyak 17 untuk menghasilkan barisan 17 bilangan kompleks, kemudian digambar bilangan-bilangan kompleksnya.

```
>x=sequence("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox(latex("Spiral \ Theodorus"),0.4):
```



Selanjutnya dihubungkan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end:
```

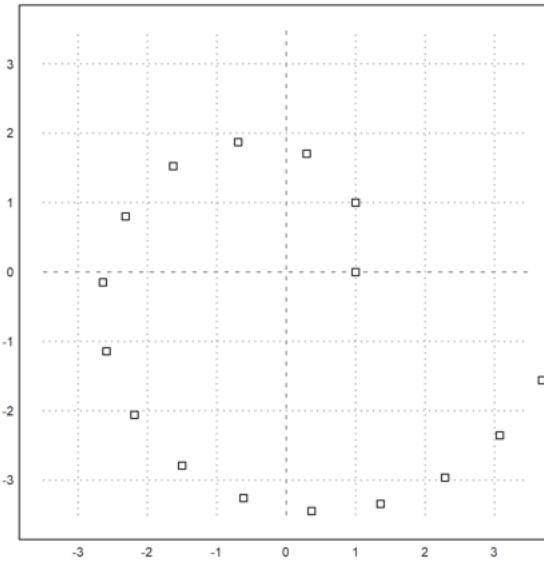


Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memerlukan indeks dan bilangan kompleks. Dalam hal ini digunakan vektor kolom pada bidang.

```
>function gstep (v) ...
w=[-v[2];v[1]];
return v+w/norm(w);
endfunction
```

Jika dilakukan iterasi 16 kali dimulai dari [1;0] akan didapatkan matriks yang memuat vektor-vektor dari setiap iterasi.

```
>x=iterate("gstep",[1;0],16); plot2d(x[1],x[2],r=3.5,>points):
```



Kekonvergenan

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```
>iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1.
```

0.739085133216

Iterasi tersebut konvergen ke penyelesaian persamaan

$$x = \cos(x).$$

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.

```
>hasil := iterate("cos(x)",~1,2~) //iterasi x(n+1)=cos(x(n)), dengan interval awal (1, 2)
```

~0.739085133211,0.7390851332133~

Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan $x=\cos(x)$.

```
>h=expand(hasil,100), cos(h) << h
```

~0.73908513309,0.73908513333~
1

Iterasi juga dapat digunakan pada fungsi yang didefinisikan.

```
>function f(x) := (x+2/x)/2
```

Iterasi $x(n+1)=f(x(n))$ akan konvergen ke akar kuadrat 2.

```
>iterate("f",2), sqrt(2)
```

1.41421356237
1.41421356237

Jika pada perintah iterate diberikan tambahan parameter n, maka hasil iterasinya akan ditampilkan mulai dari iterasi pertama sampai ke-n.

```
>iterate("f",2,5)
```

[2, 1.5, 1.41667, 1.41422, 1.41421, 1.41421]

Untuk iterasi ini tidak dapat dilakukan terhadap interval.

```
>niterate("f",~1,2~,5)
```

[~1,2~, ~1,2~, ~1,2~, ~1,2~, ~1,2~, ~1,2~]

Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya adalah perhitungan dengan interval bersifat terlalu longgar. Untuk meingkatkan perhitungan pada ekspresi dapat digunakan pembagian intervalnya, menggunakan fungsi ieval().

```
>function s(x) := ieval("(x+2/x)/2",x,10)
```

Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal, dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang memuat akar persamaan:

$$x = \frac{1}{2} \left(x + \frac{2}{x} \right).$$

Satu-satunya solusi adalah

$$x = \sqrt{2}.$$

```
>iterate("s",~1,2~)
```

```
Variable s not found!
Use global variables or parameters for string evaluation.
Error in expression: s
Try "trace errors" to inspect local variables after errors.
iterate:
x=f$(x,args());
```

Fungsi "iterate()" juga dapat bekerja pada vektor. Berikut adalah contoh fungsi vektor, yang menghasilkan rata-rata aritmetika dan rata-rata geometri.

$$(a_{n+1}, b_{n+1}) = \left(\frac{a_n + b_n}{2}, \sqrt{a_n b_n} \right)$$

Iterasi ke-n disimpan pada vektor kolom x[n].

```
>function g(x) := [(x[1]+x[2])/2,sqrt(x[1]*x[2])]
```

Iterasi dengan menggunakan fungsi tersebut akan konvergen ke rata-rata aritmetika dan geometri dari nilai-nilai awal.

```
>iterate("g",[1;5])
```

```
2.60401
2.60401
```

Hasil tersebut konvergen agak cepat, seperti kita cek sebagai berikut.

```
>iterate("g",[1;5],4)
```

1	3	2.61803	2.60403	2.60401
5	2.23607	2.59002	2.60399	2.60401

Iterasi pada interval dapat dilakukan dan stabil, namun tidak menunjukkan bahwa limitnya pada batas-batas yang dihitung.

```
>iterate("g",[~1~;~5~],4)
```

```
Interval 2 x 5 matrix
~0.99999999999999778,1.0000000000000022~ ...
~4.999999999999911,5.0000000000000089~ ...
```

Iterasi berikut konvergen sangat lambat.

$$x_{n+1} = \sqrt{x_n}.$$

```
>iterate("sqrt(x)",2,10)
```

```
[2, 1.41421, 1.18921, 1.09051, 1.04427, 1.0219, 1.01089,
1.00543, 1.00271, 1.00135, 1.00068]
```

Kekonvergenan iterasi tersebut dapat dipercepat dengan percepatan Steffenson:

```
>steffenson("sqrt(x)",2,10)
```

```
[1.04888, 1.00028, 1, 1]
```

Iterasi menggunakan Loop yang ditulis Langsung

Berikut adalah beberapa contoh penggunaan loop untuk melakukan iterasi yang ditulis langsung pada baris perintah.

```
>x=2; repeat x=(x+2/x)/2; until x^2~=2; end; x,
```

```
1.41421356237
```

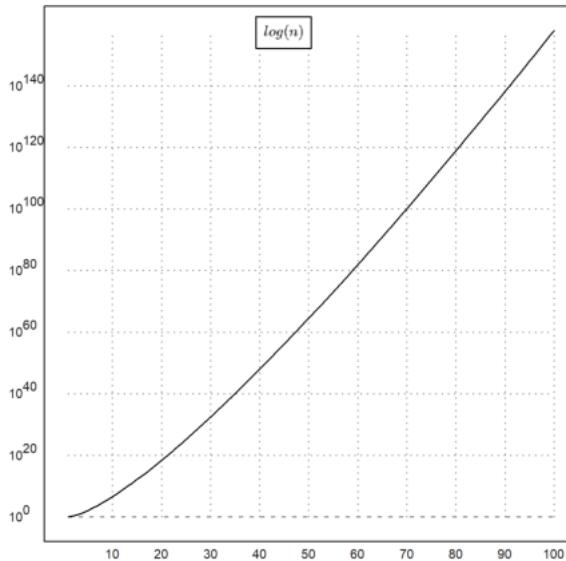
Penggabungan matriks menggunakan tanda "[" dapat digunakan untuk menyimpan semua hasil iterasi.

```
>v=[1]; for i=2 to 8; v=v|(v[i-1]*i); end; v,
```

```
[1, 2, 6, 24, 120, 720, 5040, 40320]
```

hasil iterasi juga dapat disimpan pada vektor yang sudah ada.

```
>v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]*i; end; ...
plot2d(v,logplot=1); textbox(latex(&log(n)),x=0.5):
```



```
>A =[0.5,0.2;0.7,0.1]; b=[2;2]; ...
x=[1;1]; repeat xnew=A.x-b; until all(xnew~≈x); x=xnew; end; ...
x,
```

```
-7.09677
-7.74194
```

Iterasi di dalam Fungsi

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.

Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

```
>function map hiter(f$,x0) ...
x=x0;
maxiter=0;
repeat
  xnew=f$(x);
  maxiter=maxiter+1;
  until xnew~≈x;
  x=xnew;
end;
return maxiter;
endfunction
```

Misalnya, berikut adalah iterasi untuk mendapatkan hampiran akar kuadrat 2, cukup cepat, konvergen pada iterasi ke-5, jika dimulai dari hampiran awal 2.

```
>hiter("(x+2/x)/2",2)
```

5

Karena fungsinya didefinisikan menggunakan "map". maka nilai awalnya dapat berupa vektor.

```
>x=1.5:0.1:10; hasil=hiter("(x+2/x)/2",x); ...
plot2d(x,hasil):
```

Dari gambar di atas terlihat bahwa kekonvergenan iterasinya semakin lambat, untuk nilai awal semakin besar, namun penambahan tidak kontinu. Kita dapat menemukan kapan maksimum iterasinya bertambah.

```
>hasil[1:10]
```

```
Row index 2 out of bounds!
Error in:
  hasil[1:10] ...  
^
```

```
>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 (aslinya 10 kali).

```
>function iterasi(f$,x,n=10) ...
loop 1 to n; x=f$(x); end;
return x;
endfunction
```

Kita mulai dengan menentukan titik-titik grid pada bidang kompleksnya.

```
>r=1.5; x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z);
```

Berikut adalah akar-akar polinomial di atas.

```
>z=&solve(p) ()

[ -0.5+0.866025i, -0.5-0.866025i, 1+0i ]
```

Untuk menggambar hasil iterasinya, dihitung jarak dari hasil iterasi ke-10 ke masing-masing akar, kemudian digunakan untuk menghitung warna yang akan digambar, yang menunjukkan limit untuk masing-masing nilai awal.

Fungsi plotrgb() menggunakan jendela gambar terkini untuk menggambar warna RGB sebagai matriks.

```
>C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1)); ...
plot2d(None,-r,r,-r,r); plotrgb(C);
```

```
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 berpangkat terkecil
```

```
true
```

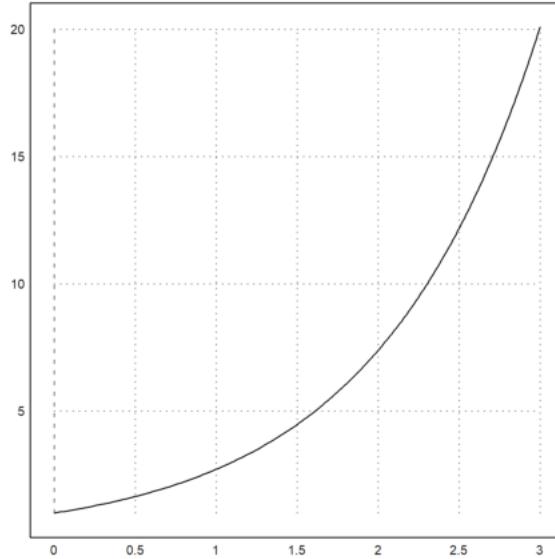
```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Taylor untuk e^x
```

```
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 hampiran fungsi tersebut
```

```
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.

```
>$deret[3]
>plot2d(["exp(x)",&deret[1],&deret[2],&deret[3]],0,3,color=1:4):
>$sum(sin(k*x)/k,k,1,5)
```

Berikut adalah cara menggambar kurva

$$y = \sin(x) + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \dots$$

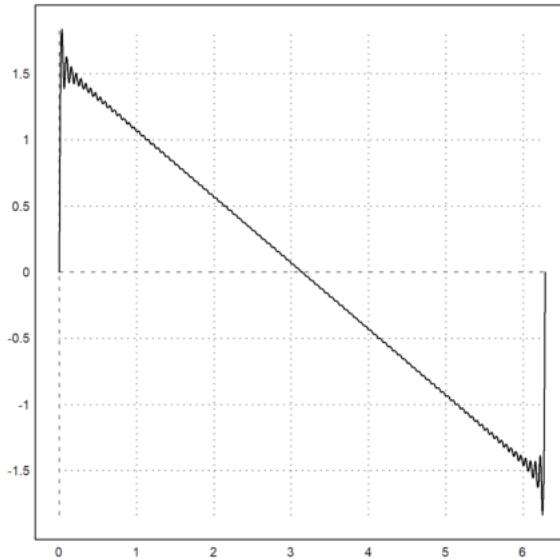
```
>plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi):
```

```
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^n 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*x,1.33330666692022e-6*x,4.499797504338432e-6*x,1.066581336583994e-
-- an error. To debug this try: debugmode(true);

%mxmtable:
return mxm("@expr,@var=@value")();
Try "trace errors" to inspect local variables after errors.
mxmtable:
y[#,1]=%mxmtable(expr,var,x[#]);

>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:menghitung deret secara simbolik

$$\sum_{k=1}^n k = \frac{n(1+n)}{2}$$


>$'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))

$$\sum_{k=0}^{\infty} \frac{1}{k+3^k} = \sum_{k=0}^{\infty} \frac{1}{k+3^k}$$

```

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai ekspresi

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$


>$'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))

$$\sum_{k=1}^{\infty} \frac{(-1)^{-1+k}}{k} = -\sum_{x=1}^{\infty} \frac{(-1)^x}{x}$$

```

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum((-1)^k/(2*k-1), k, 1, inf) = factor(ev(sum((-1)^k/(2*k-1), k, 1, inf),simpsum=true))

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{-1+2k} = \sum_{k=1}^{\infty} \frac{(-1)^k}{-1+2k}$$

```

```
>$ev(sum(1/n!, n, 0, inf),simpsum=true)
```

$$\sum_{n=0}^{\infty} \frac{1}{n!}$$

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
>&assume(abs(x)<1); $'sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forget(abs(x)<1);
```

```

Answering "Is -94914474571+15819*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)
```

```
>$limit(sum(x^k/k!,k,0,n),n,inf)
```

```
>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 diferensiable sampai tak hingga di sekitar $x=a$ adalah:

$$f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^k f^{(k)}(a)}{k!}.$$

```
>$'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai suku ke-11
```

```

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 ...

```

Tugas Individu Pekan11-12_Muhammad Lutfi Ramadhan_23030630021

Nama : Muhammad Lutfi Ramadhan
Kelas : Matematika B 2023
NIM : 23030630021

Visualisasi dan Perhitungan Geometri dengan EMT

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan geometri tersebut disimpan di dalam file program "geometry.e", sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

```
>load geometry
```

Numerical and symbolic geometry.

Fungsi-fungsi Geometri

Fungsi-fungsi untuk Menggambar Objek Geometri:

```
defaultd:=textheight()*1.5: nilai asli untuk parameter d
setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang koordinat
setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas sumbu-x dan y adalah -r sd r
plotPoint (P, "P"): menggambar titik P dan diberi label "P"
plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label "AB" sejauh d
plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d
plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"
plotLabel (label, P, V, d): menuliskan label pada posisi P
```

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

```
turn(v, phi): memutar vektor v sejauh phi
turnLeft(v): memutar vektor v ke kiri
turnRight(v): memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang 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
angleBisection(A, B, C): garis bagi sudut <ABC
circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r
getCircleCenter(c): pusat lingkaran c
getCircleRadius(c): jari-jari lingkaran c
circleThrough(A,B,C): lingkaran melalui A, B, C
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan lingkaran c
circleCircleIntersections (c1, c2): titik potong lingkaran c1 dan c2
planeThrough(A, B, C): bidang melalui titik A, B, C
```

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

```
getLineEquation (g,x,y): persamaan garis g dinyatakan dalam x dan y
getHesseForm (g,x,y,A): bentuk Hesse garis g dinyatakan dalam x dan y dengan titik A pada sisi positif (kanan/atasi) garis
quad(A,B): kuadrat jarak AB
spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni sin(alpha)^2 dengan alpha sudut yang menghadap sisi a.
crosslaw(a,b,c,sa): persamaan 3 quads dan 1 spread pada segitiga dengan panjang sisi a, b, c
triplespread(sa,sb,sc): persamaan 3 spread sa,sb,sc yang memebntuk suatu segitiga
doublespread(sa): Spread sudut rangkap Spread 2*phi, dengan sa=sin(phi)^2 spread a.
```

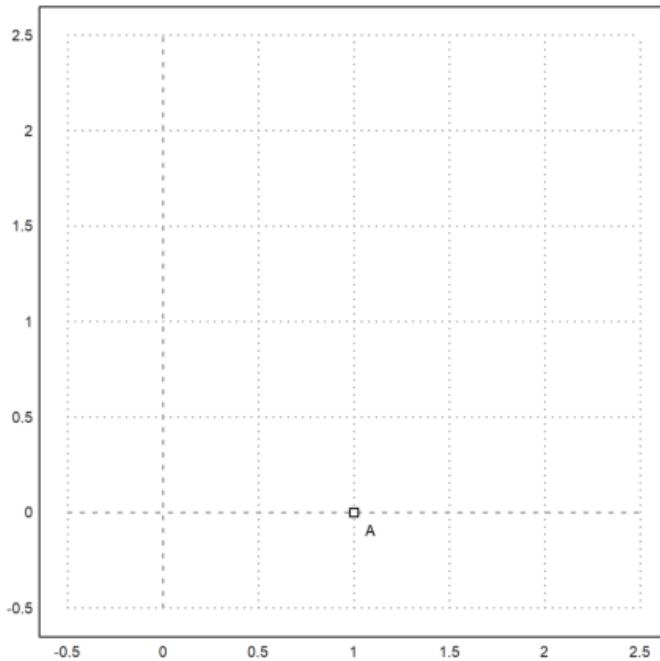
Contoh 1: Luas, Lingkaran Luar, Lingkaran Dalam Segitiga

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.

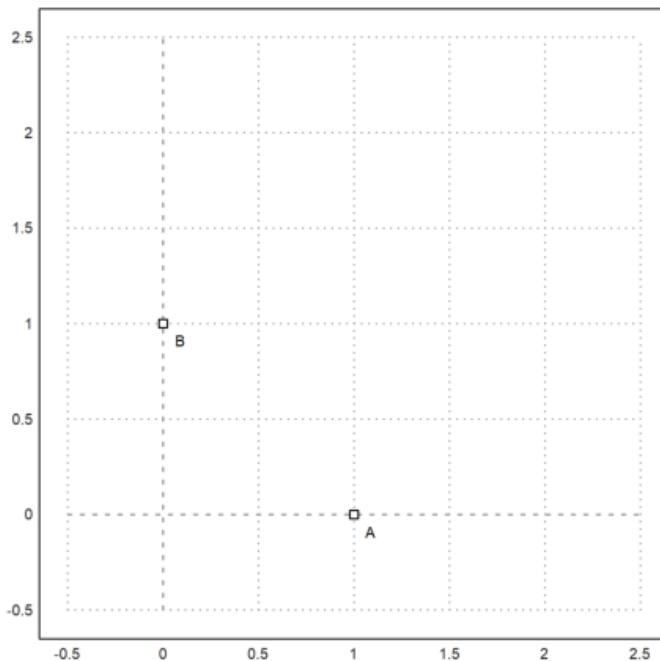
```
>setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
```

Sekarang tentukan tiga titik dan gambarkan.

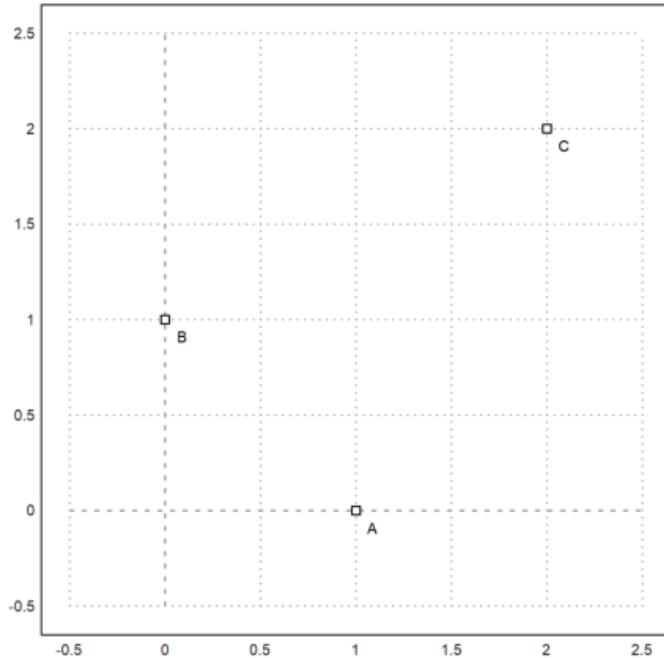
```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
```



```
>B=[0,1]; plotPoint(B,"B");
```

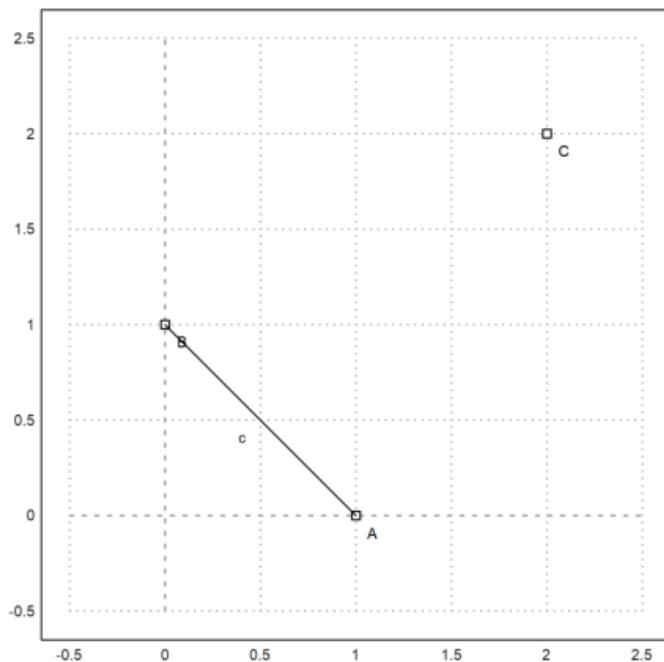


```
>C=[2,2]; plotPoint(C,"C");
```

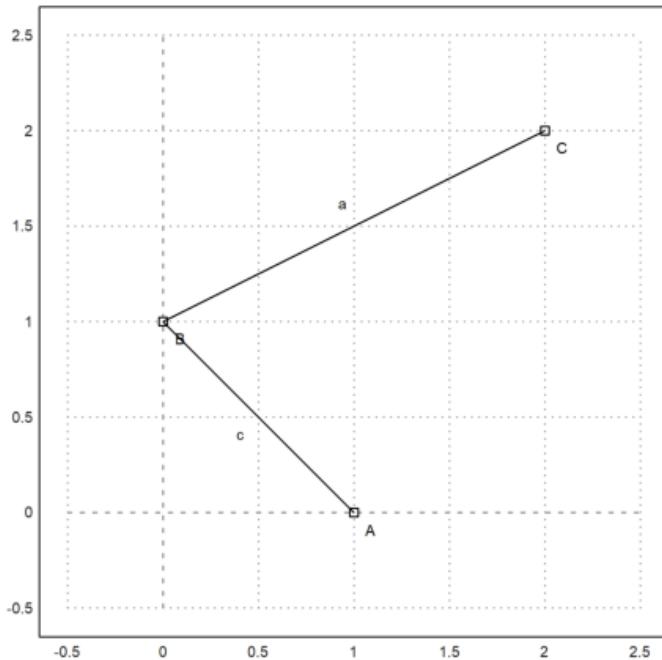


Lalu tiga segmen.

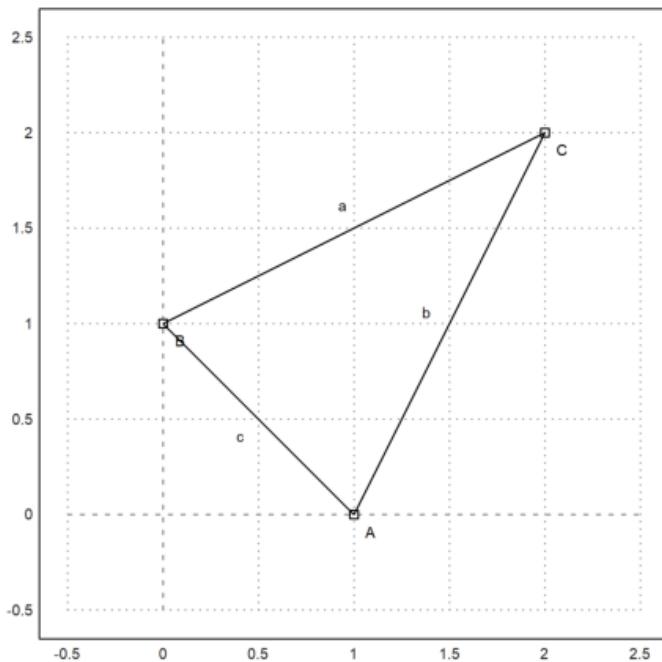
```
>plotSegment(A,B,"c") : // c=AB
```



```
>plotSegment(B,C,"a") : // a=BC
```



```
>plotSegment(A,C,"b"): // b=AC
```



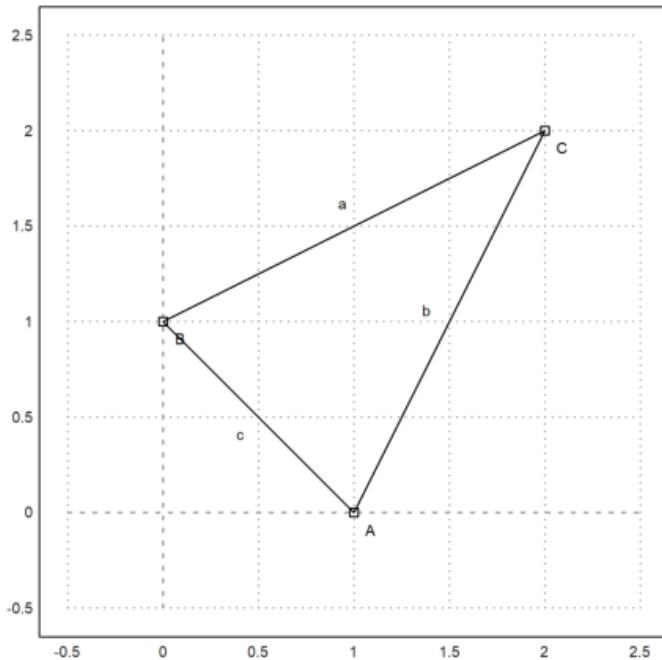
Fungsi-fungsi geometri mencakup fungsi untuk membuat garis dan lingkaran. Format untuk garis adalah [a,b,c], yang mewakili garis dengan persamaan $ax+by=c$.

```
>lineThrough(B,C) // garis yang melalui B dan C
```

```
[-1, 2, 2]
```

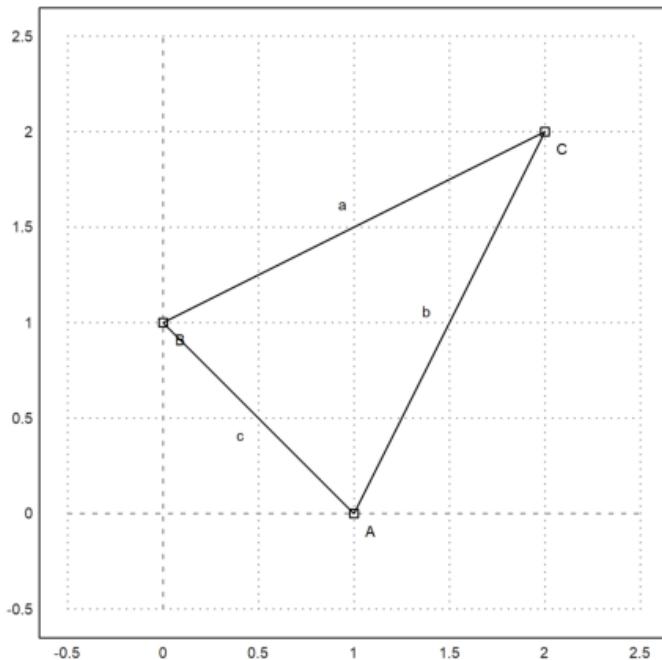
Hitung garis tegak lurus melalui A pada BC.

```
>h=perpendicular(A,lineThrough(B,C)): // garis h tegak lurus BC melalui A
```



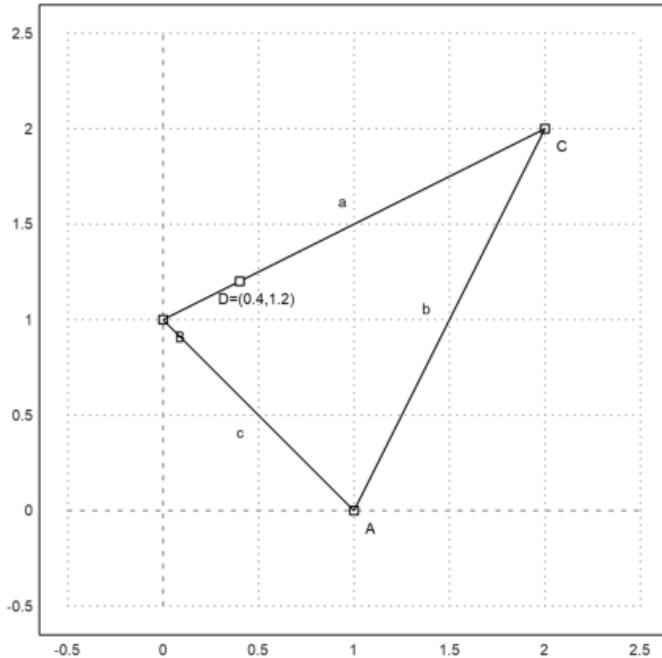
Dan titik potongnya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)): // D adalah titik potong h dan BC
```

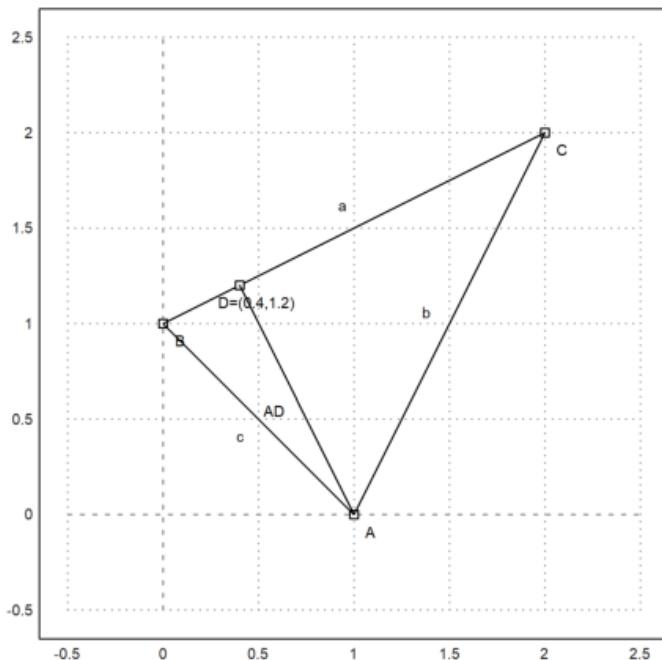


Gambarkan itu.

