

Tugas Latex Aplikasi Komputer



**Raden Mas Farrel Maheswara Kusuma Dewa**

22305141042

Matematika E 2022

**PRODI MATEMATIKA  
DEPARTEMEN PENDIDIKAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI YOGYAKARTA  
2022**

---

## DAFTAR ISI

1 Pekan 2: Belajar Menggunakan Software EMT	2
2 Pekan 3: Menggunakan EMT untuk menyelesaikan masalah-masalah Aljabar	18
3 Pekan 4: Menggunakan EMT untuk mengambar grafik 2 dimensi (2D)	78
4 Pekan 5: Menggunakan EMT untuk mengambar grafik 3 dimensi (3D)	163
5 Pekan 6: Menggunakan EMT untuk kalkulus	246
6 Pekan 8: Menggunakan EMT untuk Geometri	290
7 Pekan 10; Menggunakan EMT untuk Statistika	357

---

---

# BAB 1

---

## PEKAN 2: BELAJAR MENGGUNAKAN SOFTWARE EMT

[a4paper,10pt]article eumat

### Pendahuluan dan Pengenalan Cara Kerja EMT

---

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

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

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

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

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

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

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

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

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

### Komentar (Teks Uraian)

---

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

```
- * Judul
- ** Sub-Judul
- latex: F (x) = \int_a^x f (t) dt
- mathjax: \frac{x^2-1}{x-1} = x + 1
- maxima: 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

---

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

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

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

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

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



---

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

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

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

## Baris Perintah

---

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

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

- Titik koma menyembunyikan output (hasil) dari perintah.

- Sebuah koma mencetak hasilnya.

- Setelah perintah terakhir, koma diasumsikan secara otomatis (boleh tidak ditulis).

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

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

4.90873852123

7.85398163397

## Latihan untuk Anda

---

- Sisipkan beberapa baris perintah baru

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

---

### Jawab :

---

```
>x=3; y=5; z=2; (x*y)^z
```

225

```
>a=50; b=2; c=30; (a/b)*c
```

750

```
>x=10; y=15; (x*y/2)^2
```

5625

```
>p=120; l=60; t=30; p*l*t
```

216000

```
>p=5; q=25; (p^3/q)^p
```

3125

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

- Pastikan untuk menggunakan titik desimal, bukan koma desimal untuk bilangan!

- Gunakan \* untuk perkalian dan ^ untuk eksponen (pangkat).

- Seperti biasa, \* dan / bersifat lebih kuat daripada + atau -.

- ^ mengikat lebih kuat dari \*, sehingga  $\pi * r^2$  merupakan rumus luas lingkaran.

- Jika perlu, Anda harus menambahkan tanda kurung, seperti pada  $2^2(2^3)$ .

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

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

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

1.25

Argumen atau input untuk fungsi ditulis di dalam tanda kurung.

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

0.707106781187

-1

0.5

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

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

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

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

1.61803398875

2

## Latihan untuk Anda

---

- Buka berkas Reference dan baca fungsi-fungsi matematika yang tersedia di EMT.

- Sisipkan beberapa baris perintah baru.

- Lakukan contoh-contoh perhitungan menggunakan fungsi-fungsi matematika di EMT.

---

## Jawab :

---

```
>sqrt(16)*sqrt(25)/10
```

2

```
>(sin(90°)+tan(45°))/cos(60°)
```

4

```
>sqrt(144)+sin(90°)
```

13

```
> sqrt(225)/3, (%*4/10)^5 // Memeriksa solusi (x*4/10)^5=0
```

5

32

---

## Satuan

---

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

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

1609.344

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

```
kilometer$:=1000;
km$:=kilometer$;
cm$:=0.01;
mm$:=0.001;
minute$:=60;
min$:=minute$;
minutes$:=minute$;
hour$:=60*minute$;
h$:=hour$;
hours$:=hour$;
day$:=24*hour$;
days$:=day$;
d$:=day$;
year$:=365.2425*day$;
years$:=year$;
y$:=year$;
inch$:=0.0254;
in$:=inch$;
feet$:=12*inch$;
foot$:=feet$;
ft$:=feet$;
yard$:=3*feet$;
yards$:=yard$;
yd$:=yard$;
mile$:=1760*yard$;
```

```
miles$:=mile$;
kg$:=1;
sec$:=1;
ha$:=10000;
Ar$:=100;
Tagwerk$:=3408;
Acre$:=4046.8564224;
pt$:=0.376mm;
```

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

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

2.48548476895

101.6 mm

## Format Tampilan Nilai

---

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

```
>pi
```

3.14159265359

```
>longest pi
```

3.141592653589793

```
>long pi
```

3.14159265359

```
>short pi
```

3.1416

```
>shortest pi
```

3.1

```
>fraction pi
```

312689/99532

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

```
1612.7  
2.71828182846  
3.141592653589793
```

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

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

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

```
3.141592653589793  
3.14159265359
```

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

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

```
0.392857142857  
11/28
```

## Perintah Multibaris

---

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

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

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

```
-0.138444501319  
-3.61155549868
```

## Menampilkan Daftar Variabel

---

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

```
>listvar
```

```

z          2
r          1.25
x          10
y          15
a          4
b          15
c          2
p          5
l          60
t          30
q          25
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
```

## Matematika Simbolik

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

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

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

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

$$(b + a)^2$$

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

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

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

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

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

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

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

$$a - b$$

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

3628800

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

3628800

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

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

[2, 2, 2, 5, 5]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7

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

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

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

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

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

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

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

## Tampilan Matematika Simbolik dengan LaTeX

---

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

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

$$(b + a)^2$$

```
>${\it expand}((a+b)^2), ${\it factor}(x^2+5*x+6)
```

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

```
>${\it solve}(a*x^2+b*x+c,x) // rumus abc
```

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

```
>${\it ratsimp}((a^2-b^2)/(a+b))
```

$$a - b$$

## Selamat Belajar dan Berlatih!

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

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

- Carilah soal-soal matematika dari buku-buku Matematika.
  - Tambahkan beberapa baris perintah EMT pada notebook ini.
  - Selesaikan soal-soal matematika tersebut dengan menggunakan EMT.
- Pilih soal-soal yang sesuai dengan perintah-perintah yang sudah dijelaskan dan dicontohkan di atas.

Soal no 1

$$125 \times (-20) \times (-14) = \dots$$

```
>125*(-20)*(-14)
```

35000

Soal no 2

$$\mathbf{28 : (-4) \times 5 = \dots}$$

>28 / (-4) \*5

-35

Soal no 3

$$\sin 60^\circ + \cos 45^\circ$$

> $\sin(60^\circ) + \cos(45^\circ)$

1.57313218497

Soal no 4

$$\tan 30^\circ - \cos 30^\circ$$

> $\tan(30^\circ) - \cos(30^\circ)$

-0.288675134595

---

---

## BAB 2

---

# PEKAN 3: MENGGUNAKAN EMT UNTUK MENYELESAIKAN MASALAH-MASALAH ALJABAR

[a4paper,10pt]article eumat  
Raden Mas Farrel Maheswara Kusuma Dewa  
Matematika E  
22305141042

### **EMT untuk Perhitungan Aljabar**

---

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

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

### **Contoh pertama**

---

Menyederhanakan bentuk aljabar:

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

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

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

Menjabarkan:

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

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

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

## Baris Perintah

---

Baris perintah Euler terdiri dari satu atau beberapa perintah Euler diikuti dengan titik koma ";" atau koma ",". Titik koma mencegah pencetakan hasil. Koma setelah perintah terakhir dapat dihilangkan. Baris perintah berikut hanya akan mencetak hasil ekspresi, bukan tugas atau perintah format.

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

16.7551608191

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

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

50.2654824574  
100.530964915

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

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

0.540302305868

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

0.857553215846

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

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

1.41666666667  
1.41421568627  
1.41421356237

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

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

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

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

81

```
>
```

Euler mendukung loop di baris perintah, 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=1; for i=1 to 10; x := (x+2/x)/2, end; // menghitung akar 2
```

```
1.5  
1.4166666667  
1.41421568627  
1.41421356237  
1.41421356237  
1.41421356237  
1.41421356237  
1.41421356237  
1.41421356237  
1.41421356237  
1.41421356237
```

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

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

```
1.41421356237
```

Struktur bersyarat juga berfungsi.

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

```
Thought so!
```

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

Saat Anda menggerakkan kursor di sepanjang garis, pasangan tanda kurung atau kurung buka dan tutup akan disorot. 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(10°)/cos(20°))
```

0.429875017772

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

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

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

2.5

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

## Sintaks Dasar

---

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

Untuk menyetel variabel, gunakan "=" atau ":=". Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir. Spasi tidak masalah. Tapi ruang antara perintah diharapkan.

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

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

30.65625

EMT menggunakan sintaks pemrograman untuk ekspresi. Memasuki

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

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

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

8.77908249441

Untuk menghitung ekspresi rumit seperti

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

Anda harus memasukkannya dalam bentuk baris.

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

23.2671801626

Letakkan tanda kurung dengan hati-hati di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyorot ekspresi 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.

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

0.47619047619

10/21

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

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

14.4978445072

Operator numerik Euler meliputi

```
+ unary atau operator plus  
- unary atau operator minus  
*, /  
. 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. Untuk log.

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

2  
0.5

```
>sin(30°)
```

0.5

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 1/3
```

0.3333333333333333

Representasi ganda internal membutuhkan 8 byte.

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

1.01\*2^-2

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

5.5555555555554\*16^-1

>

## 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.

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

The area of the circle with radius 2 cm is 12.5663706144 cm<sup>2</sup>.

Fungsi print juga 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

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

```
>none
```

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

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

1234.5

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

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

```
affe  
charlie  
bravo
```

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

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

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

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

String Unicode dapat digabungkan seperti string lainnya.

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

= 45°

I

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

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

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

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

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

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

```
Ü is a German letter
```

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

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

```
We have =.
```

Dimungkinkan juga untuk menggunakan entitas numerik.

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

```
Ähnliches
```

## Nilai Boolean

---

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

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

```
0  
1
```

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

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

```
1
```

Operator Boolean mematuhi aturan bahasa matriks.

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

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

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

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

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

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

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

## Format Keluaran

---

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

```
>defformat; pi
```

```
3.14159265359
```

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

```
>longest pi
```

```
3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhex(pi)
```

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

Format output dapat diubah secara permanen dengan perintah format.

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

```
0.33333  
3.14159  
0.84147
```

Standarnya adalah format (12).

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

0.333333333333

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

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

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

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

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

3.1416

Fungsi "format terpanjang" mengatur format skalar juga.

```
>longestformat; pi
```

3.141592653589793

Untuk referensi, berikut adalah daftar format output yang paling penting.

format terpendek format pendek format panjang, format terpanjang  
format(panjang,digit) format baik(panjang)  
fracformat (panjang)  
mengubah bentuk

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

Tetapi format output EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

3.141592653589793

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

3.14159

Standarnya adalah deformat().

```
>deformat; // 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 menggunakannya di atas.

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

```
25/12
```

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

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

```
-1.110223024625157e-16
```

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

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

```
0
```

## Ekspresi

---

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

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

```
12.56637061435917
```

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

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

```
-0.919535764538
```

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

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

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

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

45

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

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

45

Ekspresi dalam x sering digunakan seperti fungsi.

Perhatikan bahwa mendefinisikan fungsi dengan nama yang sama seperti ekspresi simbolik global menghapus variabel ini untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

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

12

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

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

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

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

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

41

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.

Seperi yang Anda lihat dalam contoh berikut, variabel global terlihat selama evaluasi.

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

-0.475

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

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

-0.425

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

## Matematika Simbolik

---

EMT melakukan matematika simbolis dengan bantuan Maxima. Untuk detailnya, mulailah dengan tutorial berikut, atau telusuri referensi untuk Maxima. Para ahli di Maxima harus 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.

```
>$&44!
```

26582715747884487680436258110146158903196385280000000000

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

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

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

2481256778

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

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

2481256778

Untuk mempelajari lebih lanjut tentang fungsi tertentu klik dua kali di atasnya. Misalnya, coba klik dua kali pada "&binomial" di baris perintah sebelumnya. Ini membuka dokumentasi Maxima 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 "dengan".

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

120

Dengan begitu Anda dapat menggunakan solusi persamaan dalam persamaan lain.

Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Alasan untuk ini adalah bendera simbolis 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
```

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

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 "mode kompatibilitas").

```
>&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 melakukannya dengan ":::".

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

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

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

$$\begin{array}{r} 3 \quad x \\ x \quad E \end{array}$$

$$x^3 e^x$$

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(%)
```

$$\begin{array}{r} 5 \\ 125 \quad E \end{array}$$

$$125 e^5$$

$$18551.64488782208$$

```
>fx(5)
```

$$18551.6448878$$

Untuk evaluasi ekspresi dengan nilai variabel tertentu, Anda dapat menggunakan operator "with".

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

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

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

$$2.20079141499189e+7$$

```
>$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)
```

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

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=1+t
```

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

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 opsional 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 memberikan Maxima menemukan nilai numerik.

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

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

$$[-3.23607, 1.23607]$$

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

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

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

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

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

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

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

$$2$$

Ekspresi simbolis dapat memiliki 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(%)
```

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

## Fungsi

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. Alih-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 "menimpa". Menimpa 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°) // redefine sine in degrees  
>sin(45)
```

0.707106781187

Lebih baik kita menghapus redefinisi dosa ini.

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

0.707106781187

## Parameter Default

---

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

16

Menyetelnya akan menimpa nilai default.

```
>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 harus global. Fungsi satu baris dapat melihat variabel global.

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

24

Tetapi parameter yang ditetapkan menimpa nilai global.

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 "&=". Mereka didefinisikan dalam Euler dan Maxima, dan bekerja di kedua dunia. Ekspresi yang mendefinisikan dijalankan melalui Maxima sebelum definisi.

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

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

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

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

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

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

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

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

178.635099908

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik lainnya.

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

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

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

0.703467422498

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

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

0.703467422498

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

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

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

$$(x + 2)^n$$

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

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

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

625

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

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

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

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

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

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

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

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

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

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

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

$$x^3 - x$$

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

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

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

Dengan `:=` fungsinya numerik. Contoh yang baik adalah integral 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 "peta" 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 atau tanpa parameter "basis".

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

2  
6.7

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

2

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

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

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

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

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

17

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

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

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

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

0

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

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

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

Untuk meringkas

- &= 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 ekspresi simbolis. Ambil fungsi berikut.

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

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

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

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

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

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

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

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

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

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

Sekarang kita mencari titik, di mana polinomialnya adalah 2. Dalam solve(), nilai target default y=0 dapat diubah dengan variabel yang ditetapkan.

Kami menggunakan y=2 dan memeriksa dengan mengevaluasi polinomial pada hasil sebelumnya.

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

$$0.966715594851$$

2

Memecahkan 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()
```

$$-0.6180339887498949 \quad 1.618033988749895$$

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

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

$$0$$

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

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

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

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

$$a^x - x^a = 0.1$$

dengan  $a=3$ .

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

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

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

2.54116291558

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

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

2.54116291558

## Menyelesaikan Pertidaksamaan

---

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

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

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

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

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

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

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

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

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

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

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

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

$$\emptyset$$

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

$$\text{universal set}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## Bahasa Matriks

---

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.

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

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

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

Produk matriks dilambangkan dengan titik.

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

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

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

$$[3, 4]$$

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

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

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

```
11  
25
```

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

```
1 0  
0 1
```

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

```
>A.A
```

```
7 10  
15 22
```

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

```
1 4  
9 16
```

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

```
37 54  
81 118
```

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

```
37 54  
81 118
```

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

```
1 1  
1 1
```

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

```
0.333333 0.666667  
0.75 1
```

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

```
-2  
2.5
```

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

```
-2  
2.5
```

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

1	0
0	1

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

1	0
0	1

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

1	4
9	16

Ini bukan produk matriks, tetapi perkalian elemen demi elemen. Hal yang sama berlaku untuk vektor.

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

9
16

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

```
>2*A
```

2	4
6	8

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

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

1	4
3	8

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

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

2	6
6	12

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

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

1	2
1	2

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

1	4
3	8

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

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

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

Sekali lagi, ingat bahwa ini bukan produk matriks!

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

55

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

55

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

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

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

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

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

5

Euler memiliki operator perbandingan, seperti "`==`", yang memeriksa kesetaraan. Kami mendapatkan vektor 0 dan 1, di mana 1 berarti benar.

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

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

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

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

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

[8, 9, 10]

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

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

[64, 81, 100]

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

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

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

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

Kita dapat memeriksa keutamaan. Mari kita cari tahu, berapa banyak kuadrat ditambah 1 adalah bilangan prima.

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

112

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

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

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

Ini mengembalikan indeks elemen, yang bukan nol.

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

1	4
2	1
2	2
3	2

Indeks ini dapat digunakan untuk mengatur elemen ke beberapa nilai.

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

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

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

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

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

Dan dimungkinkan untuk mendapatkan elemen dalam vektor.

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

[0.267829, 0.13673, 0.390567, 0.006085]

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

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

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

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

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

[0.765761, 0.952814, 0.548138]

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

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

[0.765761, 0.952814, 0.548138]

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

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

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

## Fungsi Matriks Lainnya (Membangun Matriks)

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

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

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

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

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

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

```
>A|1
```

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

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

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

1	2	3
1	2	3

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

1	1
2	2
3	3

Tujuan utama dari ini adalah untuk menafsirkan vektor ekspresi untuk vektor kolom.

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

1	1
2	4
3	9

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut.

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

2	1
4	4
[2,	4]
4	

Untuk vektor, ada panjang().

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

9

Ada banyak fungsi lain, yang menghasilkan matriks.

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

1	1
1	1

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

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

[ 6, 6, 6, 6, 6 ]

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

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

0.66566	0.831835
0.977	0.544258

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

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

1	2	3
4	5	6
7	8	9

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

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

Mari kita uji.

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

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

Fungsi multdup() menduplikasi elemen vektor.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
[0, 5, 0, 6, 0, 0, 0, 7, 0, 8, 0]
```

```
[2, 3, 5, 7, 23, 29, 31, 37, 41, 43, 47]
```

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

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

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

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

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

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Kemudian kita atur diagonal bawah (-1) menjadi 1:4.

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

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

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

Berikut adalah fungsi, yang mengembalikan matriks tri-diagonal.

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

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

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

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

1	2	3
4	5	6
7	8	9

Sekarang kita dapat mengekstrak diagonal.

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

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

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

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

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

## Vektorisasi

---

Hampir semua fungsi di Euler 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(1:3)
```

[1, 1.41421, 1.73205]

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

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

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

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

$$s = t^3 - t$$

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

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

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

Misalnya, vektor kolom dikalikan vektor baris menjadi matriks, jika operator diterapkan. Berikut ini, `v'` adalah vektor yang ditransposisikan (vektor kolom).

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

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

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

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

55

Secara default, vektor baris dicetak dalam format yang ringkas.

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

[1, 2, 3, 4]

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

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

5  
25

Untuk mentranspos matriks kita menggunakan apostrof.

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

1  
2  
3  
4

Jadi kita dapat menghitung matriks A kali vektor b.

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

30  
70

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

```
>v'.v
```

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

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

```
>v.v'
```

30

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

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

30

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 menduplikasinya.

Berikut ini adalah kasus sederhana dengan operator ^.

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

[1, 4, 9]

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

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

1	2	3
2	4	6
3	6	9

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

```
>v.v'
```

14

Ada banyak fungsi 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)
```

[1, 4, 9]

14

[1, 5, 14]

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

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

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

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

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

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

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

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

```
6
```

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

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

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

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

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

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

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

```
4
```

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

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

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

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

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

Dalam contoh berikut, kita menghitung  $t[j]^i$  untuk  $i$  dari 1 hingga  $n$ . Kami mendapatkan matriks, di mana setiap baris adalah tabel  $t^i$  untuk satu  $i$ . Yaitu, matriks memiliki elemen

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

Fungsi yang tidak berfungsi untuk 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([1:5])
```

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

## Sub-Matriks dan Matriks-Elemen

Untuk mengakses elemen matriks, gunakan notasi braket.

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

1	2	3
4	5	6
7	8	9
5		

Kita dapat mengakses satu baris matriks yang lengkap.

```
>A[2]
```

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

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

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

```
2
```

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

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

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

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

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

1	2	3
4	5	6

Kita bahkan dapat menyusun ulang A menggunakan vektor indeks. Tepatnya, kami tidak mengubah A di sini, 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[1:3,2:3]
```

2	3
5	6
8	9

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

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

3
6
9

Atau, biarkan indeks pertama kosong.

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

2	3
5	6
8	9

Kita juga bisa mendapatkan baris terakhir dari A.

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

[ 7, 8, 9 ]

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

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

4	2	3
4	5	6
7	8	9

Kami juga dapat menetapkan nilai ke baris A.

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

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

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

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

5	6	-1
7	8	6
7	8	9

Selain itu, beberapa jalan pintas diperbolehkan.

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

0	0	-1
0	0	6
7	8	9

Peringatan: Indeks di luar batas 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[4]
```

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

## Menyortir dan Mengacak

---

Fungsi sort() mengurutkan vektor baris.

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

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

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

Mari kita mengocok vektor.

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

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

Indeks berisi urutan yang tepat dari v.

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

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

Ini bekerja untuk vektor string juga.

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

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

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

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

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

Fungsi unik mengembalikan daftar elemen unik vektor yang diurutkan.

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

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

Ini bekerja untuk vektor string juga.

```
>unique(s)
```

```
a  
aa  
d  
e
```

## Aljabar linier

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

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

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

```
-4  
4.5
```

Untuk contoh lain, kami membuat matriks 200x200 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(200,200); b=sum(A); longest totalmax(abs(inv(A).b-1))
```

8.790745908981989e-13

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

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

1	2	3
4	5	6
7	8	9

Determinan matriks ini adalah 0.

```
>det(A)
```

0

## 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(%)
```

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

```
>$&invert(A) with a=0
```

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

```
>A &= [1,a;b,2]; $A
```

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

Seperti semua variabel simbolik, matriks ini dapat digunakan dalam ekspresi simbolik lainnya.

```
>$&det(A-x*ident(2)), $&solve(% ,x)
```

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

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

Nilai eigen juga dapat dihitung secara otomatis. Hasilnya adalah 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=4,b=5)
```

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

Dalam ekspresi simbolik, gunakan dengan.

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

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

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)
```

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

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

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

```
1      3.14159  
4          5
```

Masih ada perbedaan antara bentuk numerik dan simbolik. Saat mentransfer matriks ke bentuk simbolis, pendekatan fraksional untuk real akan digunakan.

```
>$&A
```

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

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

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

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

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))
```

```
1.414213562373095
```

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)
```

```
-5.42477960769379
```

## 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 5000 (katakanlah dalam dolar).

```
>K=5000
```

```
5000
```

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

```
>K*1.03
```

```
5150
```

Euler akan memahami sintaks berikut juga.

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

```
5150
```

Tetapi lebih mudah menggunakan faktornya

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

```
1.03
```

```
5150
```

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

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

6719.58189672

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

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

6719.58

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

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

Starting from 5000\$ you get 6719.58\$.

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

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

Real 1 x 11 matrix

5000.00      5150.00      5304.50      5463.64      ...

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

```
>short 0:10
```

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

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

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

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

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

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

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 (K) := round(K*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.

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

```
Real 1 x 11 matrix  
5000.00 5150.00 5304.50 5463.64 ...
```

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

```
>VKr'
```

```
5000.00  
5150.00  
5304.50  
5463.64  
5627.55  
5796.38  
5970.27  
6149.38  
6333.86  
6523.88  
6719.60
```

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

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

```
5150.00  
5000.00 5150.00 5304.50
```

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

```
>VKr[-1], VK[-1]
```

```
6719.60  
6719.58
```

Perbedaannya sangat kecil.

## Memecahkan Persamaan

---

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

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

Kita tidak perlu menentukan q atau R untuk definisi fungsi. Hanya jika kita menjalankan perintah, kita harus mendefinisikan nilai-nilai ini. Kami memilih R=200.

```
>R=200; iterate("onepay",5000,10)
```

```
Real 1 x 11 matrix  
5000.00 5350.00 5710.50 6081.82 ...
```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

```
>R=-200; iterate("onepay",5000,10)
```

```
Real 1 x 11 matrix  
5000.00 4950.00 4898.50 4845.45 ...
```

Kami melihat bahwa 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.

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

```
Real 1 x 51 matrix  
5000.00 4950.00 4898.50 4845.45 ...
```

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

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

48.00

Alasan untuk ini adalah bahwa 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 47 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",5000,till="x<0"); x, n,
```

-19.83  
47.00

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

Iterasinya sama seperti di atas

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

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

-19.83

Sekarang kita bisa menyelesaikan masalah kita.

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

3.15

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(5000,x,3,20)",-200)
```

-336.08

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

$R + qK$

Kita sekarang dapat mengulangi ini.

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

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

Kami melihat sebuah pola. Setelah n periode yang kita miliki

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

Rumusnya adalah rumus 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.

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

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

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

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

```
-19.82504734650985  
-19.82504734652684
```

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

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

```
20.51
```

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

Kita juga dapat menggunakan sisi 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(K, R, P, n) = Kn; $&equ
```

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

Biasanya rumus ini diberikan dalam bentuk

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

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

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

Kita dapat memecahkan tingkat R secara simbolis.

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

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

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(5000, 0, x, 10), x, 0)
```

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

Jelas, tanpa bunga kita harus membayar kembali 10 tarif 500.

Persamaan juga dapat diselesaikan untuk n. Kelihatannya lebih bagus, jika kita menerapkan beberapa penye-derhanaan untuk itu.

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

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

```
>
```

### Contoh Soal

Operasi bentuk aljabar

$$2x^3 + 7y^2 - x^3 + 2y^2$$

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

$$9y^2 + x^3$$

$$(4x^{-2} + y^3)(2x^2 - y^{-4})$$

```
> $&showev (' expand( (4*x^(-2)+y^3)*(2*x^2-y^(-4)) ) )
```

$$\text{expand} \left( \left( 2x^2 - \frac{1}{y^4} \right) \left( y^3 + \frac{4}{x^2} \right) \right) = 2x^2 y^3 - \frac{1}{y} - \frac{4}{x^2 y^4} + 8$$

### Bilangan Asli

```
> longest 1/11
```

0.09090909090909091

### Ekspresi

```
> r:=4; fx="pi*r^3"; longest fx()
```

201.0619298297468

### Matematika Simbolik

```
> $&20!/ (15!*5!)
```

15504

### Fungsi

```
> function f(x) := 2*x+x^2
> f(7)
```

63.00

```
> f(0:2:4)
```

0.00            8.00            24.00

### Memecahkan ekspresi

```
> $&solve (x^2+3*x=30, x)
```

$$\left[ x = \frac{-\sqrt{129} - 3}{2}, x = \frac{\sqrt{129} - 3}{2} \right]$$

### Menyelesaikan Pertidaksamaan

```
>$&fourier_elim([x^2-5<0], [x])
```

$$[5 - x^2 > 0]$$

## Matrix

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

2.00	4.00
6.00	8.00

```
>A'
```

2.00	6.00
4.00	8.00

```
>B=[1,3;5,7]
```

1.00	3.00
5.00	7.00

```
>A.B
```

22.00	34.00
46.00	74.00

## Menyortir dan Mengacak

```
>sort([6,2,9,7,3])
```

2.00	3.00	6.00	7.00	9.00
------	------	------	------	------

```
>v=shuffle(1:3)
```

1.00	3.00	2.00
------	------	------

## Aljabar Linear

```
>A=[1,3;5,7]; B=[4,6]; A/B
```

0.25	0.50
1.25	1.17

## Memecahkan Persamaan

```
> $&fourier_elim([x^2+10=35], [x])
```

$$[x = -5] \vee [x = 5]$$

---

---

## BAB 3

---

# PEKAN 4: MENGGUNAKAN EMT UNTUK MENGAMBAR GRAFIK 2 DIMENSI (2D)

[a4paper,10pt]article eumat  
Raden Mas Farrel Maheswara Kusuma Dewa  
22305141042  
Matematika E 2022

### **Sub Bab 1 \* Menggambar Grafik Fungsi Satu Variabel \* dalam Bentuk**

---

Ekspresi Langsung Ekspresi tunggal

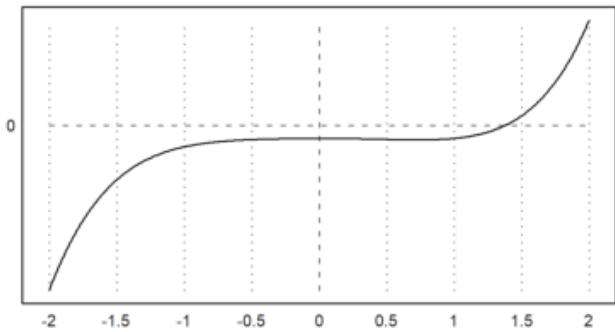
Di dalam program numerik EMT, ekspresi adalah string. Jika ditandai sebagai simbolis, mereka akan mencetak melalui Maxima, jika tidak melalui EMT. Ekspresi dalam string digunakan untuk membuat plot dan banyak fungsi numerik. Untuk ini, variabel dalam ekspresi harus "x".  
ekspresi dalam string

```
>expr := "x^5-x^2-3"
```

```
x^5-x^2-3
```

```
plot ekspresi
```

```
>plot2d(expr,-2,2) :
```

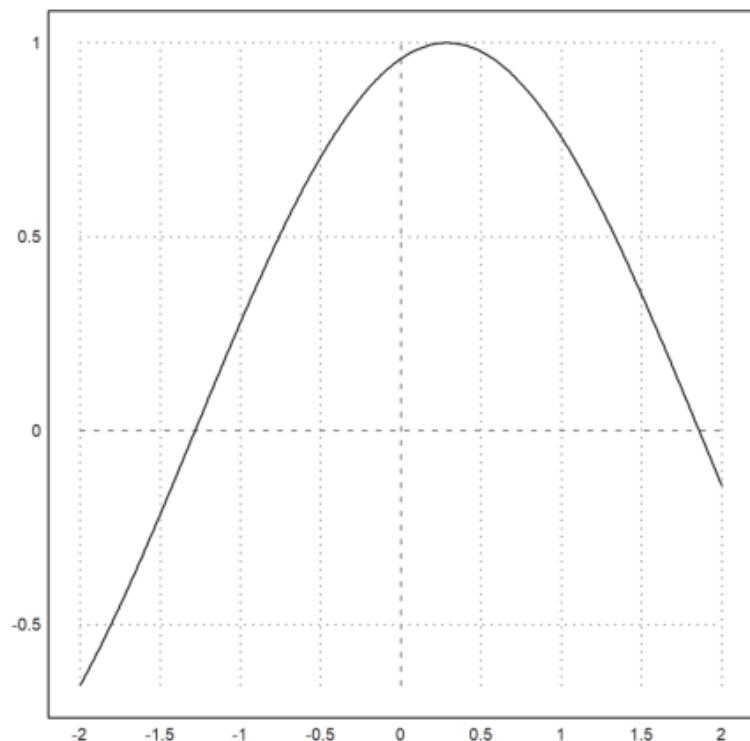


contoh 1

```
>expr := "sin (x-5)"
```

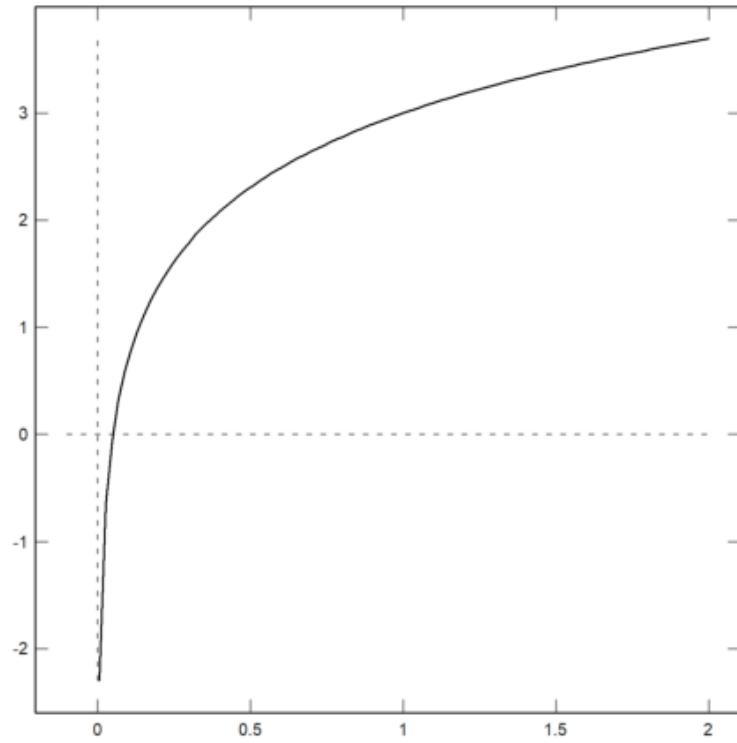
$\sin (x-5)$

```
>aspect (1) ; plot2d(expr,-2,2):
```



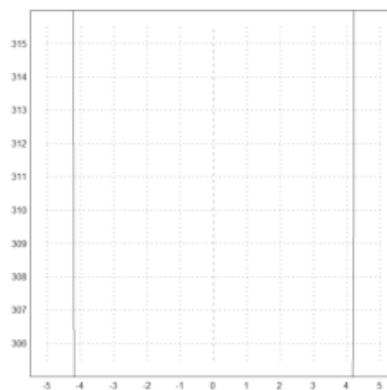
contoh 2 dan penggunaan grid

```
>aspect(1) ; plot2d("log(x) + 3", -0.1, 2, grid=6):
```

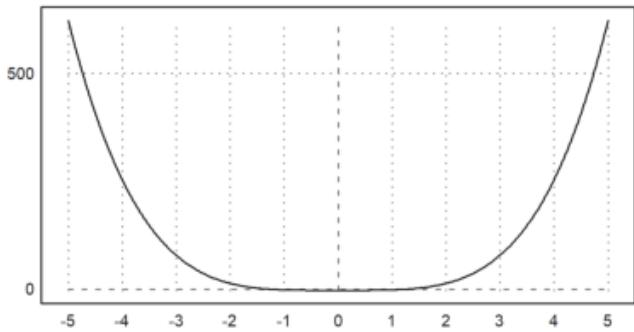


contoh 3 dan penggunaan parameter square (atau >square) untuk memilih y-range secara otomatis

```
>aspect(1,1) ; plot2d("x^4-2", -5, 5, >square); insimg(15)
```

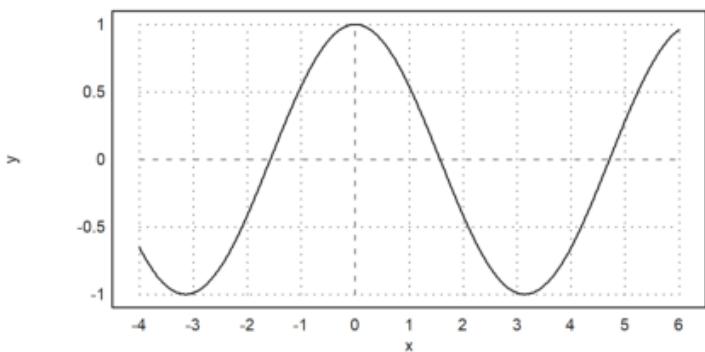


```
>aspect(2) ; plot2d("x^4-2", -5, 5) :
```



contoh 4 dan memberikan nama atau label pada garis sumbu

```
>plot2d("cos(x)", -4, 6, xl="x", yl="y") :
```



## Sub Bab 2

---

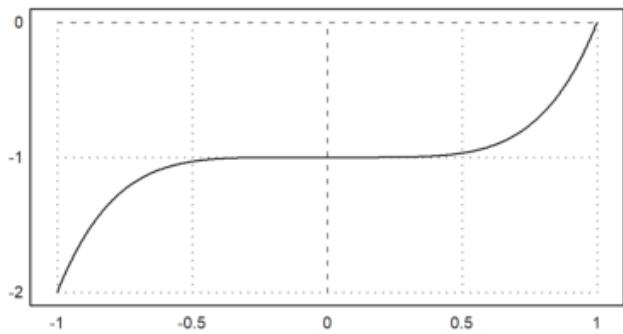
\* Menggambar Grafik Fungsi Satu Variabel \* yang Rumusnya Disimpan dalam Variabel Ekspresi ekspresi

```
>expr &= x^5-1
```

$$x^5 - 1$$

plot dari ekspresi diatas

```
>aspect(2); plot2d(expr,-1,1):
```

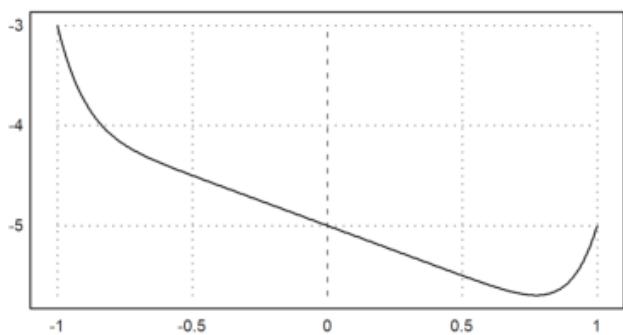


contoh 1

```
>expr := "x^10-x-5"
```

$x^{10}-x-5$

```
>aspect(2); plot2d(expr,-1,1):
```



menggunakan variabel lokal

Ekspresi dapat dievaluasi secara numerik. Variabel x,y,z ditetapkan secara otomatis. Variabel lain dapat ditetapkan berdasarkan parameter yang ditetapkan (variabel lokal) atau melalui variabel global. variabel global adalah variabel yang selalu bisa diakses kapan pun dan di mana pun.

```
>expr &= a*x^5
```

$$a \ x^5$$

menggunakan variabel global

```
>a=6; expr(2.5)
```

585.9375

menggunakan variabel lokal

```
>expr(2.5,a=6)
```

585.9375

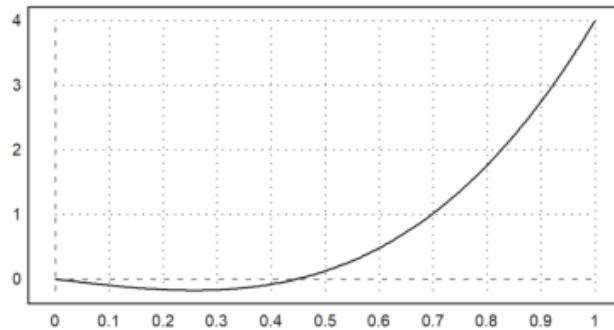
evaluasi langsung

```
>"a*x^5" (3, 4)
```

1458

Oleh karena itu, banyak algoritma EMT yang dapat menggunakan ekspresi dalam x, bukan fungsi. Namun jika parameter tambahan yang tidak bersifat global dilibatkan, fungsi harus diutamakan.  
menggunakan variabel global "a"

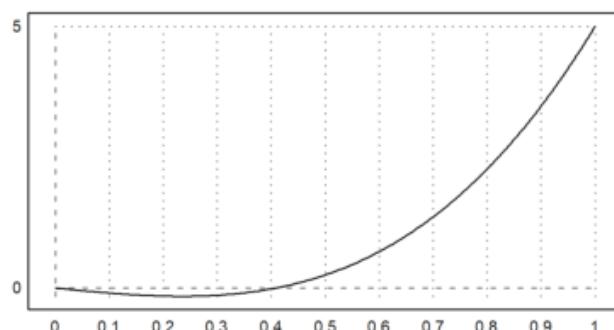
```
>a=5; plot2d("a*x^3-x", 0, 1);
```



```
>function f(x,a) := a*x^3-x
```

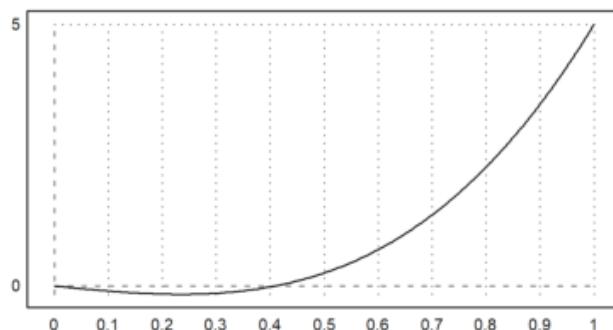
gunakan "a=6" sebagai parameter

```
>plot2d("f", 0, 1; 6);
```



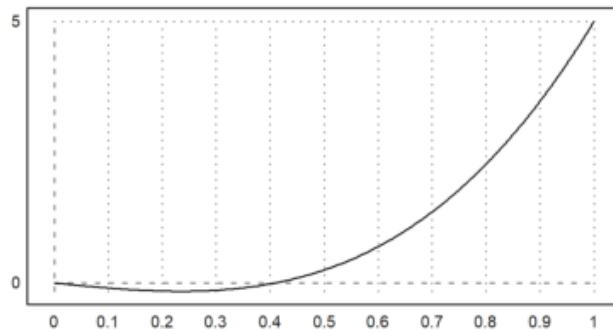
alternatif lain

```
>plot2d({{"f", 6}}, 0, 1):
```



alternatif lain

```
>plot2d("f(x, 6)", 0, 1):
```



## Sub Bab 3

\* Menggambar Fungsi Simbolik Fungsi Plot yang paling penting untuk plot planar adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat diawal program.

plot2d() menerima ekspresi, fungsi, dan data.

Rentang plot diatur dengan parameter yang ditetapkan sebagai berikut

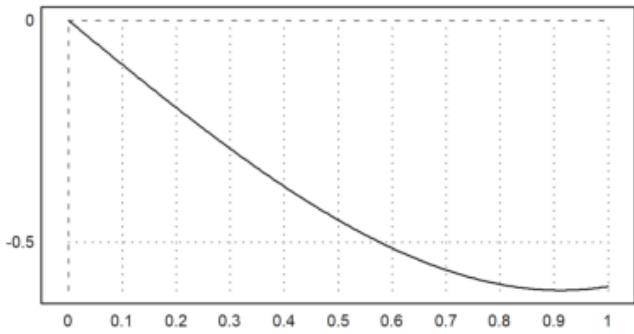
- a,b: rentang x (default -2,2)
- c,d: rentang y (default: skala dengan nilai)
- r: alternatifnya radius di sekitar pusat plot
- cx,cy: koordinat pusat plot (default 0,0)

Keterangan:(menggambar grafik fungsi satu variabel yang fungsinya didefinisikan sebagai fungsi simbolik)

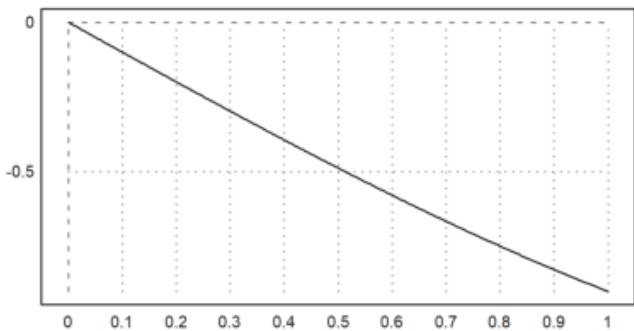
- &: untuk menampilkan variabel pada teks

Berikut adalah beberapa contoh menggunakan fungsi. Seperti biasa di EMT, fungsi yang berfungsi untuk fungsi atau ekspresi lain, jadi kita dapat meneruskan parameter tambahan (selain x) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan koleksi panggilan.

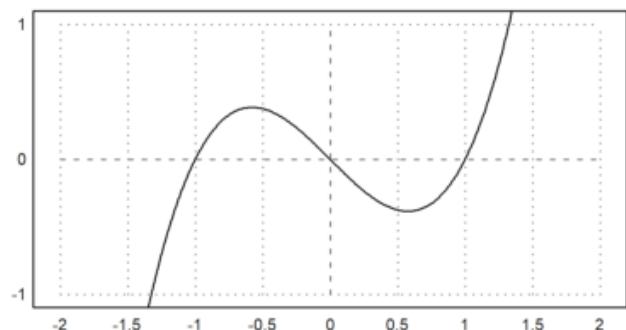
```
>plot2d("f", 0, 1; 0.4): // plot with a=0.4
```



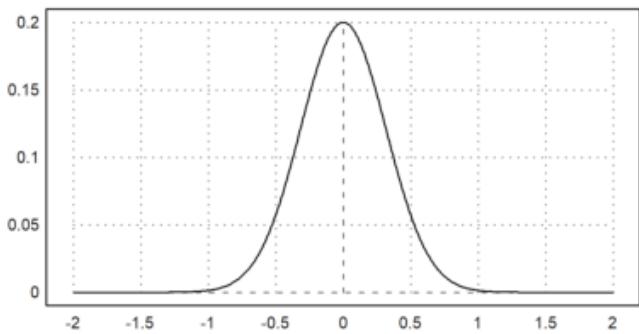
```
>plot2d({{"f", 0.2}}, 0, 1);
>plot2d({{"f(x, b)", b=0.1}}, 0, 1);
```



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



```
>plot2d("exp(-a*x^2)/a");
```



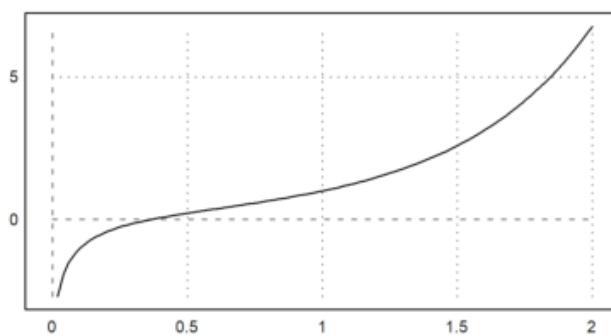
- Berikut merupakan ringkasan dari fungsi yang diterima
- ekspresi atau ekspresi simbolik dalam x
  - fungsi atau fungsi simbolis dengan nama sebagai "f"
  - fungsi simbolis hanya dengan nama f

Fungsi `plot2d()` juga menerima fungsi simbolis. Untuk fungsi simbolis, hanya nama saja yang berfungsi.

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

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

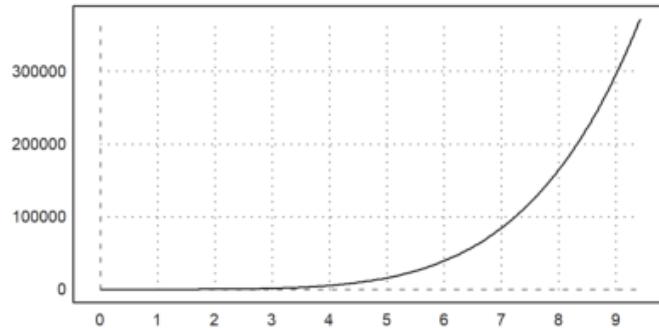
```
>plot2d(f, 0, 2):
```



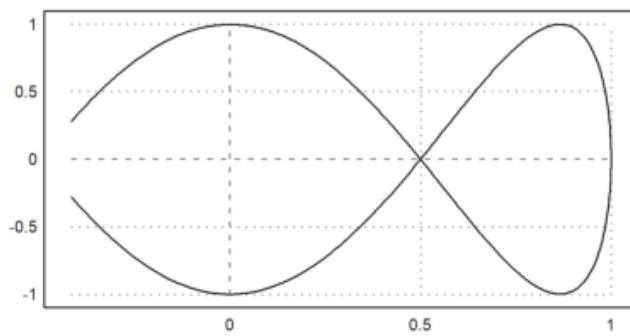
```
>$&expr = sin (x)*exp (-x)
```

$$a x^5 = e^{-x} \sin x$$

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



```
>plot2d("cos(x)","sin(3*x)") :
```



## Sub Bab 4

\* Menggambar Fungsi Numerik Fungsi Numerik adalah sebuah fungsi dengan himpunan bilangan cacah sebagai domain dan himpunan mendasar yang melibatkan hubungan matematis antara bilangan yang menjadi domain dan bilangan sebagai kodomain.

```
>
```

Fungsi numerik memiliki 1 atau lebih variabel independen, yang sering dilambangkan sebagai "X". Variabel X adalah nilai atau parameter yang dapat berubah, dan fungsi numerik menggambarkan bagaimana variabel ini memengaruhi variabel dependen. Variabel dependen adalah hasil perhitungan atau keluaran dari fungsi numerik yang bergantung pada nilai atau perubahan dalam variabel independen.

Dalam EMT cara mendefinisikan fungsi menggunakan syntax function. untuk mendefinisikan fungsi numerik menggunakan tanda ":"

Fungsi numerik menjelaskan bagaimana bilangan dalam domain berhubungan dengan bilangan sebagai ko-domain, biasanya diberikan dalam bentuk rumus matematik(persamaan) atau aturan yang memetakan setiap domain kedalam kodomain yang sesuai. contoh:

$$f(x)=2x+3$$

>

(x)(variabel dependen) adalah fungsi yang memetakan setiap nilai x(variabel independen) kedalam nilai  $2x+3$ . Terdapat berbagai jenis fungsi yang termasuk ke dalam fungsi numerik, diantaranya:

Fungsi linier dengan bentuk umum

$$f(x) = ax + b$$

Fungsi kuadrat dengan bentuk umum

$$f(x) = ax^2 + bx + c$$

Fungsi eksponensial dengan bentuk umum

$$f(x) = ax$$

Fungsi logaritma dengan bentuk umum

$$f(x) = \log a(x)$$

Fungsi trigonometri dengan bentuk umum

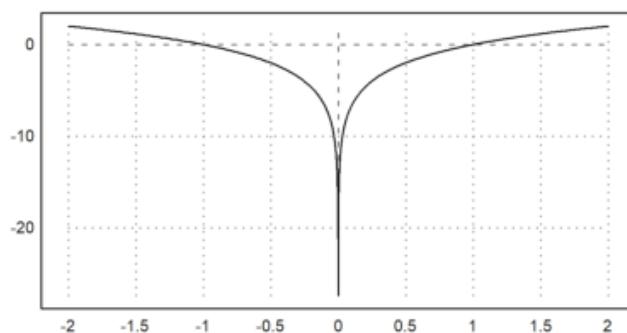
$$f(x) = \sin(x), f(x) = \cos(x)$$

Salah satu cara yang umum digunakan untuk memvisualisasikan fungsi numerik adalah dengan menggambar grafiknya. Grafik ini menggambarkan bagaimana variabel dependen berubah seiring perubahan variabel independen dan membantu dalam memahami sifat-sifat fungsi, seperti titik ekstrim **Contoh soal**

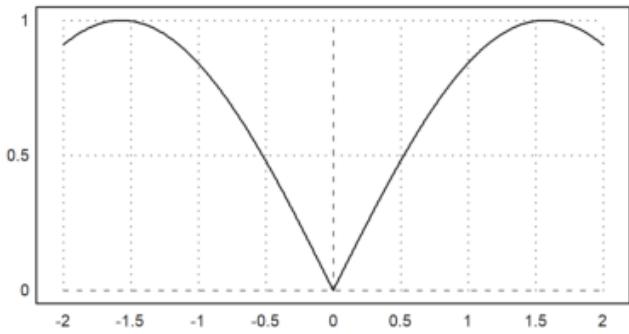
```
>function r(x):= abs(x-10)
>function s(x):= abs(sin(x))
>r(-5)
```

15

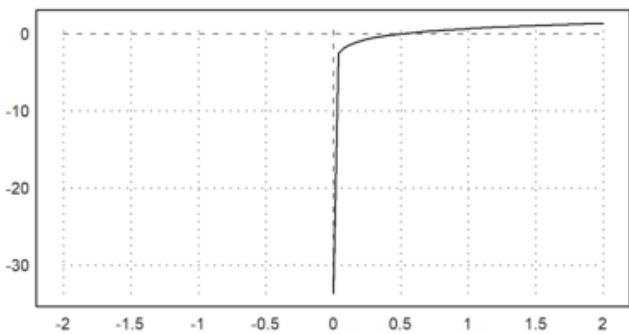
```
>function t(x):=log(x*(2+sin(x/1000)))
>function u(x):=integrate("(sin(x)*exp(-x^2))",0,x)
>function v(x):=logbase((x^2),2)
>plot2d("v"):
```



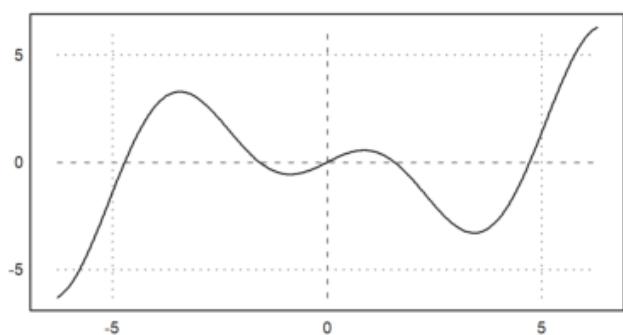
```
>plot2d("s"):
```



```
>plot2d("t", -2, 2);
```



```
>function P(x):=x*cos(x)
>plot2d("P", -2*pi, 2*pi);
```



Fungsi `plot2d()` adalah fungsi serbaguna untuk membuat grafik dalam bidang (grafik 2D). Fungsi ini dapat digunakan untuk membuat grafik fungsi-fungsi satu variabel, grafik data, kurva-kurva dalam bidang, grafik batang (bar plots), grid dari bilangan kompleks, dan grafik implisit dari fungsi dua variabel.

Parameter

`x,y` : persamaan, fungsi, atau vektor data `a,b,c,d` : area plot (default `a=-2, b=2`)

`r` : jika `r` diatur, maka `a=cx-r, b=cx+r, c=cy-r, d=cy+r` `r` bisa berupa vektor `[rx,ry]` atau vektor `[rx1,rx2,ry1,ry2]`.

`xmin,xmax` : rentang parameter untuk kurva

`auto` : tentukan rentang `y` secara otomatis (default)

square : jika benar, mencoba menjaga rentang x-y tetap persegi n : jumlah interval (default adalah adaptif)  
grid : 0 = tanpa grid dan label, 1 = hanya sumbu,

2 = grid normal (lihat di bawah untuk jumlah garis grid) 3 = di dalam sumbu

4 = tanpa grid

5 = grid penuh termasuk margin 6 = tanda di pinggiran

7 = hanya sumbu

8 = hanya sumbu, sub-ticks frame : 0 = tanpa bingkai

framecolor: warna bingkai dan grid

margin : angka antara 0 dan 0,4 untuk margin di sekitar plot color : Warna kurva. Jika ini adalah vektor warna, akan digunakan untuk setiap baris matriks plot. Dalam hal grafik titik, harus berupa vektor kolom. Jika vektor baris atau matriks penuh warna digunakan untuk grafik titik, akan digunakan untuk setiap titik data.

thickness : ketebalan garis untuk kurva

Nilai ini dapat lebih kecil dari 1 untuk garis yang sangat tipis.

style: Gaya plot untuk garis, penanda, dan isian.

Untuk titik gunakan

"[]", "<>", ".", "...", "\*", "+", " ", "-.", "o"

"[]", "<>", "o" (bentuk terisi)

"[]w", "<>w", "ow" (tidak transparan)

Untuk garis gunakan

"-", "-", "-.", ".-", "-.-", "->"

Untuk poligon terisi atau plot batang gunakan

"", "O", "O", "/", "", "/+", " ", "-.", "t"

points : plot titik tunggal sebagai gantinya garis segmen addpoints : jika benar, plot segmen garis dan titik  
add : tambahkan plot ke plot yang ada

user : aktifkan interaksi pengguna untuk fungsi delta : ukuran langkah untuk interaksi pengguna

bar : plot batang (x adalah batas interval, y adalah nilai interval) histogram : plot frekuensi x dalam n subinterval

distribusi=n : plot distribusi x dengan n subinterval even : gunakan nilai antar untuk histogram otomatis.

steps : plot fungsi sebagai fungsi langkah (steps=1,2)

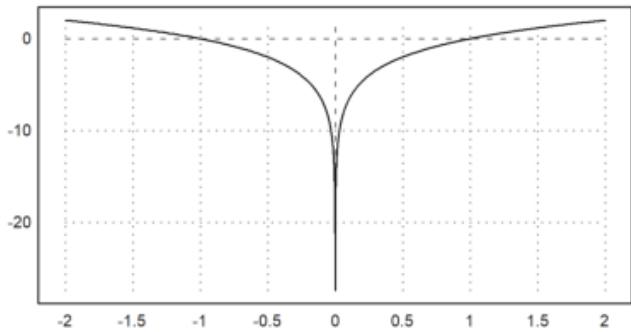
adaptive : gunakan plot adaptif (n adalah jumlah minimal langkah) level : plot garis level dari fungsi implisit dua variabel

outline : menggambar batas rentang level.

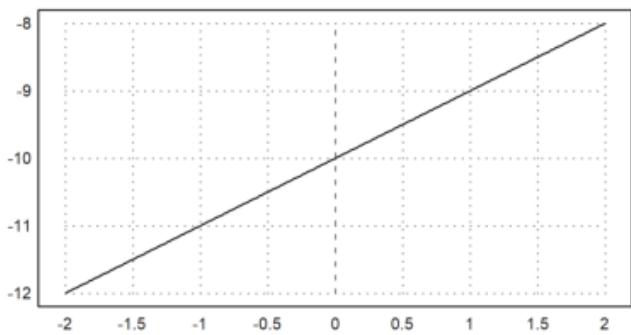
```
>function s(x):=(x-10)
>function r(x):=abs(sin(x))
>s(-5)
```

-15

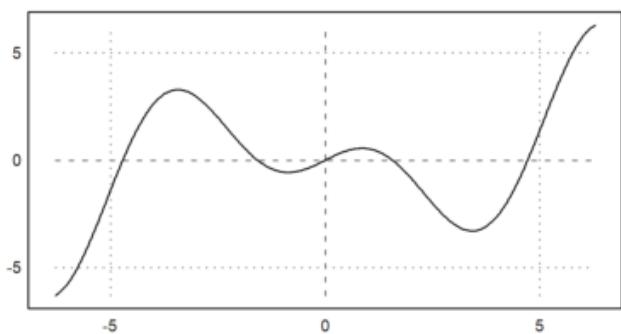
```
>function t(x):=log(x*(2+sin(x/1000)))
>function u(x):=integrate("(sin(x)*exp(-x^2))", 0, x)
>function v(x):=logbase((x^2), 2)
>plot2d("v"):
```



```
>plot2d("s"):
```



```
>function P(x):=x*cos(x)
>plot2d("P", -2*pi, 2*pi):
```



## Sub Bab 5 \* Menggambar Beberapa Kurva Sekaligus Dalam subplot ini,

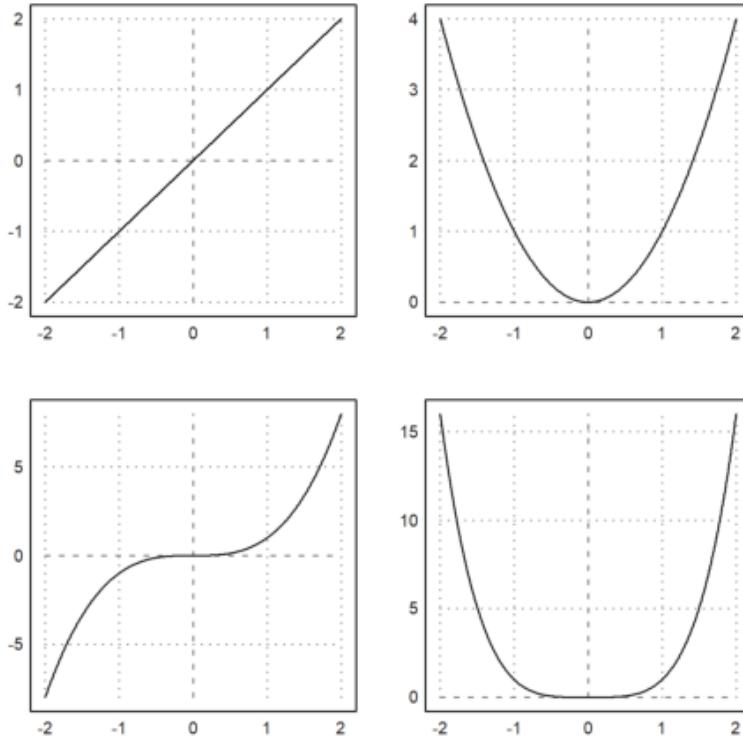
kita akan membahas mengenai cara menggambar beberapa kurva sekaligus. Dalam hal ini kita dapat menggambar beberapa kurva dalam jendela grafik yang berbeda secara bersama-sama. Untuk membuat ini kita dapat menggunakan perintah figure(). Berikut contoh dari menggambar beberapa kurva sekaligus Menggambar plot fungsi

$$x^n, 1 \leq n \leq 4$$

```

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

```



### Penjelasan sintaks dari plot fungsi

$$x^n, 1 \leq n \leq 4$$

- reset;

Perintah ini berguna untuk menghapus grafik yang telah ada sebelumnya, sehingga kita dapat memulai dari awal untuk menggambar grafik

- figure(2x2);

Perintah figure() digunakan untuk membuat jendela grafik dengan ukuran

axb. Dalam kasus ini perintah figure(2,2) memiliki makna bahwa jendela grafik yang dibuat berukuran 2x2. Artinya, akan ada empat jendela grafik yang akan ditampilkan dengan tata letak 2 baris dan 2 kolom.

- for n=1 to 4;

Perintah ini digunakan untuk melakukan pengulangan (looping) perintah sebanyak empat kali, yaitu dari 1 hingga 4.

- figure(n);

Perintah ini digunakan untuk beralih dari jendela grafik satu ke jendela grafik lainnya (jendela grafik ke-n).

- plot2d("x^n");

Perintah plot2d() digunakan untuk membuat plot fungsi matematika.

Dalam hal ini fungsi yang diplot adalah  $x^n$ , di mana n adalah nilai dari variabel yang sedang diulang. Dengan kata lain, ini akan membuat

plot dari  $x^1, x^2, x^3$ , dan  $x^4$  dalam jendela grafik yang sesuai

- end;

Perintah ini menandakan akhir dari looping.

- figure(0);

Perintah ini digunakan untuk beralih kembali ke jendela grafik utama.

Dari sini dapat kita perhatikan untuk membuat kurva fungsi  $x^n$  ( $x$  pangkat  $n$ ) perintahnya tidak ditulis dengan  $(x^n)$  melainkan ditulis dengan (" $x^n$ "). Tanda petik dua ("...") digunakan untuk mengidentifikasi bahwa teks tersebut merupakan ekspresi matematika.

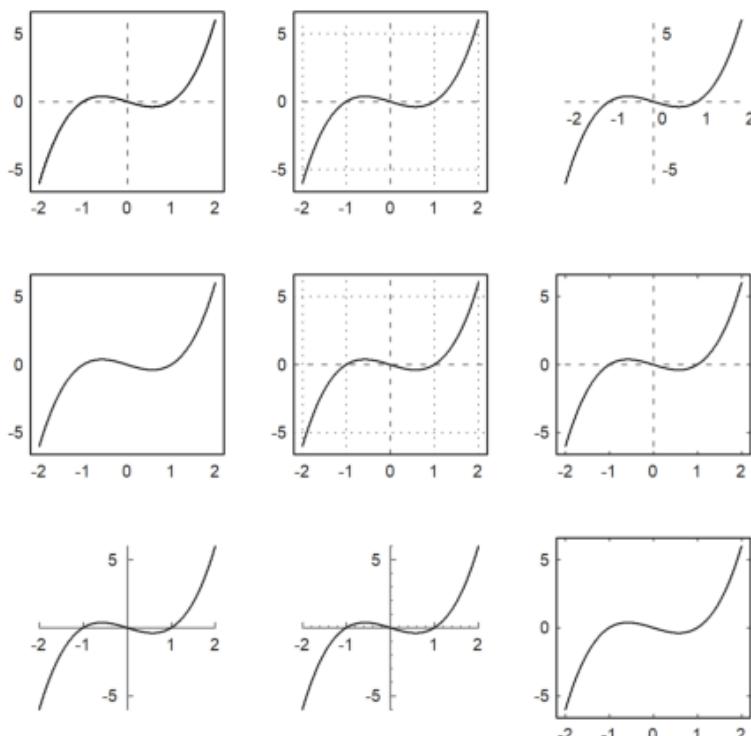
Sedangkan tanda (+) digunakan untuk menggabungkan string dengan nilai yang berubah-ubah atau variabel.

Contoh lain:

Menggambar plot fungsi

$$f(x) = x^3 - x, -2 < x < 2$$

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



Penjelasan sintaks dari plot fungsi

$$f(x) = x^3 - x, -2 < x < 2$$

- reset;

Perintah ini berguna untuk menghapus grafik yang telah ada sebelumnya, sehingga kita dapat memulai dari awal untuk menggambar grafik

- figure (3,3);

Perintah ini digunakan untuk membuat jendela grafik dengan ukuran 3x3. Artinya, akan ada empat jendela grafik yang akan ditampilkan dengan tata letak 3 baris dan 3 kolom.

- for k=1:9;

Perintah ini digunakan untuk melakukan pengulangan (looping) perintah sebanyak sembilan kali.

- figure(n);

Perintah ini digunakan untuk beralih dari jendela grafik satu ke

jendela grafik lainnya (jendela grafik ke-n).

- plot2d("x^3-x",-2,2,grid=k);

Perintah plot2d() digunakan untuk membuat plot fungsi matematika.

Dalam hal ini fungsi yang diplot adalah  $x^3-x$ , dengan batas sumbu x dari -2 hingga 2. Argumen grid=k digunakan untuk mengaktifkan grid pada jendela grafik ke-k.

- end;

Perintah ini menandakan akhir dari looping.

- figure(0);

Perintah ini digunakan untuk beralih kembali ke jendela grafik utama.

Dari contoh diatas dapat kita perhatikan bahwa tampilan plot dari yang ke-1 hingga ke-9 memiliki tampilan yang berbeda-beda. Dalam EMT memiliki berbagai gaya plot 2D yang dapat dijalankan menggunakan perintah grid=n dimana n adalah jumlah langkah minimal. Setiap nilai n memiliki tampilan plot adaptif yang berbeda dalam plot 2D, diantaranya yaitu:

0 : tidak ada grid (kisi), frame, sumbu, dan label, hanya kurva saja

1 : dengan sumbu, label-label sumbu di luar frame jendela grafik

2 : tampilan default

3 : dengan grid pada sumbu x dan y, label-label sumbu berada di dalam jendela grafik

4 : tidak ada grid (kisi), sumbu x dan y, dan label berada di luar frame jendela grafik

5 : tampilan default tanpa margin di sekitar plot

6 : hanya dengan sumbu x y dan label, tanpa grid

7 : hanya dengan sumbu x y dan tanda-tanda pada sumbu.

8 : hanya dengan sumbu dan tanda-tanda pada sumbu, dengan tanda-tanda yang lebih halus pada sumbu.

9 : tampilan default dengan tanda-tanda kecil di dalam jendela

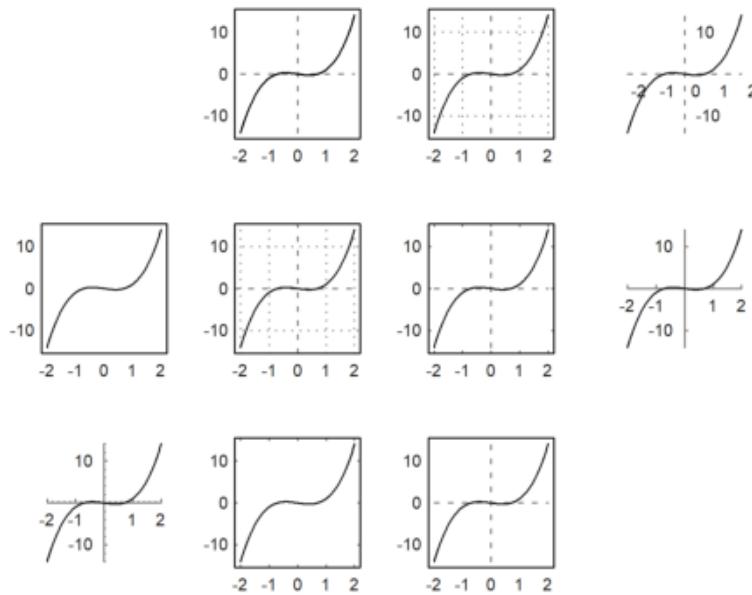
10: hanya dengan sumbu-sumbu, tanpa tanda

Contoh lain:

Menggambar plot fungsi

$$g(x) = 2x^3 - x$$

```
>reset;
>aspect(1.2);
>figure(3,4); ...
> figure(2); plot2d("2x^3-x",grid=1); ... // x-y-axis
> figure(3); plot2d("2x^3-x",grid=2); ... // default ticks
>figure(4); plot2d("2x^3-x",grid=3); ... // x-y- axis with labels inside
> figure(5); plot2d("2x^3-x",grid=4); ... // no ticks, only labels
>figure(6); plot2d("2x^3-x",grid=5); ... // default, but no margin
>figure(7); plot2d("2x^3-x",grid=6); ... // axes only
>figure(8); plot2d("2x^3-x",grid=7); ... // axes only, ticks at axis
>figure(9); plot2d("2x^3-x",grid=8); ... // axes only, finer ticks at axis
>figure(10); plot2d("2x^3-x",grid=9); ... // default, small ticks inside
>figure(11); plot2d("2x^3-x",grid=10); ...// no ticks, axes only
>figure(0):
```



### Penjelasan sintaks dari plot fungsi

$$g(x) = 2x^3 - x$$

- aspect(1.2);

Perintah aspect() digunakan untuk mengatur rasio aspek dari jendela grafik. Hal ini berarti perintah aspect(1.2); akan menghasilkan plot dengan perbandingan rasio panjang dan lebar 2:1.

- figure(3,4);

Perintah ini digunakan untuk membuat jendela grafik dengan ukuran 3x4.

Jadi, akan ada total 12 jendela grafik yang akan ditampilkan dalam tata letak 3 baris dan 4 kolom.

- figure(1); plot2d("x^3-x",grid=0); ...

Adalah perintah untuk beralih ke jendela grafik pertama dan menggambar plot dari fungsi  $x^3 - x$  tanpa grid, frame, atau sumbu.

- figure(2); plot2d("x^3-x",grid=1); ...

Adalah perintah untuk beralih ke jendela grafik kedua dan menggambar plot dari fungsi  $x^3 - x$  dengan grid hanya pada sumbu x dan y.

- figure(3); plot2d("x^3-x",grid=2); ...

Adalah perintah untuk beralih ke jendela grafik ketiga dan menggambar plot dari fungsi  $x^3 - x$  dengan tampilan default, termasuk tanda-tanda default pada sumbu.

- figure(4); plot2d("x^3-x",grid=3); ...

Adalah perintah untuk beralih ke jendela grafik keempat dan menggambar plot dari fungsi  $x^3 - x$  dengan grid pada sumbu x dan y, serta label-label sumbu yang ada di dalam jendela.

- figure(5); plot2d("x^3-x",grid=4); ...

Adalah perintah untuk beralih ke jendela grafik kelima dan menggambar plot dari fungsi  $x^3 - x$  tanpa tanda-tanda sumbu, hanya label-label yang ada.

- figure(6); plot2d("x^3-x",grid=5); ...

Adalah perintah untuk beralih ke jendela grafik keenam dan menggambar plot dari fungsi  $x^3 - x$  dengan tampilan default, tetapi tanpa margin di sekitar plot.

- figure(7); plot2d("x^3-x",grid=6); ...

Adalah perintah untuk beralih ke jendela grafik ketujuh dan menggambar plot dari fungsi  $x^3 - x$  hanya dengan sumbu-sumbu (tanpa grid atau label).

- figure(8); plot2d("x^3-x",grid=7); ...

Adalah perintah untuk beralih ke jendela grafik kedelapan dan menggambar plot dari fungsi  $x^3 - x$  hanya dengan sumbu-sumbu dan tanda-tanda pada sumbu.

- figure(9); plot2d("x^3-x",grid=8); ...

Adalah perintah untuk beralih ke jendela grafik kesembilan dan menggambar plot dari fungsi  $x^3 - x$  hanya

dengan sumbu-sumbu dan tanda-tanda pada sumbu, dengan tanda-tanda yang lebih halus pada sumbu.

- figure(10); plot2d("x^3-x",grid=9); ...

Adalah perintah untuk beralih ke jendela grafik kesepuluh dan menggambar plot dari fungsi  $x^3 - x$  dengan tanda-tanda default kecil di dalam jendela.

- figure(11); plot2d("x^3-x",grid=10); ...

Adalah perintah untuk beralih ke jendela grafik kesebelas dan menggambar plot dari fungsi  $x^3 - x$  hanya dengan sumbu-sumbu, tanpa tanda-tanda.

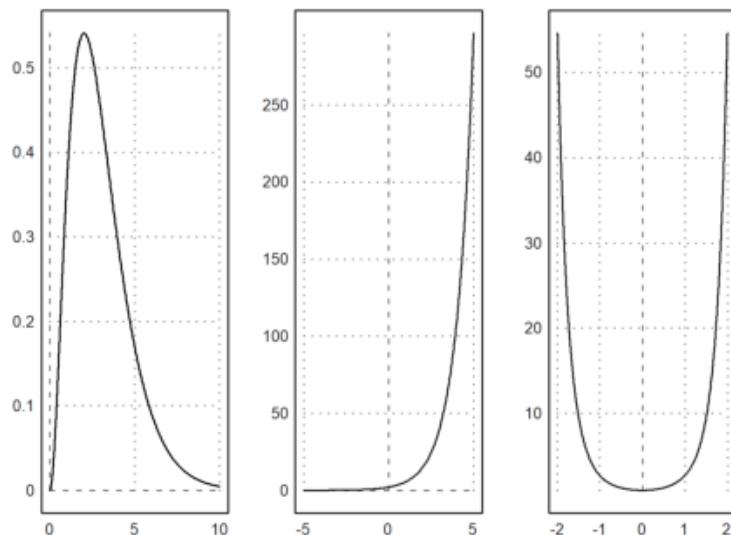
- figure(0);

Adalah perintah untuk beralih kembali ke jendela grafik utama atau jendela grafik dengan nomor 0 setelah semua perintah dalam urutan selesai dieksekusi.

Dari ketiga contoh di atas, dapat kita katakan bahwa untuk menggambar beberapa kurva sekaligus itu dapat dilakukan dengan satu baris perintah ataupun dengan cara mendefinisikannya 1 per 1.

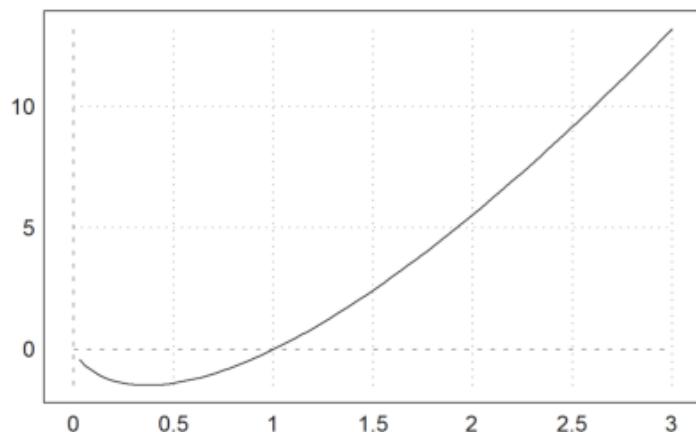
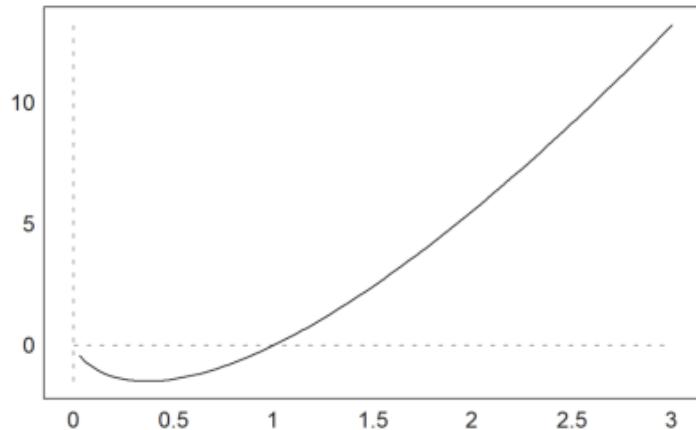
Terlihat beberapa jenis grid memiliki tampilan yang mirip atau sama, seperti 1 dan 2, 2 dan 5, 4 dan 9, 7 dan 8, untuk dapat membedakannya secara lebih jelas, ubah grid dari contoh di bawah ini.

```
>reset;
>aspect(1.3);
>figure(1,3);...
>figure(1); plot2d("x^2*exp(-x)",0,10);...
>figure(2); plot2d("2*exp(x)",-5,5);...
>figure(3); plot2d("exp(x^2)",-2,2);...
>figure(0);
```



Contoh lain:

```
>reset;
>aspect(3/4);
>figure(2,1);...
>for a=1:2; figure(a); plot2d("2*x*log(x^2)",0,3,grid=a); end;...
>figure(0);
```



## **Sub Bab 6 \* Menggambar Beberapa Kurva pada bidang \* koordinat yang**

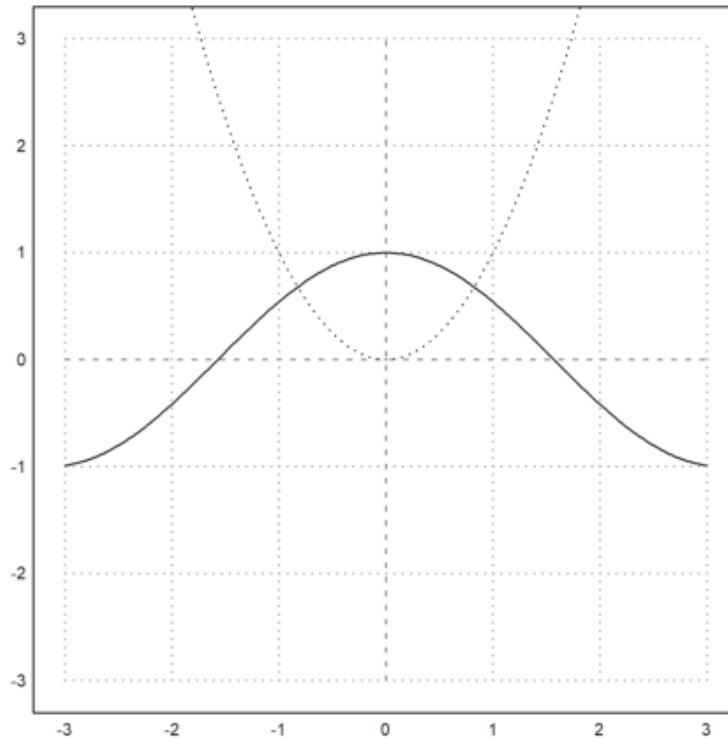
sama Plot lebih dari satu fungsi (multiple function) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu caranya adalah menggunakan >add untuk beberapa panggilan ke plot2d secara keseluruhan, kecuali panggilan pertama.

Berikut contohnya:  
menggambar kurva

$$f(x) = \cos(x)$$

$$f(x) = x^2$$

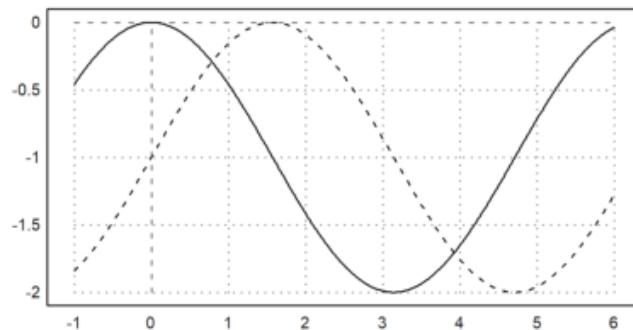
```
>aspect(); plot2d("cos(x)", r=3); plot2d("x^2", style=". ", >add):
```



$$f(x) = \cos(x) - 1$$

$$f(x) = \sin(x) - 1$$

```
>aspect(2); plot2d("cos(x)-1",-1,6); plot2d("sin(x)-1",style="--",>add):
```



Selain menggunakan >add kita juga bisa menambahkannya secara langsung

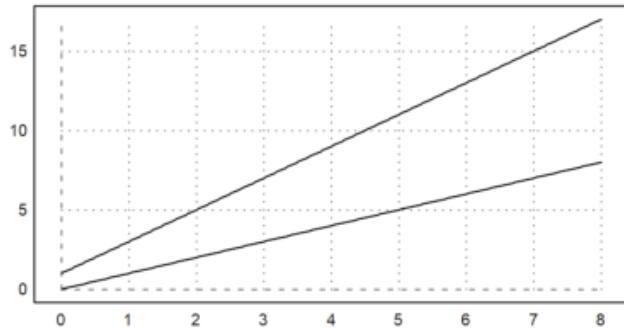
Berikut contohnya:

Menggambar kurva

$$f(x) = 2x + 1$$

$$f(x) = -2x + 1$$

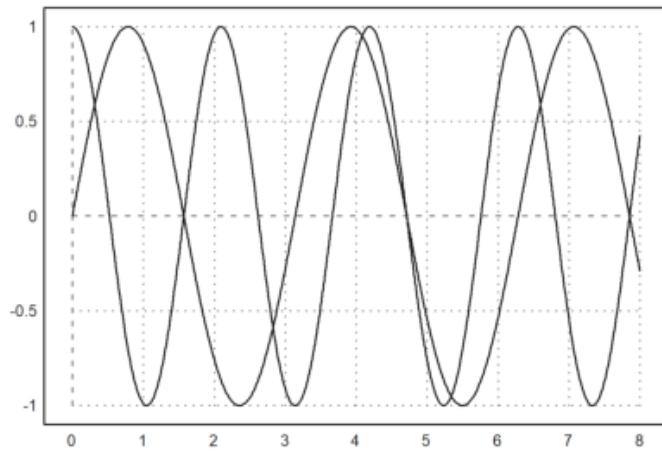
```
>plot2d(["2x+1", "x"], 0, 8) :
```



$$f(x) = \sin(2x)$$

$$f(x) = \cos(3x)$$

```
>aspect(1.5); plot2d(["sin(2x)", "cos(3x)"], 0, 8) :
```



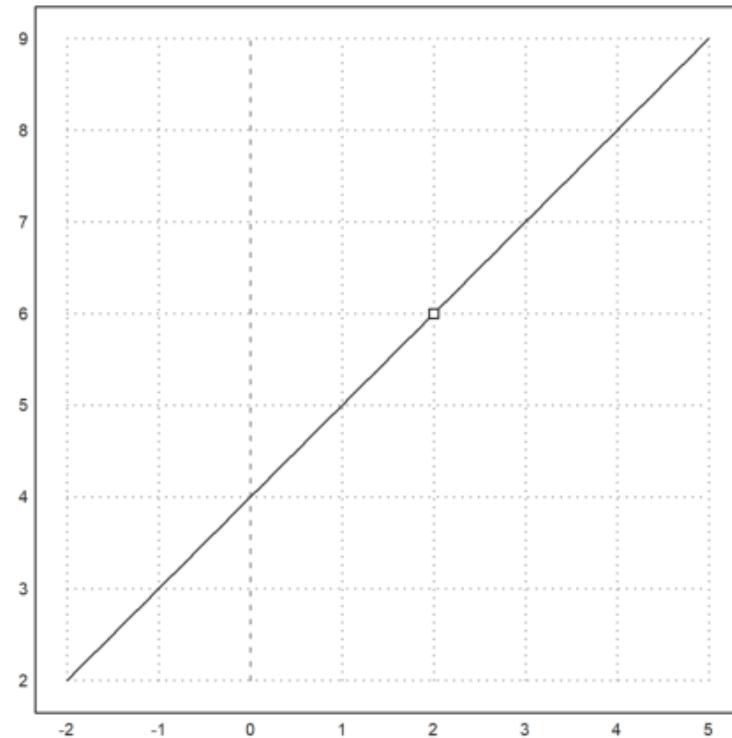
Kegunaan >add yang lain juga bisa untuk menambahkan titik pada kurva.

