

# **MENGGUNAKAN LATEX DAN**

## **MARKDOWN**

Laporan Proposal ini Disusun Untuk Memenuhi Tugas Mata Kuliah Aplikasi Komputer

Dosen Pengampu: Bapak Drs. Sahid M.Sc



**Disusun oleh:**

Adiyatma  
23030630062  
Matematika B 2023

Email: [adiyatma.2023@student.uny.ac.id](mailto:adiyatma.2023@student.uny.ac.id)

**DEPARTEMEN PENDIDIKAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI YOGYAKARTA  
TAHUN 2024**

# Pendahuluan dan Pengenalan Cara Kerja EMT

---

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

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

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

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

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

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

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

## Komentar (Teks Uraian)

---

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

```
- * Judul
- ** Sub-Judul
- latex: F (x) = \int_a^x f (t) \, dt
- mathjax: \frac{x^2-1}{x-1} = x + 1
- maxima: 'integrate(x^3,x) = integrate(x^3,x) + C
```

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

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

## Judul

### Sub-Judul

---

$$x^2 - 1x - 1 = x + 1 \quad x^2 - 1x - 1 = x + 1$$

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

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

[Google](#)

image: hati.png

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

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

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

## Baris Perintah

---

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

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

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

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

```
>r=1.25; pi*r^2, 2*pi*r
4.90873852123
7.85398163397
```

## Latihan untuk Anda

---

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

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

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

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

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

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

Argumen atau input untuk fungsi ditulis di dalam tanda kurung.

```
>sin(45°), cos(pi), log(sqrt(E))
0.707106781187
-1
0.5
```

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

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

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

```
>(sqrt(5)+1)/2, %^2-%+1 // Memeriksa solusi x^2-x+1=0
1.61803398875
2
```

## Latihan untuk Anda

---

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

---

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

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

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

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

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

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

```
>remvalue a,b,c,D
>D
Variable D not found!
Error in:
D ...
^
```

## Menampilkan Panduan

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

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

```
>intrandom(10,6)
[4, 2, 6, 2, 4, 2, 3, 2, 2, 6]
```

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

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

```

>random(10)
[0.270906, 0.704419, 0.217693, 0.445363, 0.308411, 0.914541,
0.193585,
0.463387, 0.095153, 0.595017]
>normal(10)
[-0.495418, 1.6463, -0.390056, -1.98151, 3.44132, 0.308178, -
0.733427,
-0.526167, 1.10018, 0.108453]

```

## Matriks dan Vektor

---

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

Vektor dan matriks dapat diberi nama seperti variabel biasa.

```

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

```

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

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

```

>c=1:5
[1, 2, 3, 4, 5]
>w=0:0.1:1
[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
>mean(w^2)
0.35

```

## Bilangan Kompleks

---

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

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 biasa. Untuk mengubah bilangan riil menjadi kompleks, Anda dapat menambahkan 0i atau menggunakan fungsi "complex".

```

>complex(-1), sqrt(%)
-1+0i
0+1i

```

## Matematika Simbolik

---

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

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

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

```

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

>&expand((a+b)^2), &factor(x^2+5*x+6)
2          2
b      + 2 a b + a
(x + 2)(x + 3)

```

```
>&solve(a*x^2+b*x+c,x) // rumus abc
[ $x = \frac{(-\sqrt{b^2 - 4ac}) - b}{2a}, x = \frac{\sqrt{b^2 - 4ac} - b}{2a}]$ 

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

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


>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)

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


>::: factor(20!)

$$2^{18} \cdot 3^8 \cdot 5^4 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 \cdot 19$$

```

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

```
>::: binomial(5,2); // nilai C(5,2)
10

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

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


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

$$\frac{\cos(x)\cos(y) - \sin(x)\sin(y)}{\cos(x)\sin(y) + \sin(x)\cos(y)}$$


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

$$\frac{\sin(x)\cos(y) + \sin(y)\cos(x)}{\sin(x)\sin(y) + \cos(x)\cos(y)}$$


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

$$\frac{\sin(x) + \cos(x)^4}{\cos(x)^2}$$

```

$$\cos^3(x)$$

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

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

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

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

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

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

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

7

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

$$\frac{(b + a)^3 + 1}{b^3 + 3ab^2 + 3a^2b + a^3 + 1}$$

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

```
>&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.

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

## **Selamat Belajar dan Berlatih!**

---

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

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

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

# 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^(-4)*y^5*-7*x^2*y^(-9)
```

$$-\frac{42}{x^2 y^4}$$

Menjabarkan:

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

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

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

## Baris Perintah

---

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

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

```
>r:=3; h:=4; pi*r^2*h/3  
37.6991118431
```

Perintah harus dipisahkan dengan spasi. Baris perintah berikut mencetak dua hasilnya.

```
>pi*2*r*h, %+2*pi*r*h // Ingat tanda % menyatakan hasil perhitungan  
terakhir sebelumnya  
75.3982236862  
150.796447372
```

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

```
>x := 3;  
>x := cos(x) // nilai cosinus (x dalam radian)  
-0.9899924966  
>x := cos(x)  
0.548696133603
```

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

```
>x := 1.7; ...  
x := (x+2/x)/2, x := (x+2/x)/2, x := (x+2/x)/2,  
1.43823529412  
1.41441417058  
1.4142135766
```

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

Untuk melipat semua baris yang terdiri dari beberapa baris, tekan Ctrl+L. Kemudian baris-baris berikutnya hanya akan terlihat, jika salah satunya menjadi fokus. Untuk melipat satu baris yang terdiri dari beberapa baris, mulailah baris pertama dengan "%+".