```
>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

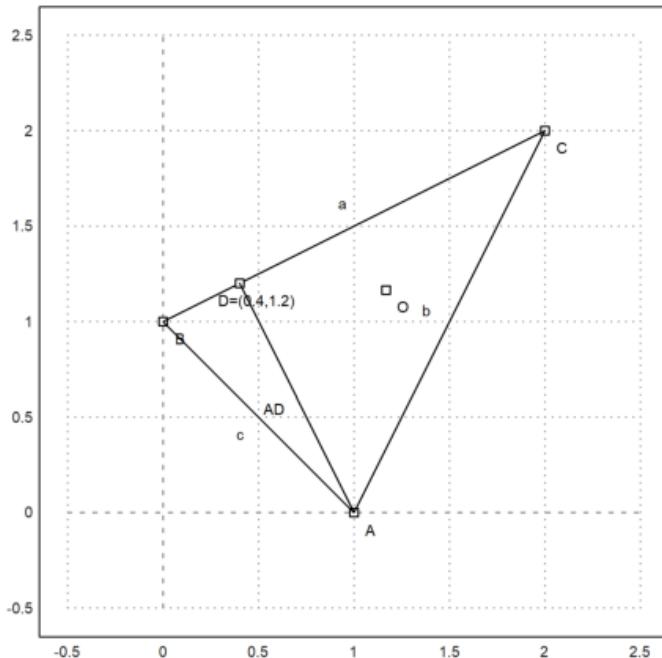
Sudut di C.

```
>degprint(computeAngle(B,C,A))
```

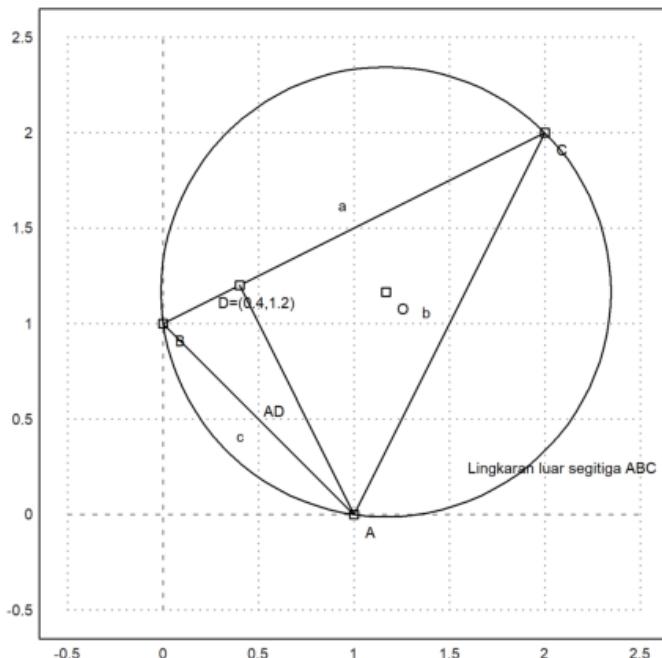
36°52'11.63''

Sekarang lingkaran juring dari segitiga.

```
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC  
>R=getCircleRadius(c); // jari2 lingkaran luar  
>O=getCircleCenter(c); // titik pusat lingkaran c  
>plotPoint(O,"O"); // gambar titik "O"
```



```
>plotCircle(c,"Lingkaran luar segitiga ABC"):
```



Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

>O, R

```
[1.16667, 1.16667]  
1.17851130198
```

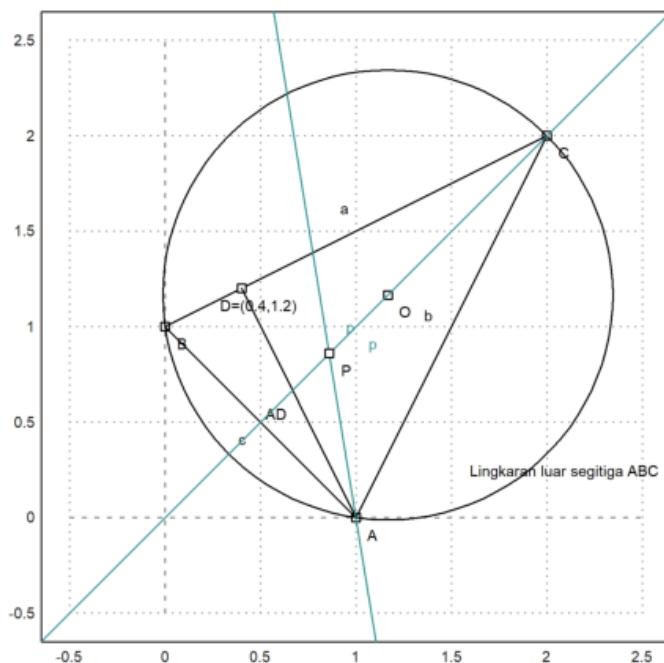
Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

```
>l=angleBisector(A,C,B); // garis bagi <ACB  
>g=angleBisector(C,A,B); // garis bagi <CAB  
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

```
[0.86038, 0.86038]
```

Tambahkan semuanya ke dalam plot.

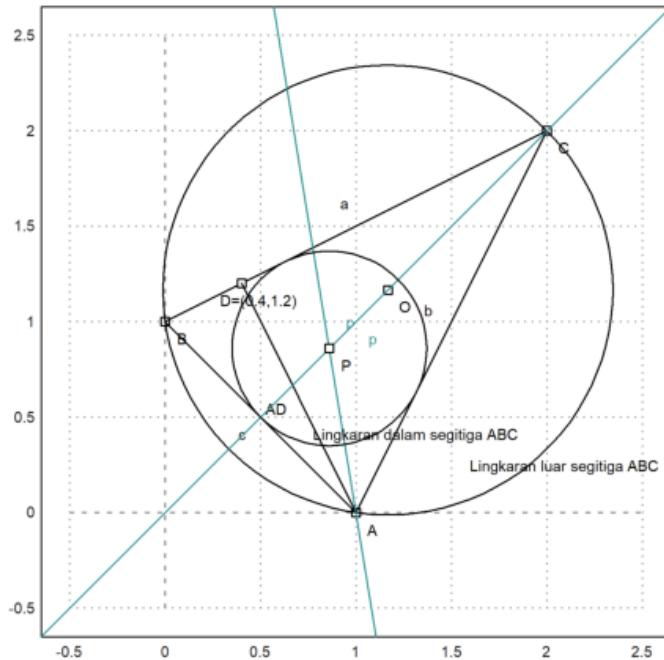
```
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut  
>plotPoint(P,"P"); // gambar titik potongnya
```



```
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

```
0.509653732104
```

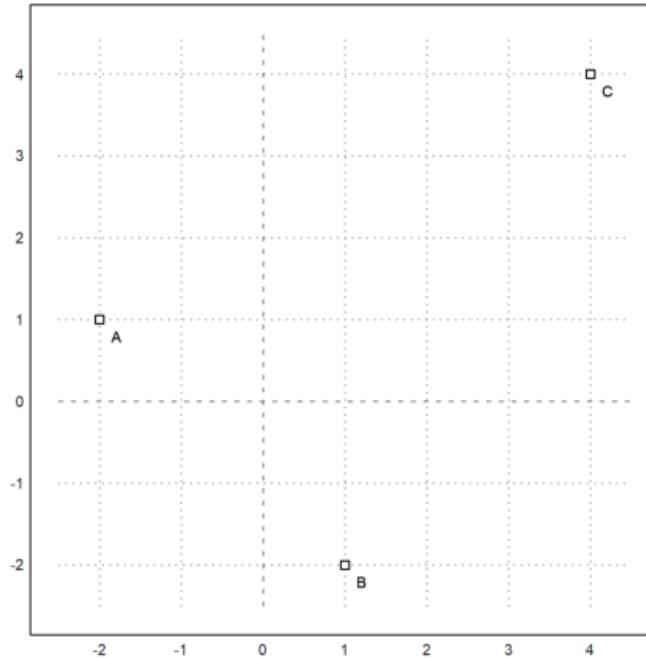
```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"); // gambar lingkaran dalam
```



Latihan

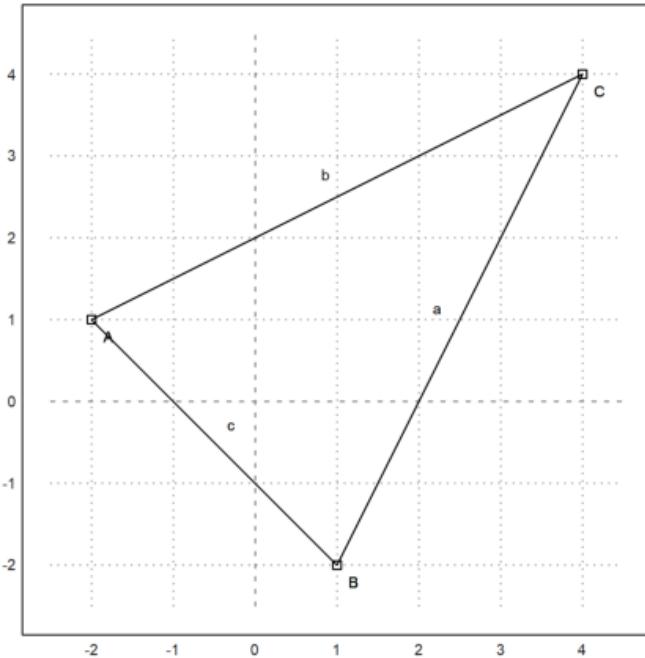
1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

```
>setPlotRange(-2.5,4.5,-2.5,4.5);
>A=[-2,1]; plotPoint(A,"A");
>B=[1,-2]; plotPoint(B,"B");
>C=[4,4]; plotPoint(C,"C");
```



2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. Merupakan segitiga apakah itu?

```
>plotSegment(A,B,"c")
>plotSegment(B,C,"a")
>plotSegment(A,C,"b")
>aspect(1);
```



3. Hitung luas segitiga tersebut.

```
>areaTriangle(A,B,C)
```

13.5

4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

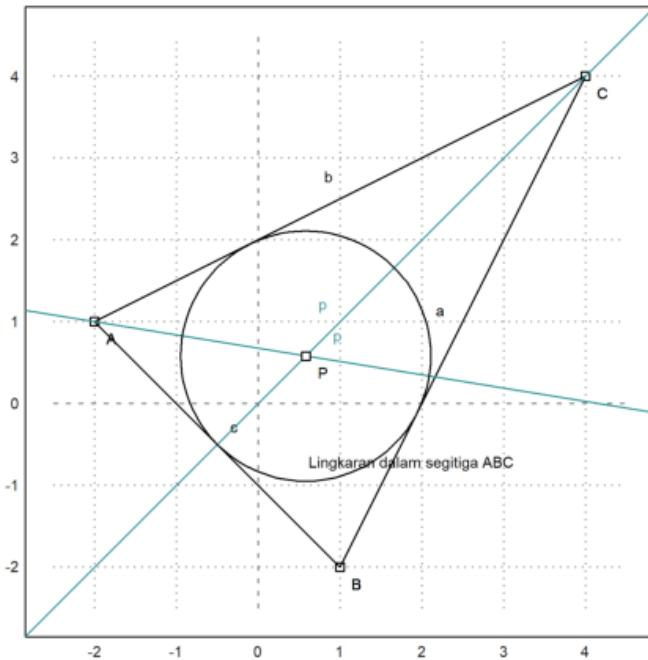
```
>l=angleBisector(A,C,B);
>g=angleBisector(C,A,B);
>P=lineIntersection(l,g)
```

[0.581139, 0.581139]

```
>color(5); plotLine(l); plotLine(g); color(1);
>plotPoint(P,"P");
>r=norm(P-projectToLine(P,lineThrough(A,B)))
```

1.52896119631

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC");
```

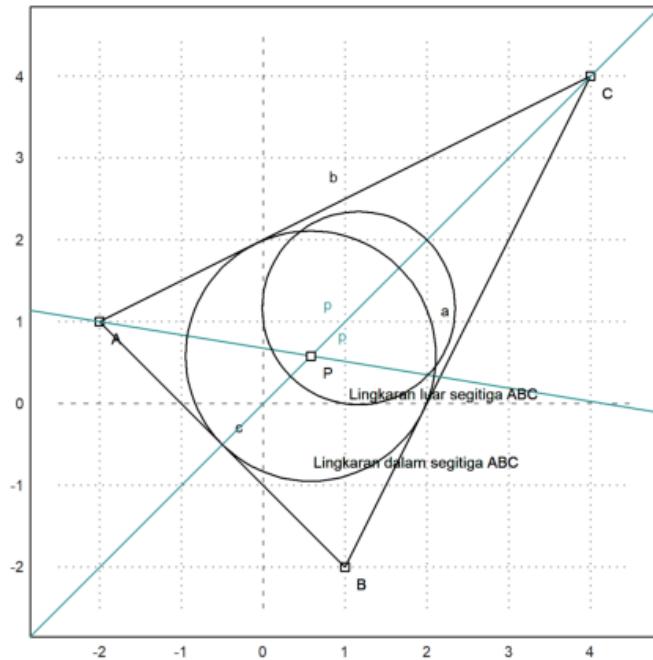


5. Gambar jari-jari lingkaran dalam.

```
>r=norm(P-projectToLine(P,lineThrough(A,B)))
```

1.52896119631

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):
```



Contoh 2: Geometri Simbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometry.e menyediakan fungsi yang sama (dan lebih banyak) di Maxima. Namun, sekarang kita dapat menggunakan perhitungan simbolik.

```
>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 perhitungan simbolik.

```
>c &= lineThrough(B,C) // c=BC
```

[- 1, 2, 2]

Kita dapat dengan mudah mendapatkan persamaan untuk sebuah garis.

```
>$getLineEquation(c,x,y), $solve(% ,y) | expand // persamaan garis c
2 y - x = 2
[ y = x / 2 + 1 ]

>$getLineEquation(lineThrough([x1,y1],[x2,y2]),x,y), $solve(% ,y) // persamaan garis melalui(x1, y1)
x (y1 - y2) + (x2 - x1) y = x1 (y1 - y2) + (x2 - x1) y1
[ y = -(x1 - x) y2 - (x - x2) y1 / x2 - x1 ]

>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
(x1 - 1) y - x y1 = -y1

>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
[2, 1, 2]

>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
[ 2 / 5, 6 / 5 ]

>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
[ 2 / 5, 6 / 5 ]

>$distance(A,Q) // jarak AQ
3 / sqrt(5)

>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
[ 7 / 6, 7 / 6, 5 / (3 * sqrt(2)) ]

>r=>getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
5 / 3 * sqrt(2)
1.178511301977579

>$computeAngle(A,C,B) // nilai <ACB
arccos(4 / 5)

>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
y = x

>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 garis bagi s
```

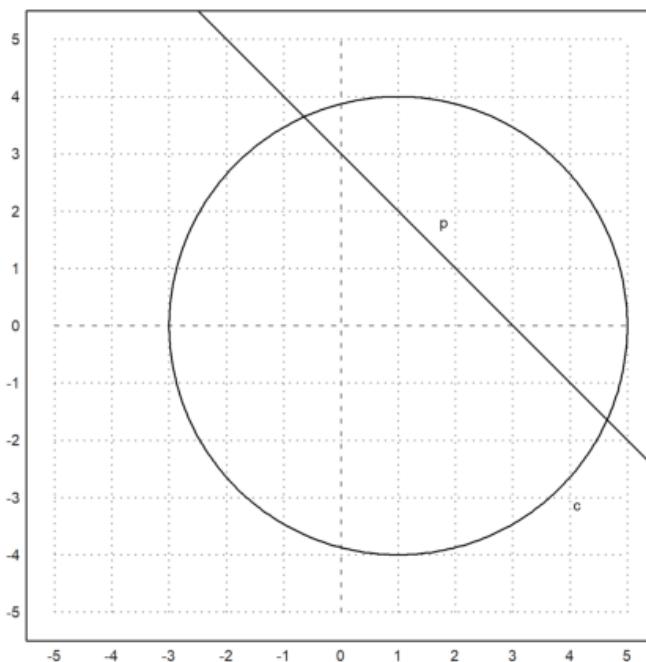
$$\left[\frac{\sqrt{2}\sqrt{5}+2}{6}, \frac{\sqrt{2}\sqrt{5}+2}{6} \right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
[0.86038, 0.86038]
```

Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga dapat 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):
```

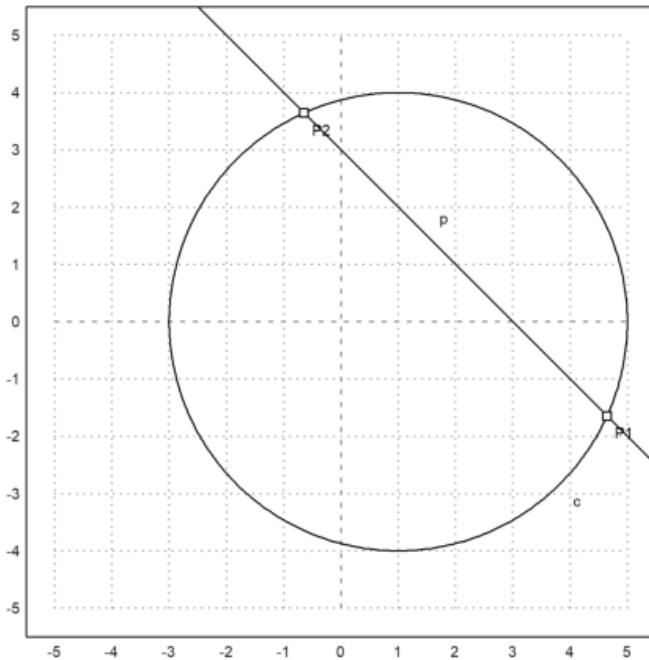


Pertemuan sebuah garis dengan lingkaran menghasilkan dua titik dan jumlah titik perpotongan.

```
>{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 di Maxima.

```
>c &= circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

```
[1, 0, 4]
```

```
>l &= lineThrough(B,C) // garis l melalui B dan C
```

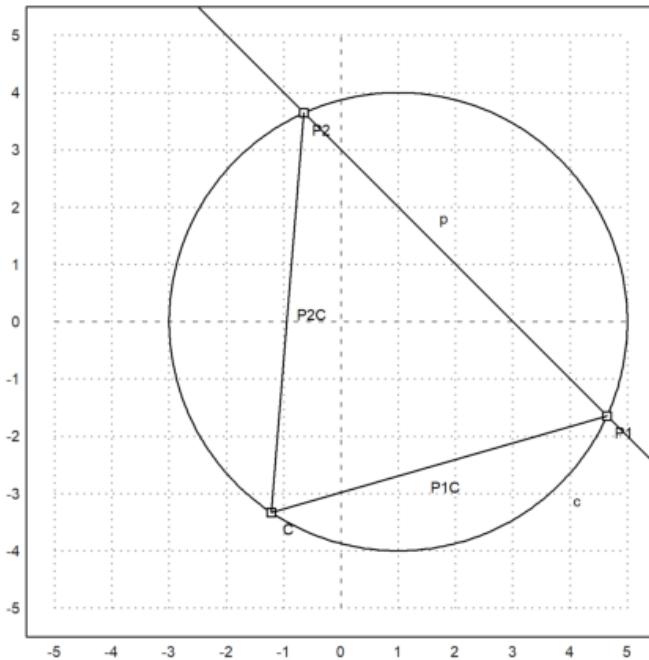
```
[1, 1, 3]
```

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan garis l
```

$$\left[\left[\sqrt{7} + 2, 1 - \sqrt{7} \right], \left[2 - \sqrt{7}, \sqrt{7} + 1 \right] \right]$$

Akan ditunjukkan bahwa sudut-sudut yang menghadap bsumur yang sama adalah sama besar.

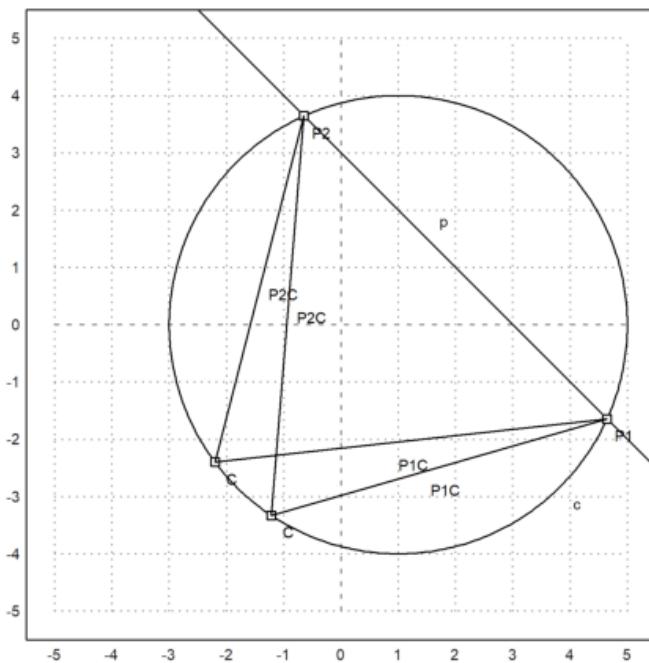
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C):
```



```
>degprint(computeAngle(P1,C,P2))
```

$69^\circ 17' 42.68''$

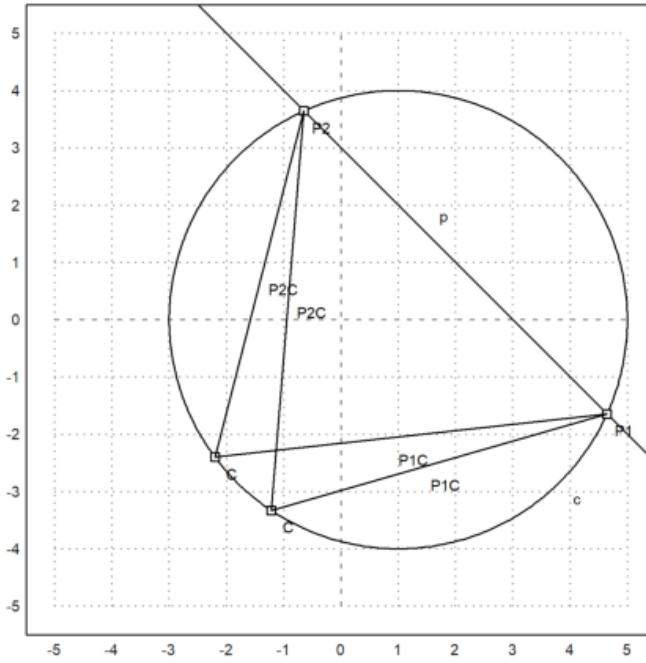
```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C):
```



```
>degprint(computeAngle(P1,C,P2))
```

$69^\circ 17' 42.68''$

```
>insimng;
```

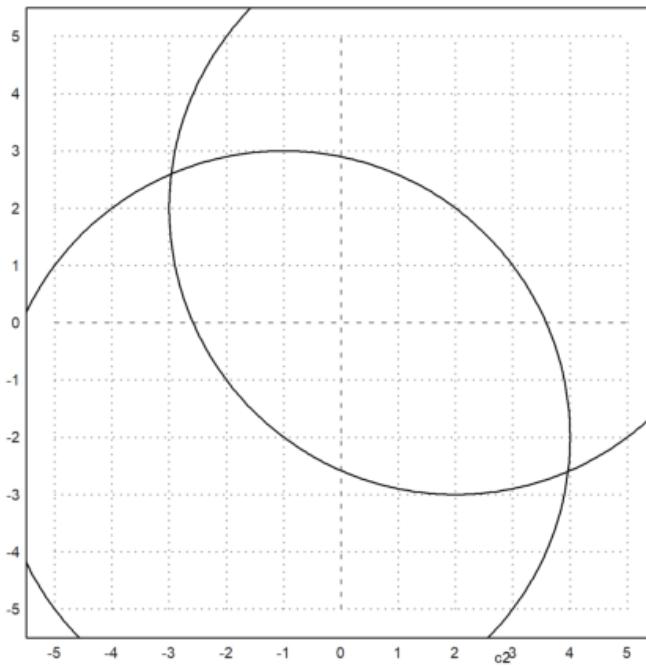


Garis Sumbu

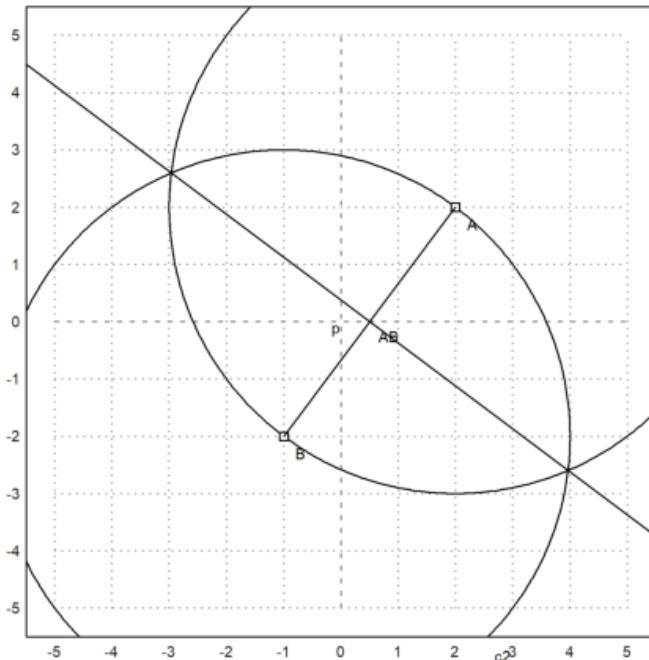
Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B.
2. Gambar lingkaran dengan pusat B melalui A.
3. Tarik garis melalui kedua titik potong kedua lingkaran tersebut. Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

```
>A=[2,2]; B=[-1,-2];
>c1=circleWithCenter(A,distance(A,B));
>c2=circleWithCenter(B,distance(A,B));
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(5); plotCircle(c1); plotCircle(c2);
```



```
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l);
```



Selanjutnya, kita 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 perpotongan cukup rumit. Namun, 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 garis 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:

$$L = \frac{1}{4} \sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}$$

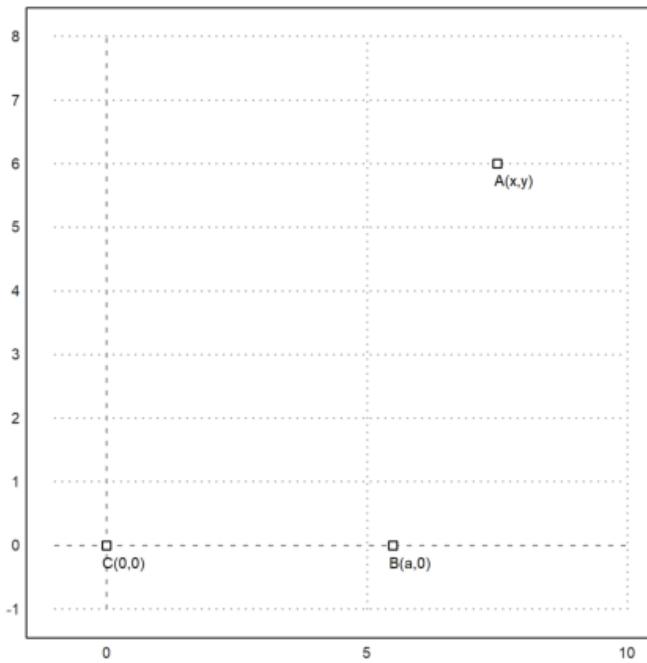
Untuk membuktikan hal ini kita misalkan $C(0,0)$, $B(a,0)$ dan $A(x,y)$, $b=AC$, $c=AB$. Luas segitiga ABC adalah

$$L_{\triangle ABC} = \frac{1}{2}a \times y.$$

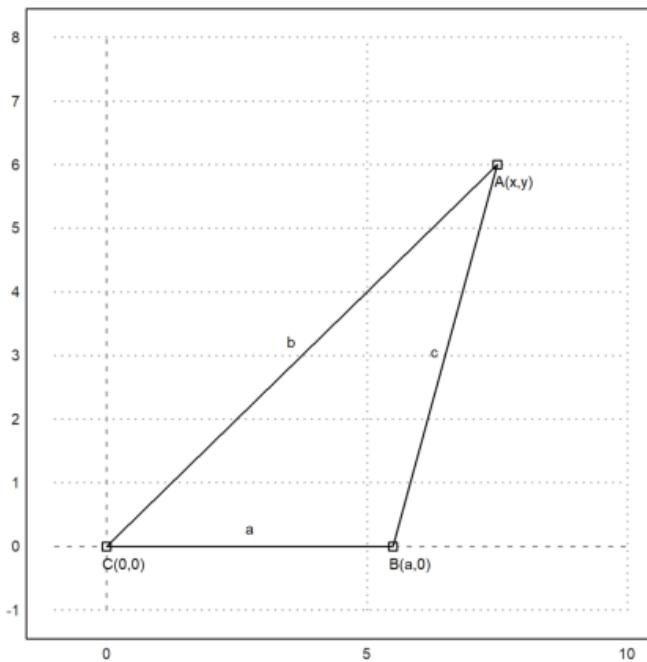
Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2 + y^2 = b^2, \quad (x - a)^2 + y^2 = c^2.$$

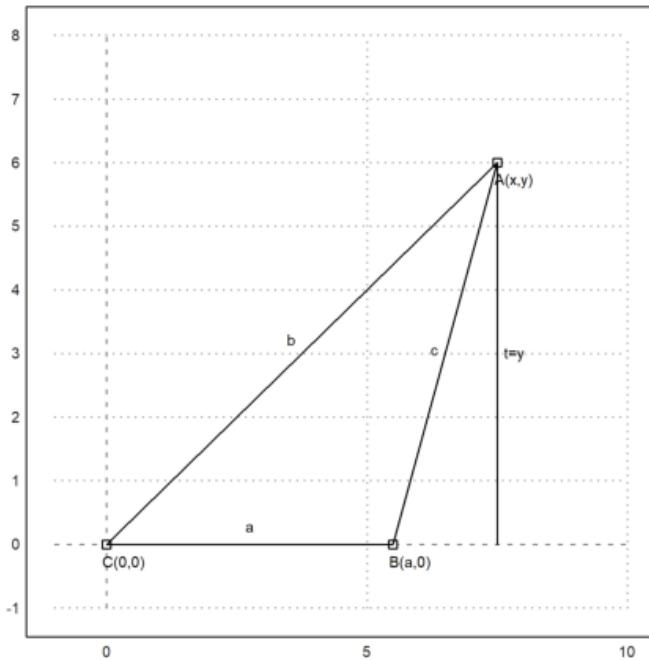
```
>setPlotRange(-1,10,-1,8); plotPoint([0,0], "C(0,0)"); plotPoint([5.5,0], "B(a,0)"); ...
plotPoint([7.5,6], "A(x,y)":
```



```
>plotSegment([0,0],[5.5,0], "a",25); plotSegment([5.5,0],[7.5,6],"c",15); ...
plotSegment([0,0],[7.5,6],"b",25):
```



```
>plotSegment([7.5,6],[7.5,0],"t=y",25):
```



```
>&assume(a>0); sol &= solve([x^2+y^2=b^2, (x-a)^2+y^2=c^2], [x, y])
```

```
[]
```

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 dapar Rumus Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H(a, b, [1, 0, 4]) = \frac{a |ysol|}{2}$$

```
>$'Luas=H(2,5,6) // luas segitiga dengan panjang sisi-sisi 2, 5, 6
```

$$Luas = |ysol|$$

Tentu saja, setiap segitiga siku-siku adalah kasus yang sudah dikenal.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

```
Function H not found.
Try list ... to find functions!
Error in:
H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...  
^
```

Dan juga jelas, bahwa ini adalah segitiga dengan luas maksimal dan dua sisinya 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7); // Kurva luas segitiga sengan panjang sisi 3, 4, x (1<= x <=7)
```

```
Function H not found.
Try list ... to find functions!
Error in expression: H(3,4,x)
%ploteval:
```

```

y0=f$(x[1],args());
adaptiveevalone:
s=%ploteval(g$,t;args());
Try "trace errors" to inspect local variables after errors.
plot2d:
dw/n,dw/n^2,dw/n,auto;args());

```

Kasus umum juga berfungsi.

```

>$solve(diff(H(a,b,c)^2,c)=0,c)

Maxima said:
diff: second argument must be a variable; found [1,0,4]
-- an error. To debug this try: debugmode(true);

Error in:
$solve(diff(H(a,b,c)^2,c)=0,c) ...

```

Sekarang mari kita cari himpunan semua titik di mana $b+c=d$ untuk beberapa konstanta d. Diketahui bahwa ini adalah ellips.

```

>s1 &= subst(d-c,b,sol[2]); $s1

Maxima said:
part: invalid index of list or matrix.
-- an error. To debug this try: debugmode(true);

Error in:
s1 &= subst(d-c,b,sol[2]); $s1 ...

```

Dan buat fungsi 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)
0
0

```

Sekarang kita bisa menggambar setnya. Sisi b bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan ellips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```

Kita dapat memeriksa persamaan umum untuk ellips 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 dan v adalah setengah sumbu.

```
>$ratsimp((fx(a,c,d)-a/2)^2/u^2+fy(a,c,d)^2/v^2 with [u=d/2,v=sqrt(d^2-a^2)/2])
```

$$\frac{a^2}{d^2}$$

Kita lihat bahwa tinggi dan luas segitiga adalah maksimal untuk $x=0$. Jadi luas segitiga dengan $a+b+c=d$ maksimal jika segitiga sama sisi. Kami ingin menurunkan ini secara analitis.

```
>eqns &= [diff(H(a,b,d-(a+b))^2,a)=0,diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

$$\left[2H(a,b,d-b-a) \left(\frac{d}{da} H(a,b,d-b-a) \right) = 0, 2H(a,b,d-b-a) \left(\frac{d}{db} H(a,b,d-b-a) \right) = 0 \right]$$

Kami mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusinya $a=b=c=d/3$.

```
>$solve(eqns,[a,b])
```

□

Ada juga metode Lagrange, memaksimalkan $H(a,b,c)^2$ terhadap $a+b+d=d$.

```
>&solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la
```

```

Closing bracket missing.
Found: solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la
Brackets: 1 (, 1 [
Symbolic expression expected.
Error in:
... 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])
```

```

Variable or function a not found.
Error in:
diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la]) ...

```

Kita bisa membuat plot situasinya. Pertama-tama atur poin di Maxima.

```
>A &= at([x,y],sol[2]); $A
```

```

Maxima said:
part: invalid index of list or matrix.
-- an error. To debug this try: debugmode(true);

Error in:
A &= at([x,y],sol[2]); $A ...

```

```
>B &= [0,0]; $B, C &= [a,0]; $C
```

[0,0]

[a,0]

Kemudian atur rentang plot, dan plot titik-titiknya.

```
>setPlotRange(0,5,-2,3); ...
a=4; b=3; c=2; ...
plotPoint(mxmeval("B"), "B"); plotPoint(mxmeval("C"), "C"); ...
plotPoint(mxmeval("A"), "A");
```

```

Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
    return evaluate(mxm(s));
Error in:
... otPoint(mxmeval("C"), "C"); plotPoint(mxmeval("A"), "A"): ...

```

Plot segmen.

```
>plotSegment(mxmeval("A"),mxmeval("C")); ...
plotSegment(mxmeval("B"),mxmeval("C")); ...
plotSegment(mxmeval("B"),mxmeval("A"));
```

```

Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
    return evaluate(mxm(s));
Error in:
plotSegment(mxmeval("A"),mxmeval("C")); plotSegment(mxmeval("B ...

```

Hitung tegak lurus tengah di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Kami mendapatkan rumus untuk jari-jari lingkaran.

```
>U &= lineIntersection(h,g);
```

Kami mendapatkan rumus untuk jari-jari lingkaran.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

$$\frac{\sqrt{a_2^2 + a_1^2} \sqrt{a_2^2 + a_1^2 - 2 a a_1 + a^2}}{2 |a_2|}$$

Mari kita tambahkan ini ke plot.

```
>plotPoint(U()); ...
plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
```

```
Variable a2 not found!
Use global variables or parameters for string evaluation.
Error in ^
Error in expression: [a/2, (a2^2+a1^2-a*a1)/(2*a2)]
Error in:
plotPoint(U()); plotCircle(circleWithCenter(mxmeval("U"),mxmev ...
```

Menggunakan geometri, kami memperoleh rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk radiusnya. Kami dapat memeriksa, apakah ini benar dengan Maxima. Maxima akan memfaktorkan ini hanya jika kita kuadratkan.

```
>$c^2/sin(computeAngle(A,B,C))^2 | factor
```

$$\left[\frac{a_2^2 + a_1^2}{a_2^2}, 0, \frac{16 (a_2^2 + a_1^2)}{a_2^2} \right]$$

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari sembarang segitiga yang tidak sama sisi. Ini adalah garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk orthocenter, circumcenter, centroid, titik Exeter dan pusat lingkaran sembilan titik segitiga.

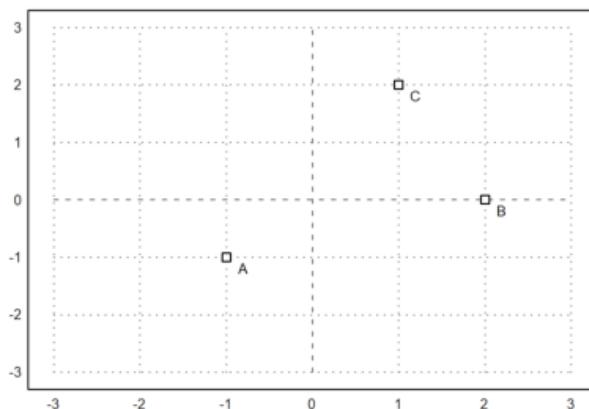
Untuk demonstrasi, kami menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kita mendefinisikan sudut-sudut segitiga di Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolis.

```
>A::=[-1,-1]; B::=[2,0]; C::=[1,2];
```

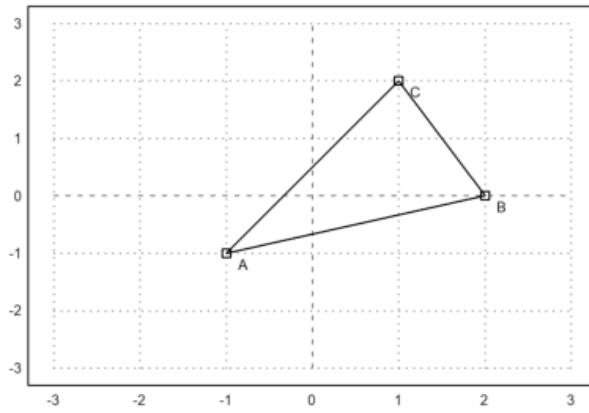
Untuk memplot objek geometris, kami menyiapkan area plot, dan menambahkan titik ke sana. Semua plot objek geometris ditambahkan ke plot saat ini.

```
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```



Kita juga bisa menambahkan sisi segitiga.

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,"");
```



Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

```
>$areaTriangle(A,B,C)
```

$$-\frac{7}{2}$$

Kita dapat menghitung koefisien sisi c.

```
>c &= lineThrough(A,B)
```

$$[-1, 3, -2]$$

Dan jugaapatkan rumus untuk baris 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 menghasilkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C, $at(%,[x=C[1],y=C[2]]))
```

$$\frac{3y - x + 2}{\sqrt{10}}$$

$$\frac{7}{\sqrt{10}}$$

Sekarang kita hitung lingkaran luar ABC.

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

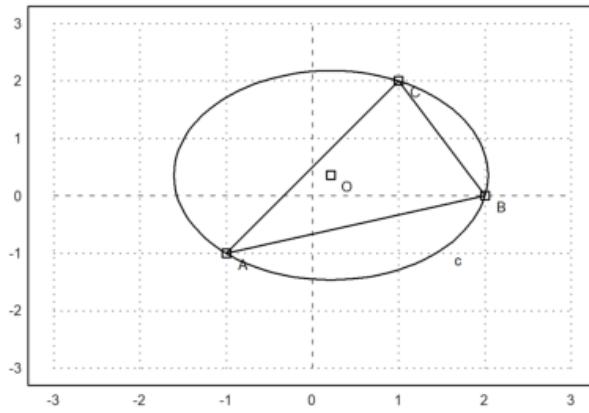
$$\left(y - \frac{5}{14}\right)^2 + \left(x - \frac{3}{14}\right)^2 = \frac{325}{98}$$

```
>O &= getCircleCenter(LL); $O
```

$$\left[\frac{3}{14}, \frac{5}{14}\right]$$

Gambarkan lingkaran dan pusatnya. Cu dan U adalah simbolis. Kami mengevaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O");
```



Kita dapat menghitung perpotongan ketinggian di ABC (orthocenter) secara numerik dengan perintah berikut.