Berikut contohnya:

Menambahkan sebuah titik di

$$f(x) = x + 4$$

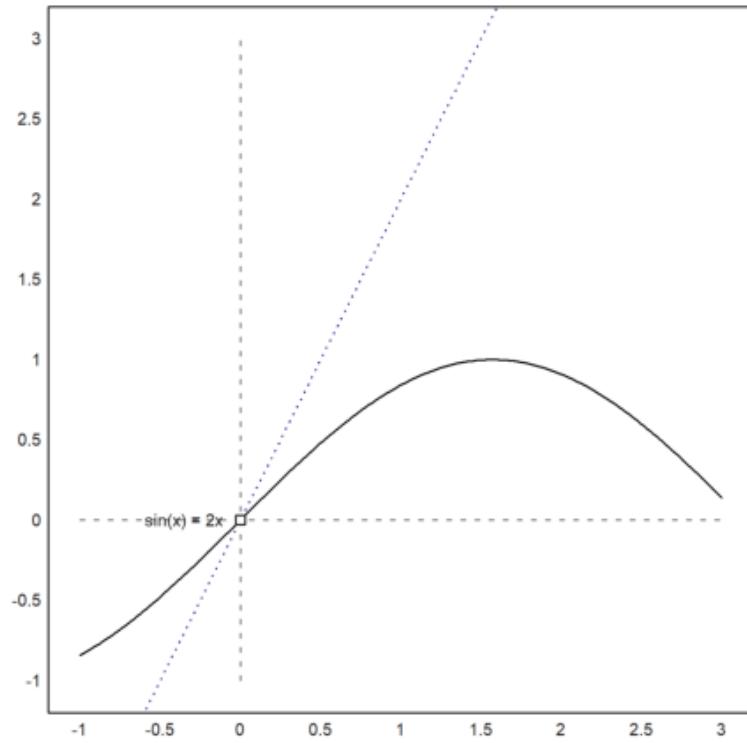
```
>aspect(); plot2d("x+4", -2, 5,); plot2d(2, 6, >points, >add) :
```



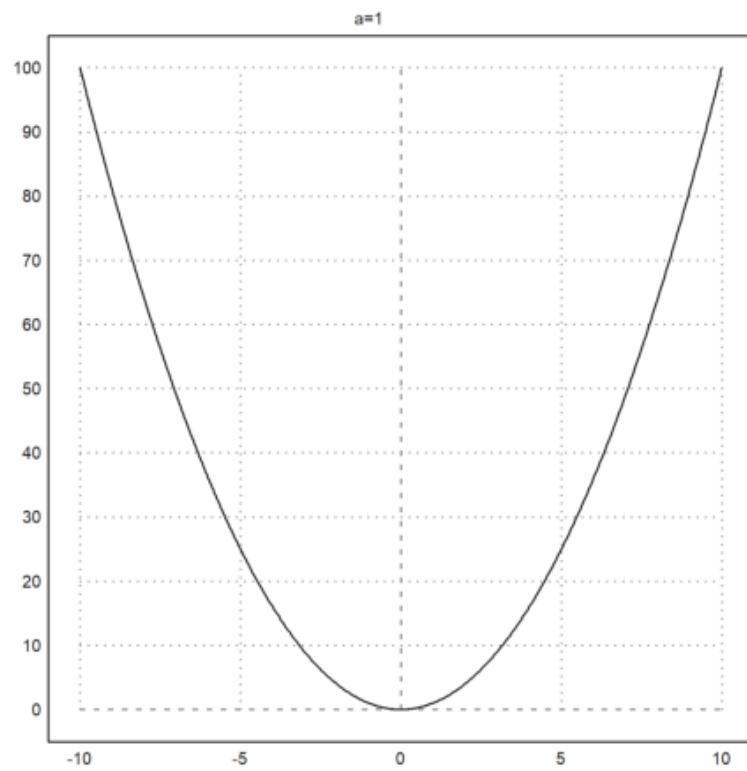
Kita juga bisa mencari titik perpotongan dengan cara berikut:

$$\sin(x) = 2x$$

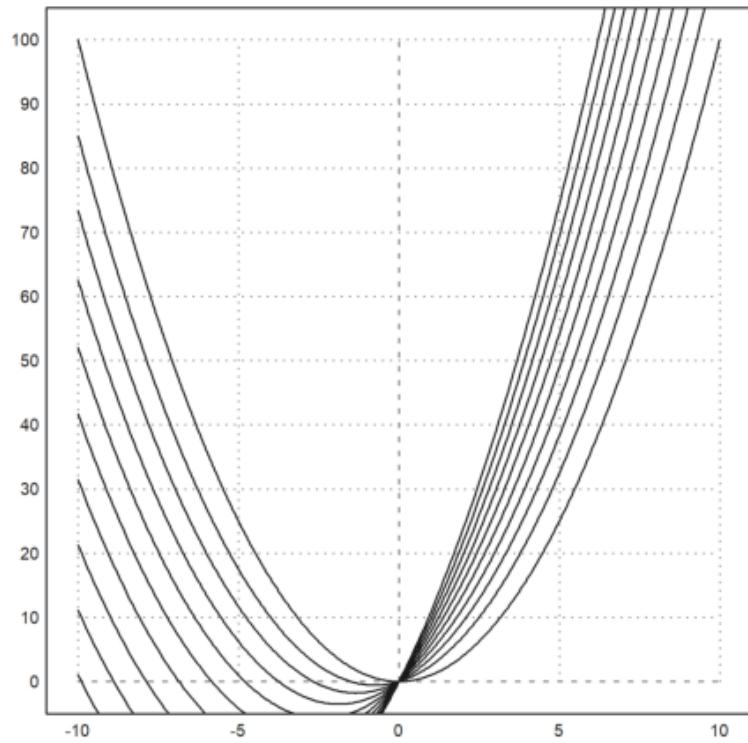
```
>plot2d(["sin(x)","2x"],r=2,cx=1,cy=1, ...
> color=[black,blue],style=["-","."], ...
> grid=1);
>x0=solve("sin(x)-2x",1); ...
> plot2d(x0,x0,>points,>add); ...
> label("sin(x) = 2x",x0,x0,pose="cl",offset=20):
```



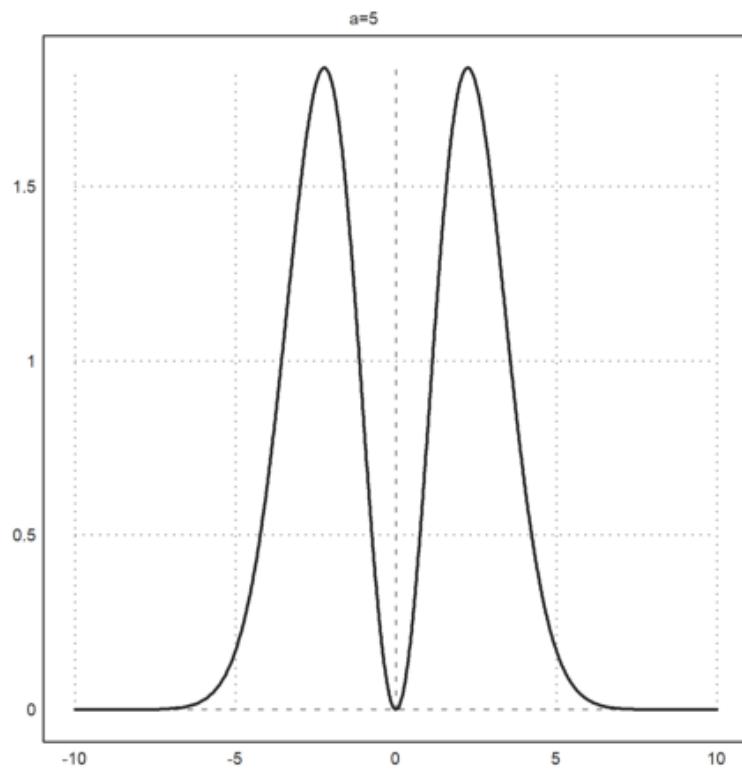
```
>function f(x,a) := x^2+a*x-x/a; ...
>plot2d("f",-10,10;1,title="a=1"):
```



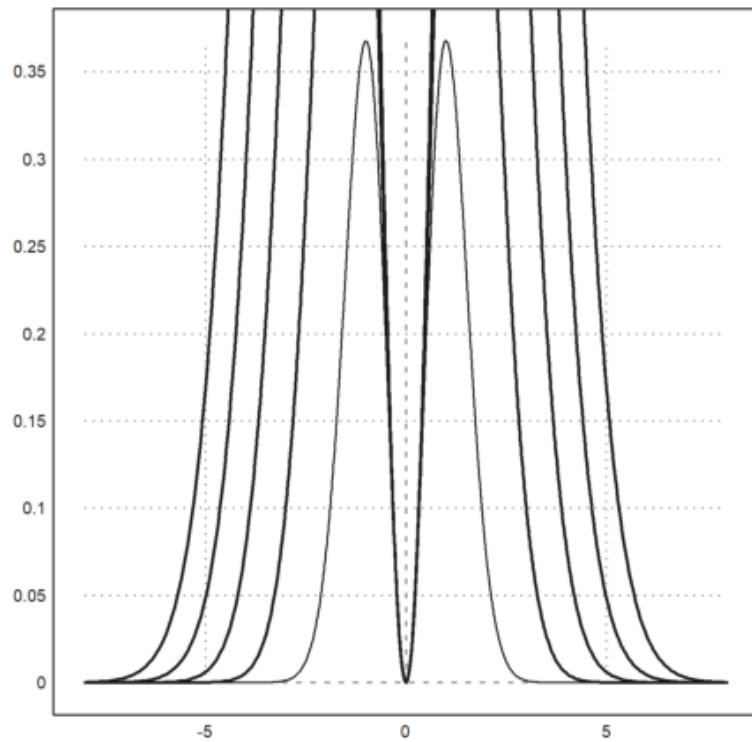
```
> plot2d({{"f",1}},-10,10); ...
>for a=1:10; plot2d({{"f",a}},>add); end;
```



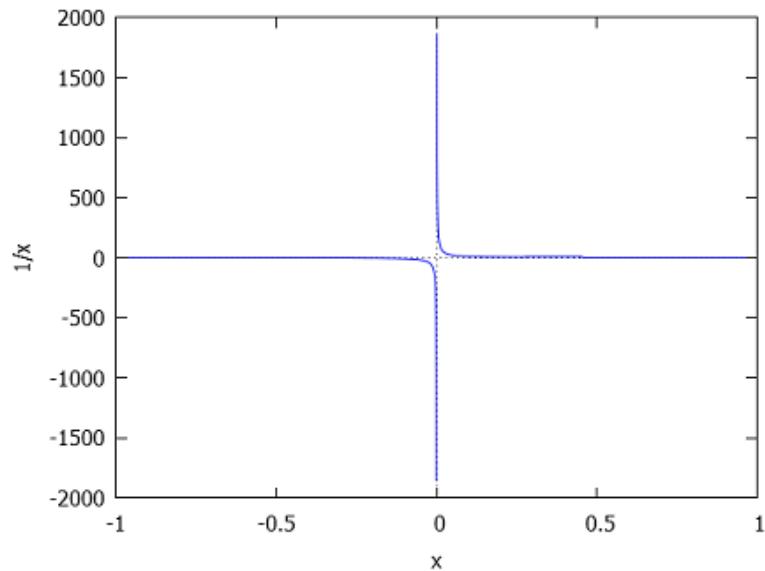
```
>function f(x,a) := x^2*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5");
```



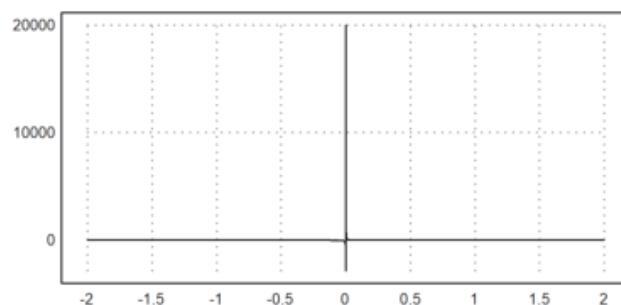
```
>plot2d({{"f",1}},-8,8); ...
>for a=2:5; plot2d({{"f",a}},>add,thickness=2); end;
```



```
>aspect(2.1); &plot2d(1/x, [x,-1,1]):
```



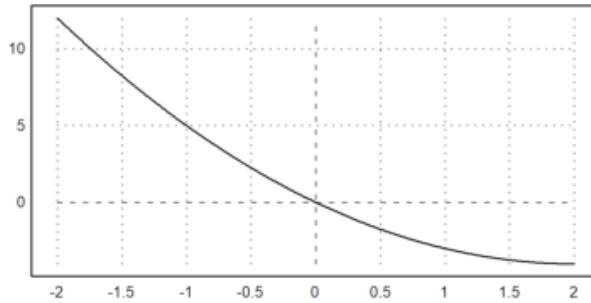
```
>x=linspace(-1,1,50);...
>plot2d("1/x"):
```



## Sub Bab 7

\* Menuliskan Label koordinat, label kurva, \* dan keterangan kurva (legend) Dalam EMT, untuk menambahkan judul dapat dilakukan dengan title="..."  
untuk menambahkan sumbu x dan sumbu y dapat dilakukan dengan x1="...", y1="..." sebagai contoh:

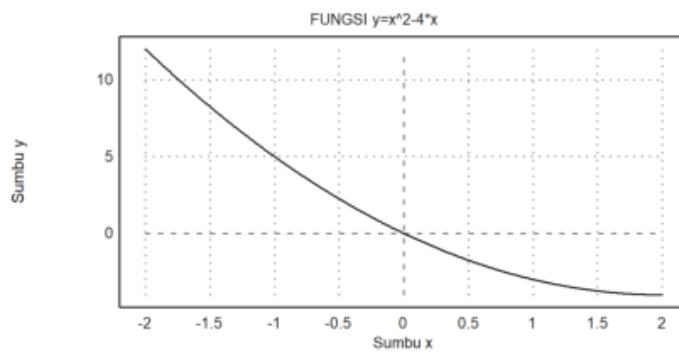
```
>plot2d("x^2-4*x"):
```



untuk menambahkan judul dapat dilakukan dengan title="..."

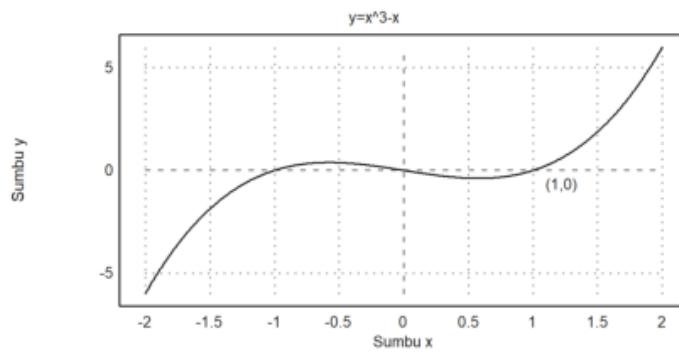
untuk menambahkan sumbu x dan sumbu y dapat dilakukan dengan x1="...", y1="..."

```
>plot2d("x^2-4*x",title="FUNGSI y=x^2-4*x",yl="Sumbu y",xl="Sumbu x") :
```



Selain itu juga dapat dengan cara lain seperti contoh berikut:

```
>expr := "x^3-x"; ...
> plot2d(expr,title="y="+expr,xl="Sumbu x",yl="Sumbu y"); ...
> label("(1,0)",1,0); label("Max",E,expr(E),pos="lc") :
```



## Sub Bab 8

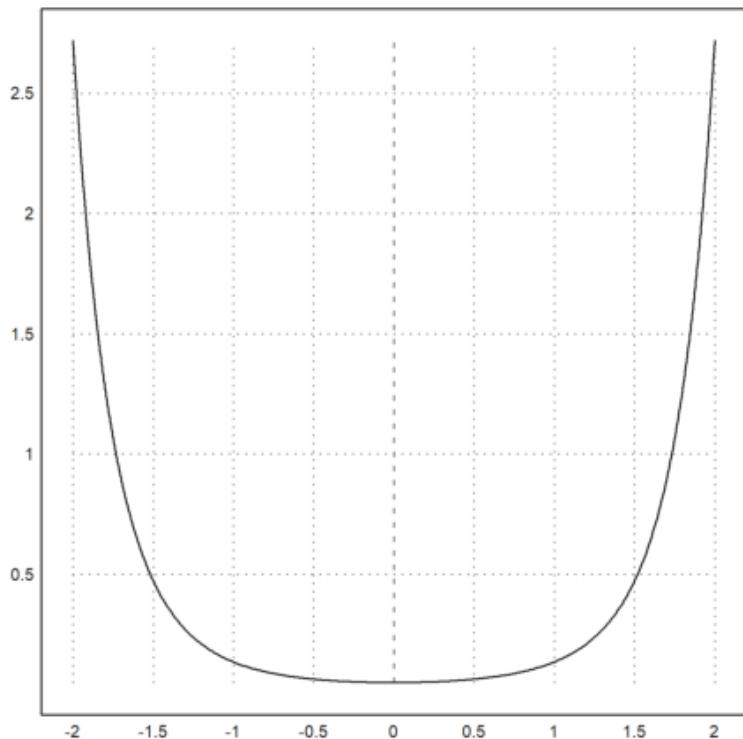
---

\* Mengatur ukuran gambar,format(style),dan warna kurva Untuk mengubah ukuran, dapat dilakukan dengan menggunakan aspect="...", semakin besar nilai aspect, maka ukuran kurva akan semakin kecil, begitupun sebaliknya

untuk mengganti style, dapat dipilih dengan berbagai pilihan  
style="...", dapat dipilih dari, misal : "-","\_","-.","-.".

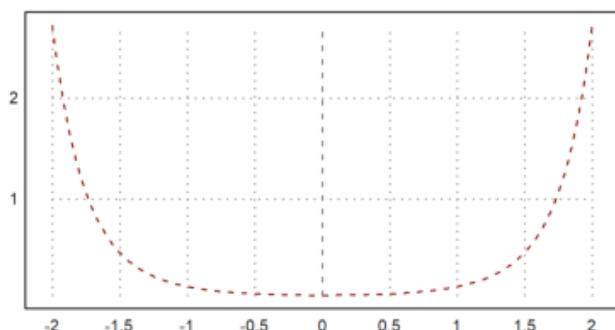
untuk warna dapat dipilih sebagai salah satu warna default  
color="...", warna default= red,green,blue,yellow, dll  
sebagai contoh:

```
>aspect(1); plot2d("exp(x^2-3)");
```



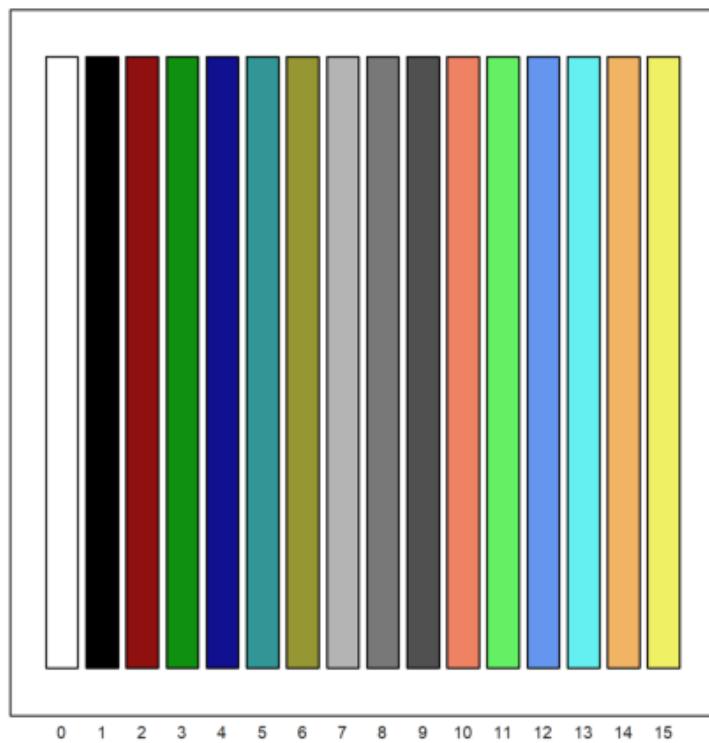
ukuran kurva dapat diganti dengan mengganti nilai aspect="...", semakin besar nilai aspect, maka ukuran kurva akan semakin kecil Untuk mengganti warna dapat ditambahkan dengan color="...", sedangkan untuk mengganti format(style) dapat dilakukan dengan menambahkan style="..."

```
>aspect(2); plot2d("exp(x^2-3)", color=red, style="--");
```



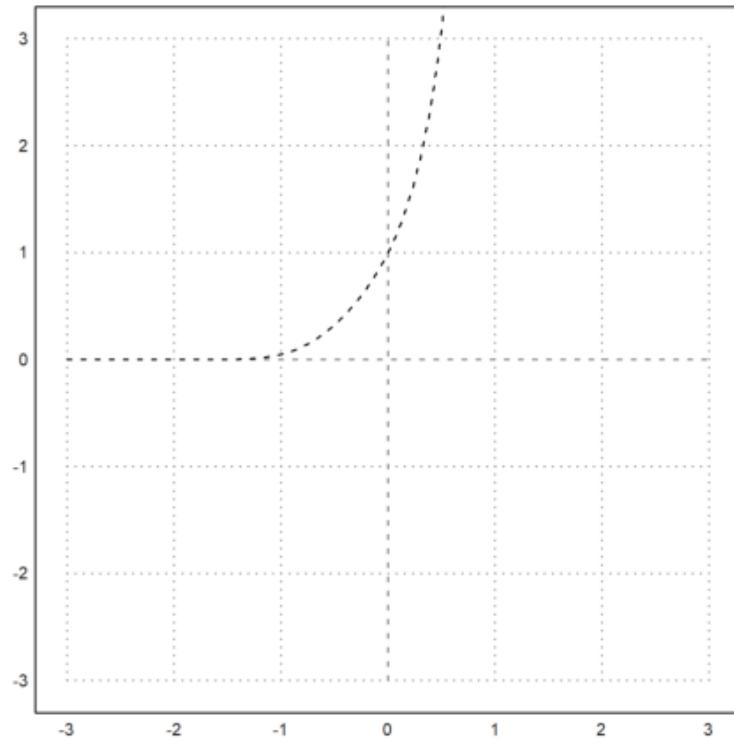
Berikut adalah tampilan warna EMT yang telah ditentukan

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



selain menggunakan warna default, untuk mengubah warna dapat juga dengan menggunakan kode warna di atas  
sebagai contoh:

```
>aspect(1); plot2d("exp(x^3+2*x)",r=3, color=1, style="--") :
```



## Sub Bab 9

---

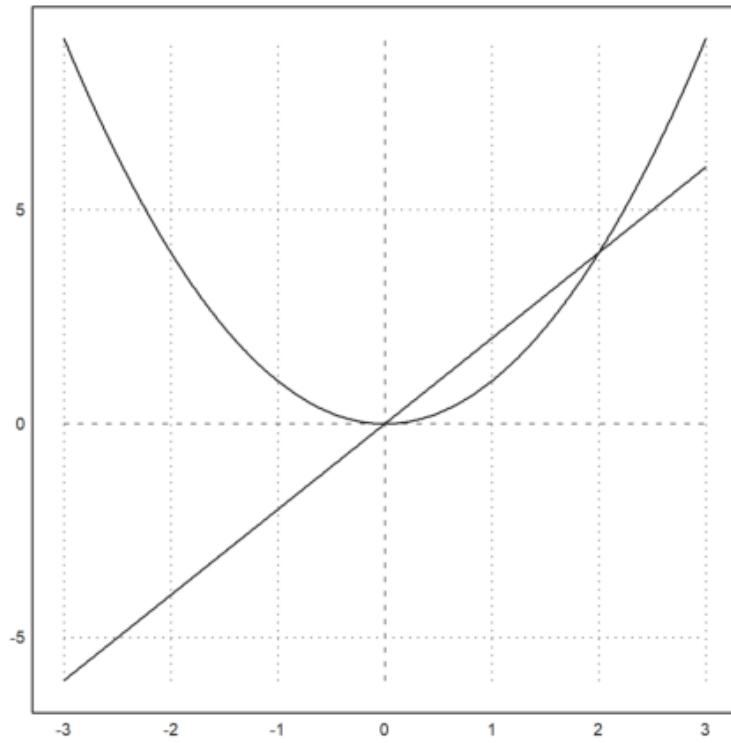
\* Menggambar Sekumpulan Kurva \* dalam satu perintah plot2d. Dalam pembahasan sub-bab 9 kali ini akan membahas mengenai bagaimana menggambar sekumpulan kurva dalam satu perintah plot2d. Menggambar sekumpulan kurva dalam satu perintah plot2d adalah teknik yang digunakan untuk memvisualisasikan beberapa fungsi dalam satu grafik. Ini memudahkan perbandingan antara beberapa kurva.

---

### Contoh

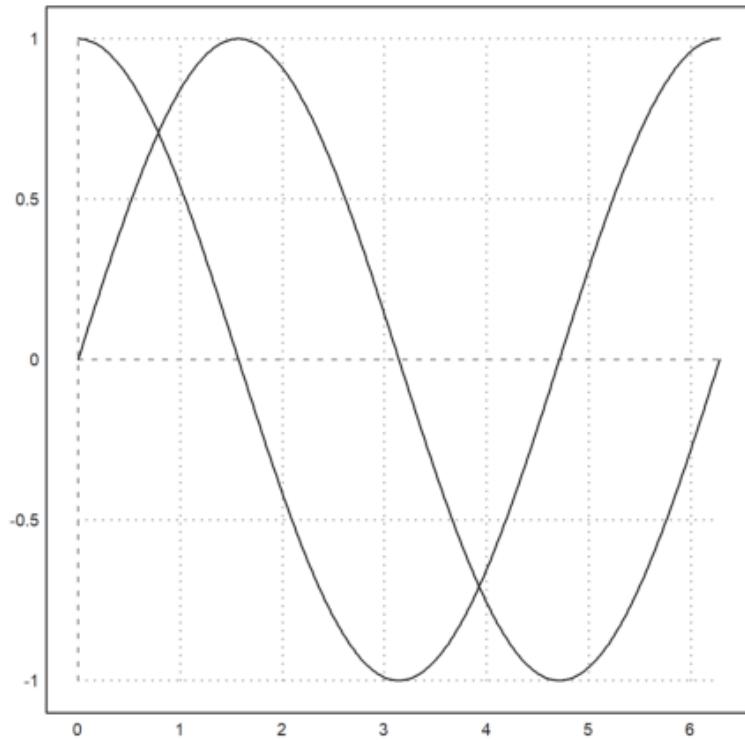
---

```
>plot2d(["x^2", "2*x"], -3, 3):
```



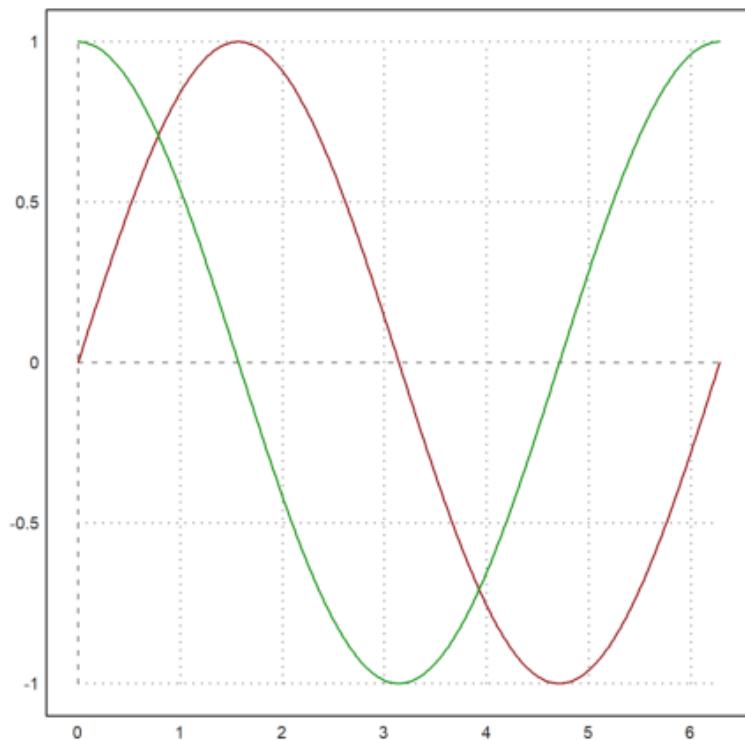
- Dalam contoh ini, merupakan gambar dua kurva sekaligus, yaitu  $x^2$  dan  $2x$ , pada rentang  $-3$  hingga  $3$ .
- Hasilnya akan menunjukkan grafik dari kedua fungsi tersebut, dan titik-titik potongan antara keduanya adalah solusi dari persamaan kuadrat.

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



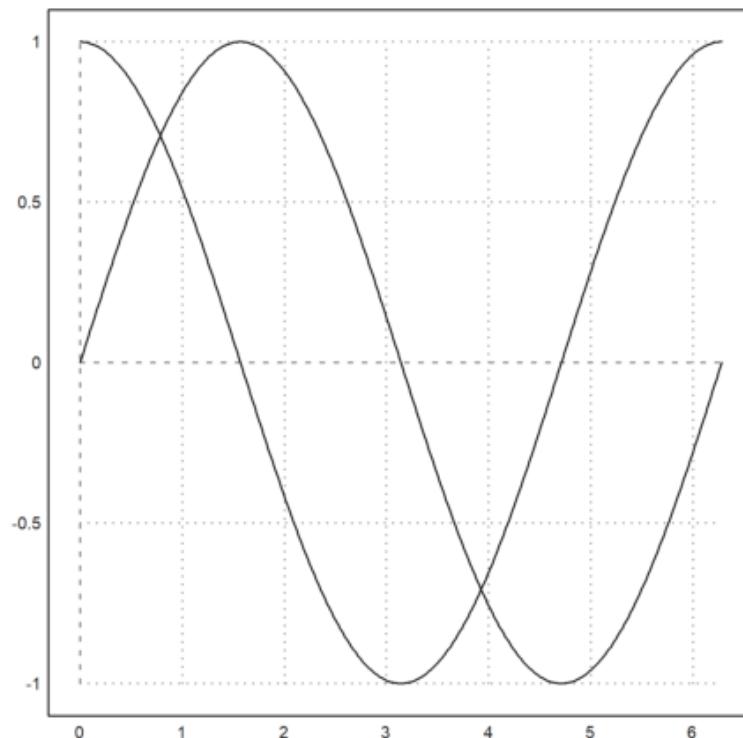
- Pada contoh ini, merupakan gambar dua fungsi trigonometri,  $\sin(x)$  dan  $\cos(x)$ , pada rentang 0 hingga  $2\pi$ .
- Ini akan menghasilkan dua grafik yang memperlihatkan hubungan antara  $\sin(x)$  dan  $\cos(x)$  dalam rentang tersebut.

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



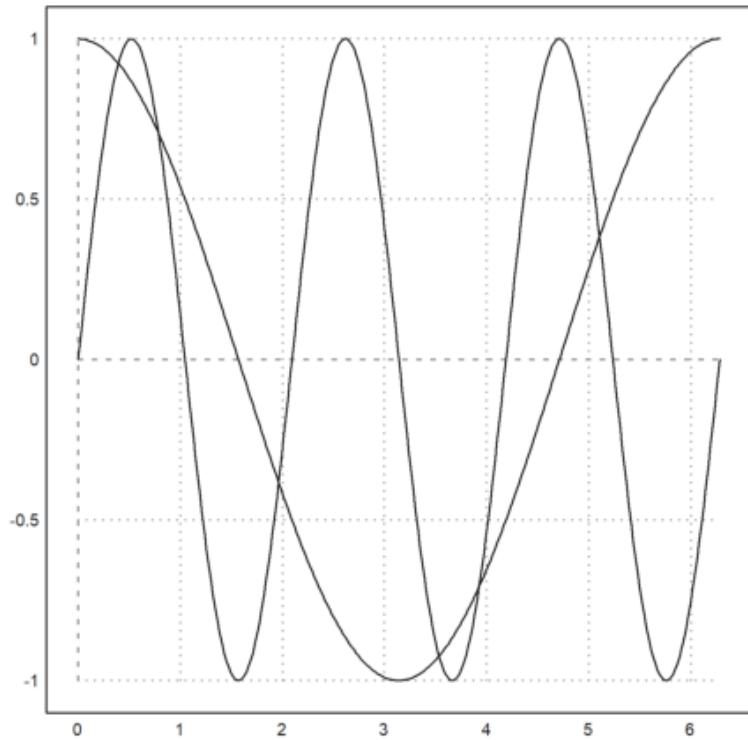
Sama seperti contoh kedua, gambar  $\sin(x)$  dan  $\cos(x)$  pada rentang 0 hingga  $2\pi$ , tetapi Anda juga memberikan warna yang berbeda pada kedua grafik ( $\sin(x)$  berwarna merah dan  $\cos(x)$  berwarna hijau).

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



Dalam contoh ini, menggunakan parameter ‘xmin’ dan ‘xmax’ untuk mengatur rentang tampilan grafik pada 0 hingga  $2\pi$ .

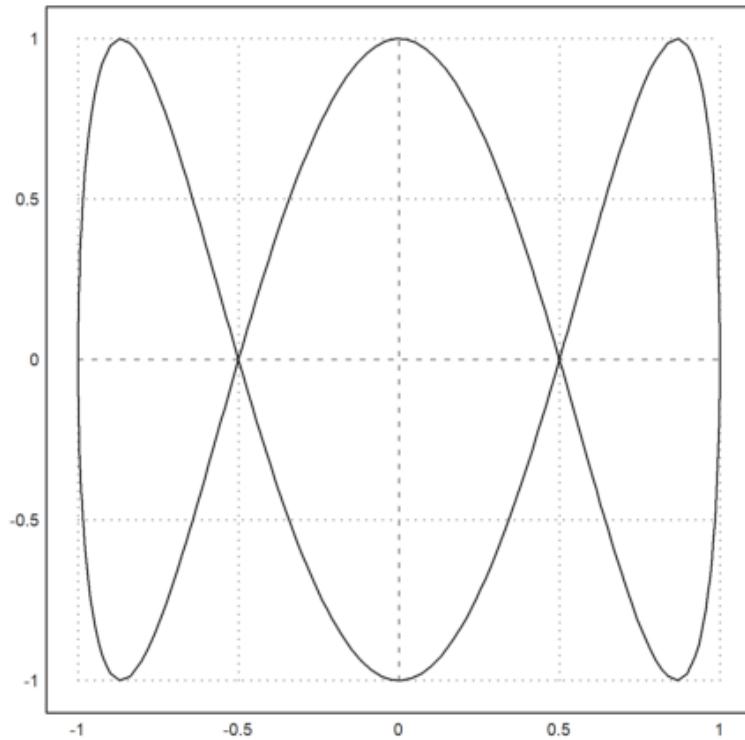
```
>  
>plot2d(["cos(x)", "sin(3*x)"], xmin=0, xmax=2pi):
```



- ini Merupakan gambar dua fungsi, yaitu  $\cos(x)$  dan  $\sin(3x)$ , dalam rentang 0 hingga  $2\pi$ .

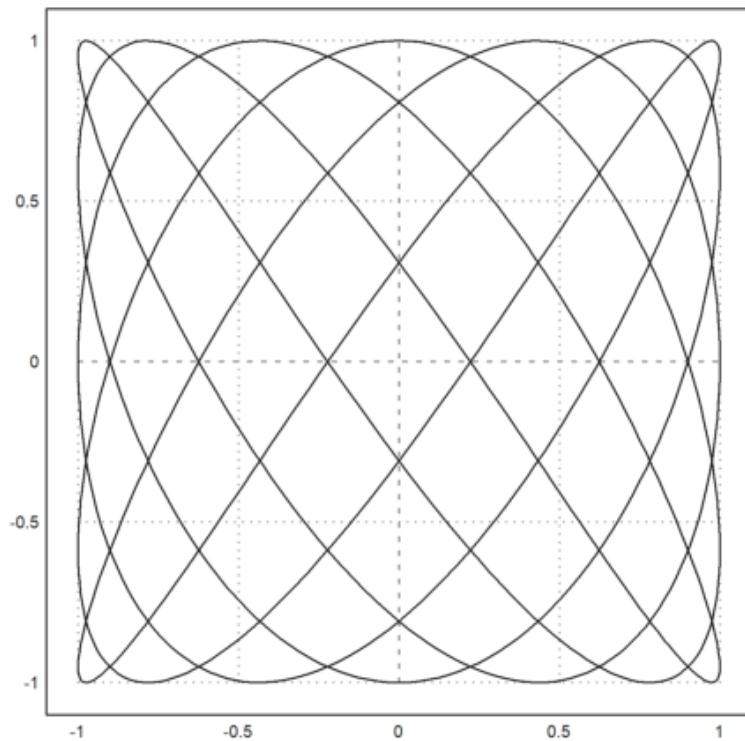
- Penjelasan mencakup konsep bahwa grafik berulang dalam rentang tertentu karena fungsi-fungsi ini memiliki frekuensi, periode, dan amplitudo yang berbeda.

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



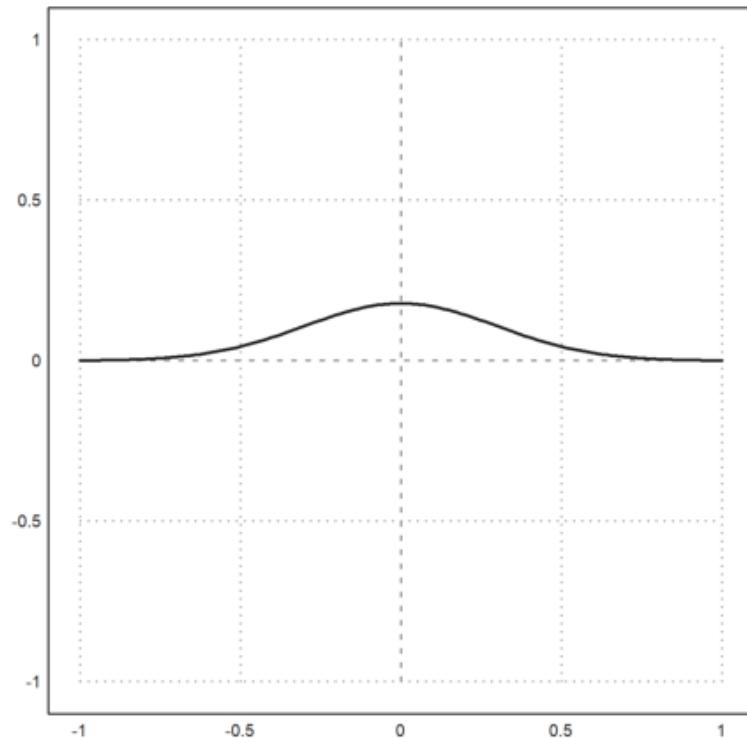
Sintaks diatas lebih menjelaskan bahaimana hubungan periodik grafik fungsi dari 2 fungsi yaitu  $\cos x$  dan  $\sin 3x$  dari rentang khusus dimana  $x$  min dari 0 sampai  $2\pi$ , hal tersebut dapat terjadi karena fungsi  $\cos x$  dan fungsi  $\sin 3x$  memiliki frekuensi, periode, dan amplitudo yang berbeda. grafik akan berulang pada rentang tertentu dan menghasilkan sebuah pola.

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



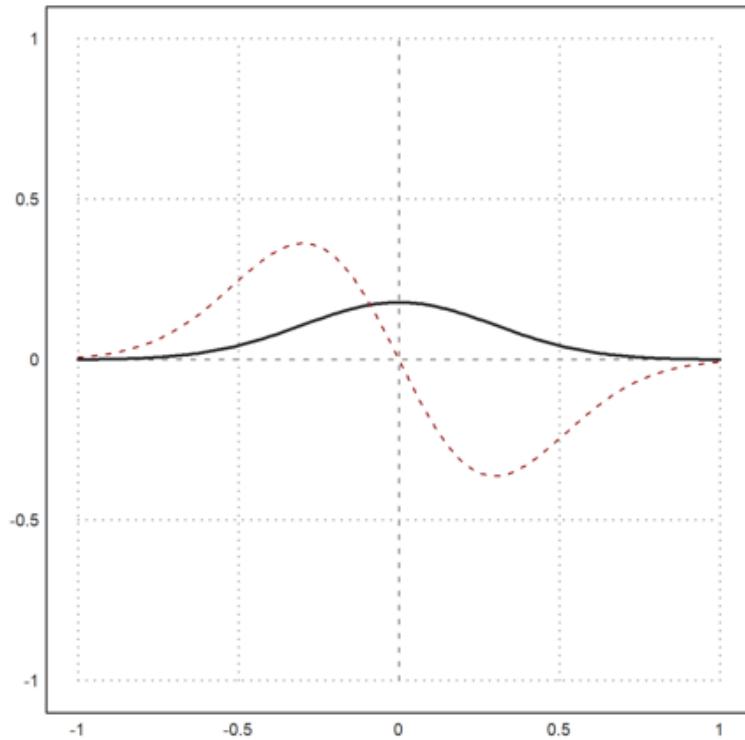
sintaks linspace digunakan untuk menghasilkan vektor x dari rentang yang telah ditentukan yaitu 0 sampai  $2\pi$  yang berisi 1000 nilai yang teratur

```
>a:=5.6; f &= exp(-a*x^2)/a;  
>plot2d(f, r=1, thickness=2):
```



- Fungsi  $f(x)$  yang merupakan hasil dari ekspresi  $\exp(-a*x^2)/a$ . Fungsi ini memiliki parameter  $a$  yang bergantung pada nilai yang diterapkan sebelumnya.
- Menggunakan perintah plot2d untuk menggambar grafik dari fungsi  $f(x)$ . Parameter  $r$  digunakan untuk mengatur rentang plot dan parameter thickness digunakan untuk mengatur ketebalan garis grafik.

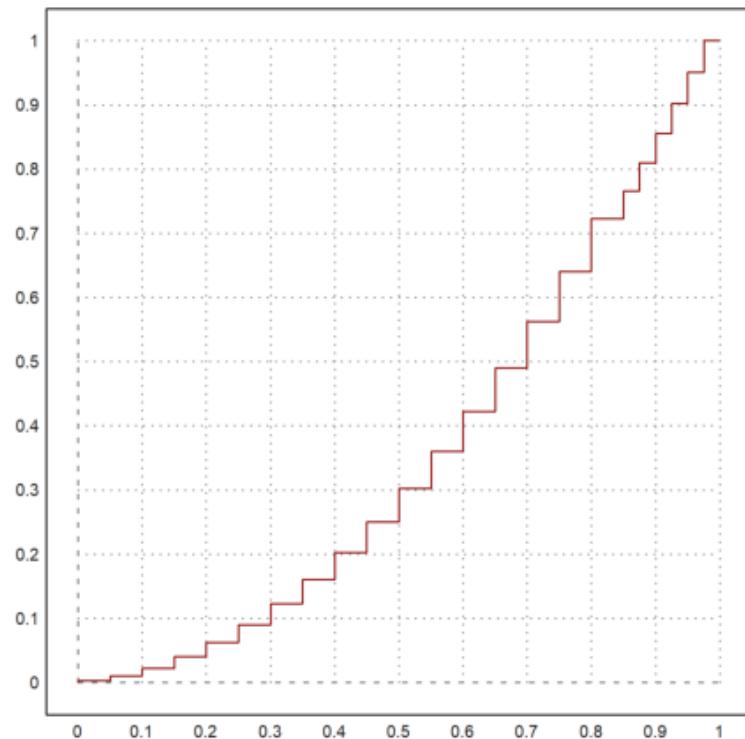
```
>plot2d(&diff(f, x), >add, style="--", color=red):
```



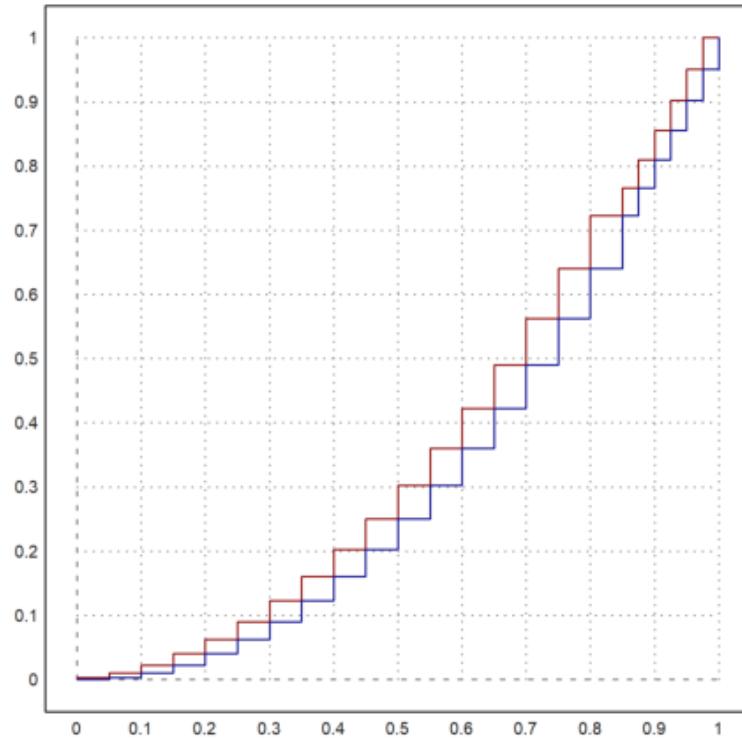
ini adalah grafik fungsi  $f$  dan grafik turunan pertama dari fungsi  $f$ . sintaks  $r=1$  digunakan untuk mengatur rentang yang akan ditampilkan pada plot,  $r=1$  berarti rentang dari  $-1$  sampai  $1$ .

sintaks  $>\text{add}$  digunakan untuk menambahkan grafik kedalam jendela grafik yang sudah ada sebelumnya.

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

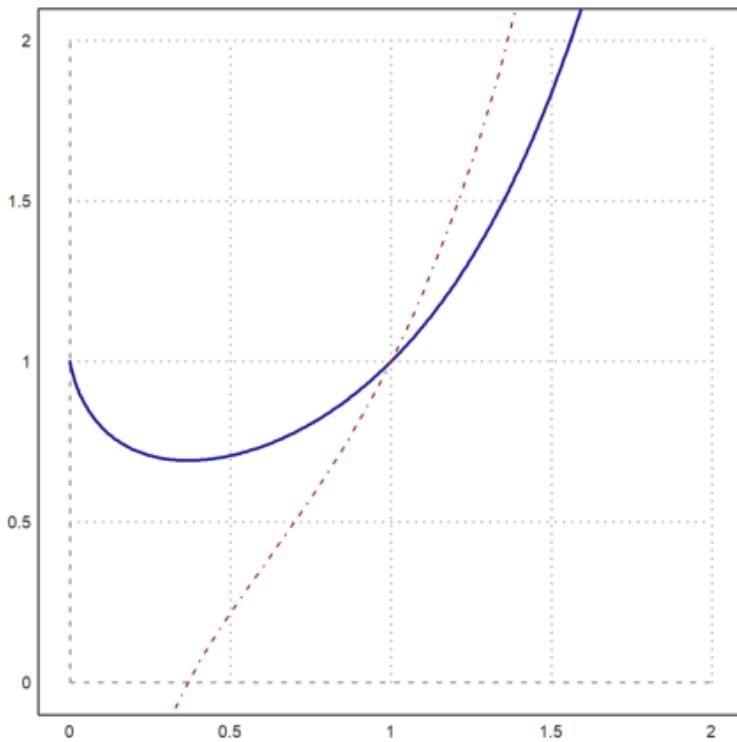


```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```



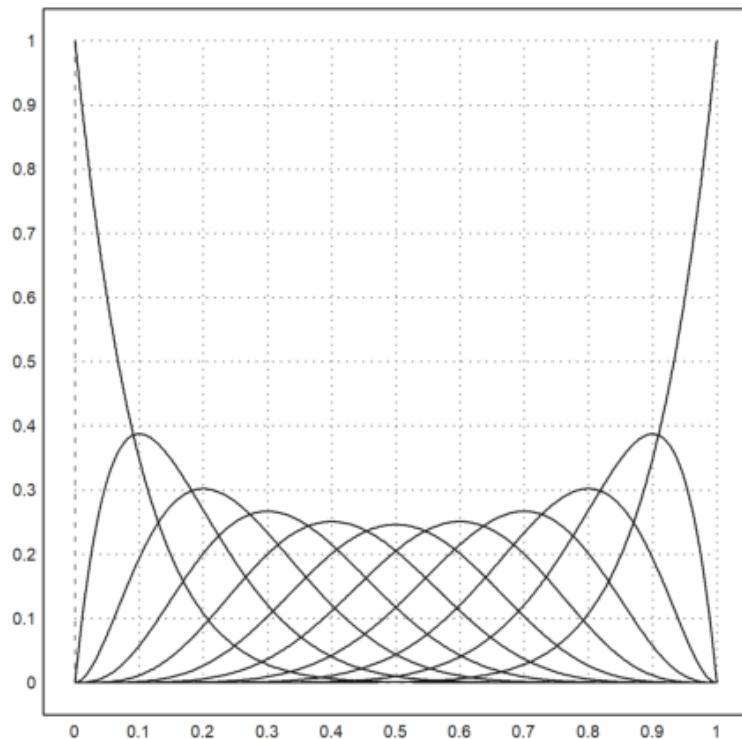
sintaks steps digunakan untuk mengatur jumlah langkah atau titik-titik yang digunakan dalam plot. dan sintaks n digunakan untuk mengatur jumlah step yang akan digunakan. semakin banyak n, maka bentuk grafik akan semakin mendekati aslinya.

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



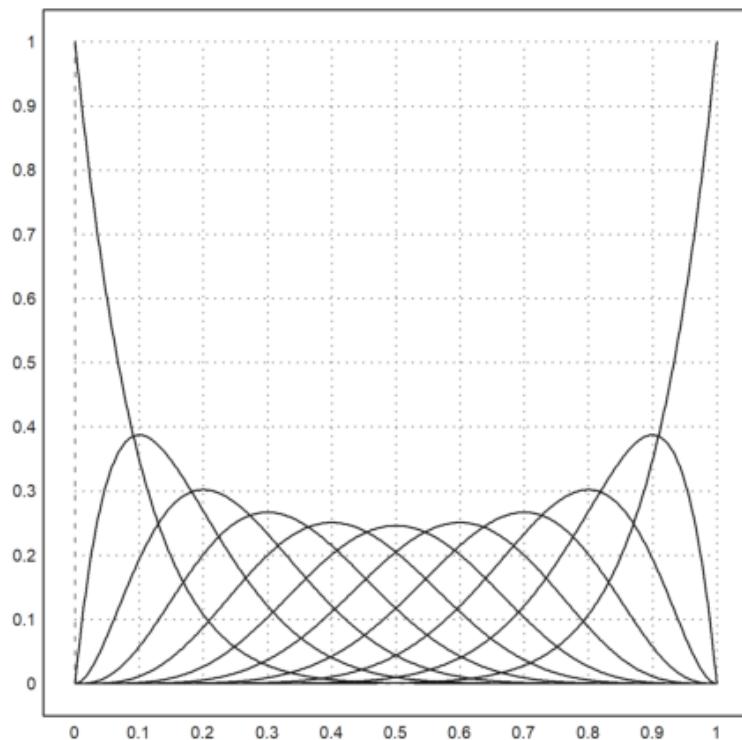
sintaks `cx=1, cy=1` digunakan untuk mengatur pusat tampilan grafik, maka plot akan diatur dengan titik pusat  $(1,1)$ .

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



dalam contoh kita menggambar serangkaian plot yang menggambarkan distribusi binomial dengan berbagai nilai  $i$  dari 1 hingga 10. kali ini kita menggunakan sintaks untuk melakukan looping pada fungsi yang berasosiasi dengan koefisien binomial dengan kombinasi 10 item. ini memungkinkan untuk memahami bagaimana distribusi binomial berubah dengan berbagai parameter.

```
>x=linspace(0,1,500);  
>n=10; k=(0:n)';  
>y=bin(n,k)*x^k*(1-x)^(n-k);  
>plot2d(x,y):
```

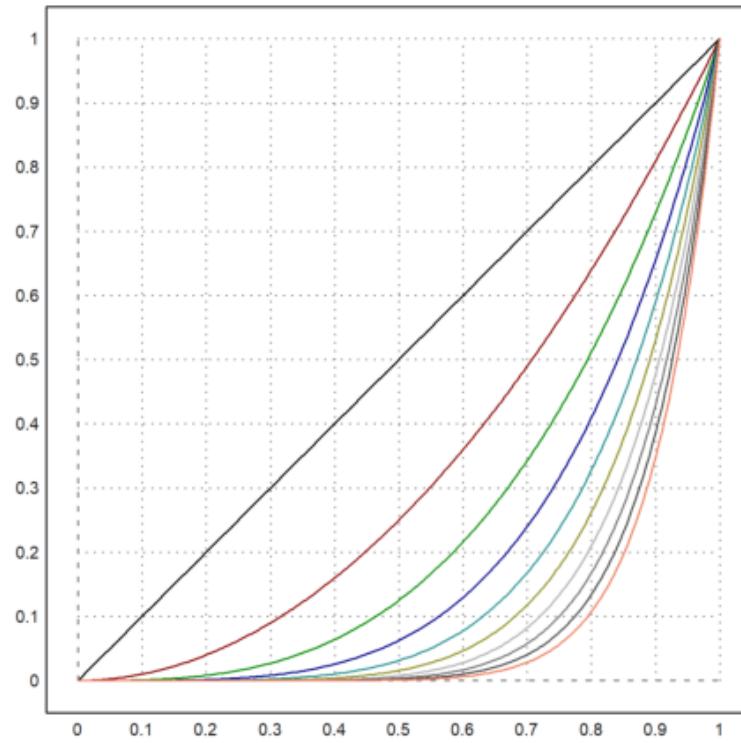


$n$  adalah vektor baris

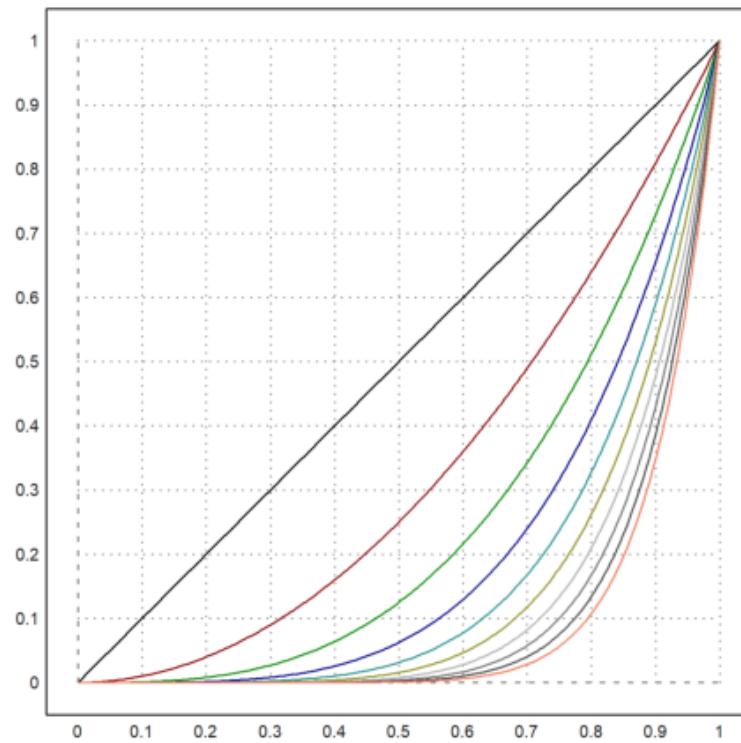
$k$  adalah vektor kolom

$y$  adalah matrik dari vektor baris dan vektor kolom tersebut dengan menggunakan fungsi binomial.

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

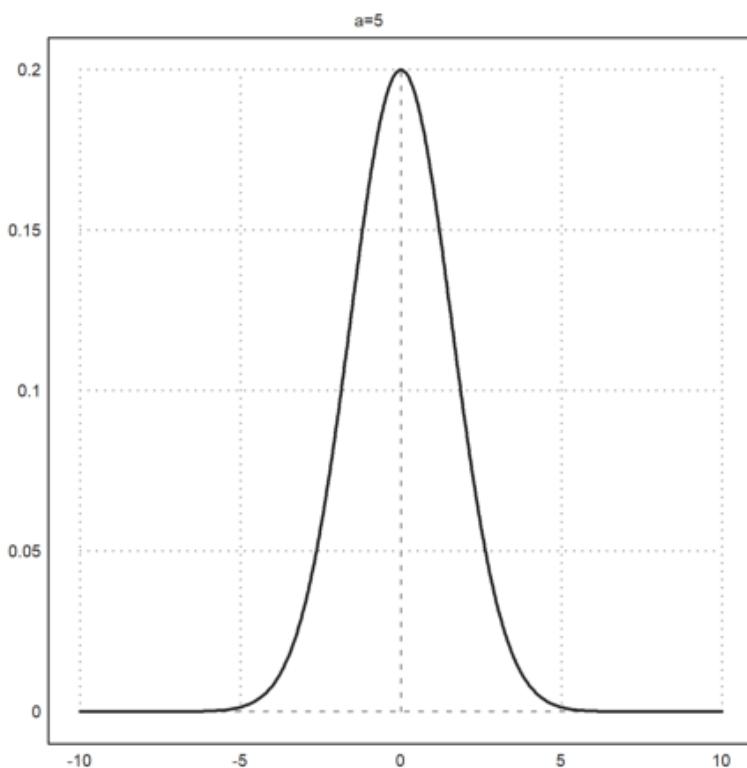


```
>n=(1:10)'; plot2d("x^n",0,1,color=1:10):
```



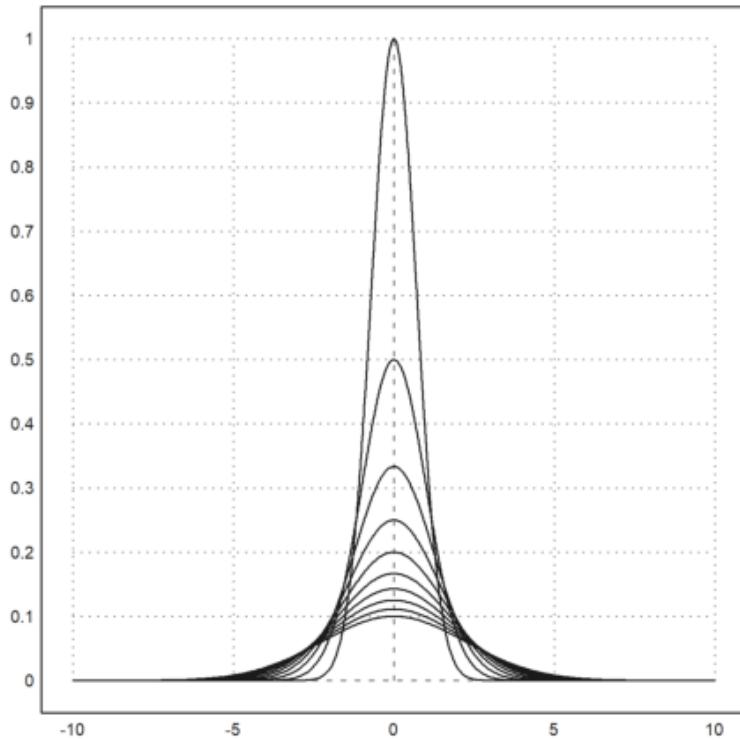
```

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



```

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



## Sub Bab 10

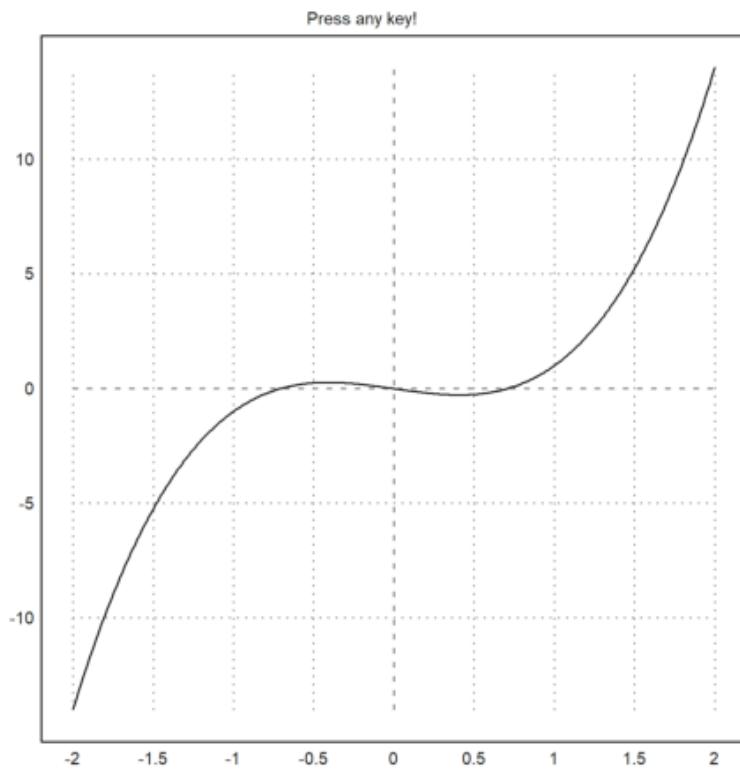
---

\* Membuat Gambar Kurva yang Bersifat Interaktif Kode ini, menggunakan ‘plot2d’ untuk membuat plot dari fungsi matematika ‘ $2*x^3-a*x$ ’ dengan parameter ‘a’. Flag ‘>user’ memungkinkan interaksi pengguna. Setelah plot ditampilkan, pengguna dapat melakukan beberapa tindakan interaktif.

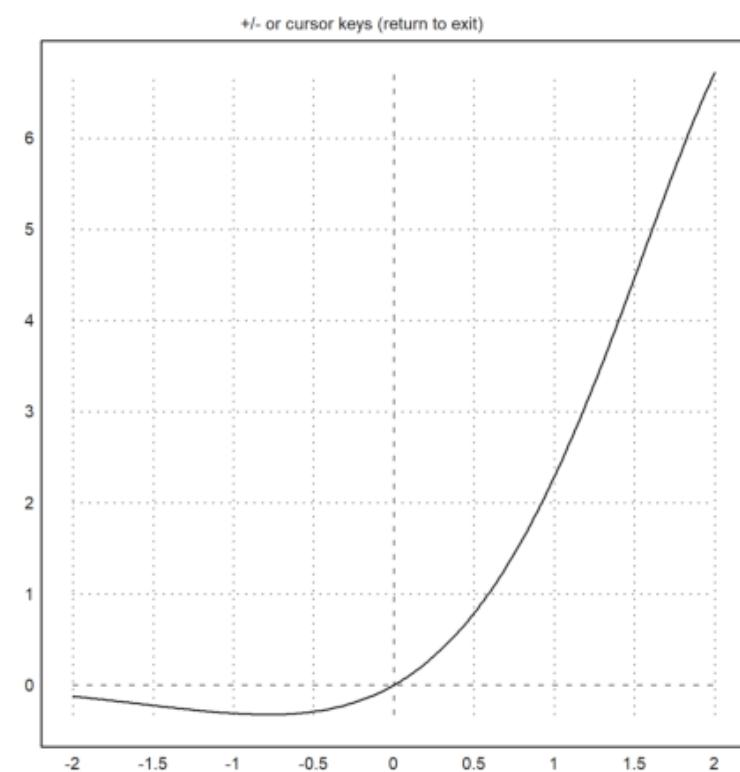
Saat plot ditampilkan dengan flag ‘>user’, pengguna dapat melakukan beberapa tindakan interaktif sebagai berikut:

- Perbesar dengan + atau -: Pengguna dapat memperbesar atau memperkecil plot dengan menggunakan tombol + atau - pada keyboard.
- Pindahkan Plot dengan Tombol Kursor: Pengguna dapat menggeser plot dengan menggunakan tombol kursor (panning).
- Pilih Jendela Plot dengan Mouse: Pengguna dapat memilih area tertentu dalam plot dengan menggunakan mouse.
- Atur Ulang Tampilan dengan Spasi: Jika pengguna menekan tombol spasi, maka tampilan plot akan diatur ulang ke jendela plot.
- Keluar dengan Kembali: Jika pengguna menekan tombol kembali, maka pengguna dapat keluar dari interaksi plot.

```
>plot2d({{"2*x^3-a*x"},a=1},>user,title="Press any key!"); ...
>insimg;
```



```
> plot2d("exp(x)*sin(x)",user=true, ...
> title="+/- or cursor keys (return to exit)":
```



Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut

Ini adalah pemanggilan fungsi plot2d yang digunakan untuk membuat plot dari fungsi matematika  $\exp(x) * \sin(x)$ . Parameter user=true menunjukkan bahwa ini adalah plot yang interaktif, yang berarti pengguna dapat berinteraksi dengan plot ini.

title="+/- or cursor keys (return to exit)": Ini adalah judul yang akan ditampilkan di atas plot. Pesan ini memberi petunjuk kepada pengguna tentang bagaimana mereka dapat berinteraksi dengan plot ini. Mereka dapat menggunakan tombol + atau - atau tombol kurSOR untuk berinteraksi dengan plot, dan tombol return (Enter) untuk keluar dari interaksi.

Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut:

- mousedrag(): Ini adalah fungsi bawaan yang digunakan untuk menunggu event mouse atau keyboard. Fungsi ini dapat mendeteksi kejadian seperti klik mouse, pergerakan mouse, atau penekanan tombol.
- dragpoints(): Fungsi ini memanfaatkan mousedrag() untuk memungkinkan pengguna menyeret titik-titik pada plot. Ini berarti pengguna dapat mengklik dan menarik titik-titik dalam plot sesuai dengan preferensi mereka.

Kita membutuhkan fungsi plot terlebih dahulu. Sebagai contoh, kita interpolasi dalam 5 titik dengan polinomial. Fungsi harus diplot ke area plot tetap.

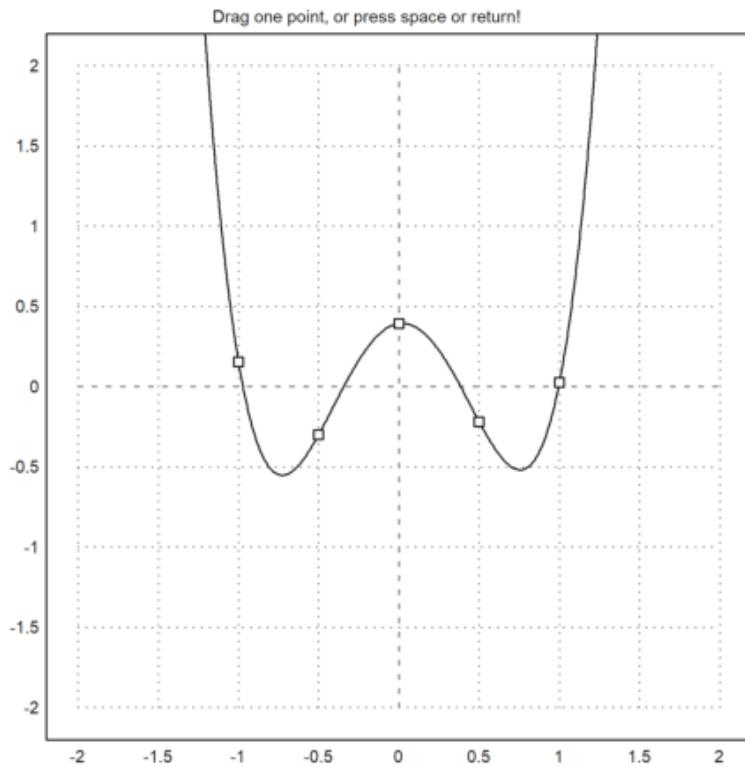
```
>function plotf(xp,yp,select) ...  
  
d=interp(xp,yp);  
plot2d("interval(xp,d,x)";d,xp,r=2);  
plot2d(xp,yp,>points,>add);  
if select>0 then  
    plot2d(xp[select],yp[select],color=red,>points,>add);  
endif;  
title("Drag one point, or press space or return!");  
endfunction
```

Perhatikan parameter titik koma di plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, mengakses nilai secara global.

Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret poin.

Kode berikut digunakan untuk menghasilkan beberapa nilai acak t dan membiarkan pengguna menyeret titik-titik pada plot dengan menggunakan fungsi dragpoints():

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



Ada juga fungsi, yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

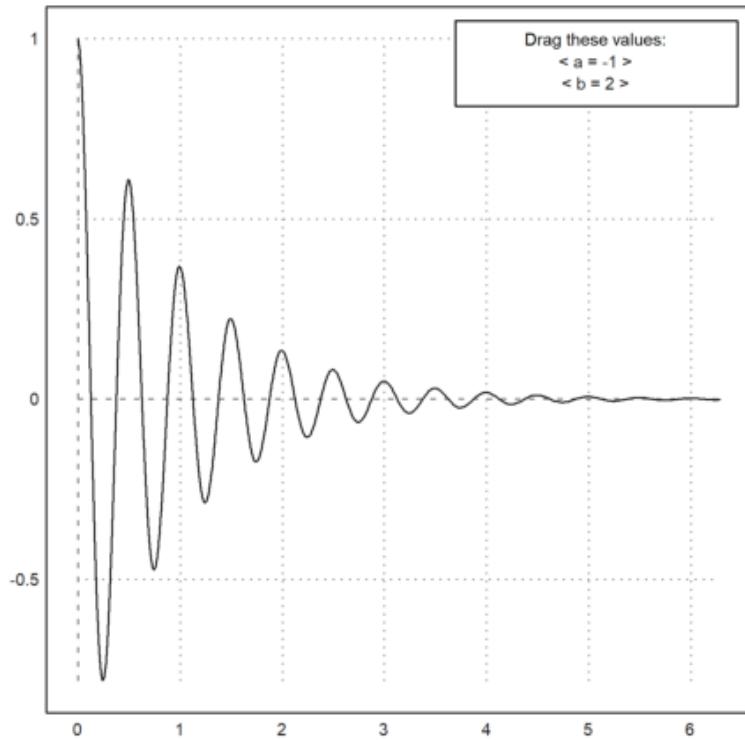
Pertama kita membutuhkan fungsi plot.

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

Kemudian kita membutuhkan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional baris judul.

Ada slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

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

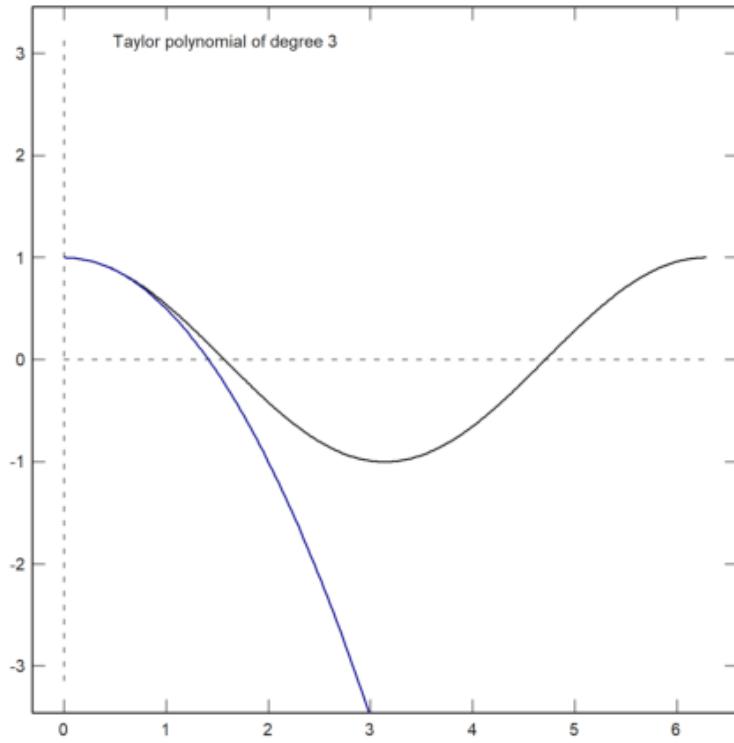


Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor derajat n ke fungsi kosinus.

```
>function plotf(n) ...
    plot2d("cos(x)",0,2pi,>square,grid=6);
    plot2d(&"taylor(cos(x),x,0,@n)",color=blue,>add);
    textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",>left);
endfunction
```

Sekarang kami mengizinkan derajat n bervariasi dari 0 hingga 20 dalam 20 pemberhentian. Hasil dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk memasukkan plot ke dalam buku catatan.

```
>nd=dragvalues("plotf","degree",3,[0,10],10,y=0.8, ...
>   heading="Drag the value:"); ...
>plotf(nd);
```



Berikut ini adalah demonstrasi sederhana dari fungsi tersebut. Pengguna dapat menggambar di atas jendela plot, meninggalkan jejak poin.

```
>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
```

## **Sub Bab 11 \* Menggambar Kurva Fungsi Parametrik Kita telah terbiasa**

---

dengan kurva yang didefinisikan oleh sebuah persamaan yang menghubungkan koordinat x dan y Contohnya

$$y = x^2$$

Atau

$$x^2 + y^2 = 13$$

dimana persamaan-persamaan ini tidak dikaitkan dengan panjang kurva s , waktu t, dan besaran lainnya. Besaran besaran ini disebut parameter

persamaan parametrik adalah persamaan yang menyatakan hubungan variabel x, y dituliskan dengan

$$x = f(t)$$

$$y = g(t)$$

dengan  $a \leq t \leq b$  tiap nilai  $t$  menentukan titik  $(x, y)$  pada kurva. Jadi, dengan berubahnya nilai  $t$ , titik

$$(x, y) = (f(t), g(t))$$

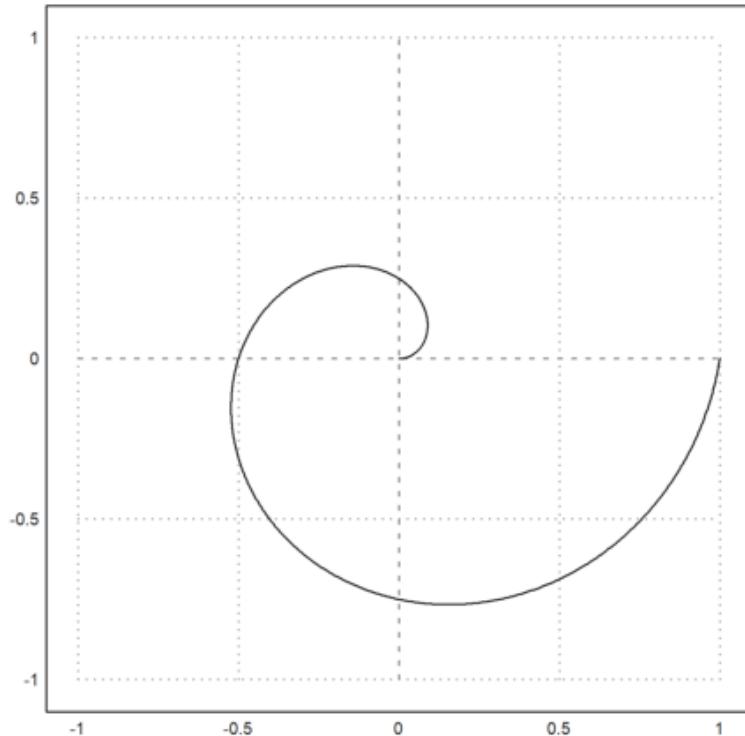
bergerak sepanjang kurva yang disebut kurva parametrik

Dalam contoh berikut, kita memplot spiral

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

Kita perlu menggunakan banyak titik untuk tampilan yang halus

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

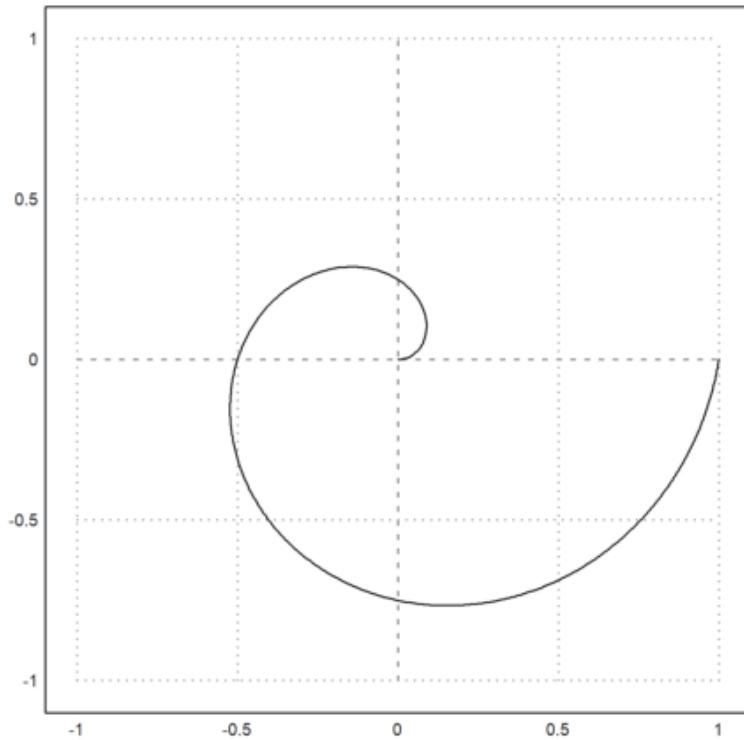


$r$  digunakan untuk mengatur radius marker titik-titik yang akan

digunakan dalam plot.

Sebagai alternatif, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

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

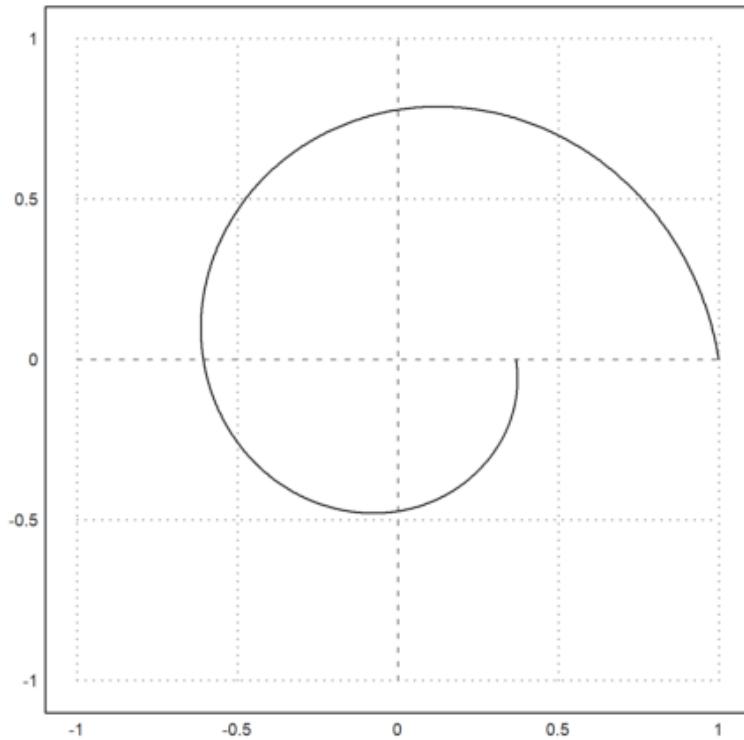


Perintah linspace digunakan untuk membuat array nilai yang terdistribusi secara merata antara dua angka tertentu. Fungsi ini sangat berguna untuk menentukan rentang nilai yang ingin digunakan pada sumbu x atau y ketika membuat plot.

```
0 : Nilai awal dari rentang.  
1 : Nilai akhir dari rentang.
```

Perintah linspace akan menghasilkan array dengan n elemen yang terdistribusi merata antara start dan stop.

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



$\exp(-t)$  menghasilkan nilai yang semakin mendekati nol seiring dengan pertambahan nilai  $t$ , karena eksponensial dari nilai negatif semakin mendekati nol saat nilai  $t$  semakin besar.

Jadi,  $r = \exp(-t)$  memberikan suatu fungsi yang menurun dengan nilai  $t$ . Dalam konteks program ini,  $r$  digunakan untuk mengontrol jari-jari dari kurva spiral dalam plot 2D. Jari-jari ini semakin kecil seiring dengan pertambahan nilai  $t$ , menciptakan efek spiral yang semakin rapat ke pusat pada bagian ujung kurva.

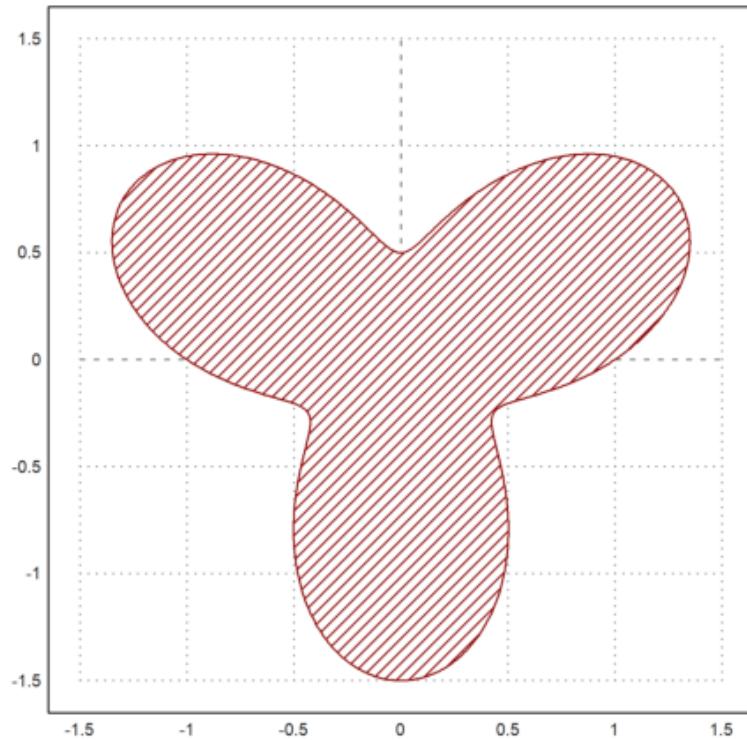
Pada contoh berikutnya, kita memplot kurvanya

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

dengan

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

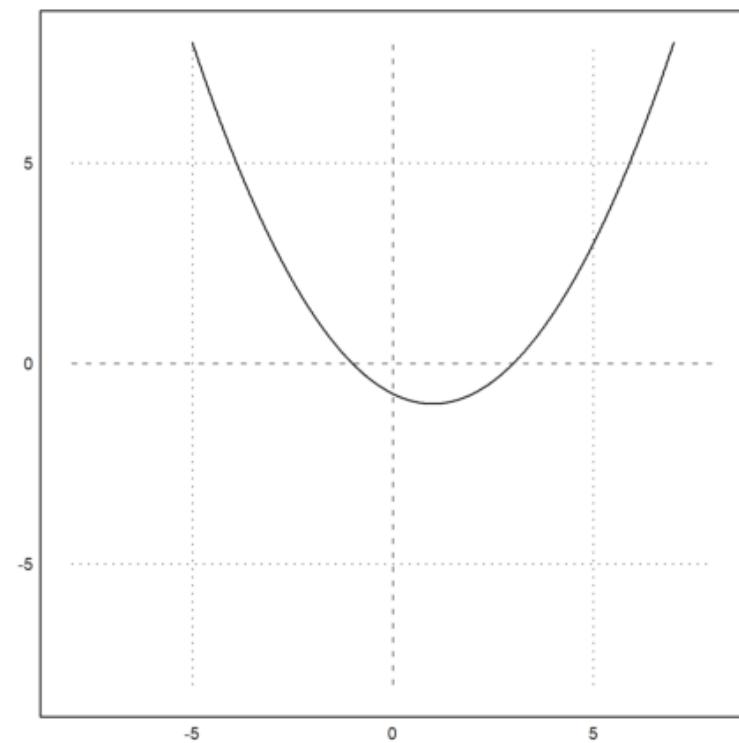
```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
>plot2d(x,y,>filled,fillcolor=red,style="/" ,r=1.5):
```



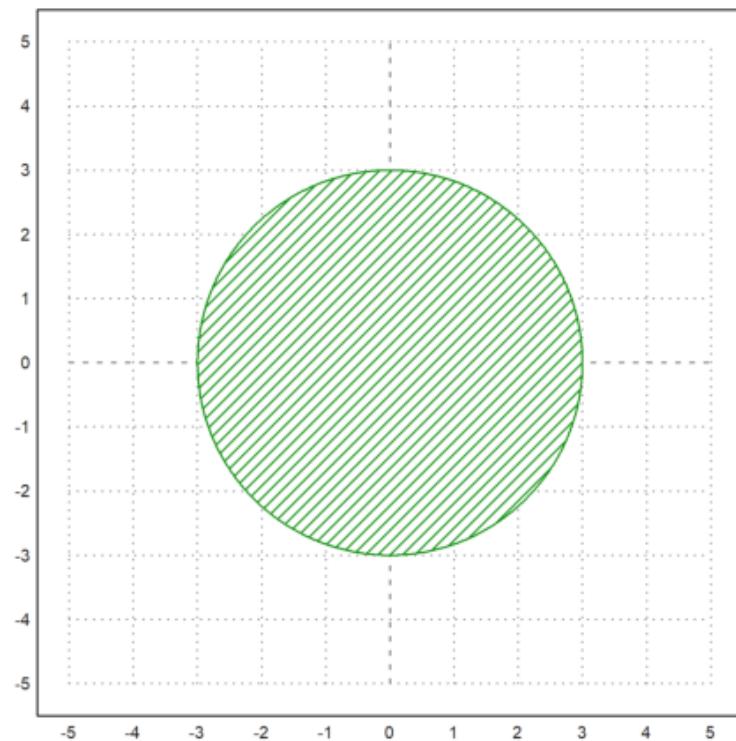
## Contoh lain

---

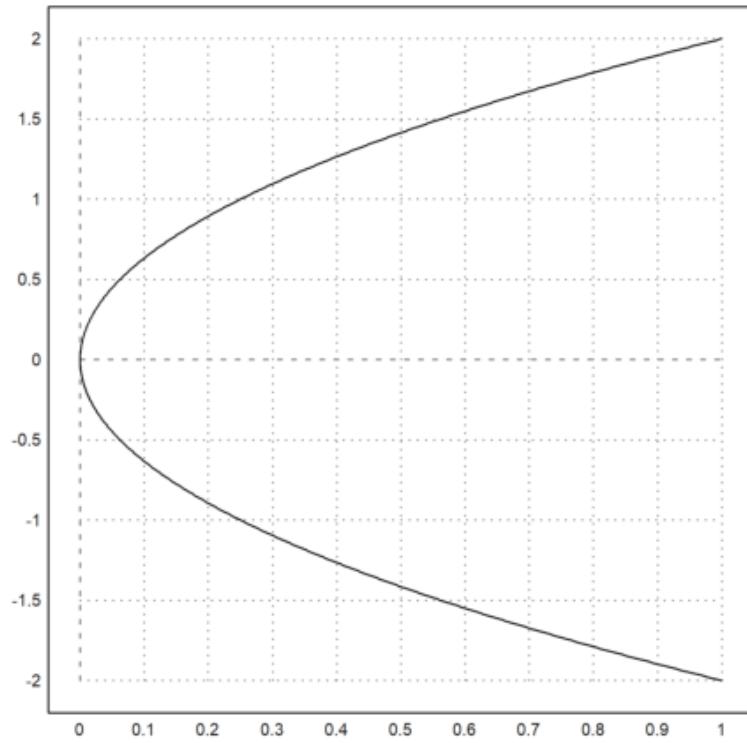
```
>t=linspace(-3,3,1000); x=2*t+1; y=t^2-1;  
>plot2d(x,y,r=8):
```



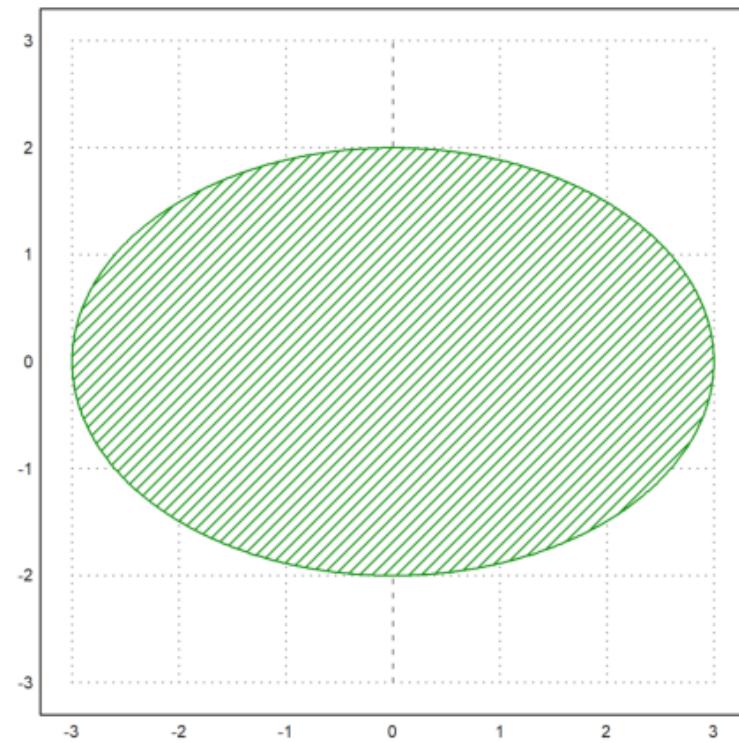
```
>t=linspace(0,2pi,1000); r=3; x=r*cos(t); y=r*sin(t);...
>plot2d(x,y,>filled,fillcolor=green,style="/",r=5):
```



```
>t=linspace(-1,1,1000); x=t^2; y=2*t;...
>plot2d(x,y):
```



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



## **Sub Bab 12 \* Menggambar Kurva Fungsi Implisit**

Fungsi implisit adalah

fungsi yang memuat lebih dari satu variabel, berjenis variabel bebas dan variabel terikat yang berada dalam satu ruas sehingga tidak bisa dipisahkan pada ruas yang berbeda.

Untuk fungsi implisit, harus berupa fungsi atau ekspresi dari parameter x dan y.

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

Untuk menggambar himpunan  $f(x,y)=c$  untuk satu atau lebih konstanta c, dapat menggunakan plot2d().

Fungsi implisit juga dapat diisi dengan persamaan tingkat

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

Untuk fungsi ini harus berupa matriks 2xn dimana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval.

Plot implisit menunjukkan garis level yang menyelesaikan  $f(x,y)=\text{level}$ , di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika level="auto", akan ada garis level nc, yang akan menyebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan >hue untuk menunjukkan nilai fungsi. Untuk fungsi implisit, xv harus berupa fungsi atau ekspresi dari parameter x dan y, atau, sebagai alternatif, xv dapat berupa matriks nilai.

Euler dapat menandai garis level

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

dari fungsi apapun.

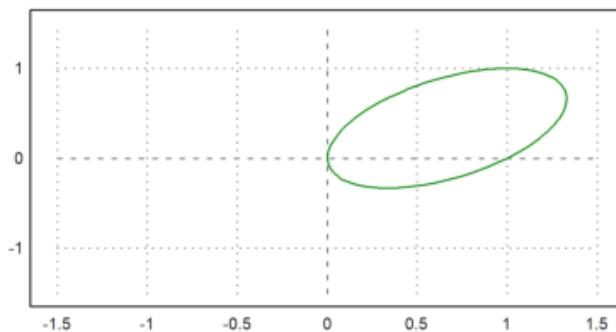
Untuk menggambar himpunan  $f(x,y)=c$  untuk satu atau lebih konstanta c, Anda dapat menggunakan plot2d() dengan plot implisitnya di dalam bidang. Parameter untuk c adalah level=c, di mana c dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi untuk setiap titik dalam plot. Parameter "n" menentukan kehalusan plot.

## Contoh Soal

---

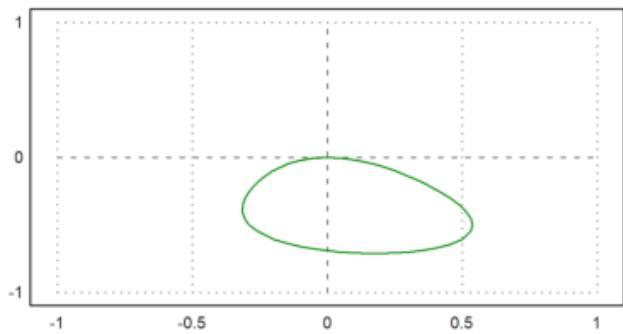
$$x^2 + y^2 - xy - x = 0$$

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

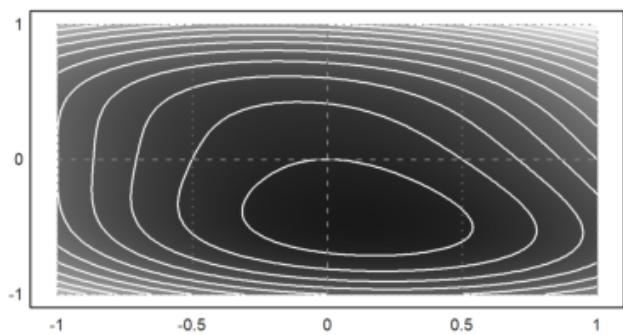


$$2x^2 + xy + 3y^4 + y = 0$$

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr, level=0, contourcolor=green): // Solutions of f(x,y)=0
```

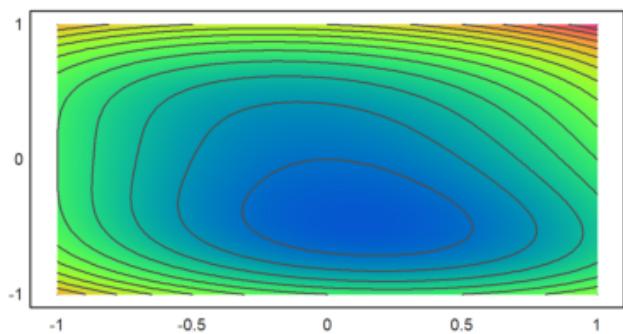


```
>plot2d(expr,level=0:0.5:20,>hue,contourcolor=white,n=200): // nice
```



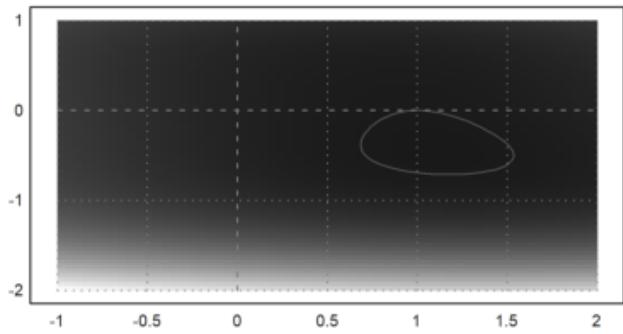
Parameter `>hue` digunakan untuk memberikan warna pada kontur sesuai dengan levelnya. Kontur dengan level yang lebih tinggi akan memiliki warna yang berbeda.