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

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

81

Euler mendukung perulangan dalam baris perintah, asalkan dapat dimasukkan ke dalam satu baris atau beberapa baris. Dalam program, pembatasan ini tentu saja tidak berlaku. Untuk informasi lebih lanjut, lihat pengantar berikut.

```
>x=1; for i=1 to 5; x := (x+2/x)/2, end; // menghitung akar 2  
1.5  
1.41666666667  
1.41421568627  
1.41421356237  
1.41421356237
```

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

```
>x := 1.5; // comments go here before the ...
repeat xnew:=(x+2/x)/2; until xnew~:=x; ...
  x := xnew; ...
end; ...
x,
1.41421356237
```

Struktur kondisional juga berfungsi.

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

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

Saat Anda menggerakkan kursor di sepanjang baris, pasangan tanda kurung buka dan tutup akan disorot. Perhatikan juga baris status. Setelah tanda kurung buka fungsi `sqrt()`, baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol return.

```
>sqrt(sin(10°)/cos(20°))
0.429875017772
```

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

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

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

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

## Contoh Soal Baris Perintah

---

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

```
>a:= 2; (3*a^2) * (-7*a^4)
-1344
```

## Sintaksis Dasar

---

Euler mengetahui fungsi matematika yang umum. Seperti yang telah Anda lihat di atas, fungsi trigonometri bekerja dalam radian atau derajat. Untuk mengonversi ke derajat,

tambahkan simbol derajat (dengan tombol F7) ke nilai, atau gunakan fungsi rad(x). Fungsi akar kuadrat disebut sqrt di Euler. Tentu saja,  $x^{(1/2)}$  juga memungkinkan.

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

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma menghilangkan keluaran perintah. Di akhir baris perintah, "," diasumsikan, jika ";" tidak ada.

```
>g:=9.81; t:=2.5; 1/2*g*t^2  
30.65625
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

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

Anda harus menetapkan 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 perlu memasukkannya dalam bentuk baris.

```
>((1/7 + 1/8 + 2) / (1/3 + 1/2))^2 * pi  
23.2671801626
```

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

Hasil perhitungan ini adalah angka floating point. Secara default, angka ini dicetak dengan akurasi sekitar 12 digit. Pada baris perintah berikut, kita juga mempelajari cara 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 terdiri dari operator dan fungsi. Jika perlu, ekspresi harus berisi tanda kurung untuk memaksakan urutan

eksekusi yang benar. Jika ragu, sebaiknya gunakan tanda kurung. Perhatikan bahwa EMT menampilkan tanda kurung buka dan tutup saat mengedit baris perintah.

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

14.4978445072

Operator numerik Euler meliputi

- + unary atau operator plus
- unary atau operator minus
- \* , /
- . perkalian matriks
- $a^b$  pangkat untuk  $a$  positif atau integer  $b$  ( $a^{**}b$  juga berfungsi)
- $n!$  operator faktorial

dan masih banyak lagi.

Berikut ini beberapa fungsi yang mungkin Anda perlukan. Masih banyak lagi.

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

Beberapa perintah memiliki alias, misalnya ln untuk log.

```
>ln(E^2), arctan(tan(0.5))  
2  
0.5  
>sin(30°)  
0.5
```

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

$> 2^{\wedge}3^{\wedge}4, \quad (2^{\wedge}3)^{\wedge}4, \quad 2^{\wedge}(3^{\wedge}4)$   
2.41785163923e+24  
4096  
2.41785163923e+24

## Bilangan Riil

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

```
>longest 1/7
```

Representasi ganda internal membutuhkan 8 byte.

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

## String

---

String dalam Euler didefinisikan dengan "...".

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

String dapat dirangkai dengan | atau dengan +. Ini juga berlaku untuk angka, yang dalam kasus tersebut diubah menjadi string.

```
>"The area of the circle with radius " + 2 + " cm is " + pi*4 + " cm^2."  
The area of the circle with radius 2 cm is 12.5663706144 cm^2.
```

Fungsi cetak juga mengonversi angka menjadi string. Fungsi ini dapat mengambil sejumlah digit dan sejumlah tempat (0 untuk keluaran padat), dan optimalnya satu unit.

```
>"Golden Ratio : " + print((1+sqrt(5))/2,5,0)  
Golden Ratio : 1.61803
```

Ada string khusus `none`, yang tidak dicetak. String ini dikembalikan oleh beberapa fungsi, ketika hasilnya tidak penting. (Dikembalikan secara otomatis, jika fungsi tersebut tidak memiliki pernyataan `return`.)

```
>none
```

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

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

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

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

Vektor string kosong dilambangkan dengan `[none]`. Vektor string dapat dirangkai.

```
>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 membuat string seperti itu, gunakan `u"..."` dan salah satu entitas HTML.

String Unicode dapat dirangkai seperti string lainnya.

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

|

## Contoh Soal

---

Dalam komentar, entitas yang sama seperti alpha;, beta; dll. dapat digunakan. Ini mungkin merupakan alternatif cepat untuk Latex. (Rincian lebih lanjut pada komentar di bawah).

```
>a:=1; b:=2; c:=3; (a+b+c)^2  
36
```

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 α=β.
```

Dimungkinkan juga untuk menggunakan entitas numerik.

```
>u"\u00e4hnliches"  
Ähnliches
```

## Nilai Boolean

---

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

```
>2<1, "apel"<"banana"  
0  
1
```

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

```
>2<E && E<3  
1
```

Operator Boolean mematuhi aturan bahasa matriks.

```
>(1:10)>5, nonzeros(%)
[0, 0, 0, 0, 0, 1, 1, 1, 1, 1]
[6, 7, 8, 9, 10]
```

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

```
>N=2|3:2:99 // N berisi elemen 2 dan bilangan2 ganjil dari 3 s.d. 99
[2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29,
31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57,
59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85,
87, 89, 91, 93, 95, 97, 99]
>N[nonzeros(isprime(N))] //pilih anggota2 N yang prima
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,
53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

## Format Keluaran

---

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

```
>defformat; pi
3.14159265359
```

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

```
>longest pi
3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari angka ganda.

```
>printhex(pi)
3.243F6A8885A30*16^0
```

Format keluaran dapat diubah secara permanen dengan perintah format.