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...  
perpendicular(B,lineThrough(A,C))); $H
```

$$\left[\frac{11}{7}, \frac{2}{7} \right]$$

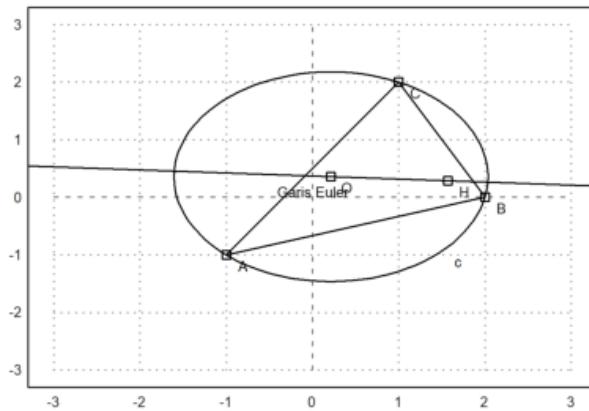
Sekarang kita dapat menghitung garis Euler dari segitiga.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$-\frac{19y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan ke plot kami.

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```

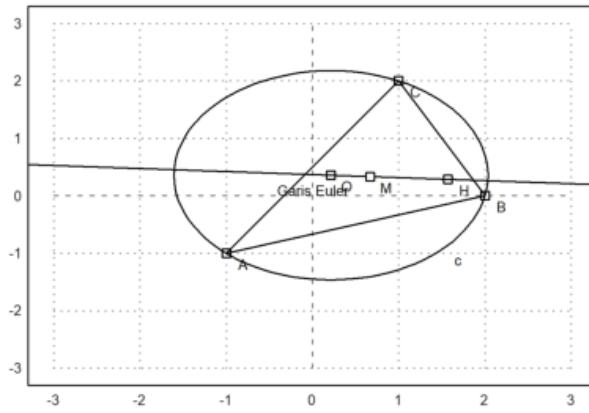


Pusat gravitasi harus berada di garis ini.

```
>M &= (A+B+C)/3; $getLineEquation(el,x,y) with [x=M[1],y=M[2]]
```

$$-\frac{1}{2} = -\frac{1}{2}$$

```
>plotPoint(M(),"M"); // titik berat
```



Teorinya memberitahu kita $MH=2\cdot MO$. Kita perlu menyederhanakan dengan radcan untuk mencapai ini.

```
> $distance(M,H) / $distance(M,O) | radcan
```

2

Fungsi termasuk fungsi untuk sudut juga.

```
> $computeAngle(A,C,B), degprint(%())
```

$$\arccos\left(\frac{4}{\sqrt{5}\sqrt{13}}\right)$$

$60^\circ 15' 18.43''$

Persamaan untuk pusat incircle tidak terlalu bagus.

```
> Q &= lineIntersection(angleBisector(A,C,B), angleBisector(C,B,A)) | radcan; $Q
```

$$\left[\frac{\left(2^{\frac{3}{2}} + 1\right)\sqrt{5}\sqrt{13} - 15\sqrt{2} + 3}{14}, \frac{(\sqrt{2} - 3)\sqrt{5}\sqrt{13} + 52^{\frac{3}{2}} + 5}{14} \right]$$

Mari kita hitung juga ekspresi untuk jari-jari lingkaran yang tertulis.

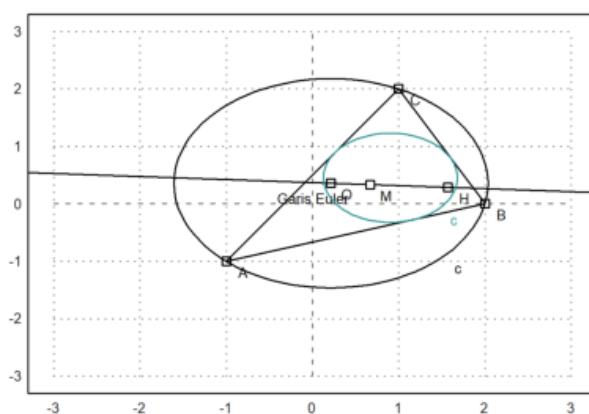
```
> r &= distance(Q, projectToLine(Q, lineThrough(A, B))) | ratsimp; $r
```

$$\frac{\sqrt{(-41\sqrt{2} - 31)\sqrt{5}\sqrt{13} + 115\sqrt{2} + 614}}{7\sqrt{2}}$$

```
> LD &= circleWithCenter(Q, r); // Lingkaran dalam
```

Mari kita tambahkan ini ke plot.

```
> color(5); plotCircle(LD());
```



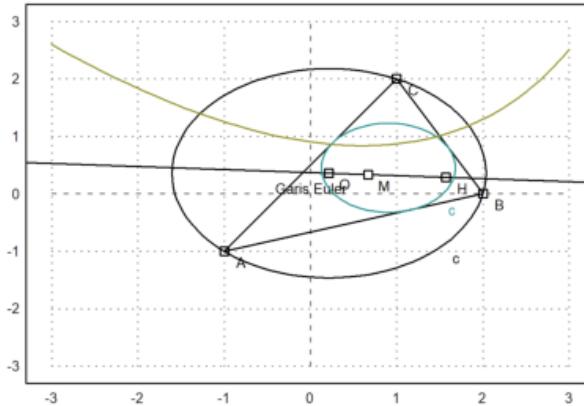
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>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 menjadi beberapa fungsi, tetapi pemecah default Maxima hanya dapat menemukan solusinya, jika kita kuadratkan persamaannya. Akibatnya, kami mendapatkan solusi palsu.

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

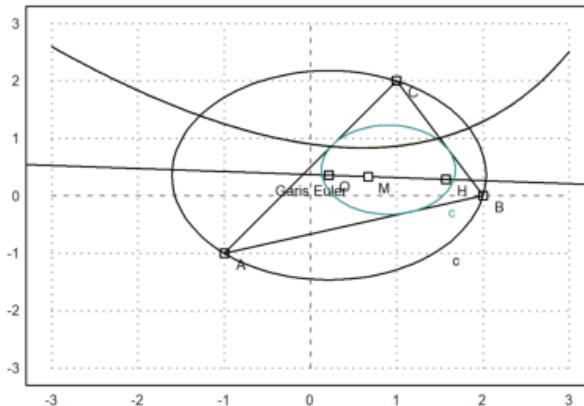
$$[y = -3x - \sqrt{70}\sqrt{9-2x} + 26, \\ y = -3x + \sqrt{70}\sqrt{9-2x} + 26]$$

Solusi pertama adalah

*akar*₁

Menambahkan solusi pertama ke plot menunjukkan, bahwa itu memang jalan yang kita cari. Teorinya memberi tahu kita bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1):
```



```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan kurva di atas
```

$$g(x) = -3x - \sqrt{70}\sqrt{9-2x} + 26$$

```
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

```
>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
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

2.135605779339061

Contoh 5: Trigonometri Rasional

Ini terinspirasi dari ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti pengertian klasik tentang jarak dan sudut dengan kuadrat dan penyebaran. Dengan menggunakan ini, memang mungkin untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

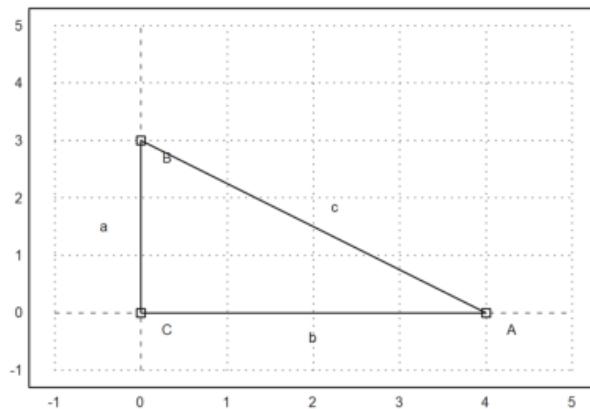
Berikut ini, saya memperkenalkan konsep, dan memecahkan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keuntungan utama dari trigonometri rasional bahwa perhitungan hanya dapat dilakukan dengan kertas dan pensil. Anda diundang untuk memeriksa hasil tanpa komputer.

Intinya adalah bahwa perhitungan rasional simbolis sering kali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

```
>load geometry;
```

Untuk pengenalan pertama, kami menggunakan segitiga persegi panjang dengan proporsi Mesir terkenal 3, 4 dan 5. Perintah berikut adalah perintah Euler untuk merencanakan geometri bidang yang terdapat dalam file Euler "geometry.e".

```
>C &:= [0,0]; A &:= [4,0]; B &:= [0,3]; ...
setPlotRange(-1,5,-1,5); ...
plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
insimg(30);
```



Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

di mana w_a adalah sudut di A. Cara yang biasa untuk menghitung sudut ini, adalah dengan mengambil invers 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, itu hanya jarak kuadran. Berikut ini, a, b, dan c menunjukkan kuadran dari sisi-sisinya.

Teorema Pythagoras menjadi $a+b=c$.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

$$25 = 25$$

Pengertian kedua dari trigonometri rasional adalah penyebaran. Spread mengukur pembukaan antar baris. Ini adalah 0, jika garis-garisnya sejajar, dan 1, jika garis-garisnya persegi panjang. Ini adalah kuadrat sinus sudut antara dua garis.

Penyebaran garis AB dan AC pada gambar di atas didefinisikan sebagai:

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

di mana a dan c adalah kuadrat dari sembarang segitiga siku-siku dengan salah satu sudut di A.

```
>sa &= a/c; $sa
```

$$\frac{9}{25}$$

Ini lebih mudah dihitung daripada sudut, tentu saja. Tetapi Anda kehilangan properti bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengonversi nilai perkiraan untuk sudut wa menjadi spread, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

$$9/25$$

Hukum kosinus trigonometri klasik diterjemahkan menjadi "hukum silang" berikut.

$$(c + b - a)^2 = 4bc(1 - s_a)$$

Di sini a , b , dan c adalah kuadrat dari sisi-sisi segitiga, dan sa adalah penyebaran sudut A. Sisi a , seperti biasa, berhadapan dengan sudut A.

Hukum ini diimplementasikan dalam file geometri.e yang kami muat ke Euler.

```
>$crosslaw(aa,bb,cc,sa)
```

$$\left[\left(bb - aa + \frac{7}{6} \right)^2, \left(bb - aa + \frac{7}{6} \right)^2, \left(bb - aa + \frac{5}{3\sqrt{2}} \right)^2 \right] = \left[\frac{14bb(1-saa)}{3}, \frac{14bb(1-saa)}{3}, \frac{52^{\frac{3}{2}}bb(1-saa)}{3} \right]$$

Dalam kasus kami, kami mendapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024 = 1024$$

Mari kita gunakan crosslaw ini untuk mencari spread di A. Untuk melakukan ini, kita buat crosslaw untuk kuadran a , b , dan c , dan selesaikan untuk spread yang tidak diketahui sa .

Anda dapat melakukannya dengan tangan dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasilnya, kami sudah memiliki.

```
>$crosslaw(a,b,c,x), $solve(% ,x)
```

$$1024 = 1600(1 - x)$$

$$\left[x = \frac{9}{25} \right]$$

Kita sudah tahu ini. Definisi spread adalah kasus khusus dari crosslaw.

Kita juga dapat menyelesaikan ini untuk umum a, b, c . Hasilnya adalah rumus yang menghitung penyebaran sudut segitiga yang diberikan kuadrat dari ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[\left[\frac{168bbz + 36bb^2 + (-72aa - 84)bb + 36aa^2 - 84aa + 49}{36}, \frac{168bbx + 36bb^2 + (-72aa - 84)bb + 36aa^2 - 84aa + 49}{36}, \frac{152^{\frac{3}{2}}bbz + 18bb^2 + (-36aa - 152^{\frac{1}{2}})bb + 18aa^2 - 152^{\frac{1}{2}}aa + 25}{18} \right] = 0 \right]$$

Kita bisa membuat fungsi dari hasilnya. Fungsi seperti itu sudah didefinisikan dalam file geometri.e dari Euler.

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Sebagai contoh, kita dapat menggunakan untuk menghitung sudut segitiga dengan sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak begitu mudah didapat jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat.

```
>degprint(arcsin(sqrt(6/7)))
```

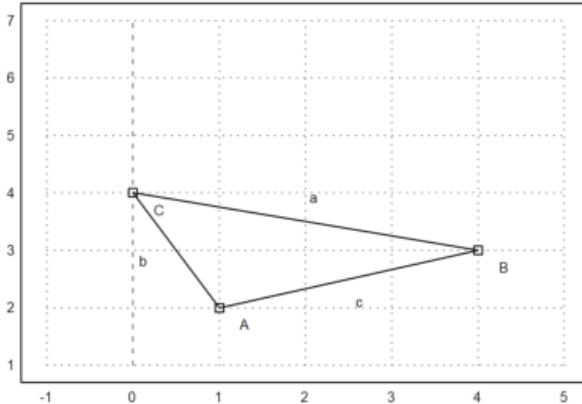
$$67^\circ 47' 32.44''$$

Contoh lain

Sekarang, mari kita coba contoh yang lebih maju.

Kami mengatur tiga sudut segitiga sebagai berikut.

```
>A&:=[1,2]; B&:=[4,3]; C&:=[0,4]; ...
setPlotRange(-1,5,1,7); ...
plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
insimg;
```



Menggunakan Pythagoras, mudah untuk menghitung jarak antara dua titik. Saya pertama kali menggunakan jarak fungsi file Euler untuk geometri. Jarak fungsi menggunakan geometri klasik.

```
>$distance(A,B)
```

$$\sqrt{10}$$

Euler juga mengandung fungsi untuk kuadran antara dua titik.

Dalam contoh berikut, karena $c+b$ bukan a , maka segitiga itu bukan persegi panjang.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

$$10$$

$$5$$

$$17$$

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan hasil kali titik dua vektor. Hasilnya adalah beberapa pendekatan floating point.

$$A = \langle 1, 2 \rangle \quad B = \langle 4, 3 \rangle, \quad C = \langle 0, 4 \rangle$$

$$\mathbf{a} = C - B = \langle -4, 1 \rangle, \quad \mathbf{c} = A - B = \langle -3, -1 \rangle, \quad \beta = \angle ABC$$

$$\mathbf{a} \cdot \mathbf{c} = |\mathbf{a}| \cdot |\mathbf{c}| \cos \beta$$

$$\cos \angle ABC = \cos \beta = \frac{\mathbf{a} \cdot \mathbf{c}}{|\mathbf{a}| \cdot |\mathbf{c}|} = \frac{12 - 1}{\sqrt{17} \sqrt{10}} = \frac{11}{\sqrt{17} \sqrt{10}}$$

```
>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. Kami memasukkan kuadran a, b, dan c ke dalam hukum silang dan menyelesaikan x.

```
>$crosslaw(a,b,c,x), $solve(% ,x), // (b+c-a)^=4b.c(1-x)
```

$$4 = 200 (1 - x)$$

$$\left[x = \frac{49}{50} \right]$$

Yaitu, apa yang dilakukan oleh penyebaran fungsi yang didefinisikan dalam "geometry.e".

```
>sb &= spread(b,a,c); $sb
```

$$\frac{49}{170}$$

Maxima mendapatkan hasil yang sama menggunakan trigonometri biasa, jika kita memaksanya. Itu menyelesaikan istilah $\sin(\arccos(...))$ menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{49}{170}$$

Setelah kita memiliki spread di B, kita dapat menghitung tinggi ha di sisi a. Ingat bahwa

$$s_b = \frac{h_a}{c}$$

Menurut definisi.

```
>ha &= c*sb; $ha
```

$$\frac{49}{17}$$

Gambar berikut telah dihasilkan dengan program geometri C.a.R., yang dapat menggambar kuadrat dan menyebar.

image: (20) Rational_Geometry_CaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadratnya.

```
>$sqrt(ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita berhadapan dengan kuadrat!

```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{7}{2}$$

Rumus determinan biasa menghasilkan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

Rumus Bangau

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue(a,b,c,sb,ha);
```

Pertama kita hitung spread di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat ("quadrea"?), faktorkan dengan Maxima, dan kita mendapatkan rumus Heron yang terkenal.

Memang, ini sulit dilakukan dengan pensil dan kertas.

```
>$spread(b^2,c^2,a^2), $factor(%*c^2*a^2/4)
```

$$\frac{-c^4 - (-2b^2 - 2a^2)c^2 - b^4 + 2a^2b^2 - a^4}{4a^2c^2}$$
$$\frac{(-c + b + a)(c - b + a)(c + b - a)(c + b + a)}{16}$$

Aturan Triple Spread

Kerugian dari spread adalah mereka tidak lagi hanya menambahkan sudut yang sama.

Namun, tiga spread dari sebuah segitiga memenuhi aturan "triple spread" berikut.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

$$(sc + sb + sa)^2 = 2(sc^2 + sb^2 + sa^2) + 4sa sb sc$$

Aturan ini berlaku untuk setiap tiga sudut yang menambah 180°.

$$\alpha + \beta + \gamma = \pi$$

Sejak menyebar

$$\alpha, \pi - \alpha$$

sama, aturan triple spread juga benar, jika

$$\alpha + \beta = \gamma$$

Karena penyebaran sudut negatif adalah sama, aturan penyebaran rangkap tiga juga berlaku, jika

$$\alpha + \beta + \gamma = 0$$

Misalnya, kita dapat menghitung penyebaran sudut 60°. Ini 3/4. Persamaan memiliki solusi kedua, bagaimanapun, di mana semua spread adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

$$\left[x = \frac{3}{4}, x = 0 \right]$$

Sebaran 90° jelas 1. Jika dua sudut dijumlahkan menjadi 90°, sebarannya menyelesaikan persamaan sebaran rangkap tiga dengan a,b,1. Dengan perhitungan berikut kita mendapatkan a+b=1.

```
>$triplespread(x,y,1), $solve(%,x)
```

$$(y + x + 1)^2 = 2(y^2 + x^2 + 1) + 4xy$$
$$[x = 1 - y]$$

Karena sebaran 180°-t sama dengan sebaran t, rumus sebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau selisih dua sudut lainnya.

Jadi kita dapat menemukan penyebaran sudut berlipat ganda. Perhatikan bahwa ada dua solusi lagi. Kami membuat ini fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))
```

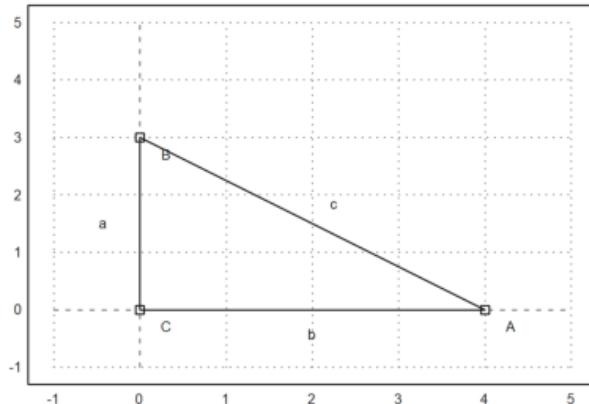
$$\left[x = 4a - 4a^2, x = 0 \right]$$

$$-4(a - 1) a$$

Pembagi Sudut

Ini situasinya, kita sudah tahu.

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
setPlotRange(-1,5,-1,5); ...
plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
insimg;
```



Mari kita hitung panjang garis bagi sudut di A. Tetapi kita ingin menyelesaikannya untuk umum a,b,c.

```
>&remvalue(a,b,c);
```

Jadi pertama-tama kita hitung penyebaran sudut yang dibagi dua di A, dengan menggunakan rumus sebaran rangkap tiga.

Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut terbelah 180° -wa.

```
>$triplespread(x,x,a/(a+b)), $solve(% ,x), sa2 &= rhs(%[1]); $sa2
```

$$\left(2x + \frac{a}{b+a}\right)^2 = 2\left(2x^2 + \frac{a^2}{(b+a)^2}\right) + \frac{4ax^2}{b+a}$$

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

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

Mari kita periksa persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

Kami dapat mencetak sudut dalam Euler, setelah mentransfer penyebaran ke radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

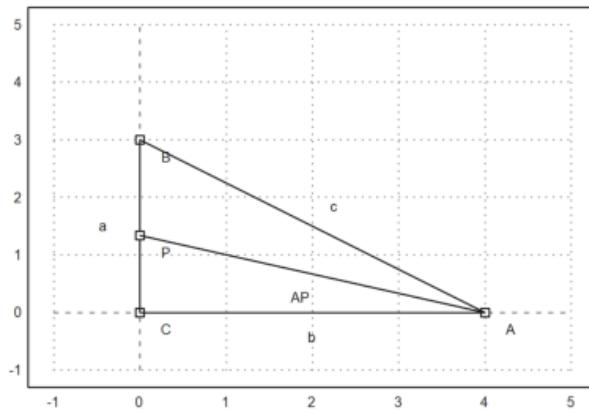
$$18^\circ 26' 5.82''$$

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 dalam contoh spesifik kita.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

```
0.321750554397
0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.

Kami menggunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

berlaku dalam segitiga apa pun. Kuadratkan, itu diterjemahkan ke dalam apa yang disebut "hukum penyebaran"

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

di mana a,b,c menunjukkan quadrances.

Karena spread CPA adalah $1 - s_a^2$, kita dapatkan darinya $b/s_a = b/\sqrt{1 - s_a^2}$ dan dapat menghitung bisa (kuadran dari garis-bagi sudut).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$\frac{2b(b+a)}{\sqrt{b}\sqrt{b+a} + b+a}$$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])")), distance(A,P)
```

```
4.21637021356
4.21637021356
```

Kita juga dapat menghitung P menggunakan rumus spread.

```
>py&=factor(ratsimp(sa2*bisa)); $py
```

$$-\frac{b(\sqrt{b}\sqrt{b+a} - b - a)}{\sqrt{b}\sqrt{b+a} + b + a}$$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

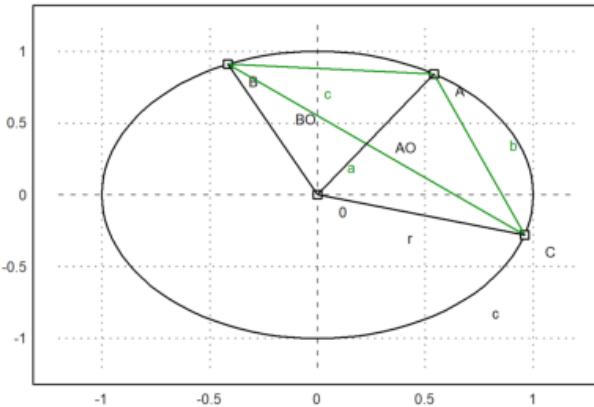
```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

```
1.333333333333
```

Sudut Akord

Perhatikan situasi berikut.

```
>setPlotRange(1.2); ...
color(1); plotCircle(circleWithCenter([0,0],1)); ...
A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...
plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
color(1); O:=[0,0]; plotPoint(O,"O"); ...
plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
insimg;
```



Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran rangkap tiga untuk sudut-sudut di pusat O untuk r . Jadi kita mendapatkan rumus untuk jari-jari kuadrat dari pericircle dalam hal kuadrat dari sisi.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); $rabc
```

$$\frac{abc}{c^2 - 2bc + a(-2c - 2b) + b^2 + a^2}$$

Kita dapat menjadikannya sebagai fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasilnya untuk poin A,B,C.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Jari-jarinya memang 1.

```
>periradius(a,b,c)
```

1

Faktanya, spread CBA hanya bergantung pada b dan c . Ini adalah teorema sudut chord.

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Sebenarnya spreadnya adalah $b/(4r)$, dan kita melihat bahwa sudut chord dari chord b adalah setengah dari sudut pusat.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

0

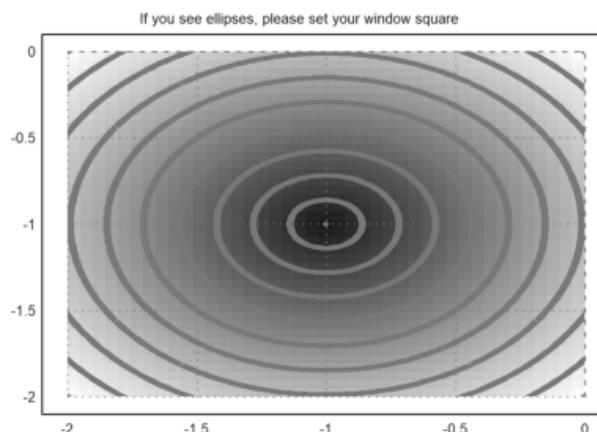
Contoh 6: Jarak Minimal pada Bidang

Catatan awal

Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis level yang agak sederhana: lingkaran berpusat di A.

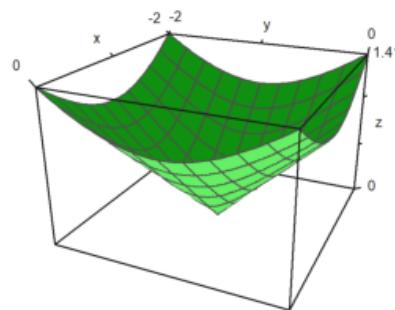
```

>&remvalue();
>A=[-1,-1];
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...
  title="If you see ellipses, please set your window square"):
```



dan grafiknya juga agak sederhana: bagian atas kerucut:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```



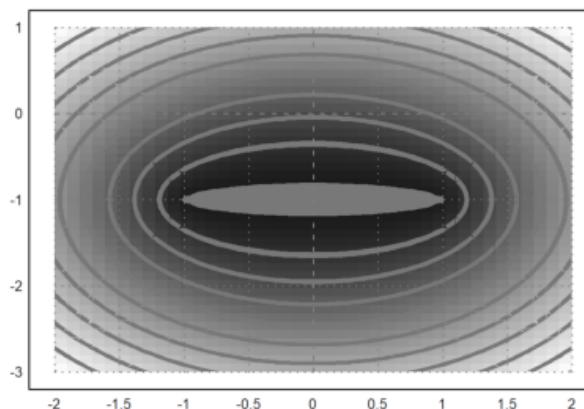
Tentu saja minimal 0 dicapai di A.

Dua poin

Sekarang kita lihat fungsi MA+MB dimana A dan B adalah dua titik (tetap). Ini adalah "fakta yang diketahui" bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali untuk AB minimum yang konstan pada segmen [AB]:

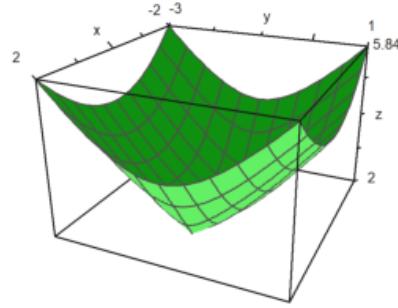
```

>B=[1,-1];
>function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2+(y-B[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



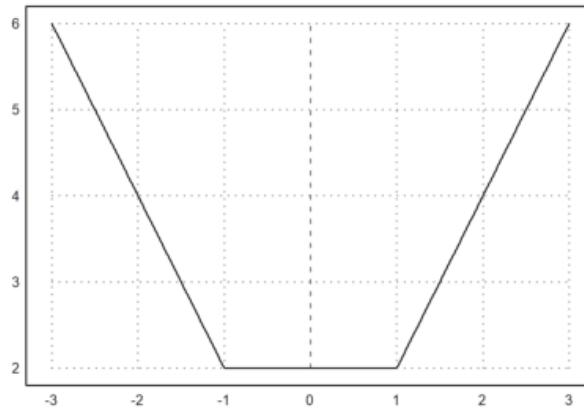
Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Pembatasan garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



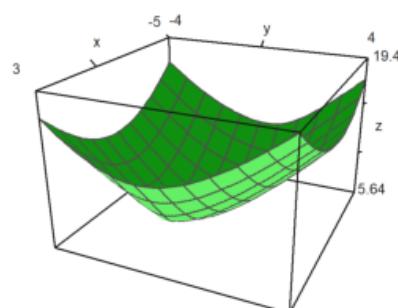
Tiga poin

Sekarang hal-hal yang kurang sederhana: Ini sedikit kurang terkenal bahwa $MA+MB+MC$ mencapai minimum pada satu titik pesawat tetapi untuk menentukan itu kurang sederhana:

- 1) Jika salah satu sudut segitiga ABC lebih dari 120° (katakanlah di A), maka minimum dicapai pada titik ini (misalnya $AB+AC$).

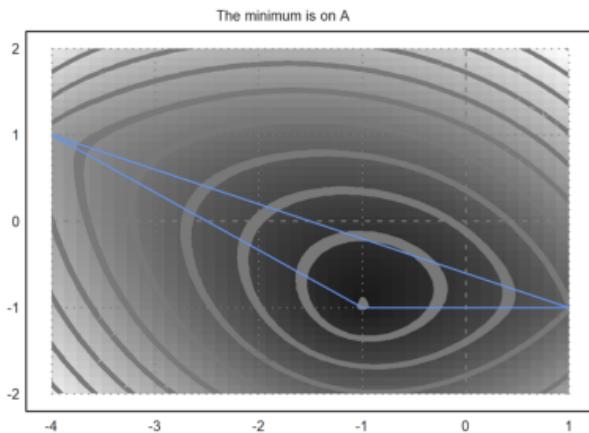
Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
>insimg;
```



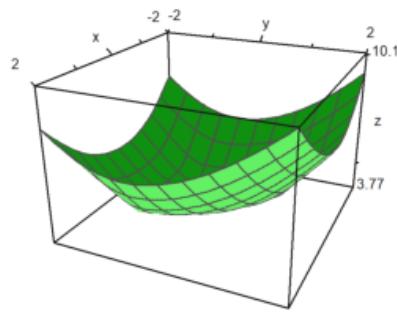
```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)';
>plot2d(P[1],P[2],add=1,color=12);
```

```
>insimg;
```

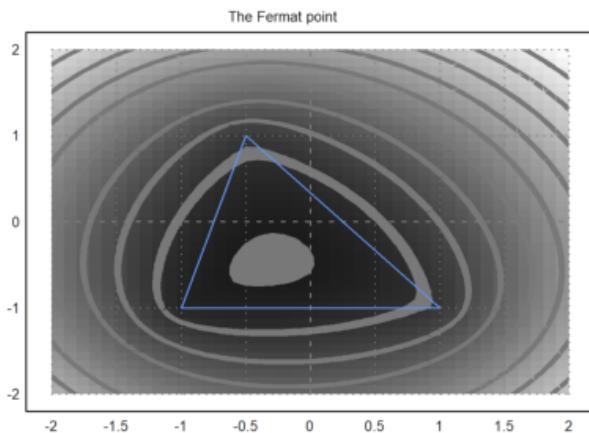


- 2) Tetapi jika semua sudut segitiga ABC kurang dari 120° , minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (maka masing-masing 120°):

```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):
```



```
>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");
>P=(A_B_C_A)';
>plot2d(P[1],P[2],add=1,color=12);
>insimg;
```



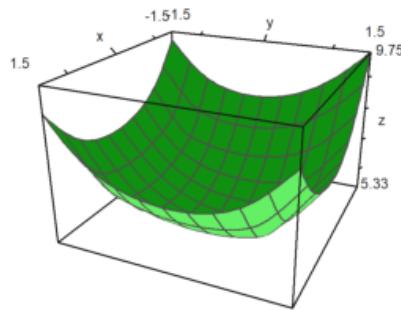
Merupakan kegiatan yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya tahu soft yang ditulis di Jawa yang memiliki instruksi "garis kontur" ...

Semua ini di atas telah ditemukan oleh seorang hakim Perancis bernama Pierre de Fermat; dia menulis surat kepada dilettants lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di pajak penghasilan. Jadi titik unik F sedemikian rupa sehingga $FA+FB+FC$ minimal, disebut titik Fermat segitiga. Tetapi tampaknya beberapa tahun sebelumnya, Torricelli Italia telah menemukan titik ini sebelum Fermat melakukannya! Bagaimanapun tradisinya adalah mencatat poin ini F...

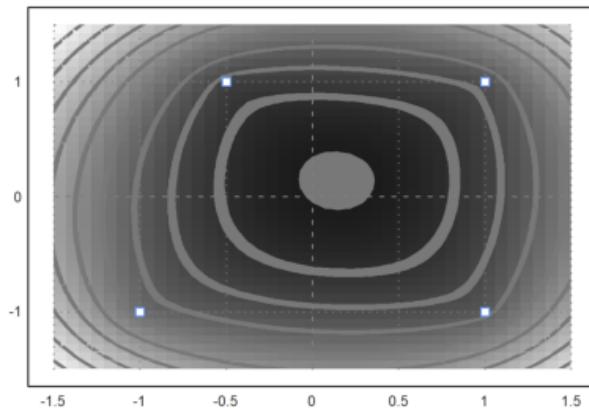
Empat poin

Langkah selanjutnya adalah menambahkan 4 titik D dan mencoba meminimalkan MA+MB+MC+MD; katakan bahwa Anda adalah operator TV kabel dan ingin mencari di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
```



```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
>insimg;
```



Masih ada minimum dan tidak tercapai di salah satu simpul A, B, C atau D:

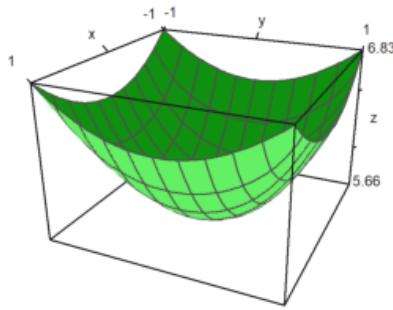
```
>function f(x):=d4(x[1],x[2])
>neldermin("f", [0.2,0.2])
```

```
[0.142858, 0.142857]
```

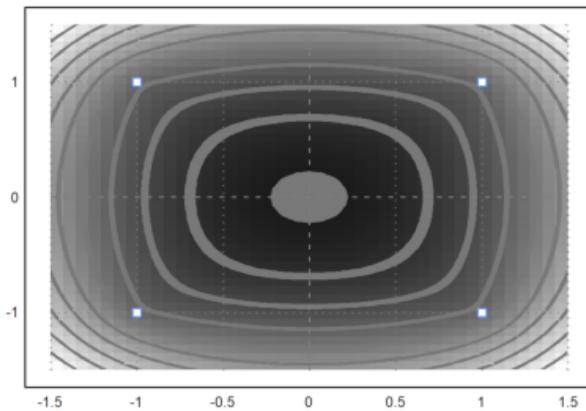
Tampaknya dalam kasus ini, koordinat titik optimal adalah rasional atau mendekati rasional...

Sekarang ABCD adalah persegi, kami berharap bahwa titik optimal akan menjadi pusat ABCD:

```
>C=[-1,1];
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1):
```



```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);
>insimg;
```



Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan pvengine.exe di jalur program.

Pertama kita hitung jari-jari bola.

Jika Anda melihat gambar di bawah, Anda melihat bahwa kita membutuhkan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file geometri.e dari Euler untuk ini.

```
>load geometry;
```

Pertama dua garis yang membentuk kerucut.