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

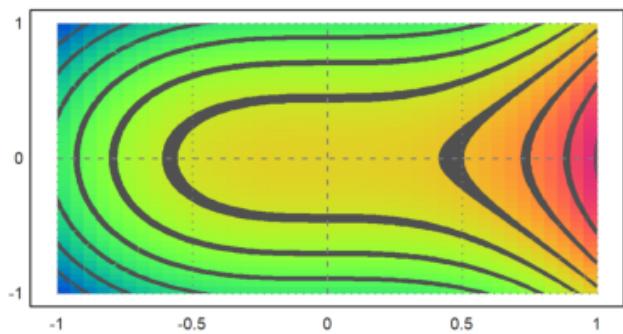


`>spectral` digunakan untuk mengatur palet warna yang akan digunakan pada kontur. Dalam hal ini, digunakan palet warna "spectral".

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



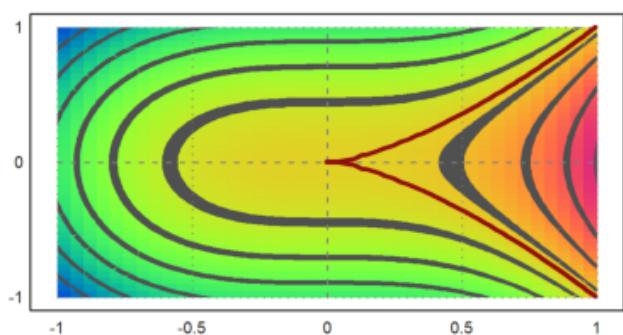
```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



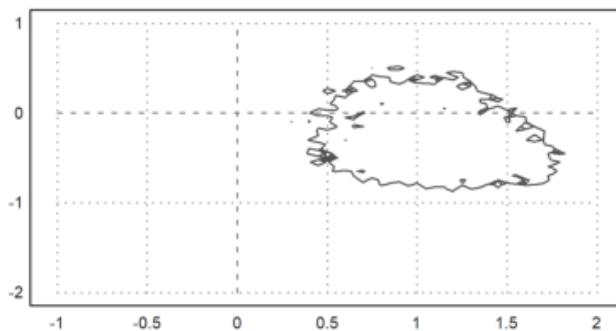
Perintah `>contour` adalah cara untuk menghasilkan plot kontur, yaitu plot yang menunjukkan garis-garis kontur yang mewakili tingkat-tingkat dari suatu fungsi. Jumlah dan posisi garis kontur akan secara otomatis diatur oleh Euler Math Toolbox berdasarkan distribusi nilai-nilai fungsi.

Penggunaan level memungkinkan Anda secara eksplisit menentukan tingkat kontur yang ingin kita tampilkan pada plot. kita dapat mengatur level kontur sesuai dengan preferensi kita, dan plot akan menampilkan garis kontur pada tingkat-tingkat yang kita tentukan.

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

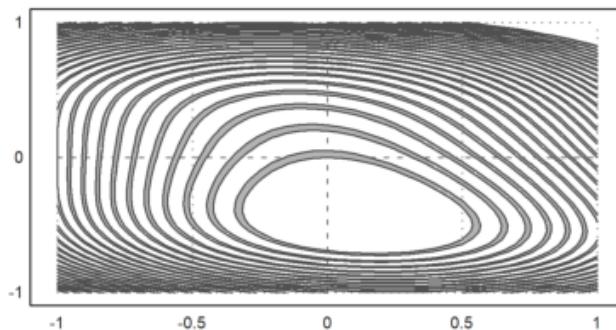


```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```

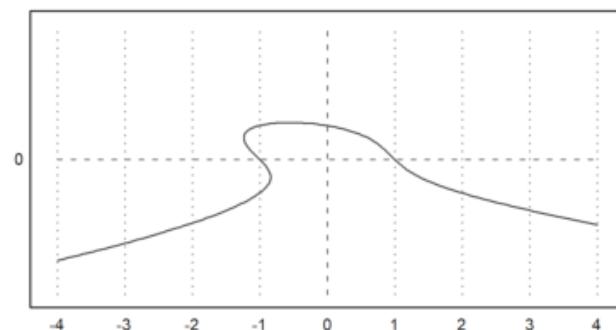


`normal(size(z))` menghasilkan matriks dengan ukuran yang sama dengan matriks `z`, dan setiap elemennya diambil dari distribusi normal standar (mean 0, deviasi standar 1). Kemudian, matriks `z` diubah dengan menambahkan nilai-nilai acak ini, yang telah dikalikan dengan 0.2. Ini menciptakan variasi acak dalam matriks `z`.

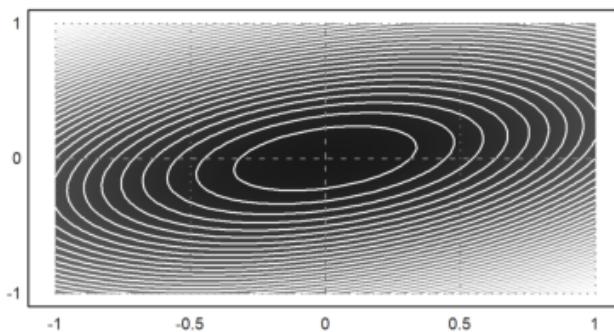
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



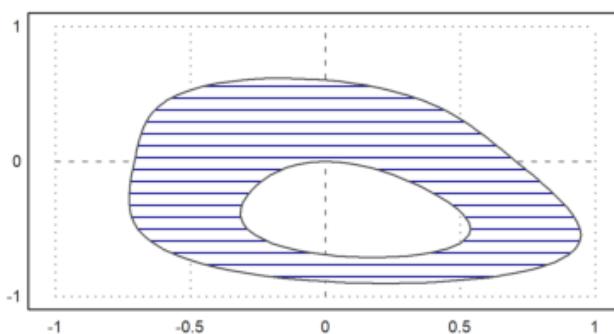
```
>plot2d("x^2+y^3+x*y",level=1,r=4,n=100):
```



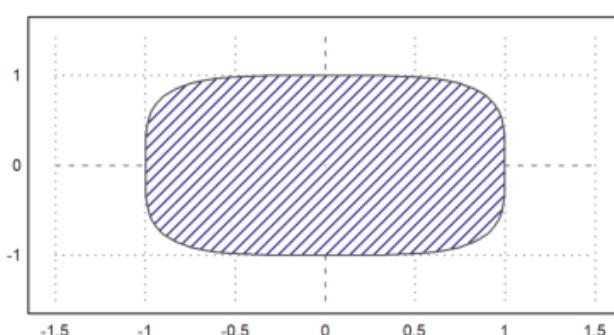
```
>plot2d("x^2+2*y^2-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):
```



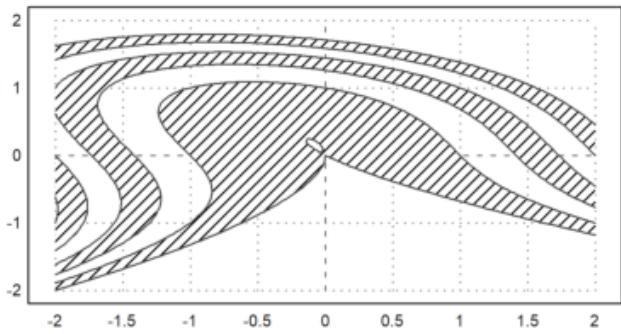
```
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



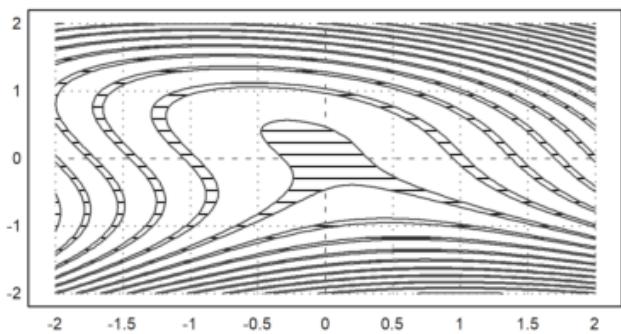
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



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

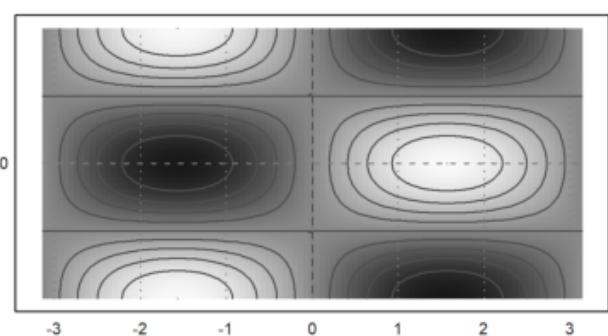


```
>plot2d("x^2+y^3+x*y",level=-10:20,r=2,style="-",dl=0.1,n=100):
```

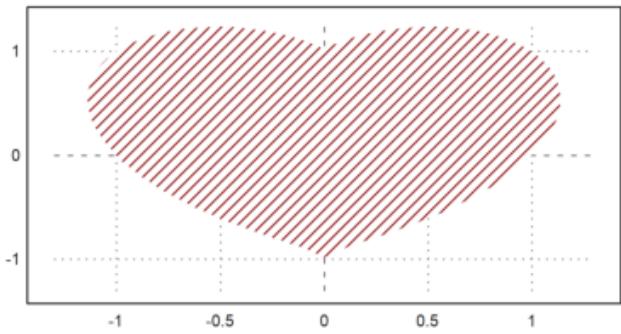


dl Parameter ini mengatur tingkat penghalusan pada plot. Semakin kecil nilai ini, semakin halus plotnya.

```
>plot2d("sin(x)*cos(y)",r=pi,>hue,>levels,n=100):
```



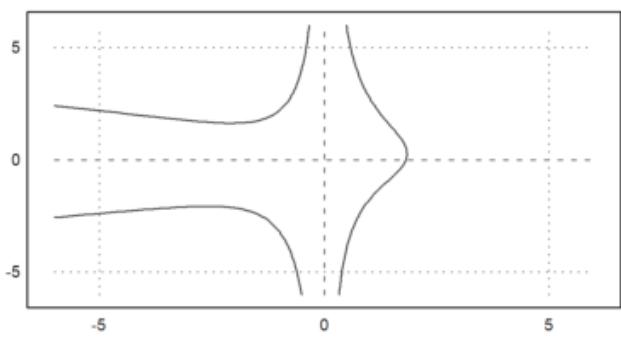
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>style="/",color=red,<outline, ...
>level=[-2;0],n=100):
```



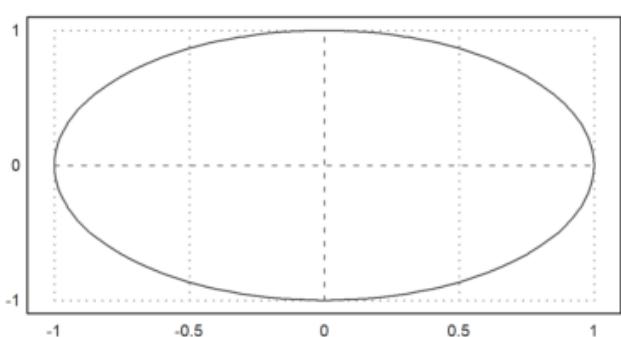
<outline: Parameter ini mengatur plot agar hanya memiliki kontur saja tanpa diisi.  
Misal plot solusi dari persamaan

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

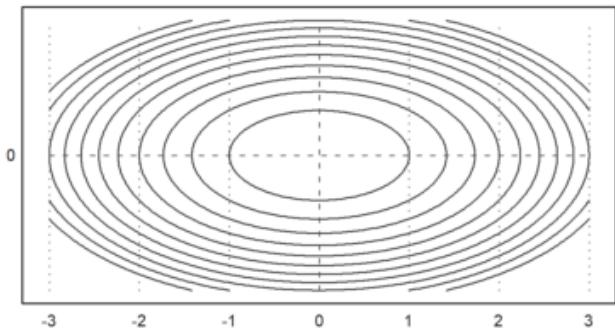
```
>plot2d("x^3-x*y+x^2*y^2", r=6, level=6, n=100) :
```



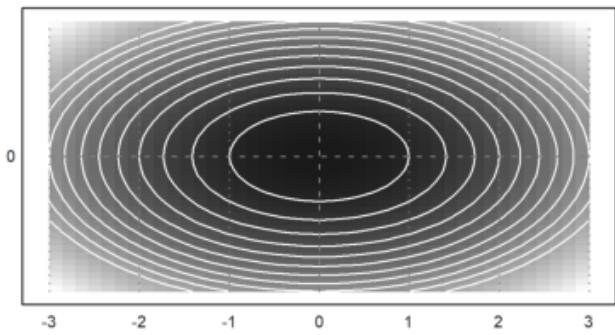
```
>plot2d("x^2+y^2-1", level=0) :
```



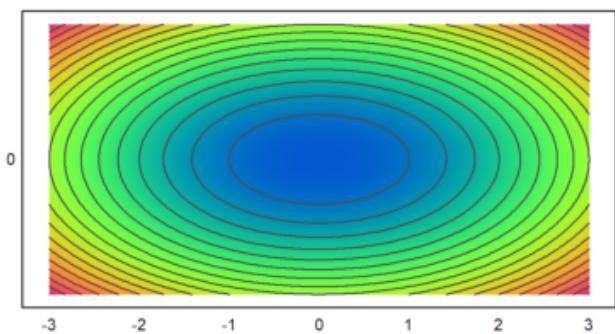
```
>plot2d("x^2+y^2-1", r=3, level=0:1:10, n=200) :
```



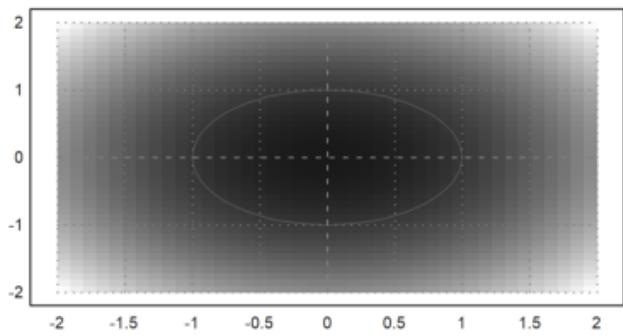
```
>plot2d("x^2+y^2-1",r=3,level=0:1:10,>hue,contourcolor=white):
```



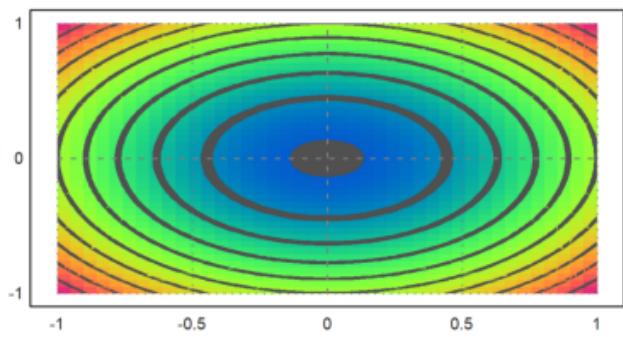
```
>plot2d("x^2+y^2-1",r=3,level=0:1:20,>hue,>spectral,n=200,grid=4):
```



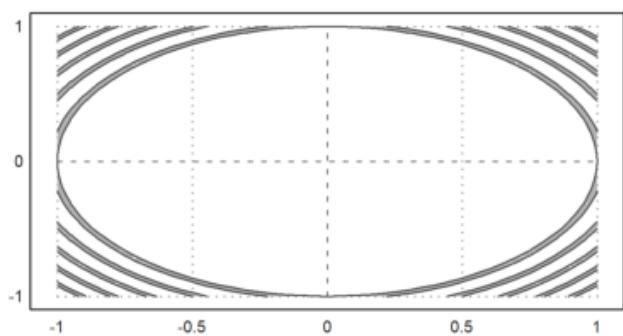
```
>plot2d("x^2+y^2-1",level=0,a=-2,b=2,c=-2,d=2,>hue):
```



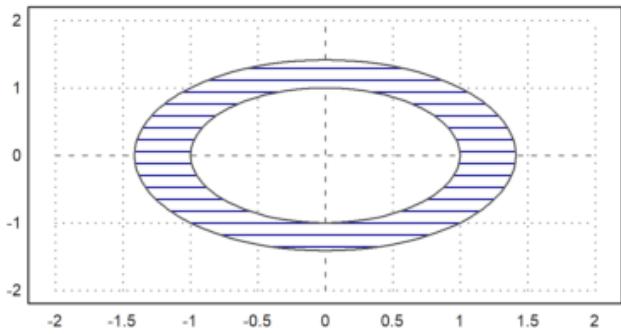
```
>plot2d("x^2+y^2-1",>contour,>hue,>spectral):
```



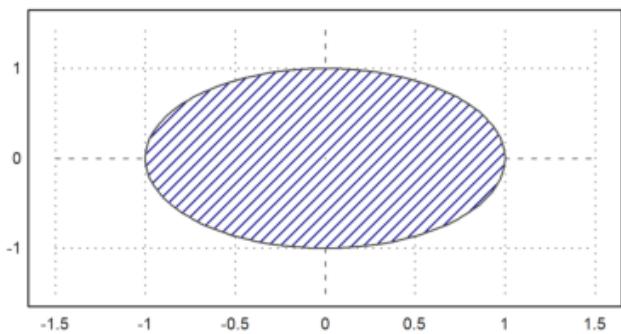
```
>plot2d("x^2+y^2-1",level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



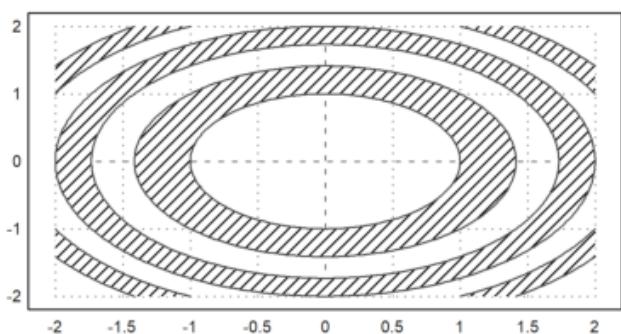
```
>plot2d("x^2+y^2-1",r=2,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



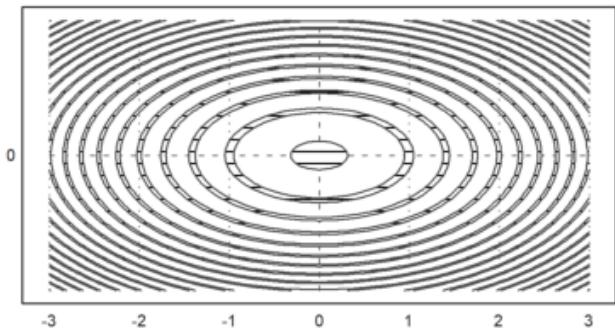
```
>plot2d("x^2+y^2",r=1.5,level=[0;1],color=blue,style="/"):
```



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



```
>plot2d("x^2+y^2-1",level=-10:20,r=3,style="-",dl=0.1,n=100):
```



## Sub bab 13

---

\* Menggambar Grafik Bilangan Kompleks Bilangan kompleks secara visual dapat direpresentasikan sebagai sepasang angka  $(a, b)$  membentuk vektor pada diagram yang disebut diagram Argand, mewakili yang bidang kompleks. Sumbu-x adalah sumbu nyata dan sumbu-y adalah sumbu imajiner.

Menggambar kurva fungsi kompleks sendiri adalah proses visualisasi grafis dari fungsi matematika kompleks (yaitu fungsi yang melibatkan bilangan kompleks, yaitu bilangan dengan bagian real dan imajiner) berperilaku dalam koordinat kompleks. Hal tersebut memungkinkan untuk melihat bagaimana pola, bentuk, dan sifat dari fungsi kompleks tersebut.

Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi  $1 \times 2$ ) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.

Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.

> Definisi fungsi kompleks, mendefinisikan fungsi kompleks yang dianalisis atau digambarkan. Fungsi ini memiliki variabel kompleks  $z$ , yang melibatkan bagian real dan imajiner.

> Selanjutnya kita dapat menggunakan fungsi linspace. Fungsi linspace sendiri adalah salah satu fungsi yang umum digunakan dalam pemrograman, terutama dalam konteks pemrograman numerik dan ilmu data. Ini sering digunakan untuk menghasilkan urutan nilai dalam rentang tertentu dengan jumlah titik yang sama di antara dua titik ujungnya. Penggunaannya tidak terbatas pada pemrosesan sinyal atau elektromagnetik, tetapi bisa digunakan dalam berbagai konteks di mana Anda perlu membuat urutan nilai.

> Penentuan rentang, memilih rentang nilai  $z$  yang ingin ditampilkan di dalam plot. Rentang ini mencakup wilayah kompleks tertentu yang ingin diamati.

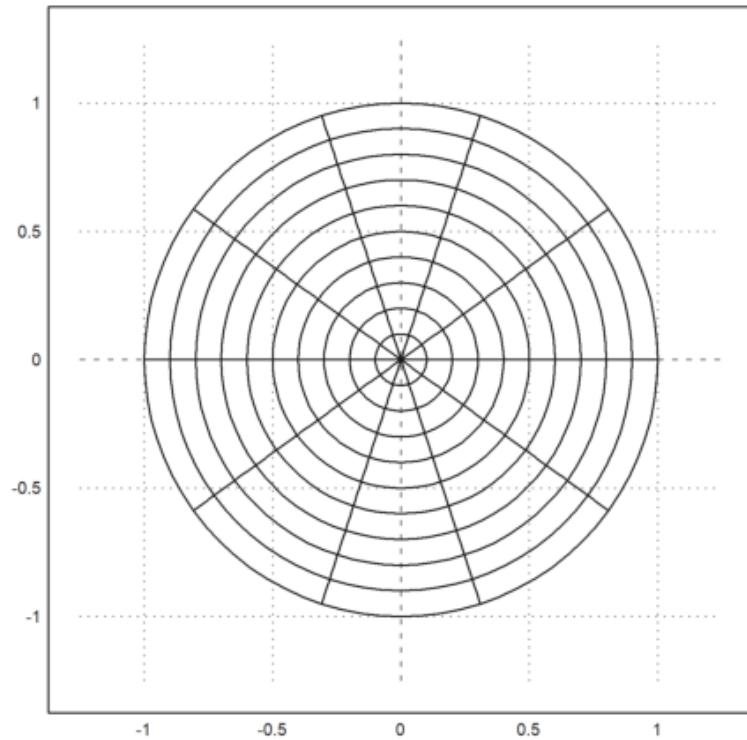
> Menggunakan sintaks plot2d.

> Penyesuaian plot, mengubah plot sesuai yang diinginkan (mengubah warna, format (style), dan sebagainya). Dalam contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

## Contoh

---

```
>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):
```



#### Penjelasan sintaks

`z` : sebuah ekspresi atau fungsi yang akan digambar dalam koordinat kompleks.

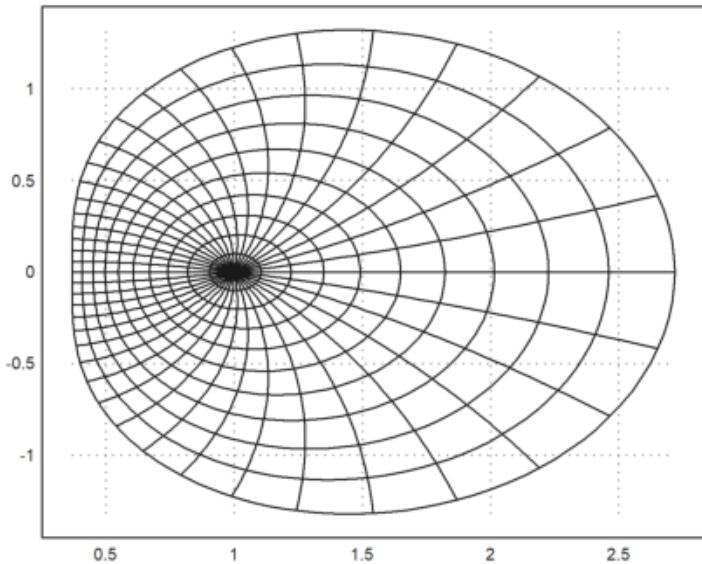
`a,b,c,d` : parameter-parameter yang digunakan untuk mengatur jendela tampilan (viewport) dalam koordinat kompleks. Parameter-parameter ini akan menentukan rentang sumbu x dan sumbu y yang akan ditampilkan di dalam plot.

`cgrid` : parameter ini mengontrol tampilan grid pada plot. Jika `cgrid=n`, maka grid akan ditampilkan, jika `cgrid=0`, maka grid akan disembunyikan.

## Bentuk lain

---

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



Penjelasan :

Perintah tersebut merupakan perintah untuk menggambar kurva dari fungsi kompleks eksponensial "exp(z)" dalam koordinat kompleks. Dalam perintah tersebut juga menggunakan parameter cgrid dengan nilai [40,10] untuk mengatur grid pada plot.

Dalam sintaks ini,

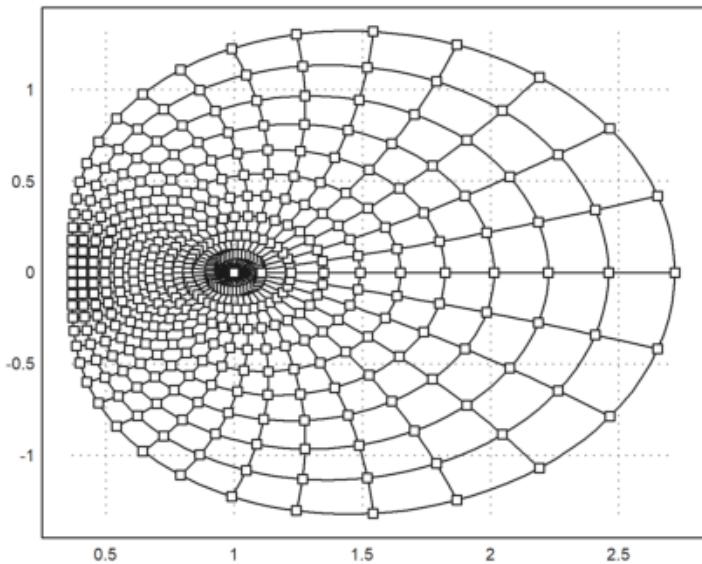
`exp(z)` : fungsi eksponensial kompleks yang akan digambar

`cgrid=[40,10]` : mengatur grid pada plot. cgrid tersebut adalah jumlah garis grid yang akan digunakan pada sumbu x dan sumbu y. Nah di dalam plot ini, akan ada 40 garis grid pada sumbu x dan 10 grid pada sumbu y.

## Bentuk lain

---

```
>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 pada bidang kompleks dengan bagian real dan bagian imajiner.

Penjelasan :

Perintah plot2d di atas adalah perintah untuk menggambar kurva fungsi kompleks dalam koordinat kompleks, namun dengan opsi yang berbeda,

`exp(z)`: fungsi kompleks yang akan digambar

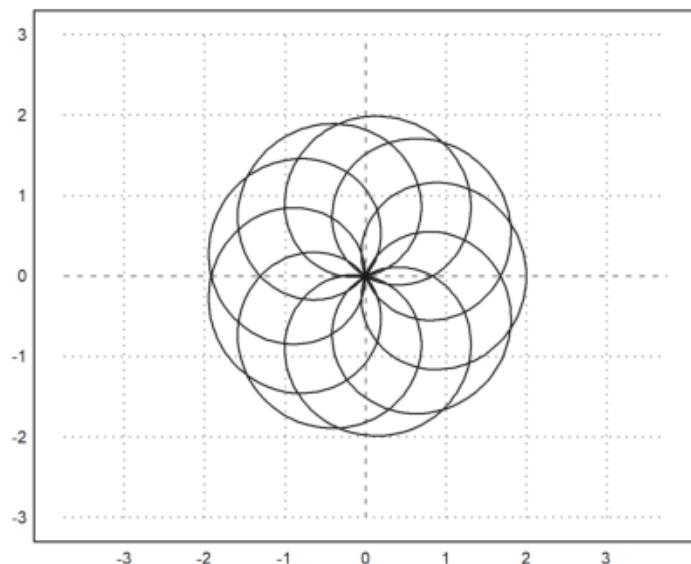
`>points` : opsi ini mengubah cara plot untuk dilakukan. Dengan menggunakan `>points`, plot ini akan menggunakan titik-titik diskrit untuk merepresentasikan fungsi ke dalam bentuk titik-titik

`>add` : sintaks ini menginstrusikan perintah untuk menambahkan plot ini ke plot sebelumnya jika ada.

## Contoh

---

```
>t=linspace(0,2pi,1000); ...
>plot2d(exp(I*t)+exp(10*I*t), r=3):
```



Penjelasan :

Perintah plot2d di atas menggambarkan kurva dari fungsi kompleks yang diberikan dalam koordinat kompleks dengan parameter-parameter tertentu.

Sintaks yang digunakan yaitu,

`exp(I*t)+exp(10*I*t)` : fungsi kompleks yang akan digambar. Fungsi ini terdiri dari dua bagian yang masing-masing merupakan fungsi kompleks eksponensial. Dengan 10 adalah berapa kali putaran dalam gambar tersebut.

`r` : parameter `r` digunakan untuk menentukan rentang nilai dari variabel `t`. Dalam contoh ini, `r=3`, yaitu mengatur rentang nilai `t` dari 3 hingga 3.

## Sub Bab 14

---

\* Menggambar Daerah Yang Dibatasi Kurva Plot data benar-benar poligon. Kita juga dapat memplot kurva atau kurva terisi.

Pada subtopik sebelumnya telah kita ketahui dan pelajari bersama bahwa EMT dapat melakukan visualisasi plot mulai dari bentuk ekspresi langsung hingga plot dari fungsi-fungsi. Pada subtopik ini merupakan kelanjutan dari subtopik sebelumnya, bahwa kita dapat membentuk/menggambar daerah dari perpotongan beberapa kurva yang telah didefinisikan. Hal ini dapat bermanfaat untuk membantu dalam menyelesaikan

permasalahan dalam matematika, salah satu contohnya seperti optimasi program linear, dimana disajikan beberapa fungsi-fungsi kendala beserta dengan fungsi tujuannya dan perlu divisualisasikan dalam bentuk grafik untuk melihat dimana letak daerah layaknya untuk menentukan nilai optimum.

Dalam EMT ada beberapa perintah yang digunakan untuk menggambar daerah yang dibatasi oleh beberapa kuva, diantaranya yaitu:

- plot2d

Digunakan untuk melakukan plotting.

- filled=true

Digunakan untuk memberikan isian/arsiran pada daerah/area di bawah kurva saat plotting.

- style="..."

Digunakan untuk memilih gaya kurva yang akan digunakan saat plotting. Anda dapat memilih dari beberapa gaya, seperti "", "/", "\", atau "-". Dan hal ini mempengaruhi tampilan daerah kurva yang terbentuk.

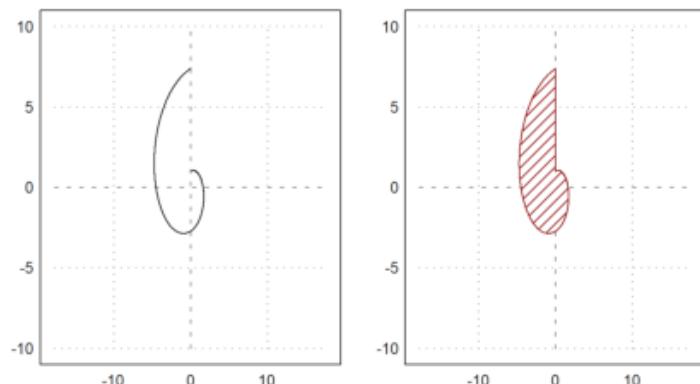
- fillcolor

Digunakan untuk menentukan warna isian yang akan digunakan untuk mengiri area di bawah kurva.

## Contoh

---

```
>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):
```



Penjelasan:

- t=linspace(0,2pi,1000);

Pada langkah pertama yaitu mendefinisikan parameter t sebagai serangkaian 1000 titik antara 0 dan 2pi. Parameter t ini akan digunakan sebagai parameter untuk menggambar kurva.

- x=sin(t)\*exp(t/pi); y=cos(t)\*exp(t/pi); // x(t) and y(t)

Kemudian kita definisikan dua vektor x dan y yang merupakan koordinat x dan y dari kurva yang akan digambar. Fungsi

$$\sin(t) * \exp(t/\pi)$$

digunakan untuk menghitung komponen x ( $x(t)$ ), dan

$$\cos(t) * \exp(t/\pi)$$

digunakan untuk menghitung komponen y ( $y(t)$ ) dari kurva.

- figure(1,2); aspect(16/9)

Perintah ini digunakan untuk mengatur tampilan gambar. Perintah figure(1,2) digunakan membuat dua gambar (1 dan 2) dalam satu jendela gambar. Dan perintah aspect(16/9) mengatur rasio aspek gambar menjadi 16:9, yang mempengaruhi bentuk dan ukuran gambar yang akan digambar.

- figure(1); plot2d(x,y,r=10); // membuat plot kurva

Perintah ini memilih gambar pertama (1) dan menggunakan perintah plot2d untuk menggambar kurva yang dihitung sebelumnya. Parameter r=10 mengatur lebar garis plot. Ini menghasilkan kurva tanpa adanya isi atau arsiran di dalamnya.

- figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // fill curve

Selanjutnya pada perintah ini beralih ke gambar kedua (2) dan menggunakan perintah plot2d lagi untuk menggambar kurva yang sama dengan pengisian area di bawahnya. Perintah >filled digunakan untuk mengisi area di bawah kurva, style="/" digunakan untuk mengatur gaya garis menjadi garis miring, dan fillcolor=red digunakan untuk mengatur warna isian menjadi merah.

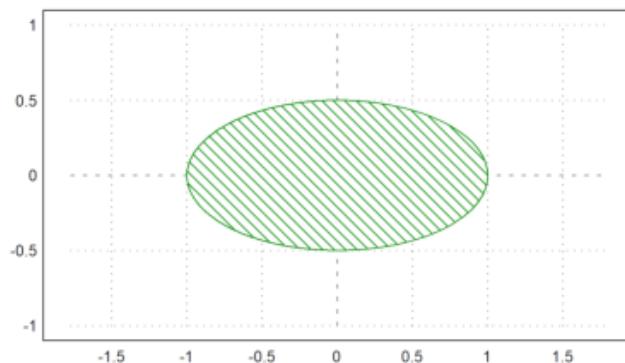
-figure(0):

Baris perintah ini digunakan untuk mengakhiri gambar dan kembali ke tampilan biasa tanpa gambar. Ini berfungsi untuk menyelesaikan proses penggambaran.

## Contoh

---

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



Penjelasan:

- `x=linspace(0,2pi,100);`

Mendefinisikan vektor x dengan menggunakan perintah linspace. linspace digunakan untuk membuat vektor dengan 100 titik yang secara merata tersebar antara 0 dan 2phi. Dalam konteks ini, vektor x akan digunakan sebagai parameter saat menggambar kurva.

- `plot2d(cos(x),sin(x)*0.5,r=1,>filled,style="\");`

Ini merupakan perintah utama yang digunakan untuk menggambar plot. Perintah ini memiliki beberapa parameter sebagai berikut:

> cos(x) adalah komponen x dari kurva. Ini adalah hasil dari fungsi kosinus yang dihitung pada vektor x.

> sin(x)\*0.5 adalah komponen y dari kurva. Ini adalah hasil dari fungsi sinus yang dihitung pada vektor x dan kemudian dikalikan dengan 0,5, yang mengubah amplitudonya.

> r=1 mengatur lebar garis plot menjadi 1.

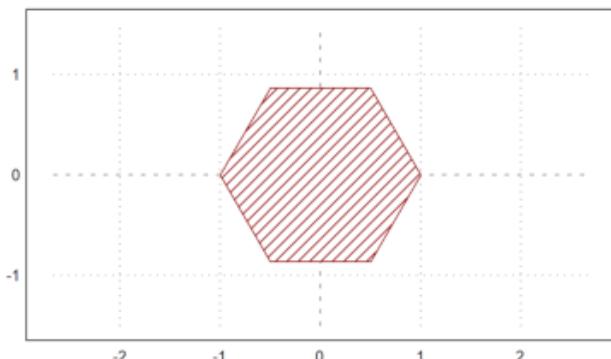
> filled digunakan untuk mengisi area di bawah kurva, sehingga menciptakan daerah yang terisi.

> style="/" mengatur gaya garis kurva untuk membentuk garis miring yang gunanya menutupi semua bagian kurva dengan garis miring.

## Contoh

---

```
>t=linspace(0,2pi,6); ...
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.5):
```



Penjelasan:

- `t=linspace(0,2pi,6); ...`

Pada perintah ini, kita definisikan vektor t dengan menggunakan

perintah linspace. linspace digunakan untuk membuat vektor dengan 6 titik yang terletak secara merata antara 0 dan 2pi. Dalam konteks ini, vektor t akan digunakan sebagai parameter saat menggambar kurva.

- `plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.5):`

Ini adalah perintah utama yang digunakan untuk menggambar plot.

Perintah ini memiliki beberapa parameter sebagai berikut:

> cos(t) adalah komponen x dari kurva.

> sin(t) adalah komponen y dari kurva.

> filled digunakan untuk mengisi area di bawah kurva, sehingga menciptakan bentuk yang terisi. Ini berarti daerah di bawah kurva akan diwarnai.

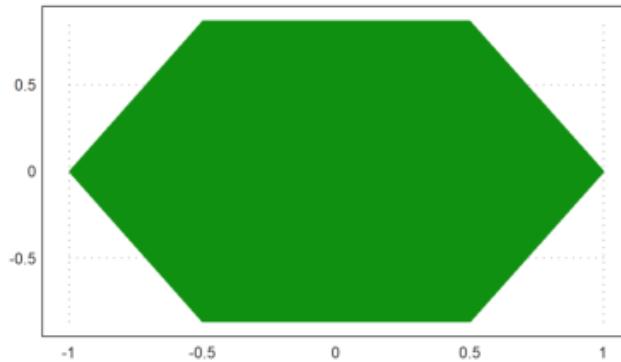
> style="/" mengatur gaya garis kurva menjadi garis miring ("/").

> fillcolor=orange mengatur warna isian daerah di bawah kurva menjadi oranye.  
> r=1.5 mengatur lebar garis plot menjadi 1.5.

## Contoh

---

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



Penjelasan:

- t=linspace(0,2pi,6);

Pada perintah ini, kita definisikan vektor t dengan menggunakan perintah linspace. linspace digunakan untuk membuat vektor dengan 6 titik yang terletak secara merata antara 0 dan 2phi. Dalam konteks ini, vektor t akan digunakan sebagai parameter saat menggambar kurva.

- plot2d(cos(t),sin(t),>filled,style=""):

Ini adalah perintah utama yang digunakan untuk menggambar plot. Perintah ini memiliki beberapa parameter sebagai berikut:

> cos(t) adalah komponen x dari kurva.

> sin(t) adalah komponen y dari kurva.

> filled digunakan untuk mengisi area di bawah kurva, sehingga menciptakan bentuk yang terisi. Ini berarti daerah di bawah kurva akan diisi dengan warna atau pola tertentu.

> style="" mengatur isian kurva menjadi warna solid dengan menggunakan simbol tanda pagar ("")

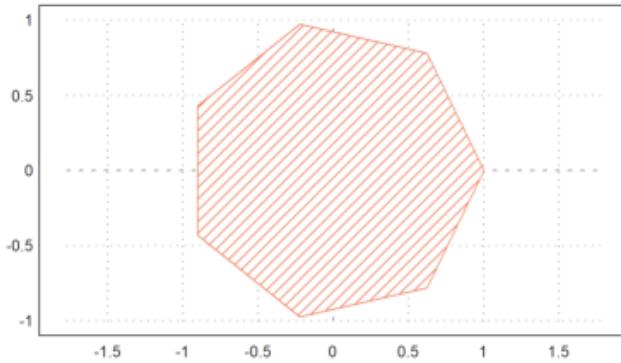
Pada contoh ini tidak ada perintah untuk mengatur warna, maka warna yang dihasilkan pada plot ini akan mengikuti pada warna yang disetting pada bagian sebelumnya.

## Contoh

---

Contoh lainnya adalah segi empat, yang kita buat dengan 7 titik pada lingkaran satuan.

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



Penjelasan:

- `t=linspace(0,2pi,7);`

Fungsi linspace digunakan untuk membuat array berisi sejumlah nilai yang merata dalam rentang tertentu. Dalam hal ini, rentangnya adalah dari 0 hingga  $2\pi$  (dua kali nilai  $\pi$ ) dan sebanyak 7 titik akan dihasilkan. Ini akan digunakan sebagai sudut dalam koordinat polar untuk menggambarkan data.

- `plot2d(cos(t),sin(t),r=1,>filled,style="/",fillcolor=orange);`

Ini adalah perintah untuk melakukan plotting data. Terdapat beberapa argumen di sini:

> `cos(t)`: Ini adalah nilai kosinus dari setiap elemen dalam array `t`. Ini akan digunakan sebagai komponen sumbu Y dalam koordinat polar.

> `sin(t)`: Ini adalah nilai sinus dari setiap elemen dalam array `t`. Ini akan digunakan sebagai komponen sumbu X dalam koordinat polar.

> `r=1`: Ini adalah argumen opsional yang menentukan radius plot. Dalam hal ini, radiusnya diatur menjadi 1.

> `filled`: Ini adalah argumen yang menginstruksikan untuk mengisi area di dalam kurva plot.

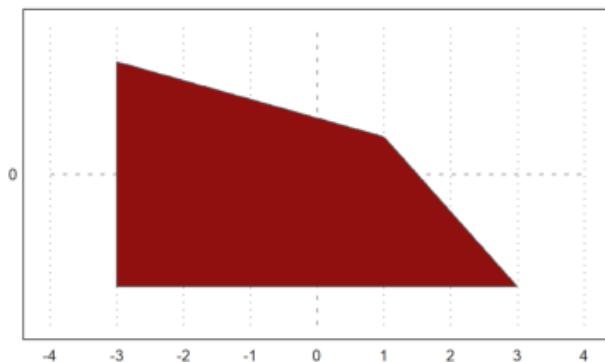
> `style="/"`: Ini adalah argumen yang menentukan gaya garis yang digunakan untuk plot. Di sini, garisnya akan berbentuk garis miring (/).

> `fillcolor=orange`: Ini adalah argumen yang menentukan warna pengisian untuk area di dalam kurva plot. Dalam hal ini, warnanya diatur menjadi oranye.

## Contoh

---

```
>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=red,n=111):
```



Penjelasan:

- A=[2,1;1,2;-1,0;0,-1];

Ini adalah perintah untuk membuat matriks A. Matriks ini memiliki dimensi 4x2, yang berarti memiliki 4 baris dan 2 kolom. Isinya adalah:

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

- function f(x,y) := max([x,y].A');

Ini adalah perintah untuk mendefinisikan sebuah fungsi bernama  $f(x, y)$ . Fungsi ini mengambil dua argumen input, yaitu  $x$  dan  $y$ . Fungsi ini melakukan operasi berikut:

>  $[x, y]$  menghasilkan vektor baris dengan elemen  $[x, y]$ .

>  $[x, y].A'$  adalah perkalian dot (dot product) dari vektor baris  $[x, y]$  dengan transpose dari matriks A.

>  $\max([x, y].A')$  menghitung nilai maksimum dari hasil perkalian dot tersebut.

Dengan kata lain, fungsi ' $f(x, y)$ ' mengambil vektor ' $[x, y]$ ' sebagai input, mengalikannya dengan matriks ' $A'$ , dan mengembalikan nilai maksimum dari hasil perkalian tersebut.

- plot2d("f",r=4,level=[0;3],color=red,n=111):

Ini adalah perintah untuk membuat plot 2D dari fungsi ' $f(x, y)$ ' yang telah didefinisikan. Rincian perintah ini adalah sebagai berikut:

> "f" adalah nama fungsi yang akan diplot.

> r=4 menentukan rentang plot, yang dalam hal ini adalah  $[-4, 4]$  untuk kedua sumbu x dan y.

> level=[0;3] menentukan tingkat kontur (contour levels) yang akan digunakan dalam plot. Ada dua tingkat kontur: 0 dan 3.

> color=green mengatur warna kontur plot menjadi merah.

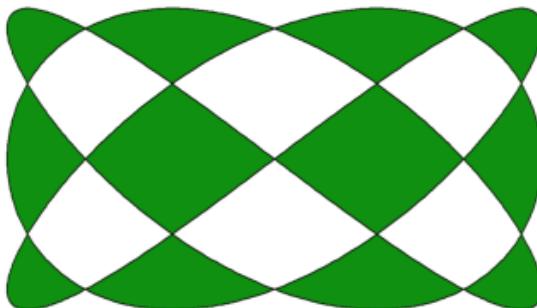
> n=111 mengendalikan jumlah titik yang digunakan dalam plot.

Hasilnya akan menjadi sebuah grafik kontur 2D dari fungsi ' $f(x, y)$ ' dengan kontur berwarna merah pada tingkat 0 dan 3, yang mencakup rentang -4 hingga 4 pada kedua sumbu x dan y.

## Contoh

---

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);  
>plot2d(x,y,<grid,<frame,>filled):
```



Penjelasan:

- t = linspace(0, 2\*pi, 1000);

Ini adalah perintah untuk membuat vektor t yang berisi 1000 nilai yang merata terdistribusi antara 0 hingga  $2\pi$ . Vektor t ini akan digunakan sebagai parameter waktu atau sudut dalam parameterisasi lingkaran.

linspace(0, 2\*pi, 1000) membuat 1000 titik antara 0 hingga  $2\pi$ , memberikan sudut-sudut yang merata di sepanjang satu putaran lingkaran.

-  $x = \cos(3*t)$ ;  $y = \sin(4*t)$ ;

Ini adalah perintah untuk menghitung vektor x dan y yang menggambarkan lintasan dalam koordinat polar.  
->  $x = \cos(3*t)$ ; menghitung nilai x sebagai hasil dari fungsi kosinus dari 3 kali nilai t. Ini akan menghasilkan osilasi yang lebih cepat pada sumbu x.

>  $y = \sin(4*t)$ ; menghitung nilai y sebagai hasil dari fungsi sinus dari 4 kali nilai t. Ini akan menghasilkan osilasi yang lebih cepat pada sumbu y.

- plot2d(x, y, <grid, <frame, >filled);

Ini adalah perintah untuk membuat plot dari vektor x dan y. Berikut adalah rincian perintah ini:

x adalah vektor yang digunakan sebagai data untuk sumbu x.

y adalah vektor yang digunakan sebagai data untuk sumbu y.

<grid mengaktifkan garis-garis koordinat (grid) di latar belakang plot, membantu dalam visualisasi.

<frame mengaktifkan bingkai (frame) di sekitar plot.

>filled mengisi area di bawah kurva dengan warna, membuat plot menjadi lebih berwarna.

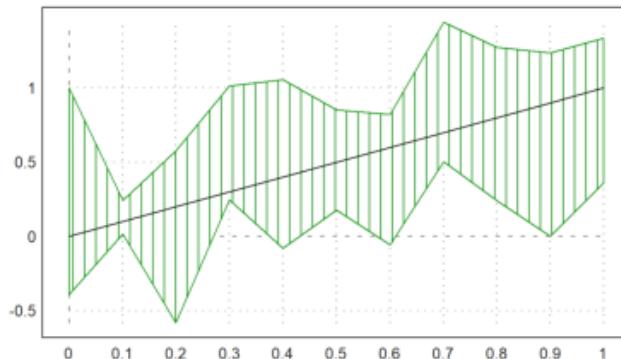
## Contoh

---

Sebuah vektor interval diplot terhadap nilai x sebagai daerah terisi antara nilai interval bawah dan atas.

Ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga 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):
```



Penjelasan:

-  $t = 0:0.1:1$ ;

Ini adalah perintah untuk membuat vektor t yang berisi nilai-nilai dari 0 hingga 1 dengan interval 0.1. Hasilnya adalah vektor  $[0, 0.1, 0.2, 0.3, \dots, 0.9, 1]$ .

- plot2d(t, interval(t - random(size(t)), t + random(size(t))), style="|");

Ini adalah perintah untuk membuat plot pertama. Rincian perintah ini adalah sebagai berikut:

> interval(t - random(size(t)), t + random(size(t))) adalah interval yang digunakan untuk menggambar "garis" pada plot. Setiap titik pada sumbu x (t) akan dihubungkan oleh dua garis vertikal yang dibuat secara acak di sekitar titik tersebut menggunakan random(size(t)). Hasilnya adalah plot dengan garis-garis vertikal yang mewakili interval acak di sekitar setiap titik pada sumbu x.

> style="|" mengatur gaya plot menjadi garis vertikal ("|").

- plot2d(t, t, add=true);

Ini adalah perintah untuk membuat plot kedua dan menambahkannya ke dalam plot yang sudah ada dari perintah sebelumnya. Rincian perintah ini adalah sebagai berikut:

> t adalah sumbu x dan y plot ini, sehingga plot ini akan menjadi plot garis diagonal dengan kemiringan 45 derajat.

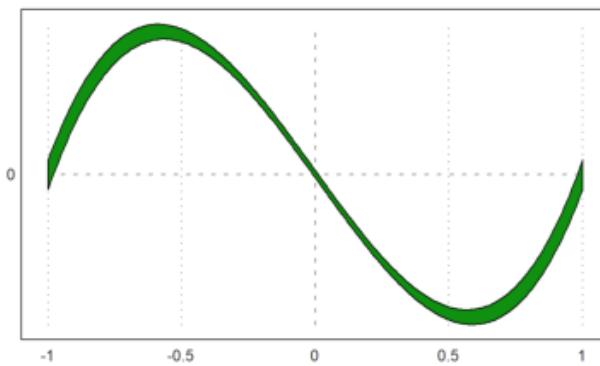
> add=true digunakan untuk menambahkan plot ini ke dalam plot sebelumnya, sehingga kedua plot akan ditampilkan dalam satu plot yang sama.

## Contoh

---

Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

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



Penjelasan:

- t = -1:0.01:1;

Ini adalah perintah untuk membuat vektor t yang berisi nilai-nilai dari -1 hingga 1 dengan interval 0.01. Hasilnya adalah vektor t yang berisi nilai-nilai seperti [-1, -0.99, -0.98, ..., 0.99, 1]. Vektor t ini akan digunakan sebagai sumbu x pada plot.

- x = t - 0.01, t + 0.01 ;

Ini adalah perintah yang menghitung vektor x. Tanda ~ digunakan di sini untuk mendefinisikan dua interval, yaitu [ t - 0.01, t + 0.01 ]. Ini menghasilkan vektor x yang memiliki dua interval, satu yang kurang dari t - 0.01 dan satu yang lebih dari t + 0.01.

- y = x^3 - x;

Ini adalah perintah yang menghitung vektor y sebagai fungsi dari x. Fungsi ini menghitung nilai y dengan memasukkan setiap nilai x ke dalam rumus  $x^3 - x$ .

- plot2d(t, y);

Ini adalah perintah untuk membuat plot dari fungsi y sebagai fungsi dari t. Rincian perintah ini adalah sebagai berikut:

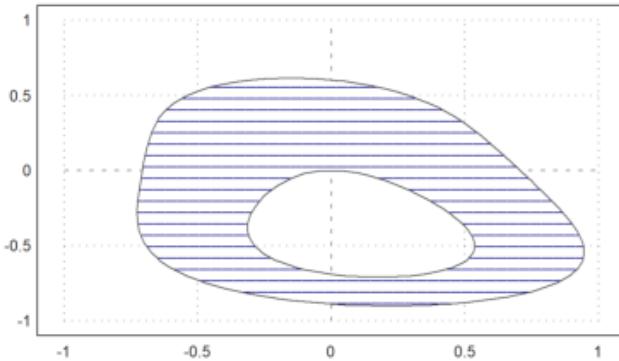
> t adalah sumbu x pada plot, yang berisi vektor t yang telah didefinisikan sebelumnya.

> y adalah sumbu y pada plot, yang berisi vektor y yang dihitung dari rumus  $x^3 - x$ .

## Contoh

---

```
>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
```



Penjelasan:

- expr := "2\*x^2+x\*y+3\*y^4+y";

Ini adalah perintah untuk mendefinisikan ekspresi matematika yang disimpan dalam variabel expr. Ekspresi ini merupakan suatu fungsi  $f(x, y)$  yang tergantung pada dua variabel, yaitu  $x$  dan  $y$ . Ekspresi ini memiliki bentuk matematika yang terdiri dari berbagai suku, seperti kuadrat dari  $x$ , perkalian  $x*y$ , kuadrat dari  $y$ , dan lainnya.

- plot2d(expr, level=[0;1], style="-", color=blue);

Ini adalah perintah untuk membuat plot dari fungsi  $f(x, y)$  yang telah didefinisikan sebelumnya. Berikut adalah rincian perintah ini:

> expr adalah ekspresi yang akan digunakan sebagai fungsi yang akan diplotkan. Dalam hal ini, ekspresi  $2*x^2+x*y+3*y^4+y$  adalah fungsi  $f(x, y)$  yang telah didefinisikan sebelumnya.

> level=[0;1] mengatur tingkat kontur (contour levels) yang akan digunakan dalam plot. Dalam hal ini, tingkat kontur adalah 0 hingga 1, yang berarti plot akan menunjukkan wilayah di mana  $f(x, y)$  memiliki nilai antara 0 hingga 1.

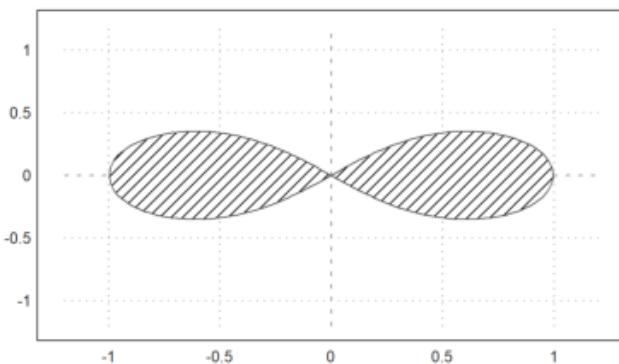
> style="-" mengatur gaya plot menjadi garis berjenis -, yang akan menghasilkan plot kontur.

> color=blue mengatur warna garis plot menjadi biru.

## Contoh

---

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



Penjelasan:

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

Ini adalah perintah untuk membuat plot dari fungsi matematika yang didefinisikan dalam bentuk string: "(x^2+y^2)^2-x^2+y^2". Fungsi ini tergantung pada dua variabel, yaitu  $x$  dan  $y$ .

$(x^2+y^2)^2-x^2+y^2$  adalah rumus dari fungsi matematika yang akan diplotkan.

`r=1.2` mengatur rentang (range) plot untuk kedua sumbu x dan y. Dalam hal ini, rentangnya adalah [-1.2, 1.2], yang berarti plot akan berada dalam wilayah ini.

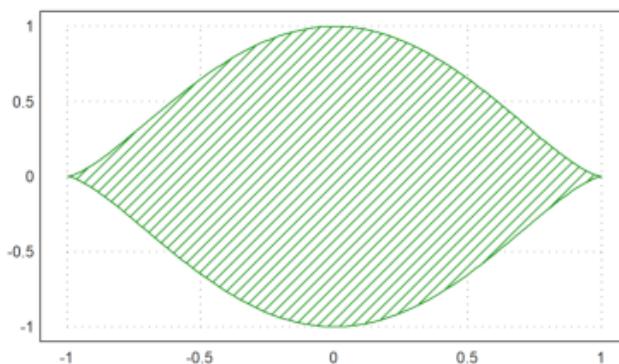
`level=[-1;0]` mengatur tingkat kontur (contour levels) yang akan digunakan dalam plot. Dalam hal ini, ada dua tingkat kontur: -1 dan 0. Ini akan menentukan wilayah kontur dalam plot.

`style="/"` mengatur gaya plot menjadi garis miring ("/"). Ini akan menghasilkan plot dengan garis-garis miring yang menggambarkan kontur fungsi.

## Contoh

---

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



Penjelasan:

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

Ini adalah perintah untuk membuat plot dari dua fungsi matematika, yaitu  $\sin(x)^3$  dan  $\cos(x)$ , dalam satu plot yang sama. Berikut adalah rincian perintah ini:

" $\sin(x)^3$ " adalah ekspresi pertama yang akan diplotkan. Ini adalah fungsi trigonometri  $\sin(x)$  yang dipangkatkan tiga. Fungsi ini tergantung pada variabel x.

" $\cos(x)$ " adalah ekspresi kedua yang akan diplotkan. Ini adalah fungsi trigonometri  $\cos(x)$ . Fungsi ini juga tergantung pada variabel x.

`xmin=0` dan `xmax=2*pi` mengatur rentang (range) plot untuk sumbu x dari 0 hingga  $2\pi$ . Ini adalah rentang yang akan ditampilkan dalam plot.

`>filled` mengisi area di bawah kurva fungsi dengan warna, sehingga area di bawah kurva fungsi akan diisi dengan warna.

`style="/"` mengatur gaya plot menjadi garis miring ("\"). Ini akan menghasilkan plot dengan garis-garis miring. **Sub Bab 15**

---

\* Menggambar Segi Banyak Data plot merupakan poligon atau segi banyak. Kita juga dapat membuat kurva atau mengisi kurva.

Fungsi perintah yang digunakan untuk menggambar segi banyak atau poligon.

Membentuk poligon dengan fungsi:

```
x=linspace(0,2pi,n); plot2d(cos(x),sin(x),r=1,>filled,style="...");
```

atau

```
x=linspace(0,2pi,n); plot2d(sin(x),cos(x),r=1,>filled,style="...",fillcolor=red);
```

Keterangan

- `filled=true`, mengisi plot.

- `style="...":` Pilih dari "", "/", "\", "\/" dan gaya gaya lainnya.

- `fillcolor:` untuk memberikan warna.

Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional mencegah menggambar batas untuk semua gaya kecuali yang default.

Poligon dalam EMT dapat digambar dengan fungsi maksimal. Dengan fungsi maksimal ini, poligon yang dihasilkan dapat berupa poligon tak beraturan.

```
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);
```

Keterangan:

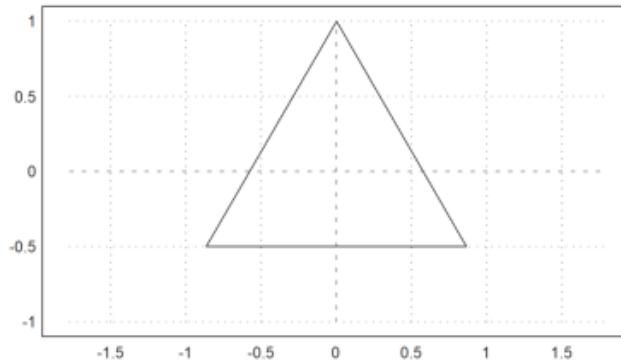
-A adalah titik koordinat dari poligon yang akan dibuat.

-"r" untuk menentukan ukuran bidang koordinat.

Berikut adalah himpunan nilai maksimal dari empat kondisi linear yang kurang dari atau sama dengan 3. Ini merupakan  $A[k].v \leq 3$  untuk semua baris A. Untuk mendapatkan sudut yang bagus, kita menggunakan n yang relatif besar.

### 1. Menggambar Segitiga

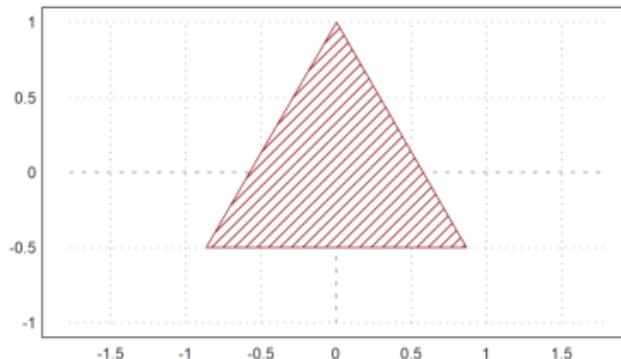
```
>x=linspace(0,2pi,3); ...
>plot2d(sin(x),cos(x),r=1);
```



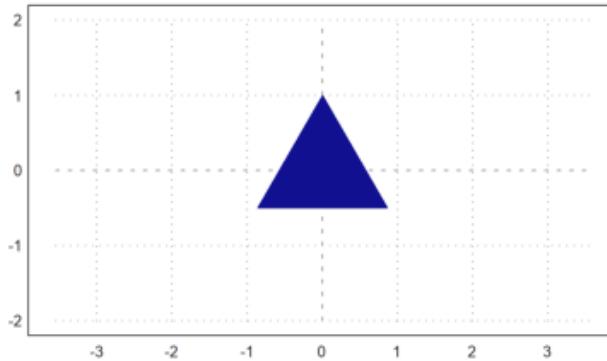
Segitiga diatas digambar dari kurva tertutup dengan 3 titik.

Kita dapat membuat segitiga dengan gaya yang berbeda-beda. Seperti pada contoh berikut ini.

```
>x=linspace(0,2pi,3); ...
>plot2d(sin(x),cos(x),>filled,style="/",fillcolor=red,r=1);
```



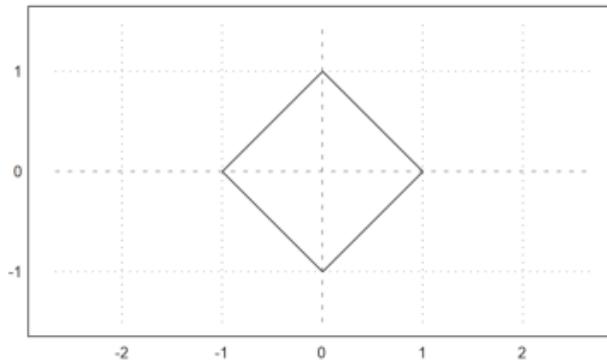
```
>x=linspace(0,2pi,3); ...
>plot2d(sin(x),cos(x),>filled,style="#",fillcolor=blue,r=2):
```



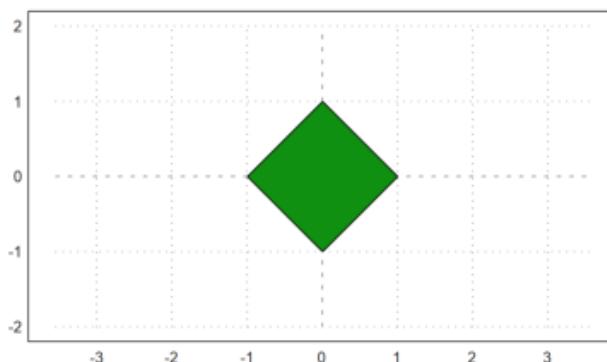
Dua gambar segitiga diatas memiliki gaya yang berbeda, dengan menggunakan fungsi perintah "style=". Gambar segitiga juga dapat dibuat dengan posisi yang berbeda, tergantung pada fungsi yang akan diplot.

## 2. Menggambar Segiempat

```
>x=linspace(0,2pi,4); ...
>plot2d(cos(x),sin(x),r=1.5):
```



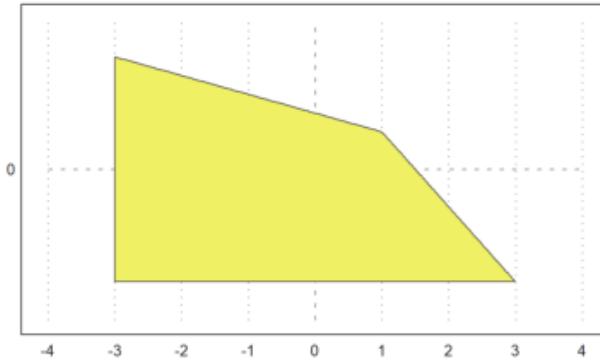
```
>x=linspace(0,2pi,4);
>plot2d(cos(x),sin(x),r=2,>filled,outline=1):
```



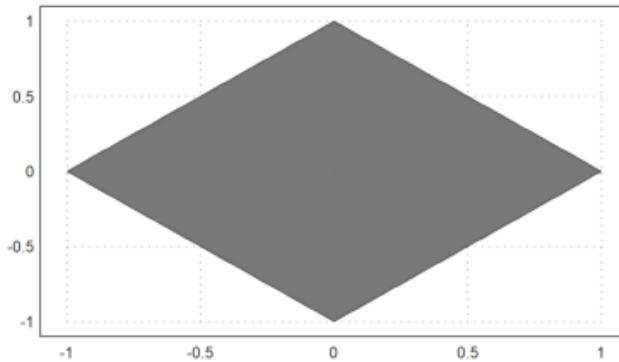
Gambar diatas merupakan salah satu contoh segiempat yang dapat digambar di EMT. Fungsi perintah yang digunakan masih sama seperti fungsi perintah untuk menggambar segitiga.

Selain fungsi perintah diatas, untuk menggambar segi banyak, dapat menggunakan fungsi maksimum.

```
>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=yellow,n=111):
```



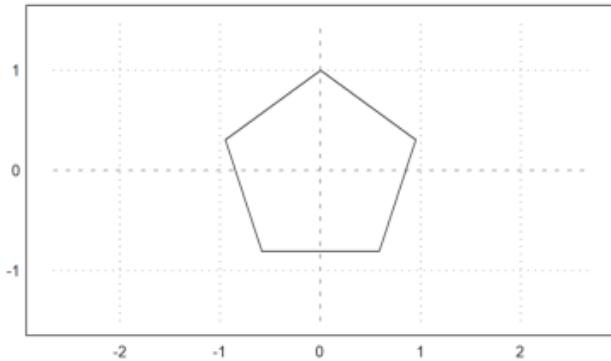
```
>A=[1,1;-1,1;-1,-1;1,-1];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=1,level=[0;1],color=gray,n=90):
```



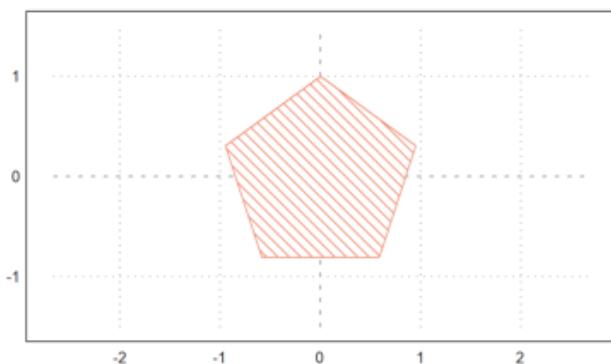
Dengan fungsi maksimal ini, kita dapat menggambar segiempat atau segi banyak sebarang.

### 3. Menggambar Segilima

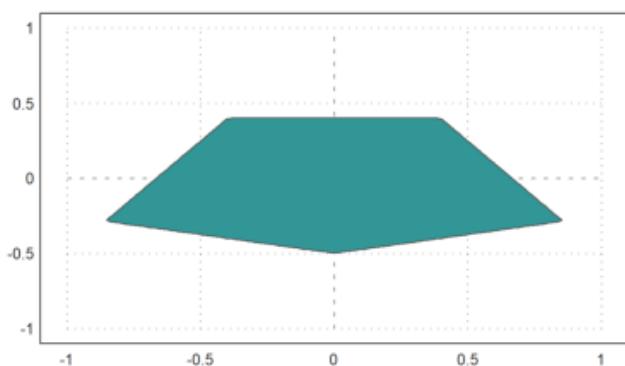
```
>t=linspace(0,2pi,5); plot2d(sin(t),cos(t),r=1.5):
```



```
>t=linspace(0,2pi,5); ...
>plot2d(sin(t),cos(t),r=1.5,>filled,style="\\",fillcolor=orange):
```

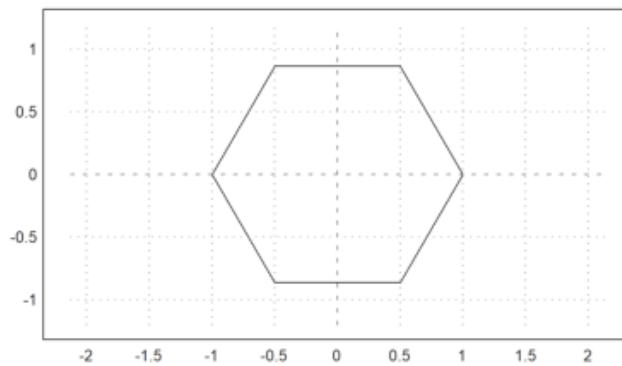


```
>A=[0,5;3,2;1,-4;-1,-4;-3,2];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=1,level=[0;2],color=cyan,n=111):
```

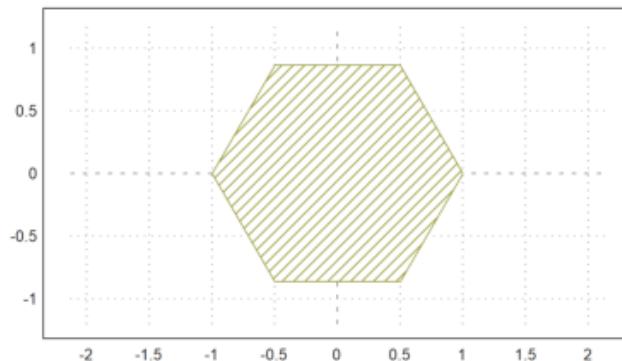


#### 4. Menggambar Segienam

```
>t=linspace(0,2pi,6); ...
>plot2d(cos(t),sin(t),r=1.2):
```

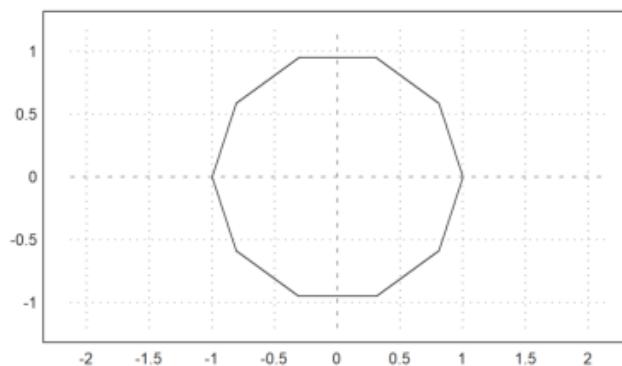


```
>t=linspace(0,2pi,6); ...
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=olive,r=1.2):
```

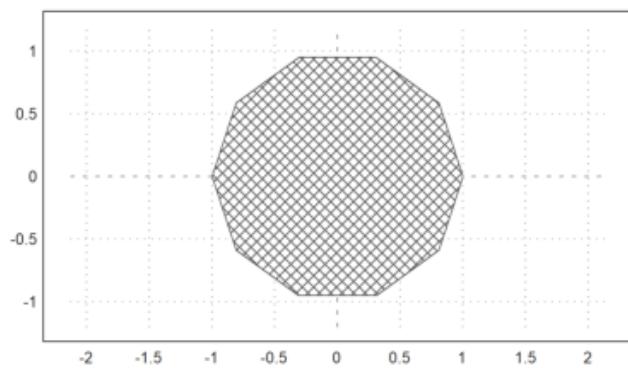


## 5. Menggambar dekagon

```
>t=linspace(0,2pi,10); ...
>plot2d(cos(t),sin(t),r=1.2):
```



```
>t=linspace(0,2pi,10); ...
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=darkgray,r=1.2):
```



---

---

## BAB 4

---

# PEKAN 5: MENGGUNAKAN EMT UNTUK MENGAMBAR GRAFIK 3 DIMENSI (3D)

[a4paper,10pt]article eumat  
Raden Mas Farrel Maheswara Kusuma Dewa  
22305141042  
Matematika E

### Menggambar Plot 3D dengan EMT Ini adalah pengenalan plot 3D di

---

Euler. Kami membutuhkan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

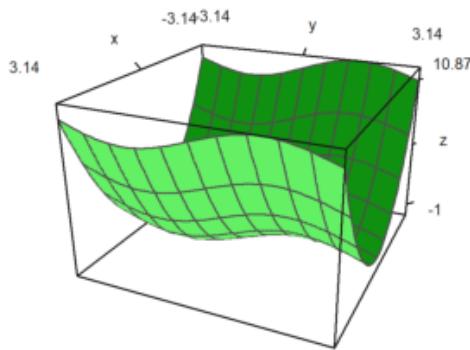
Euler menggambar fungsi seperti itu menggunakan algoritma pengurutan untuk menyembunyikan bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Defaultnya adalah dari kuadran x-y positif ke arah asal  $x = y = z = 0$ , tetapi sudut = 0 ? melihat dari arah sumbu y. Sudut pandang dan ketinggian dapat diubah.

Euler bisa merencanakan

- permukaan dengan bayangan dan garis level atau rentang level,
- awan poin,
- kurva parametrik,
- permukaan implisit.

Plot 3D dari suatu fungsi menggunakan plot3d. Cara termudah adalah dengan memplot ekspresi dalam x dan y. Parameter r mengatur kisaran plot sekitar (0,0).

```
>aspect(1.5); plot3d("x^2+sin(y)",r=pi):
```



## Fungsi Dua Variabel

---

Untuk grafik suatu fungsi, gunakan

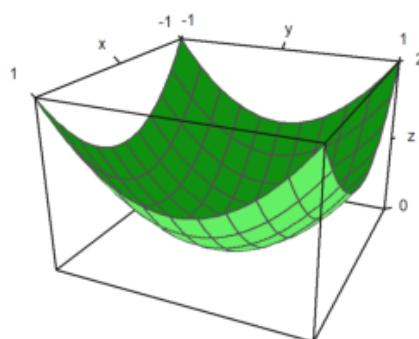
- ekspresi sederhana dalam  $x$  dan  $y$ ,
- nama fungsi dari dua variabel l
- atau matriks data.

Defaultnya adalah kisi kawat yang diisi dengan warna berbeda di kedua sisi. Perhatikan bahwa jumlah default interval kisi adalah 10, tetapi plot menggunakan jumlah default persegi panjang  $40 \times 40$  untuk membuat permukaan. Ini bisa diubah.

- $n = 40, n = [40,40]$ : jumlah garis grid di setiap arah
- $\text{grid}=10, \text{grid}=[10,10]$ : jumlah garis kisi di setiap arah.

Kami menggunakan default  $n = 40$  dan  $\text{grid} = 10$ .

```
>plot3d("x^2+y^2") :
```

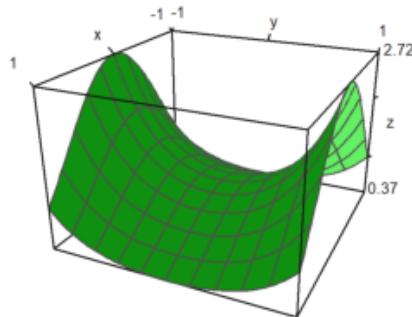


Interaksi pengguna dimungkinkan dengan > user parameter. Pengguna dapat menekan tombol berikut.

- left,right,up,down: putar sudut pandang
- +, -: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l: beralih memutar sumber cahaya (lihat di bawah)
- space: reset ke default
- return: interaksi akhir

```
>plot3d("exp(-x^2+y^2)",>user, ...
> title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



Rentang plot untuk fungsi dapat ditentukan dengan

- a, b: rentang-x
- c, d: rentang y
- r: persegi simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

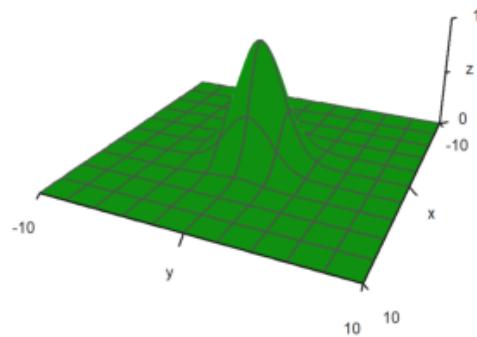
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: menskalakan ke nilai fungsi (defaultnya adalah <fscale>).

scale: angka atau vektor 1x2 untuk skala ke arah x dan y.

frame: jenis bingkai (default 1)

```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3):
```



Tampilan dapat diubah dengan berbagai cara.

- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut ke sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.

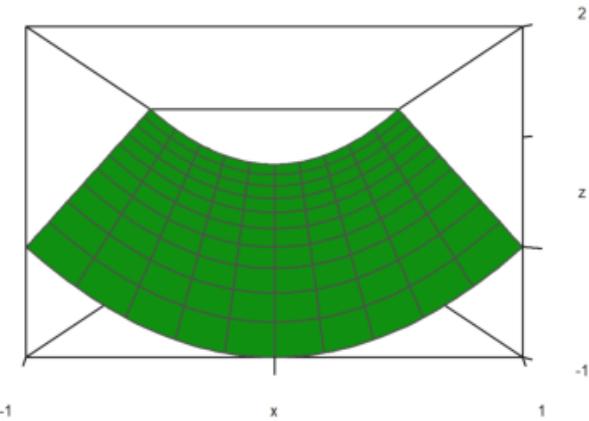
Nilai default bisa diperiksa atau diubah dengan fungsi view (). Ini mengembalikan parameter dalam urutan di atas.

```
>view
```

```
[5, 2.6, 2, 0.4]
```

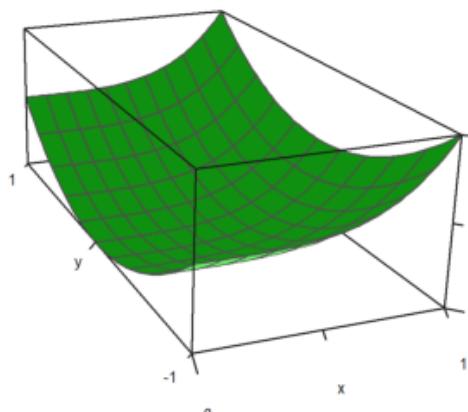
Jarak yang lebih dekat membutuhkan lebih sedikit zoom. Efeknya lebih seperti lensa sudut lebar. Dalam contoh berikut, sudut = 0 dan tinggi = 0 dilihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^2+y", distance=3, zoom=2, angle=0, height=0) :
```



Plot selalu terlihat ke tengah plot kubus. Anda dapat memindahkan pusat dengan parameter tengah.

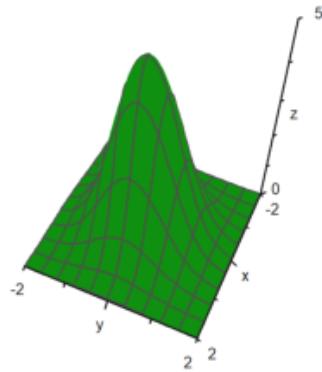
```
>plot3d("x^4+y^2", a=0, b=1, c=-1, d=1, angle=-20°, height=20°, ...
> center=[0.4, 0, 0], zoom=5) :
```



Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung ukuran plot. Namun, label mengacu pada ukuran sebenarnya.

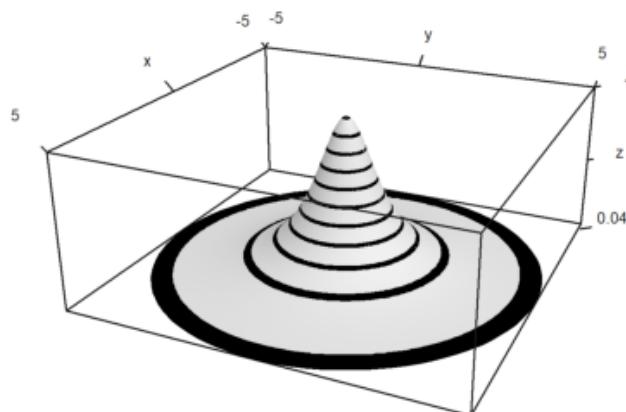
Jika Anda mematikannya dengan scale = false, Anda harus berhati-hati, bahwa plot masih pas dengan jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan pusat.

```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
> center=[0,0,-2],frame=3):
```

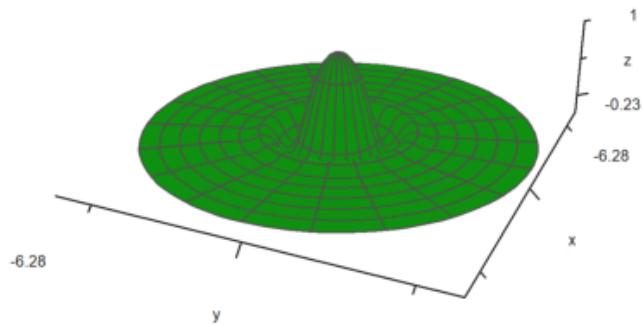


Plot kutub juga tersedia. The parameter polar= true menggambarkan plot kutub. Fungsi tersebut harus tetap merupakan fungsi dari x dan y. Parameter "fscale" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi diskalakan agar sesuai dengan kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=gray):
```



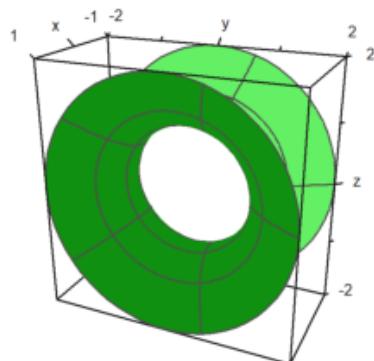
```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=2pi,frame=3,zoom=4):
```



Parameter memutar memutar fungsi dalam x di sekitar sumbu x.

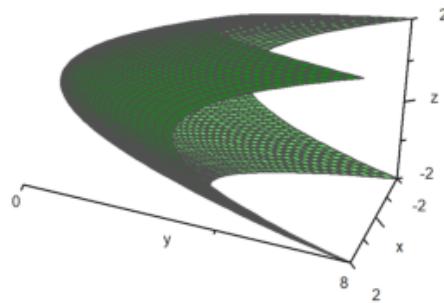
- rotate = 1: Menggunakan sumbu x
- rotate = 2: Menggunakan sumbu z

```
>plot3d("x^2+1", a=-1, b=1, rotate=true, grid=5):
```



Berikut adalah plot dengan tiga fungsi.

```
>plot3d("x", "x^2+y^2", "y", r=2, zoom=3.5, frame=3):
```



## Plot Kontur

Untuk plot, Euler menambahkan garis kisi. Alih-alih, dimungkinkan untuk menggunakan garis level dan corak satu warna atau corak warna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan shading. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah / cyan.

-> hue: Mengaktifkan bayangan terang, bukan kabel.

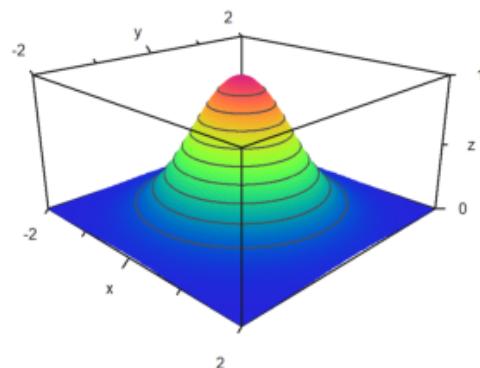
-> contour: Membuat plot garis kontur otomatis pada plot.

- level = ... (atau level): Vektor nilai untuk garis kontur.

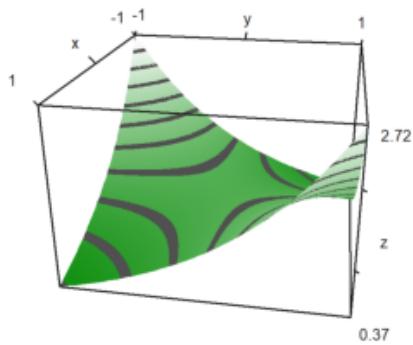
Standarnya adalah level = "auto", yang menghitung beberapa baris level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut, kami menggunakan kisi yang lebih halus untuk 100x100 titik, menskalakan fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...
>>contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):
```



```
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```



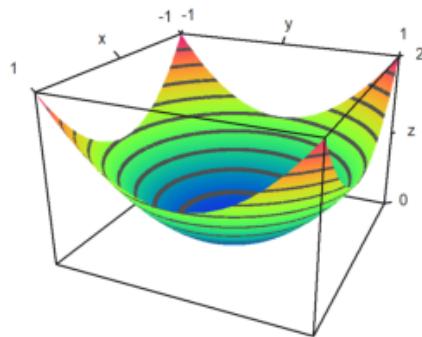
Bayangan default menggunakan warna abu-abu. Tetapi berbagai spektrum warna juga tersedia.

-> spectral: Menggunakan skema spektral default

- color = ...: Menggunakan warna khusus atau skema spektral

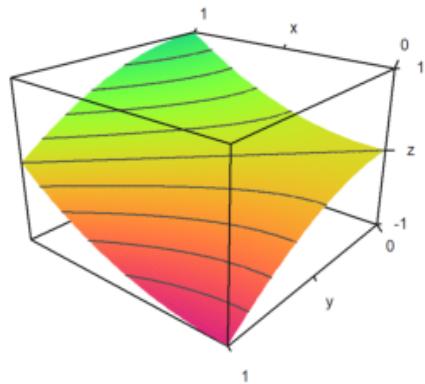
Untuk plot berikut, kami menggunakan skema spektral default dan menambah jumlah poin untuk mendapatkan tampilan yang sangat mulus.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```



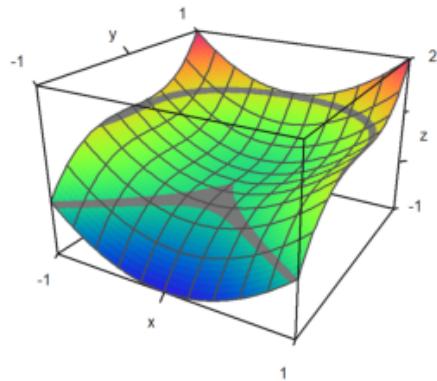
Alih-alih garis level otomatis, kami juga dapat mengatur nilai garis level. Ini akan menghasilkan garis level tipis, bukan rentang level.

```
>plot3d("x^2-y^2",0,1,0,1,angle=220°,level=-1:0.2:1,color=redgreen):
```



Dalam plot berikut, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas level sebagai kolom.  
Selain itu, kami melapisi kisi dengan 10 interval di setiap arah.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...
> >spectral,angle=30°,grid=10,contourcolor=gray):
```

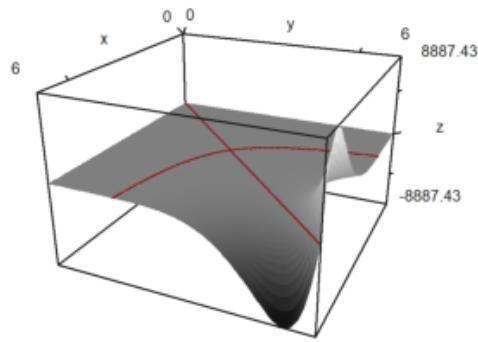


Dalam contoh berikut, kami memplot set, di mana

$$f(x, y) = x^y - y^x = 0$$

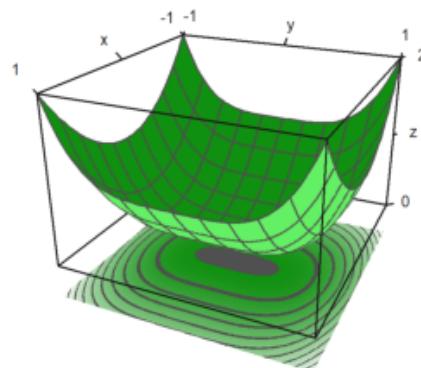
Kami menggunakan satu garis tipis untuk garis level.

```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```



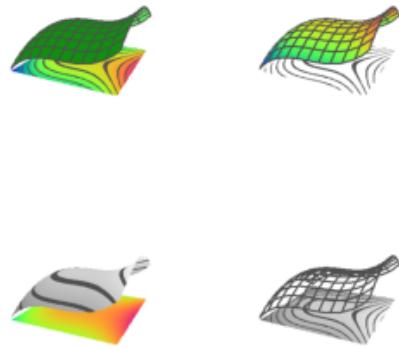
Dimungkinkan untuk menunjukkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

```
>plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2):
```



Berikut beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi.