```
>format(12,5); 1/3, pi, sin(1)
0.33333
3.14159
0.84147
```

Format defaultnya adalah(12).

```
>format(12); 1/3
0.333333333333
```

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

```
>shortestformat; random(5,7)
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   0.53   0.5     0.17   0.26  
0.87   0.54   0.49   0.6    0.66   0.97   0.19
```

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

```
>setscalarformat(5); pi  
3.1416
```

Fungsi "longestformat" juga mengatur format skalar.

```
>longestformat; pi  
3.141592653589793
```

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

shortestformat shortformat longformat, longestformat

format(length,digits) goodformat(length)

fracformat(length)

defformat

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

Namun, format output EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,  
3.141592653589793  
>format(10,5); pi  
3.14159
```

Standarnya adalah defformat().

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

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

```
>longest pi^2/2  
4.934802200544679
```

Ada juga operator pendek untuk mencetak hasil dalam format pecahan. Kami telah menggunakan 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 terwakili secara tepat. Kesalahannya bertambah sedikit, seperti yang Anda lihat dalam perhitungan berikut.

```
>longest 0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1-1  
-1.110223024625157e-16
```

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

```
>0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1-1  
0
```

## Ekspresi

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

Variabel global dapat digunakan dalam evaluasi.

```
>r:=2; fx:="pi*r^2"; longest fx()  
12.56637061435917
```

Parameter ditetapkan ke x, y, dan z dalam urutan tersebut. Parameter tambahan dapat ditambahkan 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 suatu fungsi dengan nama yang sama. (Jika tidak, evaluasi ekspresi dalam fungsi dapat memberikan hasil yang sangat membingungkan bagi pengguna yang memanggil fungsi tersebut.)

```
>at:=4; function f(expr,x,at) := expr(x); ...  
f("at*x^2",3,5) // computes 4*3^2 not 5*3^2  
36
```

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

```
>at:=4; function f(expr,x,a) := expr(x,at=a); ...  
f("at*x^2",3,5)  
45
```

Sebagai referensi, kami mencatat bahwa koleksi panggilan (dibahas di tempat lain) dapat berisi ekspresi. Jadi, kita dapat 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.

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

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

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

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

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

Bentuk khusus dari suatu ekspresi memperbolehkan variabel apa pun sebagai parameter tanpa nama untuk evaluasi ekspresi, bukan hanya "x", "y", dst. Untuk ini, awali ekspresi dengan "@(variabel) ...".

```
>"@ (a,b) a^2+b^2", %(4,5)
@ (a,b) a^2+b^2
41
```

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

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

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

```
>fx &= x^3-a*x; ...
a=1.2; fx(0.5)
-0.475
```

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

```
>fx(0.5,a=1.1)
-0.425
```

Suatu ekspresi tidak harus simbolis. Hal ini diperlukan, jika ekspresi tersebut memuat fungsi, yang hanya diketahui dalam kernel numerik, bukan dalam Maxima.

## Matematika Simbolis

EMT mengerjakan matematika simbolis dengan bantuan Maxima. Untuk detailnya, mulailah dengan tutorial berikut, atau telusuri referensi untuk Maxima. Para ahli di Maxima harus memperhatikan bahwa terdapat perbedaan dalam sintaksis antara sintaksis asli Maxima dan sintaksis default ekspresi simbolis dalam EMT.

Matematika simbolis terintegrasi dengan mulus ke dalam Euler dengan &. Setiap ekspresi yang dimulai dengan & adalah ekspresi simbolis. Ekspresi tersebut dievaluasi dan dicetak oleh Maxima.

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

```
>$&44!
```

2658271574788448768043625811014615890319638528000000000

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

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

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

2481256778

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

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

2481256778

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

Anda akan mempelajari bahwa hal berikut juga berfungsi.

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

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

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

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

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

120

Dengan cara itu Anda dapat menggunakan solusi persamaan dalam persamaan lain.

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

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

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

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

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

Ekspresi simbolik diurai oleh Euler. Jika Anda memerlukan sintaksis yang kompleks dalam satu ekspresi, Anda dapat melampirkan ekspresi tersebut dalam "...". Menggunakan lebih dari satu ekspresi sederhana dimungkinkan, tetapi sangat tidak disarankan.

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

25

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

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

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

Sekali lagi, % merujuk pada hasil sebelumnya.

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

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

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

Input langsung perintah Maxima juga tersedia. Awali baris perintah dengan "::". Sintaks Maxima disesuaikan dengan sintaks EMT (disebut "mode kompatibilitas").

```
>&factor(30!)
```

265252859812191058636308480000000

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

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

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

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

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

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

```
          2  
          g
```

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

```
          3   x  
          x   E
```

$$x^3 e^x$$

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

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

```
          5  
          125 E
```

$$125 e^5$$

```
18551.64488782208
```

```
>fx(5)
```

```
18551.6448878
```

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

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

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

```
          10          5  
          1000 E      - 125 E
```

```
2.20079141499189e+7
```

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

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

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

```
>tex(fx)
x^3\,e^{x}
```

Ekspresi simbolik dapat dievaluasi seperti halnya ekspresi numerik.

```
>fx(0.5)
0.206090158838
```

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

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

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

Penugasan tersebut juga dapat bersifat simbolis.

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

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

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

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

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

Bandingkan dengan perintah numerik "`solve`" di Euler, yang memerlukan nilai awal, dan secara opsional nilai target.

```
>solve("x^2+x",1,y=4)
1.56155281281
```

Nilai numerik dari solusi simbolik dapat dihitung dengan mengevaluasi hasil simbolik. Euler akan membaca ulang penugasan `x= dst`. Jika Anda tidak memerlukan hasil numerik untuk perhitungan lebih lanjut, Anda juga dapat membiarkan Maxima menemukan nilai numeriknya.