```
>g1 &= lineThrough([0,0],[1,a])
[- a, 1, 0]
```

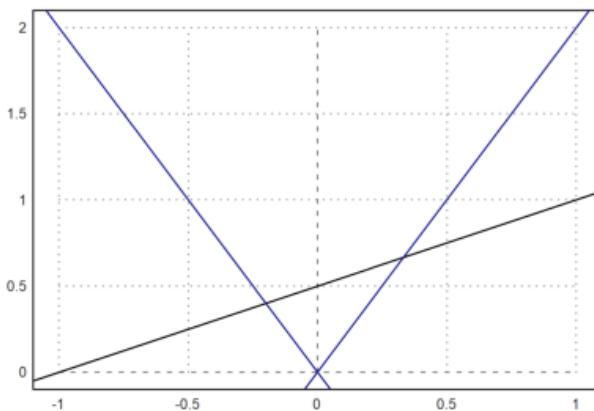
```
>g2 &= lineThrough([0,0],[-1,a])
[- a, - 1, 0]
```

Kemudian saya 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 sama.

```
>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]

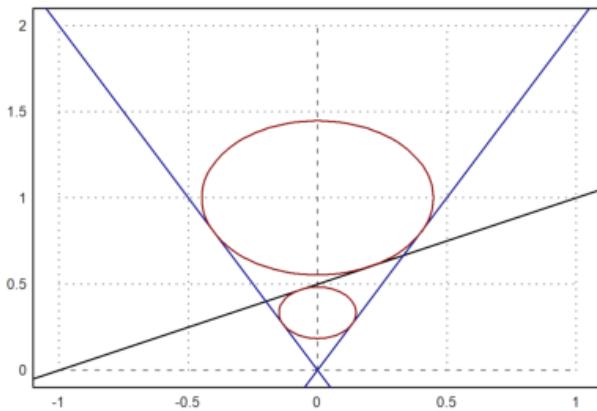
```
>dd := d()
```

[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 kami merencanakan semuanya dengan Povray. Perhatikan bahwa Anda mengubah perintah apa pun dalam urutan perintah Povray berikut, dan menjalankan kembali semua perintah dengan Shift-Return.

Pertama kita memuat fungsi povray.

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Kami mengatur adegan dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Next we write the two spheres to the Povray file.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

Dan kerucutnya, transparan.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

Kami menghasilkan bidang terbatas pada kerucut.

```
>gp=g();
>pc=povcone([0,0,0],0,[0,0,a],1,"");
>vp=[gp[1],0, gp[2]]; dp=gp[3];
>writeln(povplane(vp,dp,povlook(blue,0.5),pc));
```

Sekarang kita menghasilkan dua titik pada lingkaran, di mana bola menyentuh kerucut.

```
>function turnz(v) := return [-v[2],v[1],v[3]]
>P1=projectToLine([0,u[1]],g()); P1=turnz([P1[1],0,P1[2]]);
>writeln(povpoint(P1,povlook(yellow)));
>P2=projectToLine([0,u[2]],g()); P2=turnz([P2[1],0,P2[2]]);
>writeln(povpoint(P2,povlook(yellow))));
```

Kemudian kami menghasilkan dua titik di mana bola menyentuh bidang. Ini adalah fokus dari elips.

```
>P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
>writeln(povpoint(P3,povlook(yellow)));
>P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
>writeln(povpoint(P4,povlook(yellow))));
```

Selanjutnya kita hitung perpotongan P1P2 dengan bidang.

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
>writeln(povpoint(P5,povlook(yellow))));
```

Kami menghubungkan titik-titik dengan segmen garis.

```
>writeln(povsegment(P1,P2,povlook(yellow)));
>writeln(povsegment(P5,P3,povlook(yellow)));
>writeln(povsegment(P5,P4,povlook(yellow)));



```

Sekarang kita menghasilkan pita abu-abu, di mana bola menyentuh kerucut.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);
>pcl=povcylinder([0,0,P1[3]-defaultpointsiz/2],[0,0,P1[3]+defaultpointsiz/2],1);
>writeln(povintersection([pcw,pcl],povlook(gray)));
>pc2=povcylinder([0,0,P2[3]-defaultpointsiz/2],[0,0,P2[3]+defaultpointsiz/2],1);
>writeln(povintersection([pcw,pc2],povlook(gray)));



```

Mulai program Povray.

```
>povend();

Command was not allowed!
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);



```

Untuk mendapatkan Anaglyph ini kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali kemudian.

```
>function scene () ...
global a,u,dd,g,g1,defaultpointsiz;
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);
pcl=povcylinder([0,0,P1[3]-defaultpointsiz/2],[0,0,P1[3]+defaultpointsiz/2],1);
writeln(povintersection([pcw,pcl],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsiz/2],[0,0,P2[3]+defaultpointsiz/2],1);
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction



```

Anda membutuhkan kacamata merah/sian untuk menghargai efek berikut.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);



```

```
Variable scene not found!
Use global variables or parameters for string evaluation.
Error in expression: scene
Try "trace errors" to inspect local variables after errors.
povanaglyph:
    scene$(args());



```

Contoh 8: Geometri Bumi

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan sferis. Fungsi-fungsi tersebut terdapat dalam file "spherical.e" di folder contoh. Kita perlu memuat file itu terlebih dahulu.

```
>load "spherical.e";



```

Untuk memasukkan posisi geografis, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut koordinat Kampus FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569, 1.92657]
```

Anda dapat mencetak posisi ini dengan sposprint (cetak posisi spherical).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
S 7°46.467' E 110°23.050'
```

Mari kita tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,24.533)];  
>sposprint(Solo), sposprint(Semarang),
```

```
S 7°34.333' E 110°49.683'  
S 6°59.050' E 110°24.533'
```

Pertama kita menghitung vektor dari satu ke yang lain pada bola ideal. Vektor ini [pos,jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang 7° .

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan jarak FMIPA-Solo
```

```
65°20'26.60''  
53.8945384608
```

Ini adalah perkiraan yang baik. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak yang begitu pendek hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->" km" // perkiraan jarak FMIPA-Semarang
```

```
Commands must be separated by semicolon or comma!  
Found: // perkiraan jarak FMIPA-Semarang (character 32)  
You can disable this in the Options menu.  
Error in:  
esdist(FMIPA,Semarang)->" km" // perkiraan jarak FMIPA-Semarang ...  
^
```

Ada fungsi untuk heading, dengan mempertimbangkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

```
65.34°
```

Sudut segitiga melebihi 180° pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,Semarang,Solo); degprint(
```

```
180°0'10.77''
```

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, ini tidak akurat karena kesalahan pengurangan dalam $\text{asum}-\pi$.

```
>(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FMIPA-Solo-Semarang
```

```
Commands must be separated by semicolon or comma!  
Found: // perkiraan luas segitiga FMIPA-Solo-Semarang (character 32)  
You can disable this in the Options menu.  
Error in:  
(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FM ...  
^
```

Ada fungsi untuk ini, yang menggunakan garis lintang rata-rata segitiga untuk menghitung jari-jari bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi esarea()
```

2123.64310526 km²

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi heading dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kami menggunakan vektor. Untuk menambahkan vektor ke posisi, kami menggunakan vektor saddr.

```
>v=svector(FMIPA,Solo); sposprint(saddrvector(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(Solo),
```

```
S 7°34.333' E 110°49.683'  
S 7°34.333' E 110°49.683'
```

Mari kita beralih ke contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];  
>sposprint(Tugu), sposprint(Monas)
```

```
S 7°46.998' E 110°21.966'  
S 6°10.500' E 106°48.717'
```

Menurut Google Earth, jaraknya adalah 429,66 km. Kami mendapatkan pendekatan yang baik.

```
>esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Monas Jakarta
```

```
Commands must be separated by semicolon or comma!  
Found: // perkiraan jarak Tugu Jogja - Monas Jakarta (character 32)  
You can disable this in the Options menu.  
Error in:  
esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Mona ...  
^
```

Judulnya sama dengan judul yang dihitung di Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

```
294°17'2.85''
```

Namun, kita tidak lagi mendapatkan posisi target yang tepat, jika kita menambahkan heading dan jarak ke posisi semula. Hal ini terjadi, karena kita tidak menghitung fungsi invers secara tepat, tetapi mengambil perkiraan jari-jari bumi di sepanjang jalan.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

```
S 6°10.500' E 106°48.717'
```

Namun, kesalahannya tidak besar.

```
>sposprint(Monas),
```

```
S 6°10.500' E 106°48.717'
```

Tentu kita tidak bisa berlayar dengan tujuan yang sama dari satu tujuan ke tujuan lainnya, jika kita ingin menempuh jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti heading yang konstan!

Perhitungan berikut menunjukkan bahwa kami jauh dari tujuan yang benar, jika kami menggunakan pos yang sama selama perjalanan kami.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh dari jarak, menggunakan pos ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya jauh.

```
>sposprint(p), skmprint(esdist(p,Monas))
```

```
S 6°11.250' E 106°48.372'  
1.529km
```

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran garis lintang 30° , melainkan jalur terpendek yang dimulai 10° lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

```
79.69°
```

Tapi, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kami menyesuaikannya pada $1/10$ dari total jarak.

```
>p=P1; dist=esdist(P1,P2); ...  
loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
79.69°  
81.67°  
83.71°  
85.78°  
87.89°  
90.00°  
92.12°  
94.22°  
96.29°  
98.33°
```

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti heading yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

```
0.203km
```

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan pos setelah setiap $1/100$ dari total jarak dari Tugu ke Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...  
loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;  
>skmprint(esdist(p,Monas))
```

```
0.000km
```

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS di sepanjang lingkaran besar menuju Monas dengan fungsi navigasi.

```
>load spherical; v=navigate(Tugu,Monas,10); ...  
loop 1 to rows(v); sposprint(v[#]), end;
```

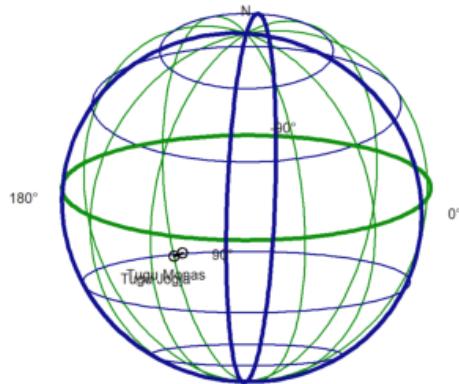
```
S 7°46.998' E 110°21.966'  
S 7°37.422' E 110°0.573'  
S 7°27.829' E 109°39.196'  
S 7°18.219' E 109°17.834'  
S 7°8.592' E 108°56.488'  
S 6°58.948' E 108°35.157'  
S 6°49.289' E 108°13.841'  
S 6°39.614' E 107°52.539'  
S 6°29.924' E 107°31.251'  
S 6°20.219' E 107°9.977'  
S 6°10.500' E 106°48.717'
```

Kami menulis sebuah fungsi, yang memplot bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
useglobal;
plotearth;
plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");
plotposline(v);
endfunction
```

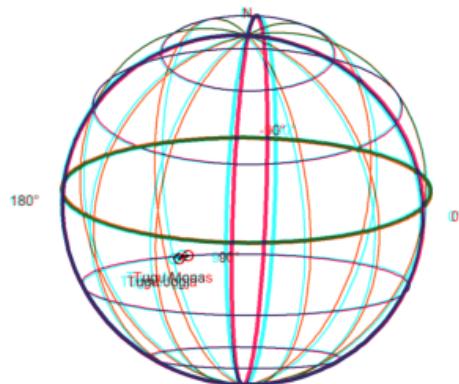
Sekarang rencanakan semuanya.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```



Atau gunakan plot3d untuk mendapatkan tampilan anaglyph. Ini terlihat sangat bagus dengan kacamata merah/sian.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```



MENCoba RUMUS-RUMUS PADA MATEI DI ATAS

Geometri Simbolik

```
>A &:= [2,0]; B &:= [0,2]; C &:= [3,3]; // menentukan tiga titik A, B, C
>c &:= lineThrough(B,C) // c=BC
```

```
[-1, 3, 6]
```

```
>$getLineEquation(c,x,y), $solve(% ,y) | expand // persamaan garis c
```

$$3y - x = 6$$

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

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

```
[3, 1, 6]
```

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

$$\left[\begin{array}{cc} 6 & 12 \\ -, & -- \\ 5 & 5 \end{array} \right]$$

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\left[\frac{6}{5}, \frac{12}{5} \right]$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{2^{\frac{5}{2}}}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[\frac{7}{4}, \frac{7}{4}, \frac{5}{2^{\frac{3}{2}}} \right]$$

```
>r=>getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

$$\frac{5}{2^{\frac{3}{2}}}$$

```
1.767766952966368
```

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{3}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

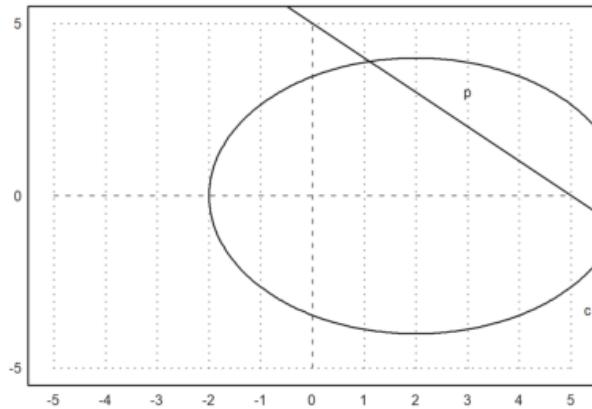
$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2
```

$$\left[\frac{\sqrt{2}\sqrt{10} + 2}{4}, \frac{\sqrt{2}\sqrt{10} + 2}{4} \right]$$

Garis dan Lingkaran yang berpotongan

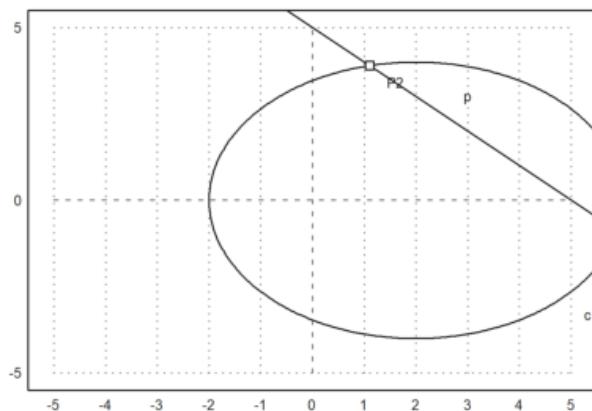
```
>A &:= [2,0]; c =circleWithCenter(A,4);
>B &:= [2,3]; C &:= [3,2]; l =lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l):
```



```
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2,
```

```
[5.89792, -0.897916]
[1.10208, 3.89792]
```

```
>plotPoint(P1); plotPoint(P2);
```



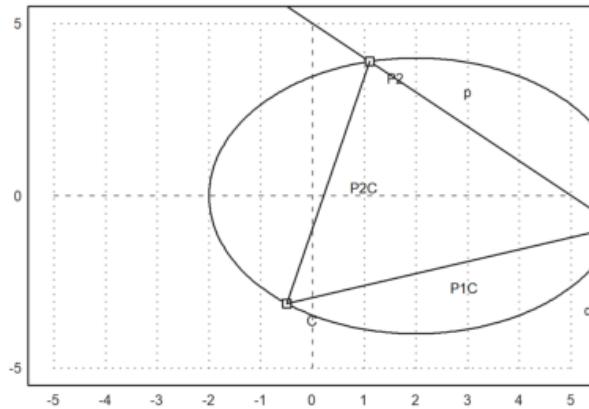
```
>c &= circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

```
[2, 0, 4]
```

```
>l &= lineThrough(B,C) // garis l melalui B dan C
```

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

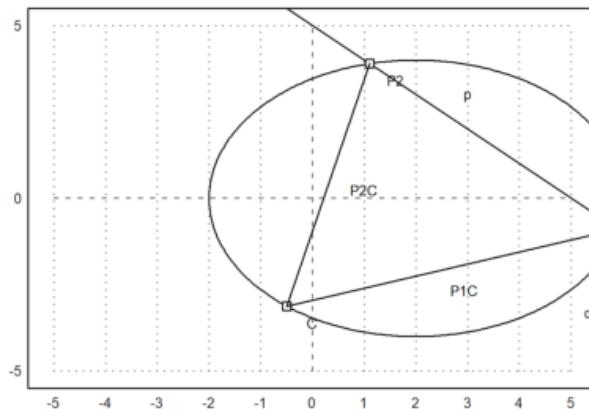
```
>C=A+normalize([-4,-5])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
```



```
>degsprint(computeAngle(P1,C,P2))
```

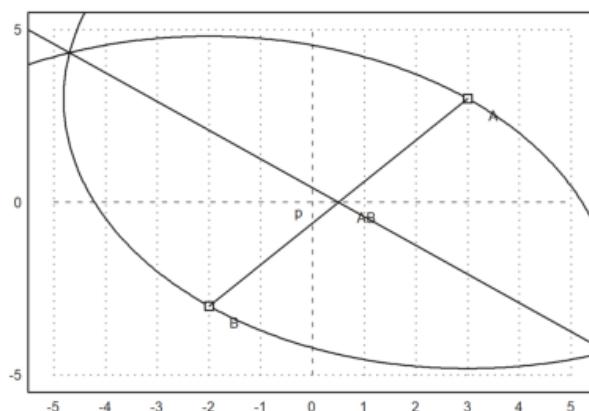
$57^\circ 58' 20.06''$

```
>insimg;
```



Garis Sumbu

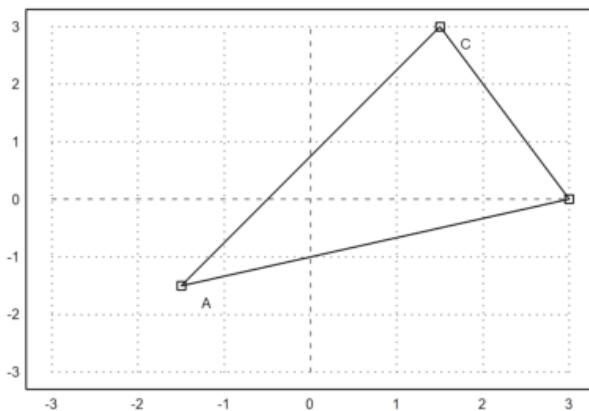
```
>A &:=[3,3]; B &:=[-2,-3];
>c1=circleWithCenter(A,distance(A,B));
>c2=circleWithCenter(B,distance(A,B));
>(P1,P2,f)=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(5); plotCircle(c1); plotCircle(c2);
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):
```



Garis Euler dan Parabola

```
>A &:=[-1.5,-1.5]; B &:=[3,0]; C &:=[1.5,3];
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,"");
```



```
>$areaTriangle(A,B,C)
```

$$-\frac{63}{8}$$

```
>c &= lineThrough(A,B)
```

$$\left[-\frac{3}{2}, \frac{9}{2}, -\frac{9}{2} \right]$$

```
>$getLineEquation(c,x,y)
```

$$\frac{9y}{2} - \frac{3x}{2} = -\frac{9}{2}$$

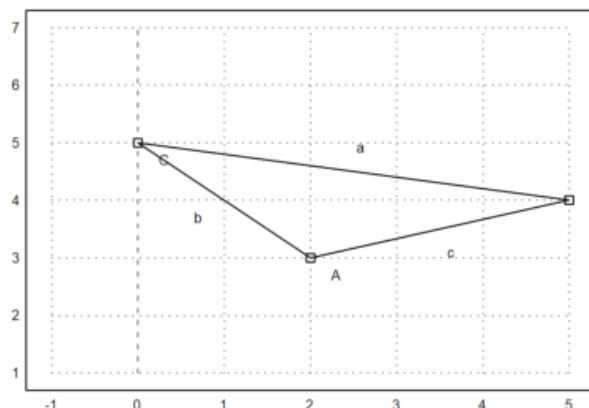
```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$$\frac{\sqrt{2} \left(\frac{9y}{2} - \frac{3x}{2} + \frac{9}{2}\right)}{3\sqrt{5}}$$

$$\frac{21}{2^{\frac{3}{2}}\sqrt{5}}$$

contoh lain dari materi trigonometri rasional

```
>A &:=[2,3]; B &:=[5,4]; C &:=[0,5]; ...
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;
```



```

>$distance(A,B)

$$\sqrt{10}$$


>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,

$$10$$


$$8$$


$$26$$


>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180) ()

$$\arccos\left(\frac{14}{\sqrt{10}\sqrt{26}}\right)$$


29.7448812969

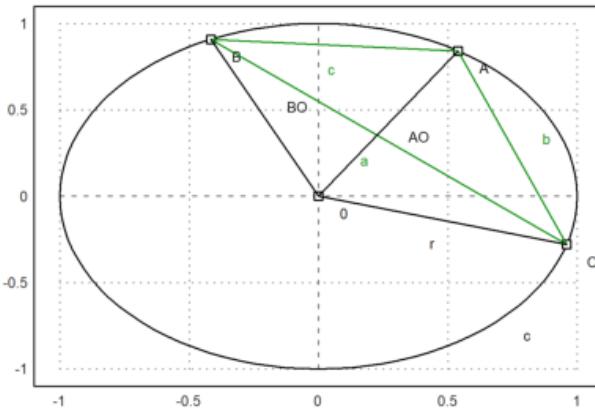
```

Aturan penyebaran 3 kali lipat

```

>setPlotRange(1); ...
color(1); plotCircle(circleWithCenter([0,0],1)); ...
A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...
plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
color(1); O:=[0,0]; plotPoint(O,"O"); ...
plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
insimg;

```



```

>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); $rabc

```

$$\frac{abc}{c^2 - 2bc + a} \frac{(-2c - 2b) + b^2 + a^2}{}$$

```

>function periradius(a,b,c) &= rabc;
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
>periradius(a,b,c)

```

1

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

0

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:

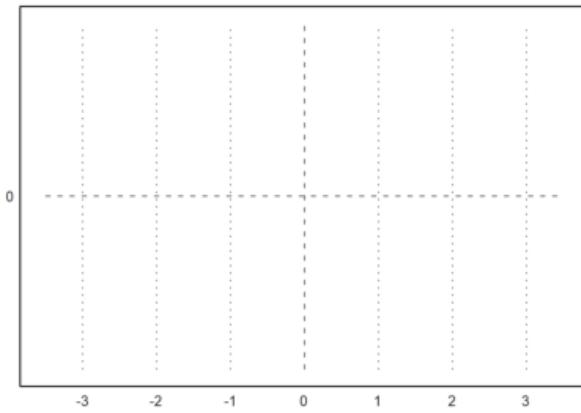
- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah $(360/n)$.
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan $(360/n)$.
- Untuk n ganjil, pilih salah satu titik sudut adalah di atas.
- 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.

Penyelesaian :

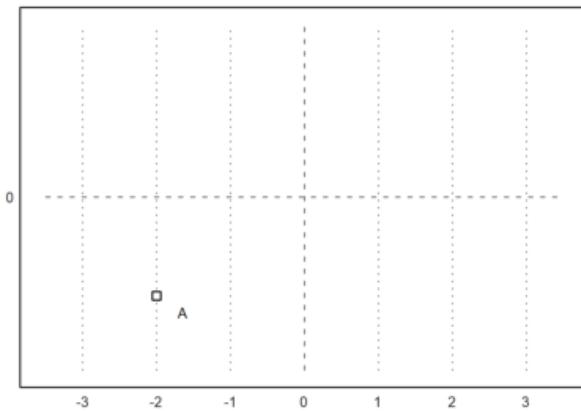
```
>load geometry
```

Numerical and symbolic geometry.

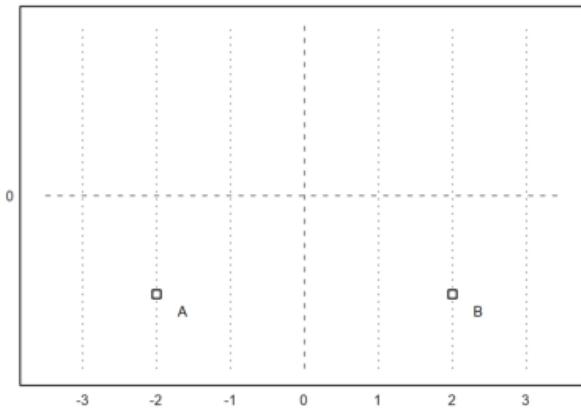
```
>setPlotRange(-3.5,3.5,-3.5,3.5):
```



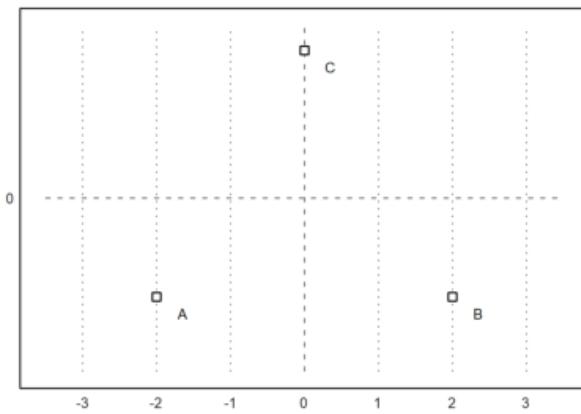
```
>A=[-2,-2]; plotPoint(A,"A"):
```



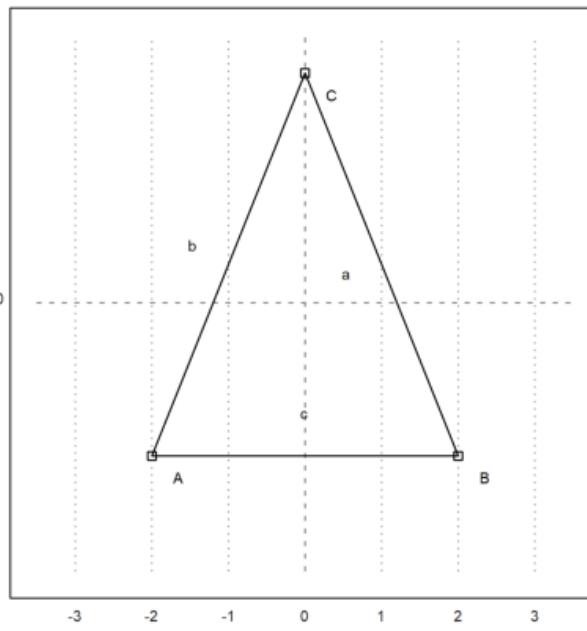
```
>B=[2,-2]; plotPoint(B,"B"):
```



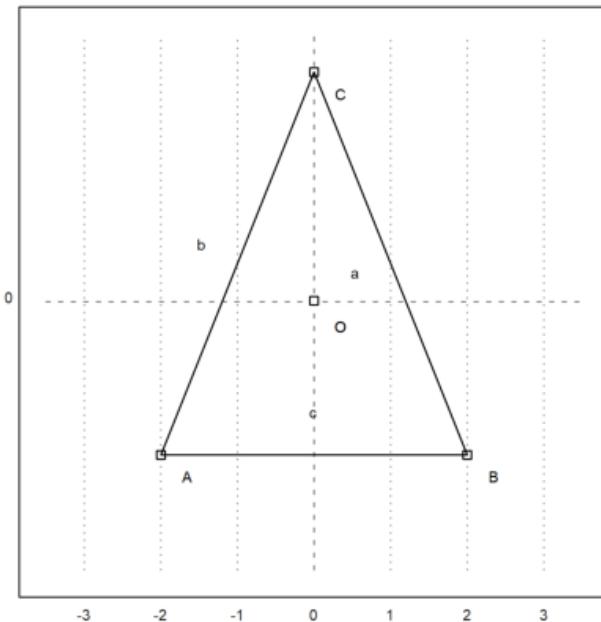
```
>C=[0,3]; plotPoint(C,"C"):
```



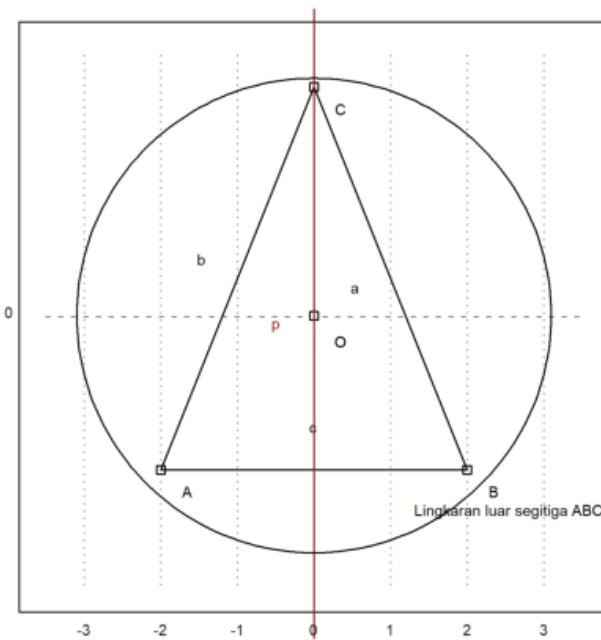
```
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
>plotSegment(A,C,"b");
>aspect(1):
```



```
>c=circleThrough(A,B,C);
>R=getCircleRadius(c);
>O=getCircleCenter(c);
>plotPoint(O,"O"):
```



```
>l=angleBisector(A,C,B);
>color(2); plotLine(l); color(1);
>plotCircle(c,"Lingkaran luar segitiga ABC"):
```



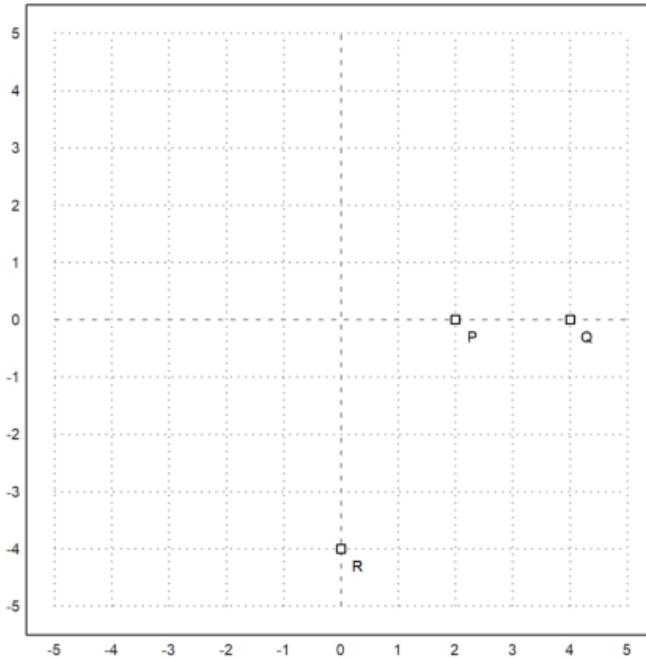
2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

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.

Penyelesaian :

```
>load geometry;
>setPlotRange(5); P=[2,0]; Q=[4,0]; R=[0,-4];
>plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R"):
```

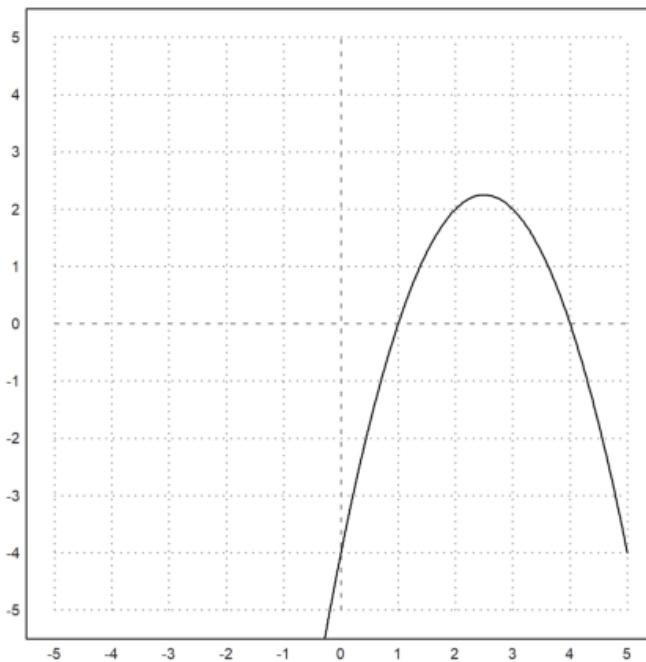


```
>sol &= solve([a+b=-c, 16*a+4*b=-c, c=-4], [a,b,c])
[[a = -1, b = 5, c = -4]]
```

Sehingga didapatkan nilai $a = -1$, $b = 5$ dan $c = -4$

```
>function y&=-x^2+5*x-4
           2
      - x    + 5 x - 4
```

```
>plot2d("-x^2+5*x-4", -5, 5, -5, 5):
```



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.

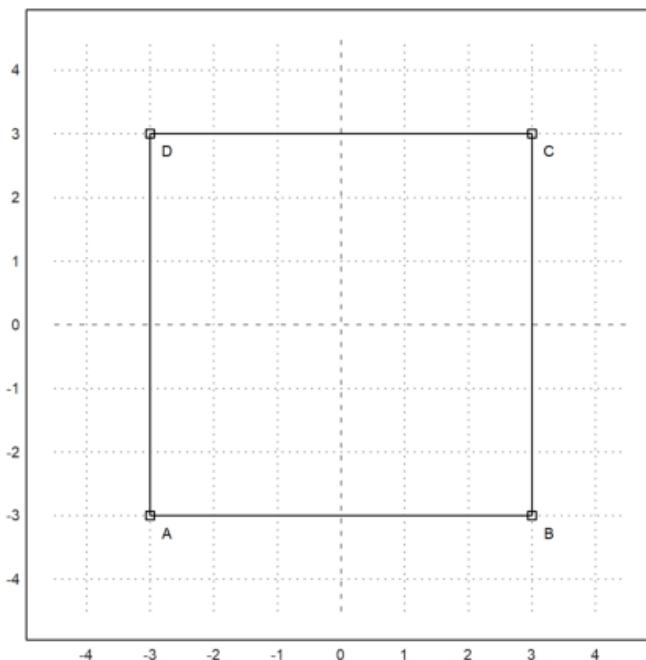
- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

Penyelesaian :

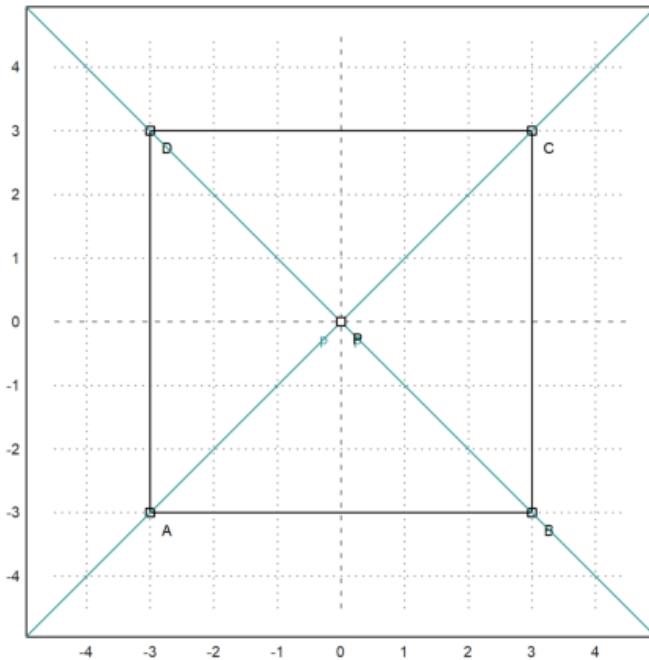
```
>load geometry
```

Numerical and symbolic geometry.