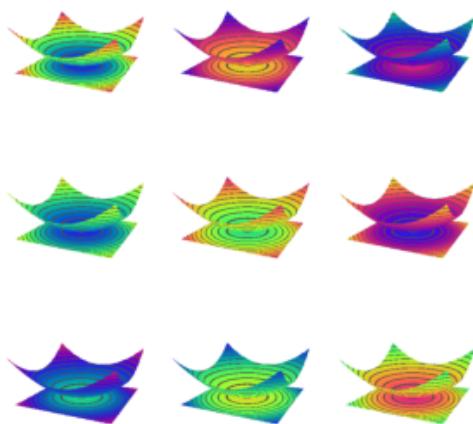
```
>figure(2,2); ...
>expr="y^3-x^2"; ...
>figure(1); ...
> plot3d(expr,<frame,>cp,cpcolor=spectral); ...
>figure(2); ...
> plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
>figure(3); ...
> plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
>figure(4); ...
> plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...
>figure(0):
```



Ada beberapa skema spektral lain, diberi nomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan color=value, di mana nilai

- spectral: untuk rentang dari biru hingga merah
- white: untuk rentang yang lebih redup
- yellowblue,purplegreen,blueyellow,greenred
- blueyellow, greenpurple,yellowblue,redgreen

```
>figure(3,3); ...
>for i=1:9; ...
> figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4); ...
>end; ...
>figure(0):
```



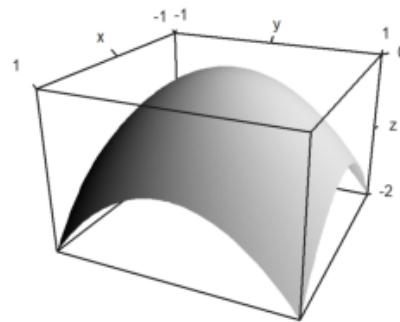
Sumber cahaya dapat diubah dengan l dan tombol cursor selama interaksi pengguna. Itu juga dapat diatur dengan parameter.

- light : arah cahaya
- amb: cahaya ambient antara 0 dan 1

Perhatikan bahwa program tidak membuat perbedaan antara sisi-sisi plot. Tidak ada bayangan. Untuk ini, Anda membutuhkan Povray.

```
>plot3d("-x^2-y^2", ...
>  hue=true,light=[0,1,1],amb=0,user=true, ...
>  title="Press l and cursor keys (return to exit)":
```

Press l and cursor keys (return to exit)



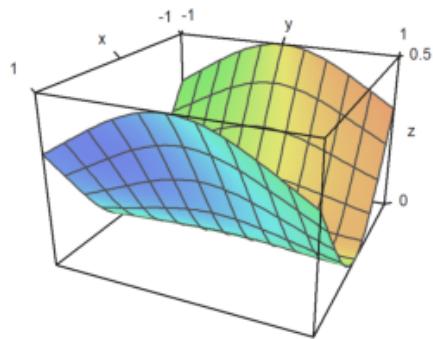
Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
>  zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01):
```



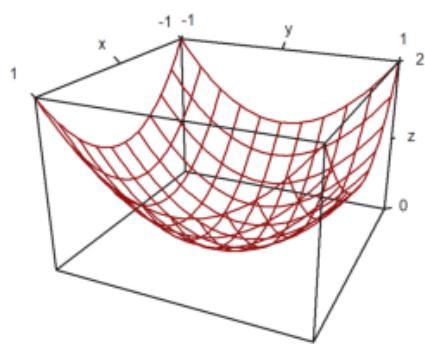
Warna 0 memberikan efek pelangi khusus.

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



Permukaannya juga bisa transparan.

```
>plot3d("x^2+y^2", >transparent, grid=10, wirecolor=red) :
```



## Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan pemotongan melalui objek. Fitur plot3d termasuk plot implisit. Plot ini menunjukkan himpunan nol fungsi dalam tiga variabel.

Solusi dari

$$f(x, y, z) = 0$$

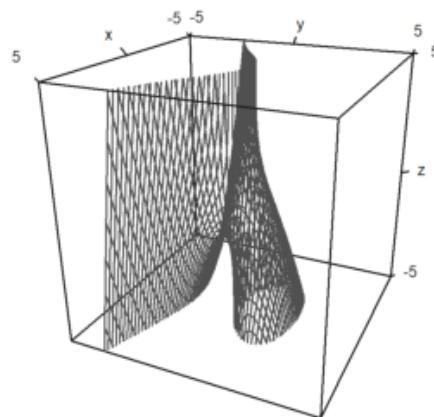
dapat divisualisasikan dalam potongan sejajar dengan bidang x-y-, x-z-, dan y-z.

- implisit = 1: potong sejajar bidang y-z
- implisit = 2: potong sejajar dengan bidang x-z
- implisit = 4: potong sejajar bidang x-y

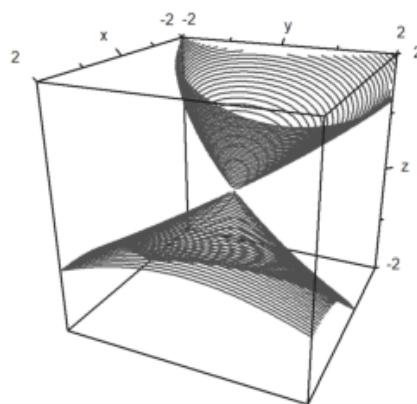
Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh yang kami plot

$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

```
>plot3d("x^2+y^3+z*y-1", r=5, implicit=3):
```



```
>plot3d("x^2+y^2+4*x*z+z^3", >implicit, r=2, zoom=2.5):
```



## Plotting 3D Data

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu memberikan matriks nilai x, y dan z, atau tiga fungsi atau ekspresi  $f_x(x, y)$ ,  $f_y(x, y)$ ,  $f_z(x, y)$ .

$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

Karena x, y, z adalah matriks, kita asumsikan bahwa (t, s) berjalan melalui kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di luar angkasa.

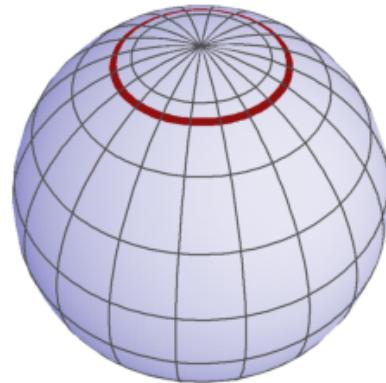
Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

Dalam contoh berikut, kami menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar kita bisa menandai daerah, dalam kasus kita daerah kutub.

```

>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...
>plot3d(x,y,z,>hue, ...
>color=blue,<frame,grid=[10,20], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.4,height=50°):

```

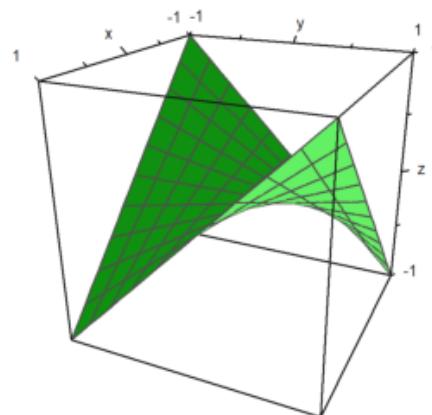


Berikut adalah contoh grafik dari suatu fungsi.

```

>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):

```



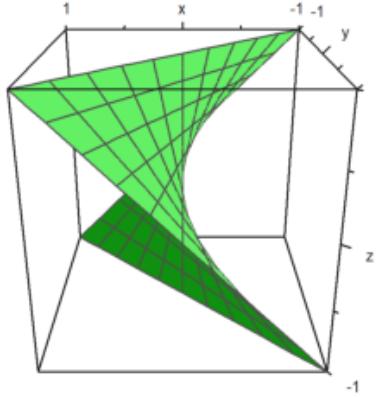
Namun, kita bisa membuat semua jenis permukaan. Ini adalah permukaan yang sama sebagai suatu fungsi

$$x = yz$$

```

>plot3d(t*s,t,s,angle=180°,grid=10):

```



Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

Dalam contoh berikut, kami membuat tampilan berbayang dari bola yang terdistorsi. Koordinat biasa untuk bola adalah

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

dengan

$$0 \leq t \leq 2\pi, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2}.$$

Kami menyimpangkan ini dengan sebuah faktor

$$d(t, s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

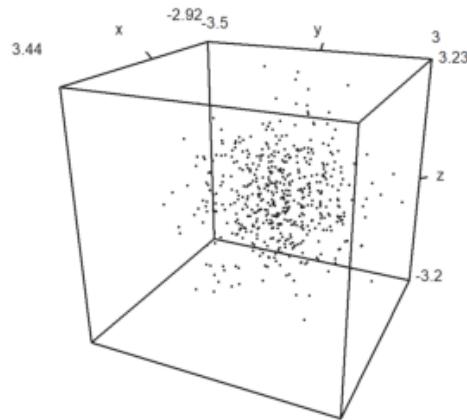
```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
> light=[1,0,1],frame=0,zoom=5):
```



Tentu saja, point cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor sebagai koordinat titik.

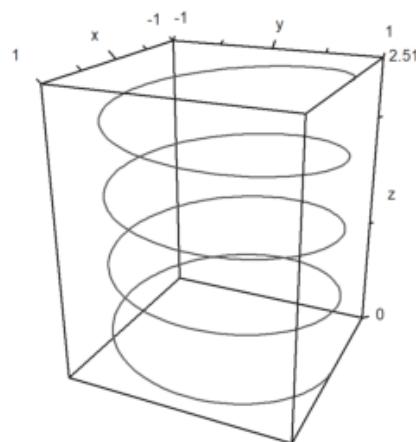
Gayanya sama seperti di plot2d dengan points = true;

```
>n=500; ...
> plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

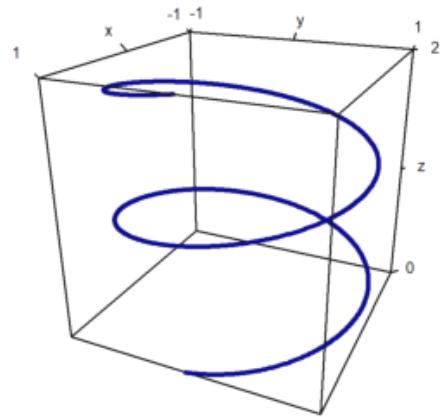


Juga dimungkinkan untuk memplot kurva dalam 3D. Dalam kasus ini, lebih mudah untuk menghitung sebelumnya titik-titik kurva. Untuk kurva di bidang kita menggunakan urutan koordinat dan parameter wire = true.

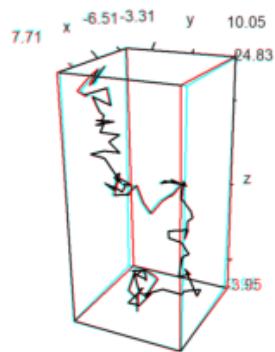
```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
>linewidth=3, wirecolor=blue):
```

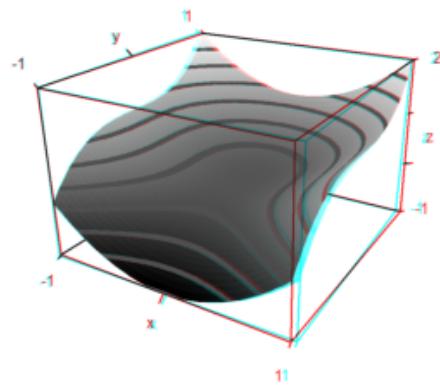


```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire) :
```



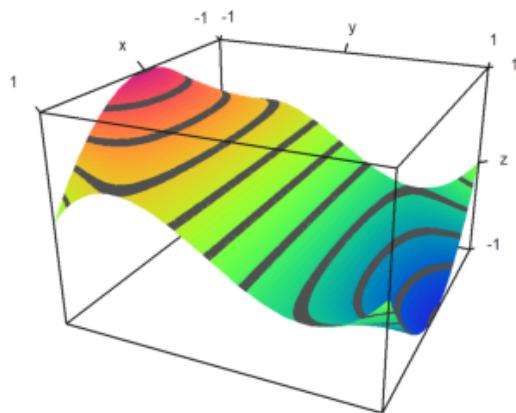
EMT juga dapat memplot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda membutuhkan red/cyan glasses.

```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°) :
```



Seringkali, skema warna spektral digunakan untuk plot. Ini menekankan ketinggian fungsinya.

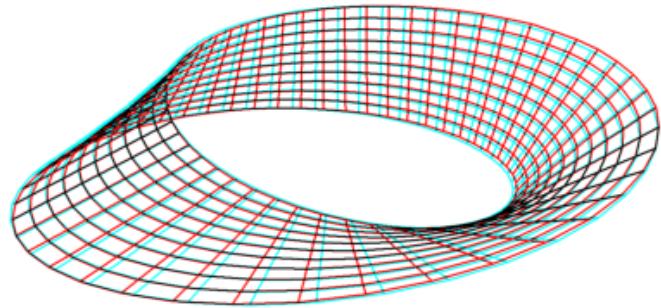
```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```



Euler juga dapat memplot permukaan berparameter, jika parameternya adalah nilai x, y, dan z dari gambar kisi persegi panjang di dalam ruang.

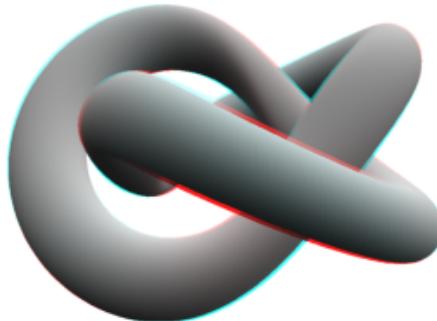
Untuk demo berikut, kami menyiapkan parameter u- dan v-, dan menghasilkan koordinat ruang dari ini.

```
>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 megah dengan red/cyan glasses.

```
>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

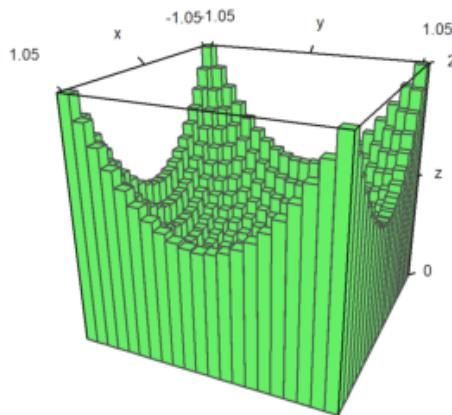
Plot batang juga dimungkinkan. Untuk ini, kami harus menyediakan

- x: vektor baris dengan  $n + 1$  elemen
- y: vektor kolom dengan  $n + 1$  elemen
- z: nxn matriks nilai.

z bisa lebih besar, tetapi hanya nilai nxn yang akan digunakan.

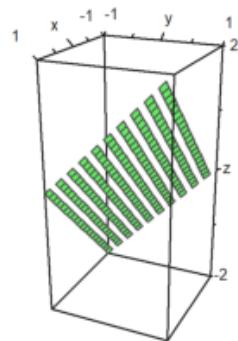
Dalam contoh, pertama-tama kita menghitung nilainya. Kemudian kita menyesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...
>plot3d(xa,ya,z,bar=true):
```



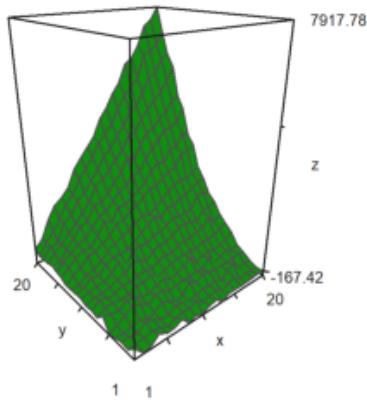
Dimungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20);
```

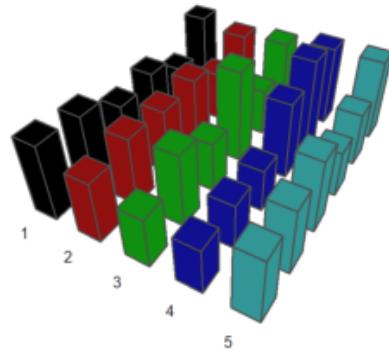


Jika memuat atau membuat matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan skala (M), atau menskalakan matriks dengan >zscale. Ini dapat dikombinasikan dengan faktor penskalaan individu yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8);
```



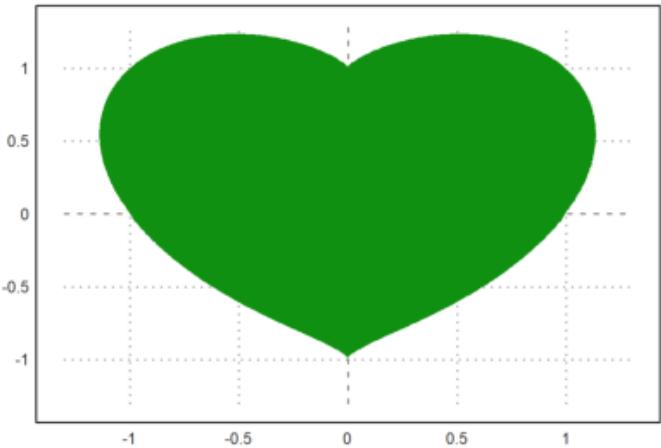
```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v',scols=1:5,ccols=[1:5]):
```



## Permukaan Benda Putar

---

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



```
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

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

Kami ingin memutar kurva jantung di sekitar sumbu y. Inilah ungkapan yang mendefinisikan hati:

$$f(x, y) = (x^2 + y^2 - 1)^3 - x^2 \cdot y^3.$$

Selanjutnya kita atur

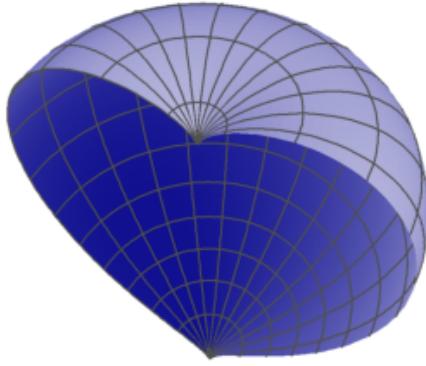
$$x = r \cdot \cos(a), \quad y = r \cdot \sin(a).$$

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2\sin a) r^5}{16}$$

Hal ini memungkinkan untuk menentukan fungsi numerik, yang menyelesaikan r, jika a diberikan. Dengan fungsi itu kita dapat memplot jantung yang berubah sebagai permukaan parametrik.

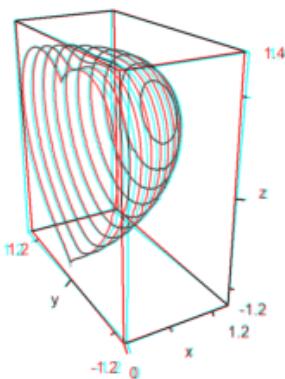
```
>function map f(a) := bisect("fr",0,2;a); ...
>t=linspace(-pi/2,pi/2,100); r=f(t); ...
>s=linspace(pi,2pi,100)'; ...
>plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>>hue,<frame,color=blue,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```



Berikut ini adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kami mendefinisikan fungsi, yang mendeskripsikan objek.

```
>function f(x,y,z) ...
r=x^2+y^2;
return (r+z^2-1)^3-r*z^3;
endfunction
```

```
>plot3d("f(x,y,z)", ...
>xmin=0, xmax=1.2, ymin=-1.2, ymax=1.2, zmin=-1.2, zmax=1.4, ...
>implicit=1, angle=-30°, zoom=2.5, n=[10, 60, 60], >anaglyph):
```



## Special 3D Plots

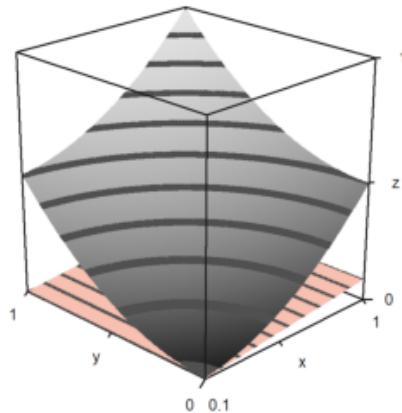
Fungsi `plot3d` bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda bisa mendapatkan plot berbingkai dari objek apa pun yang Anda suka.

Meskipun Euler bukan program 3D, Euler dapat menggabungkan beberapa objek dasar. Kami mencoba untuk memvisualisasikan paraboloid dan garis singgung-nya.

```
>function myplot ...
y=0:0.01:1; x=(0.1:0.01:1)';
plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ..
    hues=0.5,>contour,color=orange);
h=holding(1);
plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
holding(h);
endfunction
```

Sekarang framedplot () menyediakan bingkai, dan menyetel tampilan.

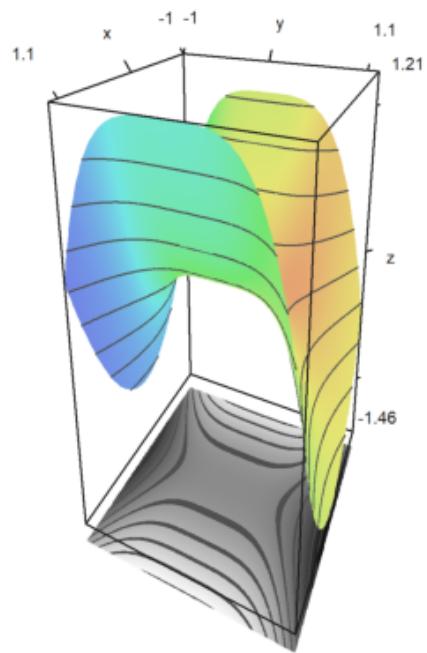
```
>framedplot("myplot", [0.1,1,0,1,0,1],angle=-45°, ...
> center=[0,0,-0.7],zoom=6):
```



Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d () menyetel jendela ke fullwindow () secara default, tetapi plotcontourplane () mengasumsikannya.

```
>x=-1:0.02:1.1; y=x'; z=x^2-y^4;
>function myplot (x,y,z) ...
zoom(2);
wi=fullwindow();
plotcontourplane(x,y,z,level="auto",<scale);
plot3d(x,y,z,>hue,<scale,>add,color=white,level="thin");
window(wi);
reset();
endfunction

>myplot(x,y,z);
```



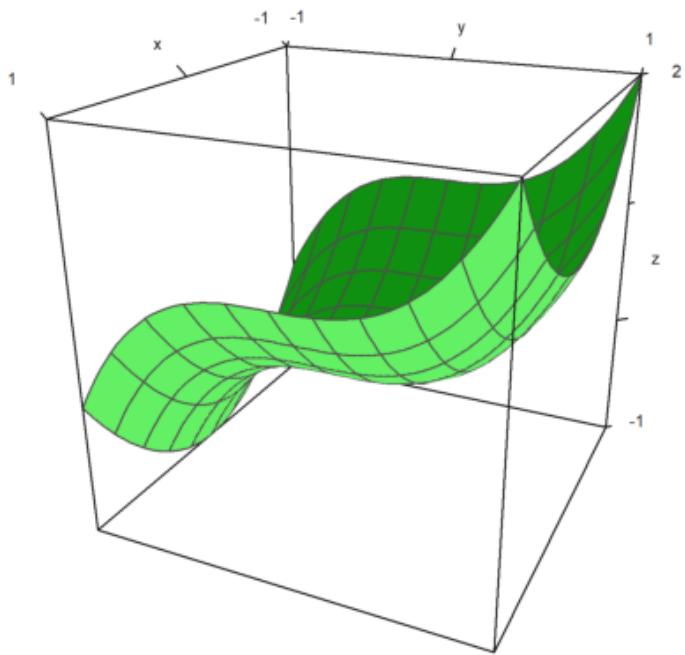
## Animasi

---

Salah satu fungsi yang memanfaatkan teknik ini adalah memutar. Itu dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi tersebut memanggil `addpage()` untuk setiap plot baru. Akhirnya itu menjawai plot.

Harap pelajari sumber rotasi 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.

Anda perlu menginstal Povray (32bit atau 64bit)dari <http://www.povray.org/>,

dan meletakkan sub-direktori "bin" Povray ke dalam jalur lingkungan, atau menyetel variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe".

Antarmuka Povray dari Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah current.pov, dan direktori 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 memiliki semangat yang sama dengan plot3d. Ini dapat menghasilkan grafik fungsi  $f(x, y)$ , atau permukaan dengan koordinat X, Y, Z dalam matriks, termasuk garis level opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat pemandangan ke dalam notebook Euler.

Selain pov3d (), ada banyak fungsi yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string, berisi kode Povray untuk objek. Untuk menggunakan fungsi ini, mulai file Povray dengan povstart (). Kemudian gunakan writeln (...) untuk menulis objek ke file adegan. Terakhir, akhiri file dengan povend (). Secara default, raytracer akan mulai, dan PNG akan dimasukkan ke dalam notebook Euler.

Fungsi objek memiliki parameter yang disebut "look", yang membutuhkan string dengan kode Povray untuk tekstur dan penyelesaian objek. Fungsi povlook () dapat digunakan untuk menghasilkan string ini. Ini memiliki parameter untuk warna, transparansi, Phong Shading dll.

Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk ke atas secara vertikal, sumbu a nd x, y, z di tangan kanan.

Anda perlu memuat file povray.

```
> load povray;
```

Pastikan, direktori bin Povray ada di jalurnya. Jika tidak, edit variabel berikut sehingga berisi path ke povray yang dapat dieksekusi.

```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Untuk pertamakali, kami memplot fungsi sederhana. Perintah berikut menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk menelusuri file ini.

Jika Anda memulai perintah berikut, GUI Povray akan terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya, apakah Anda ingin mengizinkan file exe dijalankan. Anda dapat menekan batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK di jendela Povray untuk mengetahui dialog start-up Povray.

```
>pov3d("x^2+y^2",zoom=3);
```

```
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;
```

Kita bisa membuat fungsinya transparan dan menambahkan hasil akhir lainnya. Kita juga bisa menambahkan garis level ke plot fungsi.

```
>pov3d("x^2+y^3",axiscolor=red,angle=20°, ...
>   look=povlook(blue,0.2),level=-1:0.5:1,zoom=3.8);
```

```
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;
```

Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan. Kami memplot himpunan titik di bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=1.5, ...
>   angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=45°,n=50, ...
>   <fscale,zoom=3.8);
```

```
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;

```

## Merencanakan dengan Koordinat

---

Alih-alih fungsi, kita bisa memplot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objeknya.

Dalam contoh ini, kita memutar fungsi di sekitar sumbu z.

```

>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,8)'; ...
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,height=20°,axis=0,zoom=4,light=[10,-5,5]);

```

```

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, kami memplot gelombang teredam. Kami menghasilkan gelombang dengan bahasa matriks Euler.

Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut. Perhatikan bahwa plot3d menskalakan plot, sehingga sesuai dengan kubus satuan

```

>r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ...
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3; ...
>pov3d(x,y,z,zoom=5,axis=0,add=povsphere([0,0,0.5],0.1,povlook(green)), ...
> w=500,h=300);

```

```

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;

```

Dengan metode naungan lanjutan Povray, sangat sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya di perbatasan dan dalam bayangan, triknya mungkin menjadi jelas.

Untuk ini, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>Z &= x^2*y^3
```

$$\begin{matrix} 2 & 3 \\ x & y \end{matrix}$$

Persamaan permukaannya adalah  $[x, y, Z]$ . Kami menghitung dua turunan menjadi  $x$  dan  $y$  dari ini dan mengambil produk silang sebagai normal.

```
>dx &= diff([x,y,z],x); dy &= diff([x,y,z],y);
```

Kami mendefinisikan normal sebagai produk silang dari turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

$$\begin{bmatrix} -2x^3y^2 \\ -3x^2y^2 \\ 1 \end{bmatrix}$$

We use only 25 points.

```
>x=-1:0.5:1; y=x';
>pov3d(x,y,Z(x,y),angle=10°, ...
> xv=Nx(x,y),yv=Ny(x,y),zv=Nz(x,y),<shadow>;
```

```
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;
```

Berikut ini adalah simpul Trefoil yang dilakukan oleh A. Busser di Povray. Ada versi perbaikannya dalam contoh.

See: Examples\Trefoil Knot | Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normal bagi kami. Pertama, tiga fungsi koordinat sebagai ekspresi simbolik.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
>Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
>Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunannya menjadi  $x$  dan  $y$ .

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang normal, yang merupakan produk persilangan dari dua turunannya.

```
>dn &= crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

Vektor normal adalah evaluasi dari ekspresi simbolik dn [i] untuk  $i = 1, 2, 3$ . Sintaks untuk ini adalah & "expresi" (parameter). Ini adalah alternatif metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),axis=0,zoom=5,w=450,h=350, ...
>    <shadow, look=povlook(gray), ...
>    xv=&"dn[1]"(x,y), yv=&"dn[2]"(x,y), zv=&"dn[3]"(x,y));
```

```
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;
```

Kami juga dapat membuat 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();
```

```
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);
```

Dengan povgrid (), 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();
```

```
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 Povray

---

Di atas, kami menggunakan pov3d untuk memplot permukaan. Antarmuka povray di Euler juga dapat menghasilkan objek Povray. Objek-objek ini disimpan sebagai string di Euler, dan perlu ditulis ke file Povray. Kami memulai output dengan povstart () .

```
>povstart(zoom=4);
```

Pertama kita tentukan tiga silinder, dan simpan dalam string di Euler.

Fungsi povx () dll. Hanya mengembalikan vektor [1,0,0], yang bisa digunakan sebagai gantinya.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...  
>c2=povcylinder(-povy,povy,1,povlook(green)); ...  
>c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

```
>c1
```

```
cylinder { <-1,0,0>, <1,0,0>, 1  
texture { pigment { color rgb <0.564706,0.0627451,0.0627451> } }  
finish { ambient 0.2 }  
}
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.

Itu dilakukan oleh povlook (), yang mengembalikan string dengan kode Povray yang relevan. Kita dapat menggunakan warna Euler default, atau menentukan warna kita sendiri. Kami juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> } }  
finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek interseksi, dan menulis hasilnya ke file.

```
>writeln(povintersection([c1,c2,c3]));
```

Perpotongan tiga silinder sulit untuk divisualisasikan, jika Anda 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 berikut menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan, bagaimana Euler menangani 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);

```

Perbedaan memungkinkan pemotongan satu objek dari yang lain. Seperti persimpangan, ada bagian dari objek CSG Povray.

```

>povstart(light=[5,-5,5],fade=10);

```

Untuk demonstrasi ini, kami mendefinisikan sebuah objek di Povray, daripada menggunakan string di Euler. Definisi segera ditulis ke file.

Koordinat kotak -1 berarti [-1, -1, -1].

```
>povdefine("mycube", povbox(-1,1));
```

Kita bisa menggunakan objek ini di povobject (), yang mengembalikan string seperti biasa.

```
>c1=povobject ("mycube", povlook(red));
```

Kami menghasilkan kubus kedua, dan memutar serta menskalakannya sedikit.

```
>c2=povobject ("mycube", povlook(yellow),translate=[1,1,1], ...
>    rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita ambil perbedaan kedua objek tersebut.

```
>writeln(povdifference(c1,c2));
```

```
Variable or function c1 not found.
Error in:
writeln(povdifference(c1,c2)); ...  
^
```

Sekarang tambahkan tiga sumbu.

```
>writeAxis(-1.2,1.2,axis=1); ...
>writeAxis(-1.2,1.2,axis=2); ...
>writeAxis(-1.2,1.2,axis=4); ...
>povend();
```

```
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 dapat memplot himpunan di mana  $f(x, y, z) = 0$ , seperti parameter implisit di plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsinya sedikit berbeda. Anda tidak dapat menggunakan keluaran ekspresi Maxima atau Euler.

```
>povstart(angle=70°,height=50°,zoom=4);
```

Buat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi tersebut.

```
>writeln(povsurface ("pow(x, 2)*y-pow(y, 3)-pow(z, 2)", povlook(green))); ...
>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);
```

## Objek Jaring

---

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.

Kami ingin memaksimalkan xy di bawah kondisi  $x + y = 1$  dan mendemonstrasikan sentuhan tangensial dari garis level.

```
>povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kami tidak dapat menyimpan objek dalam string seperti sebelumnya, karena terlalu besar. Jadi kami mendefinisikan objek dalam file Povray menggunakan declare. Fungsi povtriangle () melakukan ini secara otomatis. Ia dapat menerima vektor normal seperti pov3d () .

Yang berikut ini mendefinisikan objek mesh, dan langsung menulisnya ke dalam file.

```
>x=0:0.02:1; y=x'; z=x*y; vx=-y; vy=-x; vz=1;
>mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kami mendefinisikan dua disk, yang akan berpotongan dengan permukaan.

```
>cl=povdisc([0.5,0.5,0],[1,1,0],2); ...
>ll=povdisc([0,0,1/4],[0,0,1],2);
```

Tulis permukaan dikurangi dua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tuliskan dua persimpangan tersebut.

```
>writeln(povintersection([mesh,cl],povlook(red))); ...
>writeln(povintersection([mesh,ll],povlook(gray)));
```

Tulis titik maksimal.

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));
```

Tambahkan sumbu dan selesai.

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...
>povend();
```

```
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);
```

## Anaglyph di Povray

---

Untuk menghasilkan anaglyph untuk kacamata merah / cyan, Povray harus dijalankan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph().

Tentu saja, Anda memerlukan kaca mata merah / cyan untuk melihat contoh berikut dengan benar.

Fungsi pov3d() memiliki tombol 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 Anda membuat adegan dengan objek, Anda perlu memasukkan pembuatan adegan 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);
clk=povobject(cl,rotate=xrotate(90°));
cly=povobject(cl,rotate=yrotate(90°));
c=povbox([-1,-1,0],1);
un=povunion([cl,clk,cly,c]);
obj=povdifference(s,un,povlook(red));
writeln(obj);
writeAxes();
endfunction
```

Fungsi povanaglyph () melakukan semua ini. Parameternya seperti di povstart () dan povend () digabungkan.

```
>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 Euler berisi banyak objek. Tetapi Anda tidak dibatasi untuk ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau merupakan objek yang sama sekali baru.

Kami mendemonstrasikan torus. Perintah Povray untuk ini adalah "torus". Jadi kami mengembalikan string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat pada asalnya.

```
>function povdonat (r1,r2,look "") ...
```

```
    return "torus {" + r1 + "," + r2 + look + "}";
endfunction
```

Here is our first torus.

```
>t1=povdonat (0.8,0.2)
```

```
torus {0.8,0.2}
```

Mari kita gunakan objek ini untuk membuat torus kedua, diterjemahkan dan diputar.

```
>t2=povobject (t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
object { torus {0.8,0.2}
    rotate 90 *x
    translate <0.8,0,0>
}
```

Sekarang kami menempatkan objek ini ke dalam sebuah adegan. Untuk tampilan, kami menggunakan Phong Shading.

```
>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...
>writeln(povobject(t1,povlook(green,phong=1))); ...
>writeln(povobject(t2,povlook(green,phong=1))); ...
```

```
> povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, itu tidak menampilkan kesalahan. Karena itu Anda harus menggunakan

```
> povend (<exit);
```

jika ada yang tidak berhasil. Ini akan membuat jendela Povray terbuka.

```
>povend (h=320,w=480);
```

```
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);
```

Berikut adalah contoh yang lebih lengkap. Kami menyelesaikannya

$$Ax \leq b, \quad x \geq 0, \quad c \cdot x \rightarrow \text{Max.}$$

dan menunjukkan poin yang layak dan optimal dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
>b=[10,10,10,10]';
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini memiliki solusi.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0, 1, 0.5]
```

Yes, it has.

Next we define two objects. The first is the plane

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look=""') ...
```

```
    return povplane(a,b,look)
endfunction
```

Kemudian kami mendefinisikan perpotongan dari semua setengah spasi dan sebuah kubus.

```
>function adm (A, b, r, look "") ...
```

```
    ol=[];
    loop 1 to rows(A); ol=ol|oneplane(A[#],b[#]); end;
    ol=ol|povbox([0,0,0],[r,r,r]);
    return povintersection(ol,look);
endfunction
```

We can now plot the scene.

```
>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); ...
```

The following is a circle around the optimum.

-Terjemahan

Berikut ini adalah lingkaran di sekitar optimal.

```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
> povlook(red,0.9)));
```

Dan ada kesalahan di arah optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

```
>writeln(povtext("Linear Problem", [0,0.2,1.3],size=0.05,rotate=125°)); ...
>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);
```

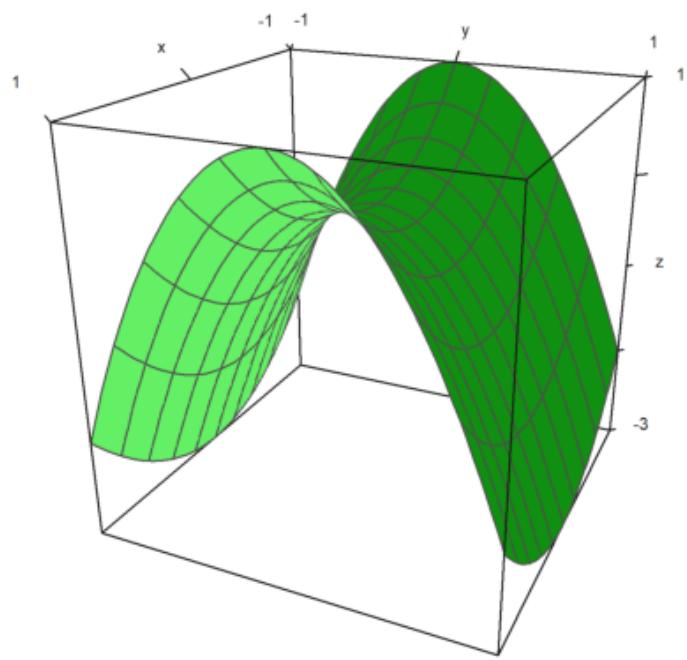
## Latihan Soal

---

1. Buatkan grafik

$$f(x) = x^2 - 3y^2$$

```
>plot3d("x^2-3*y^2"):
```

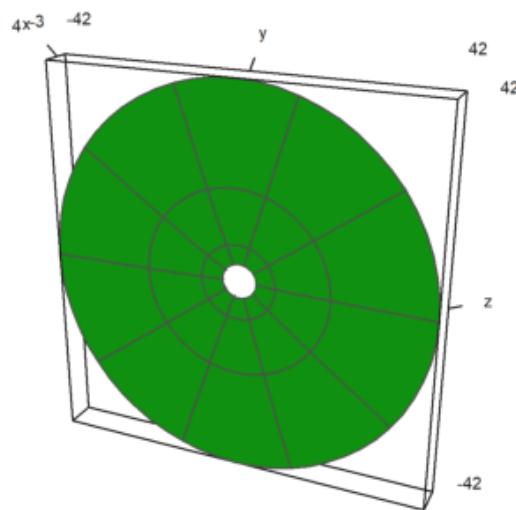


2. Buatkan grafik

$$f(x) = x^3 + 7x + 10$$

dengan a=-3, b=4, dan grid=10

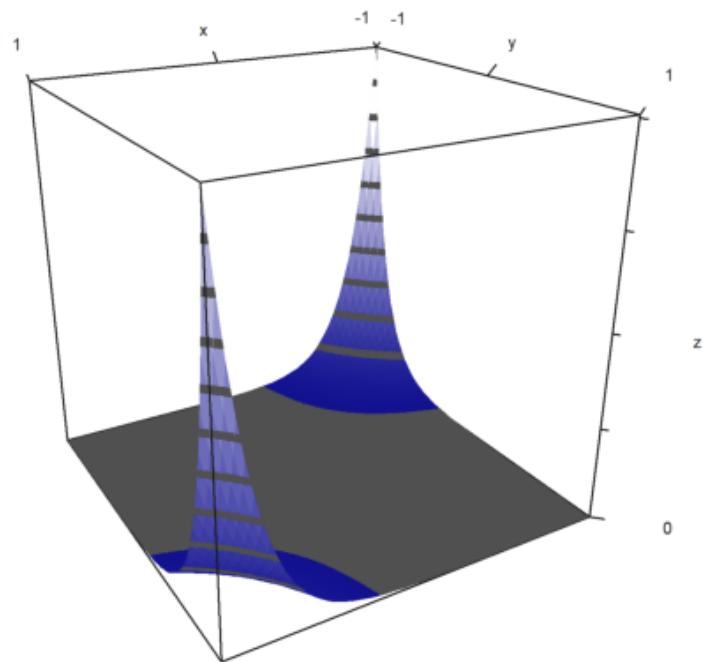
```
>plot3d("x^3-9*x+14", a=-3, b=4, rotate=true, grid=10):
```



3. Buatlah grafik plot kontur

$$f(x) = e^{6xy}$$

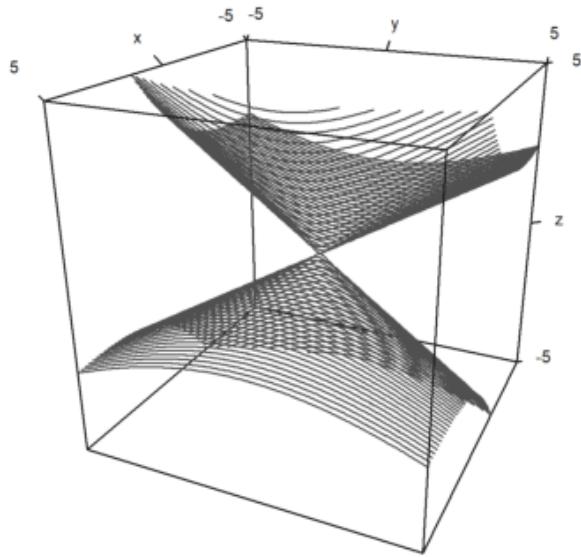
```
>plot3d("exp(6*x*y)", angle=150°, >contour, color=blue):
```



4. Buat grafik plot implisit

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

```
>plot3d("x^2+y^2+3*x*z-2*z^2",>implicit,r=5,zoom=2.2):
```



[a4paper,10pt]article eumat

Raden Mas Farrel Maheswara Kusuma Dewa  
22305141042  
Matematika E

## Menggambar Plot 3D dengan EMT

---

### 1. Menggambar Grafik Fungsi Dua Variabel \* dalam Bentuk Ekspresi

---

Langsung

Fungsi Dua Variabel didefinisikan sebagai sebuah fungsi bernilai real dari dua variabel real, yakni fungsi  $f$  yang memadankan setiap pasangan terurut  $(x,y)$  pada suatu himpunan  $D$  dari bidang dengan bilangan real tunggal  $f(x,y)$ .

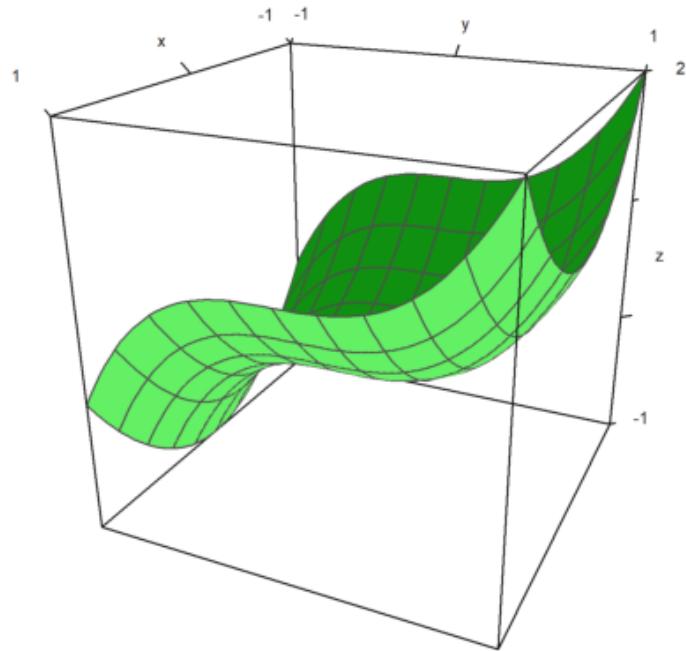
Di dalam program numerik EMT, ekspresi adalah string. Jika ditandai sebagai simbolis, mereka akan mencetak melalui Maxima, jika tidak melalui EMT. Ekspresi dalam string digunakan untuk membuat plot dan banyak fungsi numerik. Untuk ini, variabel dalam ekspresi harus "x" dan "y".

Untuk grafik suatu fungsi, gunakan

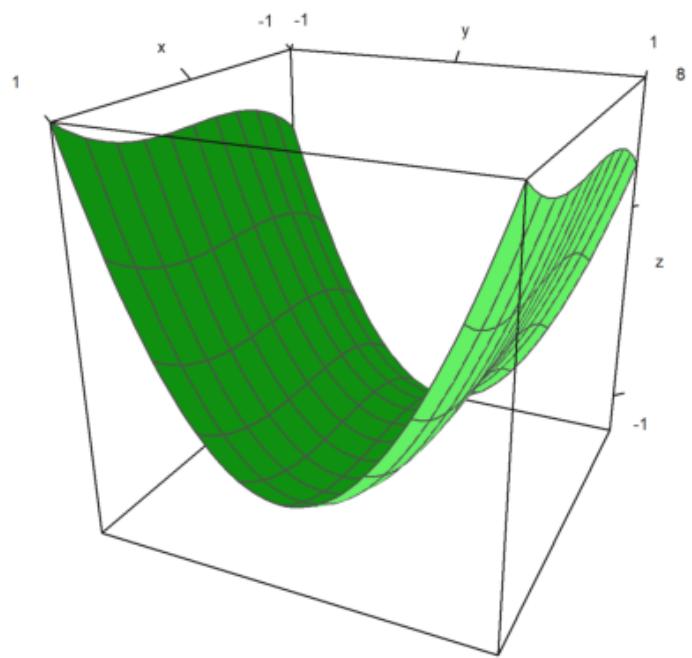
- ekspresi sederhana dalam  $x$  dan  $y$ ,
- nama fungsi dari dua variabel
- atau matriks data.

contoh:

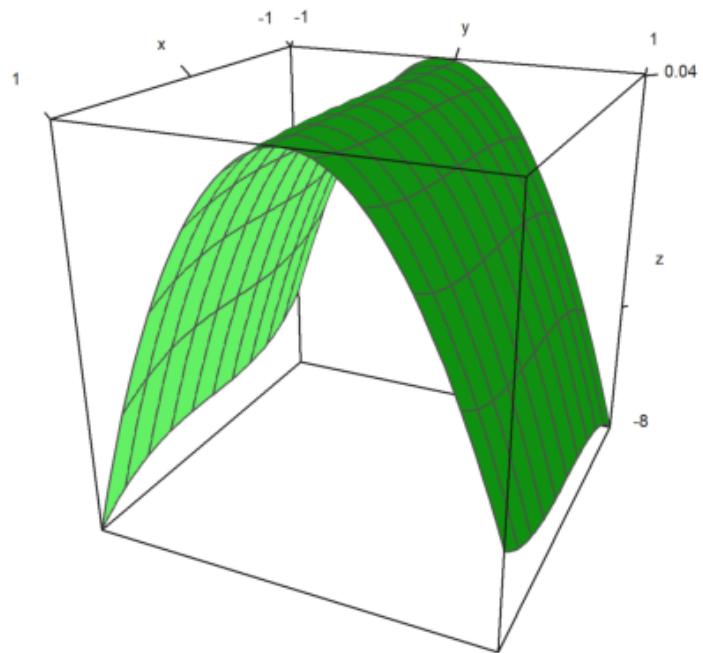
```
>plot3d("x^2+y^3"):
```



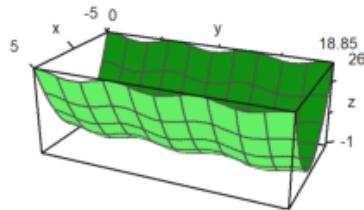
```
>plot3d("x^3+7*y^2"):
```



```
>plot3d("x^3*y-7*y^2"):
```



```
>aspect(1.5); plot3d("x^2+cos(y)",-5,5,0,6*pi):
```



1. aspect(1.5) mengatur aspek rasio pada grafik 3D.
2. plot3d("x^2+cos(y)",-5,5,0,6\*pi) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
3. -5,5 mengatur rentang sumbu x yang akan ditampilkan pada grafik.
4. 0,6\*pi mengatur rentang sumbu y yang akan ditampilkan pada grafik.

## Fungsi umum untuk plot 3D.

---

Fungsi plot3d (x, y, z, xmin, xmax, ymin, ymax, n, a, ..  
b, c, d, r, scale, fscale, frame, angle, height, zoom, distance, ..)

Rentang plot untuk fungsi dapat ditentukan dengan

- a,b: rentang x
- c,d: rentang y
- r : persegi simetris di sekitar (0,0).
- n : jumlah subinterval untuk plot.

Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

- fscale: menskalakan ke nilai fungsi (defaultnya adalah <fscale>).
- scale: angka atau vektor 1x2 untuk menskalakan ke arah x dan y.
- frame: jenis bingkai (default 1).

Tampilan dapat diubah dengan berbagai cara.

- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut terhadap sumbu y negatif dalam radian.
- height: ketinggian pandangan dalam radian.

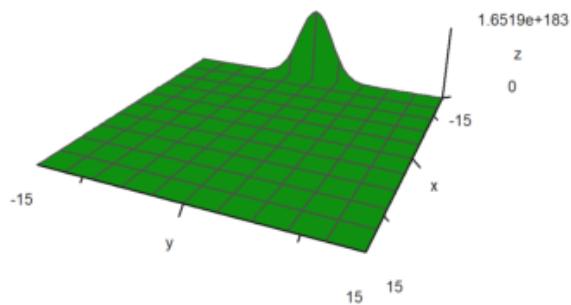
Nilai default dapat diperiksa atau diubah dengan fungsi view(). Ini mengembalikan parameter dalam urutan di atas.

```
>view
```

```
[5, 2.6, 2, 0.4]
```

Jarak yang lebih dekat membutuhkan lebih sedikit zoom. Efeknya lebih seperti lensa sudut lebar.  
contoh:

```
>plot3d("exp(-(x^3+y^2)/8)",r=15,n=60,fscale=4,scale=1.2,frame=3,>user):
```



1.  $\exp(-(\mathbf{x}^3+\mathbf{y}^2)/8)$  adalah fungsi matematika yang digunakan untuk membuat grafik 3D.

2.  $r=15$  mengatur jarak maksimum dari pusat grafik ke tepi grafik.

3.  $n=60$  mengatur jumlah titik yang digunakan untuk membuat grafik.

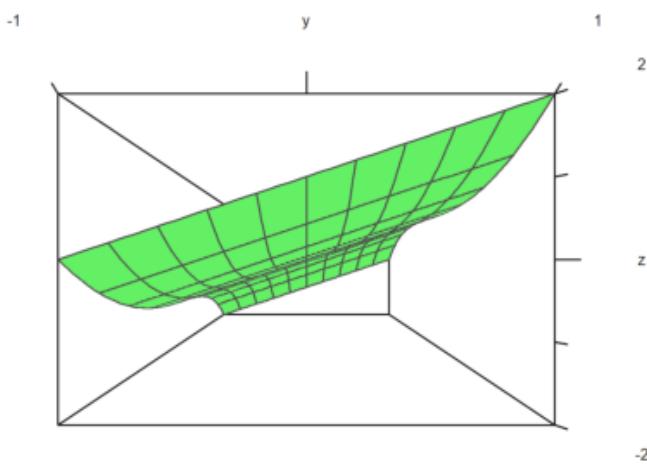
4.  $fscale=4$  mengatur faktor skala untuk warna.

5.  $scale=1.2$  mengatur faktor skala untuk ukuran grafik.

6.  $frame=3$  mengatur jenis bingkai yang digunakan untuk grafik.

Pada contoh berikut, sudut=0 dan tinggi=0 dilihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^3+y",distance=2,zoom=1,angle=pi/2,height=0):
```



1.  $x^3+y$  adalah fungsi matematika yang digunakan untuk membuat grafik 3D.

2.  $distance=2$  mengatur jarak pandang dari grafik.

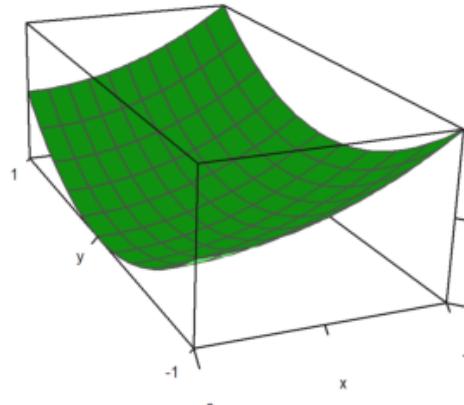
3.  $zoom=1$  mengatur faktor perbesaran grafik.

4.  $angle=pi/2$  mengatur sudut pandang grafik dalam radian.

5.  $height=0$  mengatur ketinggian pandangan dari grafik.

Plot selalu terlihat berada di tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter tengah.

```
>plot3d("x^2+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
> center=[0.4,0,0],zoom=5):
```

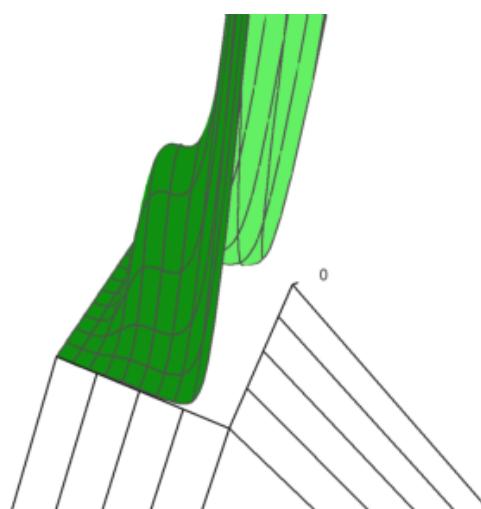


1.  $x^2+y^2$  adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2.  $a=0, b=1, c=-1, d=1$  mengatur rentang sumbu x dan y yang akan ditampilkan pada grafik.
3.  $\text{angle}=-20^\circ$  mengatur sudut pandang grafik dalam derajat.
4.  $\text{height}=20^\circ$  mengatur ketinggian pandangan dari grafik dalam derajat.
5.  $\text{center}=[0.4,0,0]$  mengatur pusat pandangan dari grafik.
6.  $\text{zoom}=5$  mengatur faktor perbesaran grafik.

Plotnya diskalakan agar sesuai dengan unit kubus untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung ukuran plot. Namun labelnya mengacu pada ukuran sebenarnya.

Jika Anda mematikannya dengan `scale=false`, Anda harus berhati-hati agar plot tetap masuk ke dalam jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan bagian tengah.

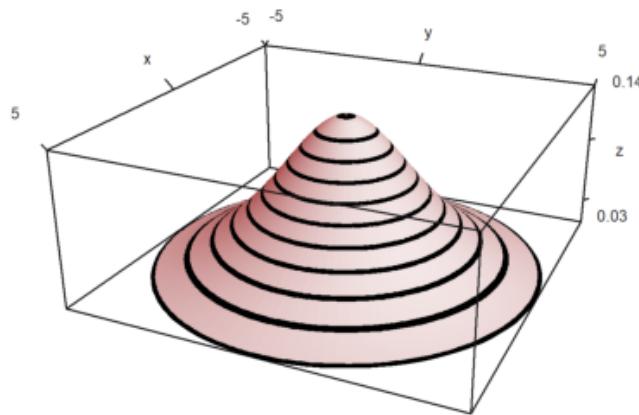
```
>plot3d("5*exp(-x^2+y^3)",r=2,<fscale,<scale,distance=13,height=50°, ...
> center=[0,0,-2],frame=3):
```



1.  $5 \cdot \exp(-x^2+y^3)$  adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2.  $r=2$  mengatur jarak maksimum dari pusat grafik ke tepi grafik.
3.  $<\text{fscale}$  mengatur faktor skala untuk warna.
4.  $<\text{scale}$  mengatur faktor skala untuk ukuran grafik.
5.  $\text{distance}=13$  mengatur jarak pandang dari grafik.
6.  $\text{height}=50^\circ$  mengatur ketinggian pandangan dari grafik dalam derajat.
7.  $\text{center}=[0,0,-2]$  mengatur pusat pandangan dari grafik.
8.  $\text{frame}=3$  mengatur jenis bingkai yang digunakan untuk grafik.

Plot kutub juga tersedia. Parameter  $\text{polar}=\text{true}$  menggambar plot kutub. Fungsi tersebut harus tetap merupakan fungsi dari  $x$  dan  $y$ . Parameter "fscale" menskalakan fungsi dengan skalanya sendiri. Kalau tidak, fungsinya akan diskalakan agar sesuai dengan kubus.

```
>plot3d("1/(x^2+y^2+7)", r=5, >polar, ...
>fscale=2, >hue, n=100, zoom=4, >contour, color=red) :
```

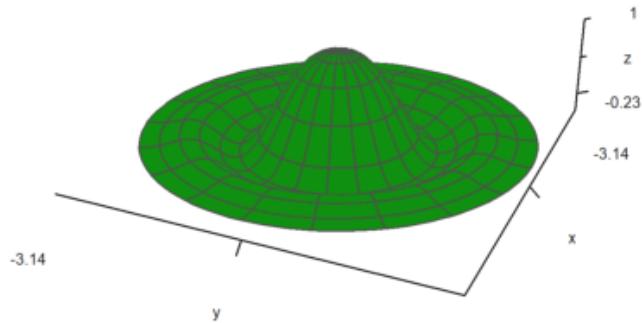


1.  $1/(x^2+y^2+7)$  adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2.  $r=5$  mengatur jarak maksimum dari pusat grafik ke tepi grafik.
3.  $\text{polar}$  mengatur tampilan grafik dalam koordinat polar.
4.  $\text{fscale}=2$  mengatur faktor skala untuk warna.
5.  $\text{hue}$  mengatur skala warna yang digunakan pada grafik.
6.  $n=100$  mengatur jumlah titik yang digunakan untuk membuat grafik.
7.  $\text{zoom}=4$  mengatur faktor perbesaran grafik.
8.  $\text{contour}$  mengatur tampilan garis kontur pada grafik.
9.  $\text{color}=red$  mengatur warna garis kontur pada grafik.

```
>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) :
```

```
Function plot3d needs at least one argument!
Use: plot3d (x {, y: none, z: none, xmin: none, xmax: none, ...})
Error in:
plot3d"f(x^2+y^2)", >polar, scale=[1,1,0.4], r=pi, frame=3, zoom=4) ...
^
```

```
>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) :
```

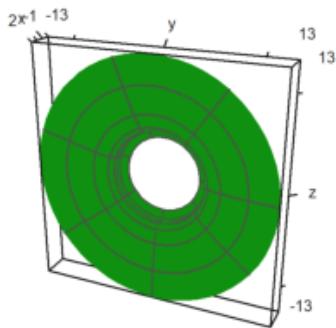


1. function  $f(r) := \exp(-r/2) * \cos(r)$  adalah fungsi matematika yang didefinisikan sebagai  $f(r) = e^{-(r/2)} * \cos(r)$ .
2. `plot3d("f(x^2+y^2)", polar, scale=[1,1,0.4], r=pi, frame=3, zoom=4)` adalah perintah untuk membuat grafik 3D dari fungsi  $f(x^2+y^2)$ .
3. polar mengatur tampilan grafik dalam koordinat polar.
4. scale=[1,1,0.4] mengatur faktor skala untuk ukuran grafik.
5.  $r=\pi$  mengatur jarak maksimum dari pusat grafik ke tepi grafik.
6. frame=3 mengatur jenis bingkai yang digunakan untuk grafik.
7. zoom=4 mengatur faktor perbesaran grafik.

Parameter memutar memutar fungsi di x di sekitar sumbu x.

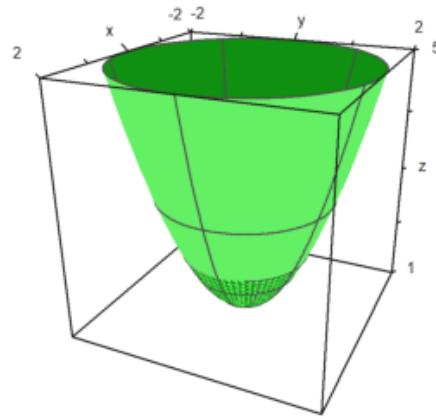
- rotate=1: Menggunakan sumbu x
- rotate=2: Menggunakan sumbu z

```
>plot3d("x^3+5", a=-1, b=2, rotate=true, grid=7):
```



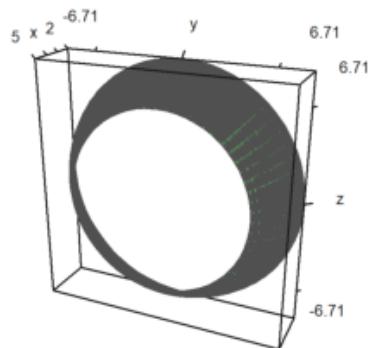
1.  $x^3+5$  adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2.  $a=-1, b=2$  mengatur rentang sumbu x yang akan ditampilkan pada grafik.
3. `rotate=true` mengatur grafik agar dapat diputar secara interaktif.
4. `grid=7` mengatur jumlah garis koordinat yang ditampilkan pada grafik.

```
>plot3d("x^2+1", a=-1, b=2, rotate=3, grid=5):
```



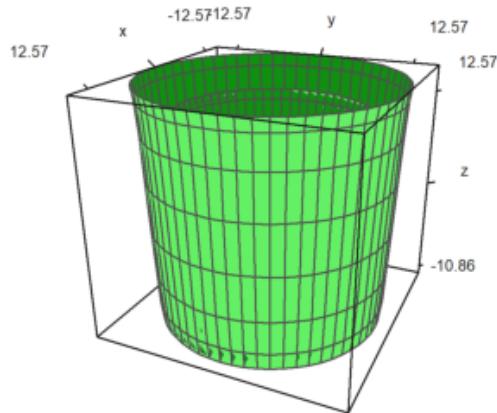
1.  $x^2+1$  adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2.  $a=-1, b=2$  mengatur rentang sumbu x yang akan ditampilkan pada grafik.
3.  $rotate=3$  mengatur grafik agar dapat diputar secara interaktif dengan menggunakan mouse.
4.  $grid=5$  mengatur jumlah garis koordinat yang ditampilkan pada grafik.

```
>plot3d("sqrt(49-x^2)", a=2, b=5, rotate=1) :
```



1.  $\sqrt{49-x^2}$  adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2.  $a=2, b=5$  mengatur rentang sumbu x yang akan ditampilkan pada grafik.
3.  $rotate=1$  mengatur grafik agar dapat diputar secara interaktif.

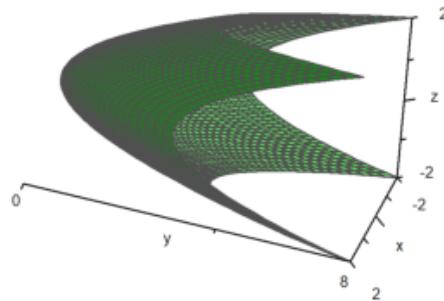
```
>plot3d("x*cos(2x)", a=0, b=4pi, rotate=2) :
```



1.  $x \cdot \cos(2x)$  adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
2.  $a=0, b=4\pi$  mengatur rentang sumbu x yang akan ditampilkan pada grafik.
3.  $\text{rotate}=2$  mengatur grafik agar dapat diputar secara interaktif.

Berikut adalah plot dengan tiga fungsi.

```
>plot3d("x", "x^2+y^2", "y", r=2, zoom=3.5, frame=3) :
```



1.  $x$  adalah fungsi matematika yang digunakan untuk menentukan nilai sumbu x pada grafik.
2.  $x^2+y^2$  adalah fungsi matematika yang digunakan untuk menentukan nilai sumbu z pada grafik.
3.  $y$  adalah fungsi matematika yang digunakan untuk menentukan nilai sumbu y pada grafik.
4.  $r=2$  mengatur jarak maksimum dari pusat grafik ke tepi grafik.
5.  $\text{zoom}=3.5$  mengatur faktor perbesaran grafik.
6.  $\text{frame}=3$  mengatur jenis bingkai yang digunakan untuk grafik.

## **2. Menggambar Grafik Fungsi Dua Variabel yang**

---

\* Rumusnya Disimpan dalam Variabel Ekspresi

Fungsi ini dapat memplot plot 3D dengan grafik fungsi dua variabel, permukaan berparameter, kurva ruang, awan titik, penyelesaian persamaan tiga variabel. Semua plot 3D bisa ditampilkan sebagai anaglyph.

fungsi `plot3d (x, y, z, xmin, xmax, ymin, ymax, n, a)`

## Parameter

- x : ekspresi dalam x dan y
- x,y,z : matriks koordinat suatu permukaan
- x,y,z : ekspresi dalam x dan y untuk permukaan parametrik
- x,y,z : ekspresi dalam x untuk memplot kurva ruang
- xmin,xmax,ymin,ymax :

x, y batas ekspresi

contoh:

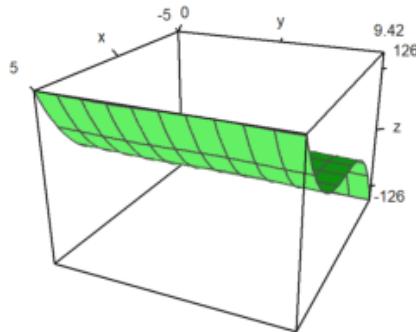
ekspresi dalam string

```
>expr := "x^3+cos(y)"
```

$x^3+\cos(y)$

plot ekspresi

```
>plot3d(expr, -5, 5, 0, 3*pi) :
```



1.  $x^3+\cos(y)$  adalah fungsi matematika yang digunakan untuk membuat grafik 3D.

2. -5,5 mengatur rentang sumbu x yang akan ditampilkan pada grafik.

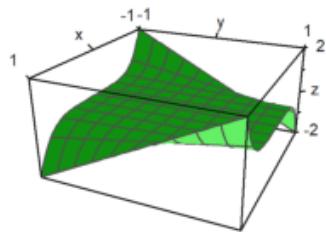
3. 0,3\*pi mengatur rentang sumbu y yang akan ditampilkan pada grafik.

contoh 1:

```
>  
>expr := "2*x^5*y"
```

$2*x^5*y$

```
>aspect(2); plot3d(expr) :
```

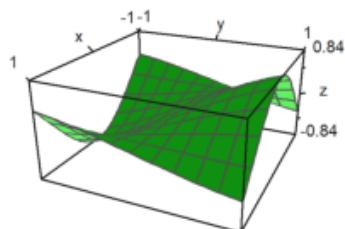


1. aspect(2) mengatur aspek rasio pada grafik 3D.
  2. plot3d(expr) adalah fungsi matematika yang digunakan untuk membuat grafik 3D.
- contoh 2:

```
>expr := "cos(2x)*sin(y)"
```

$\cos(2x)*\sin(y)$

```
>plot3d(expr):
```

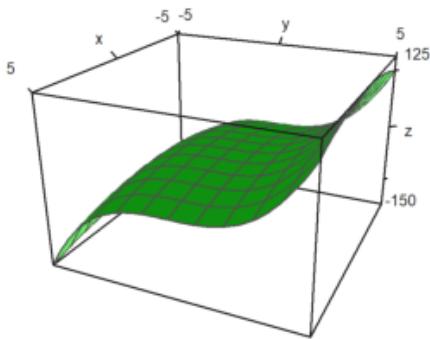


- contoh 3:

```
>expr := "y^3-x^2"
```

$y^3-x^2$

```
>aspect(1.5); plot3d(expr, -5, 5, -5, 5):
```



### 3. Menggambar Grafik Fungsi Dua Variabel yang \* Fungsinya

---

Didefinisikan sebagai Fungsi Numerik

#### Fungsi Dua Variabel

---

Fungsi dua variabel adalah sebuah fungsi yang bernilai real dari dua variabel real. Fungsi ini memadankan setiap pasangan terurut  $(x,y)$  pada suatu himpunan  $D$  dari bidang dengan bilangan real tunggal  $f(x,y)$ . Dalam matematika, fungsi dua variabel atau lebih digunakan untuk menggambarkan hubungan antara dua atau lebih variabel.

#### Fungsi Numerik

---

Fungsi numerik adalah suatu fungsi matematika yang menghasilkan nilai numerik sebagai output-nya. Fungsi ini dapat dinyatakan dalam bentuk persamaan matematika atau algoritma komputasi.

Contoh:

Fungsi

$$f(x, y) = 5x + y$$

Misal input nilai  $x=2$  dan  $y=3$ , maka akan dihasilkan nilai  $z$  yaitu

$$z = f(x, y) = 5(2) + 3 = 10 + 3 = 13$$

#### Gambar Grafik Fungsi

---

Fungsi satu baris numerik didefinisikan oleh "`:`".

Langkah-langkah untuk memvisualisasikan grafik fungsi dua variabel yang fungsi nya didefinisikan sebagai fungsi numerik dalam `plot3d`:

1. Buat fungsi numerik yang akan digunakan untuk memvisualisasikan data.

`function f(x,y):=ax+by`

dimana  $a$  dan  $b$  adalah konstanta

2. Gunakan fungsi `plot3d()` untuk membuat grafik tiga dimensi dari fungsi numerik.

`plot3d("f")`:

#### Contoh

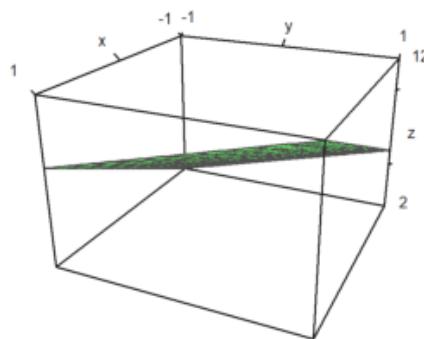
---

Fungsi matematika  $f(x,y)$  dapat digambarkan dalam bentuk grafik tiga dimensi menggunakan perintah plot3d. Berikut adalah contoh penggunaan perintah plot3d untuk menggambarkan fungsi tersebut:

### 1. Fungsi Linear Dua Variabel

$$f(x, y) = 3x + 2y + 7$$

```
>function f(x,y) := 3*x+2*y+7  
>plot3d("f") :
```



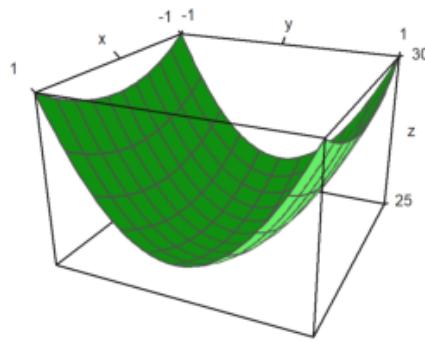
- Fungsi  $f(x,y)$  didefinisikan sebagai  $3x+2y+7$ .
- Perintah "plot3d("f")" digunakan untuk memplot grafik 3D dari fungsi  $f(x,y)$  menggunakan fungsi plot3d di EMT.
- Grafik yang dihasilkan akan menampilkan fungsi dalam tiga dimensi, dengan sumbu x dan y mewakili variabel masukan dan sumbu z mewakili nilai keluaran fungsi. Grafik akan menunjukkan bentuk fungsi dan perubahannya seiring dengan perubahan variabel masukan.

---

### 2. Fungsi Kuadrat Dua Variabel

$$f(x, y) = x^2 + 2y^2 + 25$$

```
>function f(x,y) := x^2+(2*y)^2+25  
>plot3d("f") :
```



- Perintah "function f(x,y):= x^2+(2\*y)^2+25" berarti mendefinisikan fungsi matematika  $f(x,y)$  sebagai  $x$  pangkat 2 ditambah 3 kali  $y$  pangkat 2 ditambah 25.

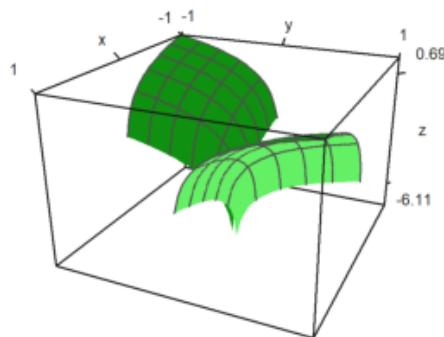
- Perintah "plot3d("f")" berarti membuat grafik tiga dimensi dari fungsi  $f(x,y)$  yang telah didefinisikan sebelumnya.

---

### 3. Fungsi Logaritma Dua Variabel

$$f(x, y) = \log(3xy)$$

```
>function f(x,y) := log((2*x)*y)
>plot3d("f"):
```



- Input yang diberikan adalah fungsi matematika dua variabel,  $f(x,y)$ , yang didefinisikan sebagai logaritma hasil kali  $2x$  dan  $y$ .

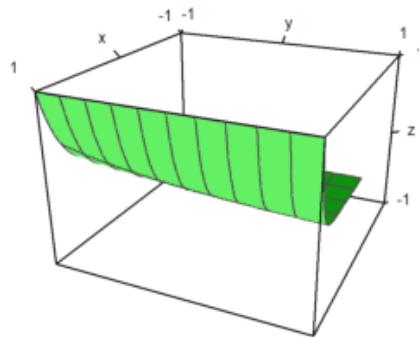
- Perintah "plot3d("f")" digunakan untuk memplot grafik fungsi  $f(x,y)$  dalam ruang tiga dimensi.

---

#### 4. Fungsi Eksponen Dua Variabel

$$f(x, y) = x^{2y-15}$$

```
>function f(x,y):= x^(5*y+10)  
>plot3d("f"):
```



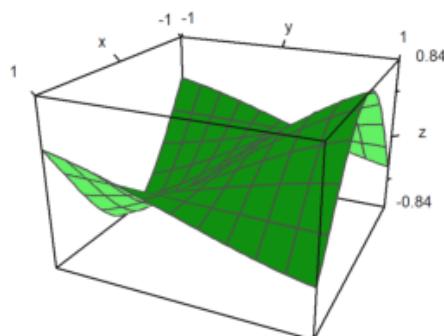
- Perintah 'fungsi f(x,y):= x^(2y-15)' adalah fungsi matematika dua variabel 'x' dan 'y' dan dengan rumus  $x^{(2y-15)}$
- Perintah 'plot3d("f")' digunakan untuk memplot fungsi dalam ruang tiga dimensi. Plot yang dihasilkan akan menampilkan nilai fungsi sebagai permukaan pada bidang x-y, dengan tinggi permukaan mewakili nilai fungsi pada titik tersebut.

---

#### 5. Fungsi Trigonometri Dua Variabel

$$f(x, y) = \cos(2x) + \sin(y)$$

```
>function f(x,y):= cos(2*x)*sin(y)  
>plot3d("f"):
```



- Perintah "function f(x,y):= cos(2x)\*sin(y)" adalah perintah untuk mendefinisikan fungsi matematika  $f(x,y)$  yang menghasilkan nilai cosinus dari  $x$  dikalikan dengan sinus dari  $y$ .
- Perintah "plot3d("f")" adalah perintah untuk membuat grafik tiga dimensi dari fungsi  $f(x,y)$  yang telah didefinisikan sebelumnya.

## 4. Menggambar Grafik Fungsi Dua Variabel yang \* Fungsinya

---

Didefinisikan sebagai Fungsi Simbolik

### Fungsi Simbolik

---

Fungsi simbolik adalah fungsi yang dinyatakan dengan menggunakan simbol-simbol matematika, seperti huruf dan operasi matematika, daripada menggunakan angka konkret atau ekspresi numerik. Fungsi simbolik sering digunakan untuk menggambarkan hubungan antara variabel-variabel matematika dalam bentuk yang lebih umum dan abstrak.

Contoh fungsi simbolik yang umum adalah:

$$g(x, y) = 2x + y$$

Dalam contoh di atas,  $g(x)$  adalah fungsi simbolik yang mengaitkan setiap nilai  $x$  dengan hasil dari ekspresi matematika  $2x + 3$ . Fungsi ini dapat digunakan untuk menghitung nilai fungsi untuk berbagai nilai  $x$ .

### Perbedaan Fungsi Numerik dan Fungsi Simbolik

---

#### 1. Fungsi Numerik

Fungsi numerik dinyatakan dalam bentuk yang lebih konkret menggunakan angka-angka nyata.

Contoh fungsi numerik adalah

$$g(x, y) = 2x + y + 3$$

dimana kita memberikan nilai numerik kepada "x dan y"

misalnya,  $x = 5$  dan  $y = 2$ , maka hasilnya adalah angka konkret yaitu  $g(5,2) = 15$

#### 2. Fungsi Simbolik

Fungsi simbolik dinyatakan menggunakan simbol-simbol matematika seperti huruf (variabel) dan operasi matematika.

Contoh fungsi simbolik adalah

$$g(x, y) = 2x + y + 3$$

" $g$ " adalah simbol fungsi

" $x, y$ " adalah variabel,

$2x + 3$  adalah ekspresi matematika yang menggambarkan hubungan antara " $x, y$ " dan hasil fungsi.

### Gambar Grafik Fungsi

---

Fungsi satu baris simbolik didefinisikan oleh "&=".

Langkah-langkah untuk memvisualisasikan grafik fungsi dua variabel yang fungsi nya didefinisikan sebagai fungsi simbolik dalam plot3d:

1. Buat fungsi simbolik yang akan digunakan untuk memvisualisasikan data.

function  $g(x,y):= ax+by;$

dimana  $a$  dan  $b$  adalah konstanta

2. Gunakan fungsi plot3d() untuk membuat grafik tiga dimensi dari fungsi numerik.

plot3d("g"):

3. Menentukan rentang variabel

misal

plot3d("g(x,y)",-10,10,-5,5):

dengan batasan x dari -10 hingga 10 dan batasan y dari -5 hingga 5

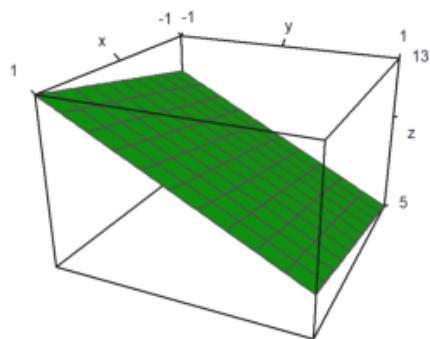
## Contoh

---

### 1. Fungsi Linear Dua Variabel

$$g(x, y) = x - 3y + 9$$

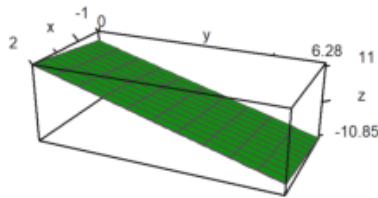
```
>function g(x,y) &= x-3*y+9;  
>plot3d("g(x,y)":
```



- Fungsi  $g(x,y)$  adalah fungsi matematika yang mengambil dua variabel,  $x$  dan  $y$ , dan menghasilkan sebuah nilai berdasarkan rumus  $x - 3y + 9$ .

- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,2,0,2*pi):
```

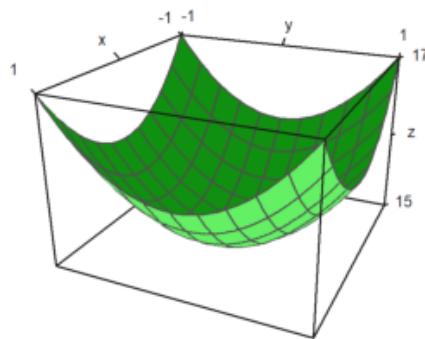


- Perintah "plot3d("g(x,y)",-1,2,0,2\*pi)" adalah perintah untuk menggambar grafik fungsi tiga dimensi "g(x,y)" pada rentang x dari -1 hingga 2 dan rentang y dari 0 hingga 2pi.
- 

## 2. Fungsi Kuadrat Dua Variabel

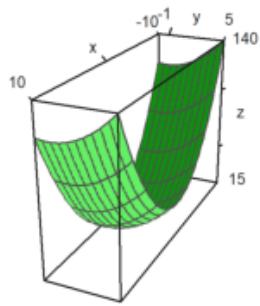
$$g(x, y) = x^2 + y^2 + 15$$

```
>function g(x,y) &= x^2+y^2+15;
>plot3d("g(x,y)":
```



- Fungsi  $g(x,y)$  adalah fungsi matematika yang mengambil dua variabel,  $x$  dan  $y$ , dan menghasilkan sebuah nilai berdasarkan rumus  $x^2+y^2+15$
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-10,10,-1,5):
```

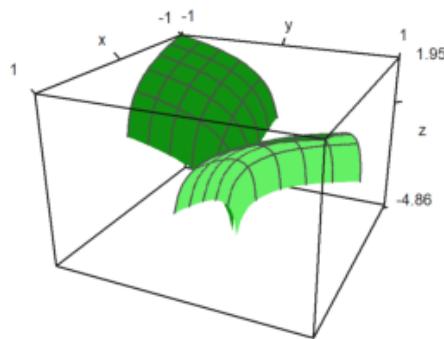


- Perintah "plot3d("g(x,y)",-10,10,-1,5)" adalah perintah untuk menggambar grafik fungsi tiga dimensi g(x,y) pada rentang x dari -10 hingga 10 dan rentang y dari -1 hingga 5
- 

### 3. Fungsi Logaritma Dua Variabel

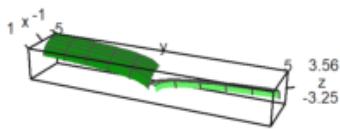
$$g(x, y) = \log(xy^7)$$

```
>function g(x,y) &= log(x*y^7);
>plot3d("g(x,y)":
```



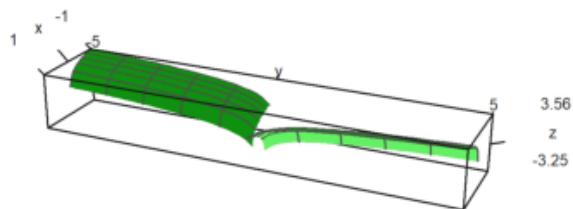
- Fungsi g(x,y) adalah fungsi matematika yang mengambil dua variabel, x dan y, dan menghasilkan sebuah nilai berdasarkan rumus logaritma x dikalikan y dikalikan 7
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,1,-5,5):
```



- Perintah "plot3d("g(x,y)",-1,1,-5,5)" adalah perintah untuk menggambar grafik fungsi tiga dimensi  $g(x,y)$  pada rentang  $x$  dari -1 hingga 1 dan rentang  $y$  dari -5 hingga 5

```
>plot3d("g(x,y)",-1,1,-5,5,zoom=4.5):
```



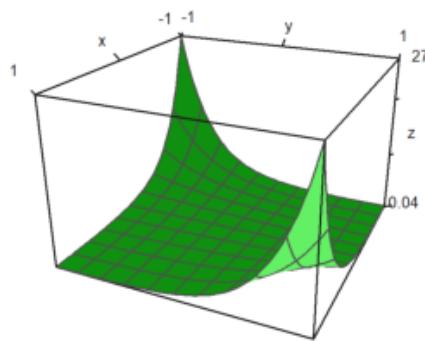
- plot3d: perintah untuk membuat grafik 3D.  
 - "g(x,y)": fungsi matematika yang akan digunakan untuk membuat grafik.  
 - -1,1: rentang nilai variabel  $x$  yang akan digunakan dalam grafik.  
 - -5,5: rentang nilai variabel  $y$  yang akan digunakan dalam grafik.  
 - zoom=4.5: perintah untuk memperbesar tampilan grafik dengan faktor 4.5.

---

#### 4. Fungsi Eksponen Dua Variabel

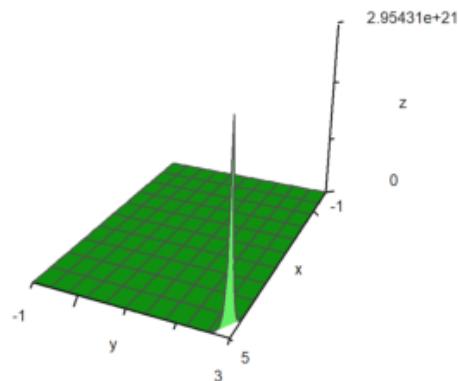
$$g(x, y) = 3^{xy^3}$$

```
>function g(x,y) &= 3^(x*y^3);  
>plot3d("g(x,y)":
```



- Fungsi  $g(x,y)$  adalah fungsi matematika yang mengambil dua variabel,  $x$  dan  $y$ , dan menghasilkan sebuah nilai berdasarkan rumus  $3^{(xy^3)}$
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,5,-1,3,frame=3,zoom=3):
```

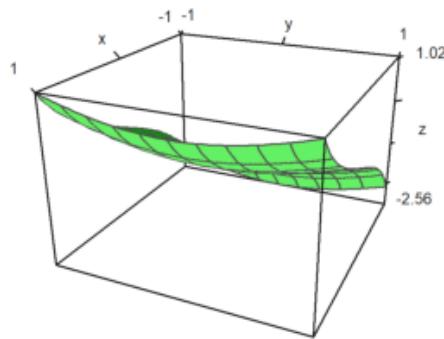


- Perintah `plot3d("g(x,y)",-1,5,-1,3,frame=3,zoom=3)` adalah perintah untuk membuat plot tiga dimensi dari fungsi ' $g(x,y)$ ' dengan batas ' $x$ ' dari '-1' hingga '5' dan batas ' $y$ ' dari '-1' hingga '3'.
- `plot3d`: perintah untuk membuat plot tiga dimensi.
- " $g(x,y)$ ": fungsi yang akan diplot.
- $(-1,5)$ : batas ' $x$ ' dari '-1' hingga '5'.
- $(-1,3)$ : batas ' $y$ ' dari '-1' hingga '3'.
- `frame=3`: menampilkan frame nomor 3.
- `zoom=3`: memperbesar tampilan plot sebanyak 3 kali.

## 5. Fungsi Trigonometri Dua Variabel

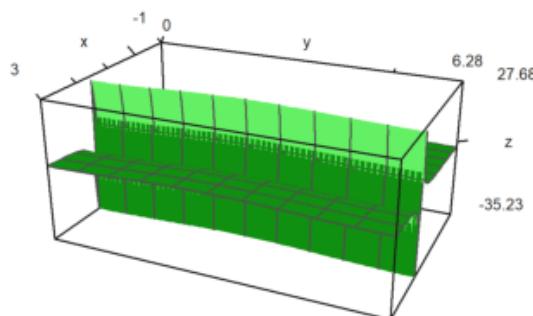
$$g(x, y) = \tan(x) - \cot(y)$$

```
>function g(x,y) &= tan(x)-cos(y);
>plot3d("g(x,y)":
```



- Fungsi  $g(x,y)$  adalah fungsi matematika yang mengambil dua variabel,  $x$  dan  $y$ , dan menghasilkan sebuah nilai berdasarkan rumus  $\tan(x)-\cos(y)$
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,3,0,2*pi,frame=1,zoom=3.5):
```

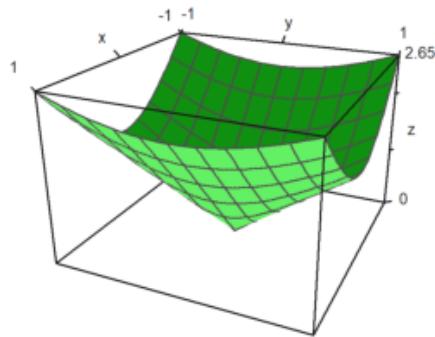


- Perintah `plot3d("g(x,y)",-1,3,0,2*pi,frame=5,zoom=3)` adalah perintah untuk membuat plot tiga dimensi dari fungsi ' $g(x,y)$ ' dengan batas 'x' dari '-1' hingga '3' dan batas 'y' dari '0' hingga '2pi'.
- `plot3d`: perintah untuk membuat plot tiga dimensi.
- " $g(x,y)$ ": fungsi yang akan diplot.
- $(-1,3)$ : batas 'x' dari '-1' hingga '3'.
- $(0,2\pi)$ : batas 'y' dari '0' hingga '2pi'.
- `frame=1`: menampilkan frame nomor 1.
- `zoom=3.5`: memperbesar tampilan plot sebanyak 3.5 kali.