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

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

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

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

Untuk mendapatkan solusi simbolis yang spesifik, seseorang dapat menggunakan "with" dan indeks.

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

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

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

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

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

Ekspresi simbolik dapat memiliki tanda, yang menunjukkan perlakuan khusus di Maxima. Beberapa tanda dapat digunakan sebagai perintah juga, yang lainnya tidak. Tanda ditambahkan dengan "|" (bentuk yang lebih baik dari "ev(...,flags)")

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

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

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

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

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

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

## Fungsi

---

Dalam EMT, fungsi adalah program yang didefinisikan dengan perintah "function". Fungsi ini dapat berupa fungsi satu baris atau fungsi multibaris.

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

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

Sebagai gambaran umum, kami tampilkan semua definisi yang mungkin untuk fungsi satu baris. Suatu fungsi dapat dievaluasi seperti fungsi Euler bawaan lainnya.

```
>f(2)
4.472135955
```

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

```
>f(0:0.1:1)
[0, 0.100499, 0.203961, 0.313209, 0.430813, 0.559017, 0.699714,
 0.854459, 1.0245, 1.21083, 1.41421]
```

Fungsi dapat diplot. Alih-alih ekspresi, kita hanya perlu memberikan nama fungsi.

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

```
>solve("f",1,y=1)
0.786151377757
```

Secara default, jika Anda perlu menimpa fungsi bawaan, Anda harus menambahkan kata kunci "overwrite". Menimpa fungsi bawaan berbahaya dan dapat menyebabkan masalah bagi fungsi lain yang bergantung padanya.

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
```

Sebaiknya kita hilangkan pendefinisian ulang dosa ini.

```
>forget sin; sin(pi/4)
0.707106781187
```

## Parameter Default

---

Fungsi numerik dapat memiliki parameter default.

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

Mengabaikan parameter ini akan menggunakan nilai default.

```
>f(4)  
16
```

Mengurnya akan menimpa nilai default.

```
>f(4,5)  
80
```

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

```
>f(4,a=1)  
16
```

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

```
>function f(x) := a*x^2  
>a=6; f(2)  
24
```

Namun, parameter yang ditetapkan akan mengantikan nilai global.

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

```
>f(2,a:=5)  
20
```

Fungsi simbolik didefinisikan dengan "&=". Fungsi ini didefinisikan dalam Euler dan Maxima, dan berfungsi 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
```

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

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

```
>g(5+g(1))  
178.635099908
```

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

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

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

```
>solve(&g(x),0.5)  
0.703467422498
```

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

```
>solve(&g,0.5)  
0.703467422498  
>function P(x,n) &= (2*x-1)^n; $&P(x,n)
```

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

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

$$(x + 2)^n$$

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

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

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

```
>P(3,4)  
625  
>$&P(x,4)+ Q(x,3), $&expand(%)
```

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

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

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

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

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

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

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

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

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

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

```
> $&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 bersifat simbolis, dan dapat digunakan dalam ekspresi simbolis lainnya.

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

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

Dengan := fungsinya bersifat numerik. Contoh yang bagus adalah integral tentu seperti

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

yang tidak dapat dievaluasi secara simbolis.

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

```
>function map f(x) := integrate("x^x",1,x)
>f(0:0.5:2)
[-0.783431, -0.410816, 0, 0.676863, 2.05045]
```

Fungsi dapat memiliki nilai default untuk parameter.

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

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

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

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

Sering kali, kita ingin menggunakan fungsi untuk vektor di satu tempat, dan untuk elemen individual di tempat lain. Hal ini dimungkinkan dengan parameter vektor.

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

$$\begin{aligned} b^2 - ab + b + a^2 \\ y^2 - xy + y + x^2 \end{aligned}$$

Fungsi simbolik semacam itu dapat digunakan untuk variabel simbolik.

Namun, fungsi tersebut juga dapat digunakan untuk vektor numerik.

```
>v=[3,4]; f(v)
17
```

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

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

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

$$\begin{aligned} y^4 - 6x^2y^2 + x^4 \\ 0 \end{aligned}$$

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

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

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

Singkatnya

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

## Menyelesaikan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolik.

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

```
>solve("x^2-2",1)
1.41421356237
```

Ini juga berlaku untuk ekspresi simbolik. Ambil fungsi berikut.

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

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

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

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

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

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

```
>$&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 cari titik, di mana polinomialnya adalah 2. Dalam `solve()`, nilai target default `y=0` dapat diubah dengan variabel yang ditetapkan.

Kita gunakan `y=2` dan periksa dengan mengevaluasi polinomial pada hasil sebelumnya.

```
>solve(px,1,y=2), px(%)
0.966715594851
2
```

Memecahkan ekspresi simbolik dalam bentuk simbolik akan menghasilkan 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 sebuah ekspresi.

```
>longest sol()
-0.6180339887498949 1.618033988749895
```

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

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

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

$$0$$

Memecahkan sistem persamaan secara simbolis dapat dilakukan dengan vektor persamaan dan penyelesaian simbolis `solve()`. Jawabannya adalah 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, sering kali 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 (cara lainnya adalah parameter titik koma).

```
>solve({{"f",3}},2,y=0.1)  
2.54116291558
```

Ini juga berlaku untuk ekspresi. Namun, elemen daftar bernama harus digunakan. (Informasi lebih lanjut tentang daftar ada di tutorial tentang sintaks EMT).

```
>solve({{"x^a-a^x",a=3}},2,y=0.1)  
2.54116291558
```

## Menyelesaikan Pertidaksamaan

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

```
>&load(fourier_elim)  
C:/Program Files/Euler x64/maxima/share/maxima/5.35.1/share/f\  
ourier_elim/fourier_elim.lisp
```

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

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

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

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

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

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

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

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

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

*emptyset*

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

*universalset*

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

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

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

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

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

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

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

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

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

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

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

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

```
> &fourier_elim([\max(x, y) > 6, x # 8, \abs(y-1) > 12], [x, y])  
[6 < x, x < 8, y < -11] or [8 < x, y < -11]  
or [x < 8, 13 < y] or [x = y, 13 < y] or [8 < x, x < y, 13 < y]  
or [y < x, 13 < y]
```

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

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

## Bahasa Matriks

---

Dokumentasi inti EMT berisi pembahasan terperinci tentang bahasa matriks Euler.