```
>setPlotRange(-4.5,4.5,-4.5,4.5);
>A=[-3,-3]; plotPoint(A,"A");
>B=[3,-3]; plotPoint(B,"B");
>C=[3,3]; plotPoint(C,"C");
>D=[-3,3]; plotPoint(D,"D");
>plotSegment(A,B,"");
>plotSegment(B,C,"");
>plotSegment(C,D,"");
>plotSegment(A,D,"");
>aspect(1):
```

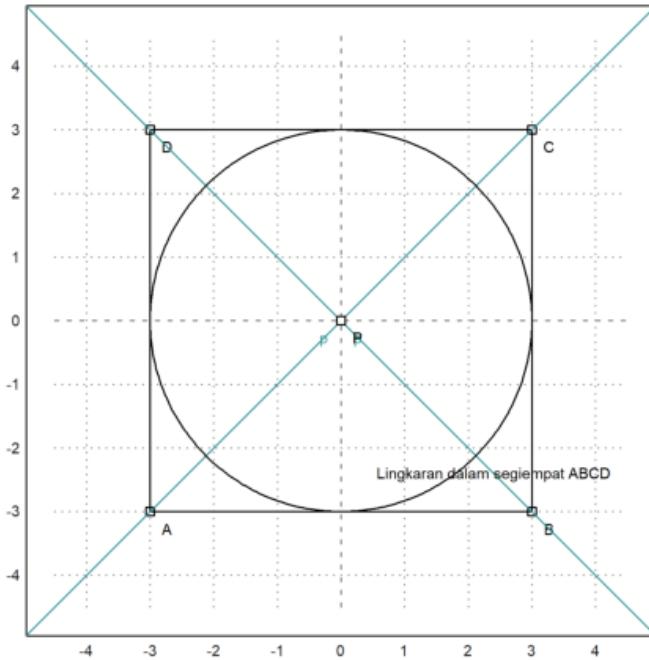


```
>l=angleBisector(A,B,C);
>m=angleBisector(B,C,D);
>P=lineIntersection(l,m);
>color(5); plotLine(l); plotLine(m); color(1);
>plotPoint(P,"P"):
```



Dari gambar diatas terlihat bahwa keempat garis bagi sudutnya bertemu di satu titik yaitu titik P.

```
>r=norm(P-projectToLine(P,lineThrough(A,B)));
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat ABCD");
```



Dari gambar diatas, terlihat bahwa sisi-sisinya merupakan garis singgung lingkaran yang sama yaitu lingkaran dalam segiempat.

Akan ditunjukkan bahwa hasil kali panjang sisi-sisi yang berhadapan sama.

```
>AB=norm(A-B) //panjang sisi AB
```

6

```
>CD=norm(C-D) //panjang sisi CD
```

6

```
>AD=norm(A-D) //panjang sisi AD
```

```
>BC=norm(B-C) //panjang sisi BC
```

```
>AB.CD
```

```
>AD.BC
```

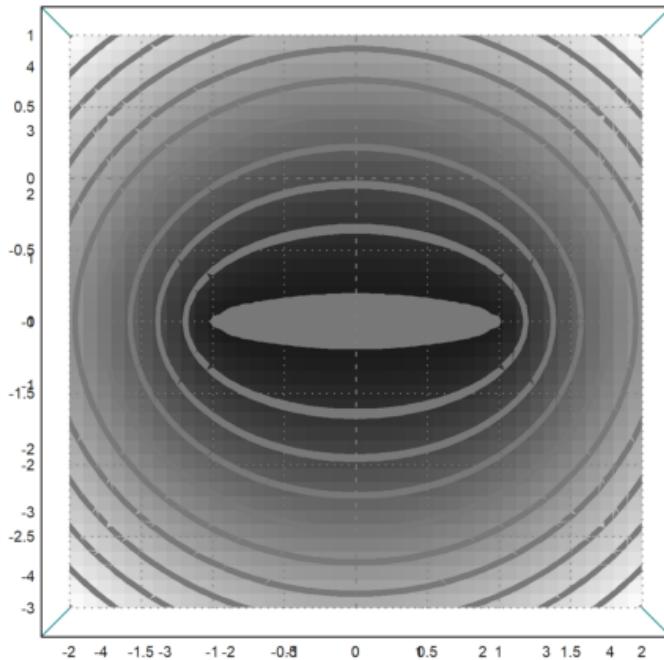
Terbukti bahwa hasil kali panjang sisi-sisi yang berhadapan sama yaitu 36. Jadi dapat dipastikan bahwa segiempat tersebut merupakan segiempat garis singgung.

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).

Penyelesaian :

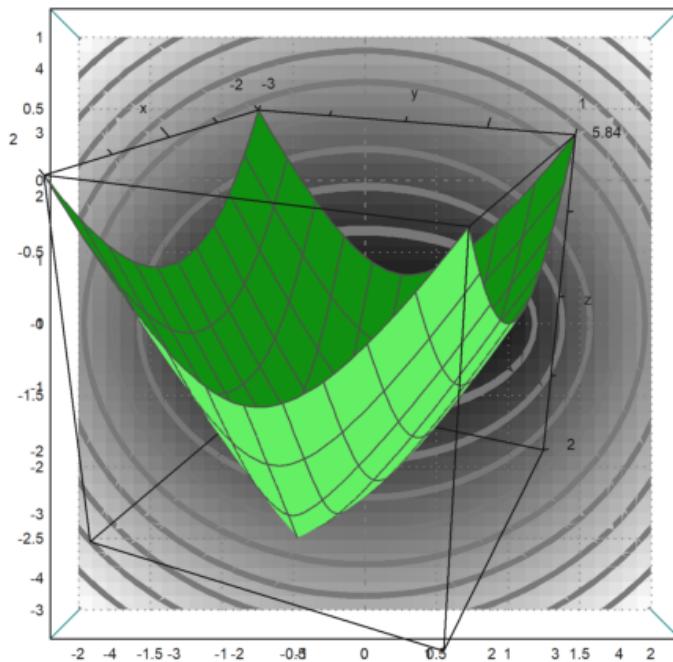
Diketahui kedua titik fokus $P = [-1, -1]$ dan $Q = [1, -1]$

```
>P=[-1,-1]; Q=[1,-1];
>function d1(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)
>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)+sqrt((x-Q[1])^2+(y-Q[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



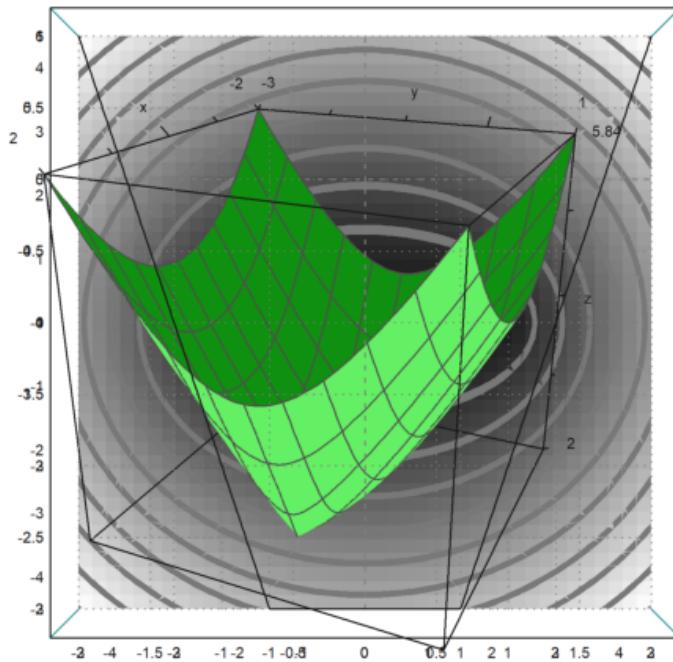
Grafik yang lebih menarik.

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Batasan ke garis PQ.

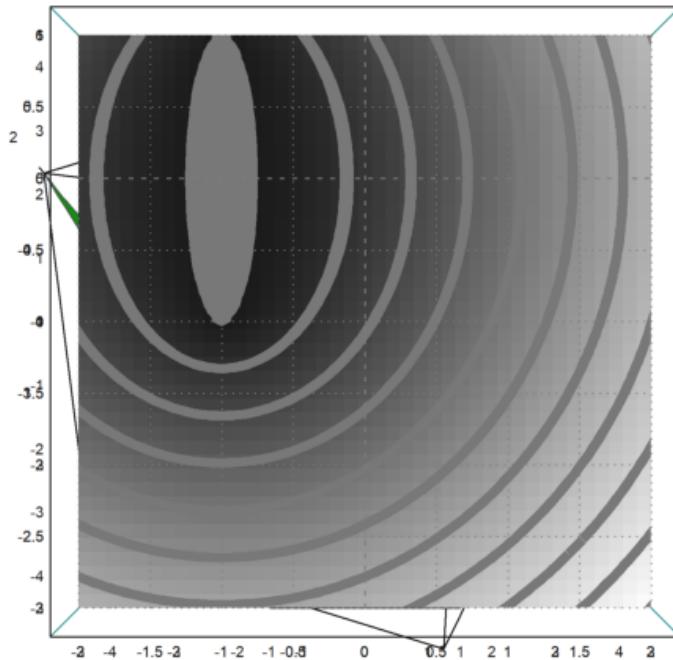
```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



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).

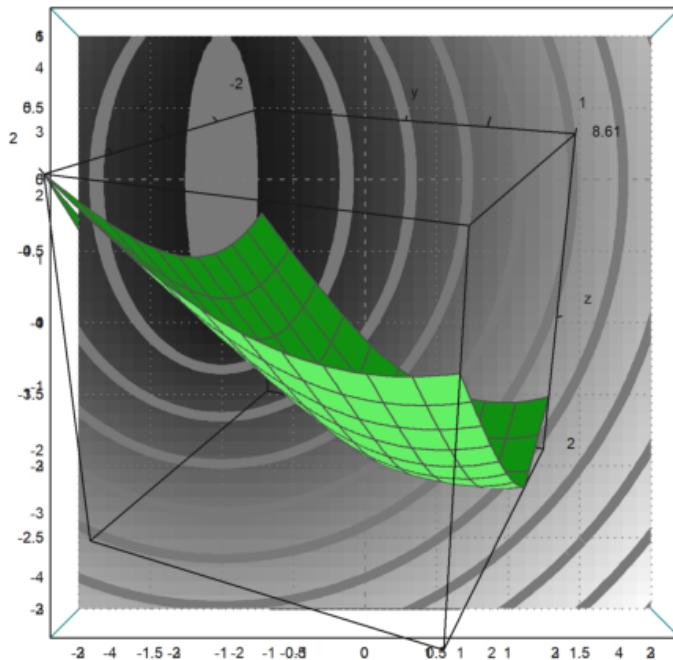
Penyelesaian :

```
>P=[-1,-1]; Q=[1,-1];
>function d1(x,y):=sqrt((x-p[1])^2+(y-p[2])^2)
>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)+sqrt((x+Q[1])^2+(y+Q[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```

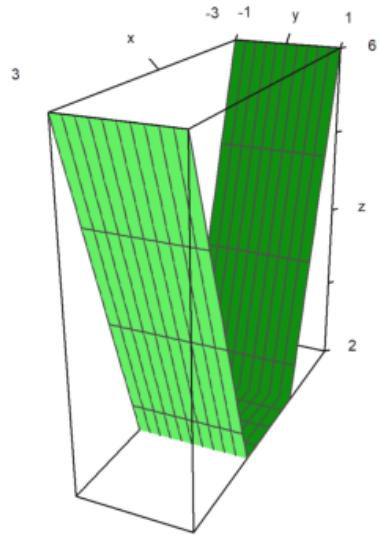


Grafik yang lebih menarik.

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1) :
```



```
>plot3d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3) :
```



Tugas Individu Pekan13-14_Muhammad Lutfi Ramadhan_23030630021

Nama : Muhammad Lutfi Ramadhan
NIM : 23030630021
Kelas : Matematika B 2023

EMT untuk Statistika

Di buku catatan ini, kami mendemonstrasikan plot statistik utama, pengujian, dan distribusi di Euler.

Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar statistik. Jadi, Anda mungkin memerlukan latar belakang untuk memahami detailnya.

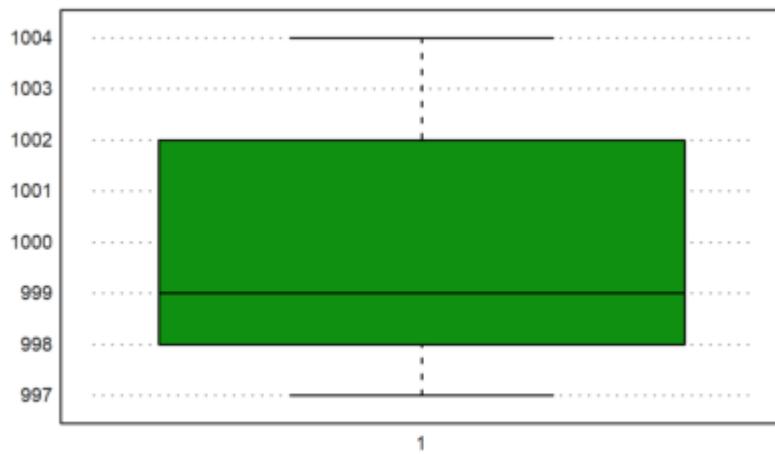
Asumsikan pengukuran berikut. Kami ingin menghitung nilai rata-rata dan deviasi standar yang diukur.

```
>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 datanya. Dalam kasus kami, tidak ada outlier.

```
>aspect(1.75); boxplot(M) :
```



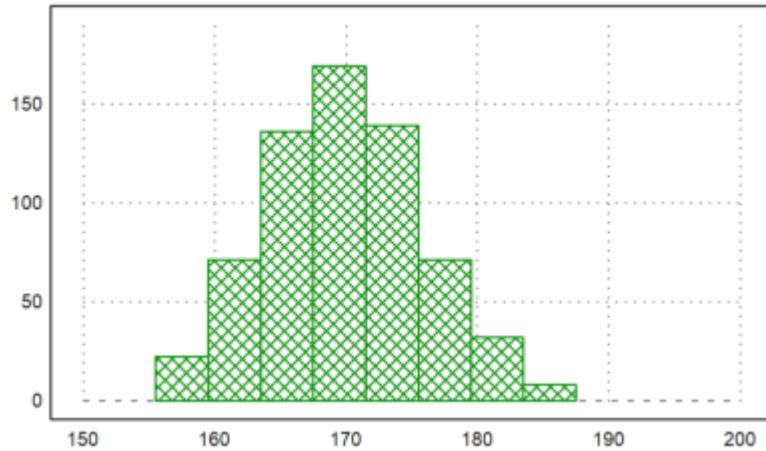
aspect (1.75) digunakan untuk mengatur rasio aspek dari plot (perbandingan antara lebar dan tinggi). boxplot(M) digunakan untuk membuat boxplot atau diagram kotak dari data di dalam variabel M. Boxplot adalah visualisasi statistik yang menunjukkan persebaran data, termasuk nilai minimum, median, dan nilai maksimum.

Contoh, kita asumsikan jumlah pria berikut dalam rentang ukuran tertentu.

```
>r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Berikut adalah alur pendistribusianya.

```
>plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="\\"/") :
```



Kita bisa memasukkan data mentah tersebut ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal jangkauan, akhir jangkauan, jumlah pria dalam jangkauan.

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 memerlukan nilai rata-rata dan statistik ukuran lainnya, kita perlu menghitung titik tengah rentang tersebut. Kita bisa menggunakan dua kolom pertama tabel kita untuk ini.

Sumbol "|" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan opsi "labc" untuk menentukan header kolom.

```
>(T[,1]+T[,2])/2
```

```
157.5
161.5
165.5
169.5
173.5
177.5
181.5
185.5
```

```
>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 mean dan deviasi sampel dengan frekuensi tertentu.

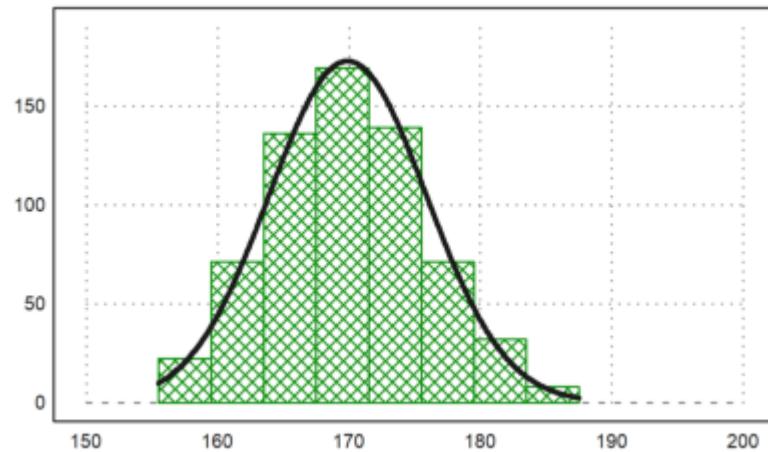
```
>{m,d}=meandev(M,v); m, d,
```

```
169.901234568  
5.98912964449
```

Mari kita tambahkan distribusi nilai normal ke diagram batang di atas. Rumus distribusi normal dengan mean m dan simpangan baku d adalah:

Karena nilainya antara 0 dan 1, maka untuk memplotnya pada bar plot 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

Di direktori buku catatan ini Anda menemukan file dengan tabel. Data tersebut merupakan hasil survei. Berikut adalah empat baris pertama file tersebut. Datanya berasal dari buku online Jerman "Einführung in die Statistik mit R" oleh A. Handl.

```
>printfile("table.dat", 4);
```

```
Person Sex Age Titanic Evaluation Tip Problem  
1 m 30 n . 1.80 n  
2 f 23 y g 1.80 n  
3 f 26 y g 1.80 y
```

Tabel berisi 7 kolom angka atau token (string). Kami ingin membaca tabel dari file. Pertama, kami menggunakan terjemahan kami sendiri untuk tokennya.

Untuk ini, kami mendefinisikan kumpulan token. Fungsi `strtokens()` mendapatkan vektor string token dari string tertentu.

```
>mf:=["m", "f"]; yn:=[ "y", "n"]; ev:=strtokens("g vg m b vb");
```

Sekarang kita membaca tabel dengan terjemahan ini.

Argumen `tok2`, `tok4` dll. adalah terjemahan dari kolom tabel. Argumen ini tidak ada dalam daftar parameter `readtable()`, jadi Anda perlu menyediakannya dengan `"::"`.

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);  
>load over statistics;  
>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 "." mewakili nilai-nilai, yang tidak tersedia.

Jika kita tidak ingin menentukan token yang akan diterjemahkan terlebih dahulu, 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);
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

ctok=[2,4,5,7]: Ini adalah untuk menentukan kolom yang akan diambil yaitu kolom ke-2, ke-4, ke-5, dan ke-7.

```
>tok
```

```
m
n
f
y
g
vg
```

Tabel berisi entri dari file dengan token yang diterjemahkan ke dalam angka.

String khusus NA = "." diartikan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) di tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAval.

```
>MT[1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Berikut isi tabel dengan nomor yang belum diterjemahkan.

```
>writetable(MT,wc=5)
```

1	1	30	2	.	1.8	2
2	3	23	4	5	1.8	2
3	3	26	4	5	1.8	4
4	1	33	2	.	2.8	2
5	1	37	2	.	1.8	2
6	1	28	4	5	2.8	4
7	3	31	4	6	2.8	2
8	1	23	2	.	0.8	2
9	3	24	4	6	1.8	4

10	1	26	2	.	1.8	2
11	3	23	4	6	1.8	4
12	1	32	4	5	1.8	2
13	1	29	4	6	1.8	4
14	3	25	4	5	1.8	4
15	3	31	4	5	0.8	2
16	1	26	4	5	2.8	2
17	1	37	2	.	3.8	2
18	1	38	4	5	.	2
19	3	29	2	.	3.8	2
20	3	28	4	6	1.8	2
21	3	28	4	1	2.8	4
22	3	28	4	6	1.8	4
23	3	38	4	5	2.8	2
24	3	27	4	1	1.8	4
25	1	27	2	.	2.8	4

Untuk kenyamanan, Anda dapat memasukkan keluaran readtable() ke dalam daftar.

```
>Table={{readtable("table.dat",ctok=ctok)};}
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
  if filename!=none then open(filename,"r"); endif;
```

Dengan menggunakan kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dll. atau menggunakan tabel daftar.

```
>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 tabel, melewatkkan baris apa pun dengan nilai NAN ("." dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

```
>{c,i}=tablecol(MT,[5,6]);
```

Kita bisa 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 suatu kolom atau nilai statistik lainnya.

```
>mean(tablecol(MT,6))
```

2.175

Fungsi getstatistics() mengembalikan elemen dalam vektor, dan jumlahnya. Kami menerapkannya pada nilai "m" dan "f" di kolom kedua tabel kami.

```
>{xu,count}=getstatistics(tablecol(MT,2)); xu, count,
```

[1, 3]
[12, 13]

Kita bisa mencetak hasilnya di tabel baru.

```
>writetable(count',labr=tok[xu])
```

m	12
f	13

Fungsi selectable() mengembalikan tabel baru dengan nilai dalam satu kolom yang dipilih dari vektor indeks. Pertama kita mencari indeks dari dua nilai kita di tabel token.

```
>v:=indexof(tok,["g","vg"])
```

```
[5, 6]
```

Sekarang kita dapat memilih baris tabel, yang memiliki salah satu nilai v pada baris ke-5.

```
>MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5);
```

Sekarang kita dapat mencetak tabel, dengan nilai yang diekstraksi 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 selanjutnya, kami ingin menghubungkan dua kolom tabel. Jadi kita ekstrak kolom 2 dan 4 dan urutkan tabelnya.

```
>i=sortedrows(MT,[2,4]); ...  
writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)
```

```
m      n  
m      n  
m      n  
m      n  
m      n  
m      n  
m      n  
m      n  
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  
f      y  
f      y  
f      y  
f      y
```

Dengan getstatistics(), kita juga bisa menghubungkan jumlah dalam dua kolom tabel satu sama lain.

```
>MT24=tablecol(MT, [2,4]); ...
{xu1,xu2,count}=getstatistics(MT24[1],MT24[2]); ...
writetable(count,labr=tok[xu1],labc=tok[xu2])
```

	n	Y
m	7	5
f	1	12

Sebuah tabel dapat ditulis ke file.

```
>filename="test.dat"; ...
writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
```

Kemudian kita bisa membaca tabel dari file tersebut.

```
>{MT2,hd,tok2,hdr}=readtable(filename,>clabs,>rlabs); ...
writetable(MT2,labr=hdr,labc=hd)
```

	n	Y
m	7	5
f	1	12

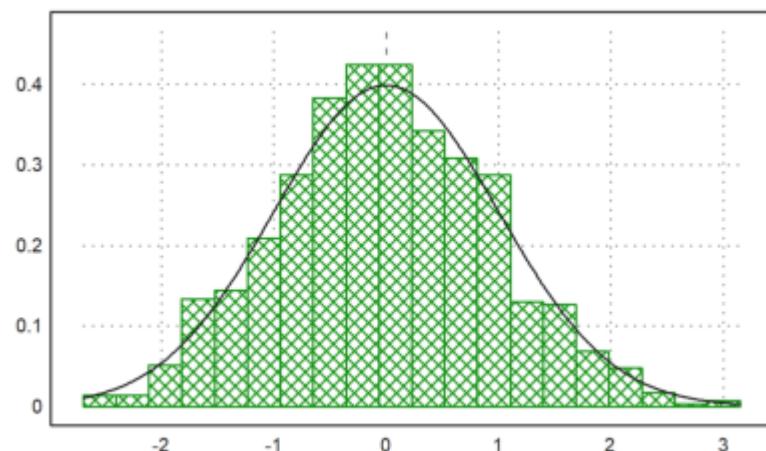
Dan hapus filenya.

```
>fileremove(filename);
```

Distribusi

Dengan plot2d, ada metode yang sangat mudah untuk memplot sebaran data eksperimen.

```
>p=normal(1,1000);
>plot2d(p,distribution=20,style="\/");
>plot2d("qnormal(x,0,1)",add=1):
```



p=normal(1,1000); digunakan untuk menciptakan 1000 sampel acak yang terdistribusi normal dengan mean (rata-rata) 1 dan standar deviasi 1000.

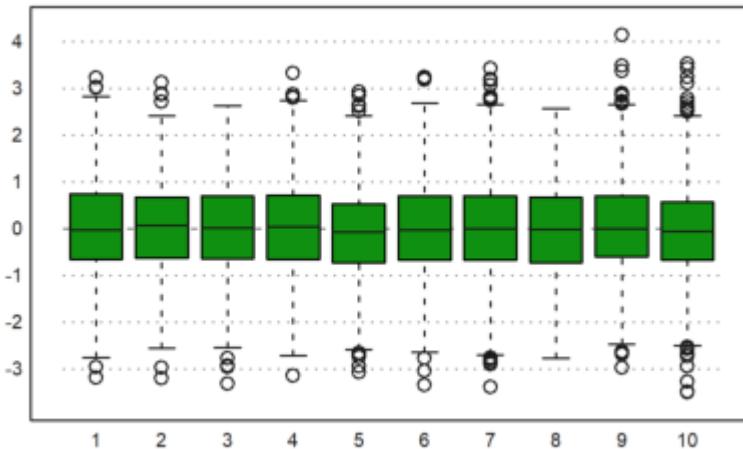
plot2d("qnormal(x,0,1)",add=1);

digunakan untuk menambahkan plot dari distribusi normal standar (dengan mean 0 dan standar deviasi 1) ke grafik yang sama. Fungsi qnormal(x,0,1) mengacu pada distribusi kumulatif dari variabel acak normal standar. add=1 menunjukkan bahwa grafik ini harus ditambahkan ke grafik yang sudah ada, bukan dibuat baru.

Perlu diperhatikan perbedaan antara bar plot (sampel) dan kurva normal (distribusi sebenarnya). Masukkan kembali ketiga perintah untuk melihat hasil pengambilan sampel lainnya.

Berikut adalah perbandingan 10 simulasi dari 1000 nilai terdistribusi normal menggunakan apa yang disebut plot kotak. Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, serta outlier.

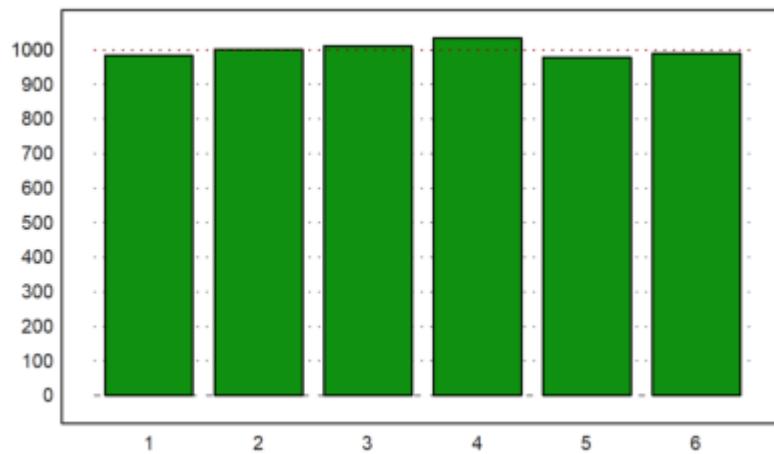
```
>p=normal(10,1000); boxplot(p):
```



Untuk menghasilkan bilangan bulat acak, Euler memiliki `intrandom`. Mari kita simulasi lemparan dadu dan plot distribusinya.

Kita menggunakan fungsi `getmultiplicities(v,x)`, yang menghitung seberapa sering elemen v muncul di x. Kemudian kita plot hasilnya menggunakan `barplot()`.

```
>k=intrandom(1,6000,6); ...
columnsplot(getmultiplicities(1:6,k)); ...
ygrid(1000,color=red):
```



Meskipun `intrandom(n,m,k)` mengembalikan bilangan bulat yang terdistribusi secara seragam dari 1 hingga k, distribusi bilangan bulat lainnya dapat digunakan dengan `randpint()`.

Dalam 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. Lihat referensinya.

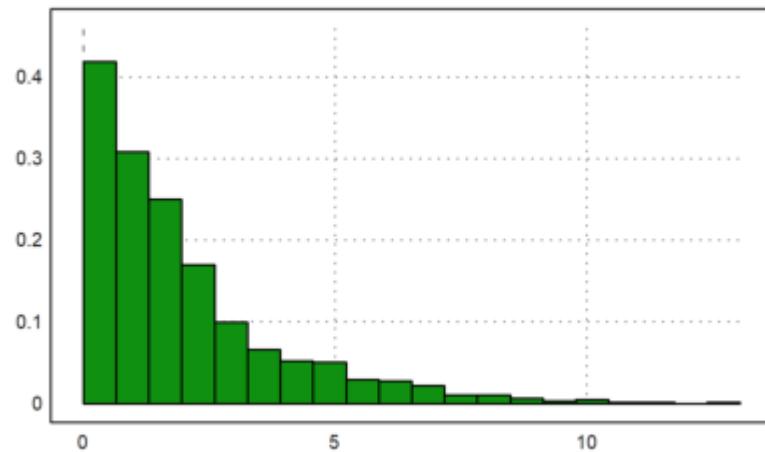
Misalnya, kita mencoba distribusi eksponensial. Variabel acak kontinu X dikatakan berdistribusi eksponensial, jika PDF-nya diberikan oleh

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

with parameter

$$\lambda = \frac{1}{\mu}, \quad \mu \text{ is the mean, and denoted by } X \sim \text{Exponential}(\lambda).$$

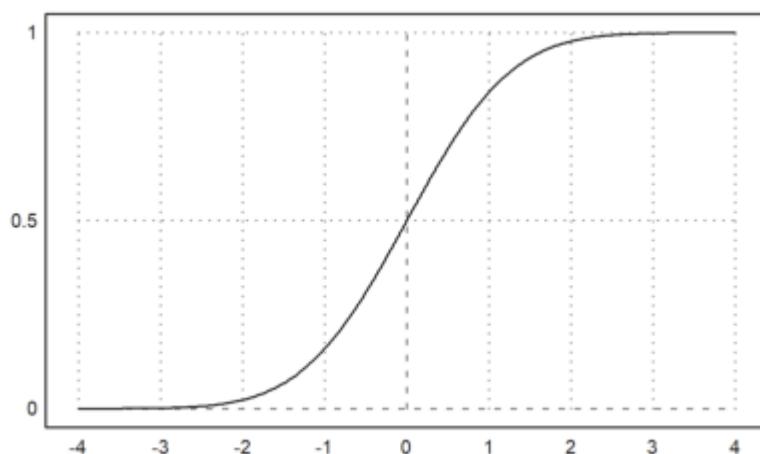
```
>plot2d(randexponential(1,1000,2),>distribution):
```



Parameter pertama (1) adalah lambda, yang merupakan parameter distribusi eksponensial. Parameter kedua (1000) menunjukkan jumlah angka acak yang dihasilkan. Parameter ketiga (2) bisa menunjukkan dimensi atau bentuk output.

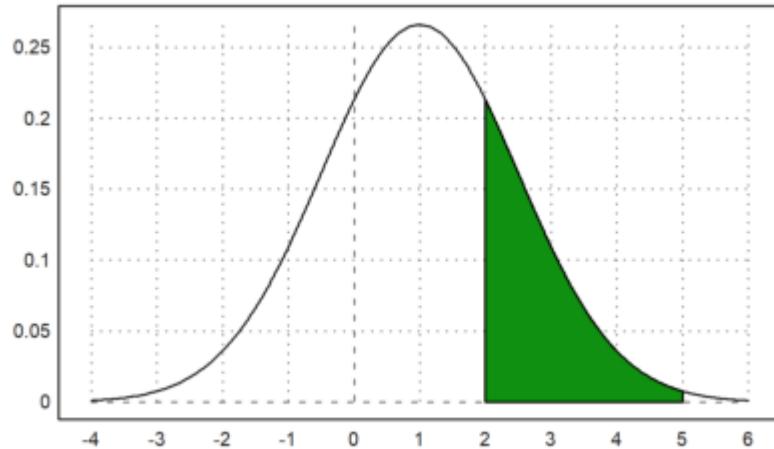
Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversnya.

```
>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):
```



Peluang berada di kawasan hijau adalah sebagai berikut.

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

0.248662156979

Ini dapat dihitung secara numerik dengan integral berikut.

```
>gauss("qnormal(x,1,1.5)",2,5)
```

0.248662156979

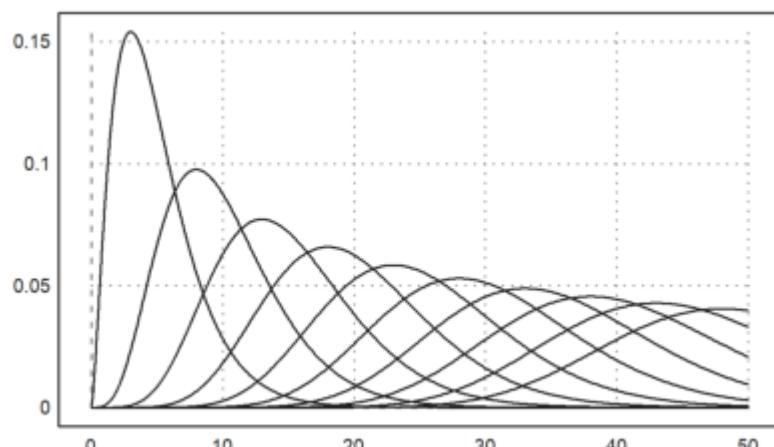
Mari kita bandingkan distribusi binomial dengan distribusi normal yang mean dan deviasinya sama. Fungsi `invbindis()` menyelesaikan interpolasi linier antara nilai integer.

```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

525.516721219
526.007419394

Fungsi `qdis()` adalah kepadatan distribusi chi-kuadrat. Seperti biasa, Euler memetakan vektor ke fungsi ini. Dengan demikian kita mendapatkan plot semua distribusi chi-kuadrat dengan derajat 5 sampai 30 dengan mudah dengan cara berikut.

```
>plot2d("qchidis(x,(5:5:50))",0,50):
```



Euler memiliki fungsi akurat untuk mengevaluasi distribusi. Mari kita periksa chidis() dengan integral.

Penamaannya mencoba untuk konsisten. Misalnya.,

- distribusi chi-kuadratnya adalah chidis(),
- fungsi kebalikannya adalah invchidis(),
- kepadatannya 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

Distribusi diskret adalah jenis distribusi probabilitas yang digunakan untuk variabel acak diskret, yaitu variabel yang hanya dapat memiliki nilai tertentu, biasanya dalam bentuk bilangan bulat.

Untuk menentukan distribusi diskrit Anda sendiri, Anda dapat menggunakan metode berikut.

Pertama kita atur 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]
```

Perintah ini menggunakan operator | dan + untuk membuat nilai dalam variabel wd.

1:6 Ini menghasilkan vektor [1, 2, 3, 4, 5, 6].

(1:6) + [-0.01, 0.01, 0, 0, 0, 0]: Operasi ini menambahkan kedua vektor elemen per elemen.
Hasilnya:

$$[1 - 0.01, 2 + 0.01, 3, 4, 5, 6] = [0.99, 2.01, 3, 4, 5, 6]$$

[1-0.01,2+0.01,3,4,5,6]=[0.99,2.01,3,4,5,6]/6 Membagi setiap elemen hasil penjumlahan tadi dengan 6.
Hasilnya:

$$\left[\frac{0.99}{6}, \frac{2.01}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}, \frac{6}{6} \right] = [0.165, 0.335, 0.5, 0.6667, 0.8333, 1]$$

Artinya dengan probabilitas wd[i+1]-wd[i] kita menghasilkan nilai acak i.