## 6. Fungsi Akar Kuadrat

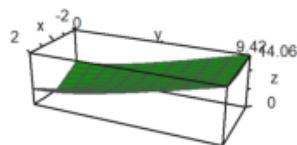
$$P(x, y) = \sqrt{5x^2 + 2y^2}$$

```
>function P(x,y) &= sqrt(5*x^2+2*y^2);  
>plot3d("P(x,y)":
```



- Fungsi  $P(x,y)$  adalah fungsi matematika yang mengambil dua variabel,  $x$  dan  $y$ , dan menghasilkan sebuah nilai berdasarkan rumus akar kuadrat dari  $5x^2+2y^2$
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("P(x,y)", -2, 2, 0, 3*pi, frame=5, zoom=2, scale=1):
```



- $P(x,y)$ : Merupakan fungsi yang akan digambarkan dalam grafik tiga dimensi.
- (-2,2): Merupakan rentang nilai dari sumbu  $x$  yang akan digunakan dalam grafik.
- (0,3pi): Merupakan rentang nilai dari sumbu  $y$  yang akan digunakan dalam grafik. Nilai pi dikalikan dengan 3 agar rentang nilai  $y$  mencakup tiga putaran lingkaran penuh.
- frame=5: Menentukan nomor bingkai (frame) yang akan digunakan dalam animasi grafik.
- zoom=2: Menentukan faktor pembesaran grafik. Dengan memperbesar tampilan, kita dapat melihat detail yang lebih kecil pada plot.

- scale=1: Menentukan skala grafik. Dengan mengatur skala, kita dapat mengubah jarak antara titik-titik pada sumbu tersebut. **Menggambar Data  $x, y, z$  pada ruang Tiga Dimensi (3D)**

---

### Definisi

Menggambar data pada ruang tiga dimensi (3D) adalah proses visualisasi data yang mengubah informasi dalam tiga dimensi, yaitu panjang, lebar, dan tinggi, menjadi representasi visual yang dapat dipahami dan dianalisis.

### Tujuan:

Tujuan dari menggambar data 3D adalah untuk membantu pemahaman dan interpretasi data yang lebih baik, terutama ketika data tersebut memiliki komponen yang tidak dapat direpresentasikan dengan baik dalam dua dimensi.

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu menyediakan matriks nilai  $x$ -,  $y$ - dan  $z$ , atau tiga fungsi atau ekspresi  $f_x(x,y)$ ,  $f_y(x,y)$ ,  $f_z(x,y)$ .

$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

Karena  $x, y, z$  adalah matriks, kita asumsikan bahwa  $(t, s)$  melalui sebuah kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di ruang angkasa.

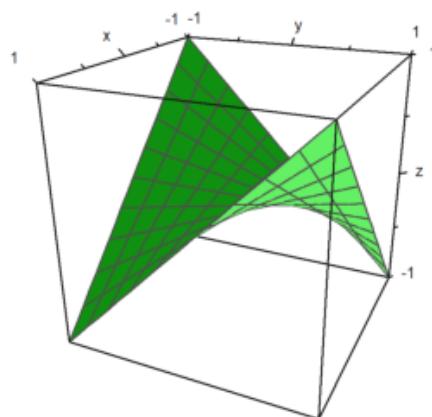
Kita dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

Dalam contoh berikut, kami menggunakan vektor nilai  $t$  dan vektor kolom nilai  $s$  untuk membuat parameter permukaan bola. Dalam gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

### Contoh 1

---

```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



Baris pertama kode "t=-1:0.1:1" membuat vektor baris  $t$  yang berisi nilai dari -1 hingga 1 dengan interval 0.1. Baris kedua "s=(-1:0.1:1)'" membuat vektor kolom  $s$  yang berisi nilai dari -1 hingga 1 dengan interval 0.1. Operator transpose ' digunakan untuk mengubah vektor baris  $t$  menjadi vektor kolom.

Baris ketiga "plot3d(t,s,t\*s,grid=10)" membuat plot tiga dimensi dari fungsi  $f(x,y) = xy$  pada domain  $[-1,1] \times [-1,1]$ . Plot dibuat menggunakan fungsi plot3d, yang mengambil tiga argumen: koordinat  $x$ ,  $y$ , dan  $z$  dari titik-titik yang akan diplot. Dalam hal ini, koordinat  $x$  diberikan oleh vektor  $t$ , koordinat  $y$  diberikan oleh vektor

s, dan koordinat z diberikan oleh hasil perkalian t dan s, yaitu ts. Parameter grid diatur menjadi 10, yang menunjukkan jumlah garis grid yang akan ditampilkan pada plot.

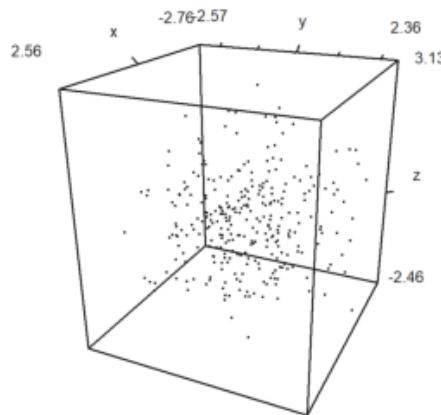
## Contoh 2

---

Tentu saja, titik cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor untuk koordinat titik-titik tersebut.

Gayanya sama seperti di plot2d dengan points=true;

```
>n=300;...
>plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```



Kode "n=300; plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=".")" digunakan untuk membuat plot tiga dimensi dari tiga vektor normal yang dihasilkan secara acak dengan menggunakan fungsi "normal()" pada Euler Math Toolbox (EMT). Parameter "n=300" menunjukkan bahwa setiap vektor normal memiliki 300 elemen. Parameter "points=true" digunakan untuk menampilkan titik-titik pada plot, sedangkan parameter "style='.'" digunakan untuk mengatur gaya titik pada plot menjadi titik bulat.

## Contoh 3

---

Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

Dalam contoh berikut, kita membuat tampilan bayangan dari bola yang terdistorsi. Koordinat biasa untuk bola adalah

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

dengan

$$0 \leq t \leq 2\pi, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2}.$$

Kami mendistorsi ini dengan sebuah 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):
```



Kode ini terdiri dari beberapa baris. Baris pertama "t=linspace(0,2pi,320)" membuat vektor t yang berisi 320 nilai yang sama terdistribusi secara merata antara 0 dan 2p. Baris kedua "s=linspace(-pi/2,pi/2,160)'" membuat vektor s yang berisi 160 nilai yang sama terdistribusi secara merata antara -p/2 dan p/2. Operator transpose ' digunakan untuk mengubah vektor baris s menjadi vektor kolom.

Baris ketiga "d=1+0.2\*(cos(4t)+cos(8s))" membuat vektor d yang berisi nilai dari  $1 + 0.2 * (\cos(4t) + \cos(8s))$ . Baris keempat "plot3d(cos(t)\*cos(s)\*d,sin(t)\*cos(s)\*d,sin(s)\*d,hue=1,light=[1,0,1],frame=0,zoom=5)" membuat plot tiga dimensi dari fungsi  $f(x,y) = 2x^2 + y^3$ . Plot dibuat menggunakan fungsi plot3d, yang mengambil empat argumen: koordinat x, y, dan z dari titik-titik yang akan diplot, serta beberapa parameter lainnya. Dalam hal ini, koordinat x diberikan oleh ekspresi  $\cos(t)*\cos(s)*d$ , koordinat y diberikan oleh ekspresi  $\sin(t)*\cos(s)*d$ , dan koordinat z diberikan oleh ekspresi  $\sin(s)*d$ . Parameter "hue=1" digunakan untuk mengatur warna pada plot berdasarkan nilai fungsinya. Parameter "light=[1,0,1]" digunakan untuk mengatur pencahayaan pada plot. Parameter "frame=0" digunakan untuk menghilangkan frame pada plot. Parameter "zoom=5" digunakan untuk mengatur level zoom pada plot. **Grafik Tiga Dimensi yang**

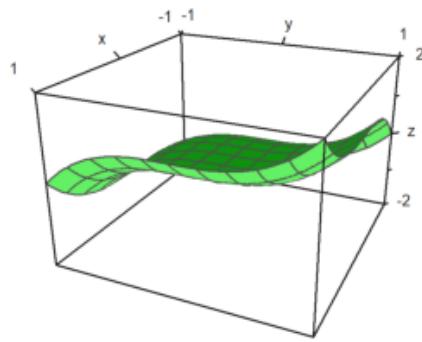
---

\* Bersifat Interaktif dan animasi grafik 3D

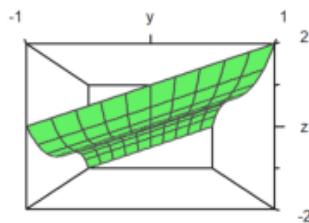
Membuat gambar grafik tiga dimensi (3D) yang bersifat interaktif dan animasi grafik 3D adalah proses menciptakan visualisasi tiga dimensi yang memungkinkan pengguna berinteraksi dengan objek-objek 3D. Interaktivitas dalam gambar 3D memungkinkan pengguna untuk melakukan tindakan seperti mengubah sudut pandang, memindahkan objek, atau berinteraksi dengan elemen-elemen dalam adegan 3D. Animasi grafik 3D dapat mencakup pergerakan, tetapi juga dapat berarti perubahan dalam tampilan atau atribut objek tanpa pergerakan fisik yang mencolok.

CONTOH GAMBAR

```
>function testplot () := plot3d("x^3+y^3"); ...
>rotate("testplot"); testplot():
```



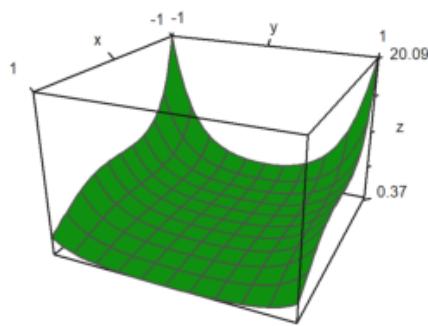
```
>function testplot () := plot3d("x^3+y",distance=3,zoom=1,angle=pi/2,height=0); ...
>rotate("testplot"); testplot():
```



Hilangkan command angle untuk bisa merotasikan grafik,dan height = 0 untuk membuat posisi sejajar dengan mata jadi tidak mempengaruhi pergerakan hanya berbeda sudut pandang saja

```
>plot3d("exp(-x^3+2*y^2)",>user, ...
> title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)

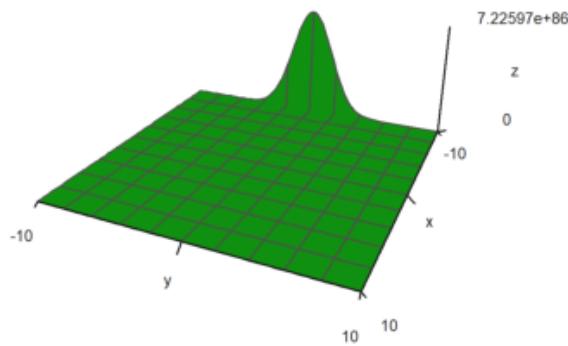


```
>plot3d("exp(x^3+y^2)",>user, ...
>title="Coba gerakan")
```

Interaksi pengguna dimungkinkan dengan parameter. Pengguna dapat menekan tombol berikut.

1. kiri, kanan, atas, bawah: memutar sudut pandang
2. +,-: memperbesar atau memperkecil
3. a: menghasilkan anaglyph (lihat di bawah)
4. l: beralih memutar sumber cahaya (lihat di bawah)
5. spasi: disetel ulang ke default
6. enter: akhiri interaksi

```
>plot3d("exp(-(x^3+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):
```



Parameter "r=10" menunjukkan jari-jari bola yang digunakan untuk membuat plot tiga dimensi. Dalam hal ini, jari-jari bola yang digunakan adalah 10.

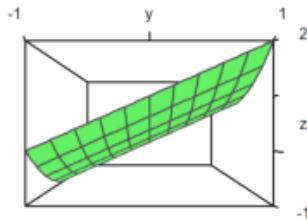
Parameter "n=80" menunjukkan jumlah titik yang digunakan untuk membuat plot. Semakin besar nilai n, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat. Parameter "fscale=4" menunjukkan faktor skala pada sumbu z. Dalam hal ini, faktor skala pada sumbu z adalah 4.

Parameter "scale=1.2" menunjukkan faktor skala pada plot. Semakin besar nilai scale, semakin besar ukuran

plot yang dihasilkan.

Parameter "frame=3" menunjukkan jenis frame yang digunakan pada plot. Dalam hal ini, jenis frame yang digunakan adalah frame kotak dengan sumbu x, y, dan z yang ditampilkan.

```
>plot3d("x^2+y", distance=3, zoom=1, angle=pi/2, height=0) :
```



Tampilan dapat diubah dengan berbagai cara.

- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut terhadap sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.

```
>plot3d("x^4+y^2", a=0, b=1, c=-1, d=1, angle=-20?, height=20?, ...
> center=[0.4,0,0], zoom=5) :
```

Closing bracket missing in function call!

Error in:

```
plot3d("x^4+y^2", a=0, b=1, c=-1, d=1, angle=-20?, height=20?,      c ...
^
```

Plot selalu terlihat berada di tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter center.

Parameter center digunakan untuk memindahkan pusat plot ke lokasi tertentu dalam ruang. Dalam hal ini, pusat plot diatur ke titik (0.4, 0, 0) dalam ruang tiga dimensi. Parameter center berguna ketika kita ingin mengubah sudut pandang plot atau ketika kita ingin menyelaraskan plot dengan objek lain dalam scene. Dengan menentukan pusat plot, kita dapat mengontrol posisi kamera dan arah tampilan plot.

Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: menskalakan ke nilai fungsi (defaultnya adalah <fscale>).

scale: angka atau vektor 1x2 untuk diskalakan ke arah x dan y.

frame: jenis bingkai (default 1).

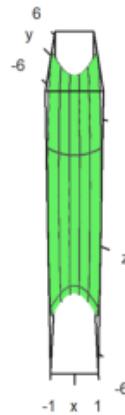
```
>function testplot () := plot3d("5*exp(-x^3-y^2)", r=2, <fscale, <scale, distance=13, height=50
>center=[0,0,-2], frame=3); ...
>rotate("testplot"); testplot():
```

```

Closing bracket missing in function call!
testplot:
  useglobal; return plot3d("5*exp(-x^3-y^2)",r=2,<fscale,<scale ...
Try "trace errors" to inspect local variables after errors.
rotate:
f$(args());

```

```
>plot3d("x^2+5",a=-1,b=1,rotate=true,grid=5):
```



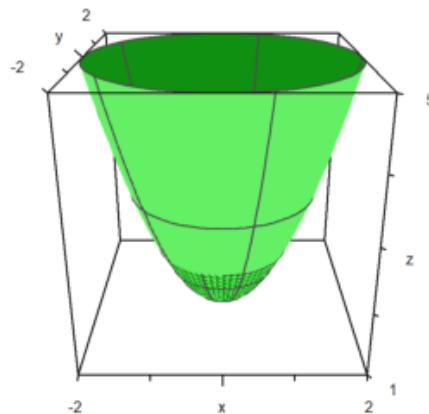
#### Penjelasan:

Secara umum, parameter "a" dan "b" digunakan untuk menentukan rentang nilai variabel independen dalam suatu fungsi. Dalam kasus ini, "a=-1" dan "b=1" menunjukkan bahwa fungsi tersebut akan diplot pada interval [-1, 1]. Parameter "rotate=true" menunjukkan bahwa grafik akan diputar untuk memberikan tampilan bentuk tiga dimensi yang lebih baik. Parameter "grid=5" menunjukkan bahwa grid dengan jarak 5 unit akan ditampilkan pada grafik.

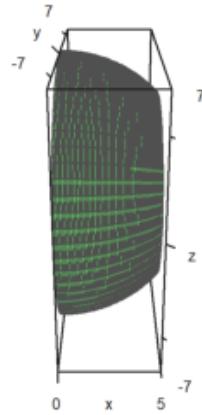
Parameter memutar memutar fungsi dalam x di sekitar sumbu x.

- rotate=1: Menggunakan sumbu x
- rotate=2: Menggunakan sumbu z

```
>plot3d("x^2+1",a=-1,b=2,rotate=3,grid=5):
```



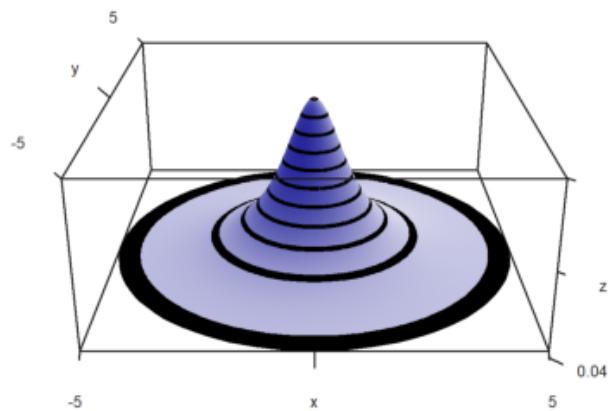
```
>function testplot () := plot3d("sqrt(49-x^2)",a=0,b=5,rotate=1); ...
>rotate("testplot"); testplot():
```



```
>function testplot () := plot3d("x^3+y^2",a=0,b=1,c=-1,d=1,height=20?, ...
>center=[0.4,0,0],zoom=5); ...
>rotate("testplot"); testplot():
```

Closing bracket missing in function call!  
 testplot:  
 useglobal; return plot3d("x^3+y^2",a=0,b=1,c=-1,d=1,height=20 ...  
 Try "trace errors" to inspect local variables after errors.  
 rotate:  
 f\$(args());

```
>function testplot () := plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=blue); ...
>rotate("testplot"); testplot():
```



Parameter "r=5" menunjukkan jari-jari bola yang digunakan untuk membuat plot tiga dimensi. Dalam hal ini, jari-jari bola yang digunakan adalah 5.

Parameter ">polar" menunjukkan bahwa plot yang dibuat adalah plot polar tiga dimensi. Plot polar adalah plot yang dibuat dengan menggunakan koordinat polar, yaitu koordinat yang terdiri dari jarak dan sudut.

Parameter "fscale=2" menunjukkan faktor skala pada sumbu z. Dalam hal ini, faktor skala pada sumbu z adalah 2.

Parameter ">hue" menunjukkan bahwa warna pada plot akan diatur berdasarkan nilai fungsinya. Semakin tinggi nilai fungsinya, semakin terang warnanya.

Parameter "n=100" menunjukkan jumlah titik yang digunakan untuk membuat plot. Semakin besar nilai n, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat.

Parameter "zoom=4" menunjukkan level zoom pada plot.

Parameter "color=blue" menunjukkan warna garis kontur pada plot. Dalam hal ini, warna yang digunakan adalah biru.

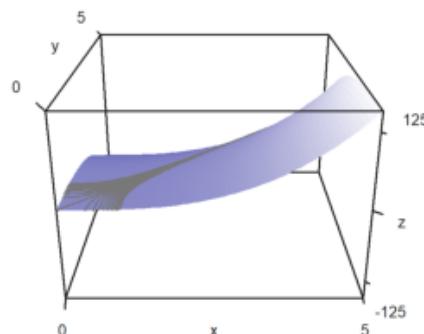
Untuk plotnya, Euler menambahkan garis grid. Sebaliknya dimungkinkan untuk menggunakan garis level dan satu warna atau warna spektral. Euler dapat menggambar ketinggian fungsi pada sebuah plot dengan bayangan. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.

-hue: Mengaktifkan bayangan cahaya, bukan kabel.

-contour: Membuat plot garis kontur otomatis pada plot.

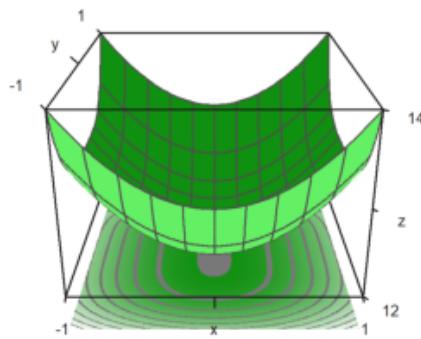
-level=... (atau level): Vektor nilai garis kontur.

```
>function testplot () := plot3d("x^3-y^3",0,5,0,5,level=-1:0.1:1,color=blue); ...
>rotate("testplot"); testplot():
```



Parameter "level=-1:0.1:1" menunjukkan rentang nilai fungsinya yang akan ditampilkan pada plot. Dalam hal ini, rentang nilai fungsinya adalah dari -1 hingga 1 dengan interval 0.1.

```
>function testplot () := plot3d("x^2+y^4+12",>cp,cpcolor=green,cpdelta=0.2); ...
>rotate("testplot"); testplot():
```

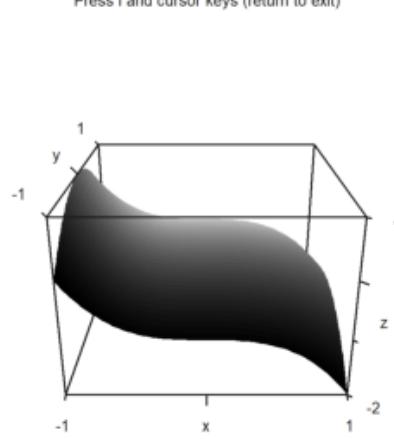


Parameter ">cp" menunjukkan bahwa titik kontrol akan ditambahkan pada plot. Titik kontrol digunakan untuk menentukan bentuk dan posisi plot tiga dimensi.

Parameter "cpcolor=green" menunjukkan warna titik kontrol yang akan digunakan. Dalam hal ini, warna yang digunakan adalah hijau.

Parameter "cpdelta=0.2" menunjukkan jarak antara titik kontrol. Semakin kecil nilai cpdelta, semakin banyak titik kontrol yang akan ditambahkan pada plot.

```
>plot3d("-x^3-y^2", ...
>hue=true, light=[0,1,1], amb=0, user=true, ...
> title="Press l and cursor keys (return to exit)":
```



parameter "hue=true" menunjukkan bahwa warna pada plot akan diatur berdasarkan nilai fungsinya. Semakin tinggi nilai fungsinya, semakin terang warnanya.

Parameter "light=light=[0,1,1] menunjukkan intensitas cahaya pada plot. Nilai light=[0,1,1] menunjukkan bahwa cahaya datang dari arah positif y dan z.

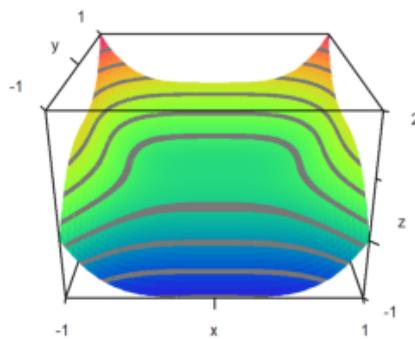
Parameter "amb=0" menunjukkan intensitas cahaya ambient pada plot. Nilai 0 menunjukkan bahwa tidak ada cahaya ambient yang digunakan.

```
>function testplot () := plot3d("-x^2-y^3",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
> zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01); ...
>rotate("testplot"); testplot():
```

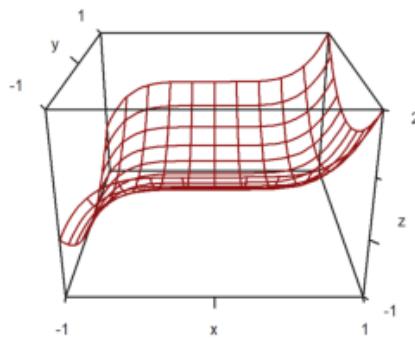


Parameter "frame=false" digunakan untuk menghilangkan frame pada plot tiga dimensi. Parameter "color=rgb(0.2,0.2,0)" menunjukkan warna dasar plot. Dalam hal ini, warna yang digunakan adalah hitam dengan nilai RGB (0.2, 0.2, 0). Parameter "dl=0.01" menunjukkan jarak antara titik-titik pada plot. Semakin kecil nilai dl, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat. Namun, semakin kecil nilai dl, semakin lama waktu yang dibutuhkan untuk membuat plot.

```
>function testplot () := plot3d("x^4+y^3", >contour, >spectral); ...
>rotate("testplot"); testplot();
```



```
>function testplot () := plot3d("x^5+y^2", >transparent, grid=10, wirecolor=red); ...
>rotate("testplot"); testplot();
```



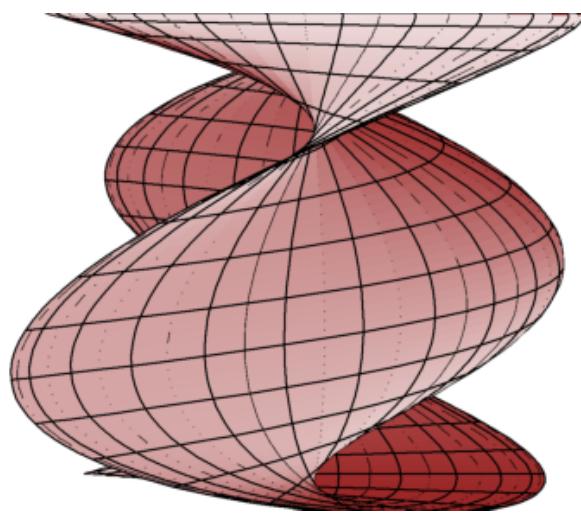
## Fungsi Parametrik 3D

---

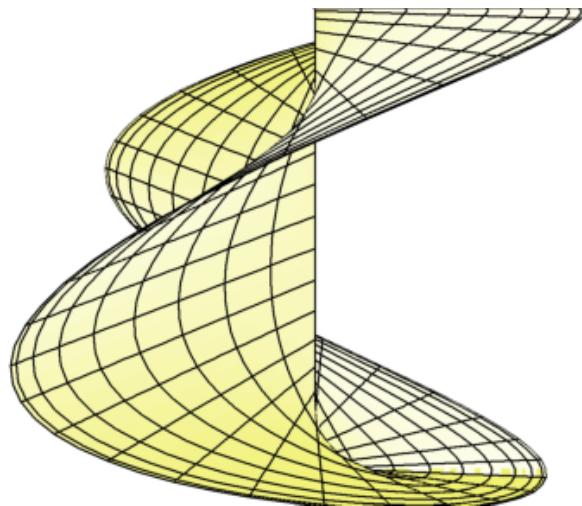
Fungsi parametrik merupakan jenis fungsi matematika yang menggambarkan hubungan antara dua atau lebih variabel, dimana masing-masing koordinat ( $x, y, z...$ ) dinyatakan sebagai fungsi lain dari beberapa parameter. Fungsi parametrik dapat digunakan untuk menggambar kurva, lintasan, atau hubungan antara berbagai variabel yang bergantung pada parameter-parameter tertentu.

Sebagai contoh :

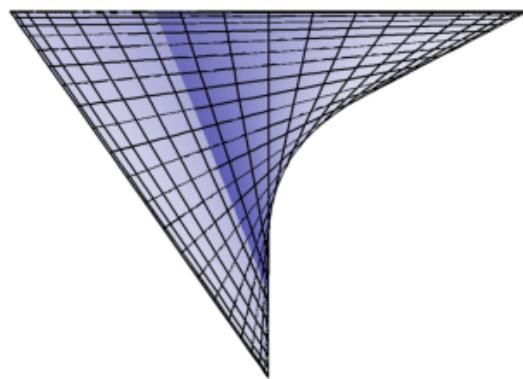
```
>plot3d("cos(2x)*cos(2y)", "sin(x)*cos(y)", "sin(y)", a=0, b=2*pi, c=pi/2, d=-pi/2, ...
>>hue, color=red, light=[0,1,3], <frame, ...
>n=90, grid=[20,50], wirecolor=black, zoom=5) :
```



```
>plot3d("cos(2x)*cos(y)", "sin(x)*cos(y)", "cos(x)", a=0, b=2*pi, c=pi/2, d=-pi/2, ...
>>hue, color=yellow, light=[0,1,3], <frame, ...
>n=90, grid=[20,50], wirecolor=black, zoom=5) :
```



```
>plot3d("cos(x)^2*sin(y)","sin(x)^2*sin(y)","cos(x)^2", a=0,b=2*pi,c=pi/2,d=-pi/2,...  
>>hue,color=blue,light=[0,1,5],<frame,...  
>n=90,grid=[20,50],wirecolor=black,zoom=5) :
```



## 8 Menggambar Fungsi Implisit Implisit

---

Fungsi implisit (implicit function) adalah fungsi yang memuat lebih dari satu variabel, berjenis variabel bebas dan variabel terikat yang berada dalam satu ruas sehingga tidak bisa dipisahkan pada ruas yang berbeda.

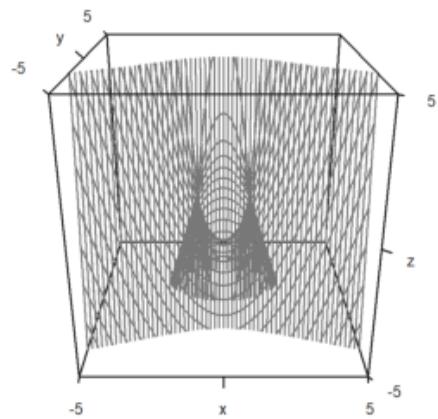
$$F(x, y, z) = 0$$

(1 persamaan dan 3 variabel), terdiri dari 2 variabel bebas dan 1 terikat  
Misalnya,

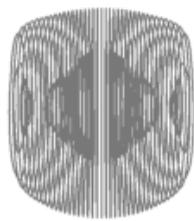
$$F(x, y, z) = x^2 + y^2 + z^2 - 1 = 0$$

adalah persamaan implisit yang menggambarkan bola dengan jari-jari 1 dan pusat di (0,0,0).

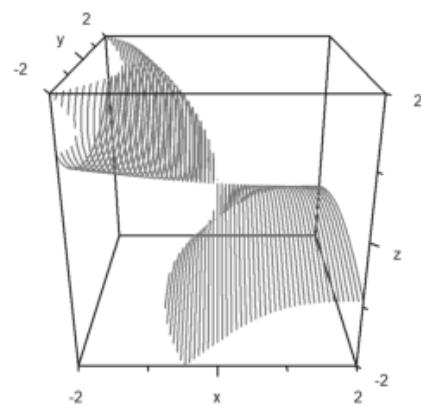
```
>plot3d("x^2+y^2+z^2-1", r=5, implicit=3) :
```



```
>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", x=-5..5, y=-5..5, z=-5..5);
```



```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit, r=2, zoom=2.5);
```



Selain plot kontur yang sudah di jelaskan sebelumnya, pada EMT juga ada plot umplisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari sebuah fungsi dalam tiga variabel.

Solusi dari

$$f(x, y, z) = 0$$

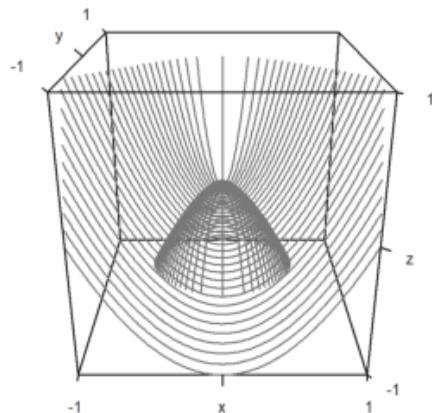
dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, bidang x-z, dan bidang y-z.

- implisit = 1: potong sejajar dengan bidang-y-z
- implicit = 2: memotong sejajar dengan bidang x-z
- implicit=4: memotong sejajar dengan bidang x-y

Ambil contoh dari persamaan latex pada fungsi implisit tadi dan tambahkan nilai-nilai ini, sehingga kita dapat memplot persamaan ini

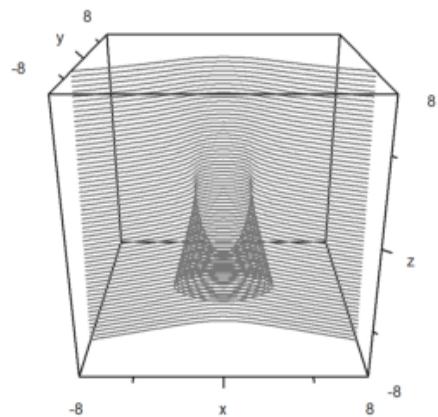
$$M = (x, y, z) : x^2 + y^3 + zy = 1$$

```
>plot3d("x^2+y^3+z*y", r=1, implicit=2) :
```

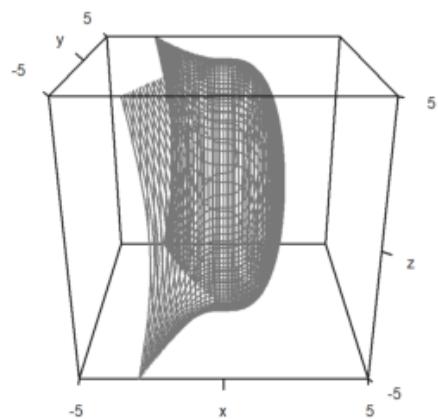


Contoh fungsi implisit yang lainnya

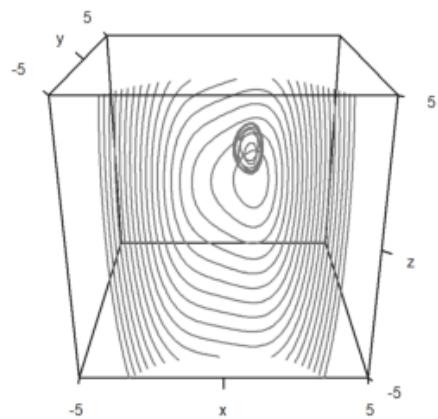
```
>plot3d("x^2+y^3+z*y-3", r=8, implicit=4) :
```



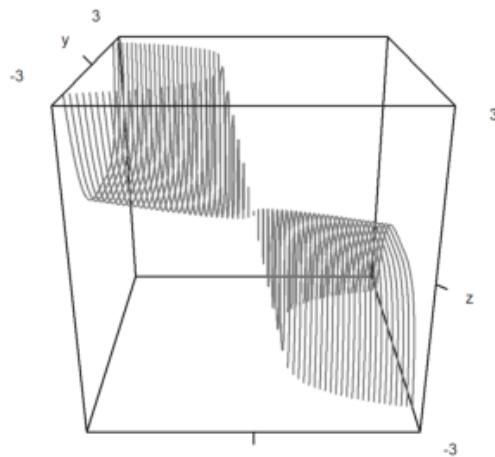
```
>plot3d("2*x^3 + 2*y^2 + z^2 - 25", r=5, implicit=3):
```



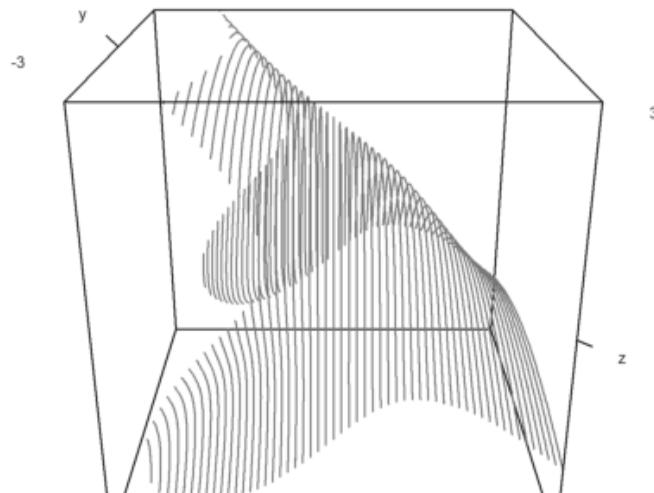
```
>plot3d("x^4 + 2*y^3 + 2*z^2 - 5*x - 7*y - 5*z + 10", r=5, implicit=2):
```



```
>plot3d("x^2+y^4+5*x*z+z^2",>implicit,r=3,zoom=3):
```



```
>plot3d("x^3+y^2+4*x*z+z^3-6",>implicit,r=3,zoom=4):
```



## Fungsi Implisit Menggunakan Povray

Povray dapat memplot himpunan di mana  $f(x,y,z)=0$ , seperti parameter implisit di plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsi-fungsi tersebut sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

$$((x^2 + y^2 - c^2)^2 + (z^2 - 1)^2) * ((y^2 + z^2 - c^2)^2 + (x^2 - 1)^2) * ((z^2 + x^2 - c^2)^2 + (y^2 - 1)^2) = d$$

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe

```
>povstart(angle=70°,height=50°, zoom=4);
>writeln(povsurface ("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(blue)));
>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);
```

---

---

## BAB 5

---

# PEKAN 6: MENGGUNAKAN EMT UNTUK KALKULUS

[a4paper,10pt]article eumat  
Raden Mas Farrel Maheswara Kusuma Dewa  
22305141042  
Matematika E 2022

### 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.

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik  
>f(0), f(1), f(pi)
```

```
1  
4.31977682472  
20.7392088022
```

```
>function g(x) := sqrt(x^2-3*x) / (x+1)  
>g(3)
```

```
0
```

```
>g(0)
```

```
0
```

```
>g(1)
```

```
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) ...  
^
```

```
>f(g(5)) // komposisi fungsi
```

```
2.20920171961
```

```
>g(f(5))
```

```
0.950898070639
```

```
>f(0:10) // nilai-nilai 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]
```

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0. \end{cases}$$

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format `:=`, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.

```
>function map f(x) ...  
  
    if x>0 then return x^3  
    else return x^2  
    endif;  
endfunction
```

```
>f(1)
```

```
1
```

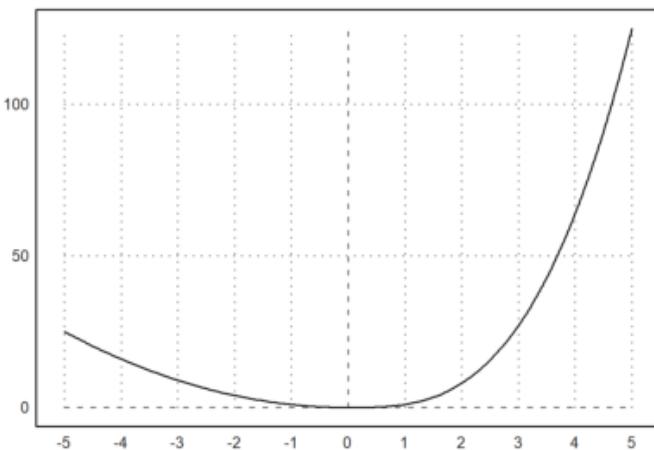
```
>f(-2)
```

```
4
```

```
>f(-5:5)
```

```
[25, 16, 9, 4, 1, 0, 1, 8, 27, 64, 125]
```

```
>aspect(1.5); plot2d("f(x)", -5, 5):
```



```
>function f(x) &= 2*x^2 // fungsi simbolik
```

$$2 \frac{x^2}{E}$$

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

$$3x + 1$$

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

$$\begin{matrix} 3x + 1 \\ 2 \rightarrow \end{matrix}$$

## 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 beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik tersebut.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

Jawab:

### A). FUNGSI 1 VARIABEL

#### 1. Fungsi 1

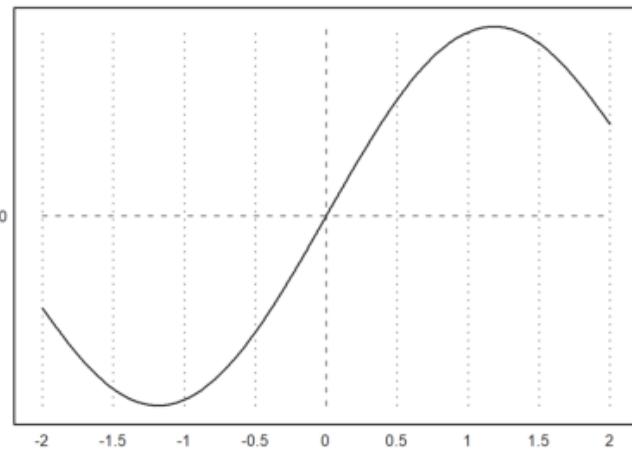
```
>function k(x) := x*(x^2-7)^2  
>k(2), k(3), k(4)
```

18  
12  
324

```
>kmap(-5:5)
```

[-1620, -324, -12, -18, -36, 0, 36, 18, 12, 324, 1620]

```
>plot2d("k(x)":
```



## 2. Fungsi 2

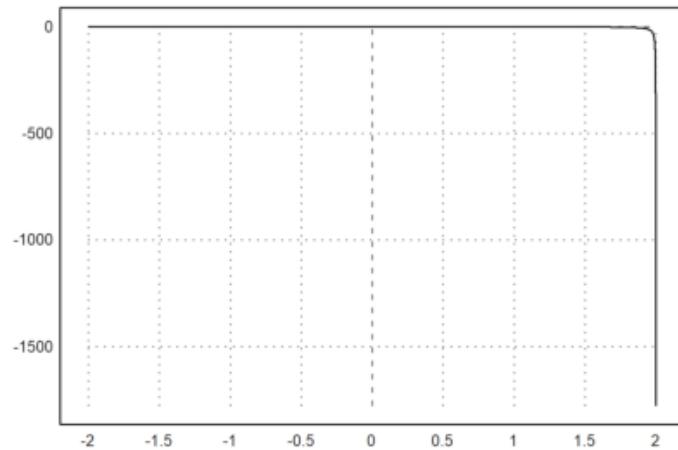
```
>function y(x) := (x)^3/(x^3-8)
>y(5), y(7), y(9)
```

```
1.06837606838
1.02388059701
1.01109570042
```

```
>ymap(-2:-1)
```

```
[0.5, 0.111111]
```

```
>plot2d("y(x)":
```



## 3. Fungsi 3

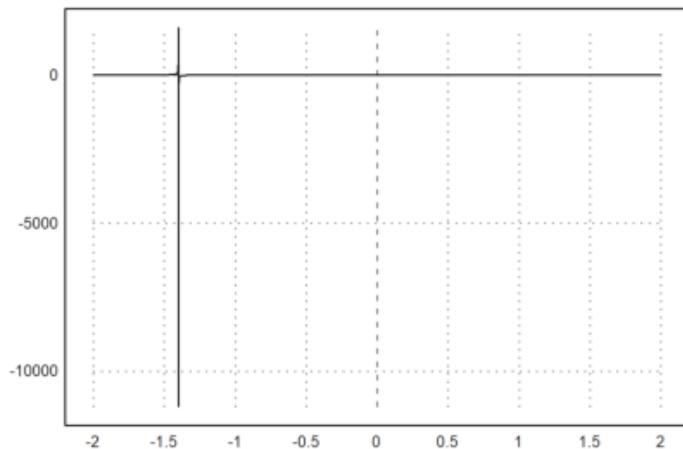
```
>function k(x) := 2*x/(5*x+7)
>k(1), k(-1), k(-3), k(3)
```

```
0.166666666667
-1
0.75
0.272727272727
```

```
>kmap(1:4)
```

```
[0.166667, 0.235294, 0.272727, 0.296296]
```

```
>plot2d("k(x)":
```



#### 4. Fungsi 4

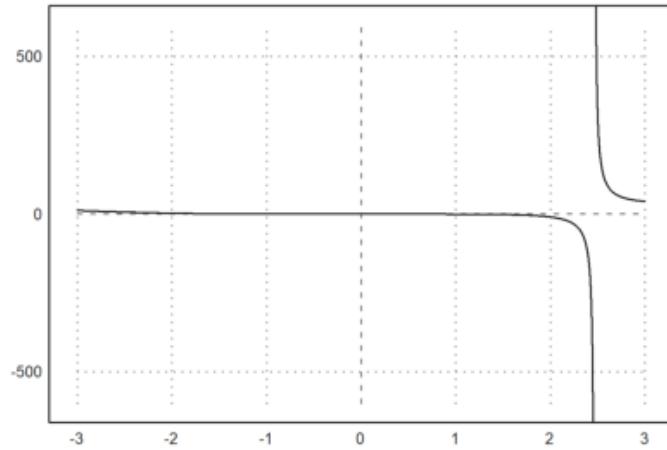
```
>function j(x) := 2*x^5/(x^3-15)
>j(2), j(4), j(6)
```

```
-9.14285714286
41.7959183673
77.3731343284
```

```
>jmap(3:5)
```

```
[40.5, 41.7959, 56.8182]
```

```
>plot2d("j(x)", -3, 3, -600, 600):
```



## 5. Fungsi 5

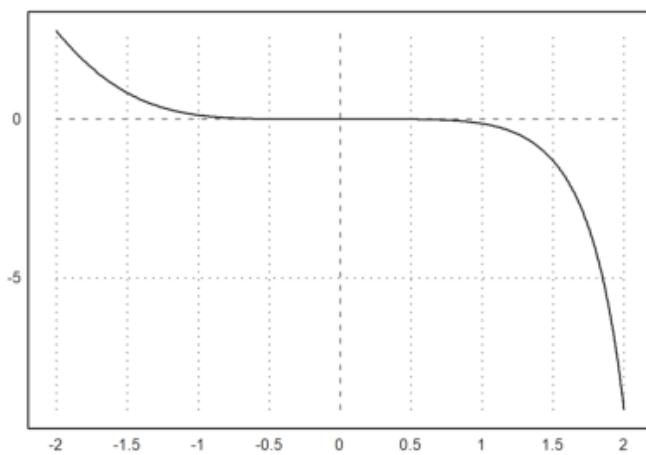
```
>function l(x) := (-cos(2x)) * cos(8x)
>l(2pi), l(0), l(pi/2)
```

```
-1
-1
1
```

```
>lmap(0:4pi)
```

```
[-1, -0.0605494, -0.625968, -0.407284, 0.12138, -0.559609,
0.540188, -0.116667, 0.375266, 0.638692, 0.0450471, 0.999334,
0.0765348]
```

```
>plot2d("j(x)");
```



## 6. Fungsi 6

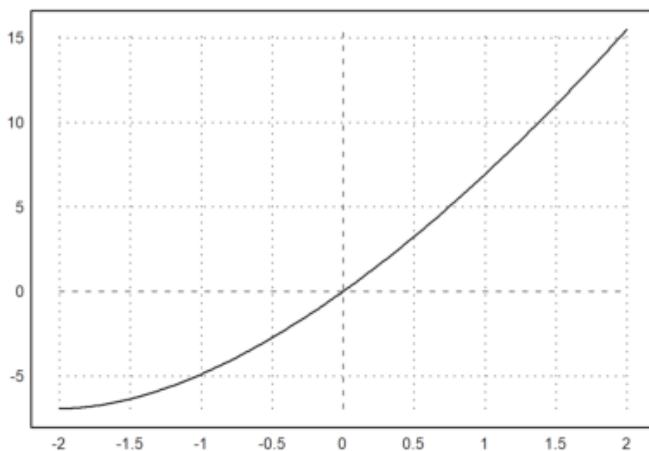
```
>function z(x) := 2x*sqrt(3x+9)
>z(6), z(7), z(8)
```

62.3538290725  
76.6811580507  
91.9130023446

```
>zmap(3:12)
```

[25.4558, 36.6606, 48.9898, 62.3538, 76.6812, 91.913, 108,  
124.9, 142.576, 160.997]

```
>plot2d("z(x)":
```



## B). FUNGSI 2 VARIABEL

### 1. Fungsi 1

```
>function a(x,y) ...
```

```
return x^2+y^2-30
endfunction
```

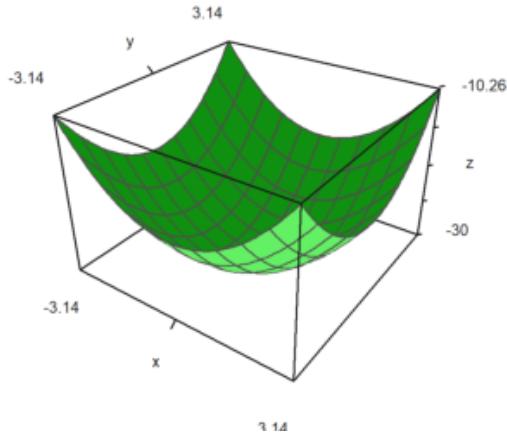
```
>a(3,1), a(7,5), a(5,3)
```

-20  
44  
4

```
>amap (-1:2,3:3)
```

```
[-20, -21, -20, -17]
```

```
>aspect=1.5; plot3d("a(x,y)",a=-100,b=100,c=-80,d=80,angle=35°,height=30°,r=pi,n=100):
```



## 2. Fungsi 2

```
>function q(x,y) ...
```

```
return y^3/(x^2/4)
endfunction
```

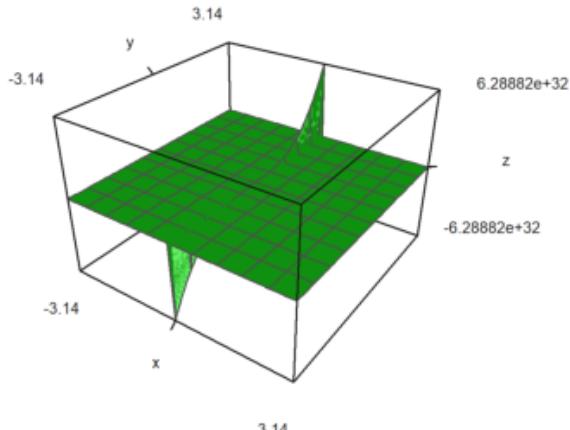
```
>q(2,3), q(5,7), q(4,5)
```

```
27
54.88
31.25
```

```
>qmap (2:2,-3:2)
```

```
[-27, -8, -1, 0, 1, 8]
```

```
>aspect=1.5; plot3d("q(x,y)",a=-100,b=100,c=-80,d=80,angle=35°,height=30°,r=pi,n=100):
```



## 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(1/(2*x-1),x,0))
```

$$\lim_{x \rightarrow 0} \frac{1}{2x-1} = -1$$

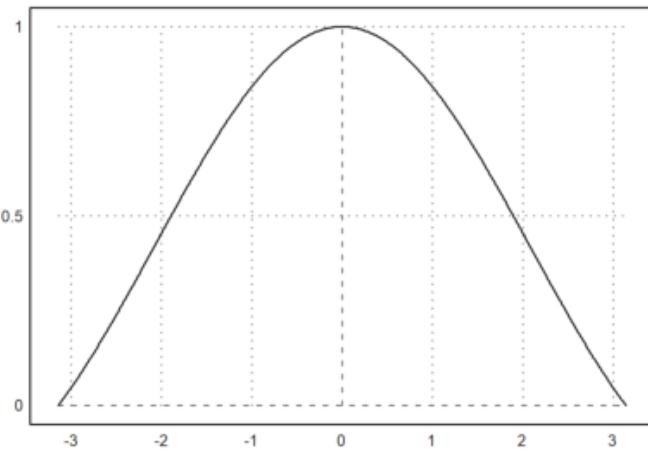
```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

$$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$$

```
>$showev('limit(sin(x)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

```
>plot2d("sin(x)/x",-pi,pi):
```



```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

```
>$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))
```

Maxima said:

```
limit: second argument must be a variable, not a constant; found:  
errexpl  
#0: showev(f='limit([-sqrt(2),-1.400673597966589,-1.387124727947029,-1.37356688476182,-1  
-- an error. To debug this try: debugmode(true);
```

Error in:

```
$showev('limit(t-sqrt(2-t),t,2,minus)) ...  
^
```

```
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
```

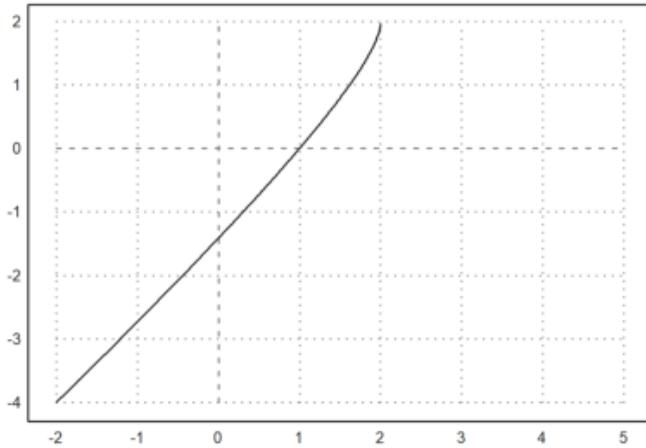
Maxima said:

```
limit: second argument must be a variable, not a constant; found:  
errexpl
```

```
#0: showev(f='limit([-sqrt(2), -1.400673597966589, -1.387124727947029, -1.37356688476182, -1
-- an error. To debug this try: debugmode(true);

Error in:
$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya ...
^
```

```
>plot2d("x-sqrt(2-x)",-2,5):
```



```
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
```

$$\lim_{x \rightarrow 3} \frac{x^2 - 9}{2x^2 - 5x - 3} = \frac{6}{7}$$

```
>$showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} = 0$$

```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

```
>$showev('limit((1+1/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{1}{x} + 1 \right)^x = e$$

```
>\$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))
```

$$\lim_{x \rightarrow 0} (x+1)^{\frac{1}{x}} = e$$

```
>\$showev('limit((x/(x+k))^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{x}{x+k} \right)^x = e^{-k}$$

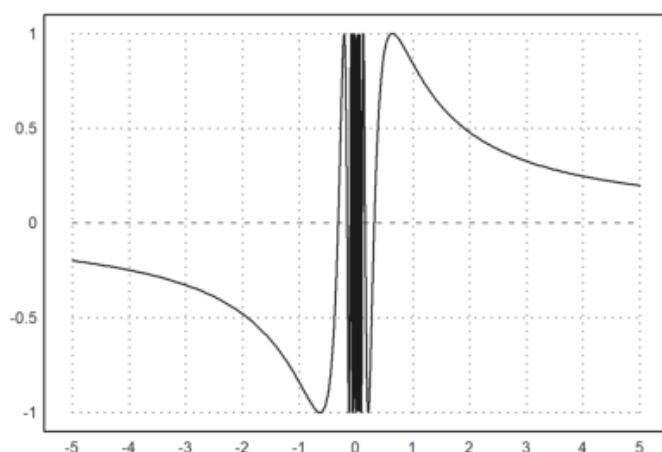
```
>\$showev('limit(sin(1/x),x,0))
```

$$\lim_{x \rightarrow 0} \sin\left(\frac{1}{x}\right) = \text{ind}$$

```
>\$showev('limit(sin(1/x),x,inf))
```

$$\lim_{x \rightarrow \infty} \sin\left(\frac{1}{x}\right) = 0$$

```
>plot2d("sin(1/x)",-5,5):
```



## 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.

Jawab:

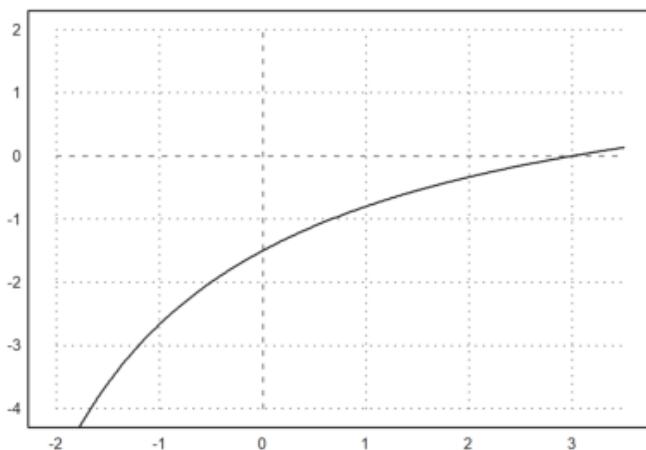
### 1. Fungsi 1

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

```
>showev('limit((2*x-6)/(x+4), x, 2))
```

$$\lim_{x \rightarrow 2} \frac{2x - 6}{x + 4} = -\frac{1}{3}$$

```
>plot2d("(2*x-6)/(x+4)", -2, 3.5, -4, 2):
```



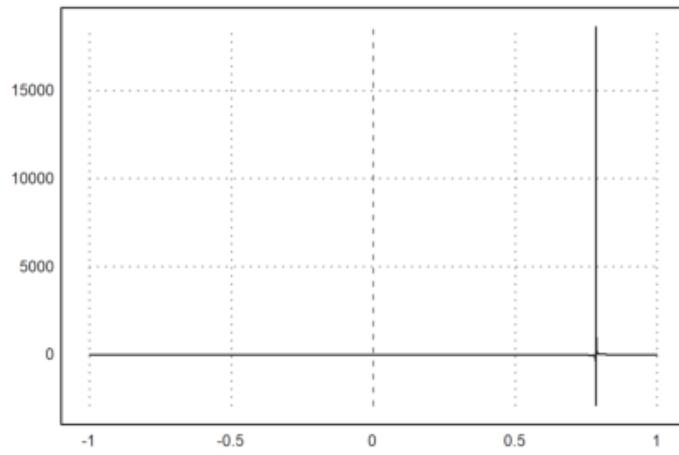
### 2. Fungsi 2

$$f(x) = \frac{\sin 3x}{\sin x - \cos x}$$

```
>showev('limit(sin(3*x)/(sin(x) - cos(x)), x, 0))
```

$$\lim_{x \rightarrow 0} \frac{\sin(3x)}{\sin x - \cos x} = 0$$

```
>plot2d("sin(3*x)/(sin(x) - cos(x))", -1, 1):
```



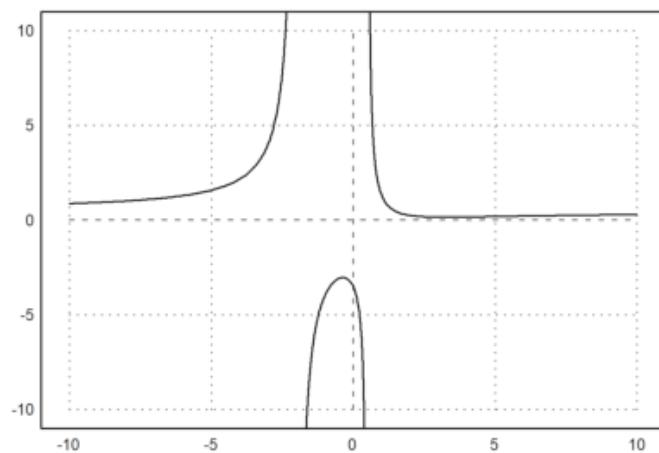
### 3. Fungsi 3

$$f(x) = \frac{x^2 - 4x + 7}{2x^2 + 3x - 2}$$

```
>$showev('limit(((x^2-4*x+7)/(2*x^2+3*x-2)),x,2))
```

$$\lim_{x \rightarrow 2} \frac{x^2 - 4x + 7}{2x^2 + 3x - 2} = \frac{1}{4}$$

```
>plot2d("(x^2-4*x+7)/(2*x^2+3*x-2)",-10,10,-10,10):
```



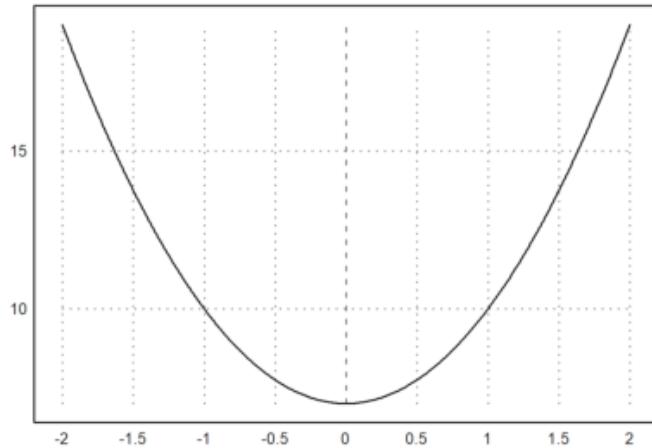
### 4. Fungsi 4

$$f(x) = 3x^2 + 7$$

```
>$showev('limit((3*x^2+7),x,3))
```

$$\lim_{x \rightarrow 3} 3x^2 + 7 = 34$$

```
>plot2d(" (3*x^2+7) ":
```



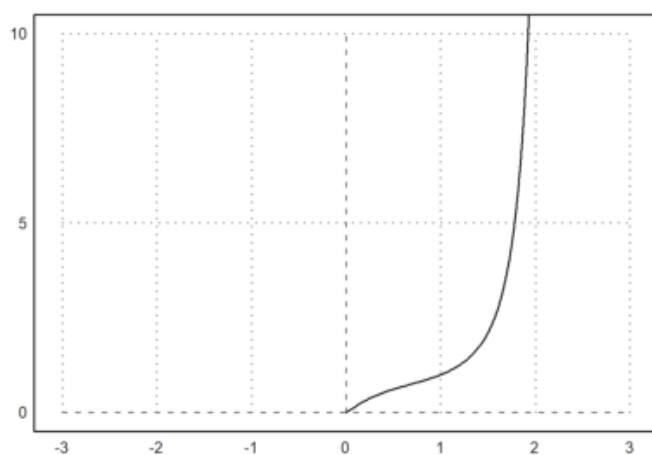
## 5. Fungsi 5

$$f(x) = x^{x^x}$$

```
>showev('limit((x^(x^(x))),x,2))
```

$$\lim_{x \rightarrow 2} x^{x^x} = 16$$

```
>plot2d(" (x^(x^(x))) ",-3,3,0,10):
```



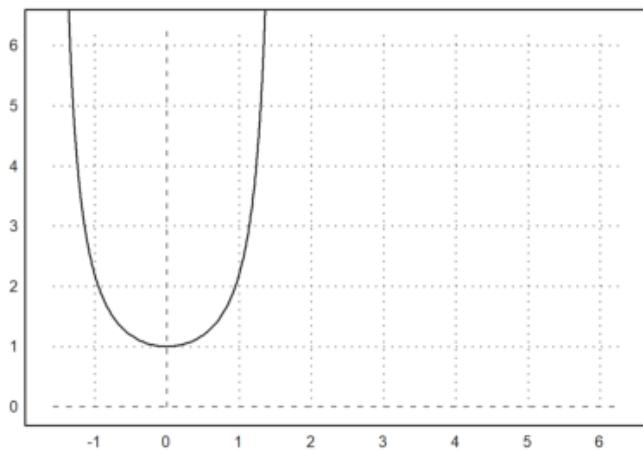
## 6. Fungsi 6

$$f(x) = \frac{2x \tan x}{1 - \cos 2x}$$

```
>$showev('limit((2*x*tan(x))/(1-cos(2*x)),x,0))
```

$$2 \left( \lim_{x \rightarrow 0} \frac{x \tan x}{1 - \cos(2x)} \right) = 1$$

```
>plot2d("(2*x*tan(x))/(1-cos(2*x))", -pi/2, 2pi, 0, 2pi):
```



## 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)^n-x^n)/h,h,0)) // turunan x^n
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} = n x^{n-1}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan  $(x+h)^n$  dengan menggunakan teorema binomial.

Jawab:

Akan ditunjukkan bahwa  $f'(x) = \lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} = n x^{n-1}$

Pertama, ekspansikan  $(x+h)^n$ , yakni:

$$\begin{aligned}
(x+h)^n &= \sum_{k=0}^n \binom{n}{k} x^{n-k} h^k \\
\Leftrightarrow (x+h)^n &= \binom{n}{0} x^n + \binom{n}{1} x^{n-1} h + \binom{n}{2} x^{n-2} h^2 + \dots + \binom{n}{n} h^n \\
\Leftrightarrow (x+h)^n &= x^n + nx^{n-1} h + \binom{n}{2} x^{n-2} h^2 + \binom{n}{3} x^{n-3} h^3 + \dots + h^n \\
\text{Sehingga, } f'(x) \text{ menjadi: } f'(x) &= \lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} \\
\Leftrightarrow f'(x) &= \lim_{h \rightarrow 0} \frac{x^n + nx^{n-1} h + \binom{n}{2} x^{n-2} h^2 + \binom{n}{3} x^{n-3} h^3 + \dots + h^n - x^n}{h} \\
\Leftrightarrow f'(x) &= \lim_{h \rightarrow 0} nx^{n-1} + \binom{n}{2} x^{n-2} h + \binom{n}{3} x^{n-3} h^2 + \dots + h^{n-1} \\
\Leftrightarrow f'(x) &= nx^{n-1}. \text{ Terbukti.}
\end{aligned}$$

```
>$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. Jawab:

Akan ditunjukkan bahwa  $\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h} = \cos x$

Diketahui bahwa:

$$1). \sin(x+h) = \sin x \cos h + \cos x \sin h$$

$$2). \lim_{h \rightarrow 0} \frac{1 - \cos h}{h} = 0$$

$$3). \lim_{h \rightarrow 0} \frac{\sin h}{h} = 1$$

$$\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h}$$

$$= \lim_{h \rightarrow 0} \frac{\sin x \cos h + \cos x \sin h - \sin x}{h}$$

$$= \lim_{h \rightarrow 0} \left[ -\sin x \cdot \frac{1 - \cos h}{h} + \cos x \cdot \frac{\sin h}{h} \right]$$

$$= (-\sin x) \left[ \lim_{h \rightarrow 0} \frac{1 - \cos h}{h} + (\cos x) \lim_{h \rightarrow 0} \frac{\sin h}{h} \right]$$

$$= (-\sin x)(0) + (\cos x)(1) = \cos x. \text{ Terbukti.}$$

```
>$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.

Jawab:

Bukti:

Ambil  $f(x) =^a \log x$ .

$$\begin{aligned} & \lim_{h \rightarrow 0} \frac{^a \log(x+h) - ^a \log x}{h} \\ &= \lim_{h \rightarrow 0} \frac{^a \log \frac{(x+h)}{x}}{h} \\ &= \lim_{h \rightarrow 0} \frac{^a \log(1 + \frac{h}{x})}{h} \\ &= \lim_{h \rightarrow 0} \frac{^a \log(1 + \frac{h}{x})}{\frac{h}{x}x} \\ &= \lim_{h \rightarrow 0} \frac{\frac{x}{h} \cdot ^a \log(1 + \frac{h}{x})}{x} \\ &= \lim_{h \rightarrow 0} \frac{^a \log(1 + \frac{h}{x})^{\frac{x}{h}}}{x} \\ &= \frac{\lim_{h \rightarrow 0} ^a \log(1 + \frac{h}{x})^{\frac{x}{h}}}{\lim_{h \rightarrow 0} x} \\ &= \frac{1}{x \cdot {}^e \log a} \\ &= \frac{1}{x \cdot \ln a} \end{aligned}$$

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))
```

$$\lim_{h \rightarrow 0} \frac{e^h - 1}{h} = 1$$

```
>$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+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x + h)^{x+h} - x^x}{h} = infinity$$

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h}.$$

Dalam hal ini diperlukan asumsi nilai x.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = x^x (\log x + 1)$$

```
>&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)
```

$$\lim_{h \rightarrow 0} \frac{\arcsin(x+h) - \arcsin x}{h} = \frac{1}{\sqrt{1-x^2}}$$

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

$$\lim_{h \rightarrow 0} \frac{\tan(x+h) - \tan x}{h} = \frac{1}{\cos^2 x}$$

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

sinh(x)

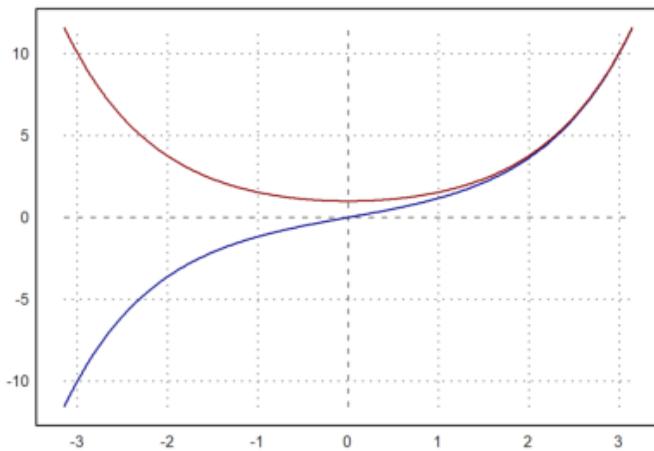
```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

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

Hasilnya adalah  $\cosh(x)$ , karena

$$\frac{e^x + e^{-x}}{2} = \cosh(x).$$

```
>plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]):
```



## 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), seperti contoh-contoh tersebut. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

Jawab:

1. Fungsi 1

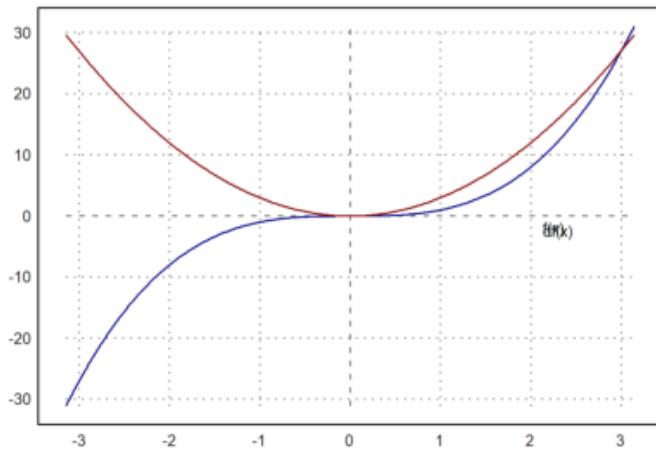
```
>function f(x) := x^3
>$showev('limit(((x+h)^3-x^3)/h,h,0)) // turunan x^3
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^3 - x^3}{h} = 3x^2$$

```
>function df(x) &= limit(((x+h)^3-x^3)/h,h,0); $df(x) // df(x) = f'(x)
```

$$3x^2$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",2,0.6), label("df(x)",2,0.1)
```



## 2. Fungsi 2

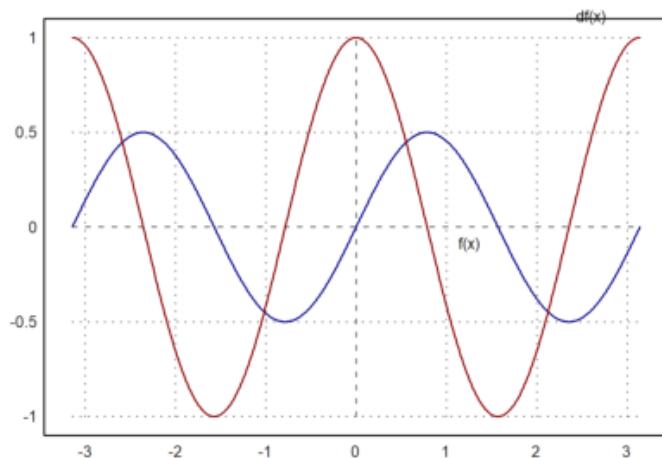
```
>function f(x) := cos(x)*sin(x)
>$showev('limit(((cos(x+h)*sin(x+h))-cos(x)*sin(x))/h,h,0)) // turunan cos(x)*sin(x)
```

$$\lim_{h \rightarrow 0} \frac{\cos(x+h) \sin(x+h) - \cos x \sin x}{h} = \cos^2 x - \sin^2 x$$

```
>function df(x) &= limit(((cos(x+h)*sin(x+h))-cos(x)*sin(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)",2.3,1.1)
```



### 3. Fungsi 3

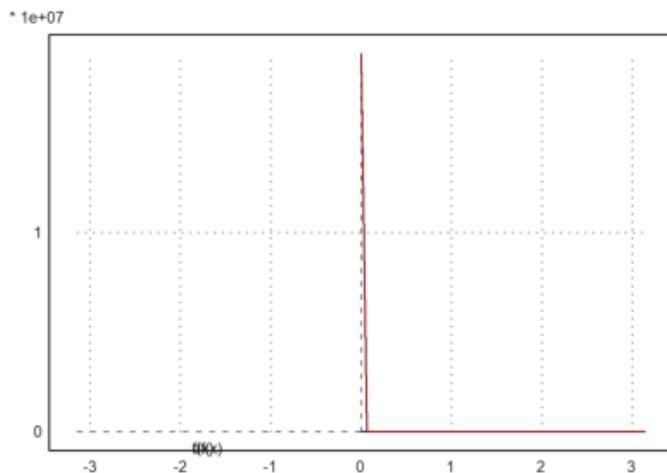
```
>function f(x) := sqrt(x)*2  
>$showev('limit((sqrt(x+h)*2-sqrt(x)*2)/h,h,0)) // turunan sqrt(x)*2
```

$$\lim_{h \rightarrow 0} \frac{2\sqrt{x+h} - 2\sqrt{x}}{h} = \frac{1}{\sqrt{x}}$$

```
>function df(x) &= limit((sqrt(x+h)*2-sqrt(x)*2)/h,h,0); $df(x) // df(x) = f'(x)
```

$$\frac{1}{\sqrt{x}}$$

```
>plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]), label("f(x)", -2, 11), label("df(x)", -2, -1)
```



### 4. Fungsi 4

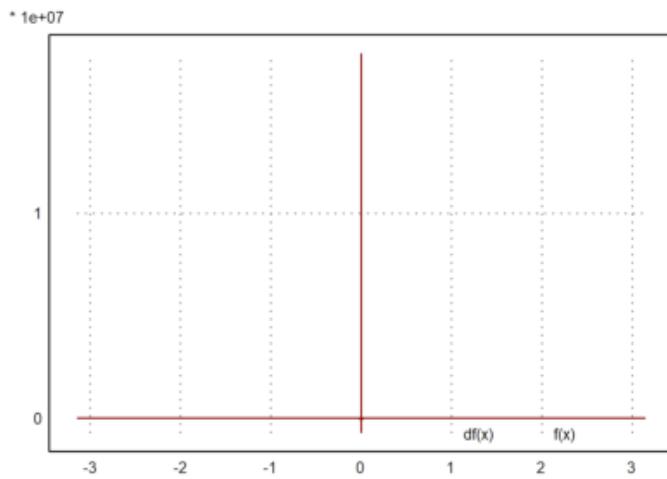
```
>function f(x) := sin(1/x)  
>$showev('limit((sin(1/(x+h))-sin(1/x))/h,h,0)) // turunan sin(1/x)
```

$$\lim_{h \rightarrow 0} \frac{\sin\left(\frac{1}{x+h}\right) - \sin\left(\frac{1}{x}\right)}{h} = -\frac{\cos\left(\frac{1}{x}\right)}{x^2}$$

```
>function df(x) &= limit((sin(1/(x+h))-sin(1/x))/h,h,0); $df(x) // df(x) = f'(x)
```

$$-\frac{\cos\left(\frac{1}{x}\right)}{x^2}$$

```
>plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]), label("f(x)", 2, 0.4), label("df(x)", 1, -0.4)
```



## 5. Fungsi 5

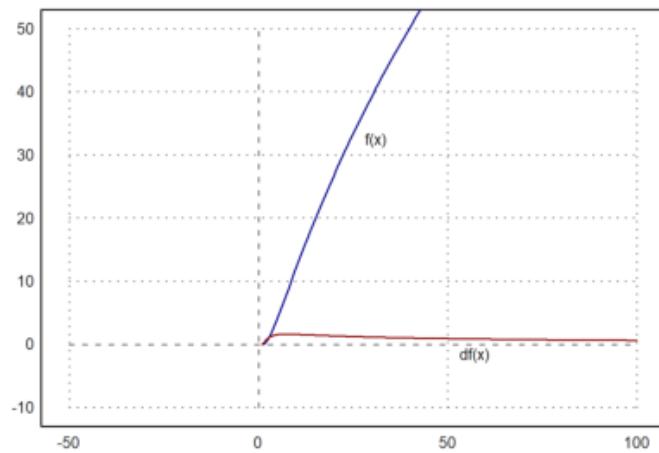
```
>function f(x) := (log(x))^3
>$showev('limit(((log(x+h))^3-(log(x))^3)/h,h,0)) // turunan (log(x))^3
```

$$\lim_{h \rightarrow 0} \frac{\log^3(x+h) - \log^3 x}{h} = \frac{3 \log^2 x}{x}$$

```
>function df(x) &= limit(((log(x+h))^3-(log(x))^3)/h,h,0); $df(x) // df(x) = f'(x)
```

$$\frac{3 \log^2 x}{x}$$

```
>plot2d(["f(x)", "df(x)"], -50, 100, -10, 50, color=[blue, red]), label("f(x)", 25, 35), label("df(x)", 75, 0)
```



## 6. Fungsi 6

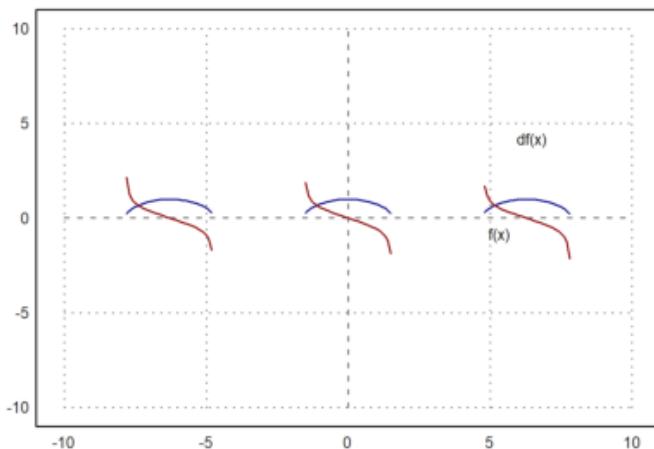
```
>function f(x) := sqrt(cos(x))
>$showev('limit((sqrt(cos(x+h))-sqrt(cos(x)))/h,h,0)) // turunan sqrt(cos(x))
```

$$\lim_{h \rightarrow 0} \frac{\sqrt{\cos(x+h)} - \sqrt{\cos x}}{h} = -\frac{\sin x}{2\sqrt{\cos x}}$$

```
>function df(x) &= limit((sqrt(cos(x+h))-sqrt(cos(x)))/h,h,0); $df(x)// df(x) = f'(x)
```

$$-\frac{\sin x}{2\sqrt{\cos x}}$$

```
>plot2d(["f(x)", "df(x)"], -10, 10, -10, 10, color=[blue, red]), label("f(x)", 4.5, 0), label("df(x)", 4.5, 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))
```

$$\int_0^\pi \sin x \, dx = 2$$

```
>showev('integrate(sin(x),x,a,b))
```

$$\int_a^b \sin x \, dx = \cos a - \cos b$$

```
>showev('integrate(x^n,x,a,b))
```

Answering "Is n positive, negative or zero?" with "positive"

$$\int_a^b x^n \, dx = \frac{b^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}$$

```
>showev('integrate(x^2*sqrt(2*x+1),x))
```

$$\int x^2 \sqrt{2x+1} \, dx = \frac{(2x+1)^{\frac{7}{2}}}{28} - \frac{(2x+1)^{\frac{5}{2}}}{10} + \frac{(2x+1)^{\frac{3}{2}}}{12}$$

```
>showev('integrate(x^2*sqrt(2*x+1),x,0,2))
```

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{25^{\frac{5}{2}}}{21} - \frac{2}{105}$$

```
>${ratsimp(%)}
```

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{25^{\frac{7}{2}} - 2}{105}$$

```
>${showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))}
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^\pi - 1) \sin a + (e^\pi + 1) \cos a$$

```
>${factor(%)}
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^\pi - 1) (\sin a - \cos a)$$

```
>function map f(x) &= E^(-x^2); $f(x)
```

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

```
>${showev('integrate(f(x),x))}
```

$$\int e^{-x^2} dx = \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

Fungsi  $f$  tidak memiliki antiturunan, integralnya masih memuat integral lain.

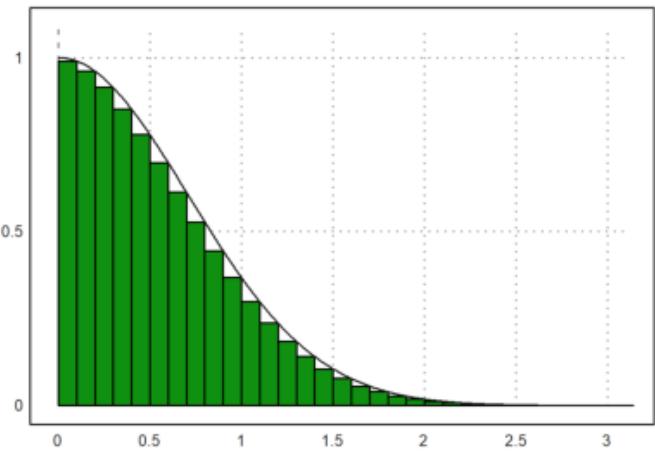
$$\operatorname{erf}(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}} dx.$$

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

$$\int_0^{\pi} e^{-x^2} dx$$

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```



### Integral tentu

$$\int_0^{\pi} e^{-x^2} dx$$

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva  $y=f(x)$  tersebut. Langkah-langkahnya adalah sebagai berikut.