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

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

1	2
3	4

Produk matriks dilambangkan dengan sebuah titik.

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

```

        4
>b' // transpose b
[3, 4]
>inv(A) //inverse A
      -2          1
      1.5       -0.5
>A.b //perkalian matriks
      11
      25
>A.inv(A)
      1          0
      0          1

```

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

```

>A.A
      7          10
      15         22
>A^2 //perpangkatan elemen2 A
      1          4
      9         16
>A.A.A
      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 // hasilkali invers A dan b, A^(-1)b
      -2
      2.5
>inv(A).b
      -2
      2.5
>A\A //A^(-1)A
      1          0
      0          1
>inv(A).A
      1          0
      0          1
>A*A //perkalin elemen-elemen matriks seletak
      1          4
      9         16

```

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

```

>b^2 // perpangkatan elemen-elemen matriks/vektor
      9
      16

```

Jika salah satu operan merupakan vektor atau skalar, ia diekspansi dengan cara alami.

```
>2*A
```

2	4
6	8

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

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

1	4
3	8

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

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

2	6
6	12

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

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

1	2
1	2

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

1	4
3	8

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

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

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

Sekali lagi, ingatlah bahwa ini bukan 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, 0]
--------------------------------

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

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

5

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

Kita memperoleh vektor 0 dan 1, di mana 1 berarti benar.

```
>t=(1:10)^2; t==25 //menguji elemen2 t yang sama dengan 25 (hanya ada 1)
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
```

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

Dalam kasus ini, kita memperoleh indeks semua elemen yang lebih besar dari 50.

```
>nonzeros(t>50) //indeks elemen2 t yang lebih besar daripada 50
[8, 9, 10]
```

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

```
>t[nonzeros(t>50)] //elemen2 t yang lebih besar daripada 50
[64, 81, 100]
```

Sebagai contoh, mari kita cari semua kuadrat 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 komputasi integer. Ia menggunakan floating point presisi ganda secara internal. Namun, ia sering kali sangat berguna.

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

```
>t=1:1000; length(nonzeros(isprime(t^2+1)))
112
```

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

```
>seed(2); A=random(3,4)
0.765761      0.401188      0.406347      0.267829
0.13673       0.390567      0.495975      0.952814
0.548138      0.006085      0.444255      0.539246
```

Ia mengembalikan indeks elemen, yang bukan nol.

```
>k=mnonzeros(A<0.4) //indeks elemen2 A yang kurang dari 0,4
1                  4
2                  1
2                  2
3                  2
```

Indeks ini dapat digunakan untuk menetapkan elemen pada nilai tertentu.

```
>mset(A,k,0) //mengganti elemen2 suatu matriks pada indeks tertentu
0.765761      0.401188      0.406347      0
0                 0             0.495975      0.952814
```

```
0.548138          0          0.444255          0.539246
```

Fungsi mset() juga dapat mengatur elemen pada indeks ke entri 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 adalah mungkin untuk mendapatkan unsur-unsur dalam sebuah vektor.

```
>mget(A, k)  
[0.267829,  0.13673,  0.390567,  0.006085]
```

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

```
>ex=extrema(A)  
0.267829          4          0.765761          1  
0.13673           1          0.952814          4  
0.006085          2          0.548138          1
```

Kita dapat menggunakan ini untuk mengekstrak nilai maksimal pada 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 sebuah matriks, kita dapat menumpuk satu matriks di atas matriks lainnya. Jika keduanya tidak memiliki jumlah kolom yang sama, matriks yang lebih pendek akan diisi dengan 0.

```
>v=1:3; v_v  
1          2          3  
1          2          3
```

Dengan cara yang sama, kita dapat menempelkan suatu matriks ke sisi lain yang 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 tidak memiliki jumlah baris yang sama, matriks yang lebih pendek diisi dengan 0.

Ada pengecualian untuk aturan ini. Bilangan riil yang dilampirkan ke matriks akan digunakan sebagai kolom yang diisi dengan bilangan riil 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 dari 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
4
[2, 4]
4
```

Untuk vektor, ada `length()`.

```
>length(2:10)
9
```

Ada banyak fungsi lain yang menghasilkan matriks.

```
>ones(2,2)
1          1
1          1
```

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

```
>ones(5)*6
[6, 6, 6, 6, 6]
```

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

```
>random(2,2)
0.66566      0.831835
0.977        0.544258
```

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

```
>rbind(1:9,3,3) // menyusun elemen2 1, 2, 3, ..., 9 ke bentuk matriks 3x3
1              2              3
4              5              6
7              8              9
```

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

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

Mari kita menguji.

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

Fungsi multdup() menduplikasi elemen suatu vektor.

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

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

```
>flipx(1:5) //membalik elemen2 vektor baris
[5, 4, 3, 2, 1]
```

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

```
>rotleft(1:5) // memutar elemen2 vektor baris
[2, 3, 4, 5, 1]
```

Fungsi khusus adalah drop(v,i), yang menghapus elemen dengan indeks di i dari vektor v.

```
>drop(10:20,3)
[10, 11, 13, 14, 15, 16, 17, 18, 19, 20]
```

Perhatikan bahwa vektor i dalam drop(v,i) merujuk pada indeks elemen dalam v, bukan nilai elemen. Jika Anda ingin menghapus elemen, Anda perlu menemukan elemen terlebih dahulu. Fungsi indexof(v,x) dapat digunakan untuk menemukan elemen x dalam vektor v yang diurutkan.