Ini hampir merupakan distribusi yang seragam. Mari kita tentukan generator nomor acak untuk ini. Fungsi find(v,x) mencari nilai x pada vektor v. Fungsi ini juga berfungsi untuk vektor x.

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahannya sangat halus sehingga kita hanya melihatnya dengan banyak iterasi.

Fungsi wrongdice mengembalikan sebuah matriks berukuran n x m, di mana setiap elemen dari matriks ini adalah indeks posisi dari elemen wd yang paling sesuai (atau mendekati) nilai acak dari random(n, m).

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```

```
Variable or function wd not found.  
Try "trace errors" to inspect local variables after errors.  
wrongdice:  
    useglobal; return find(wd,random(n,m))  
Error in:
```

```
... nsplot(getmultiplicities(1:6,wrongdice(1,1000000))): ...
```

Hasil columnsplot akan menunjukkan frekuensi relatif dari setiap angka (1 hingga 6), yang memungkinkan Anda untuk melihat apakah distribusi itu merata atau tidak.

Berikut adalah fungsi sederhana untuk memeriksa keseragaman distribusi nilai 1...K dalam v. Kita menerima hasilnya, jika untuk semua frekuensi

$$\left| f_i - \frac{1}{K} \right| < \frac{\delta}{\sqrt{n}}.$$

Metode tersebut merupakan metode statistik untuk menguji keseragaman distribusi. Distribusi dianggap seragam jika frekuensi setiap nilai dalam v mendekati frekuensi ideal 1/K, dengan deviasi yang tidak melebihi batas toleransi.

```
>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 fungsinya menolak distribusi seragam.

```
>checkrandom(wrongdice(1,1000000))
```

0

Dan ia menerima generator acak bawaan.

Manual:

- Asumsi dadu, maka peluang setiap sisi = 1/6
Dalam 1 juta lemparan maka

$$1000000 \times \frac{1}{6} \approx 166667$$

- Frekuensi setiap sisi fr. Proporsi tiap sisi = fr/n Misalkan frekuensi munculnya angka adalah [160000, 170000, 180000, 150000, 170000, 170000]

$$[160000, 170000, 180000, 150000, 170000, 170000]$$

Maka proporsi setiap angka:

$$\frac{[160000, 170000, 180000, 150000, 170000, 170000]}{1000000}$$

$$[0.16, 0.17, 0.18, 0.15, 0.17, 0.17]$$

- Deviasi maksimum $fn/n - 1/K$ $\frac{1}{6} = 0.1667$

$$\frac{1}{K} = \frac{1}{6} = 0.1667$$

$$([0.16, 0.17, 0.18, 0.15, 0.17, 0.17] - 0.1667)$$

$$\max(-0.0067, 0.0033, 0.0133, -0.0167, 0.0033, 0.0033) = 0.0133$$

- Bandingkan dengan batas toleransi. Batas= $\frac{\delta}{\sqrt{n}} = \frac{1}{\sqrt{1000000}} = \frac{1}{1000} = 0.001$

$$Batas = \frac{\delta}{\sqrt{n}} = \frac{1}{\sqrt{1000000}} = \frac{1}{1000} = 0.001$$

$$0.0133 > 0.001$$

Hasil 0 di sini mengindikasikan bahwa fungsi checkrandom telah menentukan bahwa distribusi tidak seragam.

```
>checkrandom(intrandom(1,1000000,6))
```

1

checkrandom mengembalikan 1 atau true yang berarti bahwa distribusi dari 1 juta bilangan acak rentang 1 sampai 6 dianggap cukup seragam dalam batas toleransi yang ditetapkan.

Kita dapat menghitung distribusi binomial. Pertama ada binomials() , yang mengembalikan probabilitas i atau kurang hit dari n percobaan.

Misal kita akan menghitung probabilitas dari distribusi binomial di mana terdapat 1000 percobaan (misalnya, 1000 kali pelemparan koin), dengan probabilitas sukses pada setiap percobaan sebesar 0.4, dan kita ingin mengetahui probabilitas mendapatkan tepat 410 sukses.
Secara matematis, ini dihitung dengan rumus:

$$P(X \leq 410) = \binom{1000}{410} \cdot (0.4)^{410} \cdot (0.6)^{1000-410}$$

```
>bindis(410,1000,0.4)
```

0.751401349654

```
>bindis(4,10,0.6)
```

0.1662386176

Manual:

Secara matematis, ini dihitung dengan rumus:

$$P(X \leq 4) = \binom{10}{4} \cdot (0.6)^4 \cdot (0.4)^{10-4}$$

- Untuk $k = 0$ $P(X=0) = \binom{10}{0} \cdot (0.6)^0 \cdot (0.4)^{10} \approx 0.00010$

$$P(X = 0) = \binom{10}{0} \cdot (0.6)^0 \cdot (0.4)^{10} \approx 0.00010$$

- Untuk $k = 1$ $P(X = 1) = \binom{10}{1} \cdot (0.6)^1 \cdot (0.4)^9 \approx 0.00157$

$$P(X = 1) = \binom{10}{1} \cdot (0.6)^1 \cdot (0.4)^9 \approx 0.00157$$

- Untuk $k = 2$ $P(X = 2) = \binom{10}{2} \cdot (0.6)^2 \cdot (0.4)^8 \approx 0.01061$

$$P(X = 2) = \binom{10}{2} \cdot (0.6)^2 \cdot (0.4)^8 \approx 0.01061$$

- Untuk $k = 3$ $P(X = 3) = \binom{10}{3} \cdot (0.6)^3 \cdot (0.4)^7 \approx 0.04246$

$$P(X = 3) = \binom{10}{3} \cdot (0.6)^3 \cdot (0.4)^7 \approx 0.04246$$

- Untuk $k = 4$ $P(X = 4) = \binom{10}{4} \cdot (0.6)^4 \cdot (0.4)^6 \approx 0.11147$

$$P(X = 4) = \binom{10}{4} \cdot (0.6)^4 \cdot (0.4)^6 \approx 0.11147$$

Maka,

$$P(X \leq 4) = P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3) + P(X = 4)$$

$$P(X \leq 4) = 0.00010 + 0.00157 + 0.01061 + 0.04246 + 0.11147$$

$$P(X \leq 4) \approx 0.1662$$

Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p. Tingkat defaultnya adalah alfa.

Arti dari interval ini adalah jika p berada di luar interval, hasil pengamatan 410 dalam 1000 jarang terjadi.

```
>clopperpearson(410,1000)
```

[0.37932, 0.441212]

Perintah berikut adalah cara langsung untuk mendapatkan hasil di atas. Namun untuk n yang besar, penjumlahan langsungnya tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

0.751401349655

Omong-omong, invbinsum() menghitung kebalikan dari binomialsum().

```
>invbindis(0.75,1000,0.4)
```

409.932733047

Di Bridge, kami mengasumsikan 5 kartu beredar (dari 52) 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])
```

407	105	488
397	95	508
397	108	495
378	96	526
403	97	500
410	90	500
389	115	496
385	109	506
373	90	537
396	103	501

Merencanakan Data/ Plot Data

Untuk memetakan data, kami mencoba hasil pemilu Jerman sejak tahun 1990, diukur dalam jumlah 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 beberapa bagian, kami menggunakan rangkaian nama.

```
>P:=[ "CDU/CSU", "SPD", "FDP", "Gr", "Li"];
```

Mari kita cetak persentasenya dengan baik.

Pertama kita mengekstrak kolom yang diperlukan. Kolom 3 sampai 7 adalah kursi masing-masing partai, dan kolom 2 adalah jumlah kursi seluruhnya. Kolom 1 adalah tahun pemilihan.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Kemudian statistiknya kita cetak dalam bentuk tabel. Kami menggunakan nama sebagai header kolom, dan tahun sebagai header untuk baris. Lebar default untuk kolom adalah wc=10, tetapi kami lebih memilih keluaran yang lebih padat. Kolom akan diperluas untuk 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

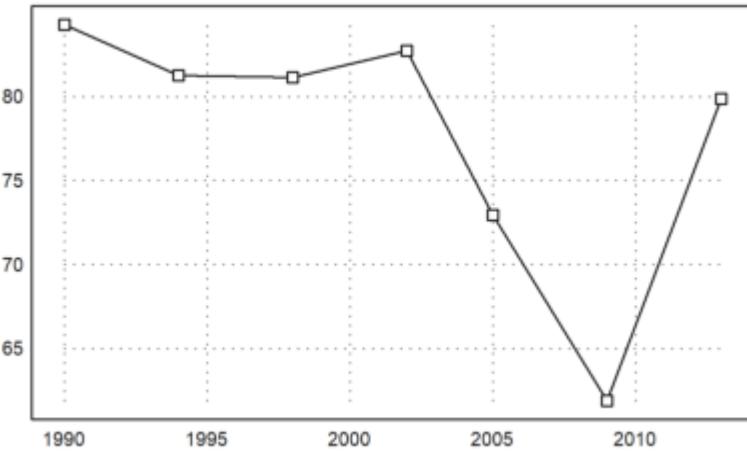
Perkalian matriks berikut ini menjumlahkan persentase dua partai besar yang menunjukkan bahwa partai-partai kecil berhasil memperoleh suara di parlemen hingga tahun 2009.

```
>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. Kami menggunakananya untuk menampilkan garis dan titik secara bersamaan. Alternatifnya adalah memanggil plot2d dua kali dengan >add.

```
>statplot(YT,BT1,"b"):
```

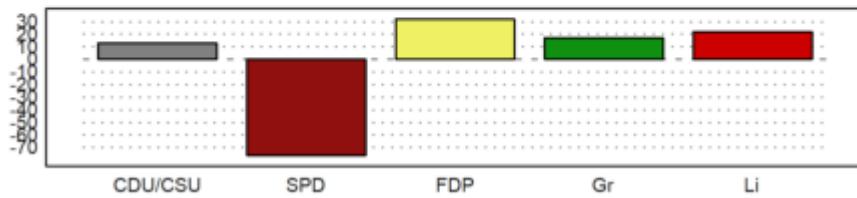


Tentukan beberapa warna untuk setiap pesta.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

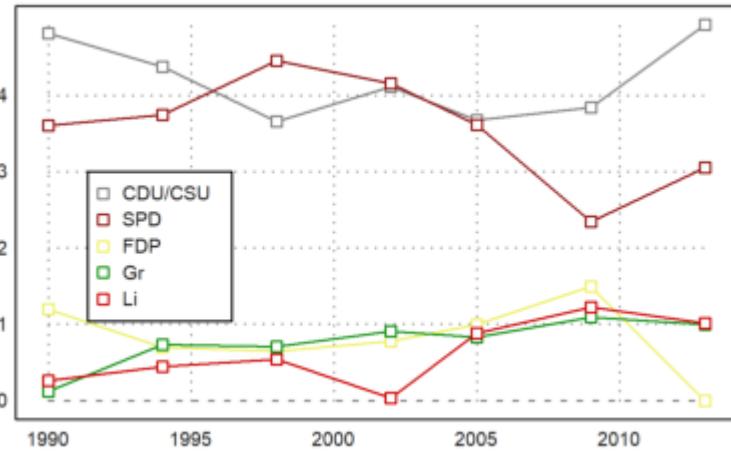
Sekarang kita bisa memplot hasil pemilu 2009 dan perubahannya menjadi satu plot dengan menggunakan gambar. Kita dapat menambahkan vektor kolom ke setiap plot.

```
>figure(2,1); ...
figure(1); columnsplot(BW[6,3:7],P,color=CP); ...
figure(2); columnsplot(BW[6,3:7]-BW[5,3:7],P,color=CP); ...
figure(0):
```



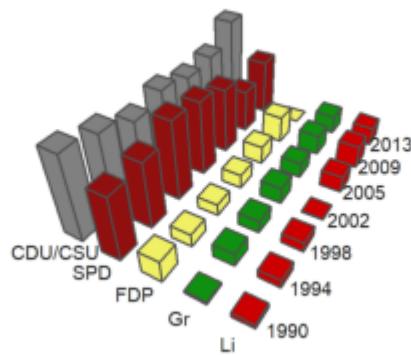
Plot data menggabungkan deretan data statistik dalam satu plot.

```
>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):
```



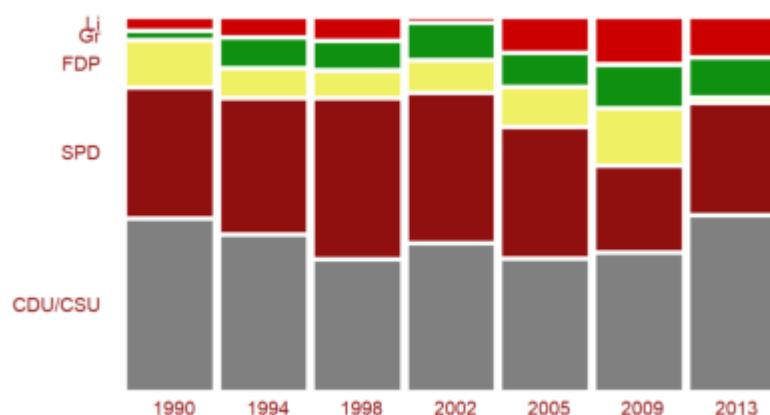
Plot kolom 3D memperlihatkan baris data statistik dalam bentuk kolom. Kami memberikan label untuk baris dan kolom. sudut adalah sudut pandang.

```
>columnplot3d(BT, scols=P, srows=YT, ...
  angle=30°, ccols=CP):
```



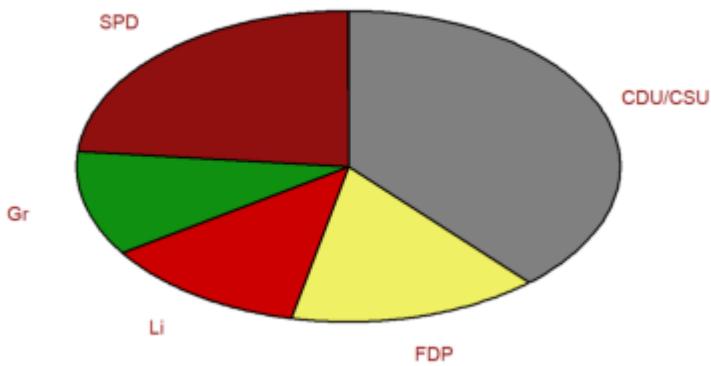
Representasi lainnya adalah plot mosaik. Perhatikan bahwa kolom plot mewakili kolom matriks di sini. Karena panjang label CDU/CSU, kami mengambil jendela yang lebih kecil dari biasanya.

```
>shrinkwindow(>smaller); ...
mosaicplot(BT', srows=YT, scols=P, color=CP, style="#"); ...
shrinkwindow():
```



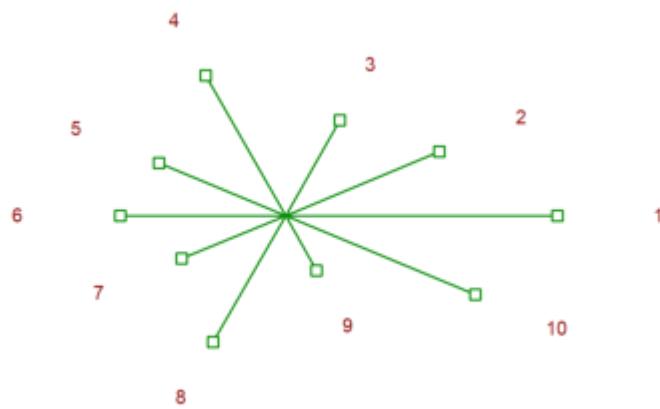
Kita juga bisa membuat diagram lingkaran. Karena hitam dan kuning membentuk koalisi, kami menyusun ulang elemen-elemennya.

```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```



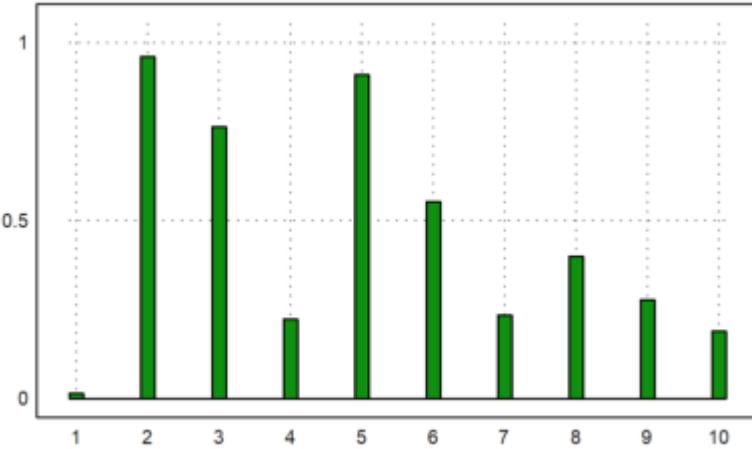
Ini adalah jenis plot lainnya.

```
>starplot(normal(1,10)+4,lab=1:10,>rays):
```



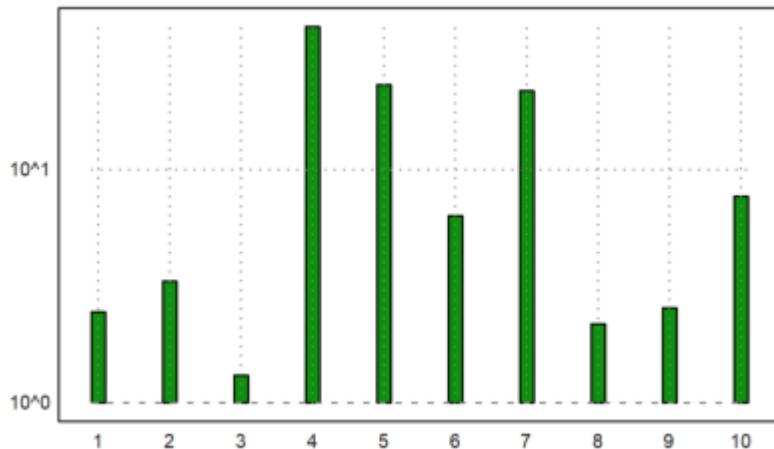
Beberapa plot di plot2d bagus untuk statika. Berikut adalah plot impuls dari data acak, terdistribusi secara seragam di [0,1].

```
>plot2d(makeimpulse(1:10,random(1,10)),>bar):
```



Namun untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

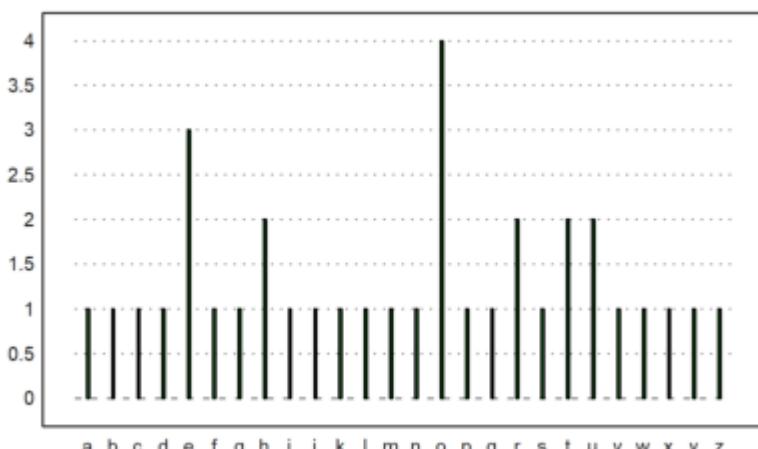
```
>logimpulseplot(1:10,-log(random(1,10))*10):
```



Fungsi Columnplot() lebih mudah digunakan, karena hanya memerlukan vektor nilai. Selain itu, ia dapat mengatur labelnya ke apa pun yang kita inginkan, kami telah mendemonstrasikannya di tutorial ini.

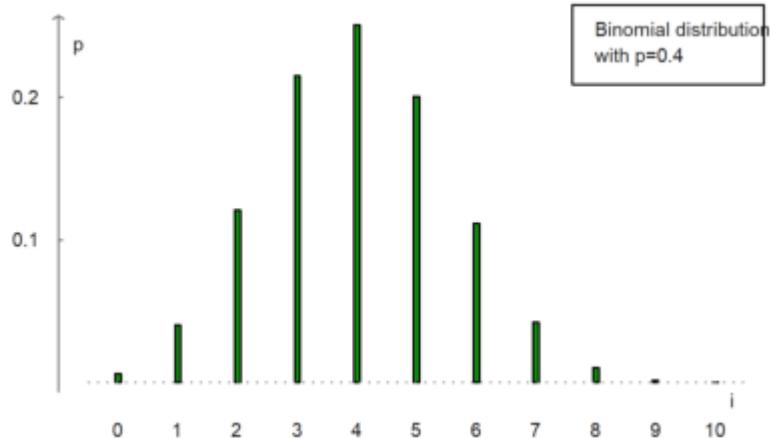
Ini adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan membuat statistik.

```
>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):
```



Dimungkinkan juga untuk mengatur 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 cara memplot frekuensi bilangan dalam suatu vektor.

Kami membuat vektor bilangan acak bilangan bulat 1 hingga 6.

```
>v:=intrandom(1,10,10)
```

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

Kemudian ekstrak nomor unik di v.

```
>vu:=unique(v)
```

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

Dan plot frekuensi dalam plot kolom.

```
>columnsplot(getmultiplicities(vu,v),lab=vu,style="/"):
```

Kami ingin mendemonstrasikan fungsi distribusi nilai empiris.

```
>x=normal(1,20);
```

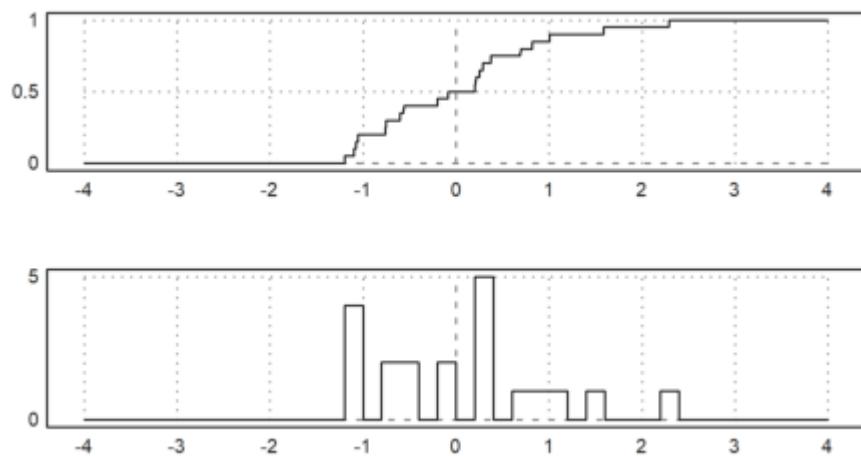
Fungsi empdist(x,vs) memerlukan array nilai yang diurutkan. Jadi kita harus mengurutkan x sebelum kita dapat menggunakannya.

```
>xs=sort(x);
```

Kemudian kita plot distribusi empiris dan beberapa batang kepadatan ke dalam satu plot. Alih-alih plot batang untuk distribusi kali ini kami menggunakan plot gigi gergaji.

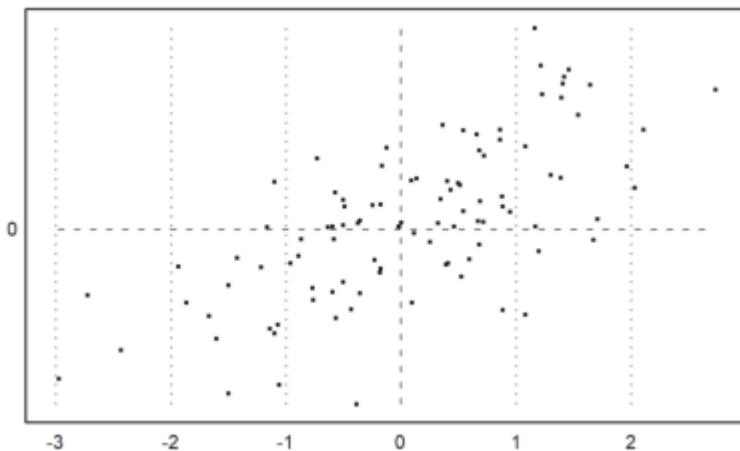
```
>figure(2,1); ...
figure(1); plot2d("empdist",-4,4;xs); ...
```

```
figure(2); plot2d(histo(x,v=-4:0.2:4,<bar)); ...  
figure(0):
```



Plot sebar mudah dilakukan di Euler dengan plot titik biasa. Grafik berikut menunjukkan bahwa X dan X+Y jelas berkorelasi positif.

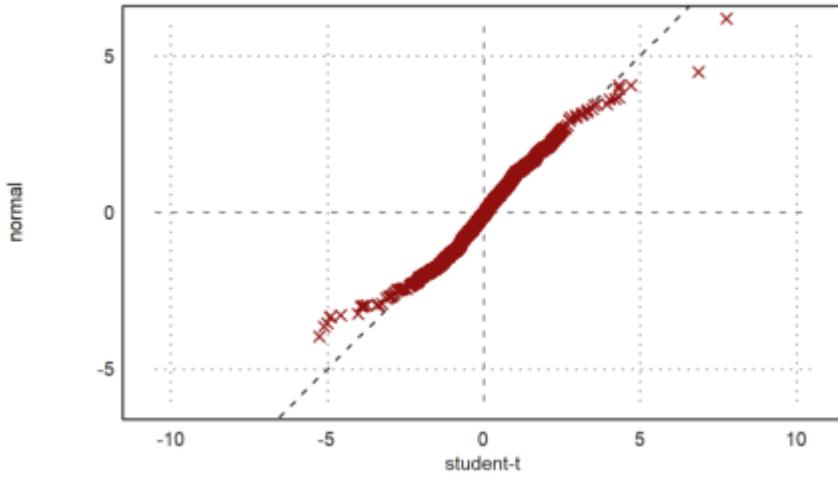
```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style=".."):
```



Seringkali kita ingin membandingkan dua sampel dengan distribusi yang berbeda. Hal ini dapat dilakukan dengan plot kuantil-kuantil.

Untuk pengujinya, 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):
```



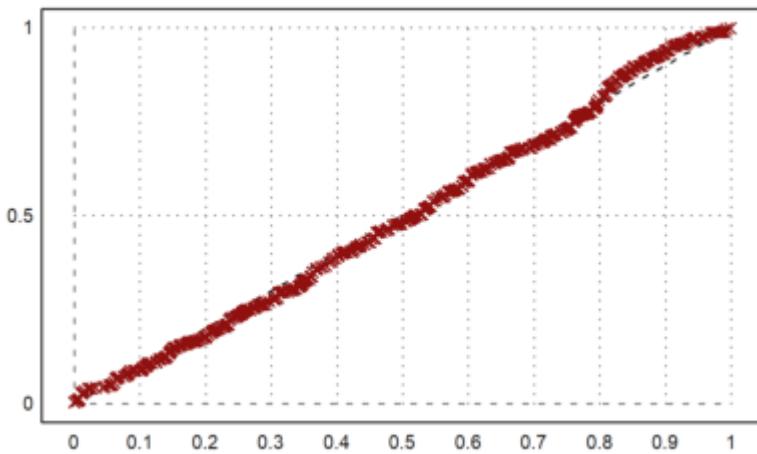
Plot tersebut dengan jelas menunjukkan bahwa nilai terdistribusi normal cenderung lebih kecil di ujung ekstrim.

Jika kita mempunyai dua distribusi yang ukurannya berbeda, kita dapat memperluas distribusi yang lebih kecil atau mengecilkan distribusi yang lebih besar. Fungsi berikut ini baik untuk keduanya. Dibutuhkan 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 permulaan kita menemukan 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

6	3
7	7
8	8
9	9
10	8

Kami ingin membandingkan kecocokan yang tidak berbobot dan berbobot. Pertama koefisien kecocokan linier.

```
>p=polyfit(x,y,1)
```

```
Need two real matrices with same number of rows for givensqr
fit:
{x,y,c}=givensqr(A,b);
Try "trace errors" to inspect local variables after errors.
polyfit:
    return fit(A,y')';
```

Regresi linear dapat ditulis dalam bentuk:

$$y = mx + b$$

dengan

$$m = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$$

$$b = \frac{\sum y - m(\sum x)}{n}$$

Kita hitung:

$$n = 10, x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], y = [2, 3, 1, 5, 6, 3, 7, 8, 9, 8]$$

$$\sum x = 55$$

$$\sum y = 52$$

$$\sum xy = 353$$

$$\sum x^2 = 385$$

Maka:

$$m = \frac{10(353) - (55)(52)}{10(385) - (55)^2} = \frac{3530 - 2860}{3850 - 3025} = 0.812121$$

$$b = \frac{52 - 0.812121(55)}{10} = 0.733333$$

Jadi, $b, m = 0.733333, 0.812121$

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 memasukkan semuanya ke dalam satu plot untuk 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):
```

```
Matrices must fit for plotarea!
plot2d:
  if auto then plotarea(xx,yy); endif;
Try "trace errors" to inspect local variables after errors.
statplot:
  plot2d(x,y,style=lststyle,xl=xl,yl=yl,color=color,vertical=vert ...
```

Contoh lain kita membaca survei siswa, usia mereka, usia orang tua mereka dan jumlah saudara kandung dari sebuah file.

Tabel ini berisi "m" dan "f" di kolom kedua. Kami menggunakan variabel tok2 untuk mengatur terjemahan yang tepat alih-alih membiarkan readtable() mengumpulkan terjemahannya.

```
>{MS,hd}:=readtable("table1.dat",tok2:=[ "m", "f" ]); ...
writetable(MS,labc=hd,tok2:=[ "m", "f" ]);
```

Person	Sex	Age	Mother	Father	Siblings
1	m	29	58	61	1
2	f	26	53	54	2
3	m	24	49	55	1
4	f	25	56	63	3
5	f	25	49	53	0
6	f	23	55	55	2
7	m	23	48	54	2
8	m	27	56	58	1
9	m	25	57	59	1
10	m	24	50	54	1
11	f	26	61	65	1
12	m	24	50	52	1
13	m	29	54	56	1
14	m	28	48	51	2
15	f	23	52	52	1
16	m	24	45	57	1
17	f	24	59	63	0
18	f	23	52	55	1
19	m	24	54	61	2
20	f	23	54	55	1

Bagaimana usia bergantung satu sama lain? Kesan pertama muncul dari plot sebar berpasangan.

```
>scatterplots(tablecol(MS,3:5),hd[3:5]):
```

Jelas terlihat 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 $s=17+0,74t$, dengan t adalah umur ibu dan s adalah umur ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tapi tidak terlalu banyak.