```
>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-nilai x
>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)
>/> jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!
```

Hasilnya adalah:

$$\int_0^{\pi} e^{-x^2} dx = 0.8362196102528469$$

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai  $f(x)$  untuk  $x = 0.1, 0.2, 0.3, \dots, 3.2$ .

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

0.836219610253

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

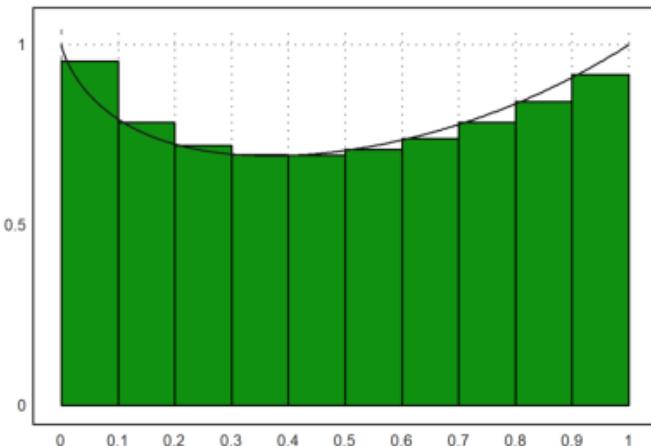
```
>function f(x) &= x^x; $f(x)
```

$$x^x$$

```
>$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));
```

## Latihan

- Bukalah buku Kalkulus.
- Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi).
- Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah integral tentu dengan batas-batas yang menarik (Anda tentukan sendiri), seperti contoh-contoh tersebut.
- Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat diintegralkan (cari sedikitnya 3 fungsi).
- Gambar grafik fungsi dan daerah integrasinya pada sumbu koordinat yang sama.
- Gunakan integral tentu untuk mencari luas daerah yang dibatasi oleh dua kurva yang berpotongan di dua titik. (Cari dan gambar kedua kurva dan arsir (warnai) daerah yang dibatasi oleh keduanya.)

- Gunakan integral tentu untuk menghitung volume benda putar kurva  $y = f(x)$  yang diputar mengelilingi sumbu x dari  $x=a$  sampai  $x=b$ , yakni

$$V = \int_a^b \pi(f(x))^2 dx.$$

(Pilih fungsinya dan gambar kurva dan benda putar yang dihasilkan. Anda dapat mencari contoh-contoh bagaimana cara menggambar benda hasil perputaran suatu kurva.)

- Gunakan integral tentu untuk menghitung panjang kurva  $y=f(x)$  dari  $x=a$  sampai  $x=b$  dengan menggunakan rumus:

$$S = \int_a^b \sqrt{1 + (f'(x))^2} dx.$$

(Pilih fungsi dan gambar kurvanya.)

Jawab:

1. Fungsi 1

```
>function f(x) &= 3*x^4; $f(x)
```

$$3x^4$$

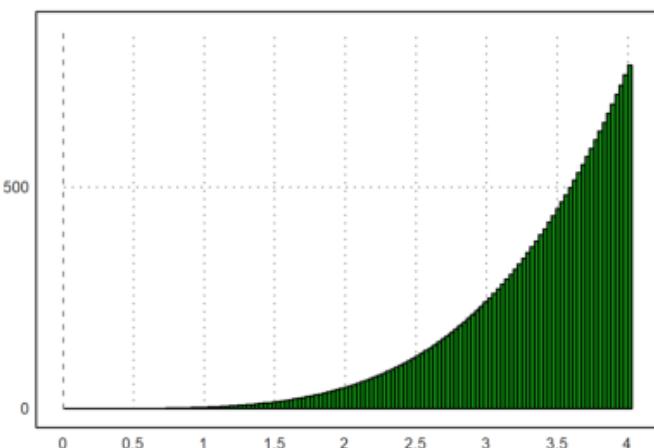
```
>$showev('integrate(f(x),x))
```

$$3 \int x^4 dx = \frac{3x^5}{5}$$

```
>$showev('integrate(f(x),x,2,3))
```

$$3 \int_2^3 x^4 dx = \frac{633}{5}$$

```
>x=0.01:0.03:4; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",2,3,>add):
```



## 2. Fungsi 2

```
>function f(x) &= tan(3*x-7); $f(x)
```

$$\tan(3x - 7)$$

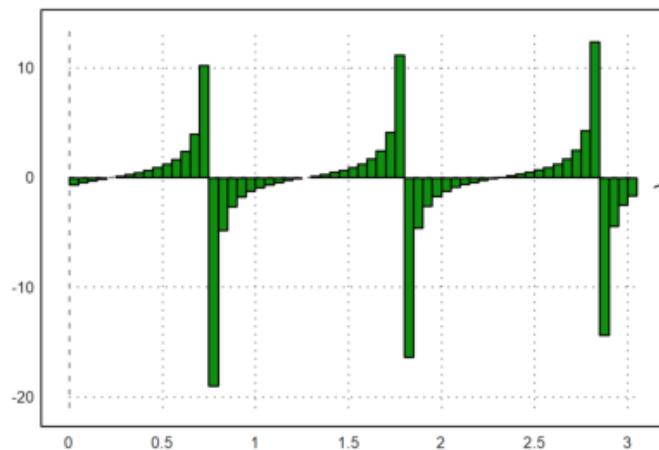
```
>$showev('integrate(f(x),x))
```

$$\int \tan(3x - 7) dx = \frac{\log \sec(3x - 7)}{3}$$

```
>$showev('integrate(f(x),x,pi,2*pi))
```

$$\int_{\pi}^{2\pi} \tan(3x - 7) dx = -\frac{\log \cos 7}{3} - \frac{\log(-\frac{1}{\cos 7})}{3}$$

```
>x=0:0.05:pi-0.1; plot2d(x,f(x+0.03),>bar); plot2d("f(x)",pi,2*pi,>add):
```



## 3. Fungsi 3

```
>function f(x) &= (cos(x))*(sin((x)))^3; $f(x)
```

$$\cos x \sin^3 x$$

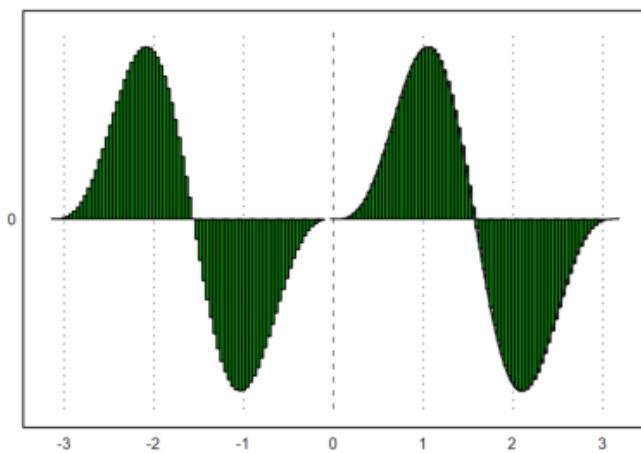
```
>$showev('integrate(f(x),x))
```

$$\int \cos x \sin^3 x dx = \frac{\sin^4 x}{4}$$

```
>$showev('integrate(f(x),x,0,2*pi))
```

$$\int_0^{2\pi} \cos x \sin^3 x \, dx = 0$$

```
>x=-pi:0.04:pi; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,pi,>add):
```



#### 4. Fungsi 4

```
>function f(x) &= (x^3 * (3+x^2)^(1/3)); $f(x)
```

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

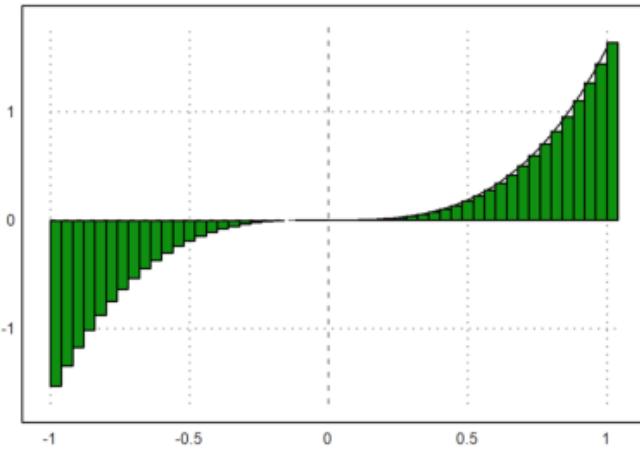
```
>$showev('integrate(f(x),x))
```

$$\int x^3 (x^2 + 3)^{\frac{1}{3}} \, dx = \frac{3 (x^2 + 3)^{\frac{7}{3}}}{14} - \frac{9 (x^2 + 3)^{\frac{4}{3}}}{8}$$

```
>$showev('integrate(f(x),x,0,2))
```

$$\int_0^2 x^3 (x^2 + 3)^{\frac{1}{3}} \, dx = \frac{37^{\frac{4}{3}}}{8} + \frac{3^{\frac{13}{3}}}{56}$$

```
>x=-1:0.04:1; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```



## 5. Fungsi 5

```
>function f(x) &= sqrt(32+x^4); $f(x)
```

$$\sqrt{x^4 + 32}$$

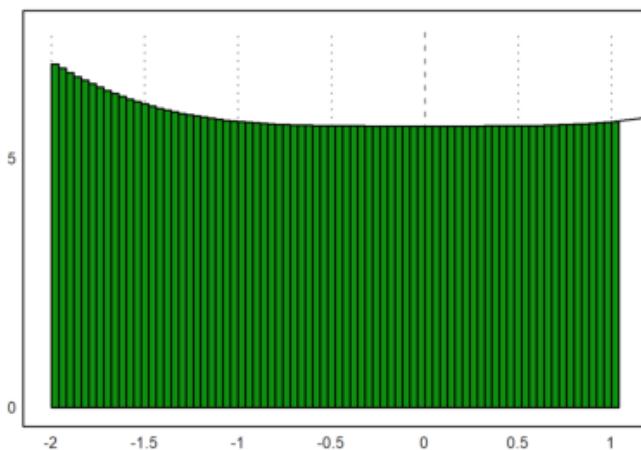
```
>$showev('integrate(f(x),x))
```

$$\int \sqrt{x^4 + 32} dx = \int \sqrt{x^4 + 32} dx$$

```
>$showev('integrate(f(x),x,1,3))
```

$$\int_1^3 \sqrt{x^4 + 32} dx = \int_1^3 \sqrt{x^4 + 32} dx$$

```
>x=-2:0.04:1; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",1,2,>add):
```

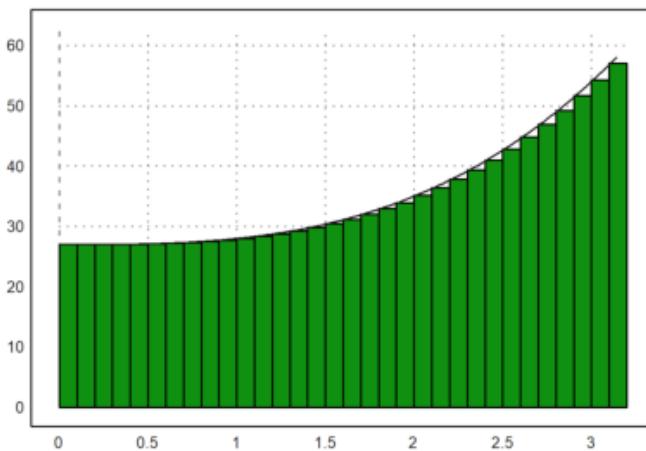


## 6. Fungsi 6

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
>function f(x) &= x^3+27; $f(x)
```

$$x^3 + 27$$

```
>x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,pi,>add):
```



```
>0.01*sum(f(x+0.01))
```

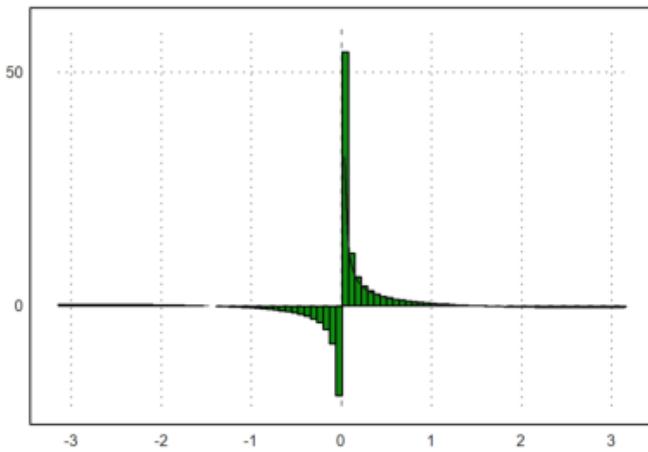
11.13155712

## 7. Fungsi 7

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
>function f(x) &= cos(x)/x; $f(x)
```

$$\frac{\cos x}{x}$$

```
>x=-pi:0.07:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,pi,>add):
```



```
>0.01*sum(f(x+0.01))
```

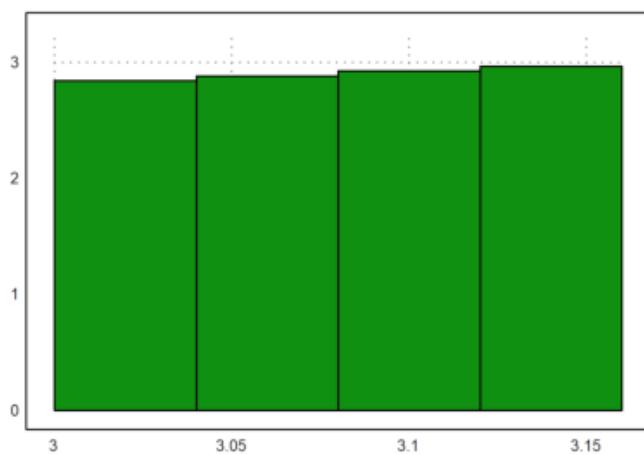
0.415163991256

### 8. Fungsi 8

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
>function f(x) &= sqrt(x^2-1); $f(x)
```

$$\sqrt{x^2 - 1}$$

```
>x=3:0.04:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,2,>add):
```



```
>0.01*sum(f(x+0.01))
```

0.11610107668

## Luas daerah dibatasi 2 kurva

1). Fungsi 1

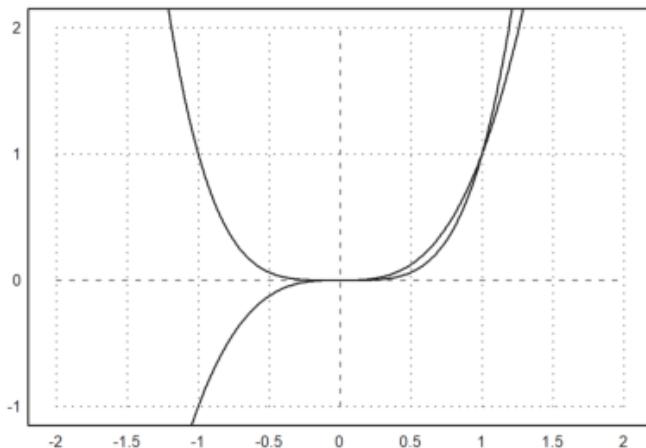
```
>function f(x) &= x^3; $f(x)
```

$$x^3$$

```
>function g(x) &= x; $g(x)
```

$$x$$

```
>plot2d(["x^4", "x^3"], -2, 2, -1, 2);
```



```
>function h(x) &= f(x)-g(x); $h(x)
```

$$x^3 - x$$

```
>$showev('integrate(h(x),x))
```

$$\int x^3 - x \, dx = \frac{x^4}{4} - \frac{x^2}{2}$$

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

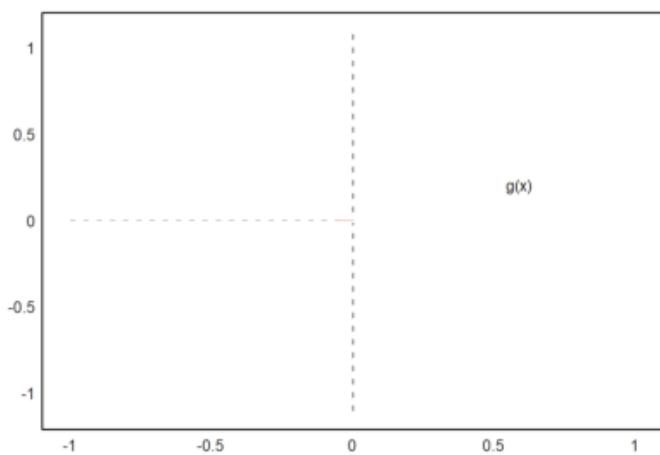
$$[x = -1, x = 1, x = 0]$$

```
>$showev('integrate(h(x),x,0,1)) // menghitung luas daerah yang dibatasi 2 kurva
```

$$\int_0^1 x^3 - x \, dx = -\frac{1}{4}$$

Arsiran daerah yang dibatasi kurva  $f(x)$  dan  $g(x)$  sebagai berikut:

```
>x=-1:0.01:1; plot2d(x,f(x),>bar,>filled,style="--",fillcolor=orange,>grid); plot2d(x,g(x),
```



## 2). Fungsi 2

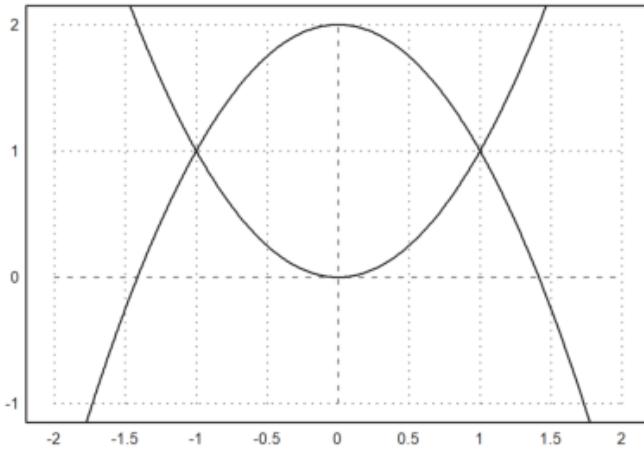
```
>function f(x) &= x^3+1; $f(x)
```

$$x^3 + 1$$

```
>function g(x) &= x^2; $g(x)
```

$$x^2$$

```
>plot2d([-x^2+2,x^2],-2,2,-1,2):
```



```
>function h(x) &= f(x)-g(x); $h(x)
```

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

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

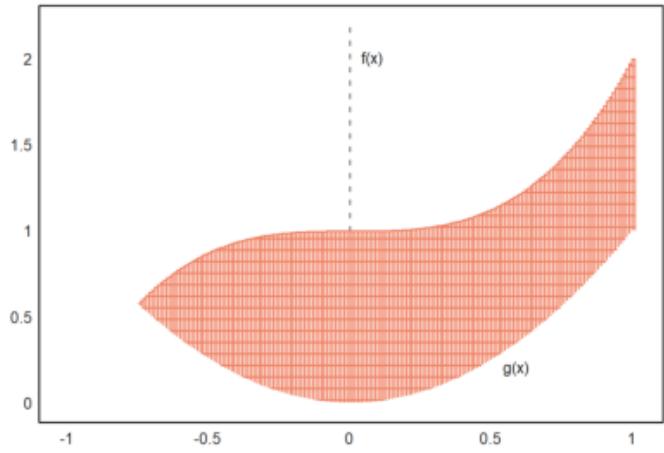
$$\left[ x = \frac{\frac{\sqrt{3}i}{2} - \frac{1}{2}}{9\left(\frac{\sqrt{23}}{23^{\frac{3}{2}}} - \frac{25}{54}\right)^{\frac{1}{3}}} + \left(\frac{\sqrt{23}}{23^{\frac{3}{2}}} - \frac{25}{54}\right)^{\frac{1}{3}} \left(-\frac{\sqrt{3}i}{2} - \frac{1}{2}\right) + \frac{1}{3}, x = \left(\frac{\sqrt{23}}{23^{\frac{3}{2}}} - \frac{25}{54}\right)^{\frac{1}{3}} \left(\frac{\sqrt{3}i}{2} - \frac{1}{2}\right) + \frac{-\frac{\sqrt{3}i}{2} - \frac{1}{2}}{9\left(\frac{\sqrt{23}}{23^{\frac{3}{2}}} - \frac{25}{54}\right)^{\frac{1}{3}}} + \frac{1}{3}, x = \right.$$

```
>$showev('integrate(h(x),x,-1,1)) // menghitung luas daerah yang dibatasi 2 kurva
```

$$\int_{-1}^1 x^3 - x^2 + 1 \, dx = \frac{4}{3}$$

Arsiran daerah yang dibatasi kurva  $f(x)$  dan  $g(x)$  sebagai berikut:

```
>x=-1:0.01:1; plot2d(x,f(x),>bar,>filled,style="--",fillcolor=orange,>grid); plot2d(x,g(x),
```



## Volume benda putar

---

Menghitung volume hasil perputaran kurva

$$m(x) = x^4 + 3$$

dari  $x=-1$  sampai  $x=0$ . Diputar terhadap sumbu-x.

Jawab:

```
>function m(x) &= x^4+3; $m(x)
```

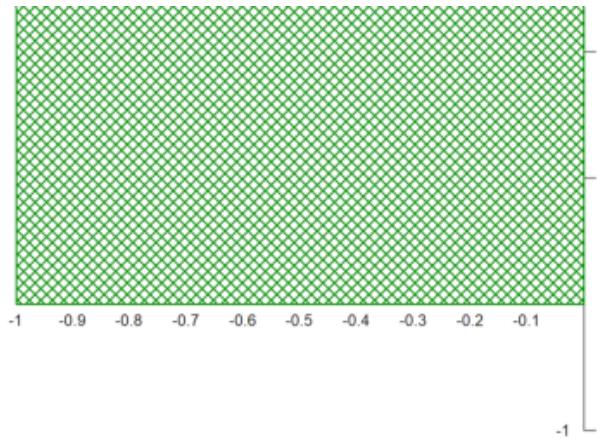
$$x^4 + 3$$

```
>$showev('integrate(pi*(m(x))^2,x,-1,0)) // Menghitung volume hasil perputaran m(x)
```

$$\pi \int_{-1}^0 (x^4 + 3)^2 dx = \frac{464\pi}{45}$$

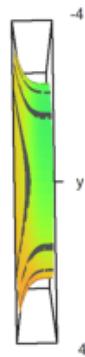
Daerah di bawah kurva yang akan dirotasi terhadap sumbu x sebagai berikut:

```
>plot2d("m(x)",-1,0,-1,2,grid=7,>filled, style="/\"):
```



Hasil perputaran  $m(x)$  terhadap sumbu x sebagai berikut:

```
>plot3d("m(x)",-1,0,-1,1,>rotate,angle=6.3,>hue,>contour,color=redgreen,height=11):
```



## Menghitung panjang kurva

---

Menghitung panjang kurva

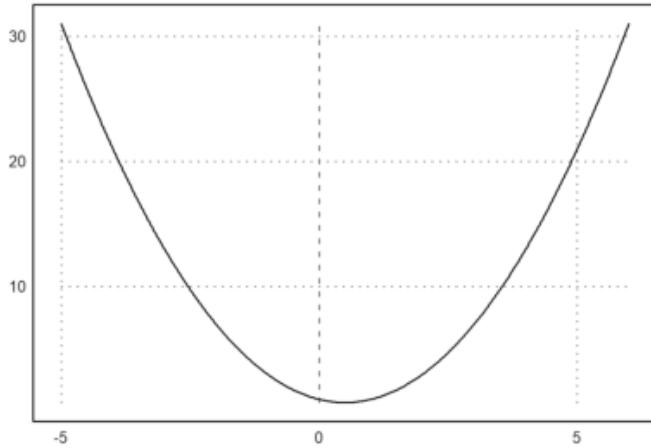
$$y = x^2 - x + 1$$

dari  $x=1$  sampai  $x=3$ .

```
>function d(x) &= x^2-x+1; $d(x)
```

$$x^2 - x + 1$$

```
>plot2d("d(x)", -5, 6); // gambar kurva d(x)
```



```
>$showev('limit((d(x+h)-d(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2 - h}{h} = 2x - 1$$

```
>function dd(x) &= limit((d(x+h)-d(x))/h,h,0); $dd(x)
```

$$2x - 1$$

```
>function q(x) &= ((dd(x))^2); $q(x)
```

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

```
>$showev('integrate(sqrt(1+q(x)),x,1,3)) // menghitung panjang kurva
```

$$\int_1^3 \sqrt{(2x-1)^2 + 1} dx = \frac{\operatorname{asinh} 5 + 5\sqrt{26}}{4} - \frac{\operatorname{asinh} 1 + \sqrt{2}}{4}$$

Jadi, panjang kurva

$$y = x^2 - x + 1$$

dari x=0 sampai x=4 adalah

$$S = \frac{\operatorname{asinh} 5 + 5\sqrt{26}}{4} - \frac{\operatorname{asinh} 1 + \sqrt{2}}{4}.$$

## Barisan dan Deret

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan (menggunakan titik dua ":");
- menggunakan perintah "sequence" dan rumus barisan (suku ke -n);
- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create\_list" atau "makelist" untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana
```

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

```
>1:2:30
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
```

```
>sum(1:2:30), sum(1/(1:2:30))
```

```
225
```

```
2.33587263431
```

```
>$' sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n), simpsum=true)) // simpsum:menghitung deret
```

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

```
>$' sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf), simpsum=true))
```

$$\sum_{k=0}^{\infty} \frac{1}{3^k + k} = \sum_{k=0}^{\infty} \frac{1}{3^k + k}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$' sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai e
```

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

```
>$' sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} = - \sum_{x=1}^{\infty} \frac{(-1)^x}{x}$$

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))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{2k-1} = \sum_{k=1}^{\infty} \frac{(-1)^k}{2k-1}$$

```
>$ev(sum(1/n!, n, 0, inf),simpsum=true)
```

$$\sum_{n=0}^{\infty} \frac{1}{n!}$$

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
>&assume(abs(x)<1); $' sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forget
```

$$a \sum_{k=0}^{\infty} x^k = \frac{a}{1-x}$$

Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1. **Deret Taylor**

---

Deret Taylor suatu fungsi f yang diferensiabel sampai tak hingga di sekitar  $x=a$  adalah:

$$f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^k f^{(k)}(a)}{k!}.$$

```
>$' e^x=taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai suku ke-11
```

$$e^x = \frac{x^{10}}{3628800} + \frac{x^9}{362880} + \frac{x^8}{40320} + \frac{x^7}{5040} + \frac{x^6}{720} + \frac{x^5}{120} + \frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1$$

```
>$' log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1
```

$$\log x = x - \frac{(x-1)^{10}}{10} + \frac{(x-1)^9}{9} - \frac{(x-1)^8}{8} + \frac{(x-1)^7}{7} - \frac{(x-1)^6}{6} + \frac{(x-1)^5}{5} - \frac{(x-1)^4}{4} + \frac{(x-1)^3}{3} - \frac{(x-1)^2}{2} - 1$$

---

---

## BAB 6

---

# PEKAN 8: MENGGUNAKAN EMT UNTUK GEOMETRI

[a4paper,10pt]article eumat

## Visualisasi dan Perhitungan Geometri dengan EMT

---

Raden Mas Farrel Maheswara Kusuma Dewa  
22305141042  
Matematika E

---

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan geometri tersebut disimpan di dalam file program "geometry.e", sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

```
>load geometry
```

Numerical and symbolic geometry.

### Fungsi-fungsi Geometri

---

Fungsi-fungsi untuk Menggambar Objek Geometri:

```
defaultd:=textheight()*1.5: nilai asli untuk parameter d  
setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang
```

koordinat

```
setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas  
sumbu-x dan y adalah -r sd r
```

```

plotPoint (P, "P"): menggambar titik P dan diberi label "P"
plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label
"AB" sejauh d

```

```

plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d
plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"
plotLabel (label, P, V, d): menuliskan label pada posisi P

```

#### Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

```

turn(v, phi): memutar vektor v sejauh phi
turnLeft(v): memutar vektor v ke kiri
turnRight(v): memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang vektor v dan w.
lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh.

```

$ax+by=c$ .

```

lineWithDirection(A,v): garis melalui A searah vektor v
getLineDirection(g): vektor arah (gradien) garis g
getNormal(g): vektor normal (tegak lurus) garis g
getPointOnLine(g): titik pada garis g
perpendicular(A, g): garis melalui A tegak lurus garis g
parallel (A, g): garis melalui A sejajar garis g
lineIntersection(g, h): titik potong garis g dan h
projectToLine(A, g): proyeksi titik A pada garis g
distance(A, B): jarak titik A dan B
distanceSquared(A, B): kuadrat jarak A dan B
quadrance(A, B): kuadrat jarak A dan B
areaTriangle(A, B, C): luas segitiga ABC
computeAngle(A, B, C): besar sudut <ABC
angleBisector(A, B, C): garis bagi sudut <ABC
circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r
getCircleCenter(c): pusat lingkaran c
getCircleRadius(c): jari-jari lingkaran c
circleThrough(A,B,C): lingkaran melalui A, B, C
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan 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 atur tiga poin dan plot.

```
>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 geometri meliputi fungsi untuk membuat garis dan lingkaran. Format untuk garis adalah [a, b, c], yang merepresentasikan garis dengan persamaan  $ax + by = c$ .

```
>lineThrough(B,C) // garis yang melalui B dan C
```

[ -1, 2, 2 ]

Hitung garis tegak lurus melalui A pada BC.

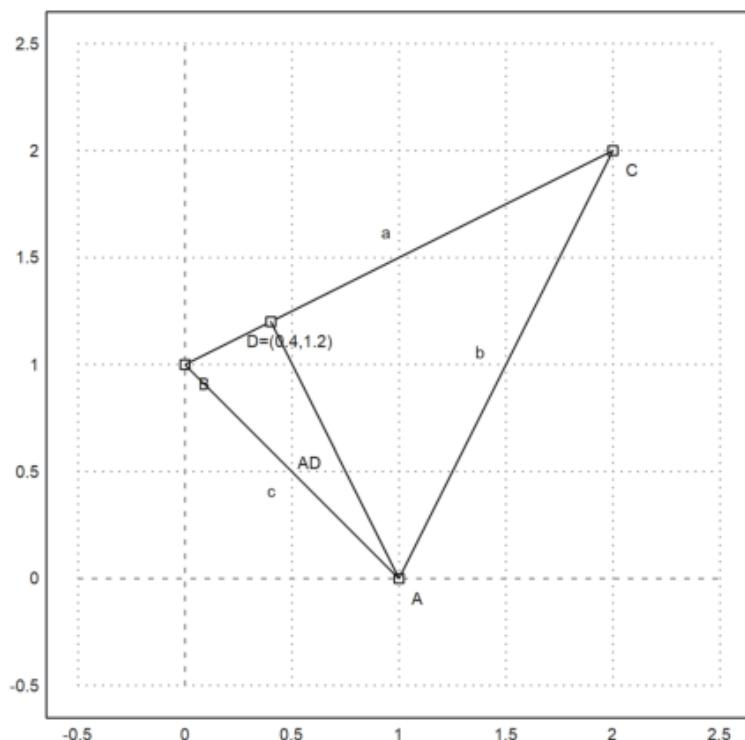
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan perpotongannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Plot itu.

```
>plotPoint(D,value=1); // koordinat D ditampilkan  
>aspect(1); plotSegment(A,D); // tampilkan semua gambar hasil plot...()
```



Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2}AD \cdot BC.$$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D) , BC=norm(B-C)
```

1.5

Cara lain menghitung 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

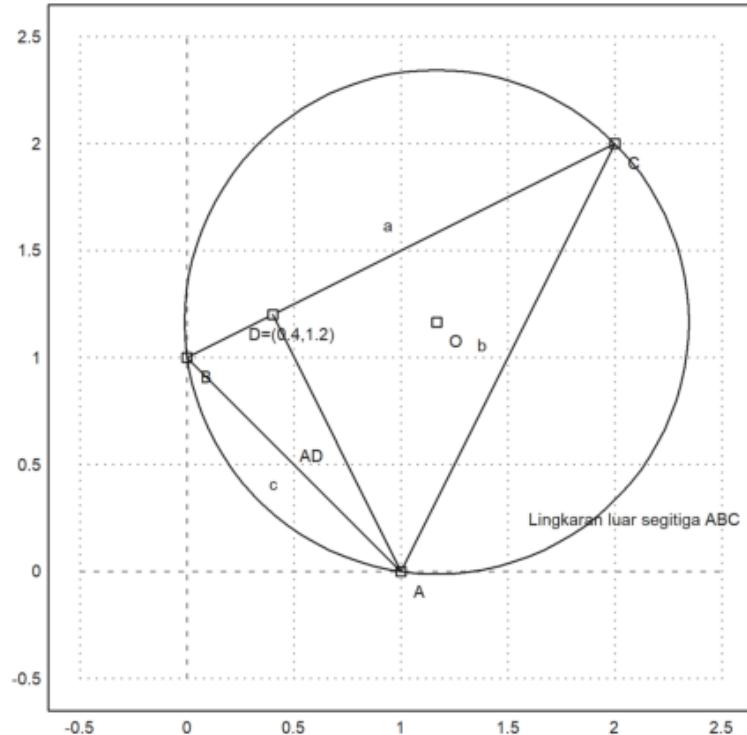
Sudut di C

```
>degsprint(computeAngle(B,C,A))
```

$36^\circ 52' 11.63''$

Sekarang lingkaran sirkit 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
```

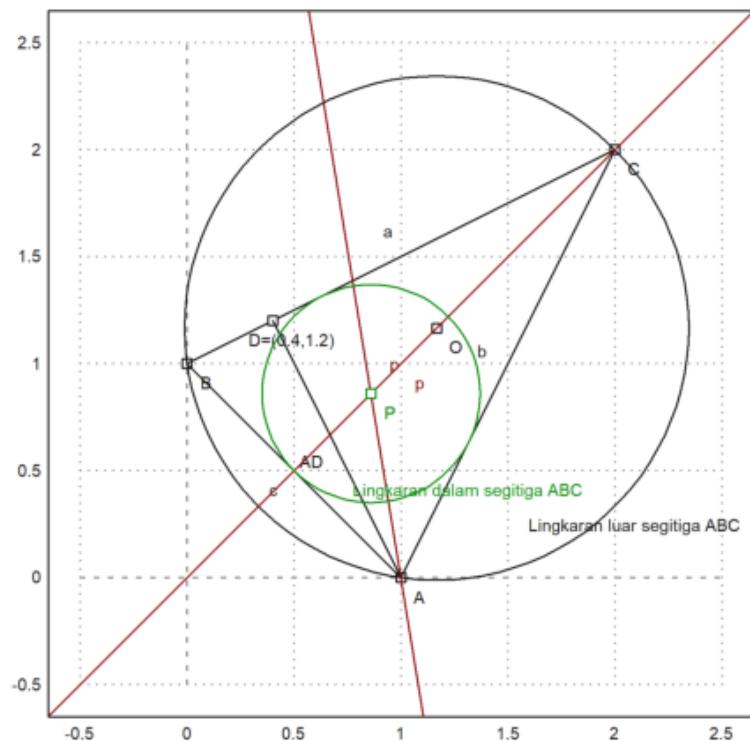
[0.86038, 0.86038]

Tambahkan semua ke 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.
2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.
3. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.
4. Gambar jari-jari lingkaran dalam.

Jawab:

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.  
Titik singgung garis BC dengan lingkaran dalam.

```
>s=lineThrough(B,C)
```

```
[-1, 2, 2]
```

```
>m=circleWithCenter(P,r)
```

```
[0.86038, 0.86038, 0.509654]
```

```
>S=lineCircleIntersections(s,m)
```

```
[0.632456, 1.31623]
```

Titik singgung garis AC dengan lingkaran dalam.

```
>p=lineThrough(A,C)
```

```
[-2, 1, -2]
```

```
>Q=lineCircleIntersections(p,m)
```

```
[1.31623, 0.632456]
```

Titik singgung garis AB dengan lingkaran dalam.

```
> q=lineThrough(A,B)
```

```
[-1, -1, -1]
```

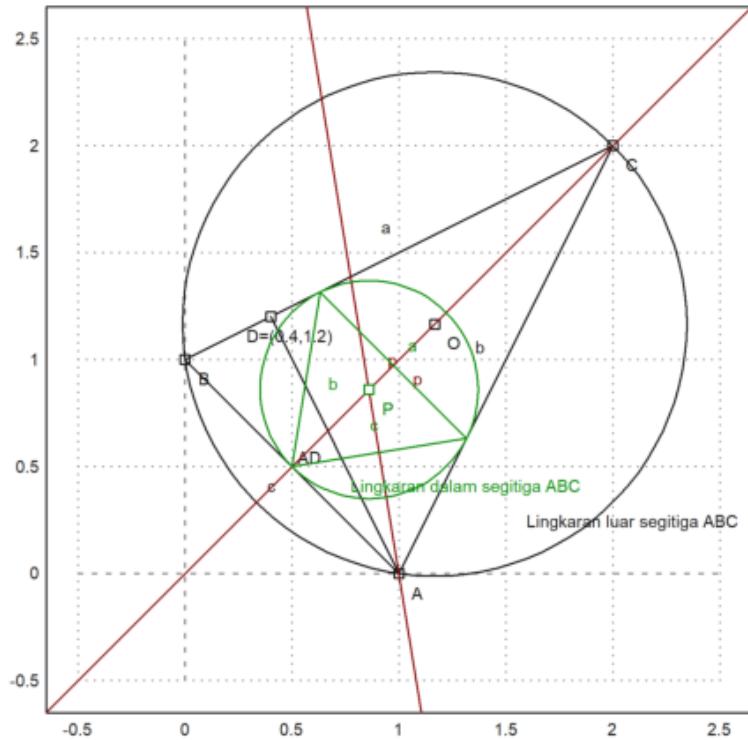
```
>L=lineCircleIntersections(q,m)
```

```
[0.5, 0.5]
```

Jadi titik singgung lingkaran dalam dengan sisi-sisi segitiga adalah  $(0.632456, 1.31623)$ ,  $(1.31632, 0.632456)$ ,  $(0.5, 0.5)$ .

2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.

```
>plotSegment(S,Q,"a");
>plotSegment(S,L,"b");
>plotSegment(L,Q,"c");
```



3. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

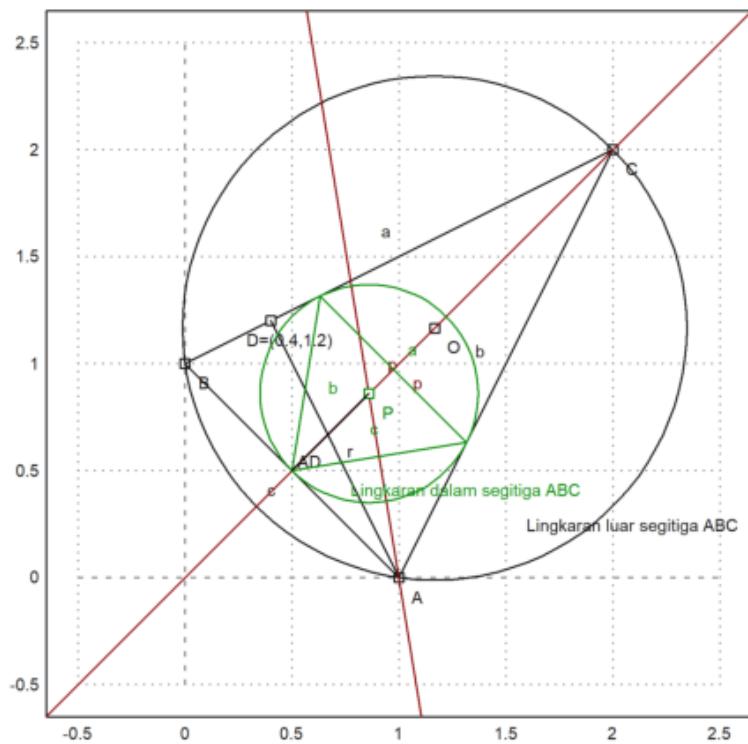
```
> P, r
```

```
[0.86038, 0.86038]
0.509653732104
```

```
>k=angleBisector(A,B,C)
```

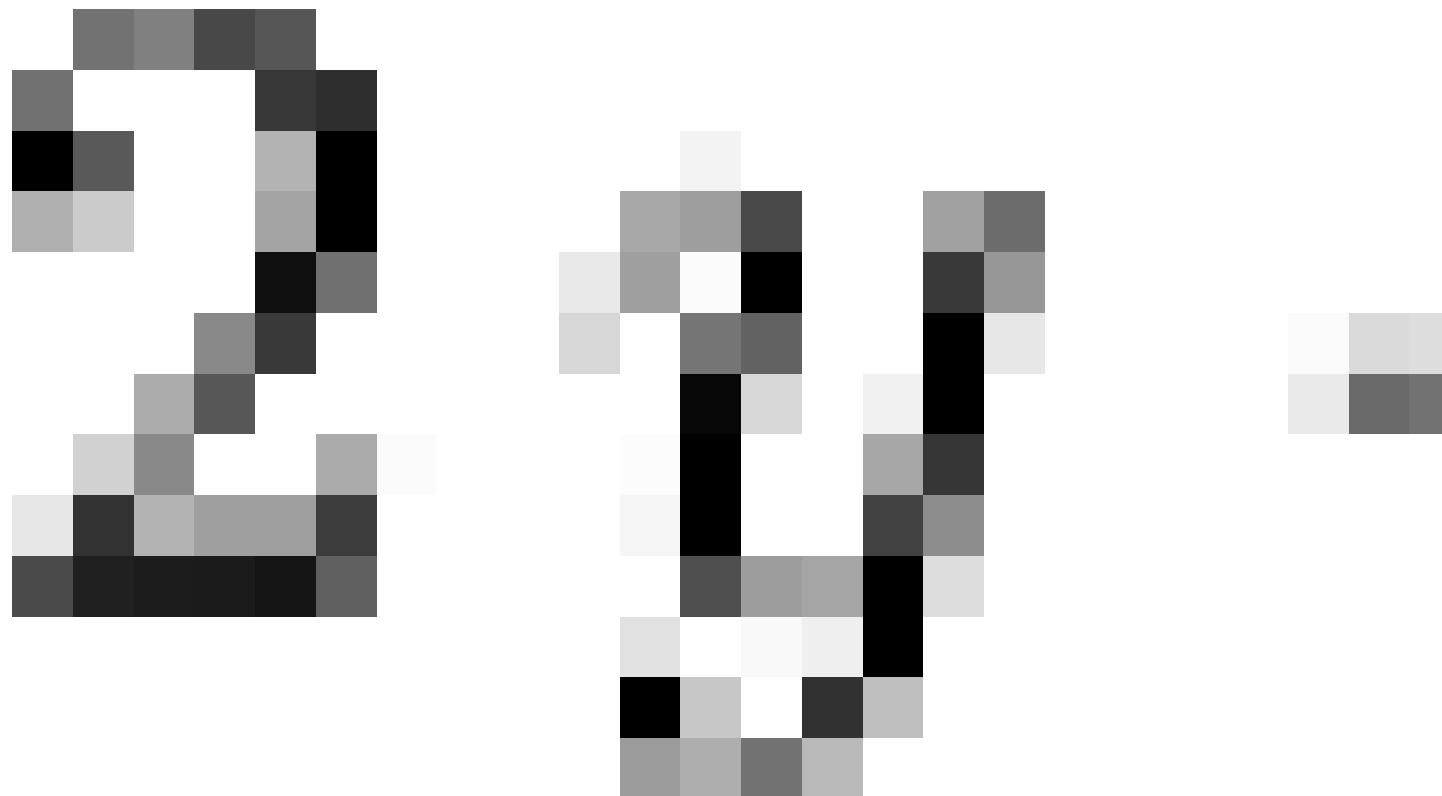
```
[-0.264911, -1.63246, -1.63246]
```

```
>color(3); plotLine(k):
```



4. Gambar jari-jari lingkaran dalam.

```
>plotSegment(P,L,"r") :
```



## Contoh 2: Geometri Smbolik

Kita dapat menghitung geometri tepat dan simbolis menggunakan Maxima.

Geometri file.e menyediakan fungsi yang sama (dan lebih banyak lagi) 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 garis dan lingkaran bekerja seperti fungsi Euler, tetapi menyediakan penghitungan simbolik.

```
>c &= lineThrough(B,C) // c=BC
```

```
[ - 1, 2, 2]
```

Kita bisa mendapatkan persamaan untuk sebuah garis dengan mudah.

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$2y - x = 2$$

$$\left[ y = \frac{x}{2} + 1 \right]$$

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
```

$$(x_1 - 1)y - x y_1 = -y_1$$

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

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

```
>q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

$$\begin{matrix} 2 & 6 \\ [-, -] \\ 5 & 5 \end{matrix}$$

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\left[ \frac{2}{5}, \frac{6}{5} \right]$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[ \frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}} \right]$$

```
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

$$\frac{5}{3\sqrt{2}}$$

1.178511301977579

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 ga
```

$$\left[ \frac{\sqrt{2}\sqrt{5} + 2}{6}, \frac{\sqrt{2}\sqrt{5} + 2}{6} \right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

[0.86038, 0.86038]

## Garis dan Lingkaran yang Berpotongan

---

Tentu saja, kita juga bisa memotong garis dengan lingkaran, dan lingkaran dengan lingkaran.

```
>A &:= [1,0]; c=circleWithCenter(A,4);
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l);
```

Perpotongan garis dengan lingkaran mengembalikan dua titik dan jumlah titik perpotongan.

```
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2,
```

```
[4.64575, -1.64575]
[-0.645751, 3.64575]
```

```
>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 bsuusr yang sama adalah sama besar.

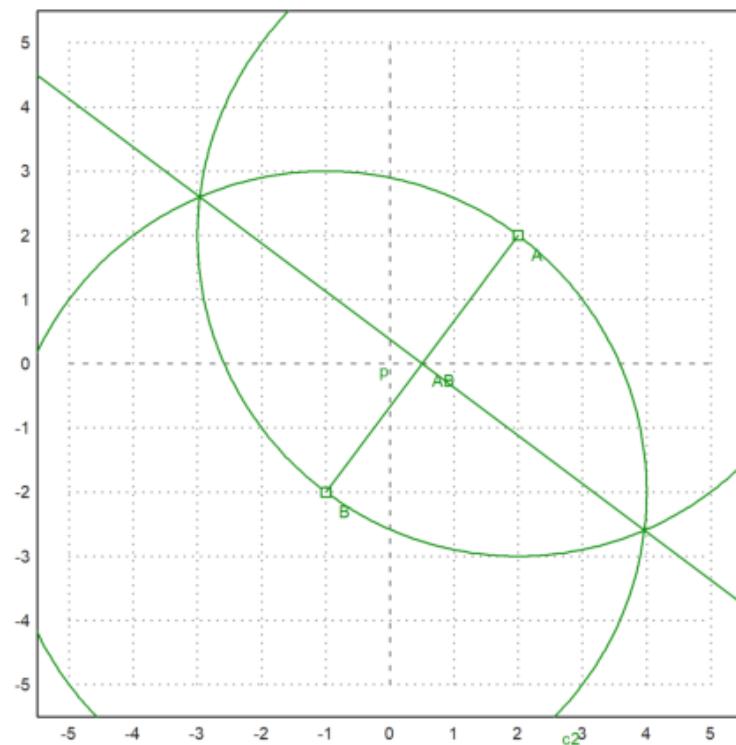
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>degsprint(computeAngle(P1,C,P2))
```

69°17' 42.68''

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>degsprint(computeAngle(P1,C,P2))
```

69°17' 42.68''

```
>insimg;
```

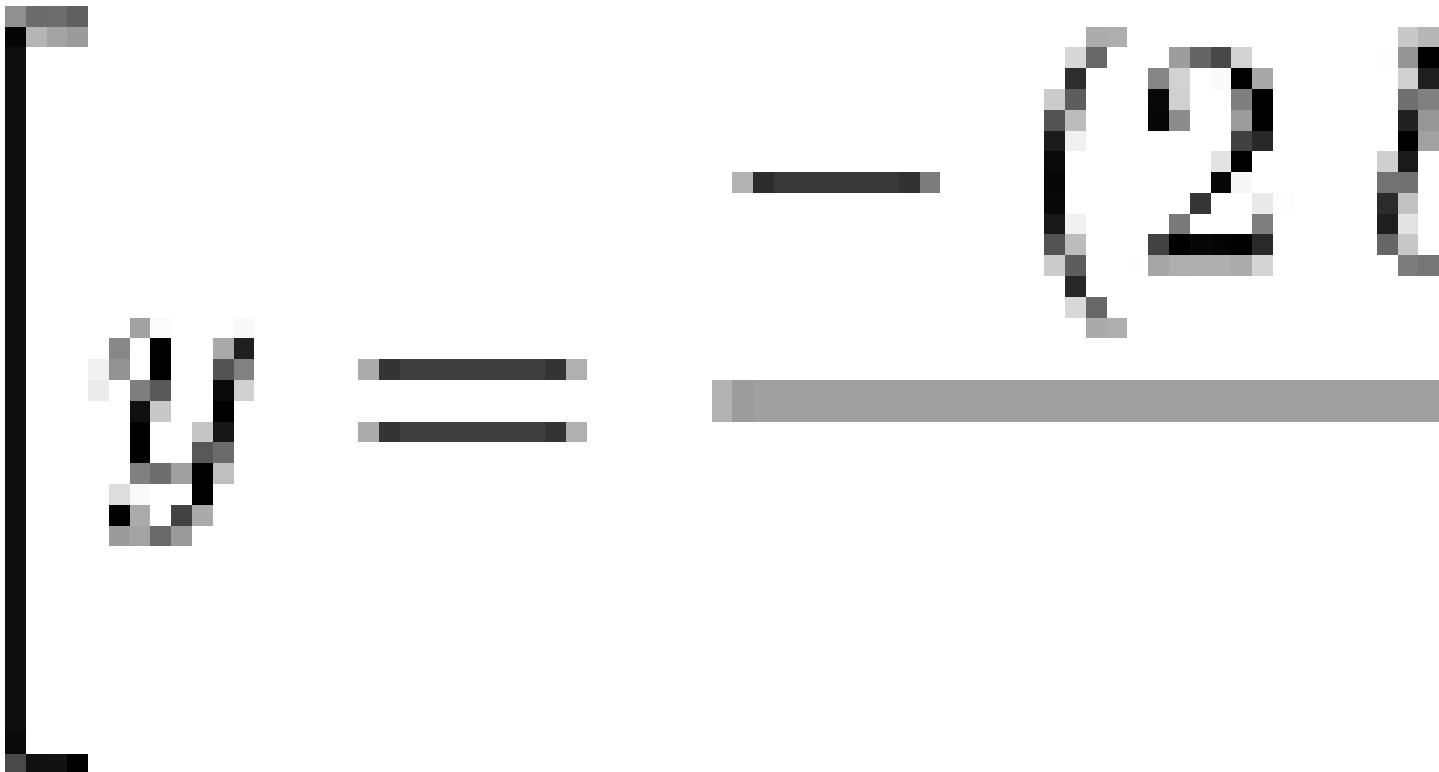


## Garis Sumbu

Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B.
2. Gambar lingkaran dengan pusat B melalui A.
3. Tarik garis 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 persimpangan cukup terlibat. Tapi kita bisa menyederhanakan, jika kita menyelesaikan 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 tengah tegak lurus, yang dihitung dengan cara yang sama sekali berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[ y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

```
>h &= getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
```

$$\left[ y = \frac{(b_2 - a_2)x - a_1b_2 + a_2b_1}{b_1 - a_1} \right]$$

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:

$$L = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{dengan } s = (a+b+c)/2.$$

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:

$$x^2 + y^2 = b^2, \quad (x-a)^2 + y^2 = c^2.$$

```
>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]; $ysol
```

```

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]; $ysol ...
^
```

Kita mendapatkan formula Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H(a, b, [1, 0, 4]) = \frac{|a| |ysol|}{2}$$

Tentu saja, setiap segitiga persegi panjang adalah kasus yang terkenal.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

```

Variable or function ysol not found.
Try "trace errors" to inspect local variables after errors.
H:
useglobal; return abs(a)*abs(ysol)/2
Error in:
H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...
^
```

Dan jelas juga, bahwa ini adalah segitiga dengan luas maksimal dan kedua sisinya 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7); // Kurva luas segitiga sengan panjang sisi 3, 4, x
```

```

Variable or function ysol not found.
Error in expression: 3*abs(ysol)/2
%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 elips.

```
>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 membuat persamaan seperti ini

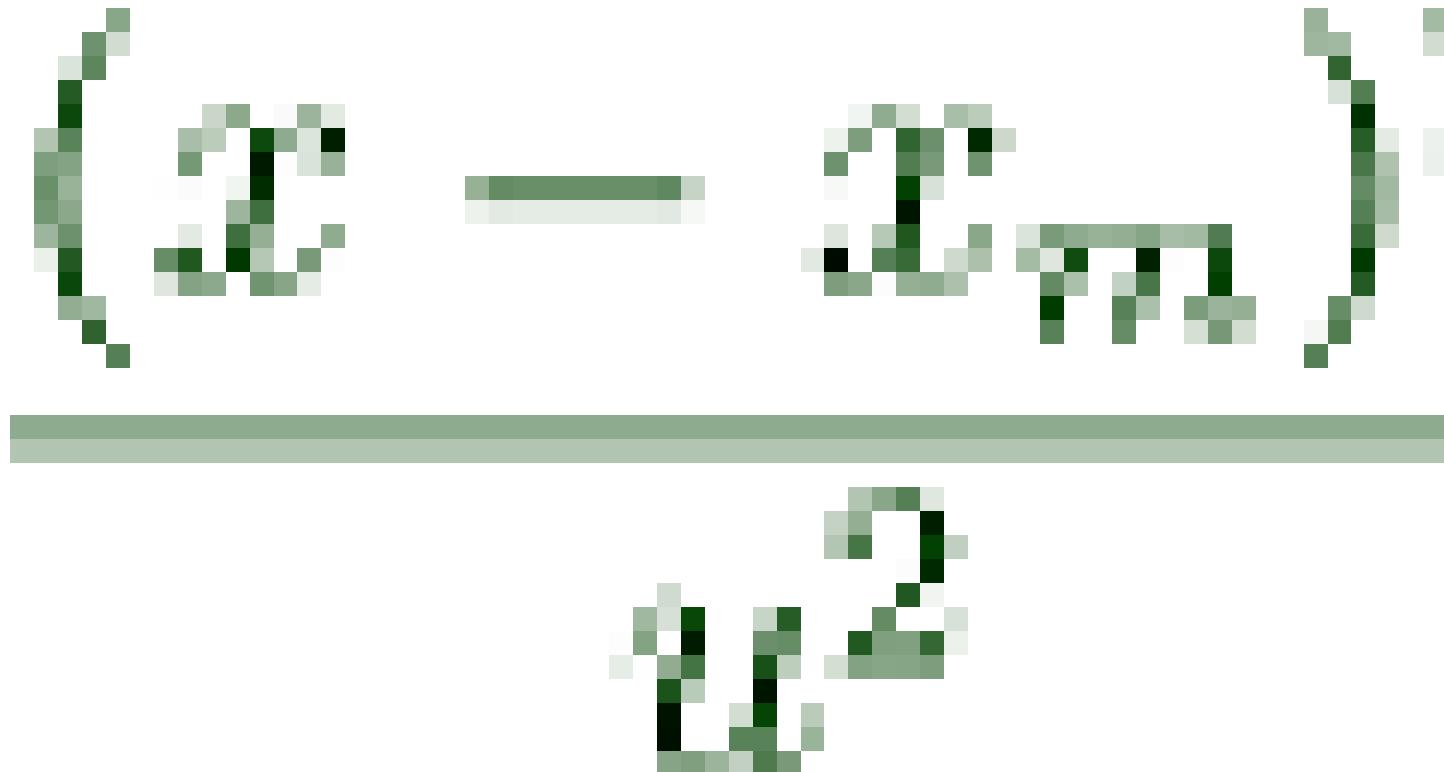
```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); $fy(a,c,d)
```

0

0

Sekarang kita bisa menggambar setnya. Sisi  $b$  bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan elips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```



Kita dapat memeriksa persamaan umum elips ini, yaitu.

$$\frac{(x - x_m)^2}{u^2} + \frac{(y - y_m)^2}{v^2} = 1,$$

di mana  $(x_m, y_m)$  adalah pusat, dan  $u$  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 melihat bahwa tinggi dan luas segitiga adalah maksimal untuk  $x = 0$ . Jadi luas segitiga dengan  $a + b + c = d$  adalah maksimal, jika sama sisi. Kami ingin mendapatkan ini secara analitis.

```
> eqns &= [diff(H(a,b,d-(a+b))^2,a)=0, diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

$$\left[ \frac{a y_{sol}^2}{2} = 0, 0 = 0 \right]$$

Kami mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusi  $a=b=c=d/3$ .

```
> $solve(eqns, [a,b])
```

$$[[a = 0, b = \%r_1]]$$

Ada juga metode Lagrange, memaksimalkan  $H(a,b,c)^2$  terhadap  $a+b+d=d$ .

```
> & solve([diff(H(a,b,c)^2,a)=la, diff(H(a,b,c)^2,b)=la, ...
>     diff(H(a,b,c)^2,c)=la, a+b+c=d], [a,b,c,la])
```

Maxima said:

```
diff: second argument must be a variable; found [1,0,4]
-- an error. To debug this try: debugmode(true);
```

Error in:

```
... la,      diff(H(a,b,c)^2,c)=la, a+b+c=d], [a,b,c,la]) ...  
^
```

Kita bisa membuat plot situasinya.

Pertama, 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 poinnya.

```
>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 segmennya.

```
>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 tengah tegak lurus di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan bagian tengah dari keliling.

```
>U &= lineIntersection(h,g);
```

Kita 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, kita mendapatkan rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk radius. Kita dapat memeriksa, apakah ini benar dengan Maxima. Maxima akan menfaktorkannya hanya jika kita mengkuadratkannya.

```
>$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 segitiga yang tidak sama sisi. Ini adalah garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk pusat ortosentrum, sirkumenter, pusat massa, titik Exeter, dan pusat lingkaran sembilan titik segitiga.

Untuk demonstrasi, kami menghitung dan memplot garis Euler dalam segitiga.

Pertama, kami menentukan sudut segitiga di Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolik.

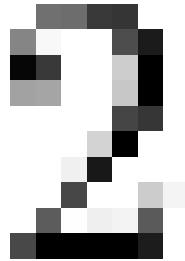
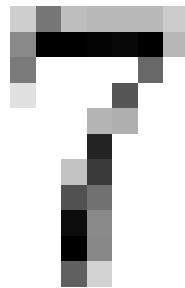
```
>A:=[-1,-1]; B:=[2,0]; C:=[1,2];
```

Untuk memplot objek geometris, kami menyiapkan area plot, dan menambahkan poin ke dalamnya. 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 dari sisi c.

```
>c &= lineThrough(A,B)
```

$$[-1, 3, -2]$$

Dan juga dapatkan rumus untuk baris ini.

```
>$getLineEquation(c,x,y)
```

$$3y - x = -2$$

Untuk bentuk Hesse, kita perlu menentukan 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 menghitung sirkit 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
```

$$\left[\frac{3}{14}, \frac{5}{14}\right]$$

Plot lingkaran dan pusatnya. Cu dan U adalah simbolik. 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
```

$$\left[ \frac{11}{7}, \frac{2}{7} \right]$$

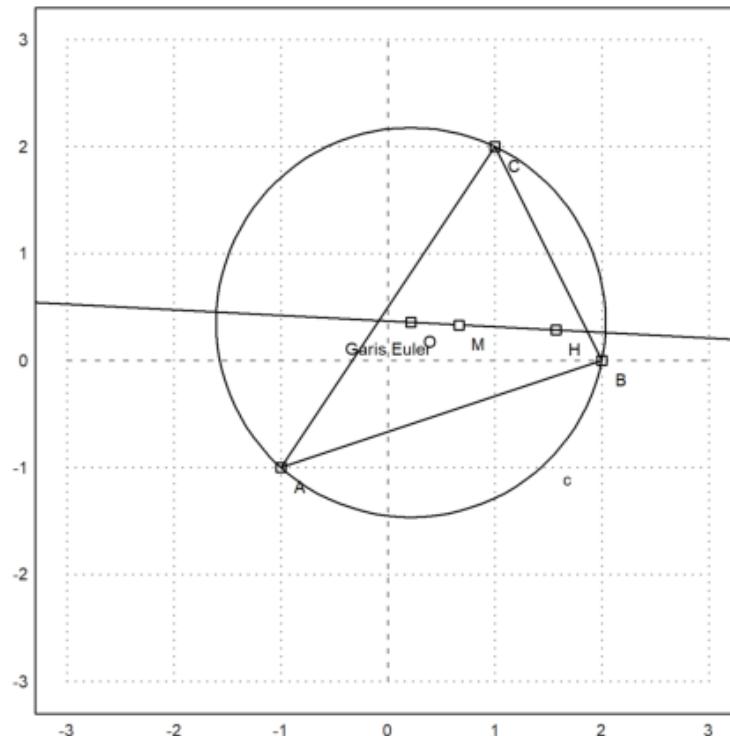
Sekarang kita dapat menghitung garis Euler dari segitiga tersebut.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$-\frac{19y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan ke plot kita.

```
>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 mengatakan bahwa  $MH=2*MO$ . Kita perlu menyederhanakan dengan radcan untuk mencapai ini.

```
>$distance(M, H) / distance(M, O) | radcan
```

2

Fungsinya 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 lingkaran tidak terlalu bagus.

```
>Q &= lineIntersection(angleBisector(A, C, B), angleBisector(C, B, A)) | radcan; $Q
```

$$\left[ \frac{\left(2^{\frac{3}{2}} + 1\right) \sqrt{5} \sqrt{13} - 15 \sqrt{2} + 3}{14}, \frac{(\sqrt{2} - 3) \sqrt{5} \sqrt{13} + 5 2^{\frac{3}{2}} + 5}{14} \right]$$

Mari kita hitung juga ekspresi jari-jari lingkaran yang tertulis.

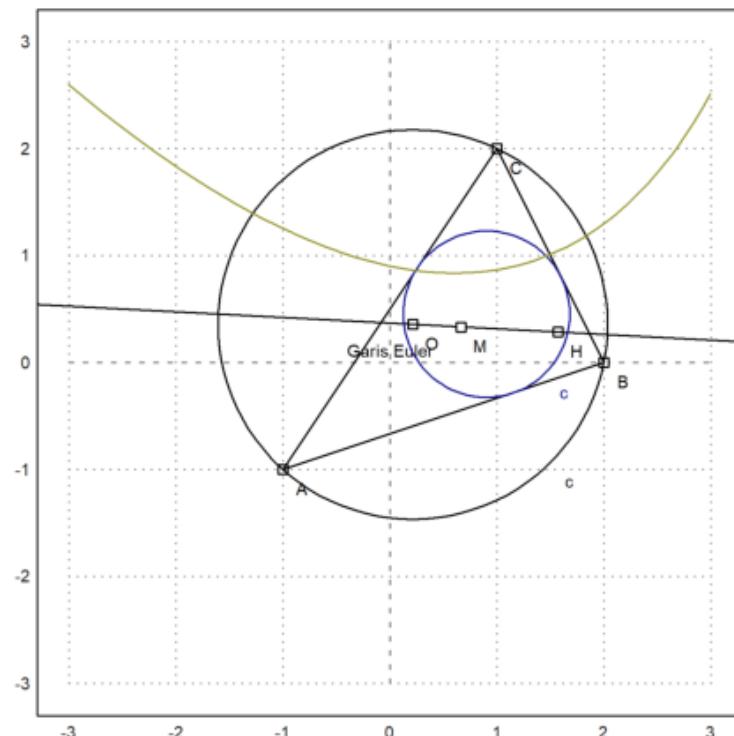
```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{(-41\sqrt{2} - 31)\sqrt{5}\sqrt{13} + 115\sqrt{2} + 614}}{7\sqrt{2}}$$

```
>LD &= circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke plot.

```
>color(4); plotCircle(LD());
```



## Parabola

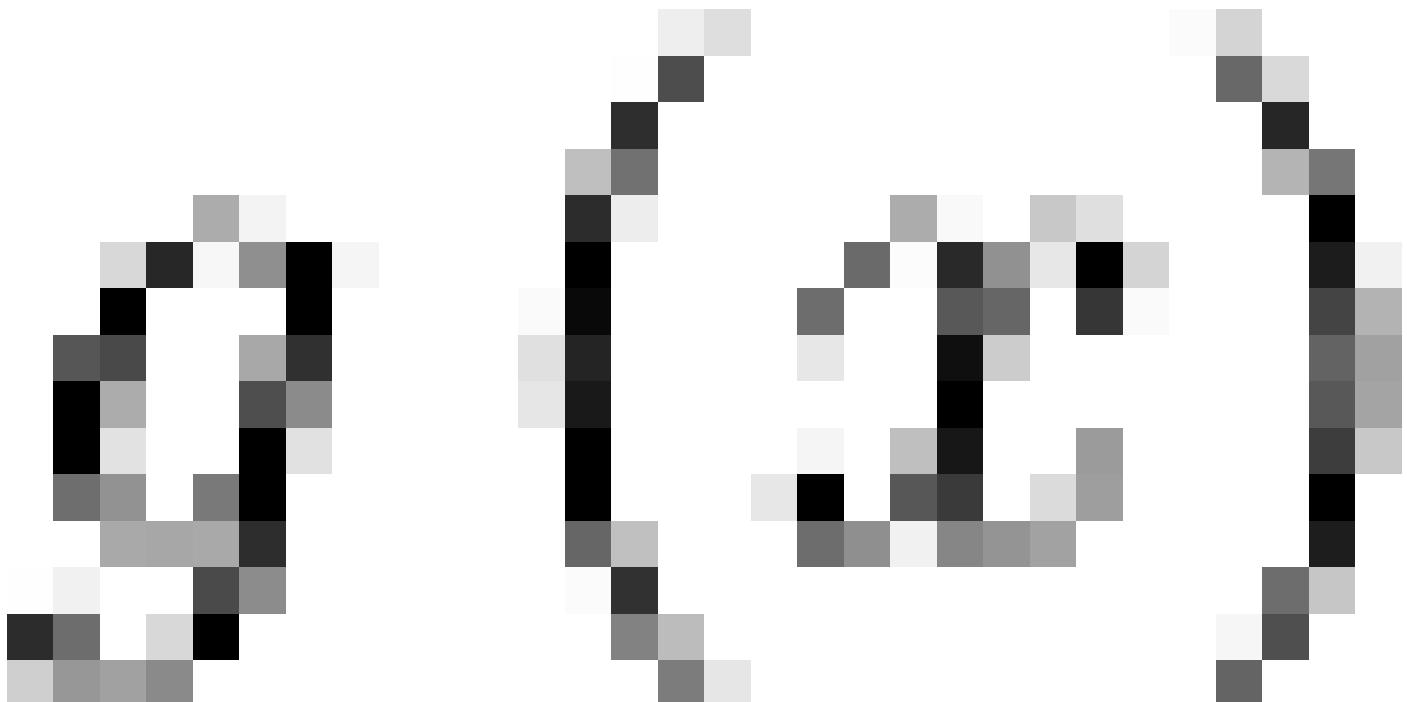
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3y - x + 2}{\sqrt{10}} - \sqrt{(2-y)^2 + (1-x)^2} = 0$$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```



Ini seharusnya menjadi beberapa fungsi, tetapi pemecah default Maxima dapat menemukan solusi hanya, jika persamaan kita kuadratkan. 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

maxima: akar[1]

Menambahkan solusi pertama ke pertunjukkan plot, bahwa itu memang jalan yang kita cari. Teori mengatakan kepada 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
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

$$2.135605779339061$$

```
>U &= projectToLine(T, lineThrough(A,B)); $U // proyeksi T pada garis AB
```

$$\left[ \frac{80 - 3\sqrt{11}\sqrt{70}}{10}, \frac{20 - \sqrt{11}\sqrt{70}}{10} \right]$$

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jarak T ke AB
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

$$2.135605779339061$$

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

## Contoh 5: Trigonometri Rasional

Ini terinspirasi oleh ceramah N.J.Wildberger. Dalam bukunya "Proporsi Agung", Wildberger mengusulkan untuk menggantikan pengertian klasik tentang jarak dan sudut dengan kuadransi 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 dapat dilakukan dengan kertas dan pensil saja. 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 mengevaluasi ke pendekatan numerik saja.

```
>load geometry;
```

Untuk pendahuluan pertama, kami menggunakan segitiga persegi panjang dengan proporsi Mesir terkenal 3, 4 dan 5. Perintah berikut adalah perintah Euler untuk memplot 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"); ...
>insimsg(30);
```



Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

di mana wa adalah sudut di A. Cara biasa untuk menghitung sudut ini, adalah dengan melakukan invers dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak secara perkiraan.

```
>wa := arcsin(3/5); degprint(wa)
```

36°52'11.63''

Trigonometri rasional mencoba menghindari hal ini.

Pengertian pertama dari trigonometri rasional adalah kuadran, yang menggantikan jarak. Faktanya, itu hanya kuadrat jarak. Berikut ini, a, b, dan c menunjukkan kuadran sisi-sisinya.

Teorema Pythagoras menjadi  $a+b=c$  lalu.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

$$25 = 25$$

Gagasan kedua dari trigonometri rasional adalah penyebarannya. Spread mengukur bukaan antar baris. Ini adalah 0, jika garis sejajar, dan 1, jika garis persegi panjang. Ini adalah kuadrat dari 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 segitiga persegi panjang mana pun dengan satu sudut di A.

```
>sa &= a/c; $sa
```

$$\frac{9}{25}$$

Ini lebih mudah dihitung daripada sudut, tentu saja. Tetapi Anda kehilangan properti yang sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengubah nilai perkiraan sudut wa menjadi sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

9/25

Hukum cosinus dari trigonometri klasik diterjemahkan menjadi "hukum silang" berikut.

$$(c + b - a)^2 = 4bc(1 - s_a)$$

Di sini a, b, dan c adalah kuadran dari sisi-sisi segitiga, dan sa adalah sebaran di sudut A. Sisi a, seperti biasa, berlawanan dengan sudut A.

Hukum ini diimplementasikan dalam file geometry.e yang kami muat ke Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

$$\left[ \left( bb - aa + \frac{7}{6} \right)^2, \left( bb - aa + \frac{7}{6} \right)^2, \left( bb - aa + \frac{5}{3\sqrt{2}} \right)^2 \right] = \left[ \frac{14 bb (1 - saa)}{3}, \frac{14 bb (1 - saa)}{3}, \frac{5 2^{\frac{3}{2}} bb (1 - saa)}{3} \right]$$

Dalam kasus kami, kita mendapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024 = 1024$$

Mari kita gunakan crosslaw ini untuk mencari sebaran di A. Untuk melakukan ini, kita menghasilkan crosslaw untuk kuadran a, b, dan c, dan menyelesaiannya untuk sebaran yang tidak diketahui sa.

Anda dapat melakukan ini dengan tangan dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasilnya, kami sudah mendapatkannya.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

$$1024 = 1600 (1 - x)$$

$$\left[ x = \frac{9}{25} \right]$$

Kami sudah tahu ini. Definisi penyebaran adalah kasus khusus dari hukum lintas hukum.

Kita juga bisa menyelesaikan ini untuk umum a, b, c. Hasilnya adalah rumus yang menghitung sebaran sudut segitiga berdasarkan kuadran ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[ \left[ \frac{168 bb x + 36 bb^2 + (-72 aa - 84) bb + 36 aa^2 - 84 aa + 49}{36}, \frac{168 bb x + 36 bb^2 + (-72 aa - 84) bb + 36 aa^2 - 84 aa + 49}{36} \right], \dots \right]$$

Kita bisa membuat fungsi dari hasilnya. Fungsi seperti itu sudah ditentukan dalam file geometry.e Euler.

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Sebagai contoh, kita bisa menggunakannya untuk menghitung sudut segitiga bersisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak mudah didapat jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat.

```
>degrint(arcsin(sqrt(6/7)))
```

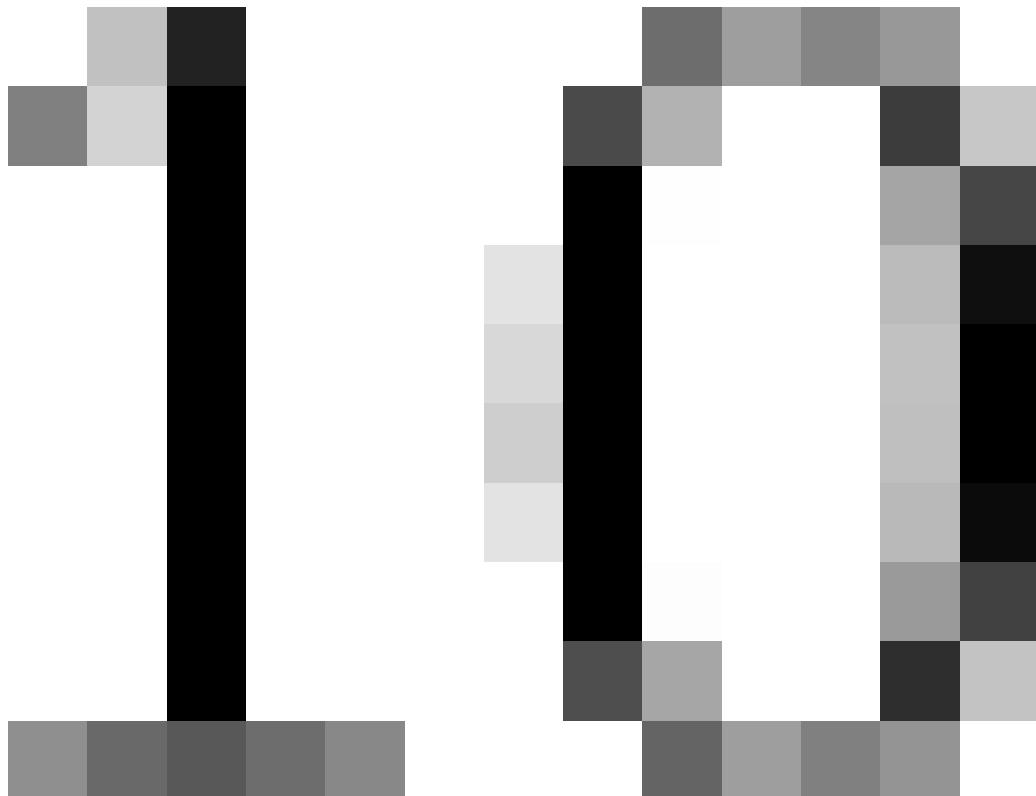
$67^\circ 47' 32.44''$

## Contoh lain

---

Sekarang, mari kita coba contoh yang lebih canggih.  
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 memiliki fungsi kuadrans antara dua titik.

Dalam contoh berikut, karena  $c + b$  bukan  $a$ , segitiga tidak 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 perkalian titik dari dua vektor. Hasilnya adalah beberapa pendekatan floating point.

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

$$\arccos\left(\frac{11}{\sqrt{10}\sqrt{17}}\right)$$

32.4711922908

Menggunakan pensil dan kertas, kita bisa melakukan hal yang sama dengan hukum silang. Kami memasukkan kuadran  $a$ ,  $b$ , dan  $c$  ke dalam hukum silang dan menyelesaikan untuk  $x$ .

```
>$crosslaw(a,b,c,x), $solve(%,x),
```

$$4 = 200(1 - x)$$

$$\left[x = \frac{49}{50}\right]$$

Artinya, fungsi penyebaran yang didefinisikan dalam "geometry.e".

```
>sb &= spread(b,a,c); $sb
```

$$\frac{49}{170}$$

Maxima mendapatkan hasil yang sama dengan menggunakan trigonometri biasa, jika kita memaksakannya. Itu menyelesaikan istilah  $\sin(\arccos(...))$  menjadi hasil pecahan. Kebanyakan siswa tidak dapat melakukan ini.

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{49}{170}$$

Setelah kita mendapatkan sebaran di  $B$ , kita bisa 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 diproduksi dengan program geometri C.a.R., yang dapat menggambar kuadran dan menyebar.

gambar : (20) Rational\_Geometry\_CaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadrannya.

```
>$sqrt(ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang kita bisa menghitung luas segitiga. Jangan lupa, bahwa kita berurusan dengan kuadran!

```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{7}{2}$$

Rumus determinan yang biasa menghasilkan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

$$\frac{7}{2}$$

## Formula Heron

---

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue(a,b,c,sb,ha);
```

Pertama-tama kita menghitung spread di B untuk segitiga dengan sisi a, b, dan c. Kemudian kami menghitung luas area yang dikuadratkan ("kuadrea"?), Memfaktorkannya dengan Maxima, dan kami 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 bahwa mereka tidak lagi hanya menambahkan sudut serupa. Namun, tiga sebaran segitiga memenuhi aturan "penyebaran rangkap tiga" berikut.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

$$(sc + sb + sa)^2 = 2 (sc^2 + sb^2 + sa^2) + 4 sa sb sc$$

Aturan ini berlaku untuk tiga sudut yang bertambah menjadi  $180^\circ$ .

$$\alpha + \beta + \gamma = \pi$$

Sejak penyebaran

$$\alpha, \pi - \alpha$$

sama, aturan penyebaran tiga kali lipat juga benar, jika

$$\alpha + \beta = \gamma$$

Karena penyebaran sudut negatif adalah sama, aturan penyebaran tiga kali lipat juga berlaku, jika

$$\alpha + \beta + \gamma = 0$$

Misalnya, kita dapat menghitung sebaran sudut  $60^\circ$ . Ini 3/4. Persamaan memiliki solusi kedua, di mana semua spread adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

$$\left[ x = \frac{3}{4}, x = 0 \right]$$

Sebaran  $90^\circ$  jelaslah 1. Jika dua sudut dijumlahkan menjadi  $90^\circ$ , penyebarannya menyelesaikan persamaan penyebaran rangkap tiga dengan a, b, 1. Dengan perhitungan berikut kita mendapatkan  $a+b=1$ .

```
>$triplespread(x,y,1), $solve(%,x)
```

$$(y + x + 1)^2 = 2 (y^2 + x^2 + 1) + 4 xy$$

$$[x = 1 - y]$$

Karena penyebaran  $180^\circ - t$  sama dengan penyebaran  $t$ , rumus penyebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau perbedaan dari dua sudut lainnya.

Jadi kita bisa menemukan sebaran sudut berlipat ganda. Perhatikan bahwa ada dua solusi lagi. Kami menjadikannya sebuah fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))
```

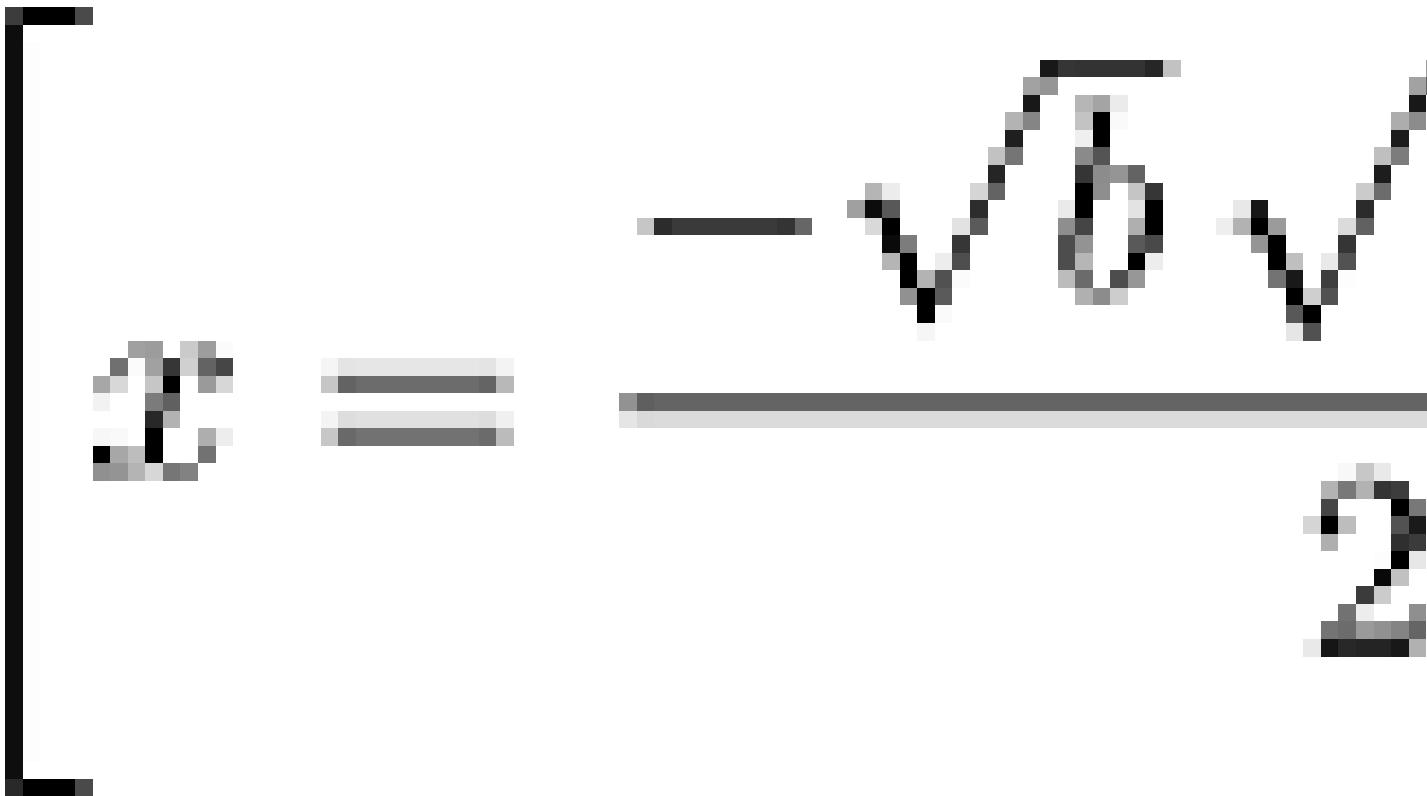
$$[x = 4a - 4a^2, x = 0]$$

$$- 4 (a - 1) a$$

## Pembagi Sudut

Ini 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 bisektor sudut pada A. Tapi kita ingin menyelesaiakannya untuk umum a, b, c.

```
>&remvalue(a,b,c);
```

Jadi pertama-tama kita menghitung sebaran sudut terbagi di A, menggunakan rumus sebaran rangkap tiga. Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kami harus memilih yang benar. Solusi lainnya mengacu pada sudut terbagi  $180^\circ$ -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 di Euler, setelah mentransfer penyebaran ke radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

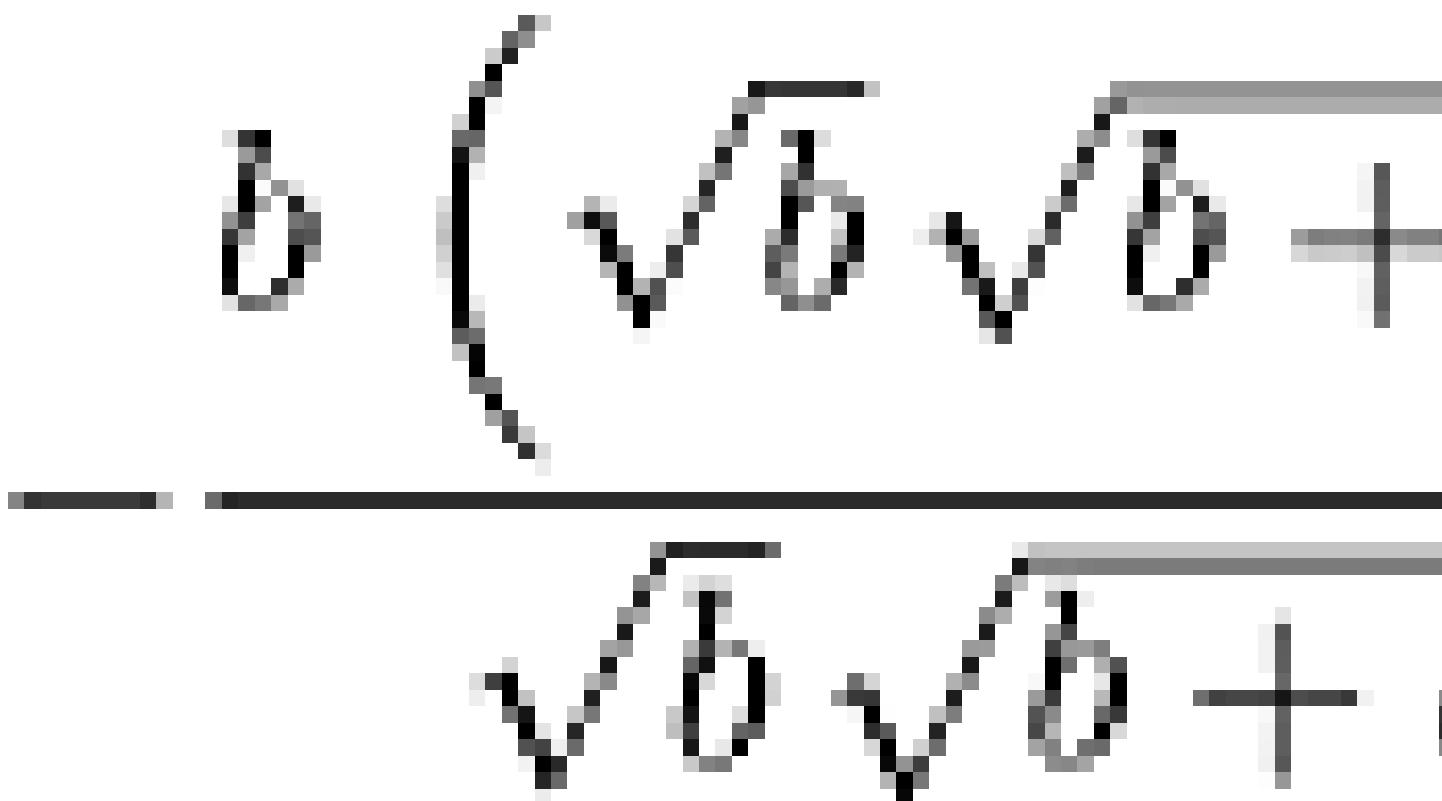
$$18^\circ 26' 5.82''$$

Titik P adalah perpotongan dari 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 menghitung panjang bisektor AP.

Kita menggunakan teorema sinus di segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

memegang di segitiga apa pun. Persegi itu, itu diterjemahkan ke dalam apa yang disebut "hukum penyebaran"

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_c}$$

dimana a, b, c menunjukkan qudrance.

Karena BPA sebaran adalah  $1-sa^2$ , kita dapatkan darinya bisa / 1 = b / (1-sa^2) dan dapat menghitung bisa (kuadran garis-garis).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$\frac{2b(b+a)}{\sqrt{b}\sqrt{b+a} + b+a}$$

Mari kita periksa rumus ini untuk nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])")), distance(A,P)
```

4.21637021356

4.21637021356

Kami juga dapat menghitung P menggunakan rumus spread.

```
>py&=factor(ratsimp(sa2*bisa)); $py
```

$$-\frac{b(\sqrt{b}\sqrt{b+a} - b - a)}{\sqrt{b}\sqrt{b+a} + b + a}$$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

1.33333333333

## Sudut Akord

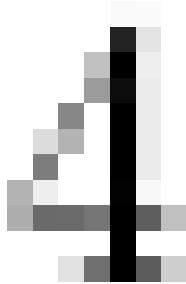
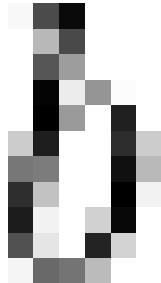
---

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 bisa menggunakan Maxima untuk menyelesaikan rumus sebaran rangkap tiga untuk sudut di pusat O untuk  $r$ . Jadi kita mendapatkan rumus untuk jari-jari kuadrat dari keliling dalam hal kuadrat sisi. Kali ini, Maxima menghasilkan beberapa angka nol yang 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 bisa menjadikannya sebagai fungsi Euler.

```

>function periradius(a,b,c) &= rabc;

```

Mari kita periksa hasilnya untuk poin A, B, C kita.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Radiusnya memang 1.

```
>periradius(a,b,c)
```

1

Faktanya, penyebaran CBA hanya bergantung pada b dan c. Ini adalah teorema sudut akord.

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Sebenarnya sebarannya adalah  $b/(4r)$ , dan kita melihat bahwa sudut akor b adalah setengah dari sudut tengah.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

0

## Contoh 6: Jarak Minimal pada Bidang

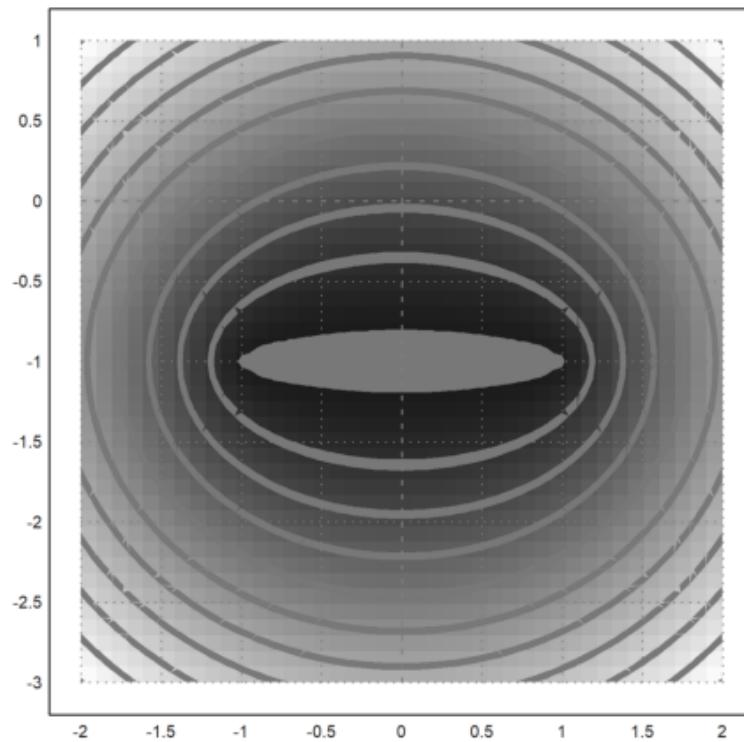
---

### Catatan awal

---

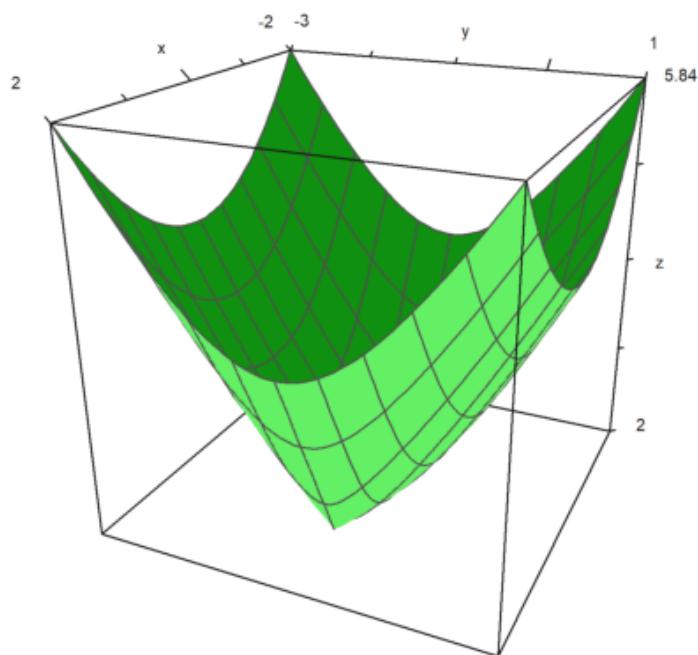
Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis level yang agak sederhana: lingkaran berpusat di A.

```
>&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) :
```



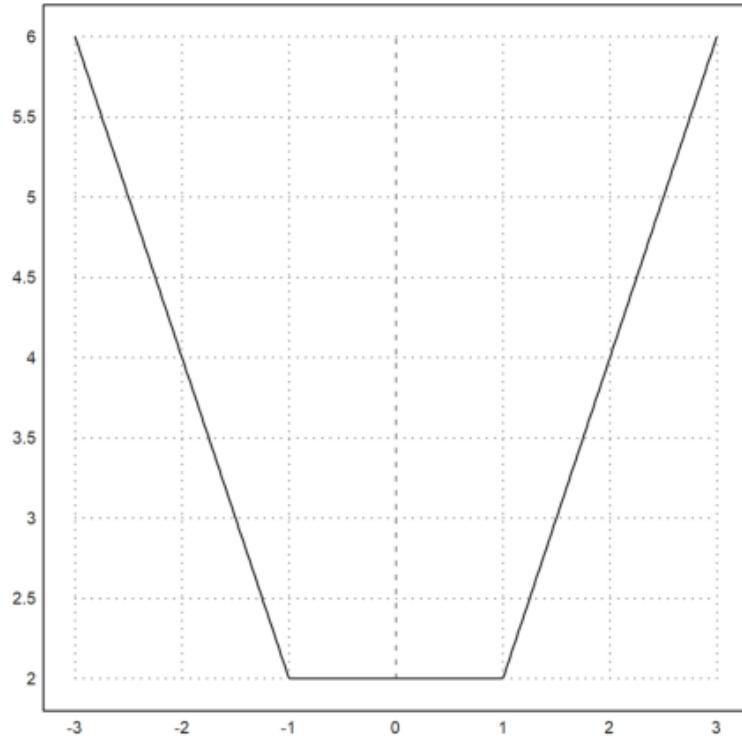
Tentu saja minimal 0 dicapai di A.

## Dua titik

---

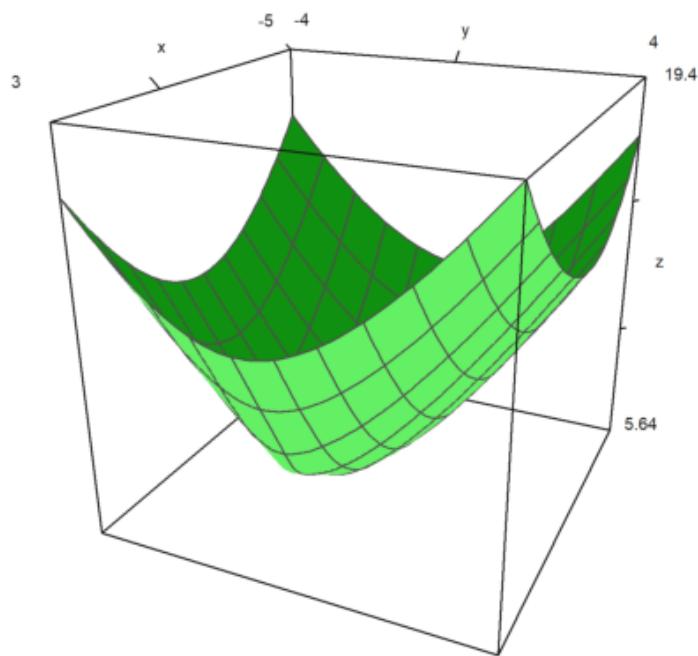
Sekarang kita melihat fungsi  $MA + MB$  dimana A dan B adalah dua titik (tetap). Ini adalah "fakta yang terkenal" bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali untuk minimum AB yang konstan pada segmen [AB]:

```
>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):
```



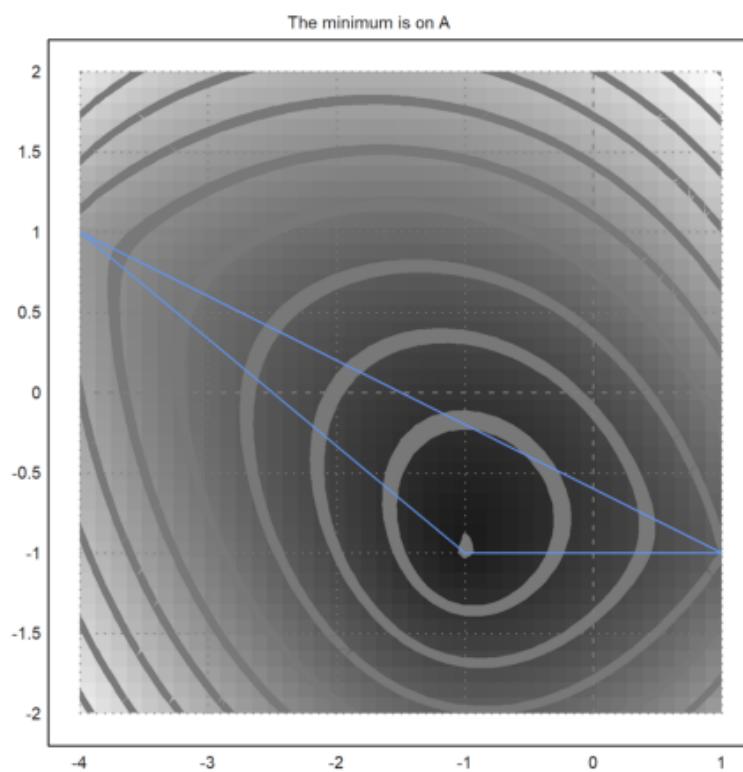
Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Batasan ke baris (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



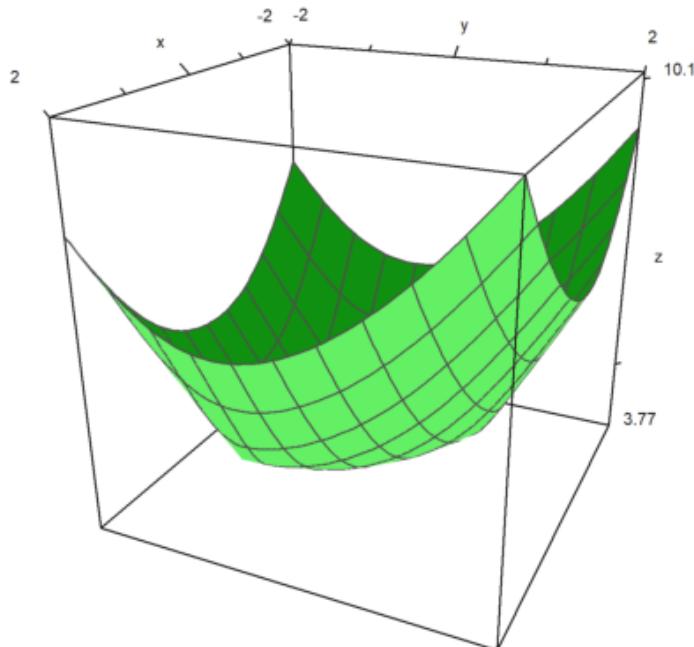
## Tiga titik

Sekarang hal-hal menjadi kurang sederhana: Sedikit kurang diketahui bahwa  $MA+MB+MC$  mencapai minimumnya pada satu titik bidang tetapi untuk menentukannya kurang sederhana:

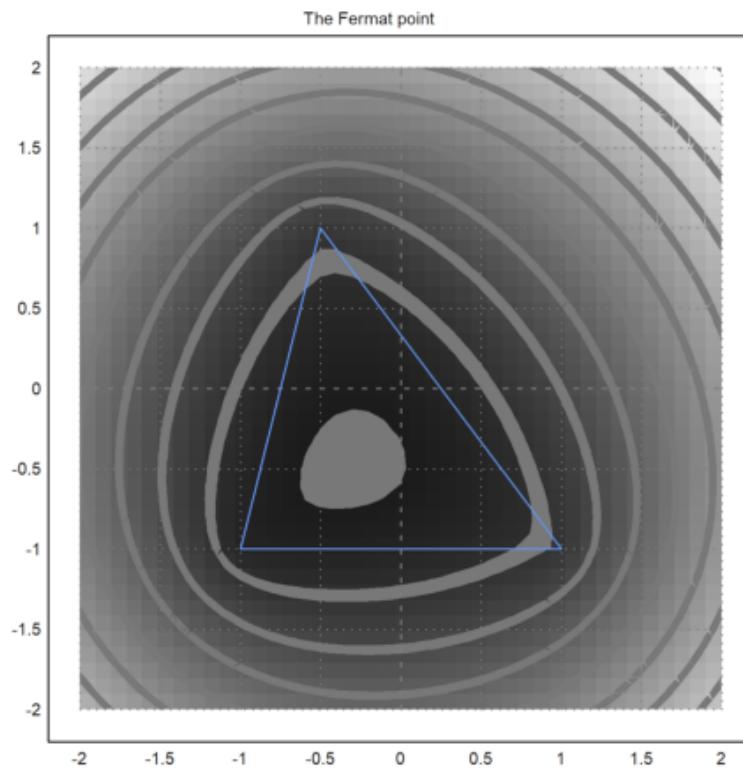
- 1) Jika salah satu sudut segitiga ABC lebih dari  $120^\circ$  (katakanlah dalam A), maka minimum tercapai pada titik ini (katakanlah AB+AC).

Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
>insimg;
```

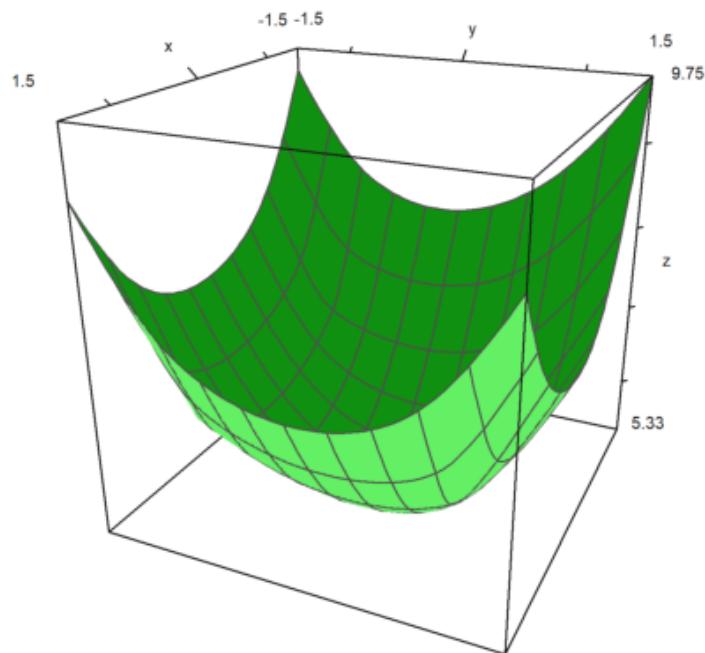


```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)';
>plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

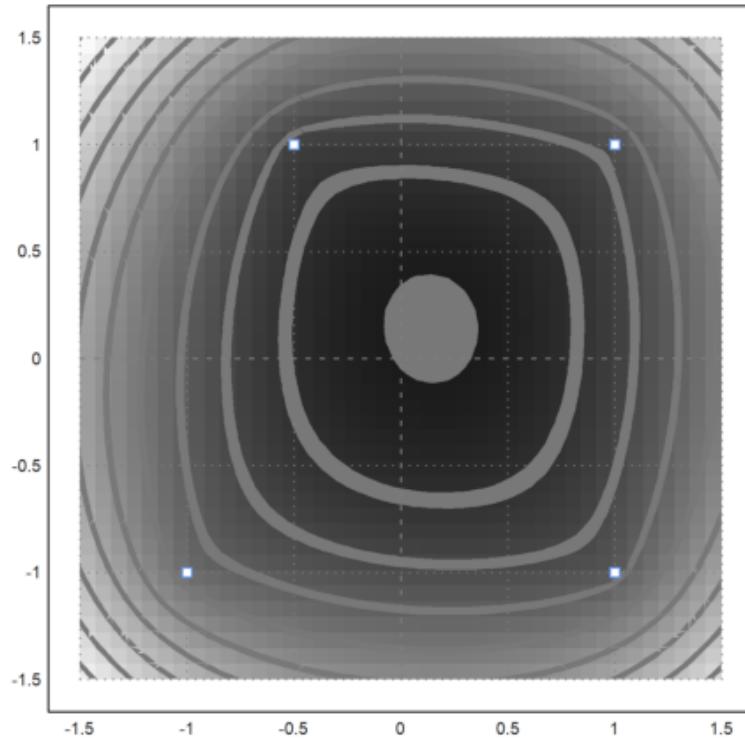


- 2) Tetapi jika semua sudut segitiga ABC kurang dari  $120^\circ$ , minimum berada pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi ABC dengan sudut yang sama (lalu masing-masing  $120^\circ$ ) :

```
>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; sebagai contoh, saya tahu soft tertulis di Java yang memiliki instruksi "garis kontur" ...