```
>v=primes(50), i=indexof(v,10:20), drop(v,i)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
[0, 5, 0, 6, 0, 0, 0, 7, 0, 8, 0]
[2, 3, 5, 7, 23, 29, 31, 37, 41, 43, 47]
```

Seperti yang Anda lihat, tidak ada salahnya menyertakan indeks di luar 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 membuat 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 kita tidak mengubah matriks A. Kita mendapatkan matriks baru sebagai hasil dari setdiag().

Berikut ini adalah fungsi yang mengembalikan matriks tri-diagonal.

```
>function tridiag (n,a,b,c) := setdiag(setdiag(b*id(n),1,c),-1,a); ...  
tridiag(5,1,2,3)  
2 3 0 0 0  
1 2 3 0 0  
0 1 2 3 0  
0 0 1 2 3  
0 0 0 1 2
```

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

```
>A=redim(1:9,3,3)  
1 2 3  
4 5 6  
7 8 9
```

Sekarang kita dapat mengekstrak diagonalnya.

```
>d=getdiag(A,0)  
[1, 5, 9]
```

Misalnya, kita dapat membagi matriks berdasarkan diagonalnya. Bahasa matriks memastikan bahwa vektor kolom d diterapkan ke matriks baris demi baris.

```
>fraction A/d'  
1 2 3  
4/5 1 6/5  
7/9 8/9 1
```

## Vektorisasi

Hampir semua fungsi di Euler juga berfungsi untuk masukan matriks dan vektor, jika 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 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`, nilai fungsi vektor dapat dibuat dengan mudah.

Dalam contoh berikut, kita membuat vektor nilai `t[i]` dengan spasi 0,1 dari -1 hingga 1. Kemudian kita membuat vektor nilai fungsi `latex: 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 mengembangkan operator untuk skalar, vektor, dan matriks dengan cara yang jelas.

Misalnya, vektor kolom dikalikan vektor baris akan mengembang 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 perkalian matriks. Perkalian matriks dilambangkan dengan titik `"` dalam EMT.

```
>(1:5).(1:5)'
55
```

Secara default, vektor baris dicetak dalam format ringkas.

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

Untuk matriks, operator khusus `.` menunjukkan perkalian matriks, dan `A'` menunjukkan transposisi. Matriks  $1 \times 1$  dapat digunakan seperti bilangan riil.

```
>v:=[1,2]; v.v', %^2
5
25
```

Untuk mentranspos suatu matriks, kita menggunakan tanda apostrof.

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

```
1  
2  
3  
4
```

Jadi kita dapat menghitung matriks A dikali vektor b.

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

```
30  
70
```

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

```
>v'.v
```

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

$v \cdot v'$  menghitung norma v kuadrat untuk vektor baris v. Hasilnya adalah vektor  $1 \times 1$ , yang bekerja seperti bilangan riil.

```
>v.v'
```

```
30
```

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

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

```
30
```

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ini ringkasan aturannya.

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

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

Berikut ini adalah kasus sederhana dengan operator ^.

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

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

Berikut ini adalah kasus yang lebih rumit. Vektor baris dikalikan vektor kolom, keduanya diekspansi dengan cara menduplikasi.

```
>v:=[1,2,3]; v*v'
1 2 3
2 4 6
3 6 9
```

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

```
>v.v'
14
```

Ada banyak fungsi untuk matriks. Kami memberikan daftar singkatnya. Anda harus merujuk ke dokumentasi untuk informasi lebih lanjut tentang perintah-perintah ini.

sum,prod menghitung jumlah dan hasil perkalian baris-baris

cumsum,cumprod melakukan hal yang sama secara kumulatif

menghitung nilai ekstrem dari setiap baris

extrema mengembalikan vektor dengan informasi ekstrem

diag(A,i) mengembalikan diagonal ke-i

setdiag(A,i,v) menetapkan diagonal ke-i

id(n) matriks identitas

det(A) determinan

charpoly(A) polinomial karakteristik

eigenvalues(A) nilai eigen

```
>v*v, sum(v*v), cumsum(v*v)
[1, 4, 9]
14
[1, 5, 14]
```

The : operator generates an equally spaces row vector, optionally with a step size.

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

Elemen-elemen suatu 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 dapat berupa vektor baris indeks. : menunjukkan semua indeks.

```
>v[1:2], A[:,2]
[2, 4]
2
5
8
```

Bentuk singkat dari : adalah menghilangkan indeks sepenuhnya.

```
>A[,2:3]
2           3
5           6
8           9
```

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

```
>A{5}
5
```

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

```
>redim(A,1,prod(size(A))), flatten(A)
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Untuk menggunakan matriks pada tabel, mari kita atur ulang ke format default, dan hitung 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 tambahkan kolom ke matriks.

```
>M = deg(w)|w|cos(w)|sin(w)
0          0          1          0
45        0.785398    0.707107   0.707107
90        1.5708      0          1
135       2.35619     -0.707107  0.707107
180       3.14159     -1          0
225       3.92699     -0.707107 -0.707107
```

```

270      4.71239      0      -1
315      5.49779      0.707107   -0.707107
360      6.28319      1      0

```

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

Dalam contoh berikut, kita menghitung  $t[j]^i$  untuk  $i$  dari 1 hingga  $n$ . Kita memperoleh matriks, yang setiap barisnya merupakan tabel  $t^i$  untuk satu  $i$ . Yaitu, matriks tersebut memiliki elemen lateks:  $a_{\{i,j\}} = t_j^i$ ,  $\quad 1 \leq j \leq 101, \quad 1 \leq i \leq n$

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

Integrasi numerik integr() hanya berfungsi untuk batas interval skalar. Jadi, kita perlu memvektorkannya.

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

Kata kunci "map" akan memvektorkan fungsi tersebut. Fungsi tersebut sekarang akan berfungsi untuk vektor angka.

```
>f([1:5])
[0, 2.05045, 13.7251, 113.336, 1241.03]
```

## Sub-Matriks dan Elemen Matriks

---

Untuk mengakses elemen matriks, gunakan notasi tanda kurung.