Sebaliknya, kami mencurigai fungsi seperti $s=a+t$. Maka a adalah mean dari $s-t$. Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

```
>da:=mean(cs[2]-cs[1])
```

3.65

Mari kita plot ini menjadi satu plot sebar.

```
>plot2d(cs[1],cs[2],>points); ...
plot2d("evalpoly(x,ps)",color=red,style=".",>add); ...
plot2d("x+da",color=blue,>add):
```

Berikut adalah plot kotak dari dua zaman tersebut. Ini hanya menunjukkan, bahwa usianya berbeda-beda.

```
>boxplot(cs, ["mothers", "fathers"]):
```

Menariknya, perbedaan median tidak sebesar perbedaan mean.

```
>median(cs[2])-median(cs[1])
```

1.5

Koefisien korelasi menunjukkan korelasi positif.

```
>Koefisien korelasi menunjukkan korelasi positif.correl(cs[1],cs[2])
```

```
Variable Koefisien not found!
Error in:
Koefisien korelasi menunjukkan korelasi positif.correl(cs[1],c ...
^
```

Korelasi pangkat merupakan ukuran keteraturan yang sama pada kedua vektor. Hal 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. Misalnya, kita mendefinisikan fungsi skewness.

$$sk(x) = \frac{\sqrt{n} \sum_i (x_i - m)^3}{(\sum_i (x_i - m)^2)^{3/2}}$$

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))
```

0.00180922922014

Berikut adalah fungsi lainnya, yang disebut koefisien skewness Pearson.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)
>skew1(data)
```

-0.0723383096094

Simulasi Monte Carlo

Kita simulasikan variabel acak berdistribusi normal 1000-5 sebanyak sejuta kali. Untuk ini, kita gunakan fungsi `normal(m,n)`, yang menghasilkan matriks nilai berdistribusi 0-1, atau `normal(n)` yang secara default bernilai `m=1`.

```
>n=1000000; x=normal(n)
```

```
[0.594669, 0.404839, 1.57377, -0.594406, 0.957243, 0.989854,
-0.750387, -1.59055, 1.11801, -0.945978, 1.01442, -0.725136,
1.16019, -0.637816, -1.00468, 0.451281, -2.60576, -0.620727,
0.247091, 0.633153, -0.120661, -0.732626, -1.02127, -0.0902834,
0.243185, -1.61108, 0.0818376, -0.632437, -0.425978, 0.0703536,
-0.209187, -0.225337, 0.445292, -0.754709, 1.53436, -0.979981,
-1.03102, 0.852066, 0.379605, -1.09705, 1.20582, -1.86178,
0.884487, -0.192057, -0.07492, 0.764815, -0.802051, 0.18518,
1.29316, -0.241986, 0.90877, 0.562809, -0.0182247, 0.213672,
0.523917, -0.271766, 0.62491, -1.1295, -0.726493, 0.638357,
-0.783011, -1.07234, 0.884189, -0.4191, -0.41695, -1.15576,
0.274547, -0.0633383, -0.164788, 0.883511, -1.97886, -2.41251,
1.44818, -0.828575, 1.90299, 0.644959, -0.362327, -0.111006,
-1.16964, 0.665621, 0.486508, -0.50645, 0.51728, -2.13836,
0.706669, -0.918796, 1.42452, 0.435429, -0.137239, -0.0160858,
-1.04948, -1.26106, -0.988632, 0.0747217, 0.124635, -2.45891,
0.00636367, 1.36691, -0.477445, -0.338019, -1.12295, -1.57755,
-0.289792, -0.313483, -0.703814, -0.190004, 0.0156531, -0.743473,
-0.399538, -0.939934, -0.765945, -0.473433, -1.30978, -1.6371,
0.879558, 0.722848, 2.26233, 0.6636, 0.454741, 0.620934,
... ]
```

```
>n=1000000; x=normal(n)*5+1000
```

```
[997.217, 1002.59, 1009.99, 998.583, 1005.33, 1000.05, 1004.29,
1008.29, 992.032, 1013.82, 996.485, 1002.4, 999.836, 1005.51,
999.52, 997.989, 1008.56, 1011.49, 999.683, 999.686, 988.919,
997.435, 1008.75, 1000.05, 1005.1, 993.698, 999.77, 998.385,
1003.1, 997.314, 999.377, 1000.46, 998.008, 995.515, 997.618,
996.455, 1004.83, 1002.92, 1002.93, 1000.02, 1001.99, 986.759,
1003.92, 993.189, 996.029, 989.77, 1006.03, 994.944, 998.395,
1004.7, 997.238, 1011.16, 1001.51, 992.181, 1000.37, 994.389,
999.321, 998.169, 1005.56, 1000.49, 993.994, 1000.62, 997.579,
1005.94, 1008.35, 999.007, 1001.37, 998.159, 1000.34, 995.868,
996.674, 997.832, 996.366, 1004.65, 1000.83, 1004.8, 1006.08,
993.986, 1006.84, 999.963, 996.663, 1000.55, 1002.6, 995.723,
996.858, 1006.43, 1000.97, 1000.79, 1002.96, 1009.19, 998.141,
1000.08, 996.891, 991.271, 998.378, 997.582, 999.068, 999.808,
1002.07, 990.138, 1000.07, 991.562, 991.012, 996.126, 994.941,
999.757, 999.237, 1007.14, 1003.09, 998.811, 1002.92, 1002.77,
997.681, 994.765, 997.174, 1002.42, 995.981, 1005.29, 1002.61,
```

```
1000.33, 1002.12, 998.899, 994.029, 991.05, 1002.69, 1004.1,  
1003.5, 999.189, 998.607, 987.859, 1004.61, 993.364, 990.851,  
993.328, 1008.34, 1004.39, 1007.6, 996.818, 999.149, 1007.52,  
... ]
```

terdapat juga fungsi `randnormal(n,m,mean,dev)`, yang dapat kita gunakan. Fungsi ini mematuhi skema penamaan "rand..." untuk generator acak.

```
>n=1000000; x=randnormal(1,n,1000,5)
```

```
[993.909, 995.449, 1002.06, 1003.15, 1007.19, 999.615, 998.355,  
999.025, 1006.23, 1002, 1000.7, 1000.2, 1005.38, 991.512,  
991.345, 1001.71, 991.031, 990.978, 998.105, 993.577, 999.242,  
989.155, 1007.17, 1001.49, 1001.54, 1001.32, 1014.04, 1005.8,  
1008.42, 1003.44, 994.948, 1010.62, 1006.41, 1005.93, 998.34,  
997.584, 997.494, 1003.14, 997.31, 997.461, 989.823, 999.147,  
1005.93, 996.811, 991.033, 1009.29, 996.466, 998.813, 1002.88,  
1005.13, 1008.62, 1005.91, 992.439, 1000.37, 997.342, 1005.15,  
1002.47, 991.082, 992.811, 995.208, 992.557, 1003.33, 1004.19,  
1002.22, 994.346, 993.58, 993.792, 998.402, 1004.58, 1008.68,  
993.079, 995.845, 1002.32, 1003.82, 1002, 999.093, 1011.66,  
1003.97, 1001.16, 1000.43, 1013.04, 1002.65, 1000.9, 997.038,  
999.367, 1003.75, 996.096, 996.257, 999.541, 1003.5, 997.854,  
995.397, 994.204, 985.007, 1005.95, 1000.34, 1006.44, 994.612,  
1003.64, 1005.92, 1000.94, 1001.69, 995.274, 992.588, 998.776,  
990.343, 1004.71, 998.007, 1000.23, 998.707, 1002.59, 1003.65,  
1000.12, 1009.27, 992.548, 998.5, 1000.64, 1000.44, 995.67,  
1003.11, 997.103, 1005.87, 1005.09, 996.395, 1002.52, 1000.84,  
1006.47, 1001.31, 1007.94, 998.893, 1003.86, 1005.1, 993.376,  
1005.83, 1000.41, 1000.2, 992.949, 998.768, 1008.35, 1005.16,  
... ]
```

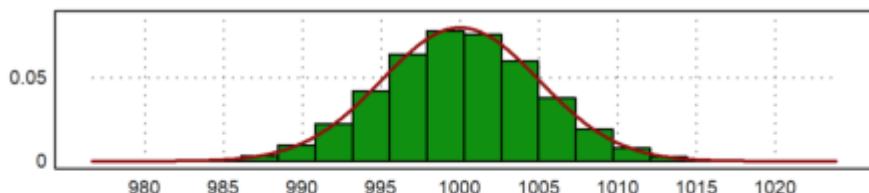
10 nilai pertama x adalah

```
>x[1:10]
```

```
[993.909, 995.449, 1002.06, 1003.15, 1007.19, 999.615, 998.355,  
999.025, 1006.23, 1002]
```

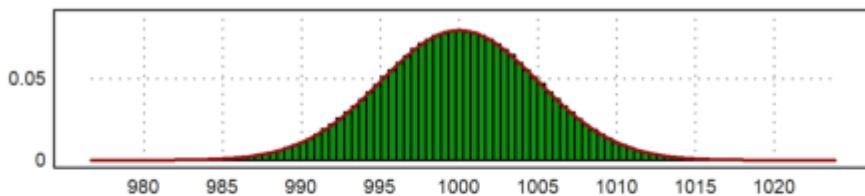
Distribusi dapat kita plot dengan flag `>distribution` dari `plot2d`.

```
>plot2d(x,>distribution); ...  
plot2d("qnormal(x,1000,5)",color=red,thickness=2,>add):
```



kita juga dapat mengatur jumlah interval untuk distribusi menjadi 100. Kemudian kita akan melihat seberapa dekat kecocokan distribusi yang diamati dan distribusi yang sebenarnya. Bagaimanapun, kita telah menghasilkan satu juta kejadian.

```
>plot2d(x,distribution=100); ...
plot2d("qnormal(x,1000,5)",color=red,thickness=2,>add) :
```



kita dapat menghitung nilai rata-rata simulasi dan deviasinya harus sangat dekat dengan nilai yang diharapkan.

```
>mean(x), dev(x)
```

```
1000.00174908
4.99977952257
```

rumus nilai rata rata

$$mean = \frac{\sum x_i}{n}$$

```
>xm=sum(x)/n
```

```
1000.00174908
```

Rumus simpangan percobaannya (deviasi)

$$deviasi = \sqrt{\frac{\sum(x - xm)^2}{n - 1}}$$

```
>sqrt(sum((x-xm)^2/(n-1)))
```

```
4.99977952257
```

Perhatikan bahwa $x - xm$ adalah vektor nilai yang dikoreksi, di mana xm dikurangi dari semua elemen vektor x .

Berikut adalah 10 nilai pertama $x - xm$.

```
>short (x-xm)[1:10]
```

```
[ -6.0931, -4.5526, 2.0616, 3.1487, 7.1846, -0.38714, -1.6464,
-0.97655, 6.2287, 2.0015]
```

Dengan menggunakan bahasa matriks, kita dapat dengan mudah menjawab pertanyaan lainnya. Misalnya, kita ingin menghitung proporsi x yang melebihi 1015. Ekspresi $x \geq 1015$ menghasilkan vektor 1 dan 0. Menjumlahkan vektor ini menghasilkan jumlah kali $x[i] \geq 1015$ terjadi.

```
>sum(x>=1015)/n
```

```
0.001318
```

Probabilitas yang diharapkan dari hal ini dapat dihitung dengan fungsi `normaldis(x)`. sehingga,

$$\text{normaldis}(c, m, s) = P(X \leq c)$$

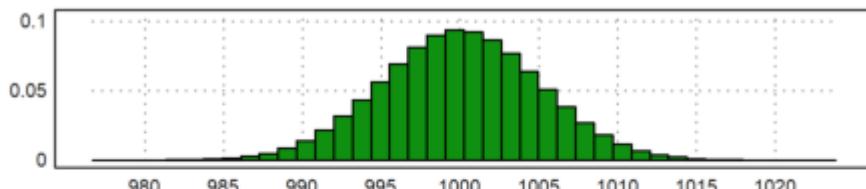
dimana X terdistribusi secara normal m-s.

```
>1-normaldis(1015,1000,5)
```

```
0.00134989803163
```

cara kerja `>distribution flag` dari `plot2d` adalah menggunakan fungsi `histo(x)`, yang menghasilkan histogram frekuensi nilai dalam x . Fungsi ini mengembalikan batas interval dan jumlah dalam interval ini. Kami menormalkan jumlah untuk mendapatkan frekuensi.

```
>{t,s}=histo(x,40); plot2d(t,s/n,>bar):
```



Fungsi `histo()` juga dapat menghitung frekuensi dalam interval yang diberikan.

```
>{t,s}=histo(x,v=[ 950,980,990,1010,1020,1050]); t, s,
```

```
[ 950, 980, 990, 1010, 1020, 1050]
[35, 22694, 954435, 22806, 30]
```

hasil tersebut merupakan semua nilai acak yang berada antara 950 dan 1050.

menghitung total jumlah nilai dalam s , yang sama dengan total jumlah elemen dalam x

```
>sum(s)
```

```
1000000
```

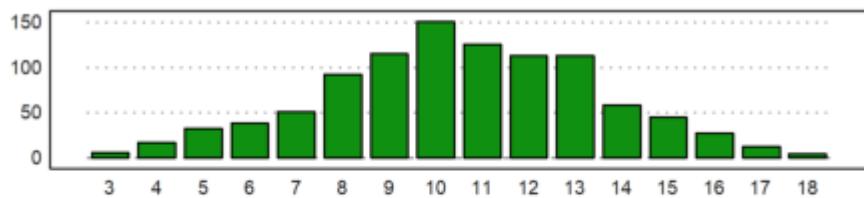
kita akan mensimulasikan 1000 kali lemparan 3 dadu, dan menanyakan pembagian jumlahnya.

```
>ds:=sum(intrandom(1000,3,6)); fs=getmultiplicities(3:18,ds)
```

```
[6, 17, 32, 38, 51, 92, 115, 151, 126, 113, 113, 58, 45, 27, 12, 4]
```

kita akan plot hasil tersebut

```
>columnsplot(fs,lab=3:18):
```



kita akan menggunakan rekursi tingkat lanjut.

Fungsi berikut menghitung banyaknya cara bilangan k dapat direpresentasikan sebagai jumlah dari n bilangan dalam rentang 1 sampai m.

```
>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 hasil pelemparan dadu sebanyak lima kali.

```
>countways(5:25,5,5)
```

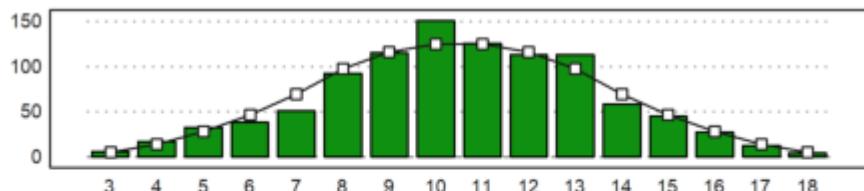
```
[1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320, 255, 185, 121, 70, 35, 15, 5, 1]
```

```
>cw=countways(3:18,3,6)
```

```
[1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3, 1]
```

Kita akan menambahkan nilai yang diharapkan ke plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```



Untuk simulasi lain, deviasi nilai rata-rata n 0-1-variabel acak terdistribusi normal adalah $1/\sqrt{n}$.

```
>longformat; 1/sqrt(10)
```

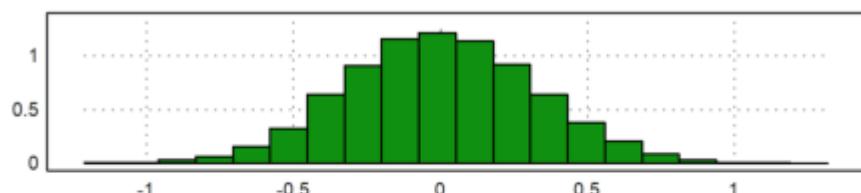
0.316227766017

Mari kita periksa ini dengan simulasi. Kami menghasilkan 10.000 kali 10 vektor acak.

```
>M=normal(10000,10); dev(mean(M)')
```

0.319021256442

```
>plot2d(mean(M)',>distribution):
```

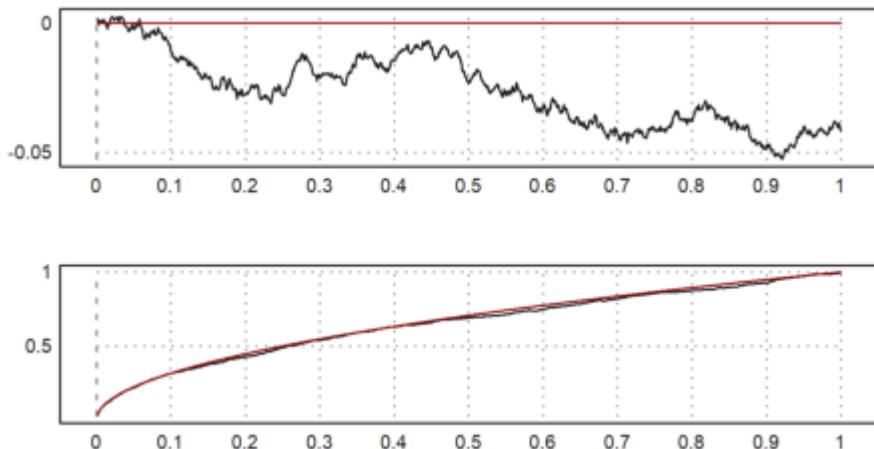


Median dari 10 bilangan acak berdistribusi normal 0-1 mempunyai deviasi yang lebih besar.

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 langkah ke-n dari proses ini bersama dengan nilai yang diharapkan berwarna 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 chi-kuadrat

uji chi-kuadrat adalah alat penting dalam statistik. Di Euler, banyak tes yang diterapkan. Semua pengujian ini mengembalikan kesalahan yang kita terima jika kita menolak hipotesis nol.

Misalnya, kami menguji lemparan dadu untuk distribusi yang seragam. Pada 600 kali lemparan, kami mendapatkan nilai berikut, yang kami masukkan ke dalam uji chi-kuadrat.

```
>chitest([90,103,114,101,103,89],dup(100,6)')
```

0.498830517952

Ini adalah nilai p-value dari uji chi-kuadrat

Uji chi-kuadrat juga memiliki mode yang menggunakan simulasi Monte Carlo untuk menguji statistiknya, menggunakan Parameter `>p` menafsirkan vektor `y` sebagai vektor probabilitas.

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

0.488

Ini adalah p-value dari uji chi-kuadrat menggunakan pendekatan Monte Carlo. Dengan simulasi Monte Carlo, kita memperoleh p-value yang mirip dengan uji chi-kuadrat standar (0,4988 di uji pertama)

Selanjutnya kita menghasilkan 1000 lemparan dadu menggunakan generator angka acak, dan melakukan tes yang sama.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

0.45357780172

Mari kita uji nilai rata-rata 100 dengan uji-t.

```
>s=200+normal([1,100])*10; ...
ttest(mean(s),dev(s),100,200)
```

0.2654993686

Fungsi ttest() memerlukan nilai mean, deviasi, jumlah data, dan nilai mean yang akan diuji.

Sekarang mari kita periksa dua pengukuran untuk mean yang sama. Kami menolak hipotesis bahwa keduanya mempunyai mean yang sama, jika hasilnya $<0,05$.

```
>tcomparedata(normal(1,10),normal(1,10))
```

0.306435607333

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)
```

0.000742363759602

Menambah nilai 2 ke salah satu distribusi menyebabkan p-value menjadi sangat kecil.

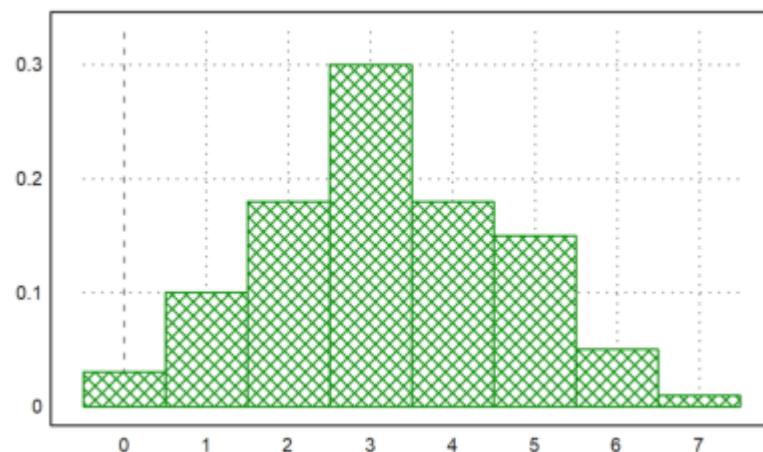
Pada contoh berikutnya, kita membuat 20 lemparan dadu acak sebanyak 100 kali dan menghitung yang ada di dalamnya. Rata-rata harus ada $20/6=3,3$.

```
>R=random(100,20); R=sum(R*6<=1)'; mean(R)
```

3.2

Sekarang kita bandingkan jumlah satuan dengan distribusi binomial. Pertama kita plot distribusinya.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\\"/") :
```



kita akan Menghitung frekuensi kemunculan setiap jumlah angka "1" dalam 20 lemparan dadu acak yang telah dilakukan 100 kali

```
>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;
```

Kita harus mengumpulkan beberapa angka untuk mendapatkan kategori yang cukup besar.

```
>t1=sum(t[1:2])|t[3:7]|sum(t[8:21]); ...
b1=sum(b[1:2])|b[3:7]|sum(b[8:21]);
```

Uji chi-square menolak hipotesis bahwa distribusi kita merupakan distribusi binomial, jika hasilnya <0,05.

```
>chitest(t1,b1)
```

0.585701058715

Contoh berikut berisi hasil dua kelompok orang (misalnya laki-laki dan perempuan) yang memilih satu dari enam partai.

```
>A=[23,37,43,52,64,74;27,39,41,49,63,76]; ...
writetable(A,wc=6,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	23	37	43	52	64	74
f	27	39	41	49	63	76

Kita akan menguji independensi suara dari jenis kelamin.

```
>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 dikoreksi. Karena sangat mendekati 0, kami menyimpulkan bahwa pemungutan suara tidak bergantung pada jenis kelamin.

```
>contingency(A)
```

0.0427225484717

uji F

Selanjutnya kita menggunakan analisis varians (uji F) untuk menguji tiga sampel data yang berdistribusi normal untuk nilai mean yang sama. Metode tersebut disebut ANOVA (analisis varians). Di Euler, fungsi varanalisis() digunakan.

```
>x1=[109,111,98,119,91,118,109,99,115,109,94]; mean(x1),
```

106.54545454545

```
>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
```

Dengan p-value sebesar 0.0138 (1,38%), kita bisa menolak hipotesis bahwa ketiga sampel memiliki mean yang sama pada tingkat signifikansi 5% (0.05) dan bahkan pada tingkat signifikansi 1% (0.01). Artinya, terdapat perbedaan yang signifikan antara mean dari setidaknya satu sampel.

Ada juga uji median, yang menolak sampel data dengan distribusi rata-rata yang berbeda, menguji median dari sampel yang disatukan.

```
>a=[56,66,68,49,61,53,45,58,54]
```

```
[56, 66, 68, 49, 61, 53, 45, 58, 54]
```

```
>b=[72,81,51,73,69,78,59,67,65,71,68,71]
```

```
[72, 81, 51, 73, 69, 78, 59, 67, 65, 71, 68, 71]
```

```
>mediantest(a,b)
```

```
0.0241724220052
```

Tes kesetaraan lainnya adalah tes peringkat. Ini jauh lebih tajam daripada tes median.

```
>ranktest(a,b)
```

```
0.00199969612469
```

Pada contoh berikut, kedua distribusi mempunyai mean yang sama.

```
>ranktest(random(1,100),random(1,50)*3-1)
```

```
0.0695597439892
```

ini menunjukkan bahwa perbedaan tidak cukup signifikan pada tingkat signifikansi 5%, sehingga hipotesis bahwa kedua distribusi memiliki median yang sama tidak dapat ditolak.

Sekarang mari kita coba mensimulasikan dua perlakuan 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];
```

Tes signum memutuskan, apakah a lebih baik dari b.

```
>signtest(a,b)
```

0.0546875

Ini kesalahan yang terlalu besar untuk menolak hipotesis. Kita tidak dapat menolak bahwa a sama baiknya dengan b, Karena $p > 0.05$.

Uji Wilcoxon lebih tajam dibandingkan uji ini, namun mengandalkan nilai kuantitatif perbedaannya.

```
>wilcoxon(a,b)
```

0.0296680599405

Mari kita coba dua tes lagi menggunakan rangkaian yang dihasilkan.

```
>wilcoxon(normal(1,20),normal(1,20)-1)
```

0.033677278493

ini menunjukkan bahwa ada perbedaan signifikan antara kedua sampel pada tingkat signifikansi 5%.

```
>wilcoxon(normal(1,20),normal(1,20))
```

0.843268467533

hasil ini jauh di atas 0.05, sehingga kita tidak bisa menolak hipotesis bahwa kedua sampel berasal dari distribusi yang sama.

Angka Acak

Berikut ini adalah pengujian pembangkit bilangan acak. Euler menggunakan generator yang sangat bagus, jadi kita tidak perlu mengharapkan adanya masalah.

Pertama kita menghasilkan sepuluh juta angka acak di [0,1].

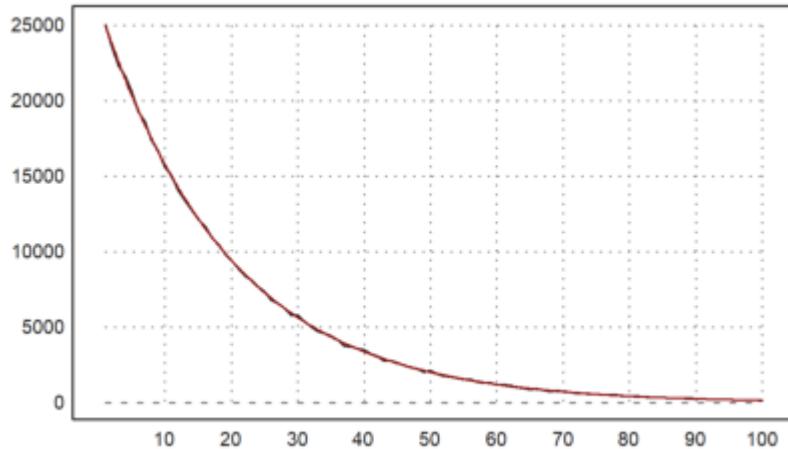
```
>n:=10000000; r:=random(1,n);
```

Selanjutnya kita hitung jarak antara dua angka yang kurang dari 0,05.

```
>a:=0.05; d:=differences(nonzeros(r<a));
```

Terakhir, kami memplot berapa kali, setiap jarak terjadi, dan membandingkannya dengan nilai yang diharapkan.

```
>m=getmultiplicities(1:100,d); plot2d(m); ...
plot2d("n*(1-a)^(x-1)*a^2",color=red,>add):
```



Hapus datanya.

```
>remvalue n;
```

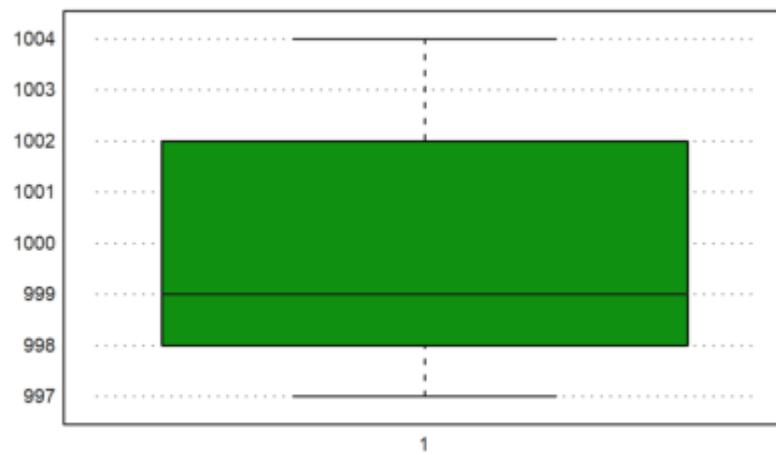
Kami ingin menghitung nilai rata-rata dan simpangan baku yang diukur.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...
mean(M), dev(M),
```

```
999.9
2.72641400622
```

Kita dapat membuat diagram kotak dan kumis untuk data tersebut. Dalam kasus kita, tidak ada outlier.

```
>boxplot(M);
```



Kami menghitung probabilitas bahwa suatu nilai lebih besar dari 1005, dengan asumsi nilai terukur dan distribusi normal.

Semua fungsi untuk distribusi dalam Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

Kami mencetak hasilnya dalam % dengan akurasi 2 digit menggunakan fungsi cetak.

```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
```

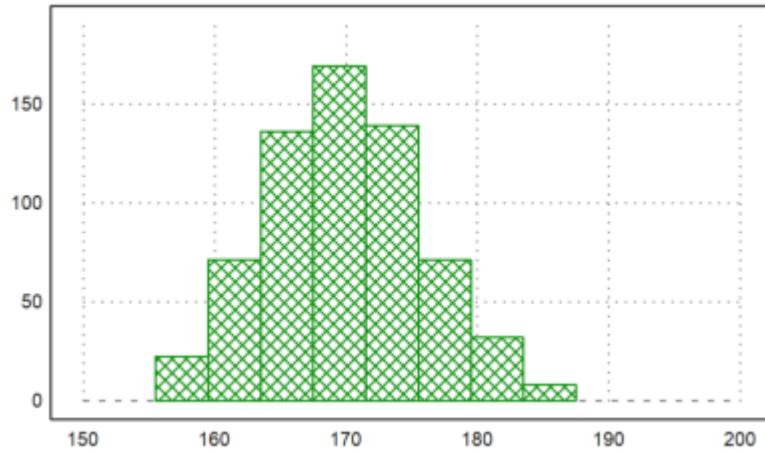
3.07 %

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 adalah plot distribusinya.

```
>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 metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal rentang, akhir rentang, jumlah orang dalam rentang.

Tabel dapat dicetak dengan tajuk. Kita menggunakan vektor string untuk mengatur tajuk.

```
>T:=r[1:8]' | r[2:9]' | v'; writetable(T,labc=["from","to","count"]);
```

from	to	count
155.5	159.5	22
159.5	163.5	71
163.5	167.5	136
167.5	171.5	169
171.5	175.5	139
175.5	179.5	71
179.5	183.5	32
183.5	187.5	8

Jika kita memerlukan nilai rata-rata dan statistik ukuran lainnya, kita perlu menghitung titik tengah rentang. Kita dapat menggunakan dua kolom pertama tabel kita untuk ini.

```
>(T[,1]+T[,2])/2
```

```
157.5  
161.5  
165.5  
169.5  
173.5  
177.5  
181.5  
185.5
```

Namun lebih mudah untuk melipat rentang dengan vektor [1/2,1/2].

```
>l=fold(r,[0.5,0.5])
```

```
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
```

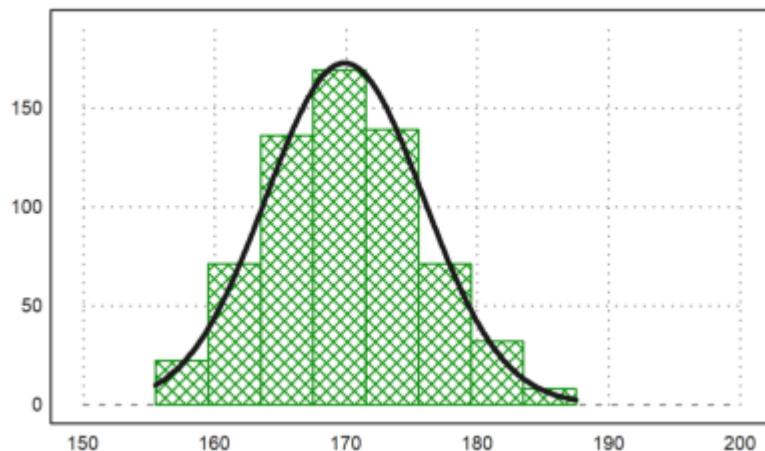
Sekarang kita dapat menghitung rata-rata dan deviasi sampel dengan frekuensi yang diberikan.

```
>{m,d}=meandev(l,v); m, d,
```

```
169.901234568  
5.98912964449
```

Mari kita tambahkan distribusi normal nilai-nilai tersebut ke plot.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...  
xmin=min(r),xmax=max(r),thickness=3,add=1):
```



Pengantar untuk Pengguna Proyek R

Jelasnya, 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.

Notebook ini cocok untuk Anda yang sudah familiar dengan R, namun perlu mengetahui perbedaan sintaksis EMT dan R. Kami mencoba memberikan gambaran umum tentang hal-hal yang sudah jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami mencari cara untuk bertukar data antara kedua sistem.

Sintaks Dasar

Hal pertama yang Anda pelajari di R adalah membuat vektor. Dalam EMT, perbedaan utamanya adalah operator : dapat mengambil ukuran langkah. Selain itu, ia mempunyai 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]
```

```
>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() di R. Kita dapat menulis fungsi ini di 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]
```

```
>function seq(a,b,c) := a:b:c; ...
seq(0,-0.5,-5)
```

```
[0, -0.5, -1, -1.5, -2, -2.5, -3, -3.5, -4, -4.5, -5]
```

```
>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]
```

Fungsi rep() dari R tidak ada di EMT. Untuk masukan vektor dapat dituliskan sebagai berikut.

Perhatikan bahwa "=" atau ":=" digunakan untuk tugas. Operator "->" digunakan untuk satuan dalam EMT.

```
>125km -> " miles"
```

```
77.6713990297 miles
```

Operator "<- " untuk penugasan memang bukan ide yang baik untuk R.

tetapi di EMT operator "<-" itu bukan penugasan melainkan perbandingan
Berikut ini akan membandingkan a dan -4 di EMT.

```
>a:=2; a<-4
```

```
0
```

EMT dan R memiliki vektor bertipe boolean. Namun dalam EMT, angka 0 dan 1 digunakan untuk mewakili salah dan benar. Di R, nilai benar dan salah tetap bisa digunakan dalam aritmatika biasa seperti di EMT.

```
>x<5, %*x
```

```
[0, 0, 1, 0, 0]
[0, 0, 3.1, 0, 0]
```

EMT memunculkan kesalahan atau menghasilkan NAN tergantung pada tanda "kesalahan".

```
>errors off; 0/0, isNaN(sqrt(-1)), errors on;
```

```
NAN
```

```
1
```

Stringnya sama di R dan EMT. Keduanya berada di lokal saat ini, bukan di Unicode.

Di R ada paket untuk Unicode. Di EMT, string dapat berupa string Unicode. String unicode dapat diterjemahkan ke pengkodean lokal dan sebaliknya. Selain itu, u"..." dapat berisi entitas HTML.

```
>u"\u00c2; Ren\u00e9 Grothmann"
```

© René Grothmann

karakter khusus (hak cipta © dan karakter aksen é),

```
>chartoutf([480])
```

Ã

Berikut ini mungkin tidak ditampilkan dengan benar pada sistem sebagai A dengan titik dan garis di atasnya. Itu tergantung pada font yang Anda gunakan.

Penggabungan string dilakukan dengan "+" atau "|". Penggabungan ini akan menghasilkan string tunggal, dan angka yang digabungkan akan dikonversi otomatis ke format string. Ini dapat mencakup angka, yang akan dicetak dalam format saat ini.

```
>"pi = "+pi
```

pi = 3.14159265359

Pengindeksan

Seringkali, ini akan berfungsi seperti di R.

Namun EMT akan menafsirkan indeks negatif dari belakang vektor, sementara R menafsirkan 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
```

```
>x, x[1:5], x[-3]
```

```
[10.4, 5.6, 3.1, 6.4, 21.7]
[10.4, 5.6, 3.1, 6.4, 21.7]
3.1
```

Untuk meniru perilaku R di EMT, kita dapat menggunakan fungsi drop(x,n)

```
>drop(x,2)
```

```
[10.4, 3.1, 6.4, 21.7]
```

Vektor logika tidak diperlakukan berbeda sebagai indeks di EMT, berbeda dengan R. Anda perlu mengekstrak elemen 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 di R, vektor indeks dapat berisi pengulangan.

```
>x[[1,2,2,1]]
```

```
[10.4, 5.6, 5.6, 10.4]
```

Tipe Data

EMT memiliki lebih banyak tipe data tetap daripada R. Jelasnya, di R terdapat vektor yang berkembang. Anda dapat mengatur vektor numerik kosong v dan memberikan nilai ke elemen v[17]. Hal ini tidak mungkin dilakukan di EMT.

Berikut ini agak tidak efisien.

```
>v=[]; for i=1 to 10000; v=v|i; end;
```

kenapa cara ini kurang efisien? karna setiap elemen baru di tambahkan EMT harus menyalin seluruh isi v kembali ke variabel v.

Semakin efisien vektor telah ditentukan sebelumnya.

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Untuk mengubah tipe data di EMT, Anda dapat menggunakan fungsi seperti kompleks().

```
>complex(1:4)
```

```
[ 1+0i , 2+0i , 3+0i , 4+0i ]
```

Konversi ke string hanya dimungkinkan untuk tipe data dasar. Format saat ini digunakan untuk peng gabungan string sederhana. Tapi ada fungsi seperti print() atau frac().

Untuk vektor, Anda dapat dengan mudah menulis fungsi Anda sendiri.

```
>function tostr (v) ...
s="[";  
loop 1 to length(v);
  s=s+print(v[#],2,0);
  if #<length(v) then s=s+","; endif;
end;
return s+"]";
endfunction
```

- Variabel s diinisialisasi sebagai string "[" untuk menyimpan hasil akhir. Awalnya, tanda kurung buka [ditambahkan ke variabel s sebagai pembuka.
- loop 1 to length(v); menjalankan perulangan dari elemen pertama hingga elemen terakhir dalam v. Fungsi length(v) mengembalikan panjang atau jumlah elemen dalam vektor v.
- print(v[#], 2, 0); adalah fungsi format yang mengonversi elemen vektor v pada posisi saat ini (v[#]) menjadi string. parameter 2 menunjukkan bahwa dua digit setelah titik desimal akan ditampilkan, sementara 0 memastikan bahwa angka ditampilkan tanpa tambahan simbol atau format lainnya.
- Bagian if #<length(v) memeriksa apakah elemen saat ini bukan elemen terakhir. Jika benar, maka koma , akan ditambahkan ke variabel s untuk memisahkan elemen.

- Setelah loop selesai, tanda kurung tutup] ditambahkan ke string s, dan string ini kemudian dikembalikan sebagai output.

```
>tostr(linspace(0,1,10));
```

Untuk komunikasi dengan Maxima, terdapat fungsi convertmxm(), yang juga dapat digunakan untuk memformat vektor untuk keluaran.

```
>convertmxm(1:10);
```

Untuk Latex perintah tex dapat digunakan untuk mendapatkan perintah Latex.

```
>tex(&[1,2,3]);
```

[1, 2, 3]

Faktor dan Tabel

Dalam pengantar R ada contoh yang disebut faktor.

Berikut ini adalah daftar wilayah 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"];
```

Perintah diatas digunakan untuk mendefinisikan sebuah array (array sendiri adalah sekumpulan variabel yang memiliki tipe data yang sama) karena pada data tersebut ada beberapa nama negara bagian yang terulang. Array ini berisi singkatan untuk negara bagian dan teritori di Australia.

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 mari kita coba mencari nilai mean dan median dari data pendapatan tersebut menggunakan perintah mean(incomes) dan median(incomes)

```
>mean(incomes)
```

54.7333333333

```
>median(incomes)
```

57

Sekarang, kami ingin menghitung rata-rata pendapatan di suatu wilayah. Menjadi program statistik, R memiliki faktor() dan tapply() untuk ini.

EMT dapat melakukan hal ini dengan menemukan indeks wilayah dalam daftar wilayah 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, 1]
```

Pada titik itu, kita dapat menulis fungsi perulangan kita sendiri untuk melakukan sesuatu hanya untuk satu faktor.

Atau kita bisa meniru fungsi tapply() dengan cara berikut.