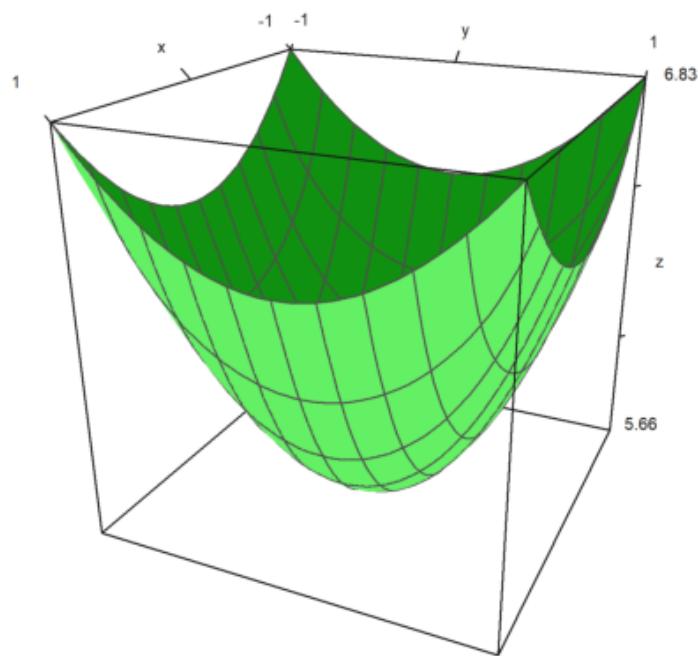
Semua ini di atas telah ditemukan oleh seorang hakim Prancis bernama Pierre de Fermat; dia menulis surat kepada para penggila lainnya seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di bagian pajak penghasilan. Jadi titik unik F sehingga  $FA + FB + FC$  minimal disebut titik Fermat segitiga. Tetapi tampaknya beberapa tahun sebelumnya, Torricelli Italia telah menemukan titik ini sebelum Fermat melakukannya! Pokoknya tradisinya adalah mencatat poin ini ...

## Empat titik

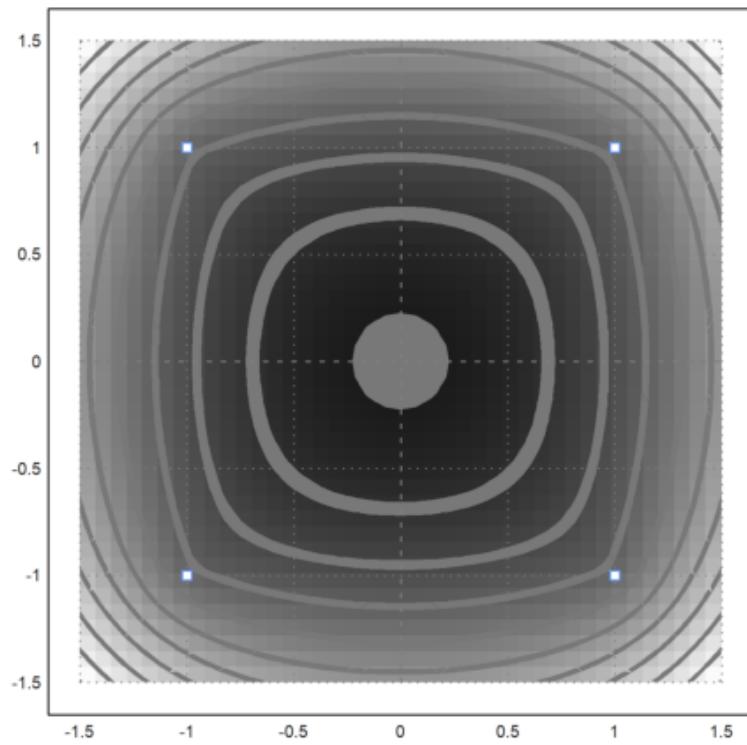
---

Langkah selanjutnya adalah menambahkan titik D ke-4 dan mencoba meminimalkan  $MA + MB + MC + MD$ ; katakanlah 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 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 ada yang dicapai pada simpul A, B, C atau D:

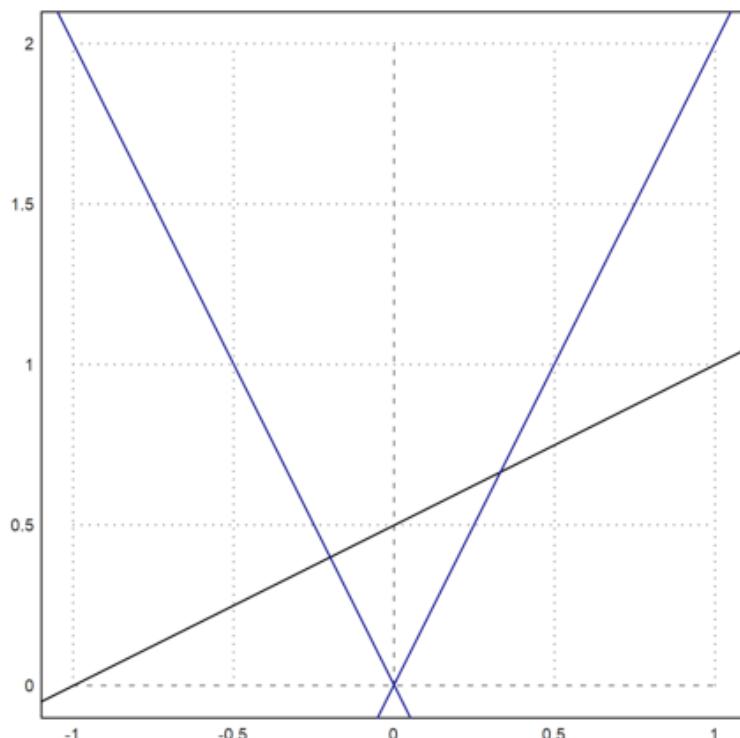
```
>function f(x):=d4(x[1],x[2])  
>neldermin("f", [0.2,0.2])
```

[0.142858, 0.142857]

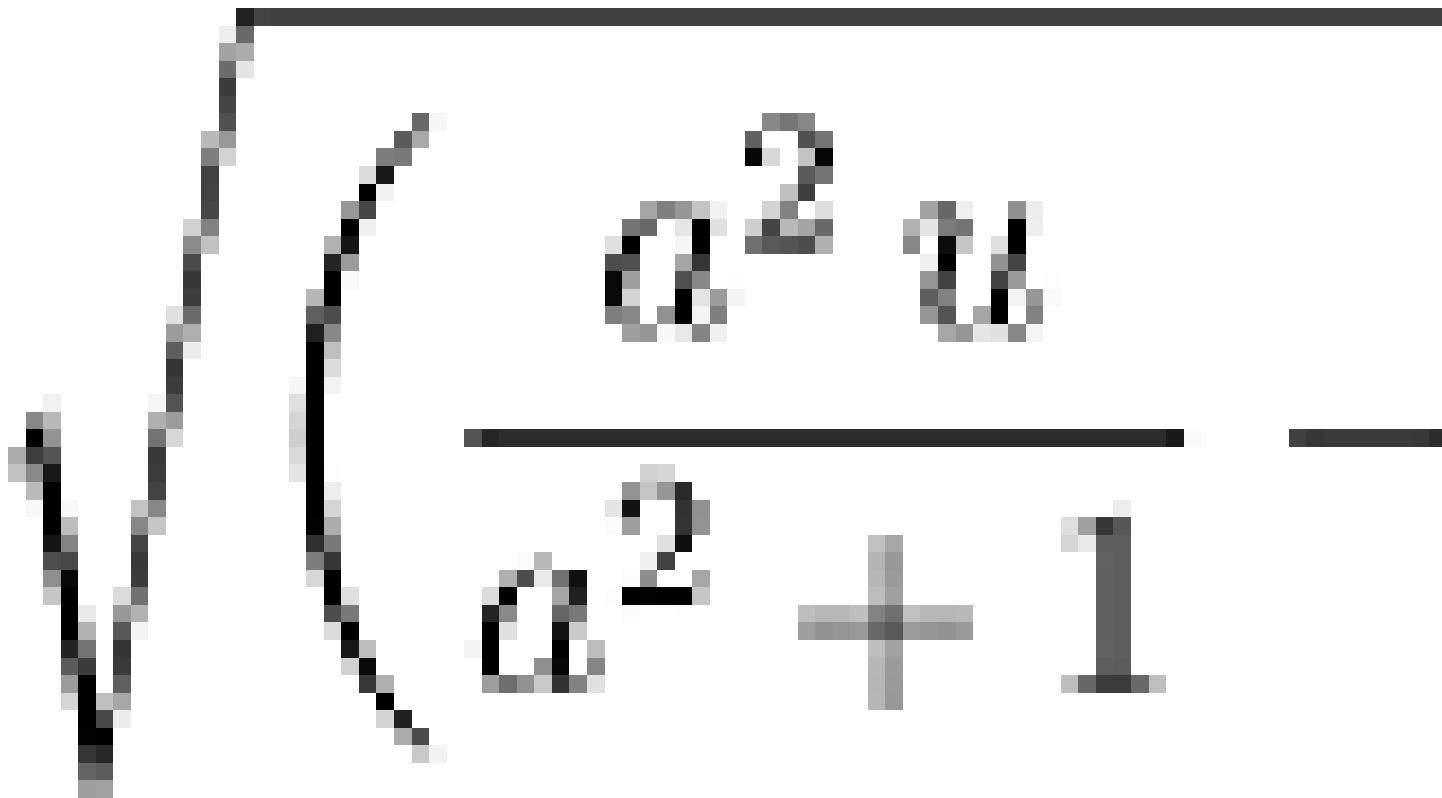
Tampaknya dalam kasus ini, koordinat titik optimal rasional atau mendekati rasional ...

Sekarang ABCD adalah bujur sangkar, kami berharap bahwa titik optimal adalah 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 memiliki Povray diinstal, dan pengine.exe di jalur program.

Pertama kami menghitung 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 geometry.e dari Euler untuk ini.

```
>load geometry;
```

Pertama, dua garis yang membentuk kerucut.

```
>g1 &= lineThrough([0,0],[1,a])
```

```
[ - a, 1, 0 ]
```

```
>g2 &= lineThrough([0,0],[-1,a])
```

```
[ - a, - 1, 0 ]
```

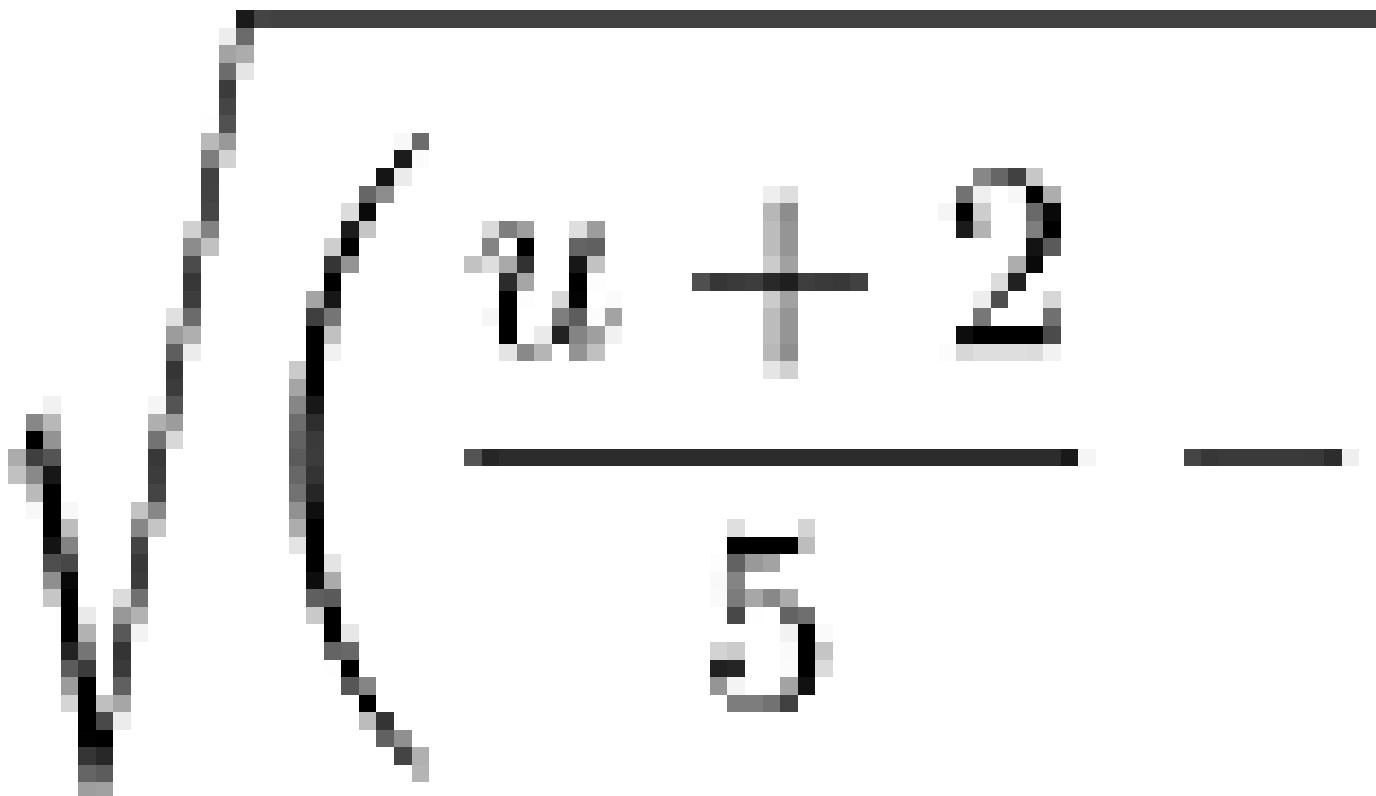
Lalu baris ketiga.

```
>g &= lineThrough([-1,0],[1,1])
```

```
[- 1, 2, 1]
```

Kita 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, di mana 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()
```

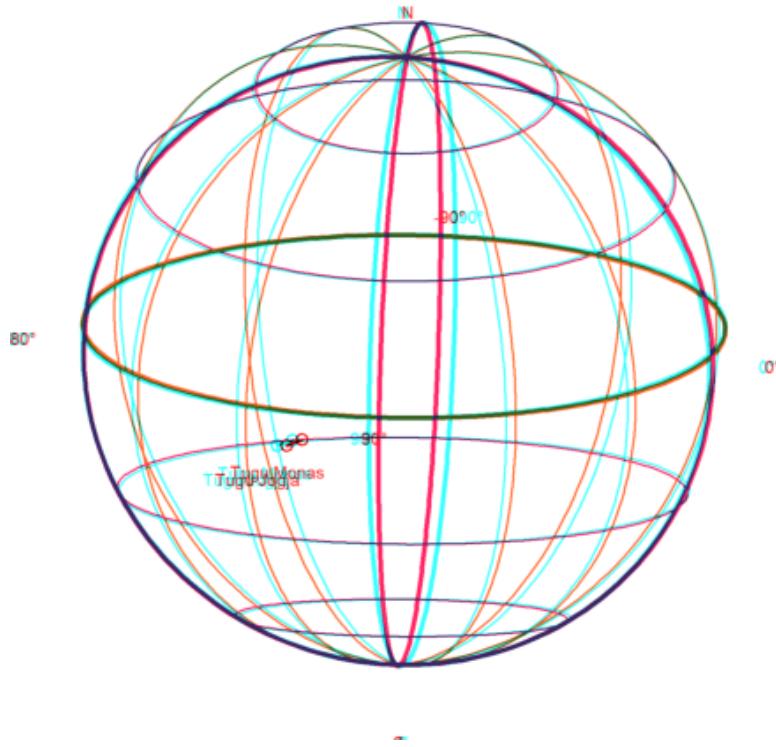
```
[0.333333, 1]
```

```
>dd := d()
```

```
[0.149071, 0.447214]
```

Plot lingkaran ke dalam gambar.

```
>color(green);
>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

Kita mengatur adegan dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Selanjutnya kita menulis dua bidang ke file Povray.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

Dan kerucutnya, transparan.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

Kami menghasilkan pesawat 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 kami 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 elips.

```
>P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];  
>writeln(povpoint(P3,povlook(yellow)));  
>P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];  
>writeln(povpoint(P4,povlook(yellow)));
```

Selanjutnya kita menghitung perpotongan P1P2 dengan bidang.

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);  
>writeln(povpoint(P5,povlook(yellow)));
```

Kami menghubungkan titik dengan segmen garis.

```
>writeln(povsegment(P1,P2,povlook(yellow)));  
>writeln(povsegment(P5,P3,povlook(yellow)));  
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Sekarang kami membuat pita abu-abu, di mana bola menyentuh kerucut.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);  
>pc1=povcylinder([0,0,P1[3]-defaultpointsiz/2],[0,0,P1[3]+defaultpointsiz/2],1);  
>writeln(povintersection([pcw,pc1],povlook(gray)));  
>pc2=povcylinder([0,0,P2[3]-defaultpointsiz/2],[0,0,P2[3]+defaultpointsiz/2],1);  
>writeln(povintersection([pcw,pc2],povlook(gray)));
```

Mulai program Povray.

```
>povend();
```

Untuk mendapatkan Anaglyph ini, kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali nanti.

```
>function scene () ...  
  
    global a,u,dd,g,g1,defaultpointsize;  
    writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));  
    writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));  
    writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));  
    gp=g();  
    pc=povcone([0,0,0],0,[0,0,a],1,"");  
    vp=[gp[1],0,gp[2]]; dp=gp[3];  
    writeln(povplane(vp,dp,povlook(blue,0.5),pc));  
    P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);  
    writeln(povpoint(P1,povlook(yellow)));  
    P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);  
    writeln(povpoint(P2,povlook(yellow)));  
    P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];  
    writeln(povpoint(P3,povlook(yellow)));  
    P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];  
    writeln(povpoint(P4,povlook(yellow)));  
    t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);  
    writeln(povpoint(P5,povlook(yellow)));  
    writeln(povsegment(P1,P2,povlook(yellow)));  
    writeln(povsegment(P5,P3,povlook(yellow)));  
    writeln(povsegment(P5,P4,povlook(yellow)));  
    pcw=povcone([0,0,0],0,[0,0,a],1.01);  
    pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);  
    writeln(povintersection([pcw,pc1],povlook(gray)));  
    pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);  
    writeln(povintersection([pcw,pc2],povlook(gray)));  
endfunction
```

Anda membutuhkan kacamata merah / cyan untuk mengapresiasi efek berikut.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

## Contoh 8: Geometri Bumi

Di notebook ini, kami ingin melakukan beberapa komputasi bola. Fungsi-fungsi tersebut terdapat dalam file "spherical.e" di folder contoh. Kita perlu memuat file itu dulu.

```
>load spherical.e
```

Spherical functions for Euler.

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 bola).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

S  $7^{\circ}46.467'$  E  $110^{\circ}23.050'$

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  $7^{\circ}34.333'$  E  $110^{\circ}49.683'$

S  $6^{\circ}59.050'$  E  $110^{\circ}24.533'$

Pertama kita menghitung vektor dari satu bola ke bola lainnya pada bola ideal. Vektor ini adalah [heading, distance] dalam radian. Untuk menghitung jarak di bumi, kita mengalikan dengan jari-jari bumi pada garis lintang  $7^{\circ}$ .

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth( $7^{\circ}$ )->km // perkiraan jarak FMIPA-Solo
```

$65^{\circ}20'26.60''$

53.8945384608

Ini adalah perkiraan yang bagus. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak yang begitu dekat hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->" km", // perkiraan jarak FMIPA-Semarang
```

88.0114026318 km

Ada fungsi untuk heading, dengan mempertimbangkan bentuk bumi yang elips. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

$65.34^{\circ}$

Sudut segitiga melebihi  $180^{\circ}$  pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,Semarang,Solo);
```

$180^{\circ}0'10.77''$

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, ini tidak akurat karena kesalahan pengurangan dalam asum- $\pi$ .

```
>(asum-pi)*rearth( $48^{\circ}$ )^2->" km^2", //perkiraan luas segitiga FMIPA-Solo-Semarang
```

2116.02948749 km $^2$

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 km<sup>2</sup>

Kami juga dapat menambahkan vektor ke posisi. Vektor berisi heading dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kami menggunakan svector. Untuk menambahkan vektor ke posisi, kami menggunakan saddvector.

```
>v=svector(FMIPA,Solo); sposprint(saddvector(FMIPA,v)), sposprint(Solo),
```

S 7°34.333' E 110°49.683'  
S 7°34.333' E 110°49.683'

Fungsi-fungsi ini mengasumsikan bola yang ideal. Hal yang sama di bumi.

```
>sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(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 429,66 km. Kami mendapatkan perkiraan yang bagus.

```
>esdist(Tugu,Monas) ->" km", // perkiraan jarak Tugu Jogja – Monas Jakarta
```

431.565659488 km

Judulnya sama dengan yang dihitung di Google Earth.

```
>degrprint(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  $6^{\circ}10.500'$  E  $106^{\circ}48.717'$

Tentunya kita tidak bisa berlayar dengan tujuan yang sama dari satu tujuan ke tujuan lainnya, jika kita ingin mengambil jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti arah yang konstan!

Perhitungan berikut menunjukkan bahwa kami jauh dari tujuan yang benar, jika kami menggunakan tajuk yang sama selama perjalanan kami.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh jaraknya, menggunakan heading ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya masih jauh.

```
>sposprint(p), skmpprint(esdist(p,Monas))
```

S  $6^{\circ}11.250'$  E  $106^{\circ}48.372'$   
1.529km

Sebagai contoh lain, mari kita ambil dua titik di bumi pada ketinggian yang sama.

```
> P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran dengan garis lintang  $30^{\circ}$ , tetapi jalur yang lebih pendek mulai  $10^{\circ}$  lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

$79.69^{\circ}$

Tapi, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah tujuan kita di sepanjang jalan. Untuk tujuan kasar, kami menyesuaikannya pada 1/10 dari jarak total.

```
>p=P1; dist=esdist(P1,P2); ...  
> loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

$79.69^{\circ}$   
 $81.67^{\circ}$   
 $83.71^{\circ}$   
 $85.78^{\circ}$

```
87.89°  
90.00°  
92.12°  
94.22°  
96.29°  
98.33°
```

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti tajuk yang sama terlalu lama.

```
>skmpprint(esdist(p,P2))
```

```
0.203km
```

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan heading 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;  
>skmpprint(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'
```

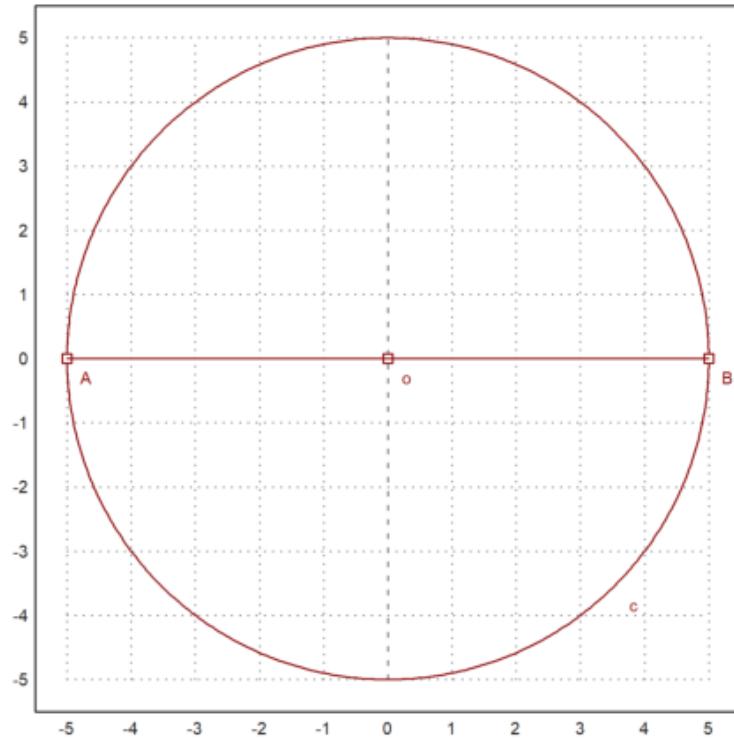
Kami menulis sebuah fungsi, yang menggambarkan bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
```

```
useglobal;  
plotearth;  
plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");  
plotposline(v);  
endfunction
```

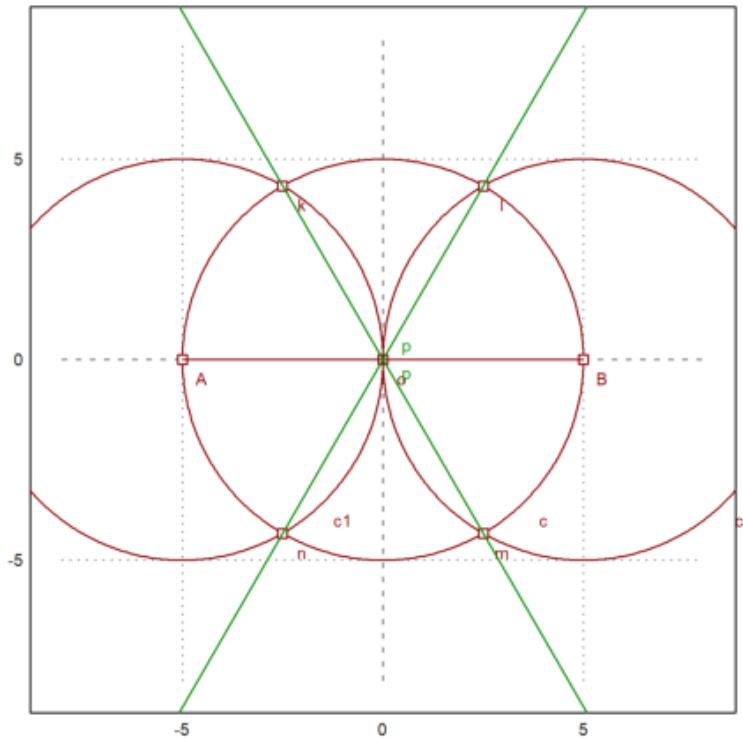
Sekarang plot semuanya.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```



Atau gunakan plot3d untuk mendapatkan tampilan anaglyphnya. Ini terlihat sangat bagus dengan kacamata merah / cyan.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```



## Latihan

---

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah  $(360/n)$ .
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan  $(360/n)$ .
- Untuk n ganjil, pilih salah satu titik sudut adalah di atas.
- Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat.
- Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:

- Misalkan persamaan parabolanya  $y = ax^2 + bx + c$ .
- Subsitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.

– Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung

(sisinya-sisinya merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).

– Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya bertemu di satu titik.

– Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar

lingkaran dalamnya.

- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

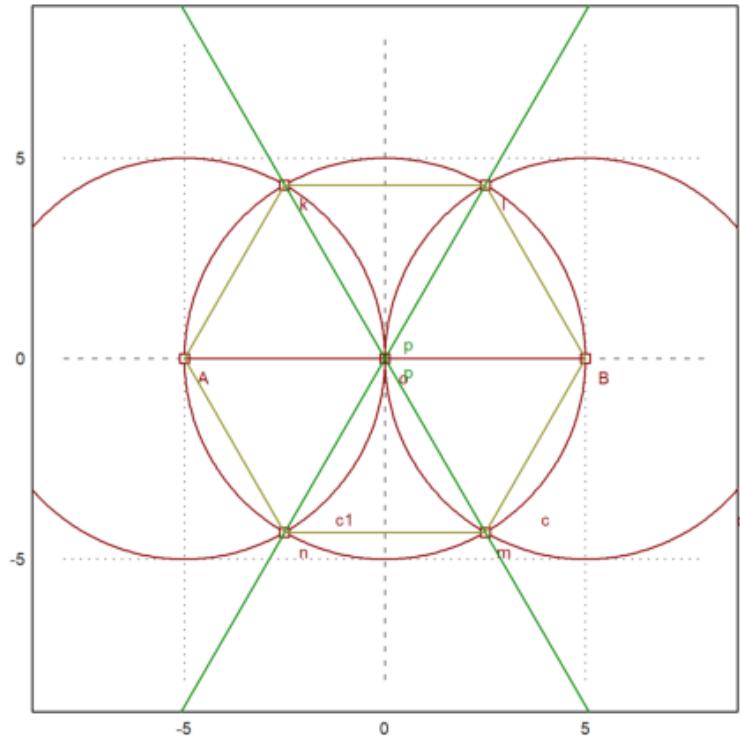
4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).

5. Gambarlah suatu hiperbolika jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

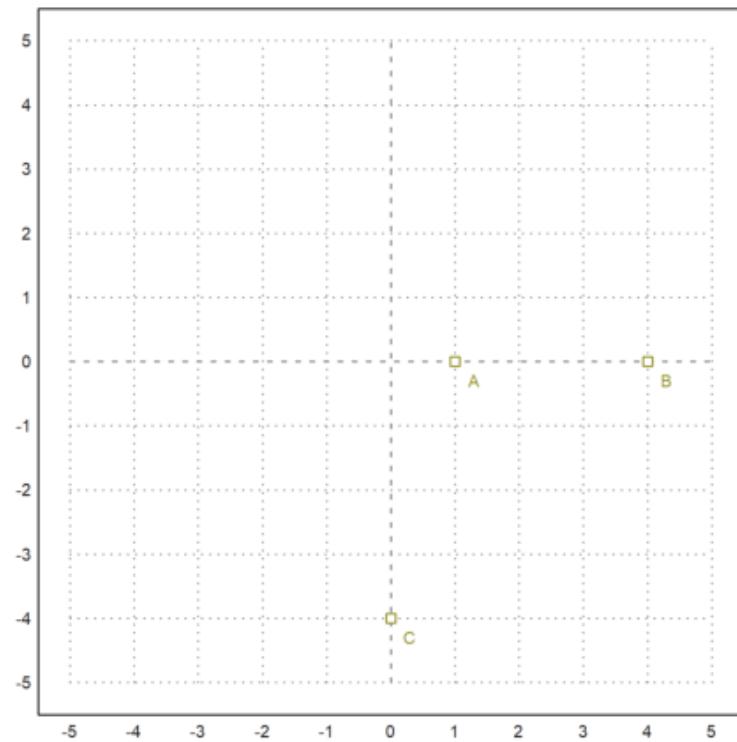
Jawab:

1.

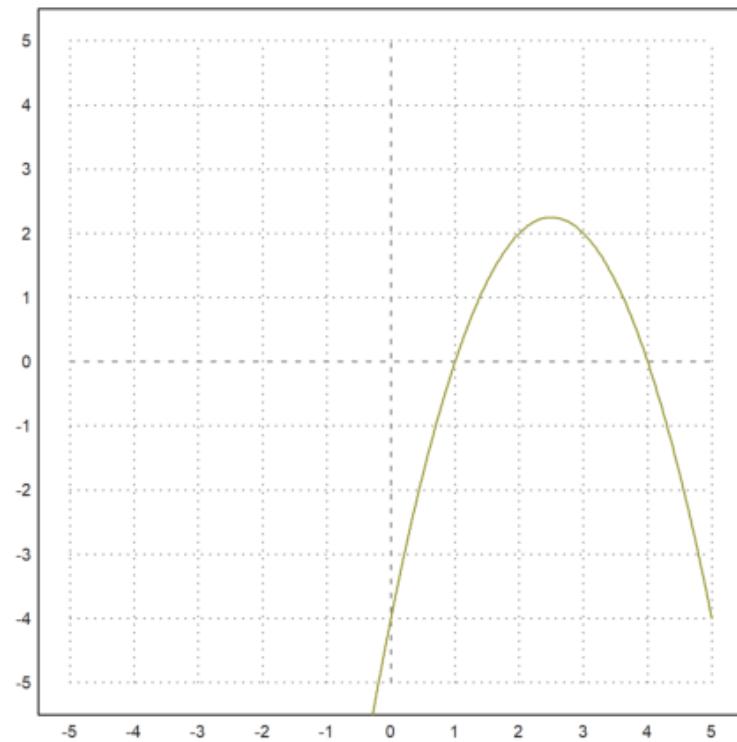
```
>o &:= [0,0]; c=circleWithCenter(o,5);
>color(2); setPlotRange(5); plotPoint(o); plotCircle(c);
>A=[-5,0]; plotPoint(A,"A");
>B=[5,0]; plotPoint(B,"B");
>plotSegment(A,B,""):
```



```
>c1=circleWithCenter(A,distance(A,o));
>c2=circleWithCenter(B,distance(B,o));
>k=circleCircleIntersections(c1,c);
>l=circleCircleIntersections(c,c2);
>m=circleCircleIntersections(c2,c);
>n=circleCircleIntersections(c,c1);
>r=lineThrough(k,m); s=lineThrough(l,n);
>setPlotRange(8); plotPoint(o); plotCircle(c); plotPoint(A,"A"); plotPoint(B,"B"); plotSeg
>color(3); plotCircle(c1); plotCircle(c2); plotPoint(k); plotPoint(l); plotPoint(m); plotP
>color(4); plotLine(r); plotLine(s):
```

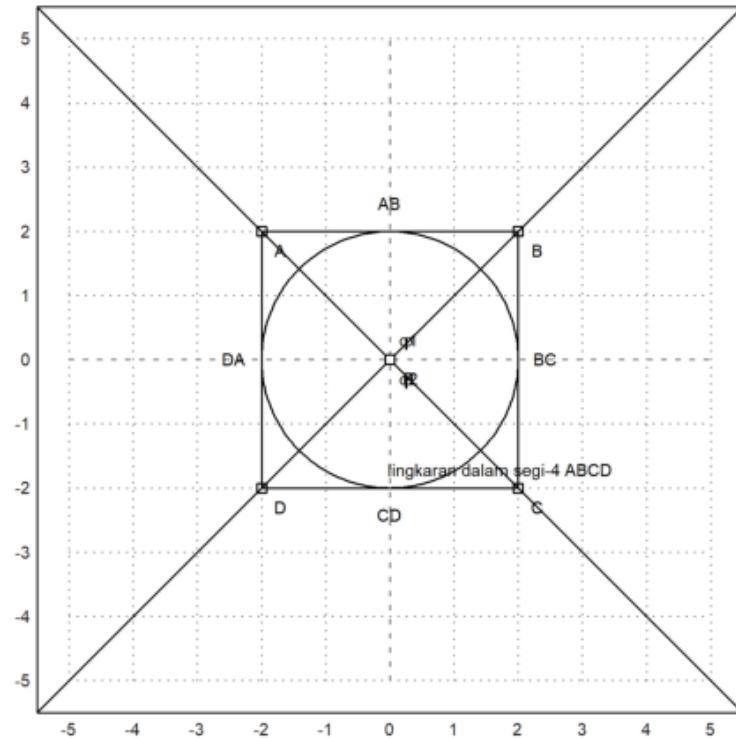


```
>color(5); plotSegment(A,k,""); plotSegment(A,n,""); plotSegment(k,l,""); ...
> plotSegment(l,B,""); plotSegment(B,m,""); plotSegment(m,n,"");
```



2.

```
>setPlotRange(5); A=[1,0]; B=[4,0]; C=[0,-4];
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```



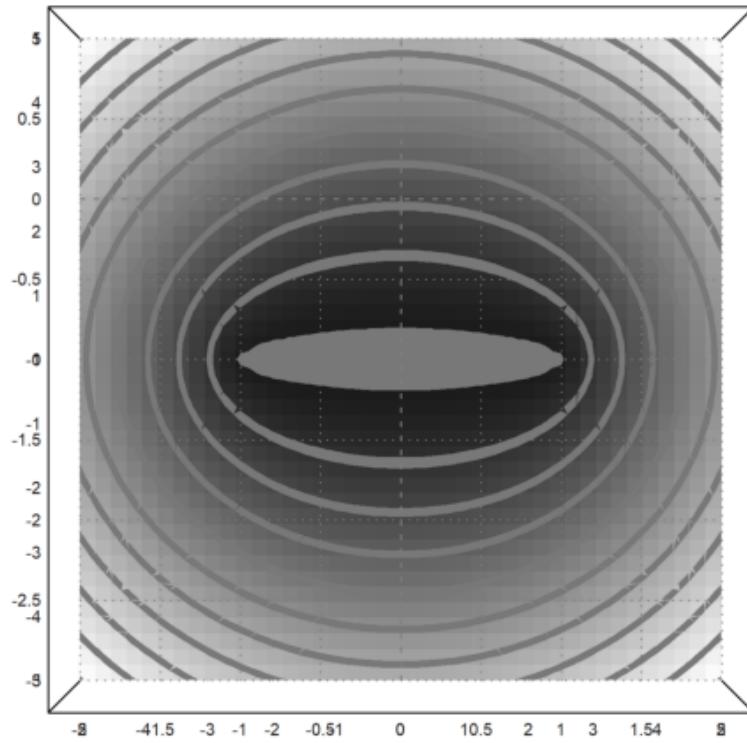
```
>sol &= solve([a+b=-c, 16*a+4*b=-c, c=-4], [a,b,c])
```

```
[[a = - 1, b = 5, c = - 4]]
```

```
>function y&=-x^2+5*x-4
```

$$-\frac{2}{x} + 5x - 4$$

```
>plot2d("-x^2+5*x-4", -5, 5, -5, 5);
```

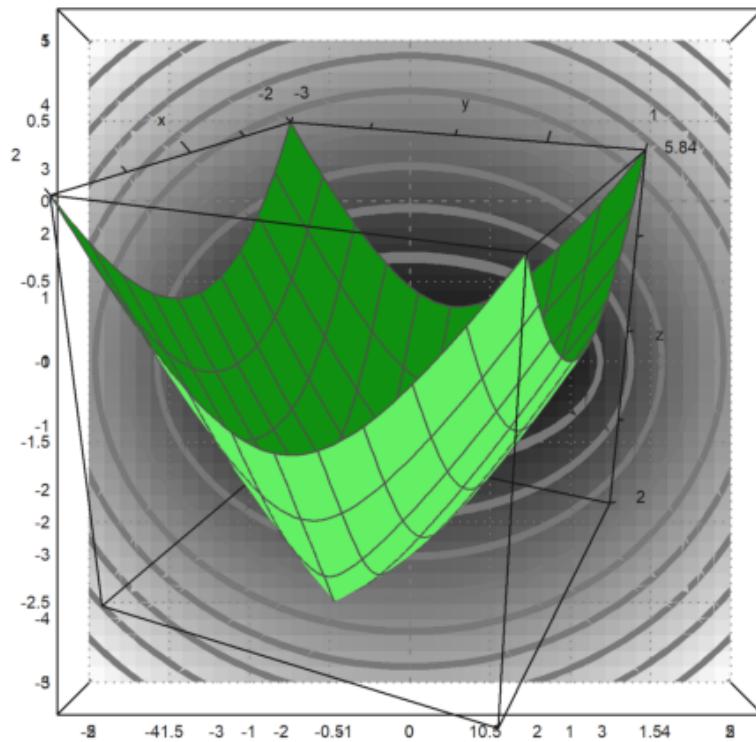


3.

```
>setPlotRange(-5,5,-5,5);
>A=[-2,2]; plotPoint(A,"A");
>B=[2,2]; plotPoint(B,"B");
>C=[2,-2]; plotPoint(C,"C");
>D=[-2,-2]; plotPoint(D,"D");
>plotSegment(A,B);
>plotSegment(B,C);
>plotSegment(C,D);
>plotSegment(D,A);
>plotSegment(A,C,"q1");
>plotSegment(B,D,"q2");
>q1=lineThrough(A,C);
>q2=lineThrough(B,D);
>p=lineIntersection(q1,q2);
>plotLine(q1); plotLine(q2);
>plotPoint(p, "P");
>r=norm(p-projectToLine(p,lineThrough(A,B)))
```

2

```
>plotCircle(circleWithCenter(p,r),"lingkaran dalam segi-4 ABCD");
```



```
>AB=norm(A-B) // panjang sisi AB
```

4

```
>CD=norm(C-D) // panjang sisi CD
```

4

```
>AD=norm(A-D) // panjang sisi AD
```

4

```
>BC=norm(B-C) // panjang sisi BC
```

4

```
>AB.CD
```

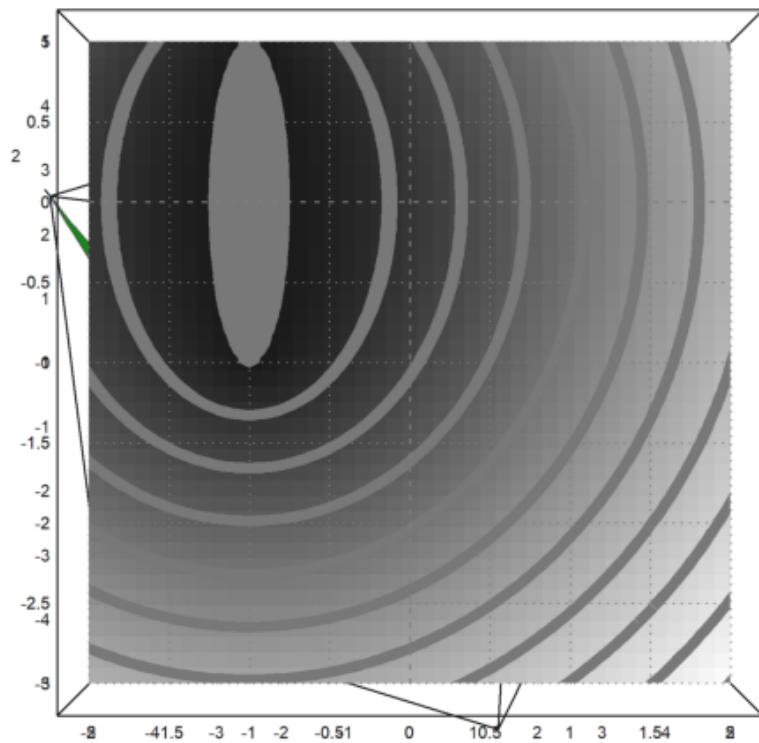
16

```
>AD.BC
```

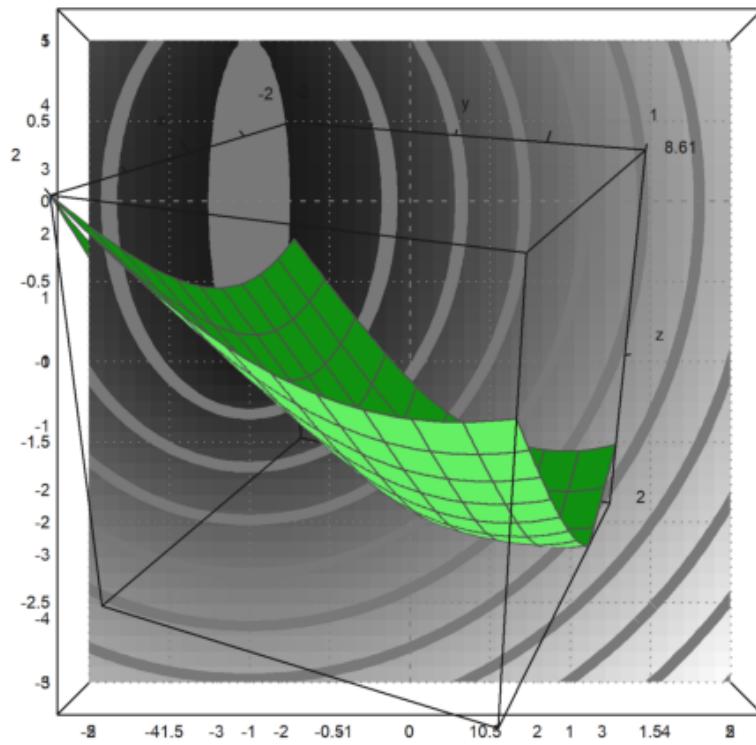
16

4.

```
>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):
```

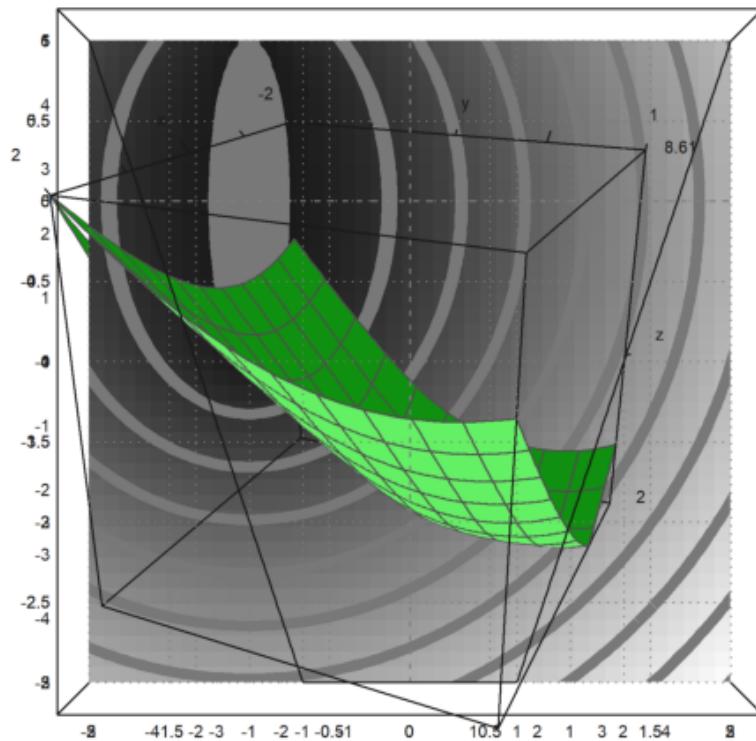


```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



5.

```
>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):
```



```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



---

---

## BAB 7

---

# PEKAN 10; MENGGUNAKAN EMT UNTUK STATISTIKA

[a4paper,10pt]article eumat

### EMT untuk Statistika

---

Raden Mas Farrel Maheswara Kusuma Dewa  
22305141042  
Matematika E 2022

---

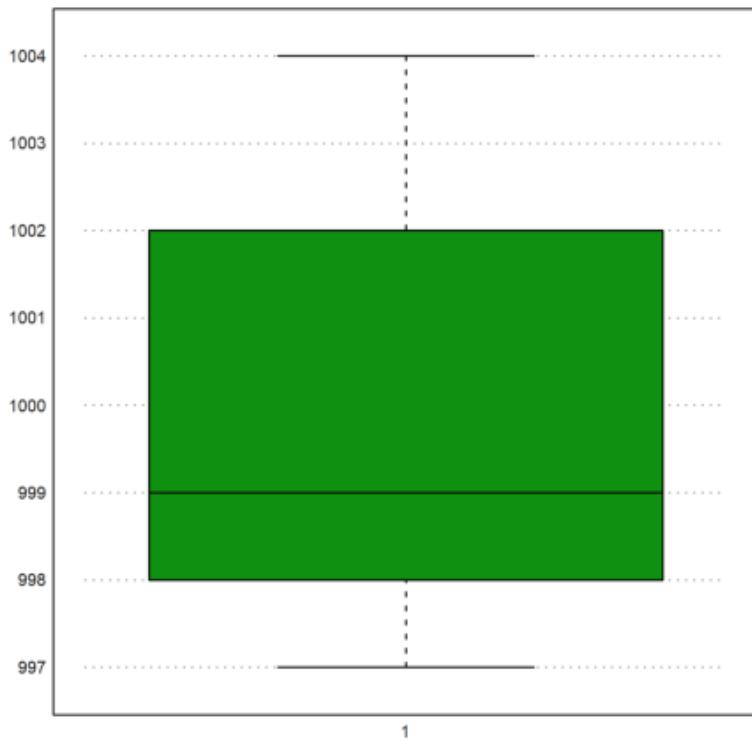
Di notebook ini, kami mendemonstrasikan plot statistik utama, tes dan distribusi di Euler. Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar statistik. Jadi, Anda mungkin memerlukan latar belakang untuk memahami detailnya.  
Asumsikan pengukuran berikut. Kami ingin menghitung nilai rata-rata dan standar deviasi yang diukur.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...  
>mean(M), dev(M),
```

999.9  
2.72641400622

Kita dapat memplot plot box-and-whiskers untuk data tersebut. Dalam kasus kami, tidak ada garis luar.

```
>boxplot(M):
```



Kita menghitung probabilitas bahwa suatu nilai lebih besar dari 1005, dengan asumsi nilai terukur dan distribusi normal.

Semua fungsi untuk distribusi di Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

$$\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} dt.$$

Kami mencetak hasilnya dalam % dengan akurasi 2 digit menggunakan fungsi cetak.

```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
```

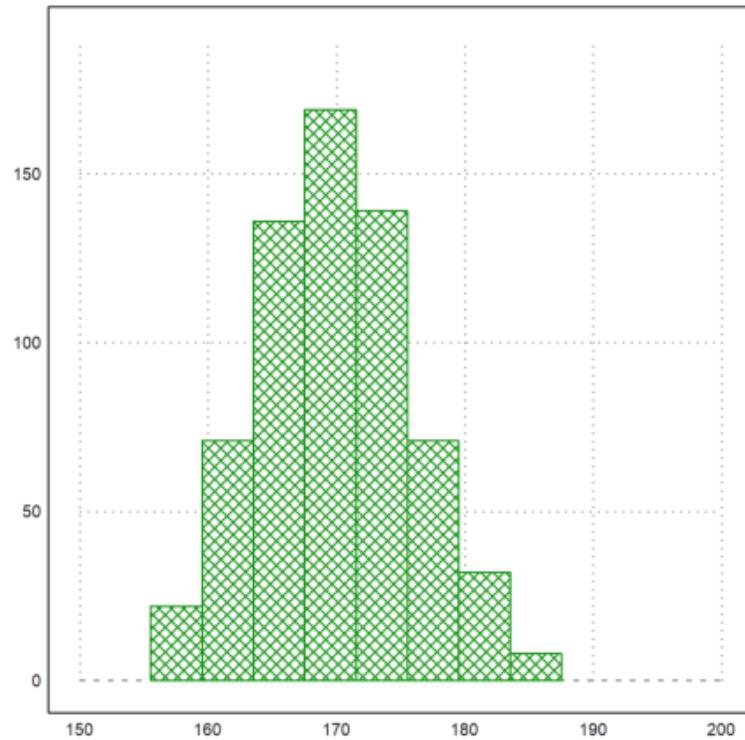
3.07 %

Untuk contoh berikut, kita 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 bisa memasukkan data mentah tersebut ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Mulai kisaran, akhir kisaran, jumlah laki-laki dalam kisaran.

Tabel dapat dicetak dengan header. Kami menggunakan vektor string untuk mengatur header.

```
>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 membutuhkan nilai rata-rata dan statistik ukuran lainnya, kita perlu menghitung titik tengah rentang. Kita dapat menggunakan dua kolom pertama dari tabel kita untuk ini.

Simbol "|" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan pilihan "labc" adalah menentukan header kolom.

```
>(T[,1]+T[,2])/2 // the midpoint of each interval
```

```
157.5
161.5
165.5
169.5
173.5
```

177.5  
181.5  
185.5

Tapi lebih mudah, melipat rentang dengan vektor [1/2, 1/2].

```
>M=fold(r, [0.5, 0.5])
```

[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]

Sekarang kita dapat menghitung mean dan deviasi sampel dengan frekuensi yang diberikan.

```
>{m, d}=meandev(M, v); m, d,
```

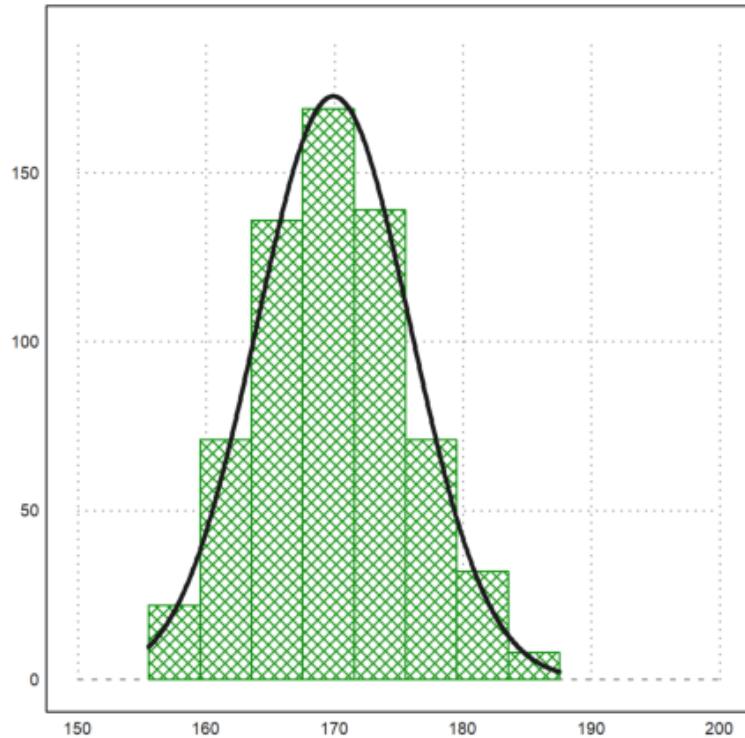
169.901234568  
5.98912964449

Mari kita tambahkan distribusi normal nilai ke plot batang di atas. Rumus distribusi normal dengan mean m dan standar deviasi d adalah:

$$y = \frac{1}{d\sqrt{2\pi}} e^{\frac{-(x-m)^2}{2d^2}}.$$

Karena nilainya antara 0 dan 1, untuk memplotnya pada diagram batang harus dikalikan dengan 4 kali jumlah data.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...  
> xmin=min(r), xmax=max(r), thickness=3, add=1):
```



## Tabel

Dalam direktori buku catatan ini Anda menemukan file dengan tabel. Data tersebut merupakan hasil survei. Berikut adalah empat baris pertama file. Data tersebut 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 token.

Untuk ini, kami mendefinisikan set 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 memberinya ":=".

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
>load over statistics;
```

Untuk mencetak, kita perlu menentukan set token yang sama. Kita mencetak empat baris pertama saja.

```
>writetable(MT[1:4],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

Titik "." mewakili nilai-nilai yang tidak tersedia.

Jika kita tidak ingin menentukan token untuk terjemahan 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);
```

Fungsi readtable() sekarang mengembalikan satu set token.

```
>tok
```

```
m
n
f
y
g
vg
```

Tabel berisi entri dari file dengan token yang diterjemahkan menjadi angka.

String khusus NA="." diartikan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) di tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAvl.

```
>MT [1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Berikut adalah isi tabel dengan bilangan 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 bisa memasukkan keluaran readtable() ke dalam daftar.

```
>Table={{readtable("table.dat", ctok=ctok)};}
```

Dengan menggunakan kolom token yang sama dan token 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 ini.

```
>{c,i}=tablecol(MT,[5,6]);
```

Kita dapat menggunakan ini untuk mengekstrak kolom dari tabel untuk tabel baru

```
>j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok)
```

Person	Evaluation	Tip
2	g	1.8
3	g	1.8
6	g	2.8
7	vg	2.8
9	vg	1.8
11	vg	1.8
12	g	1.8
13	vg	1.8
14	g	1.8
15	g	0.8
16	g	2.8
20	vg	1.8
21	m	2.8
22	vg	1.8
23	g	2.8
24	m	1.8

Tentu saja, kita perlu mengekstrak tabel itu sendiri dari Daftar Tabel dalam kasus ini.

```
>MT=Table[1];
```

Tentu saja, kami juga dapat menggunakan untuk menentukan nilai rata-rata kolom atau nilai statistik lainnya.

```
>mean(tablecol(MT, 6))
```

2.175

Fungsi `getstatistics()` mengembalikan elemen dalam vektor, dan jumlahnya. Kita menerapkannya ke nilai "m" dan "f" di kolom kedua tabel kami.

```
>{xu, count}=getstatistics(tablecol(MT, 2)); xu, count,
```

```
[1, 3]
[12, 13]
```

Kita dapat 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 dalam v di baris ke-5

```
>MT1:=MT[selectrows(MT, 5, v)]; i:=sortedrows(MT1, 5);
```

Sekarang kita dapat mencetak tabel, dengan nilai yang diekstrasi dan diurutkan di kolom-5.

```
>writetable(MT1[i], labc=hd, ctok=ctok, tok=tok, wc=7);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
6	m	28	y	g	2.8	y
18	m	38	y	g	.	n
16	m	26	y	g	2.8	n
15	f	31	y	g	0.8	n
12	m	32	y	g	1.8	n
23	f	38	y	g	2.8	n
14	f	25	y	g	1.8	y
9	f	24	y	vg	1.8	y
7	f	31	y	vg	2.8	n
20	f	28	y	vg	1.8	n
22	f	28	y	vg	1.8	y
13	m	29	y	vg	1.8	y
11	f	23	y	vg	1.8	y

Untuk statistik berikutnya, kami ingin menghubungkan dua kolom dari tabel. Jadi kami mengekstrak kolom 2 dan 4 dan mengurutkan tabel.

```
>i=sortedrows(MT,[2,4]); ...
> writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)
```

m	n
m	n
m	n
m	n
m	n
m	n
m	n
m	y
m	Y
m	Y
m	Y
m	Y
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

Dengan getstatistics(), kita juga bisa menghubungkan hitungan dalam dua kolom tabel satu sama lain.

```
>MT24=tablecol(MT,[2,4]); ...
>{xu1,xu2,count}=getstatistics(MT24[1],MT24[2]); ...
>writetable(count,labr=tok[xu1],labc=tok[xu2])
```

	n	y
m	7	5
f	1	12

Tabel dapat ditulis ke file.

```
>filename="test.dat"; ...
>writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
```

Kemudian kita dapat 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 file tersebut.

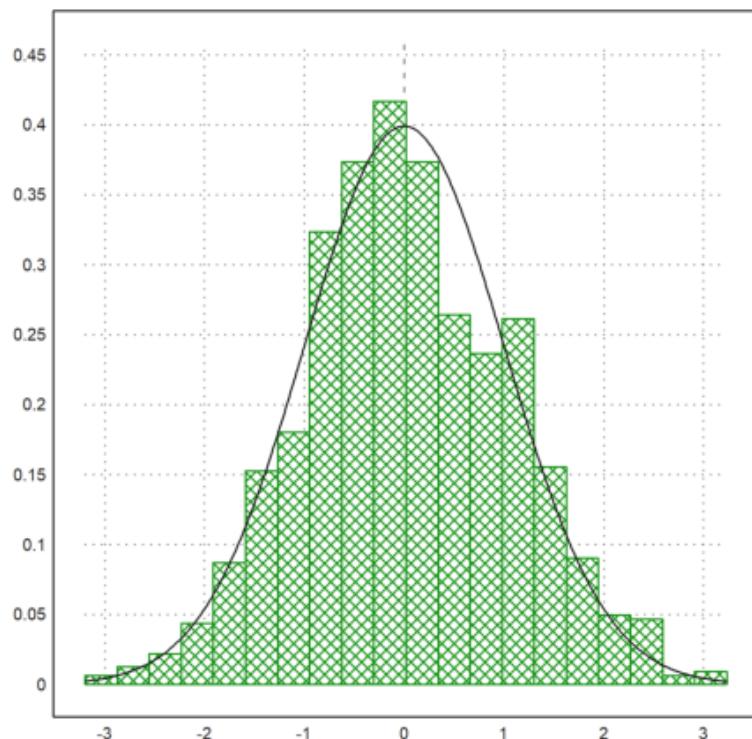
```
>fileremove(filename);
```

## Distribusi

---

Dengan plot2d, terdapat metode yang sangat mudah untuk memplot sebaran data eksperimen.

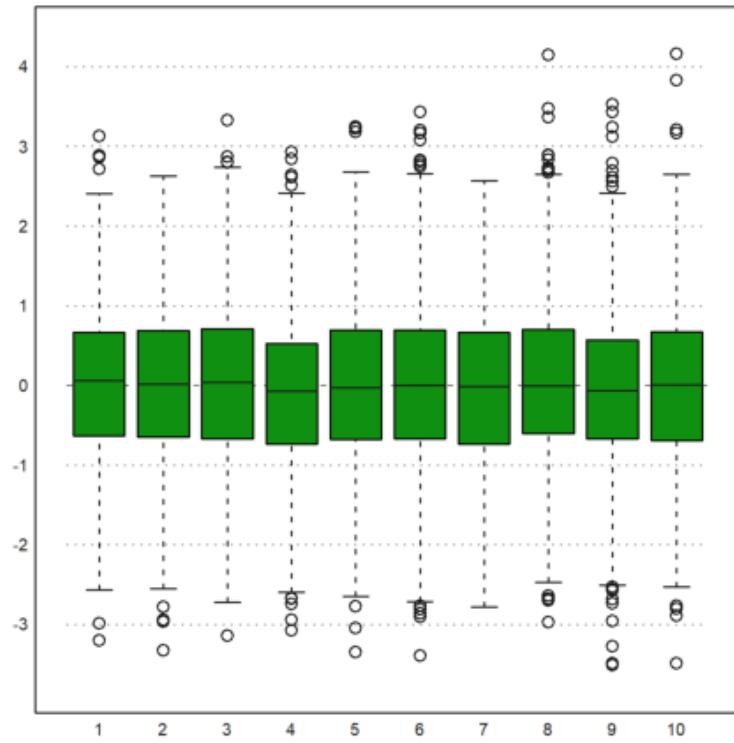
```
>p=normal(1,1000); //1000 random normal-distributed sample p
>plot2d(p,distribution=20,style="/"); // plot the random sample p
>plot2d("qnormal(x,0,1)",add=1); // add the standard normal distribution plot
```



Harap perhatikan perbedaan antara plot batang (sampel) dan kurva normal(distribusi nyata). Masukkan kembali tiga perintah untuk melihat hasil pengambilan sampel lainnya.

Berikut adalah perbandingan 10 simulasi dari 1000 nilai terdistribusi normal menggunakan apa yang disebut box plot. Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, dan outlier.

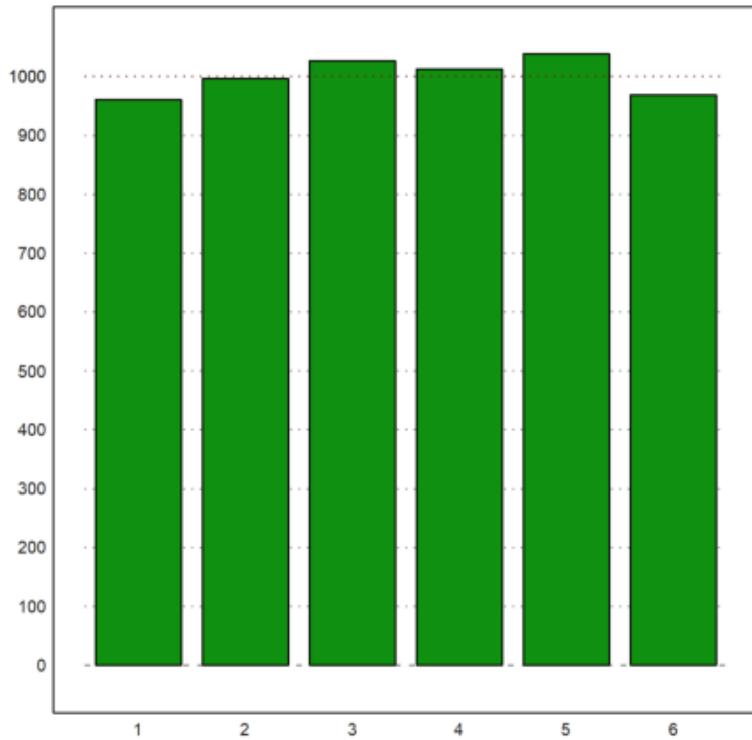
```
>p=normal(10,1000); boxplot(p);
```



Untuk menghasilkan bilangan bulat acak, Euler memiliki intrandom. Mari kita simulasi lemparan dadu dan plot distribusinya.

Kami menggunakan fungsi getmultiplicities v, x), yang menghitung seberapa sering elemen v muncul di x. Kemudian kita plot hasilnya menggunakan columnplot().

```
>k=intrandom(1,6000,6); ...
>columnspplot (getmultiplicities(1:6,k)); ...
>ygrid(1000,color=red):
```



Sementara intrandom ( $n, m, k$ ) mengembalikan bilangan bulat terdistribusi seragam dari 1 ke  $k$ , dimungkinkan untuk menggunakan distribusi bilangan bulat lain yang diberikan dengan `randpint()`.

Dalam contoh berikut, probabilitas 1,2,3 masing-masing adalah 0,4,0,1,0,5.

```
>randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,%)
```

```
[394, 107, 499]
```

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Simak referensinya.

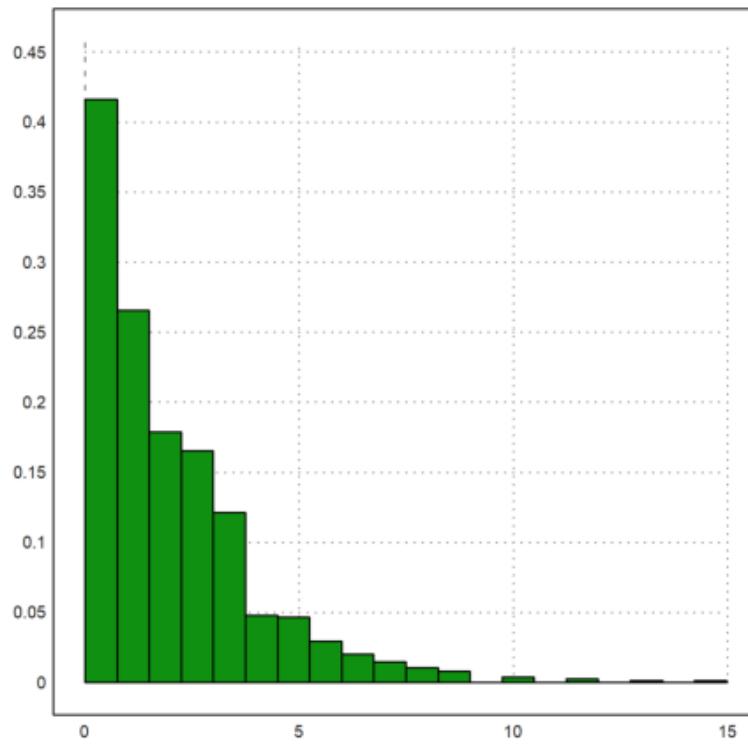
Misalnya, kami mencoba distribusi eksponensial. Variabel acak kontinu  $X$  dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

with parameter

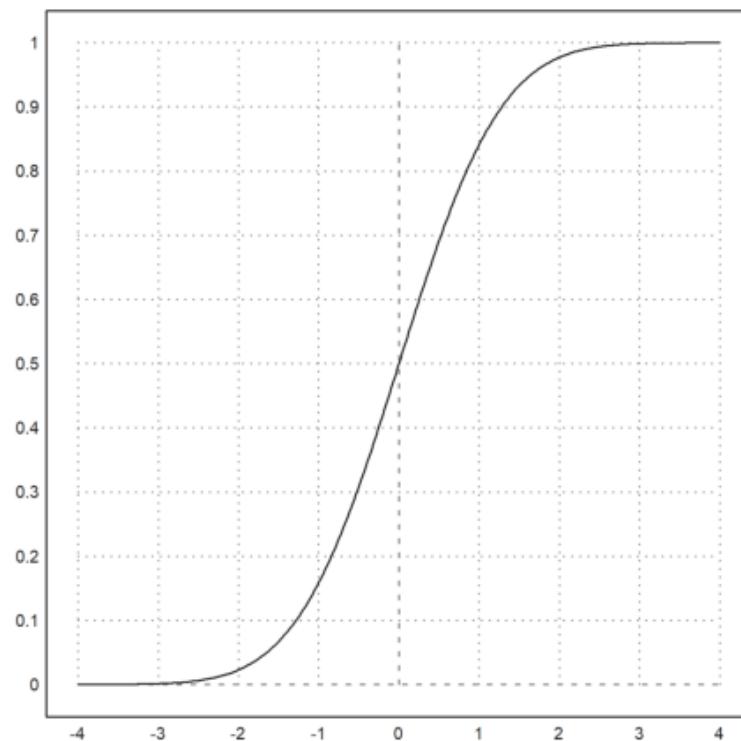
$$\lambda = \frac{1}{\mu}, \quad \mu \text{ is the mean, and denoted by } X \sim \text{Exponential}(\lambda).$$

```
>plot2d(randexponential(1,1000,2),>distribution):
```



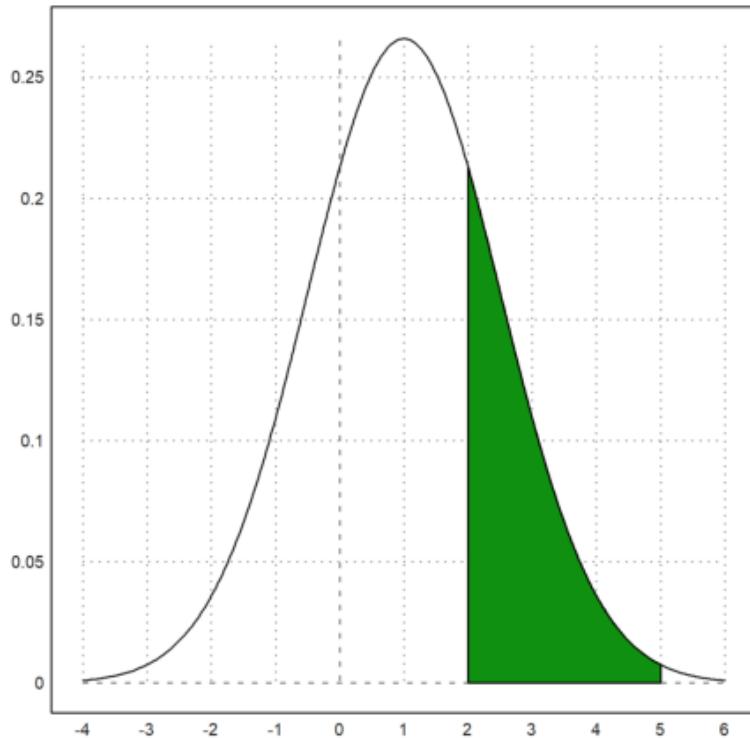
Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversnya.

```
>plot2d("normaldis", -4, 4) :
```



Berikut ini adalah salah satu cara untuk memplot sebuah kuantil.

```
>plot2d("qnormal(x,1,1.5)",-4,6); ...  
>plot2d("qnormal(x,1,1.5)",a=2,b=5,>add,>filled):
```



$$\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} dt.$$

Kemungkinan berada di area hijau adalah sebagai berikut.

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

0.248662156979

Ini dapat dihitung secara numerik dengan integral berikut.

$$\int_2^5 \frac{1}{1.5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-1}{1.5})^2} dx.$$

```
>gauss("qnormal(x,1,1.5)",2,5)
```

0.248662156979

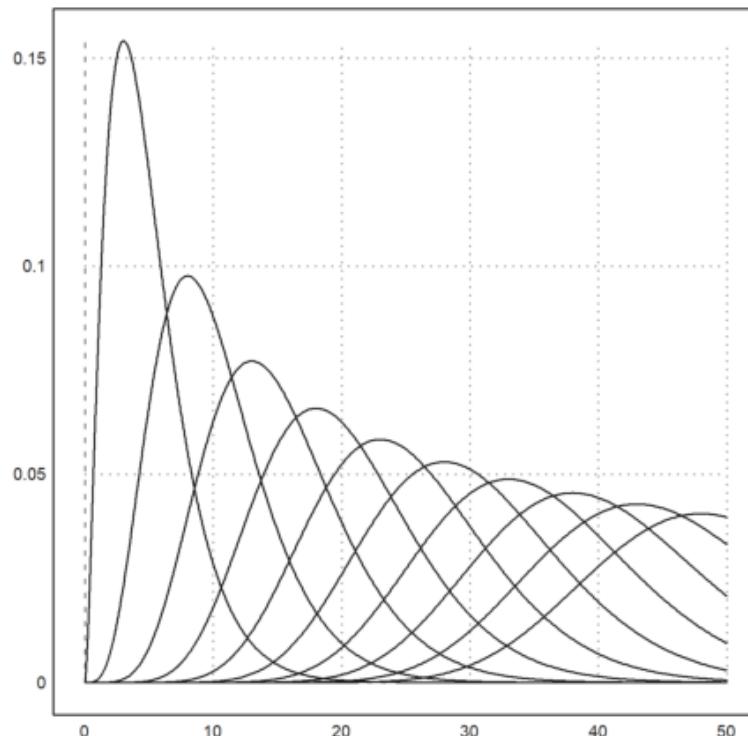
Mari kita bandingkan distribusi binomial dengan distribusi normal dari mean dan deviasi yang sama. Fungsi invbindis () memecahkan interpolasi linier antara nilai integer.

```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

```
525.516721219  
526.007419394
```

Fungsi qdis () adalah kepadatan dari distribusi chi-kuadrat. Seperti biasa, Euler memetakan vektor ke fungsi ini. Jadi kita mendapatkan plot dari 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 yang akurat untuk mengevaluasi distribusi. Mari kita periksa chidis () dengan integral. Penamaan mencoba untuk konsisten. Misalnya.,

- distribusi chi-kuadrat 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

Untuk menentukan distribusi diskrit Anda sendiri, Anda dapat menggunakan metode berikut. Pertama kita mengatur fungsi distribusi.

```
>wd = 0 | ((1:6)+[-0.01,0.01,0,0,0,0])/6
```

```
[0, 0.165, 0.335, 0.5, 0.666667, 0.833333, 1]
```

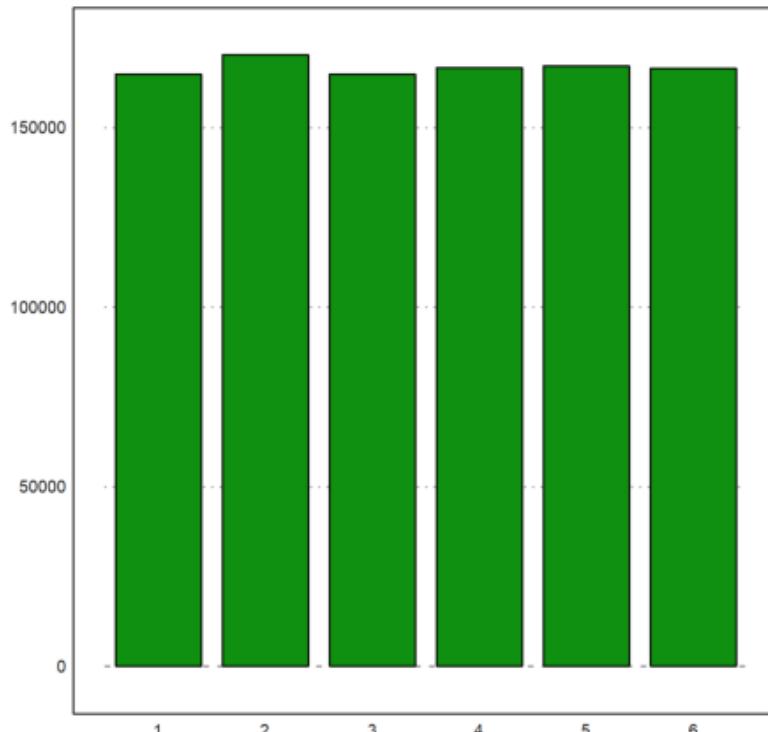
Artinya dengan probabilitas  $wd[i+1]-wd[i]$  kita menghasilkan nilai acak i.

Ini hampir merupakan distribusi yang seragam. Mari kita tentukan generator nomor acak untuk ini. Fungsi find (v, x) menemukan nilai x pada vektor v. Fungsi ini juga berlaku untuk vektor x.

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahannya begitu halus sehingga kita hanya melihatnya dengan sangat banyak iterasi.

```
>columnsplot (getmultiplicities(1:6,wrongdice(1,1000000))):
```



Berikut adalah fungsi sederhana untuk memeriksa distribusi seragam nilai 1 ... K dalam v. Kami menerima hasilnya, jika untuk semua frekuensi

$$\left| f_i - \frac{1}{K} \right| < \frac{\delta}{\sqrt{n}}.$$

```
>function checkrandom (v, delta=1) ...
```

```
K=max(v); n=cols(v);
fr=getfrequencies(v,1:K);
return max(fr/n-1/K)<delta/sqrt(n);
endfunction
```

Memang fungsinya menolak distribusi seragam.

```
>checkrandom(wrongdice(1,1000000))
```

0

Dan itu menerima generator acak bawaan.

```
>checkrandom(intrandom(1,1000000,6))
```

1

Kami dapat menghitung distribusi binomial. Pertama ada binomials() yang mengembalikan probabilitas i atau kurang dari n percobaan.

```
>bindis(410,1000,0.4)
```

0.751401349654

Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p. Tingkat defaultnya adalah alfa.

Arti dari interval ini adalah jika p berada di luar interval maka hasil observasi 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 besar, penjumlahan langsung tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

0.751401349655

Omong-omong, invbinsum() menghitung kebalikan dari binomials().

```
>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])
```

394	82	524
418	92	490
445	90	465
403	114	483
405	95	500
384	107	509
414	93	493
419	90	491
394	101	505
382	103	515

## Merencanakan Data

---

Untuk memplot data, kita coba hasil pemilu Jerman sejak 1990, diukur dalam kursi.

```
>BW := [ ...  
>1990,662,319,239,79,8,17; ...  
>1994,672,294,252,47,49,30; ...  
>1998,669,245,298,43,47,36; ...  
>2002,603,248,251,47,55,2; ...  
>2005,614,226,222,61,51,54; ...  
>2009,622,239,146,93,68,76; ...  
>2013,631,311,193,0,63,64];
```

Untuk pesta, kita menggunakan serangkain nama.

```
>P := ["CDU/CSU", "SPD", "FDP", "Gr", "Li"];
```

Mari kita cetak persentase dengan baik.