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

Kita dapat mengakses baris matriks yang lengkap.

```
>A[3]
[7, 8, 9]
```

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

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

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

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

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

Di sini kita menginginkan baris pertama dan kedua dari  $A$ .

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

1	2	3
7	8	9

Kita bahkan dapat menyusun ulang A menggunakan vektor indeks. Untuk lebih tepatnya, kita tidak mengubah A di sini, tetapi menghitung versi A yang telah disusun ulang.

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

4	5	6
7	8	9
1	2	3

Trik indeks juga berfungsi dengan kolom.

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

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

2	3
5	6

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

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

2
5
8

Atau, biarkan indeks pertama kosong.

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

1	2	3
4	5	6
7	8	9

Kita juga bisa mendapatkan baris terakhir A.

```
>A[3]
```

[7, 8, 9]
-----------

Sekarang mari kita ubah elemen A dengan menetapkan submatriks A ke suatu nilai. Hal ini pada kenyataannya mengubah matriks A yang tersimpan.

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

1	2	3
4	5	9
7	8	9

Kita juga dapat menetapkan nilai ke baris A.

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

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

Kita bahkan dapat menetapkannya ke submatriks jika ukurannya tepat.

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

```

4           5           -1
6           7           9
7           8           9

```

Selain itu, beberapa jalan pintas diperbolehkan.

```
>A[1:2,1:2]=-1
-1           -1           -1
-1           -1           9
7            8            9
```

Peringatan: Indeks yang tidak sesuai batas akan mengembalikan matriks kosong, atau pesan kesalahan, tergantung pada pengaturan sistem. Pesan kesalahan adalah standar. Namun, perlu diingat bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks yang dihitung dari akhir.

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

## Sortir dan Acak

---

Fungsi `sort()` mengurutkan vektor baris.

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

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

Mari kita acak sebuah vektor.

```
>v=shuffle(1:8)
[4,  3,  5,  7,  1,  6,  8,  2]
```

Indeks berisi urutan `v` yang tepat.

```
>{vs,ind}=sort(v); v[ind]
[1,  2,  3,  4,  5,  6,  7,  8]
```

Ini juga berlaku untuk vektor string.

```
>s=["a","d","e","a","aa","g"]
a
d
e
a
aa
g
>{ss,ind}=sort(s); ss
a
a
aa
d
e
```

g

Seperti yang Anda lihat, posisi entri ganda agak acak.

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

Fungsi unik mengembalikan daftar yang diurutkan dari elemen unik suatu vektor.

```
>inrandom(1,10,10), unique(%)  
[8, 1, 4, 4, 9, 2, 6, 5, 10, 6]  
[1, 2, 4, 5, 6, 8, 9, 10]
```

Ini juga berlaku untuk vektor string.

```
>unique(s)  
a  
aa  
d  
e  
g
```

## Aljabar Linier

---

EMT memiliki banyak fungsi untuk memecahkan sistem linier, sistem renggang, 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=[5,6;7,8]; b=[4;3]; A\b  
-7  
6.5
```

Untuk contoh lain, kita buat matriks 200x200 dan jumlah barisnya. Kemudian kita selesaikan  $Ax=b$  menggunakan matriks invers. Kita ukur kesalahan sebagai deviasi maksimal semua elemen dari 1, tentu saja merupakan solusi yang benar.

```
>A=normal(100,100); b=sum(A); longest totalmax(abs(inv(A).b-1))  
8.992806499463768e-14
```

Jika sistem tidak mempunyai solusi, penyesuaian linier meminimalkan norma kesalahan  $Ax-b$ .

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

Determinan matriks ini adalah -1.

```
>det(A)  
-1
```

## Matriks Simbolik

---

Maxima memiliki matriks simbolik. Tentu saja, Maxima dapat digunakan untuk masalah aljabar linear sederhana tersebut. Kita dapat mendefinisikan matriks untuk Euler dan Maxima dengan &:=, lalu menggunakan dalam ekspresi simbolik. Bentuk [...] yang biasa digunakan untuk mendefinisikan matriks dapat digunakan dalam Euler untuk mendefinisikan matriks simbolik.

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

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

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

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

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

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

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

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

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

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

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

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

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

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 dibutuhkan pengindeksan yang cermat.

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

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

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

```
>A(a=6,b=8)
```

$$\begin{matrix} 1 & 6 \\ 8 & 2 \end{matrix}$$

Dalam ekspresi simbolik, gunakan dengan.

```
>${&A with [a=6,b=8]}
```

$$\begin{pmatrix} 1 & 6 \\ 8 & 2 \end{pmatrix}$$

Akses terhadap baris matriks simbolik bekerja seperti halnya matriks numerik.

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

$$[1, a]$$

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

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

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

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

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

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

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

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

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

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

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

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

Euler juga memiliki fungsi xinv() yang hebat, yang melakukan upaya lebih besar dan mendapatkan hasil yang lebih tepat.

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

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

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

Misalnya nilai eigen A dapat dihitung secara numerik.

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

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

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

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

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

## Nilai Numerik dalam Ekspresi Simbolik

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

```
>A &:= [1,pi;4,5]
      1      3.14159
      4      5
```

Masih terdapat perbedaan antara bentuk numerik dan bentuk simbolik. Saat mengubah matriks ke bentuk simbolik, pendekatan pecahan untuk bilangan riil akan digunakan.

```
>$&A
```

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

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

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

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

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

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

$$1.4142135623730950488016887242097_B \times 10^0$$

$$1.414213562373095$$

Ketepatan angka floating point besar dapat diubah.

```
>&fpprec:=100; &bfloat(pi)
      3.14159265358979323846264338327950288419716939937510582097494\
      4592307816406286208998628034825342117068b0
```

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

Perlu dicatat bahwa ini hanya diperlukan jika variabel telah didefinisikan dengan ":=" atau "=" sebagai variabel numerik.

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

$$-5.424777960769379$$

**Demo - Suku Bunga**

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

Asumsikan Anda memiliki modal awal sebesar 5000 (misalnya dalam dolar).