```
>function map_tappl (i; f$:call, cat, x) ...
u=sort(unique(cat));
f=indexof(u,cat);
return f$(x[nonzeros(f==indexof(u,i))]);
endfunction
```

i: Parameter pertama biasanya adalah nilai yang digunakan untuk pencocokan atau pemetaan.
f\$:call: Parameter kedua, yang kemungkinan besar adalah sebuah fungsi yang dipanggil dalam kode tersebut. f\$ di sini merujuk pada fungsi yang diterima sebagai input.
cat: Parameter ketiga adalah array atau vektor yang berisi kategori yang akan diproses.
x: Parameter keempat adalah array atau vektor yang akan diproses atau diubah berdasarkan pemetaan kategori yang dilakukan.

Ini agak tidak efisien, karena menghitung wilayah unik untuk setiap i, tetapi berhasil.

```
>tappl(auterr,"mean",austates,incomes)
```

```
[44.5, 57.3333333333, 55.5, 53.6, 55, 60.5, 56, 52.25]
```

Perhatikan bahwa ini berfungsi untuk setiap vektor wilayah.

```
>tappl(["act","nsw"],"mean",austates,incomes)
```

```
[44.5, 57.3333333333]
```

Sekarang, paket statistik EMT mendefinisikan tabel seperti di R. Fungsi readtable() dan writetable() dapat digunakan untuk input dan output.

Sehingga kita bisa mencetak rata-rata pendapatan negara di daerah secara bersahabat.

```
>writetable(tappl(auterr,"mean",austates,incomes),labc=auterr,wc=7)
```

act	nsw	nt	qld	sa	tas	vic	wa
44.5	57.33	55.5	53.6	55	60.5	56	52.25

Fungsi writetable digunakan untuk menampilkan data dalam bentuk tabel yang terstruktur dengan label kolom dan lebar kolom yang dapat disesuaikan.

Dengan labc=auterr, berarti menetapkan label kolom untuk tabel tersebut berdasarkan kategori yang ada di auterr(yang sudah diurutkan sesuai abjad).

wc(width of columns)=7 berarti setiap kolom dalam tabel akan memiliki lebar minimal 7 karakter. sebagai contoh 44.5 itu memiliki 4 karakter (termasuk titik desimal).

karena data dalam kolom lebih pendek dari 7 karakter, kolom tersebut diberi ruang ekstra untuk tampilan yang rapi.

Kita juga bisa mencoba meniru perilaku R sepenuhnya.

Faktor-faktor tersebut harus disimpan dengan jelas dalam kumpulan beserta jenis dan kategorinya (negara bagian dan teritori dalam contoh kita). Untuk EMT, kami 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);
```

Perintah statef = makef(austates); digunakan untuk mengolah data yang ada dalam variabel austates, dan mengidentifikasi elemen unik yang ada dalam data tersebut.

Sekarang elemen ketiga dari koleksi akan berisi indeks.

```
>statef[3]
```

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

statef[3] adalah elemen ketiga dari koleksi yang dikembalikan oleh fungsi makef, yaitu indeks posisi dari elemen-elemen dalam austates yang sudah dipetakan ke urutan dalam u (data unik yang terurut). statef[3] akan mengembalikan indeks posisi dari setiap elemen dalam austates berdasarkan urutan yang ada di u.

Sekarang kita bisa meniru tapply() dengan cara berikut. Ini akan mengembalikan 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 pengecekan tipe di sini. Satu-satunya tindakan pencegahan menyangkut kategori (faktor) yang tidak memiliki data. Tetapi kita harus memeriksa panjang t yang benar dan kebenaran pengumpulan tf.

Tabel ini dapat dicetak sebagai tabel dengan writetable().

```
>writetable(tapply(incomes,statef,"mean"),wc=7)
```

act	nsw	nt	qld	sa	tas	vic	wa
44.5	57.33	55.5	53.6	55	60.5	56	52.25

Array

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Namun, akan mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau perpustakaan C untuk ini.

R memiliki lebih dari dua dimensi. Di R array adalah vektor dengan bidang dimensi.

Dalam EMT, vektor adalah matriks dengan satu baris. Itu dapat dibuat menjadi 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

Fungsi shortformat digunakan untuk mengatur format tampilan angka agar lebih ringkas dan mudah dibaca.

Perintah diatas digunakan untuk membuat matrik X dari angka 1 sampai 20 dengan ketentuan matriks dengan 4 baris dan 5 kolom.

Ekstraksi baris dan kolom, atau sub-matriks, mirip dengan R.

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

2	3
7	8
12	13
17	18

Perintah diatas digunakan untuk menampilkan matriks X kolom kedua sampai ketiga.

```
>X[,3:5]
```

3	4	5
8	9	10
13	14	15
18	19	20

Namun, di R dimungkinkan untuk menyetel daftar indeks vektor tertentu ke suatu nilai. Hal yang sama mungkin terjadi di EMT hanya dengan satu putaran.

```
>function setmatrixvalue (M, i, j, v) ...
loop 1 to max(length(i),length(j),length(v))
  M[i#{},j#{}] = v#;
end;
endfunction
```

Perintah setmatrixvalue(M, i, j, v) adalah fungsi yang digunakan untuk mengubah nilai elemen-elemen dalam matriks berdasarkan indeks tertentu.

M: Matriks yang akan dimodifikasi.

i: Indeks baris atau posisi baris dalam matriks M yang ingin diubah.

j: Indeks kolom atau posisi kolom dalam matriks M yang ingin diubah.

v: Nilai yang akan dimasukkan ke dalam elemen-elemen matriks M pada posisi yang ditentukan oleh indeks i dan j.

Kami mendemonstrasikan ini untuk menunjukkan bahwa matriks dilewatkan dengan referensi di EMT. Jika Anda tidak ingin mengubah matriks M asli, Anda perlu menyalinnya ke 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

Perkalian luar dalam EMT hanya dapat dilakukan antar vektor. Ini otomatis karena bahasa matriks. Satu vektor harus berupa vektor kolom dan vektor lainnya harus berupa vektor baris.

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

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

1:5: Ini adalah vektor baris yang berisi angka-angka dari 1 hingga 5

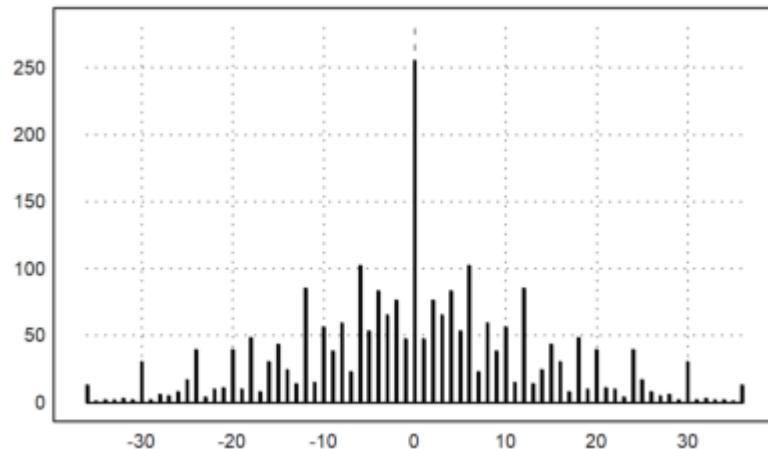
(1:5)': Tanda ' di sini menunjukkan transposisi dari vektor baris 1:5. Dengan kata lain, ini mengubah vektor baris menjadi vektor kolom.

Dalam PDF pendahuluan untuk R terdapat contoh yang menghitung distribusi ab-cd untuk a,b,c,d yang dipilih dari 0 hingga n secara acak. Solusi dalam R adalah membentuk matriks 4 dimensi dan menjalankan table() di atasnya.

Tentu saja, hal ini dapat dicapai dengan satu putaran. Tapi loop tidak efektif di EMT atau R. Di EMT, kita bisa menulis loop di C dan itu akan menjadi solusi tercepat.

Namun kita ingin meniru perilaku R. Untuk melakukannya, kita perlu meratakan perkalian ab dan membuat matriks ab-cd.

```
>a=0:6; b=a'; p=flatten(a*b); q=flatten(p-p'); ...
u=sort(unique(q)); f=getmultiplicities(u,q); ...
statplot(u,f,"h"):
```



Selain multiplisitas eksak, EMT dapat menghitung frekuensi dalam vektor.

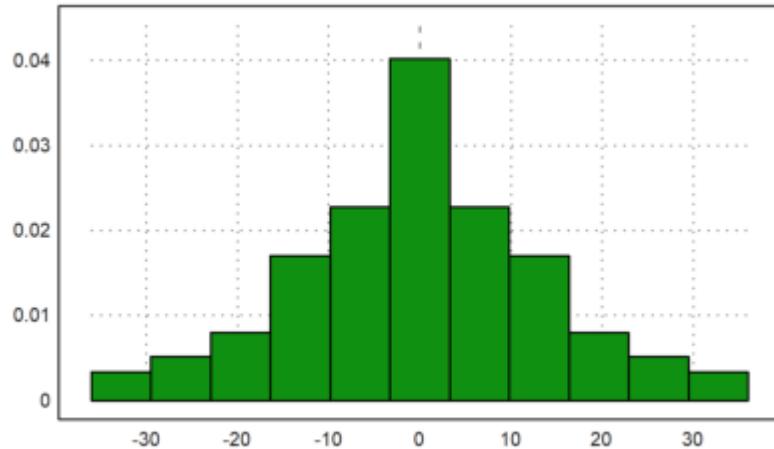
```
>getfrequencies(q,-50:10:50)
```

```
[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]
```

Perintah diatas digunakan untuk menghitung distribusi frekuensi nilai-nilai dalam vektor q dalam rentang dari -50 hingga 50, dengan interval 10. Fungsi ini menghitung berapa banyak nilai dalam q yang jatuh dalam setiap interval: [-50, -40), [-40, -30), ..., [40, 50).

Cara paling mudah untuk memplotnya sebagai distribusi adalah sebagai berikut.

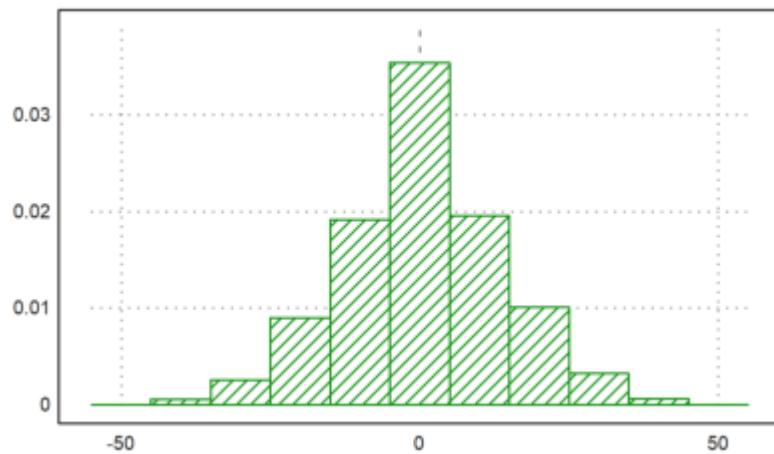
```
>plot2d(q,distribution=11):
```



Namun dimungkinkan juga untuk menghitung terlebih dahulu penghitungan 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="/"):
```



Daftar

EMT memiliki dua jenis daftar. Salah satunya adalah daftar global yang bisa berubah, dan yang lainnya adalah tipe daftar yang tidak bisa diubah. Kami tidak peduli dengan daftar global di sini.

Tipe daftar yang tidak dapat diubah disebut koleksi di EMT. Ini berperilaku seperti struktur di C, tetapi elemennya hanya diberi nomor dan tidak diberi nama.

1. Membuat list dan mengakses elemen dalam list

```
>L={ {"Fred", "Flintstone", 40, [1990,1992]} }
```

```
Fred
Flintstone
40
[1990, 1992]
```

Perintah diatas digunakan untuk membuat list L dengan nama depan Fred, nama belakang Flintstone, usia 40, dan tahun 1990, 1992.

Namun untuk tahun tersebut tidak dapat dipastikan apa arti dari tahun-tahun tersebut, bisa saja tahun kelahiran dan kematian, tahun pendidikan, tahun pekerjaan, atau yang lainnya.

Saat ini unsur-unsur tersebut tidak memiliki nama, meskipun nama dapat ditetapkan untuk tujuan khusus. Mereka diakses dengan nomor.

```
>(L[4]) [2]
```

1992

Perintah diatas digunakan untuk menampilkan list L keempat urutan kedua. Karena pada list L keempat berisi tahun yang dimana terdapat 2 tahun, tahun pertama adalah 1990 dan tahun kedua adalah 1992. Perintah tersebut ingin menampilkan tahun kedua, maka outputnya adalah 1992.

2. Menggabungkan dua list

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

[1, 2, 3]

```
>B := [4,5,6]
```

[4, 5, 6]

```
>C := [A, B]
```

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

3. Mengubah elemen dalam list

```
>D := [7,8,9,10]
```

[7, 8, 9, 10]

```
>D[3] := 99
```

[7, 8, 99, 10]

4. menghitung panjang list

```
>E := [10,20,30,40,50,60,70]
```

[10, 20, 30, 40, 50, 60, 70]

```
>len := length(E)
```

Anda sering kali ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberi tahu Anda tentang banyak cara untuk mencapai hal ini. Fungsi sederhananya adalah writematrix() dan readmatrix().

Mari kita tunjukkan cara membaca dan menulis vektor real ke file.

```
>a=random(1,100); mean(a), dev(a),
```

```
0.50877  
0.29823
```

$$mean = \frac{1}{n} \sum_{i=1}^n x_i$$

$$dev = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Untuk menulis data ke file, kita menggunakan fungsi writematrix().

Karena pengenalan ini kemungkinan besar ada di direktori, di mana pengguna tidak memiliki akses tulis, kami menulis data ke direktori home pengguna. Untuk buku catatan sendiri, hal ini tidak diperlukan, karena file data akan ditulis ke dalam direktori yang sama.

```
>filename="test.dat";
```

Sekarang kita menulis vektor kolom a' ke file. Ini menghasilkan satu nomor di setiap baris file.

```
>writematrix(a',filename)
```

Untuk membaca data, kita menggunakan readmatrix()

```
>a=readmatrix(filename)'
```

```
[0.067788, 0.0069551, 0.79177, 0.46697, 0.88572, 0.62323, 0.3563,  
0.13821, 0.81998, 0.63064, 0.2313, 0.94722, 0.050549, 0.73706,  
0.3564, 0.45624, 0.39517, 0.024241, 0.78341, 0.66949, 0.66402,  
0.92648, 0.16134, 0.22421, 0.67846, 0.17555, 0.74348, 0.13851,  
0.62084, 0.51664, 0.47372, 0.57745, 0.054384, 0.50955, 0.73305,  
0.81468, 0.46121, 0.33495, 0.99671, 0.47853, 0.48392, 0.68367,  
0.69906, 0.54403, 0.87231, 0.11038, 0.63528, 0.44927, 0.38662,  
0.75775, 0.90199, 0.10859, 0.37499, 0.0076804, 0.69771, 0.69744,  
0.70032, 0.32777, 0.71683, 0.33858, 0.95522, 0.8847, 0.62253,  
0.78447, 0.88177, 0.38756, 0.0015708, 0.1791, 0.9956, 0.10918,  
0.20354, 0.38863, 0.69405, 0.78769, 0.26061, 0.049016, 0.042538,  
0.90893, 0.96385, 0.27297, 0.97447, 0.2176, 0.37186, 0.3532,  
0.7827, 0.077767, 0.58132, 0.11367, 0.38659, 0.60442, 0.40113,  
0.92264, 0.33496, 0.088732, 0.51624, 0.18083, 0.79184, 0.60923,  
0.9795, 0.99769]
```

Dan hapus file tersebut.

```
>fileremove(filename);  
>mean(a), dev(a),
```

```
0.50877  
0.29823
```

Fungsi writematrix() atau writetable() dapat dikonfigurasi untuk bahasa lain.

Misalnya, jika Anda memiliki sistem Indonesia (titik desimal dengan koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan dengan titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan koma). File berikut "test.csv" akan muncul di folder saat ini Anda.

```
>filename="test.csv"; ...
writematrix(random(5,3),file=filename,separator=",")
```

Anda sekarang dapat membuka file ini dengan Excel bahasa Indonesia secara langsung.

```
>fileremove(filename);
```

Terkadang kita memiliki string dengan token seperti berikut.

```
>s1:="f m m f m m m f f f m m f"; ...
s2:="f f f m m f f";
```

Untuk melakukan tokenisasi ini, kami mendefinisikan vektor token.

```
>tok:=["f", "m"]
```

f
m

Kemudian kita dapat menghitung berapa kali setiap token muncul dalam string, dan memasukkan hasilnya ke dalam tabel.

```
>M:=getmultiplicities(tok,strtokens(s1))_ ...
getmultiplicities(tok,strtokens(s2));
```

Tulis tabel dengan header token.

```
>writetable(M,labc=tok,labr=1:2,wc=8)
```

	f	m
1	6	7
2	5	2

Untuk statika, EMT dapat membaca dan menulis tabel.

```
>file="test.dat"; open(file,"w"); ...
writeln("A,B,C"); writematrix(random(3,3)); ...
close();
```

The file looks like this.

```
>printfile(file)
```

```
A,B,C
0.8910393909264115,0.7342404864777072,0.1286626471841359
0.6367455442079208,0.4280982647968235,0.6285685540308836
0.5384287521086838,0.8624719326897125,0.3709559205599377
```

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 file.

```
>writetable(L,wc=10,dc=5)
```

A	B	C
0.89104	0.73424	0.12866
0.63675	0.4281	0.62857
0.53843	0.86247	0.37096

Matriks nilai adalah elemen pertama dari L. Perhatikan bahwa mean() di EMT menghitung nilai rata-rata baris matriks.

```
>mean(L[1])
```

0.58465
0.56447
0.59062

File CSV

Pertama, mari kita menulis matriks ke dalam file. Untuk outputnya, kami membuat file di direktori kerja saat ini.

```
>file="test.csv"; ...
M=random(3,3); writematrix(M,file);
```

Here is the content of this file.

```
>printfile(file)
```

0.8819682672467561,0.8665461589845007,0.6002792241162314
0.8917809421203007,0.5190138486754974,0.9963167624952852
0.1287989709283749,0.4810705319470703,0.6487943554052712

CSV ini dapat dibuka pada sistem berbahasa 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.

Namun titik desimal juga merupakan format default untuk EMT. Anda dapat membaca matriks dari file dengan readmatrix().

```
>readmatrix(file)
```

0.88197	0.86655	0.60028
0.89178	0.51901	0.99632
0.1288	0.48107	0.64879

Diumungkinkan untuk menulis beberapa matriks ke satu file. Perintah open() dapat membuka file untuk ditulis dengan parameter "w". Standarnya adalah "r" untuk membaca.

```
>open(file,"w"); writematrix(M); writematrix(M'); close();
```

Matriks dipisahkan oleh garis 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 koma). Di Excel 2007, gunakan "save as" dan "other format", lalu pilih "CSV". Pastikan tabel saat ini hanya berisi data yang ingin Anda ekspor.

Ini sebuah contoh.

```
> printfile("excel-data.csv")
```

```
Could not open the file
excel-data.csv
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
```

Seperti yang Anda lihat, sistem bahasa Jerman saya menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubahnya di pengaturan sistem atau di Excel, tetapi hal ini tidak diperlukan untuk membaca matriks menjadi EMT.

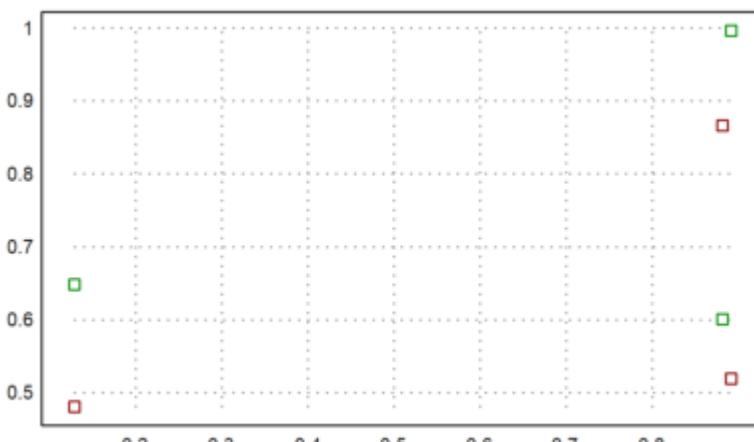
Cara termudah untuk membaca ini ke dalam Euler adalah `readmatrix()`. Semua koma diganti dengan titik dengan parameter `>koma`. Untuk CSV bahasa Inggris, hilangkan saja parameter ini.

```
>M=readmatrix("excel-data.csv",>comma)
```

```
Could not open the file
excel-data.csv
for reading!
Try "trace errors" to inspect local variables after errors.
readmatrix:
    if filename<>"" then open(filename,"r"); endif;
```

Let us plot this.

```
>plot2d(M'[1],M'[2:3],>points,color=[red,green]'):
```



Ada cara yang lebih mendasar untuk membaca data dari suatu file. Anda dapat membuka file dan membaca angka baris demi baris. Fungsi getvectorline() akan membaca angka dari sebaris data. Secara default, ini mengharapkan titik desimal. Tapi bisa juga menggunakan koma desimal, jika Anda memanggil setdecimaldot(",") sebelum Anda menggunakan fungsi ini.

Fungsi berikut adalah contohnya. Itu 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)
```

0.88197	0	0.86655	0	0.60028
0.89178	0	0.51901	0	0.99632
0.1288	0	0.48107	0	0.64879

Dimungkinkan juga untuk membaca semua angka dalam file itu dengan getvector().

```
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
```

0.88197	0	0.86655
0	0.60028	0.89178
0	0.51901	0

Oleh karena itu sangat mudah untuk menyimpan suatu vektor nilai, satu nilai di setiap baris dan membaca kembali vektor ini.

```
>v=random(1000); mean(v)
```

0.49728

```
>writematrix(v',file); mean(readmatrix(file))'
```

0.49728

Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Misalnya, kita menulis tabel dengan header baris dan kolom ke sebuah file.

```
>file="test.tab"; M=random(3,3); ...
open(file,"w"); ...
writetable(M,separator=",",labc=["one","two","three"]); ...
close(); ...
printfile(file)
```

```
one,two,three
0.65,      0.66,      0.41
0.34,      0.55,      0.74
0.24,      0.12,      0.93
```

Ini dapat diimpor ke Excel.

Untuk membaca file di EMT, kami menggunakan `readtable()`.

```
>{M,headings}=readtable(file,>clabs); ...
writetable(M,labc=headings)
```

```
one      two      three
0.65    0.66    0.41
0.34    0.55    0.74
0.24    0.12    0.93
```

Menganalisis Garis

Pada subbab ini sering digunakan untuk memproses atau mengekstrak data dari teks yang berformat khusus, seperti data tabel dalam HTML. Anda bahkan dapat mengevaluasi setiap baris dengan tangan. Misalkan, kita memiliki baris dengan format berikut.

```
>line="2020-11-03,Tue,1'114.05"
```

```
2020-11-03,Tue,1'114.05
```

Pertama, kita akan memisahkan string `line` menjadi bagian-bagian yang lebih kecil, yang dikenal sebagai "token".

```
>vt=strtoks(line)
```

```
2020-11-03
Tue
1'114.05
```

Kemudian kita dapat mengevaluasi setiap elemen garis menggunakan evaluasi yang sesuai.

```
>day(vt[1]); ...
indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])); ...
strrep(vt[3], "'", "")();
```

Dengan menggunakan ekspresi reguler, dimungkinkan untuk mengekstrak hampir semua informasi dari sebaris data.

Selanjutnya, kita akan melihat bagaimana mengekstrak data string yang berisi markup HTML menggunakan ekspresi reguler.

```
>line="<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>";
```

Untuk mengekstraknya, kami menggunakan ekspresi reguler, yang mencari

- tanda kurung tutup `>`, untuk mengindikasikan bahwa kita akan

mencari awal dari elemen yang ada di dalam tag.

- string apa pun yang tidak mengandung tanda kurung akan mencocokkan elemen di dalam tag `<td>`.
- braket pembuka dan penutup menggunakan solusi terpendek,dengan tag pembuka (`<td>`) dan penutup

(

- sekali lagi string apa pun yang tidak mengandung tanda kurung, ini akan menjamin bahwa kita akan mengambil isi yang relevan di dalam tagnya.

- dan tanda kurung buka < menandai bahwa ini adalah akhir dari tag dan awal dari tag baru.

Mencari pola tertentu dalam string line yang menggunakan ekspresi reguler.

```
>{pos,s,vt}=strxfind(line,">([^\>]+)<.+?>([^\>]+)<" );
```

Hasilnya adalah posisi kecocokan, string yang cocok, dan vektor string untuk sub-kecocokan.

Kita akan mengeksekusi elemen-elemen di dalam array atau list vt satu per satu dalam sebuah perulangan.

```
>for k=1:length(vt); vt[k] (), end;
```

```
1145.5  
5.6
```

Berikut 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=cp+strlen(s);  
end;  
return v;  
endfunction
```

Kita akan mengekstrak dan menampilkan semua nilai yang berada di antara tag <td>...</td> dalam baris, dan mencari apakah nilai tersebut numerik atau bukan.

```
>readtd(line+"<td>non-numerical</td>")
```

```
1145.45  
5.6  
-4.5  
non-numerical
```

Membaca dari Web

Situs web atau file dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris.

Dalam contoh, kita membaca versi terkini 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 2024-01-12

Contoh lain membaca URL dengan EMT
"https://mywebsite.com/version.h"

```
>function readversionmywebsite () ...
urlopen("https://mywebsite.com/version.h");
repeat
    until urleof();
    s=urlgetline();
    k=strfind(s,"Release",1);
    if k>0 then substring(s,k,strfind(s,<,k)-1); break; endif;
end;
urlclose();
endfunction
```

```
>readversionmywebsite
```

Karena string "Release" tidak ada di dalam file version.h, maka strfind(s, "Release", 1) akan mengembalikan nilai nol atau tidak menghasilkan indeks yang diperlukan untuk proses pencarian.

Input dan Output 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="tes.e"; ...
writevar(random(2,2),"M",file); ...
printfile(file,3)
```

M = [..
0.4376044954156419, 0.1788663823174511;
0.1555276728327629, 0.9887471739405346];

Sekarang kita dapat memuat file tersebut. Ini akan mendefinisikan matriks M.

```
>load(file); show M,
```

M =
0.4376 0.17887
0.15553 0.98875

Omong-omong, jika writevar() digunakan pada suatu variabel, definisi variabel dengan nama variabel tersebut akan dicetak.

```
>writevar(M); writevar(inch$)
```

```
M = [ ..  
0.4376044954156419, 0.1788663823174511;  
0.1555276728327629, 0.9887471739405346];  
inch$ = 0.0254;
```

Kita juga bisa membuka file baru atau menambahkan file yang sudah ada. Dalam contoh kita menambahkan file yang dibuat sebelumnya.

```
>open(file, "a"); ...  
writevar(random(2,2), "M1"); ...  
writevar(random(3,1), "M2"); ...  
close();  
>load(file); show M1; show M2;
```

```
M1 =  
0.91404 0.92963  
0.47914 0.44517  
M2 =  
0.0097853  
0.78846  
0.53184
```

Untuk menghapus file apa pun, gunakan fileremove().

```
>fileremove(file);
```

Vektor baris dalam suatu file tidak memerlukan koma, jika setiap angka berada pada baris baru. Mari kita buat file seperti itu, tulis 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 = [  
0.133712072096  
0.779233039  
0.250150421757  
0.938737688955  
0.874792276582  
];
```

```
>load(file); M
```

```
[0.13371, 0.77923, 0.25015, 0.93874, 0.87479]
```

LATIHAN

1. Misalkan anda memiliki vektor $x=[2,4,6,8,10]$
 - a. buatkan vektor yang menggabungkan vektor x , angka 0 dan vektor x lagi
 - b. tentukan apakah setiap elemen vektor x lebih besar dari 5 (hasil logika 1 untuk benar dan 0 untuk salah)

```
>x:=[2, 4, 6, 8, 10]; [x, 0, x]
```

```
[2, 4, 6, 8, 10, 0, 2, 4, 6, 8, 10]
```

```
>x>5, %*x
```

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

2. Tentukan matriks X dengan elemen-elemen yang berurutan dari 1 hingga 20 dan susunlah elemen tersebut menjadi matriks berukuran 5x4.

```
>shortformat; X=redim(1:20,5,4)
```

```
1      2      3      4  
5      6      7      8  
9      10     11     12  
13     14     15     16  
17     18     19     20
```

3. Seorang analis memiliki data penjualan harian selama 5 hari(150,200,250,300,350) yang disimpan dalam bentuk vektor sebagai berikut:

- a. mean(rata-rata)
- b. deviasi standar

```
>penjualan=[150,200,250,300,350]
```

```
[150, 200, 250, 300, 350]
```

atau anda bisa memanggil data yang sudah dibuat

```
>filename="penjualan.dat";  
>writematrix(penjualan',filename)  
>penjualan=readmatrix(filename)'
```

```
[150, 200, 250, 300, 350]
```

```
>mean(penjualan)
```

```
250
```

```
>dev(penjualan)
```

```
79.057
```

4. Buat fungsi yang membuka URL
["https://en.wikipedia.org/wiki/Euler_\(software\)"](https://en.wikipedia.org/wiki/Euler_(software))
dan mencari kata "Versi" di dalam URL tersebut, dan tampilkan hasilnya.

```
>function readversionwebsite () ...  
urlopen("https://en.wikipedia.org/wiki/Euler_(software)");  
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 2024-01-12

5.Diberikan data pengukuran tinggi badan pada kelas matematika B adalah sebagai berikut:

Rentang Tinggi (cm)	Jumlah Orang
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

a.)Hitung rata-rata dan deviasi standar dari distribusi tinggi badan ini.

b.)Plot distribusi frekuensi data (diagram batang).

c.)Tambahkan kurva distribusi normal untuk dibandingkan dengan data.

```
>r = 155.5:4:187.5 //Rentang ukuran tinggi badan
```

[155.5, 159.5, 163.5, 167.5, 171.5, 175.5, 179.5, 183.5, 187.5]

```
>v = [22, 71, 136, 169, 139, 71, 32, 8] //Jumlah orang dalam tiap rentang
```

[22, 71, 136, 169, 139, 71, 32, 8]

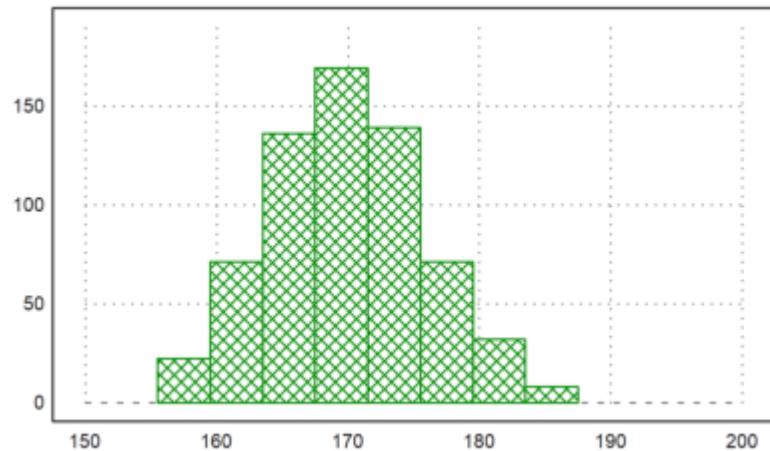
```
>l=fold(r,[0.5,0.5]) //Menghitung titik tengah dari setiap rentang tinggi badan
```

[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]

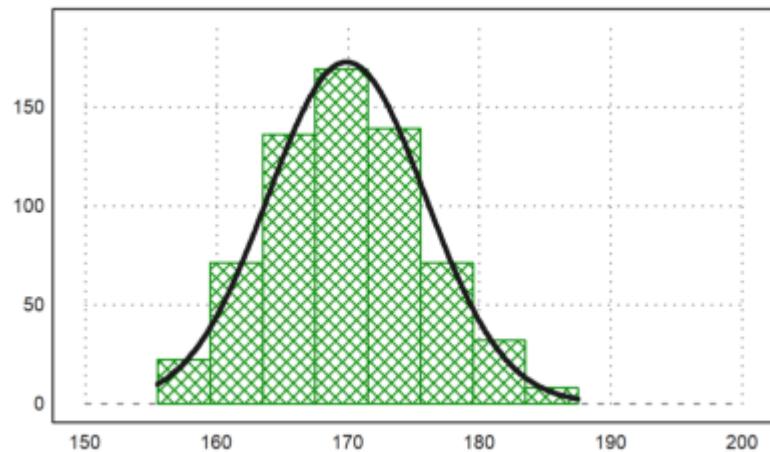
```
>{m,d}=meandev(l,v); m, d, //Hitung rata-rata dan deviasi standar
```

169.9
5.9891

```
>plot2d(r, v, a=150, b=200, c=0, d=190, bar=1, style="\"):
```



```
>plot2d("qnormal(x, m, d) * sum(v) * 4", ...
xmin=min(r), xmax=max(r), thickness=3, add=1):
```



```
>&remvalue();
```

6. Sebuah survei dilakukan untuk mengetahui jumlah jam belajar siswa SMA dalam satu minggu. Berikut data jam belajar dari 10 siswa: 8, 10, 7, 6, 9, 10, 11, 9, 8, 12.

- a) Hitung nilai rata-rata dari data di atas
- b) Tentukan median dari data tersebut.

```
>M=[8,10,7,6,9,10,11,9,8,12];
>mean(M)
```

9

```
>median(M)
```

9

7. Anda diberikan data yang menunjukkan jumlah penjualan barang selama 12 bulan dalam satu tahun berturut-turut 120, 135, 150, 160, 170, 180, 190, 210, 200, 220, 230, 240.

- a) Buatlah plot garis dari data penjualan barang tersebut.
- b) Hitung rata-rata penjualan perbulan.

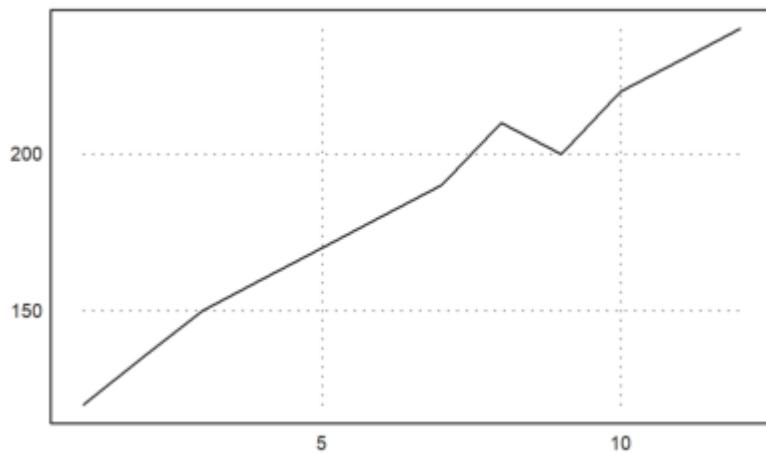
```
>X=[120,135,150,160,170,180,190,210,200,220,230,240]
```

```
[120, 135, 150, 160, 170, 180, 190, 210, 200, 220, 230, 240]
```

```
>Y=[1,2,3,4,5,6,7,8,9,10,11,12]
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

```
>statplot(Y,X,"1"):
```



```
>mean(X)
```

```
183.75
```