Pertama kami mengekstrak kolom yang diperlukan. Kolom 3 sd 7 adalah kursi masing-masing partai, dan kolom 2 adalah jumlah kursi. kolom adalah tahun pemilihan.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Kemudian kami mencetak statistik dalam bentuk tabel. Kami menggunakan nama sebagai tajuk kolom, dan tahun sebagai tajuk 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

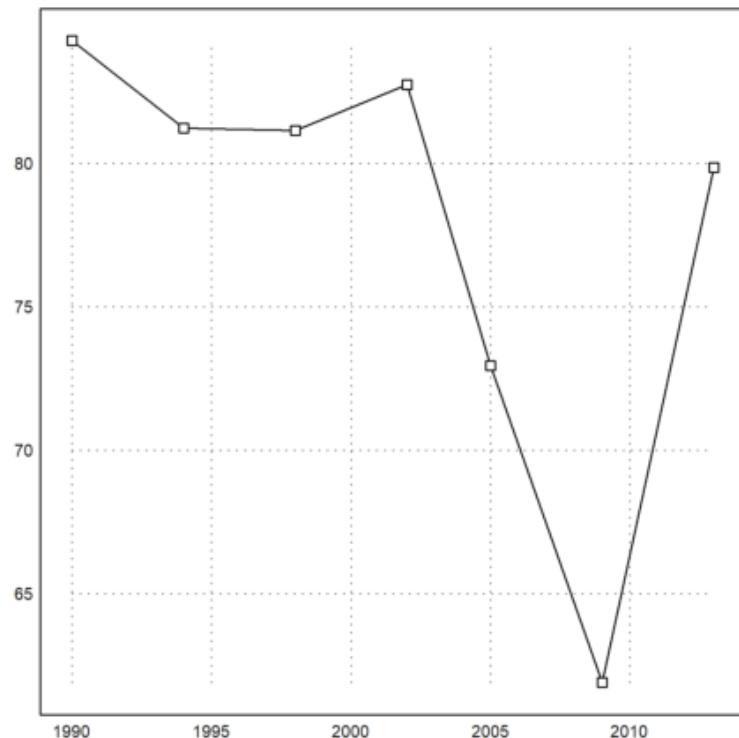
Perkalian matriks berikut mengekstrak jumlah persentase dari dua partai besar yang menunjukkan bahwa partai kecil telah mendapatkan footage di parlemen hingga tahun 2009.

```
>BT1:=(BT.[1;1;0;0])'*100
```

```
[84.29, 81.25, 81.1659, 82.7529, 72.9642, 61.8971, 79.8732]
```

Ada juga plot statistik sederhana. Kami menggunakan untuk menampilkan garis dan titik secara bersamaan. Alternatifnya adalah memanggil plot2d dua kali dengan > add.

```
>statplot(YT,BT1,"b"):
```

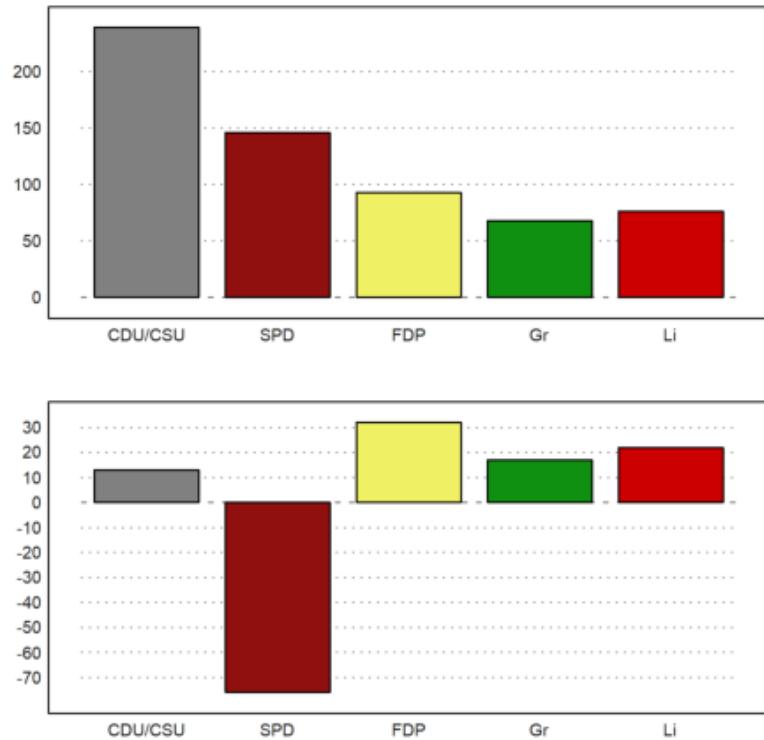


Tentukan beberapa warna untuk setiap pesta.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

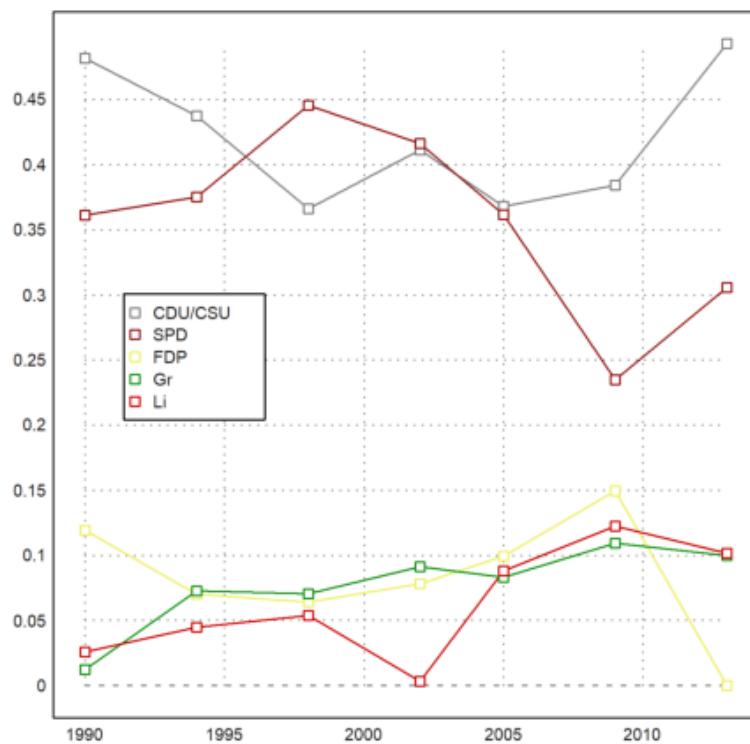
Sekarang kita bisa memplot hasil Pemilu 2009 dan perubahannya menjadi satu plot menggunakan gambar. Kita dapat menambahkan vektor kolom ke setiap plot.

```
>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):
```



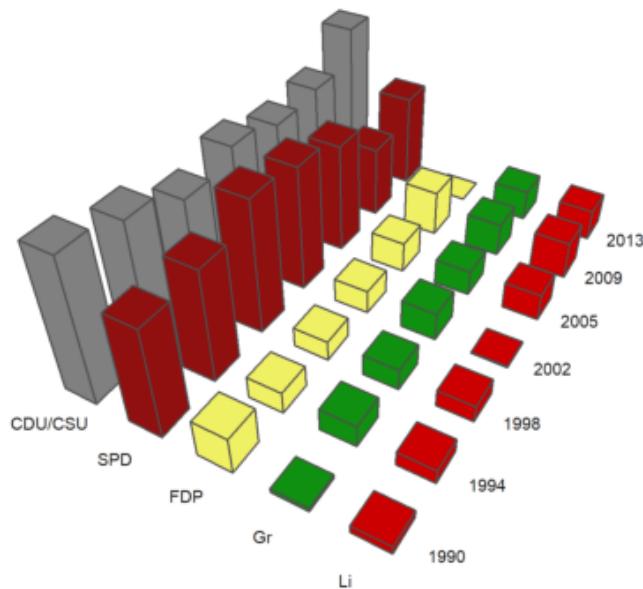
Plot data menggabungkan deretan data statistik dalam satu plot.

```
>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) :
```



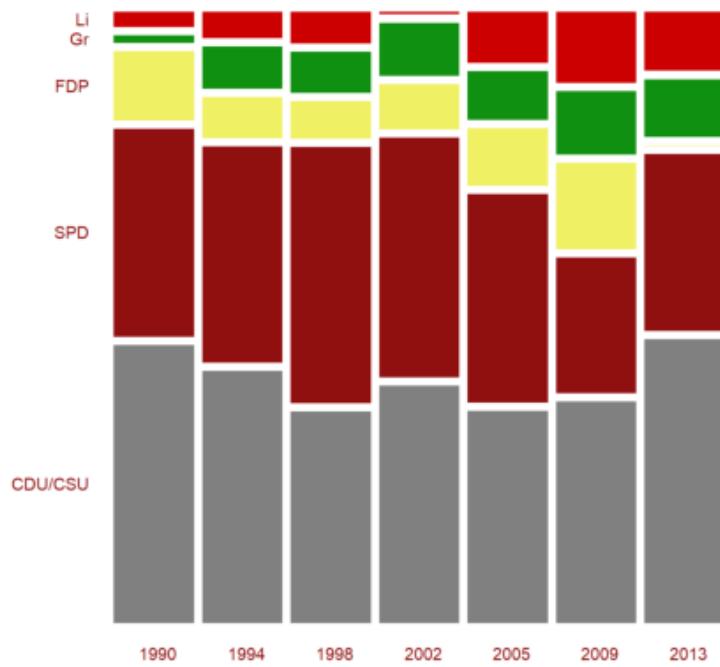
Plot kolom 3D menampilkan baris data statistik dalam bentuk kolom. Kami memberikan label untuk baris dan kolom. sudut adalah sudut pandang.

```
>columnsplot3d(BT,scols=P,srows=YT, ...
>  angle=30°,ccols=CP):
```



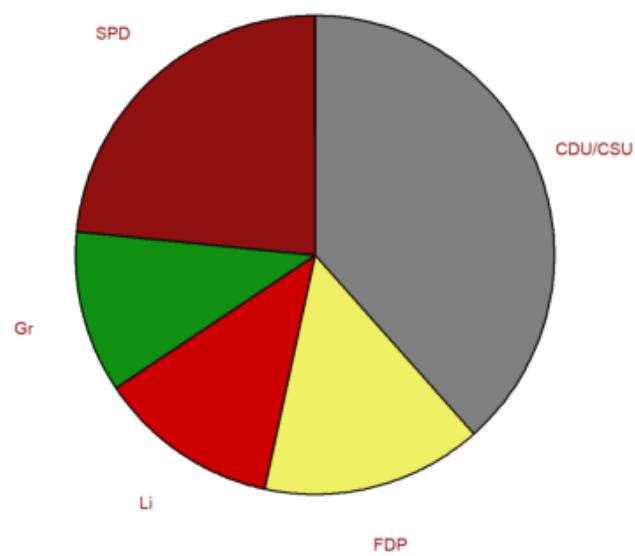
Representasi lainnya adalah plot mosaik. Perhatikan bahwa kolom plot mewakili kolom matriks di sini. Karena panjangnya label CDU / CSU, kami mengambil jendela yang lebih kecil dari biasanya.

```
>shrinkwindow(>smaller); ...
>mosaicplot(BT',srows=YT,scols=P,color=CP,style="#" ); ...
>shrinkwindow():
```



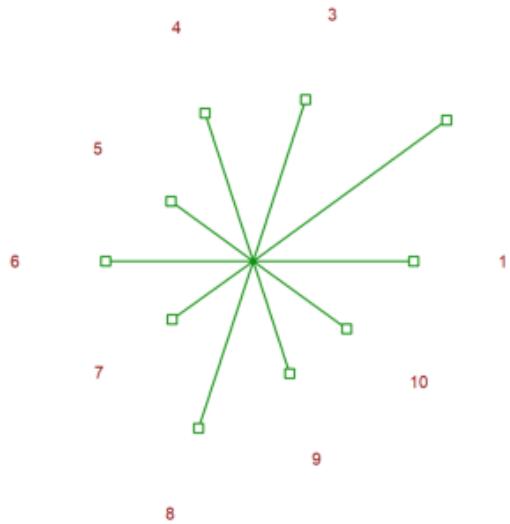
Kami juga bisa membuat diagram pie. Karena hitam dan kuning membentuk koalisi, kami menyusun ulang elemen-elemennya.

```
>i=[1, 3, 5, 4, 2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```



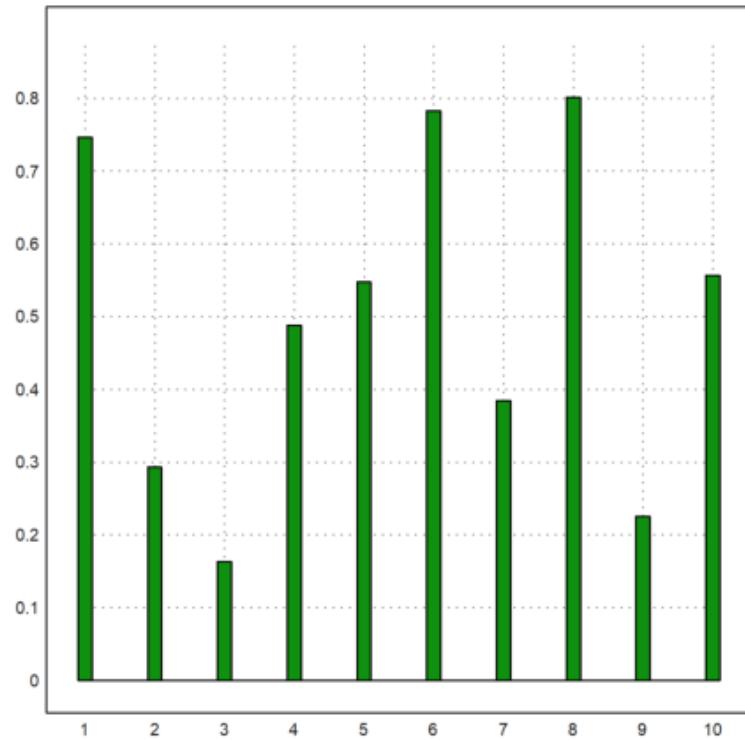
Berikut ini jenis plot lainnya.

```
>starplot(normal(1,10)+4,lab=1:10,>rays) :
```



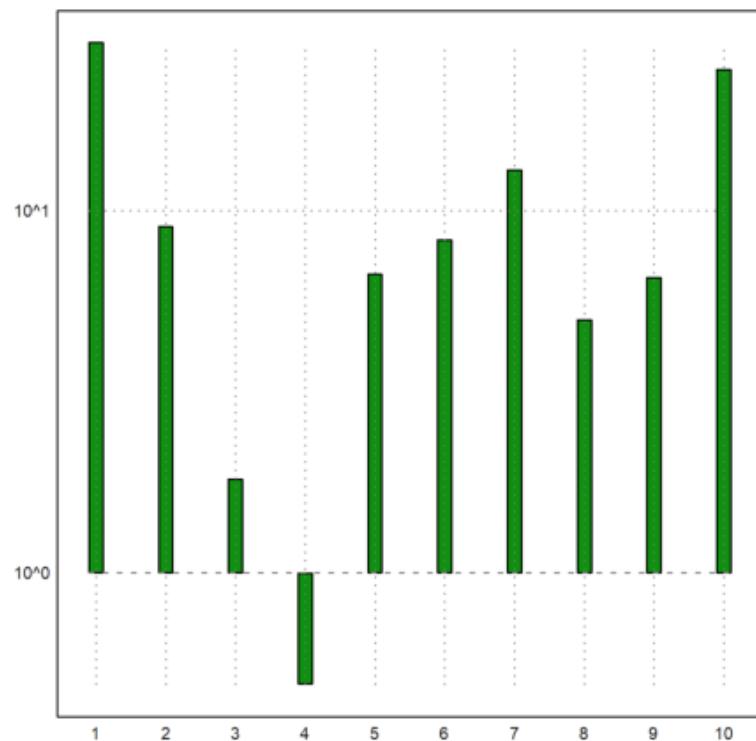
Beberapa plot di plot2d bagus untuk statika. Berikut adalah plot impuls data acak, didistribusikan secara seragam di [0,1].

```
>plot2d(makeimpulse(1:10,random(1,10)),>bar) :
```



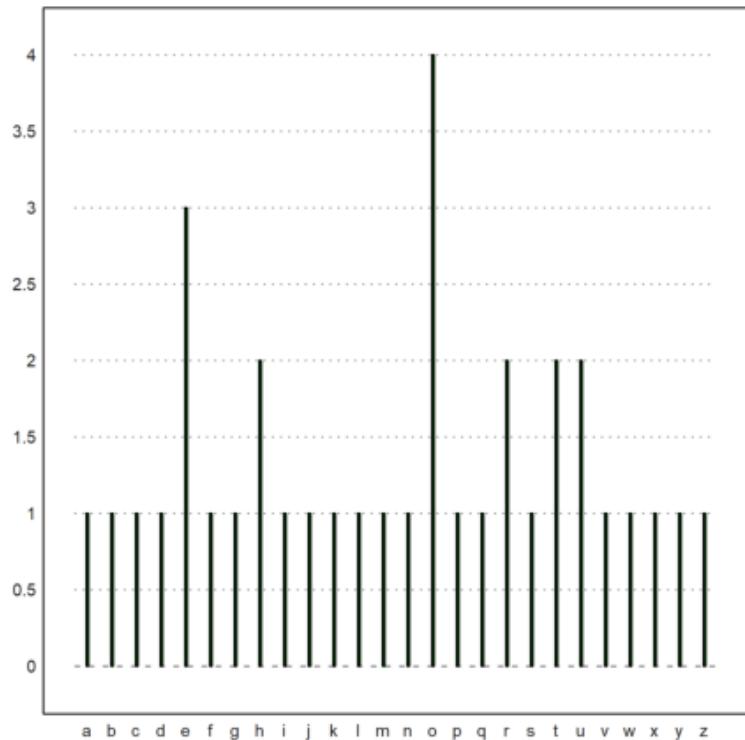
Tetapi untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

```
>logimpulseplot(1:10,-log(random(1,10))*10):
```



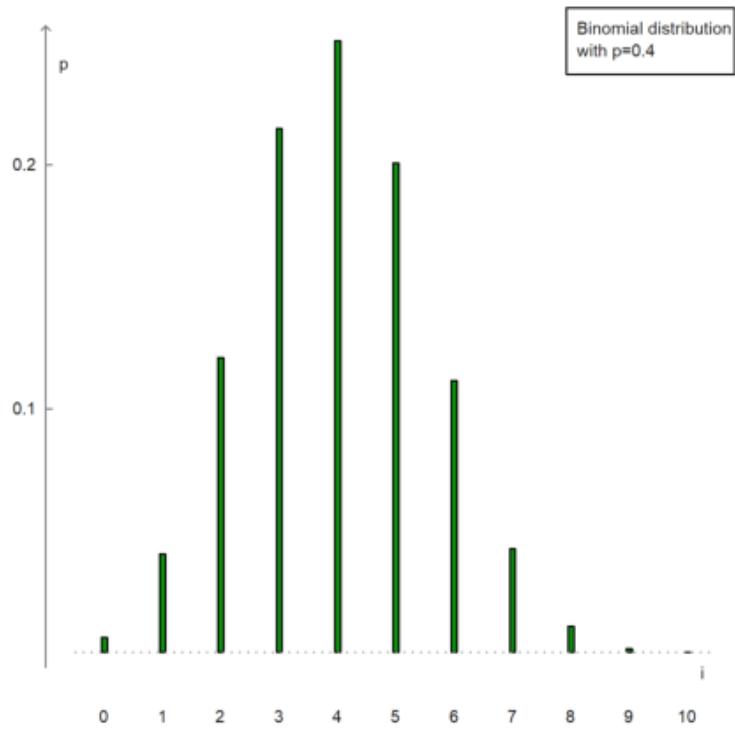
Fungsi `columnplot()` lebih mudah digunakan, karena hanya membutuhkan vektor nilai. Selain itu, ia dapat mengatur labelnya menjadi apa pun yang kami inginkan, kami telah menunjukkannya di tutorial ini. Berikut adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan membuat plot 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):
```



Sumbu juga dapat diatur secara manual.

```
>n=10; p=0.4; i=0:n; x=bin(n,i)*p^i*(1-p)^(n-i); ...
>columnsplot(x,lab=i,width=0.05,<frame,<grid); ...
>yaxis(0,0:0.1:1,style="->",>left); xaxis(0,style="."); ...
>label("p",0,0.25), label("i",11,0); ...
>textbox(["Binomial distribution","with p=0.4"]):
```



Berikut ini adalah cara untuk memplot frekuensi bilangan dalam sebuah vektor. Kami membuat vektor bilangan bulat bilangan acak 1 hingga 6.

```
>v:=intrandom(1,10,10)
```

```
[5, 1, 8, 6, 6, 10, 9, 6, 8, 3]
```

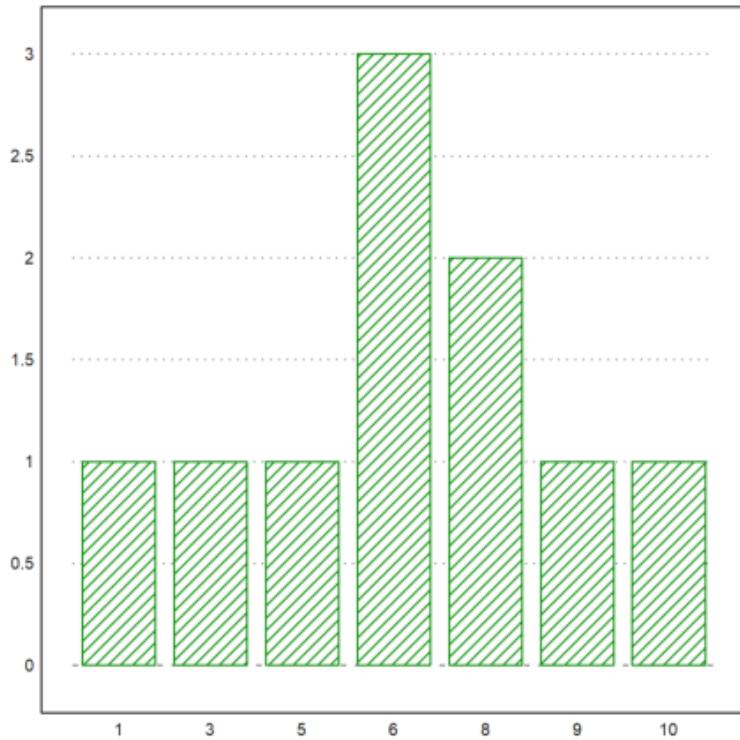
Kemudian ekstrak nomor unik di v.

```
>vu:=unique(v)
```

```
[1, 3, 5, 6, 8, 9, 10]
```

Dan plot frekuensi dalam plot kolom.

```
>columnsplot(getmultiplicities(vu,v),lab=vu,style="/"):
```



Kami ingin mendemonstrasikan fungsi untuk distribusi nilai empiris.

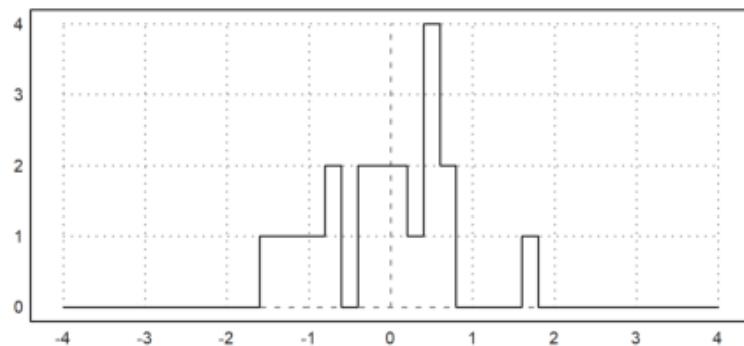
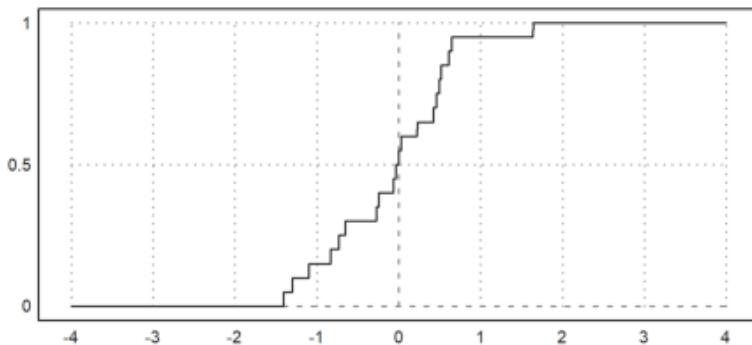
```
>x=normal(1,20);
```

Fungsi empdist ( $x$ ,  $vs$ ) membutuhkan array nilai yang diurutkan. Jadi kita harus mengurutkan  $x$  sebelum dapat menggunakannya.

```
>xs=sort(x);
```

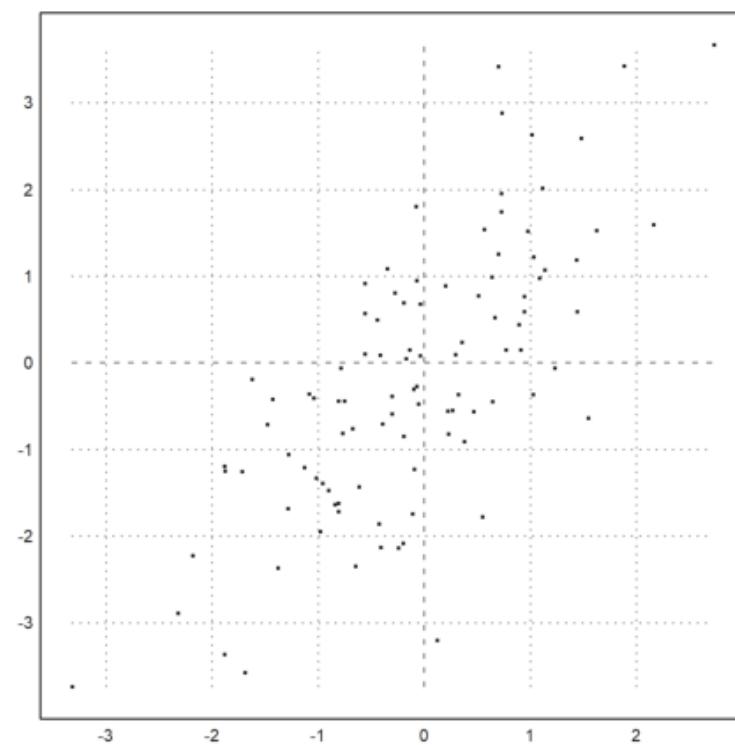
Kemudian kami memplot distribusi empiris dan beberapa batang kepadatan ke dalam satu plot. Alih-alih plot batang untuk distribusi kami menggunakan plot gigi gergaji kali ini.

```
>figure(2,1); ...
>figure(1); plot2d("empdist",-4,4;xs); ...
>figure(2); plot2d(histo(x,v=-4:0.2:4,<bar>)); ...
>figure(0):
```



Plot pencar mudah dilakukan di Euler dengan plot titik biasa. Grafik berikut menunjukkan bahwa  $X$  dan  $X+Y$  berkorelasi positif dengan jelas.

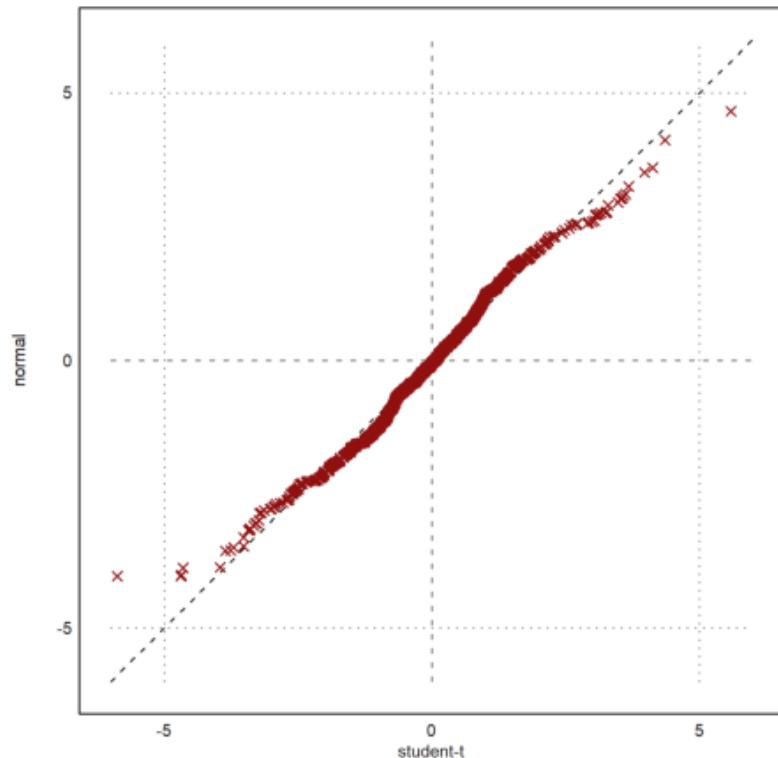
```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style="."):
```



Seringkali, kami ingin membandingkan dua sampel dari distribusi yang berbeda. Ini dapat dilakukan dengan plot-kuantil-kuantil.

Untuk pengujian, kami mencoba distribusi t siswa dan distribusi eksponensial.

```
>x=randt(1,1000,5); y=randnormal(1,1000,mean(x),dev(x)); ...
>plot2d("x",r=6,style="--",yl="normal",xl="student-t",>vertical); ...
>plot2d(sort(x),sort(y),>points,color=red,style="x",>add):
```

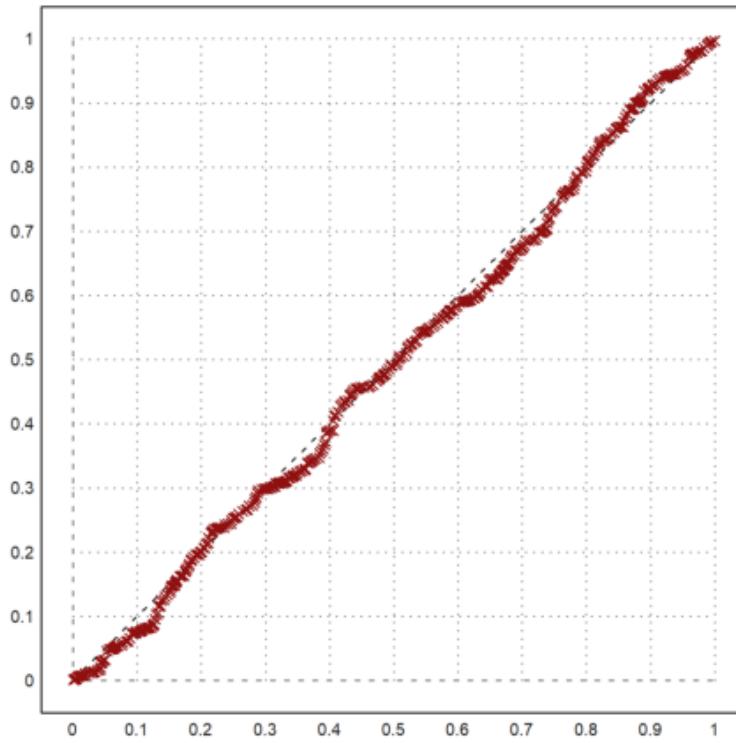


Plot dengan jelas menunjukkan bahwa nilai terdistribusi normal cenderung lebih kecil di ujung yang ekstrim. Jika kita memiliki dua distribusi dengan ukuran berbeda, kita dapat memperbesar yang lebih kecil atau mengecilkan yang lebih besar. Fungsi berikut bagus untuk keduanya. Ini mengambil nilai median dengan persentase antara 0 dan 1.

```
>function medianexpand (x,n) := median(x,p=linspace(0,1,n-1));
```

Mari 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

Kami ingin membandingkan ukuran yang tidak berbobot dan berbobot. Pertama koefisien kesesuaian linier.

```
>p=polyfit(x, y, 1)
```

```
[ 0.733333,  0.812121]
```

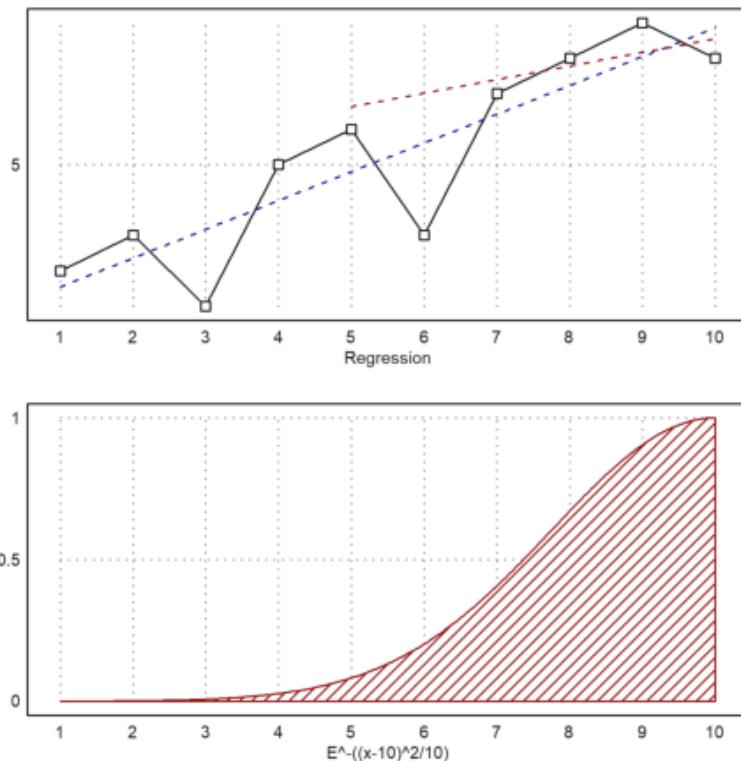
Sekarang koefisien dengan bobot yang menekankan nilai terakhir.

```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

[4.71566, 0.38319]

Kami menempatkan semuanya ke dalam satu plot untuk titik 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):
```



Untuk contoh lain kami 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 menyetel terjemahan yang tepat alih-alih membiarkan readtable () mengumpulkan terjemahan.

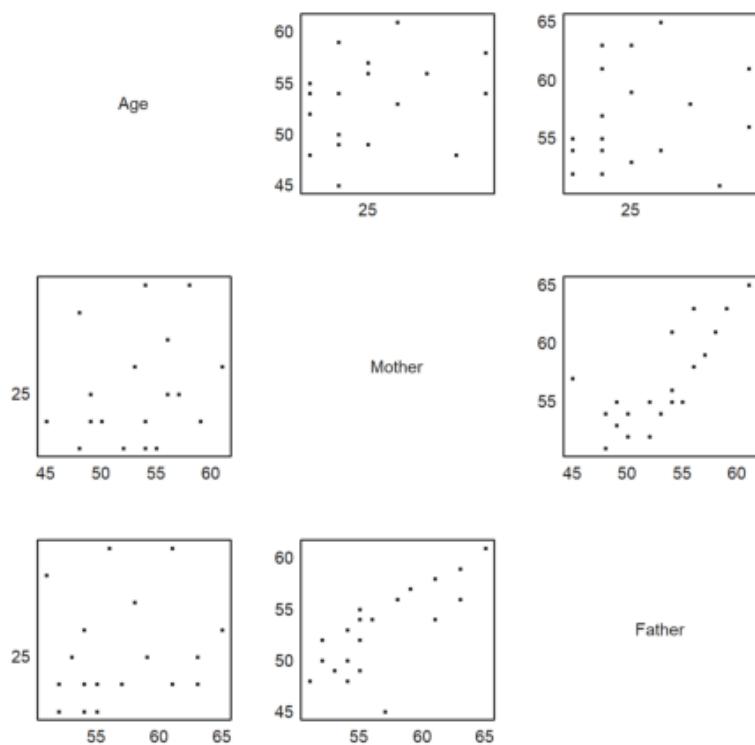
```
>{MS,hd}:=readtable("table1.dat",tok2:=[ "m", "f" ]); ...
>writetable(MS,labc=hd,tok2:=[ "m", "f" ]);
```

Person	Sex	Age	Mother	Father	Siblings
1	m	29	58	61	1
2	f	26	53	54	2
3	m	24	49	55	1
4	f	25	56	63	3

5	f	25	49	53	0
6	f	23	55	55	2
7	m	23	48	54	2
8	m	27	56	58	1
9	m	25	57	59	1
10	m	24	50	54	1
11	f	26	61	65	1
12	m	24	50	52	1
13	m	29	54	56	1
14	m	28	48	51	2
15	f	23	52	52	1
16	m	24	45	57	1
17	f	24	59	63	0
18	f	23	52	55	1
19	m	24	54	61	2
20	f	23	54	55	1

How do the ages depend on each other? A first impression comes from a pairwise scatterplot.

```
>scatterplots(tablecol(MS, 3:5), hd[3:5]):
```



Jelas bahwa usia bapak dan ibu saling bergantung. Mari kita tentukan dan plot garis regresi.

```
>cs:=MS[, 4:5]'; ps:=polyfit(cs[1], cs[2], 1)
```

```
[17.3789, 0.740964]
```

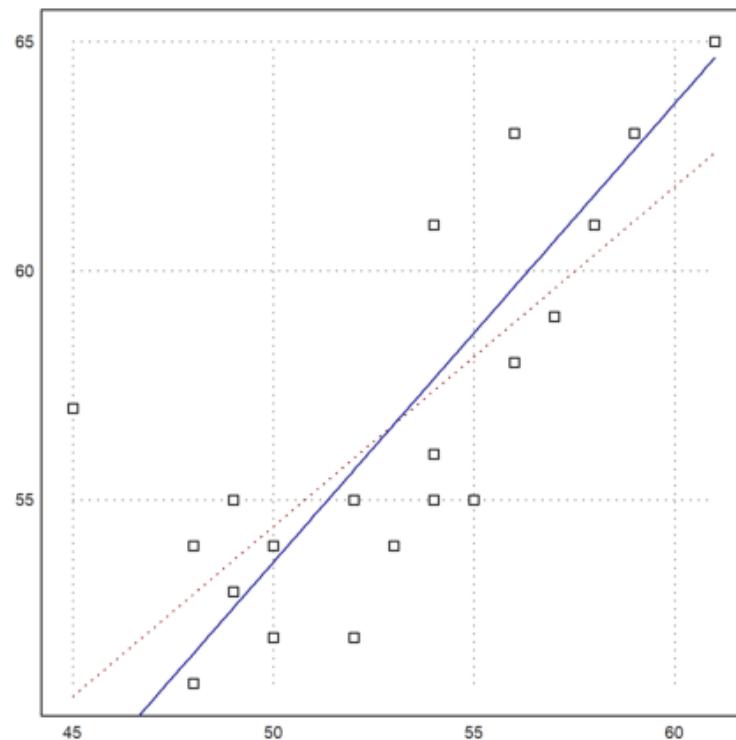
Ini jelas model yang salah. Garis regresinya adalah  $s = 17 + 0.74t$ , dimana  $t$  adalah umur ibu dan  $s$  umur bapak. Perbedaan usia mungkin sedikit bergantung pada usianya, tapi tidak terlalu banyak. Sebaliknya, kami menduga fungsi seperti  $s = a + t$ . Maka  $a$  adalah mean dari  $s-t$ . Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

```
>da:=mean(cs[2]-cs[1])
```

3.65

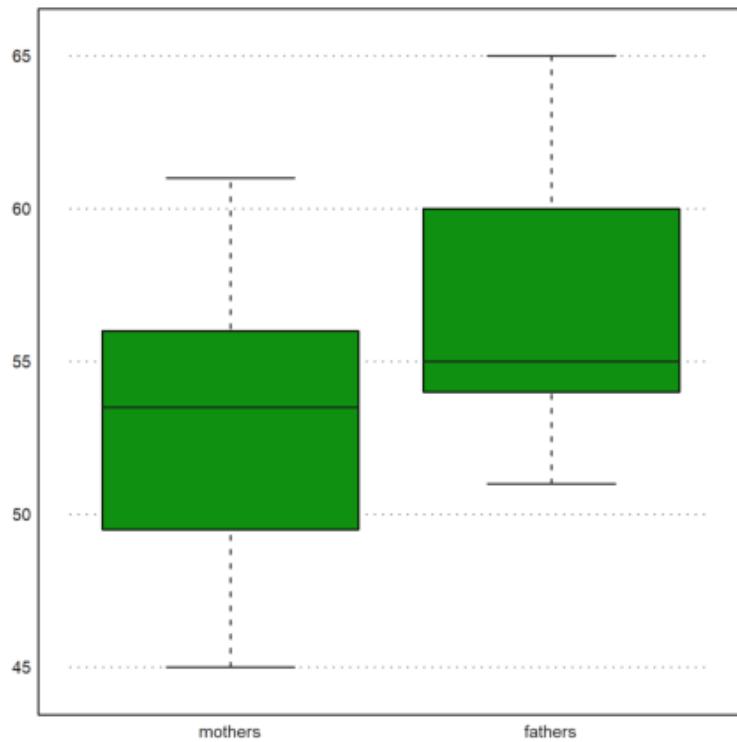
Mari kita plot ini menjadi satu plot pencar.

```
>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. Ini hanya menunjukkan, bahwa umurnya berbeda.

```
>boxplot(cs, ["mothers", "fathers"]):
```



Menariknya, perbedaan median tidak sebesar perbedaan rata rata.

```
>median(cs[2])-median(cs[1])
```

1.5

Koefisien korelasi menunjukkan korelasi positif.

```
>correl(cs[1],cs[2])
```

0.7588307236

Korelasi barisan adalah ukuran untuk urutan yang sama di kedua vektor. Ini juga cukup positif.

```
>rankcorrel(cs[1],cs[2])
```

0.758925292358

## Membuat Fungsi baru

---

Tentu saja, bahasa EMT dapat digunakan untuk memprogram fungsi baru. Misalnya, kami mendefinisikan fungsi kemiringan.

$$sk(x) = \frac{\sqrt{n} \sum_i (x_i - m)^3}{(\sum_i (x_i - m)^2)^{3/2}}$$

dimana m adalah mean 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))
```

```
0.521806329961
```

Berikut adalah fungsi lain, yang disebut koefisien kemiringan Pearson.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)  
>skew1(data)
```

```
0.573102925949
```

## Simulasi Monte Carlo

---

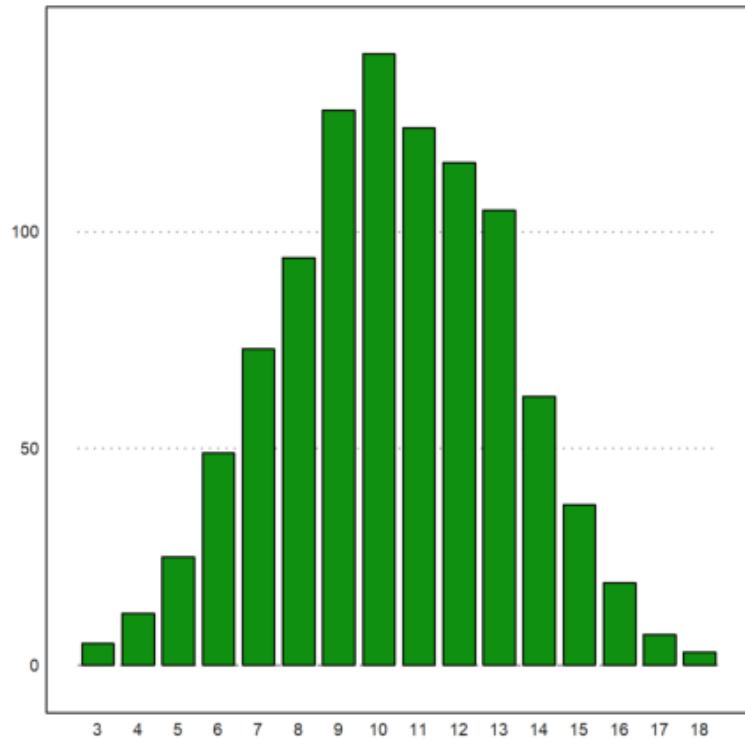
Euler dapat digunakan untuk mensimulasikan peristiwa acak. Kami telah melihat contoh sederhana di atas. Ini satu lagi, yang mensimulasikan 1000 kali lemparan 3 dadu, dan menanyakan distribusi jumlahnya.

```
>ds:=sum(intrandom(1000,3,6))'; fs=getmultiplicities(3:18,ds)
```

```
[5, 12, 25, 49, 73, 94, 128, 141, 124, 116, 105, 62, 37,  
19, 7, 3]
```

Kita bisa merencakkannya sekarang.

```
>columnsplot(fs,lab=3:18):
```



Untuk menentukan distribusi yang diharapkan tidaklah mudah. Kami menggunakan rekursi lanjutan untuk ini.

Fungsi berikut menghitung banyaknya cara bilangan k dapat direpresentasikan sebagai jumlah dari n bilangan dalam rentang 1 hingga m. Ini bekerja secara rekursif dengan cara yang jelas.

```
>function map countways (k; n, m) ...
  if n==1 then return k>=1 && k<=m
  else
    sum=0;
    loop 1 to m; sum=sum+countways(k-#,n-1,m); end;
    return sum;
  end;
endfunction
```

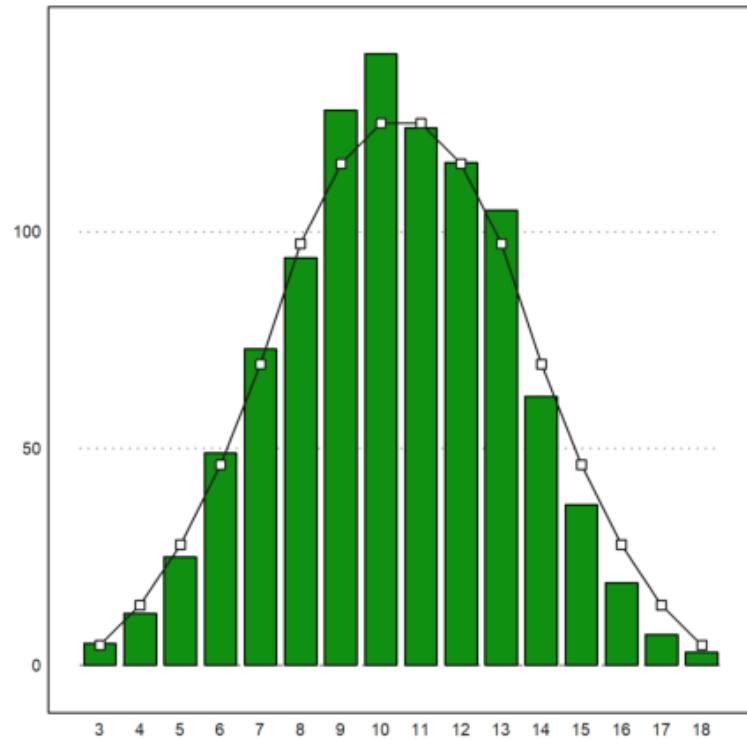
Ini adalah hasil dari tiga lemparan dadu.

```
>cw=countways(3:18,3,6)
```

```
[1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3,
1]
```

Kami menambahkan nilai yang diharapkan ke plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```



Untuk simulasi lain, deviasi nilai rata-rata  $n$  0-1-variabel acak terdistribusi normal adalah  $1 / \sqrt{n}$ .

```
>longformat; 1/sqrt(10)
```

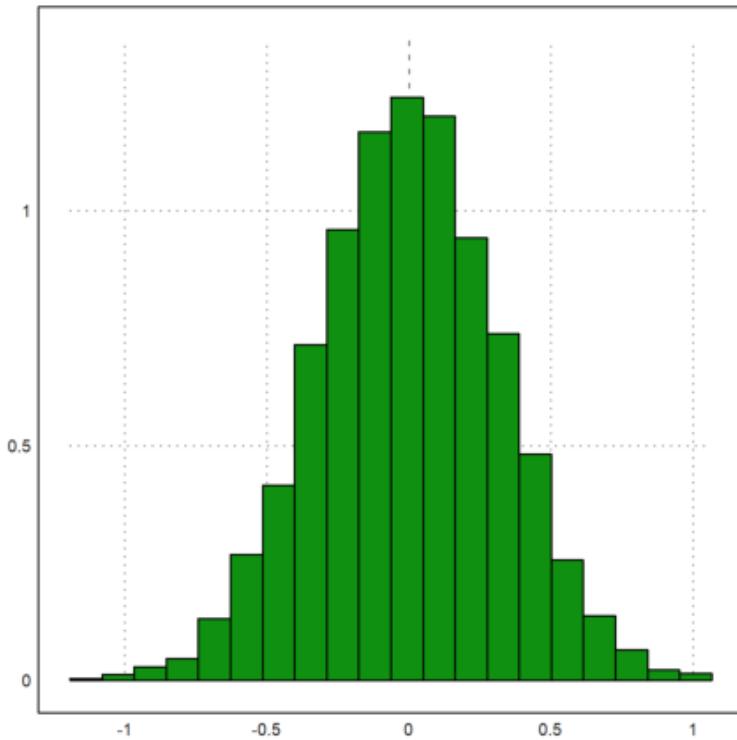
0.316227766017

Mari kita periksa dengan simulasi. Kami menghasilkan 10.000 kali 10 vektor acak.

```
>M=normal(10000,10); dev(mean(M)')
```

0.318861419326

```
>plot2d(mean(M)',>distribution):
```



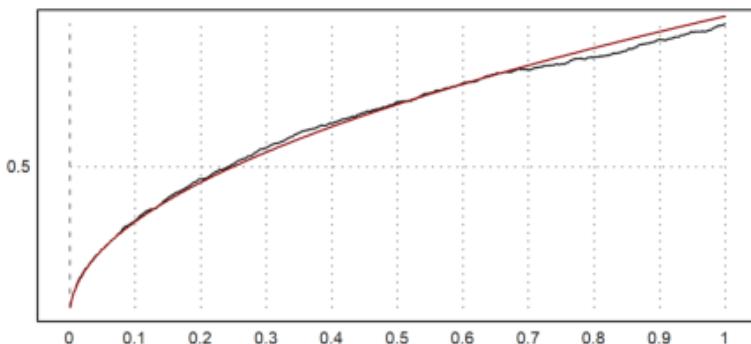
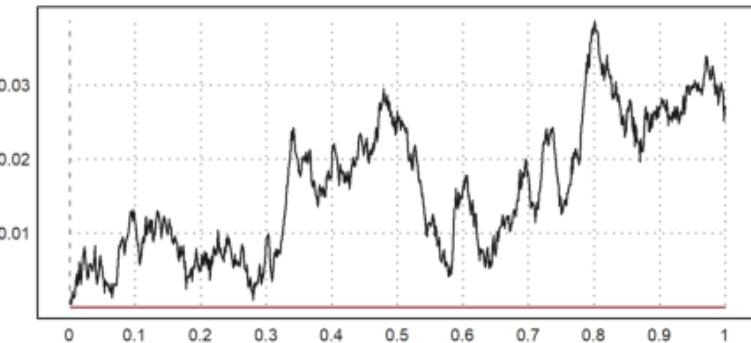
Median dari 10 bilangan acak terdistribusi normal 0-1 memiliki deviasi yang lebih besar.

```
>dev(median(M)')
```

0.376651504162

Karena kami dapat dengan mudah membuat jalan acak, kami dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami kemudian memplot deviasi standar dan mean dari 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):
```



## Tes

---

Tes adalah alat penting dalam statistik. Di Euler, banyak tes yang diterapkan. Semua pengujian ini mengembalikan kesalahan yang kami terima jika kami menolak hipotesis nol.

Sebagai contoh, kami menguji lemparan dadu untuk distribusi seragam. Pada 600 lemparan, kami mendapatkan nilai berikut, yang kami masukkan ke dalam uji chi-square.

```
>chitest([90,103,114,101,103,89],dup(100,6)')
```

0.498830517952

Uji chi-square juga memiliki mode, yang menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya harusnya hampir sama. Parameter  $\mathbf{>p}$  mengartikan vektor  $\mathbf{y}$  sebagai vektor probabilitas.

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

0.517

Kesalahan ini terlalu besar. Jadi kita tidak bisa menolak distribusi seragam. Ini tidak membuktikan bahwa dadu kami adil. Tapi kita tidak bisa menolak hipotesis kita.

Selanjutnya kami menghasilkan 1000 lemparan dadu menggunakan generator nomor acak, dan melakukan tes yang sama.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

0.159910723681

Mari kita uji nilai rata-rata 100 dengan uji-t.

```
>s=200+normal([1,100])*10; ...
>ttest(mean(s),dev(s),100,200)
```

0.353414299977

Fungsi `ttest()` membutuhkan nilai mean, deviasi, jumlah data, dan nilai mean untuk diuji. Sekarang mari kita periksa dua pengukuran untuk mean yang sama. Kami menolak hipotesis bahwa mereka memiliki mean yang sama, jika hasilnya  $<0,05$ .

```
>tcomparedata(normal(1,10),normal(1,10))
```

0.268166832384

Jika kita menambahkan bias ke satu distribusi, kita mendapatkan lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

2.17064628392e-05

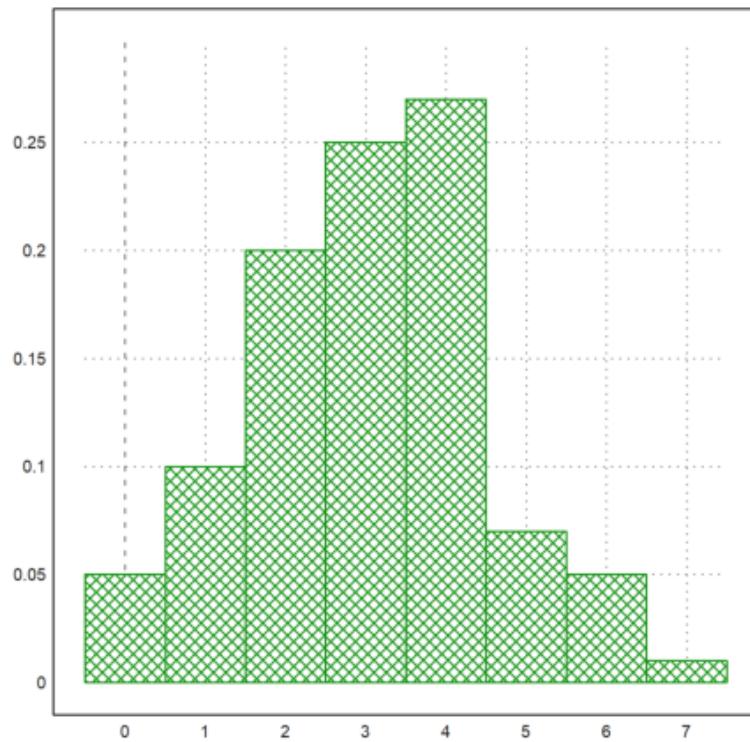
Dalam contoh berikutnya, kami menghasilkan 20 lemparan dadu acak 100 kali dan menghitung yang ada di dalamnya. Harus ada rata-rata  $20/6 = 3,3$ .

```
>R=random(100,20); R=sum(R*6<=1); mean(R)
```

3.05

Sekarang kami membandingkan jumlah satuan dengan distribusi binomial. Pertama kami memplot distribusi satu.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\\"":
```



```
>t=count(R,21);
```

Kemudian kami menghitung nilai yang diharapkan.

```
>n=0:20; b=bin(20,n)*(1/6)^n*(5/6)^(20-n)*100;
```

Kita harus mengumpulkan beberapa nomor untuk mendapatkan kategori yang cukup besar.

```
>t1=sum(t[1:2])|t[3:7]|sum(t[8:21]); ...
>b1=sum(b[1:2])|b[3:7]|sum(b[8:21]);
```

Uji chi-square menolak hipotesis bahwa distribusi kita adalah distribusi binomial, jika hasilnya <0,05.

```
>chitest(t1,b1)
```

0.262911182138

Contoh berikut berisi hasil dari dua kelompok orang (misalnya laki-laki dan perempuan) yang memberikan suara untuk satu dari enam partai.

```
>A=[23,37,43,52,64,74;27,39,41,49,63,76]; ...
> writetable(A,wc=6,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	23	37	43	52	64	74
f	27	39	41	49	63	76

Kita ingin menguji independensi suara dari jenis kelamin. Tes tabel chi<sup>2</sup> melakukan ini. Hasilnya adalah cara yang besar untuk menolak kemerdekaan. Jadi kami tidak bisa mengatakan, apakah voting tergantung jenis kelamin dari data ini.

```
>tabletest(A)
```

0.990701632326

Berikut adalah tabel yang diharapkan, jika kita mengasumsikan frekuensi pemungutan suara yang diamati.

```
>writetable(expectedtable(A), wc=6, dc=1, labr=c("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

## Berapa Tes Lagi

---

Selanjutnya kami menggunakan analisis varians (uji-F) untuk menguji tiga sampel data terdistribusi normal untuk nilai rata-rata yang sama. Metode tersebut dinamakan ANOVA (analysis of variance). Di Euler, fungsi varanalysis() digunakan.

```
>x1=[109,111,98,119,91,118,109,99,115,109,94]; mean(x1),
```

106.545454545

```
>x2=[120,124,115,139,114,110,113,120,117]; mean(x2),
```

119.111111111

```
>x3=[120,112,115,110,105,134,105,130,121,111]; mean(x3)
```

116.3

```
>varanalysis(x1,x2,x3)
```

0.0138048221371

Artinya, kami menolak hipotesis dengan nilai mean yang sama. Kami melakukan ini dengan probabilitas kesalahan 1,3%.

Ada juga uji median, yaitu menolak sampel data dengan distribusi rata-rata yang berbeda menguji median dari sampel yang bersatu.

```
>a=[56, 66, 68, 49, 61, 53, 45, 58, 54];  
>b=[72, 81, 51, 73, 69, 78, 59, 67, 65, 71, 68, 71];  
>mediantest(a,b)
```

0.0241724220052

Tes lain tentang kesetaraan adalah ujian peringkat. Ini jauh lebih tajam daripada tes median.

```
>ranktest(a,b)
```

0.00199969612469

Dalam contoh berikut, kedua distribusi memiliki mean yang sama.

```
>ranktest(random(1,100),random(1,50)*3-1)
```

0.119780211001

Sekarang mari kita coba meniru 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, jika a lebih baik dari b.

```
>signtest(a,b)
```

0.0546875

Ini terlalu banyak kesalahan. Kita tidak dapat menolak bahwa a sama baiknya dengan b.

Tes Wilcoxon lebih tajam dari tes ini, tetapi bergantung pada nilai kuantitatif perbedaannya.

```
>wilcoxon(a,b)
```

0.0296680599405

Mari kita coba dua tes lagi menggunakan seri yang dihasilkan.

```
>wilcoxon(normal(1,20),normal(1,20)-1)
```

0.0309763942686

```
>wilcoxon(normal(1,20),normal(1,20))
```

0.588619799108

## Angka Acak

---

Berikut ini adalah tes untuk generator bilangan acak. Euler menggunakan generator yang sangat bagus, jadi kami tidak perlu mengharapkan adanya masalah.  
Pertama kami menghasilkan sepuluh juta bilangan acak di [0,1].

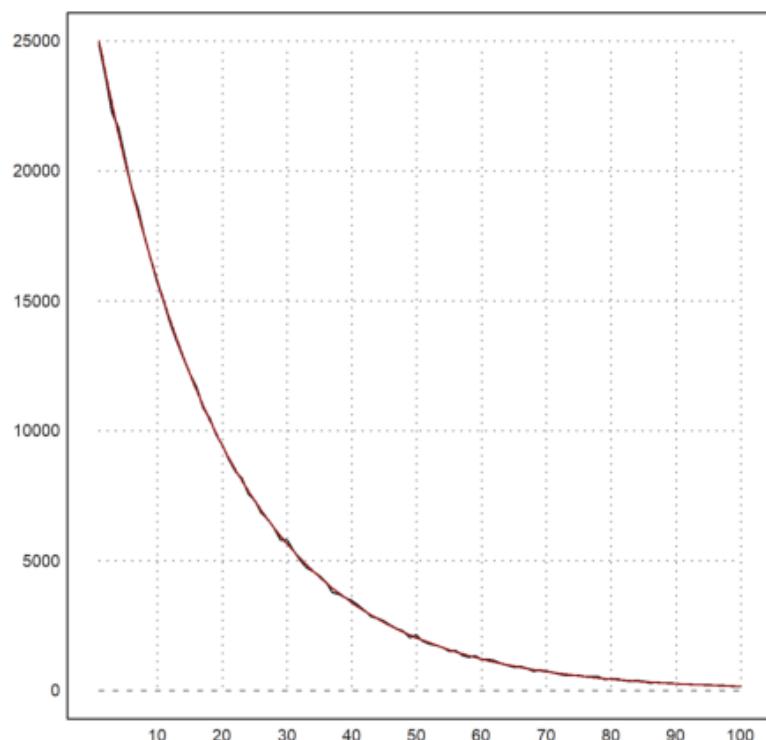
```
>n:=10000000; r:=random(1,n);
```

Selanjutnya kita menghitung jarak antara dua angka kurang dari 0,05.

```
>a:=0.05; d:=differences(nonzeros(r<a));
```

Akhirnya, kami memplot berapa kali, setiap jarak terjadi, dan membandingkan 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;
```

## Pengenalan untuk Pengguna Proyek R.

---

Jelas, EMT tidak bersaing dengan R sebagai paket statistik. Namun, ada banyak prosedur dan fungsi statistik yang tersedia di EMT juga. Jadi EMT dapat memenuhi kebutuhan dasar. Bagaimanapun, EMT hadir dengan paket numerik dan sistem aljabar komputer.

Notebook ini untuk Anda jika Anda sudah familiar dengan R, tetapi perlu mengetahui perbedaan sintaks EMT dan R. Kami mencoba memberikan gambaran umum tentang hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami mencari cara untuk bertukar data antara kedua sistem.

## Perhatikan bahwa ini adalah pekerjaan yang sedang berjalan. **Sintaks Dasar**

---

Hal pertama yang Anda pelajari di R adalah membuat vektor. Dalam EMT, perbedaan utamanya adalah: operator dapat mengambil ukuran langkah. Selain itu memiliki daya ikat yang rendah.

```
>n=10; 0:n/20:n-1
```

```
[0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5,  
7, 7.5, 8, 8.5, 9]
```

Fungsi c() tidak ada. Dimungkinkan untuk menggunakan vektor untuk menggabungkan berbagai hal.

Contoh berikut, seperti banyak contoh lainnya, dari "Introduction to R" yang disertakan dengan proyek R. Jika Anda membaca PDF ini, Anda akan menemukan bahwa saya mengikuti jalurnya dalam tutorial ini.

```
>x=[10.4, 5.6, 3.1, 6.4, 21.7]; [x,0,x]
```

```
[10.4, 5.6, 3.1, 6.4, 21.7, 0, 10.4, 5.6, 3.1, 6.4, 21.7]
```

Operator titik dua dengan ukuran langkah EMT diganti dengan fungsi seq() di R. Kita bisa 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]
```

Fungsi rep() dari R tidak ada di EMT. Untuk input vektor dapat dituliskan sebagai berikut.

```
>function rep(x:vector,n:index) := flatten(dup(x,n)); ...  
>rep(x,2)
```

```
[10.4, 5.6, 3.1, 6.4, 21.7, 10.4, 5.6, 3.1, 6.4, 21.7]
```

Perhatikan bahwa "=" atau ":=" digunakan untuk tugas. Operator "->" digunakan untuk unit di EMT.

```
>125km -> " miles"
```

```
77.6713990297 miles
```

Operator "<- untuk penugasan menyesatkan, dan bukan ide yang baik untuk R. Berikut ini akan membandingkan dan -4 di EMT.

```
>a=2; a<-4
```

0

Di R, "a <-4 <3" berfungsi, tetapi "a <-4 <-3" tidak. Saya juga memiliki ambiguitas yang serupa dalam EMT, tetapi mencoba menghilangkannya terus-menerus.

EMT dan R memiliki vektor tipe boolean. Tetapi di EMT, angka 0 dan 1 digunakan untuk mewakili salah dan benar. Di R, nilai benar dan salah dapat digunakan dalam aritmatika biasa seperti di EMT.

```
>x<5, %*x
```

```
[0, 0, 1, 0, 0]  
[0, 0, 3.1, 0, 0]
```

EMT melempar kesalahan atau menghasilkan NAN tergantung pada bendera "kesalahan".

```
>errors off; 0/0, isNaN(sqrt(-1)), errors on;
```

```
NAN  
1
```

String sama di R dan EMT. Keduanya ada di lokal saat ini, bukan di Unicode.

Di R ada paket untuk Unicode. Di EMT, string bisa berupa string Unicode. String unicode dapat diterjemahkan ke pengkodean lokal dan sebaliknya. Selain itu, u "..." dapat berisi entitas HTML.

```
>u"\u00f6; Ren\u00e4; Grothmann"
```

© René Grothmann

Berikut ini mungkin atau mungkin tidak ditampilkan dengan benar pada sistem Anda sebagai A dengan titik dan tanda hubung di atasnya. Itu tergantung pada font yang Anda gunakan.

```
>chartoutf([480])
```

Rangkaian string dilakukan dengan "+" atau "|". Ini dapat menyertakan angka, yang akan dicetak dalam format saat ini.

```
>"pi = "+pi
```

pi = 3.14159265359

## Pengindeksan

---

Biasanya, ini akan berfungsi seperti di R.

Tetapi EMT akan menafsirkan indeks negatif dari belakang vektor, sedangkan 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
```

Perilaku R dapat dicapai di EMT dengan drop().

```
>drop(x, 2)
```

```
[10.4, 3.1, 6.4, 21.7]
```

Vektor logika tidak diperlakukan secara berbeda sebagai indeks di EMT, berbeda dengan R. Anda perlu meng-ekstrak 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]
```

Tetapi nama untuk indeks tidak dimungkinkan di EMT. Untuk paket statistik, ini mungkin sering diperlukan untuk memudahkan akses ke elemen vektor.

Untuk meniru perilaku ini, kita dapat mendefinisikan fungsi sebagai berikut.

```
>function sel (v,i,s) := v[indexof(s,i)]; ...  
>s=["first","second","third","fourth"]; sel(x,[ "first","third"],s)
```

```
Trying to overwrite protected function sel!  
Error in:  
function sel (v,i,s) := v[indexof(s,i)]; ... ...  
^
```

```
Trying to overwrite protected function sel!  
Error in:  
function sel (v,i,s) := v[indexof(s,i)]; ... ...  
^  
[10.4, 3.1]
```

## Jenis Data

---

EMT memiliki lebih banyak tipe data tetap daripada R. Jelas, di R ada vektor yang tumbuh. Anda dapat menyetel vektor numerik kosong  $v$  dan menetapkan nilai ke elemen  $v[17]$ . Ini tidak mungkin dilakukan di EMT.

Berikut ini agak tidak efisien.

```
>v=[]; for i=1 to 10000; v=v|i; end;
```

EMT sekarang akan membangun vektor dengan  $v$  dan  $i$  ditambahkan pada stack dan menyalin vektor itu kembali ke variabel global  $v$ .

Semakin efisien mendefinisikan vektor sebelumnya.

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Untuk mengubah jenis data di EMT, Anda dapat menggunakan fungsi seperti `complex()`.

```
>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 penggabungan string sederhana. Tetapi ada fungsi seperti `print()` atau `frac()`.

Untuk vektor, Anda dapat dengan mudah menulis fungsi Anda sendiri.

```
>function tostr (v) ...
```

```
s="[";  
loop 1 to length(v);  
  s=s+print(v[#],2,0);  
  if #<length(v) then s=s+","; endif;  
end;  
return s+"]";  
endfunction
```

```
>tostr(linspace(0,1,10))
```

```
[0.00,0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1.00]
```

Untuk komunikasi dengan Maxima, terdapat fungsi `convertmxml()`, yang juga dapat digunakan untuk me-format vektor untuk keluaran.

```
>convertmxml(1:10)
```

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

Untuk Latex, perintah `tex` dapat digunakan untuk mendapatkan perintah Latex.

```
>tex(&[1,2,3])
```

```
\left[ 1 , 2 , 3 \right]
```

## Faktor dan Tabel

---

Dalam pengantar R ada contoh dengan apa 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"];
```

Asumsikan, kami 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, kami ingin menghitung rata-rata pendapatan di wilayah tersebut. Menjadi program statistik, R memiliki faktor() dan tapply() untuk ini.

EMT dapat melakukannya dengan mencari indeks teritori dalam daftar teritori yang unik.

```
>auterr=sort(unique(austates)); f=indexofsorted(auterr,austates)
```

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

Pada titik itu, kita bisa menulis fungsi loop 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
```

Ini sedikit tidak efisien, karena ini menghitung wilayah unik untuk setiap i, tetapi berfungsi.

```
>tappl(auterr,"mean",austates,incomes)
```

```
[44.5, 57.3333333333, 55.5, 53.6, 55, 60.5, 56, 52.25]
```

Perhatikan bahwa ini berfungsi untuk setiap vektor wilayah.

```
>tappl(["act","nsw"],"mean",austates,incomes)
```

```
[44.5, 57.3333333333]
```

Sekarang, paket statistik EMT mendefinisikan tabel seperti di R. Fungsi readtable() dan writetable() dapat digunakan untuk input dan output.

Jadi kita bisa mencetak rata-rata pendapatan negara di wilayah dengan cara yang 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

Kami juga dapat mencoba meniru perilaku R sepenuhnya.

Faktor-faktor tersebut harus disimpan dengan jelas dalam koleksi dengan tipe dan kategori (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);
```

Sekarang elemen ketiga dari koleksi akan berisi indeks.

```
>statef[3]
```

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

Sekarang kita 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
```

Kita tidak menambahkan banyak jenis pemeriksaan di sini. Tindakan pencegahan hanya menyangkut kategori (faktor) tanpa data. Tetapi seseorang harus memeriksa panjang yang benar dari t dan untuk kebenaran dari koleksi tf.

Tabel ini dapat dicetak sebagai tabel dengan writetable().

```
>writetable(tapply(incomes,statef,"mean"),wc=7)
```

act	nsw	nt	qld	sa	tas	vic	wa
44.5	57.33	55.5	53.6	55	60.5	56	52.25

## Array

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Akan mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau perpustakaan C untuk ini.

R memiliki lebih dari dua dimensi. Dalam R array adalah vektor dengan bidang dimensi.

Dalam EMT, vektor adalah matriks dengan satu baris. Itu bisa 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

Ekstraksi baris dan kolom, atau sub-matriks, sangat mirip dengan R.

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

2	3
7	8
12	13
17	18

Namun, di R dimungkinkan untuk mengatur daftar indeks tertentu dari vektor ke nilai. Hal yang sama mungkin terjadi di EMT hanya dengan satu loop.

```
>function setmatrixvalue (M, i, j, v) ...
```

```
loop 1 to max(length(i),length(j),length(v))  
  M[i#{},j#{}] = v#;  
end;  
endfunction
```

Kami mendemonstrasikan ini untuk menunjukkan bahwa matriks dilewatkan melalui referensi di EMT. Jika Anda tidak ingin mengubah matriks M asli, Anda perlu menyalinnya di fungsi.

```
>setmatrixvalue(X,1:3,3:-1:1,0); X,
```

1	2	0	4	5
6	0	8	9	10
0	12	13	14	15
16	17	18	19	20

Produk luar di EMT hanya dapat dilakukan di antara vektor. Ini otomatis karena bahasa matriks. Satu vektor harus menjadi vektor kolom dan vektor lainnya adalah vektor baris.

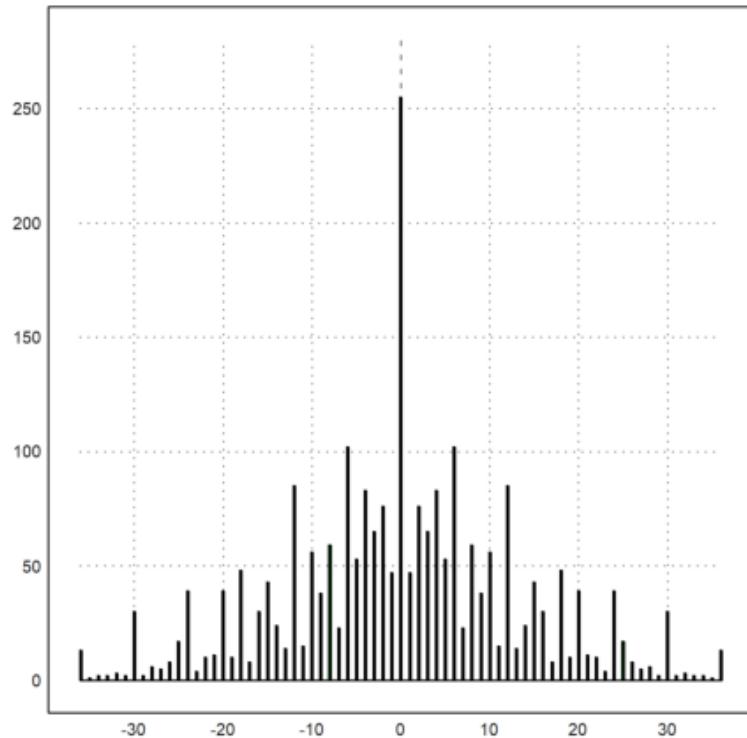
```
> (1:5) * (1:5) '
```

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

Dalam pengantar PDF untuk R ada contoh, yang menghitung distribusi ab-cd untuk a, b, c, d yang dipilih dari 0 hingga n secara acak. Solusi di R adalah membentuk matriks 4 dimensi dan menjalankan table() di atasnya. Tentu saja, ini bisa 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.

Tetapi kita ingin meniru perilaku R. Untuk ini, kita perlu meratakan perkalian ab dan membuat matriks ab-cd.

```
>a=0:6; b=a'; p=flatten(a*b); q=flatten(p-p'); ...
>u=sort(unique(q)); f=getmultiplicities(u,q); ...
>statplot(u,f,"h"):
```



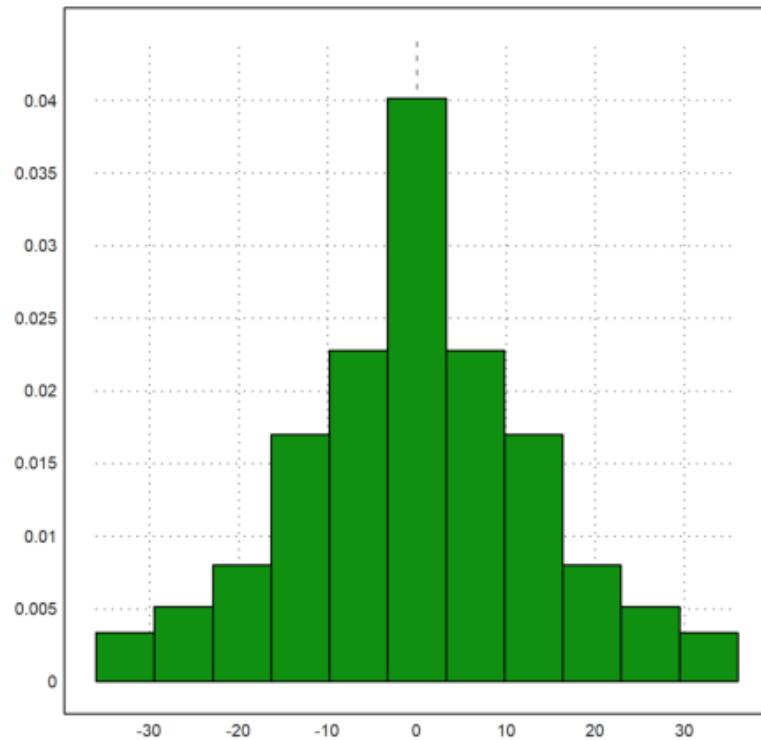
Selain perkalian yang tepat, EMT dapat menghitung frekuensi dalam vektor.

```
>getfrequencies(q,-50:10:50)
```

```
[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]
```

Cara paling mudah untuk memplotnya sebagai distribusi adalah sebagai berikut.

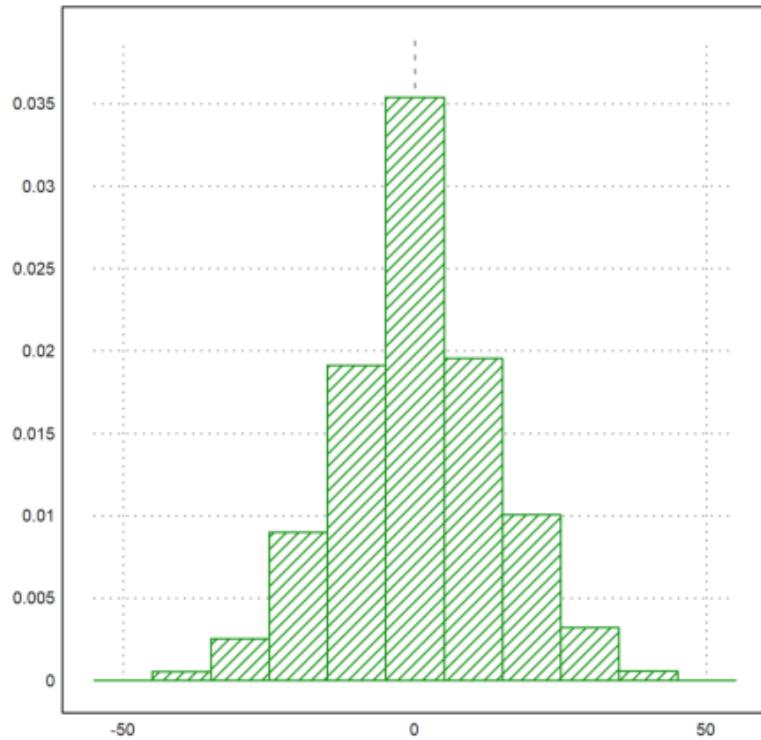
```
>plot2d(q,distribution=11):
```



Tetapi dimungkinkan juga untuk menghitung sebelumnya 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 macam daftar. Salah satunya adalah daftar global yang dapat berubah, dan yang lainnya adalah jenis daftar yang tidak dapat diubah. Kami tidak peduli dengan daftar global di sini.

Jenis daftar yang tidak dapat diubah disebut koleksi di EMT. Ini berperilaku seperti struktur di C, tetapi elemen hanya diberi nomor dan tidak dinamai.

```
>L={ {"Fred","Flintstone",40,[1990,1992]} }
```

```
Fred
Flintstone
40
[1990, 1992]
```

Saat ini elemen tidak memiliki nama, meskipun nama dapat diatur untuk tujuan khusus. Mereka diakses dengan angka.

```
> (L[4])[2]
```

```
1992
```

## Input dan Output File (Membaca dan Menulis Data)

---

Anda akan sering ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberi tahu Anda tentang banyak cara untuk mencapai ini. Fungsi sederhana adalah writematrix() dan readmatrix(). Mari kita tunjukkan bagaimana membaca dan menulis vektor real ke file.

```
>a=random(1,100); mean(a), dev(a),
```

```
0.49569  
0.2916
```

Untuk menulis data ke file, kami menggunakan fungsi writematrix().

Karena pendahuluan ini kemungkinan besar ada di direktori, di mana pengguna tidak memiliki akses tulis, kami menulis data ke direktori home pengguna. Untuk buku catatan sendiri, ini tidak perlu, karena file data akan ditulis ke 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, kami menggunakan readmatrix().

```
>a=readmatrix(filename)';
```

Dan hapus file tersebut.

```
>fileremove(filename);  
>mean(a), dev(a),
```

```
0.49569  
0.2916
```

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 oleh titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan koma). File berikut "test.csv" akan muncul di folder cuurent Anda.

```
>filename="test.csv"; ...  
>writematrix(random(5,3),file=filename,separator=",");
```

Anda sekarang dapat membuka file ini dengan Excel Indonesia secara langsung.

```
>fileremove(filename);
```

Terkadang kami 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 membuat token ini, kita mendefinisikan vektor token.

```
>tok := [ "f", "m" ]
```

f

m

Kemudian kita dapat menghitung berapa kali setiap token muncul dalam string, dan memasukkan hasilnya ke dalam 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();
```

File tersebut terlihat seperti ini.

```
>printfile(file)
```

A,B,C  
0.2325146620924334,0.5175800868283525,0.8399218481003107  
0.336655122980329,0.6942166504489329,0.7277428530427359  
0.06026396393889418,0.8443897421346642,0.7763524944847273

Fungsi readtable() dalam bentuknya yang paling sederhana bisa 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
0.23251	0.51758	0.83992
0.33666	0.69422	0.72774
0.06026	0.84439	0.77635

Matriks nilai adalah elemen pertama L. Perhatikan bahwa mean () dalam EMT menghitung nilai mean dari baris-baris matriks.

```
>mean(L[1])
```

```
0.53001  
0.5862  
0.56034
```

## File CSV

---

Pertama, mari kita tulis matriks ke dalam file. Untuk hasilnya, kami menghasilkan file di direktori kerja saat ini.

```
>file="test.csv"; ...  
>M=random(3,3); writematrix(M,file);
```

Berikut isi dari file ini.

```
>printfile(file)
```

```
0.3197946130216783,0.5787845944039014,0.2737923526542028  
0.3671130231603081,0.5275695458256693,0.6525304249790899  
0.7917330834536404,0.8603155045429328,0.527472095021572
```

CSV ini dapat dibuka pada sistem bahasa Inggris ke Excel dengan klik dua kali. Jika Anda mendapatkan file seperti itu di sistem Jerman, Anda perlu mengimpor data ke Excel dengan menggunakan titik desimal. Tetapi titik desimal adalah format default untuk EMT juga. Anda bisa membaca matriks dari file dengan readmatrix().

```
>readmatrix(file)
```

```
0.31979 0.57878 0.27379  
0.36711 0.52757 0.65253  
0.79173 0.86032 0.52747
```

Dimungkinkan 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 dipisahkan koma). Di Excel 2007, gunakan "simpan sebagai" dan "format lain", lalu pilih "CSV". Pastikan, tabel saat ini hanya berisi data yang ingin Anda ekspor.

Berikut ini contohnya.

```
>printfile("excel-data.csv")
```

```
0;1000;1000
1;1051,271096;1072,508181
2;1105,170918;1150,273799
3;1161,834243;1233,67806
4;1221,402758;1323,129812
5;1284,025417;1419,067549
6;1349,858808;1521,961556
7;1419,067549;1632,31622
8;1491,824698;1750,6725
9;1568,312185;1877,610579
10;1648,721271;2013,752707
```

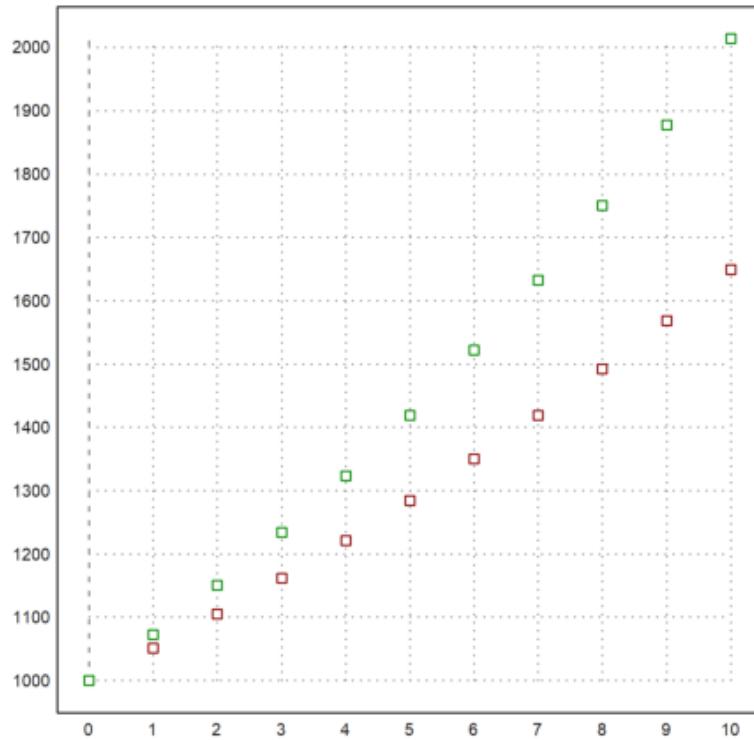
Seperti yang Anda lihat, sistem Jerman saya menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubahnya di pengaturan sistem atau di Excel, tetapi tidak perlu membaca matriks ke EMT. Cara termudah untuk membaca ini ke dalam Euler adalah readmatrix(). Semua koma diganti dengan titik dengan parameter >comma. Untuk CSV bahasa Inggris, cukup abaikan parameter ini.

```
>M=readmatrix("excel-data.csv",>comma)
```

0	1000	1000
1	1051.3	1072.5
2	1105.2	1150.3
3	1161.8	1233.7
4	1221.4	1323.1
5	1284	1419.1
6	1349.9	1522
7	1419.1	1632.3
8	1491.8	1750.7
9	1568.3	1877.6
10	1648.7	2013.8

Mari kita plot ini.

```
>plot2d(M' [1],M' [2:3],>points,color=[red,green]'):
```



Ada cara yang lebih mendasar untuk membaca data dari sebuah 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 itu juga bisa menggunakan koma desimal, jika Anda memanggil setdecimaleddot (",") sebelum Anda menggunakan fungsi ini.

Fungsi berikut adalah contoh untuk ini. Ini akan berhenti di akhir file atau baris kosong.

```
>function myload (file) ...
```

```
open(file);
M=[];
repeat
    until eof();
    v=getvectorline(3);
    if length(v)>0 then M=M_v; else break; endif;
end;
return M;
close(file);
endfunction
```

```
>myload(file)
```

```
0.31979  0.57878  0.27379
0.36711  0.52757  0.65253
0.79173  0.86032  0.52747
```

Juga dimungkinkan untuk membaca semua angka dalam file itu dengan getvector().

```
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
```

```
0.31979 0.57878 0.27379  
0.36711 0.52757 0.65253  
0.79173 0.86032 0.52747
```

Oleh karena itu, sangat mudah untuk menyimpan sebuah vektor nilai, satu nilai di setiap baris dan membaca kembali vektor ini.

```
>v=random(1000); mean(v)
```

```
0.49624
```

```
>writematrix(v',file); mean(readmatrix(file)')
```

```
0.49624
```

## Menggunakan Tabel

---

Tabel dapat digunakan untuk membaca atau menulis data numerik. Sebagai contoh, kami menulis tabel dengan judul 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  
0.59, 0.87, 0.14  
0.21, 0.22, 0.31  
0.79, 0.75, 0.79
```

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  
0.59 0.87 0.14  
0.21 0.22 0.31  
0.79 0.75 0.79
```

## Menganalisis Garis

---

Anda bahkan dapat mengevaluasi setiap baris dengan tangan. Misalkan, kita memiliki garis dengan format berikut.

```
>line="2020-11-03,Tue,1'114.05"
```

2020-11-03, Tue, 1'114.05

Pertama kita bisa membuat token baris.

```
>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])), ...
>strrepl(vt[3], "'", "")()
```

7.3816e+05

2

1114

Menggunakan ekspresi reguler, dimungkinkan untuk mengekstrak hampir semua informasi dari sebaris data. Asumsikan kita memiliki baris berikut dokumen HTML.

```
>line="<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>"
```

<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>

Untuk mengekstrak ini, kami menggunakan ekspresi reguler, yang mencari

- braket penutup>,
- string apapun yang tidak mengandung tanda kurung dengan

sub-kecocokan "(...)".

- kurung buka dan tutup menggunakan solusi terpendek,
- lagi string apapun yang tidak mengandung tanda kurung,
- dan kurung buka <.

Ekspresi reguler agak sulit dipelajari tetapi sangat kuat.

```
>{pos,s,vt}=strxfind(line,>([ ^<>]+)<.+?>([ ^<>]+)<" );
```

Hasilnya adalah posisi pertandingan, string yang cocok, dan vektor string untuk sub-pencocokan.

```
>for k=1:length(vt); vt[k](), end;
```

1145.5  
5.6

Ini adalah fungsi yang membaca semua item numerik antara <td> dan </td>.

```
>function readtd (line) ...
v=[]; cp=0;
repeat
{pos,s,vt}=strxfind(line,"<td.*?>(.+?)</td>",cp);
until pos==0;
if length(vt)>0 then v=v|vt[1]; endif;
cp=pos+strlen(s);
end;
return v;
endfunction
```

```
>readtd(line+"<td>non-numerical</td>")
```

```
1145.45
5.6
-4.5
non-numerical
```

## Membaca dari Web

---

Situs web atau file dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris.

Dalam contoh, kami membaca versi saat ini dari situs EMT. Kami menggunakan ekspresi reguler untuk memindai "Versi ..." di sebuah judul.

```
>function readversion () ...
urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html");
repeat
until urleof();
s=urlgetline();
k=strfind(s,"Version ",1);
if k>0 then substring(s,k,strfind(s,"<",k)-1), break; endif;
end;
urlclose();
endfunction
```

```
>readversion
```

```
Version 2022-05-18
```

## 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 menghasilkan file Euler di direktori kerja EMT.

```
>file="test.e"; ...
>writevar(random(2,2), "M", file); ...
>printfile(file, 3)
```

```
M = [ ..
0.5455905000563676, 0.0001549655632917177;
0.009845834149464919, 0.265649291025563];
```

Kami sekarang dapat memuat file. Ini akan mendefinisikan matriks M.

```
>load(file); show M,
```

```
M =
0.54559 0.00015497
0.0098458 0.26565
```

Ngomong-ngomong, jika writevar() digunakan pada variabel, itu akan mencetak definisi variabel dengan nama variabel ini.

```
>writevar(M); writevar(inch$)
```

```
M = [ ..
0.5455905000563676, 0.0001549655632917177;
0.009845834149464919, 0.265649291025563];
inch$ = 0.0254;
```

Kami juga dapat membuka file baru atau menambahkan ke file yang sudah ada. Dalam contoh kami 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 =
0.2939 0.34742
0.67823 0.87836
M2 =
0.49666
0.56157
0.10198
```

Untuk menghapus file apa pun, gunakan fileremove().

```
>fileremove(file);
```

Vektor baris dalam file tidak memerlukan koma, jika setiap nomor ada di 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 = [
0.900666916536
0.673346959629
0.916700922381
0.472855378963
0.346512533936
];

```

```

>load(file); M

```

```
[0.90067, 0.67335, 0.9167, 0.47286, 0.34651]
```

## Contoh Soal

---

1. Tabulasi data penelitian antara dua varibel biaya promosi (X) dan variabel penjualan rumah (Y)

```

>a=[5,17,26,34,49,52,61,77,81,86,99,104,127,149,157]

```

```
[5, 17, 26, 34, 49, 52, 61, 77, 81, 86, 99, 104, 127, 149,
157]
```

```

>b=[25,50,59,67,77,84,91,95,107,109,118,120,129,138,149]

```

```
[25, 50, 59, 67, 77, 84, 91, 95, 107, 109, 118, 120, 129,
138, 149]
```

```

>writetable(a' | b', labc=["a", "b"])

```

a	b
5	25
17	50
26	59
34	67
49	77
52	84
61	91
77	95
81	107
86	109
99	118
104	120
127	129
149	138
157	149

```
>p=polyfit(a,b,1)
```

```
[39.801, 0.73041]
```

Nilai konstanta ( $a$ )=39.801 menunjukkan besarnya variabel rata-rata penjualan rumah yang tidak dipengaruhi oleh biaya promosi atau dapat diartikan pada saat nilai biaya promosi sebesar 0, maka rata-rata penjualan rumah sebesar 39.801.

2. Dari hasil pengukuran diperoleh data tinggi badan kesepuluh siswa tersebut dalam ukuran sentimeter (cm) sebagai berikut.

```
>h=[168,169,188,158,165,156,177,160,170,164]
```

```
[168, 169, 188, 158, 165, 156, 177, 160, 170, 164]
```

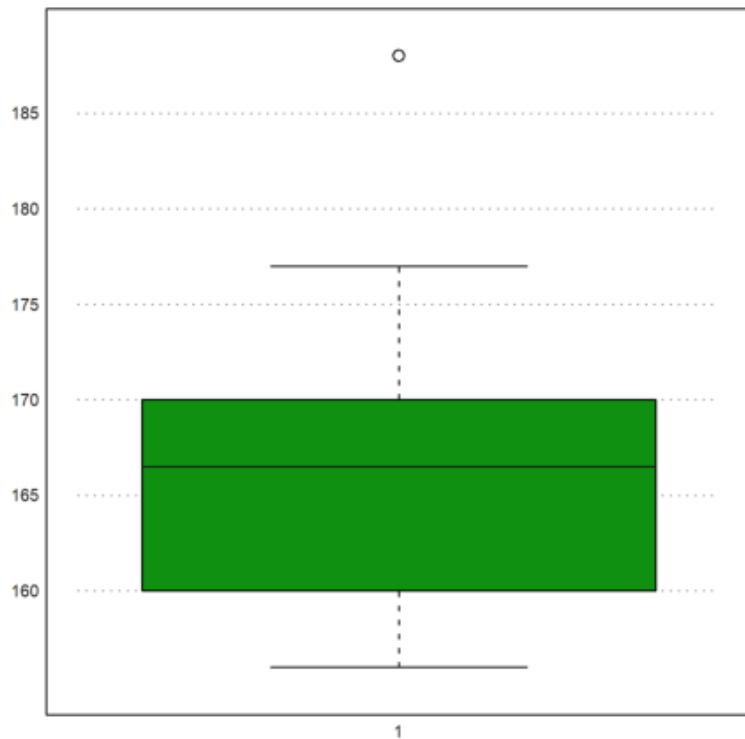
```
>mean(h)
```

```
167.5
```

```
>dev(h)
```

```
9.5248
```

```
>boxplot(h):
```



3. Suatu penelitian dilakukan untuk mengetahui apakah terdapat pengaruh perbedaan kartu kredit terhadap penggunaannya. Data di bawah ini adalah jumlah uang yang dibelanjakan ibu rumah tangga menggunakan kartu kredit (dalam \$). Empat jenis kartu kredit dibandingkan:

```
>Astra=[9,8,13,15,12]
```

```
[9, 8, 13, 15, 12]
```

```
>BCA=[10,13,17,14,11]
```

```
[10, 13, 17, 14, 11]
```

```
>CITI=[18,13,20,17,18]
```

```
[18, 13, 20, 17, 18]
```

```
>AMEX=[11,13,15,17]
```

```
[11, 13, 15, 17]
```

```
>varanalysis(Astra,BCA,CITI,AMEX)
```

```
0.026415
```

Dari hasil tersebut dapat disimpulkan bahwa tidak ada kesamaan rata rata dari data tersebut, atau Hipotesis kesamaan rata rata di tolak, dengan probabilitas kesalahan sebesar 0.026%