```
>K=5000  
5000
```

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

```
>K*1.03  
5150
```

Euler juga akan memahami sintaksis berikut.

```
>K+K*3%  
5150
```

Namun lebih mudah menggunakan faktor

```
>q=1+3%, K*q  
1.03  
5150
```

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

```
>K*q^10  
6719.58189672
```

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

```
>format(12,2); K*q^10  
6719.58
```

Mari kita cetak angka tersebut 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 hingga tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menulis loop, tetapi cukup masukkan

```
>K*q^(0:10)  
Real 1 x 11 matrix  
5000.00 5150.00 5304.50 5463.64 ...
```

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

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

Maka semua operator dan fungsi di Euler dapat diaplikasikan ke vektor elemen demi elemen. Jadi

```
>short q^(0:10)
[1, 1.03, 1.0609, 1.0927, 1.1255, 1.1593, 1.1941, 1.2299,
1.2668, 1.3048, 1.3439]
```

adalah vektor faktor  $q^0$  hingga  $q^{10}$ . Ini dikalikan dengan K, dan kita memperoleh vektor nilai.

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

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

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

Mari kita bandingkan kedua 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 mengulangnya selama bertahun-tahun. Euler menyediakan banyak solusi untuk ini.

Cara termudah adalah fungsi iterate, yang mengulang fungsi yang diberikan beberapa kali.

```
>VKr=iterate("oneyear",5000,10)
Real 1 x 11 matrix
5000.00      5150.00      5304.50      5463.64      ...

```

Kita 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 vektor tertentu, kita menggunakan indeks dalam tanda kurung siku.

```
>VKr[2], VKr[1:3]
5150.00
5000.00      5150.00      5304.50
```

Anehnya, kita juga dapat 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.

## Menyelesaikan Persamaan

Sekarang kita ambil fungsi yang lebih maju, yang menambahkan nilai 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 menentukan nilai-nilai ini. Kita pilih R=200.

```
>R=200; iterate("onepay",5000,10)  
Real 1 x 11 matrix  
  
5000.00      5350.00      5710.50      6081.82      ...
```

Bagaimana jika kita menghilangkan jumlah yang sama setiap tahun?

```
>R=-200; iterate("onepay",5000,10)  
Real 1 x 11 matrix  
  
5000.00      4950.00      4898.50      4845.45      ...
```

Kita melihat bahwa uang berkurang. Jelas, jika kita hanya memperoleh bunga sebesar 150 pada tahun pertama, tetapi mengurangi 200, kita akan kehilangan uang setiap tahun.

Bagaimana kita dapat menentukan berapa tahun uang tersebut akan bertahan? Kita harus menulis sebuah loop untuk ini. Cara termudah adalah dengan melakukan iterasi yang cukup lama.

```
>VKR=iterate("onepay",5000,50)  
Real 1 x 51 matrix  
  
5000.00      4950.00      4898.50      4845.45      ...
```

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

```
>min(nonzeros(VKR<0))  
48.00
```

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

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

Fungsi iterate() memiliki satu trik lagi. Fungsi ini dapat mengambil kondisi akhir sebagai argumen. Kemudian, fungsi ini 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. Berapa tingkat bunganya?

Ini adalah pertanyaan yang hanya dapat dijawab secara numerik. Di bawah ini, kita akan memperoleh rumus yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada rumus yang mudah untuk tingkat bunga. Namun untuk saat ini, kita bertujuan untuk mencari solusi numerik.

Langkah pertama adalah mendefinisikan fungsi yang melakukan iterasi sebanyak n kali. Kita 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

Namun, kita tidak lagi menggunakan nilai global R dalam ekspresi kita. Fungsi seperti iterate() memiliki trik khusus di Euler. Anda dapat meneruskan nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam kasus ini P dan R.

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

Mari kita coba uji coba.

```
>f(5000,-200,3,47)  
-19.83
```

Sekarang kita bisa memecahkan masalah kita.

```
>solve("f(5000,-200,x,50)",3)  
3.15
```

Rutin solve menyelesaikan ekspresi=0 untuk variabel x. Jawabannya adalah 3,15% per tahun. Kita ambil nilai awal 3% untuk algoritma tersebut. Fungsi solve() selalu membutuhkan nilai awal.

Kita dapat menggunakan fungsi yang sama untuk menyelesaikan pertanyaan berikut: Berapa banyak yang dapat kita hapus per tahun sehingga modal awal habis setelah 20 tahun dengan asumsi suku bunga 3% per tahun.

```
>solve("f(5000,x,3,20)",-200)  
-336.08
```

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

## Solusi Simbolis untuk Masalah Suku Bunga

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

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

$$R + q K$$

Sekarang kita dapat mengulanginya.

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

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

Kita melihat suatu pola. Setelah n periode kita memiliki

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

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

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

Ini agak rumit. Jumlahnya dievaluasi dengan tanda "simpsum" untuk mereduksinya menjadi hasil bagi.

Mari kita buat fungsi untuk ini.

```
>function fs(K, R, P, n) &= (1+P/100)^n*K + ((1+P/100)^n-1)/(P/100)*R;  
$&fs(K, 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 ini melakukan hal yang sama seperti fungsi f sebelumnya. Namun, fungsinya lebih efektif.

```
>longest f(5000,-200,3,47), longest fs(5000,-200,3,47)  
-19.82504734650985  
-19.82504734652684
```

Kita sekarang dapat menggunakannya untuk menanyakan waktu n. Kapan modal kita habis? Perkiraan awal kita adalah 30 tahun.

```
>solve("fs(5000,-330,3,x)",30)  
20.51
```

Jawaban ini menyatakan bahwa akan negatif setelah 21 tahun.

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

Asumsikan kita mendapatkan pinjaman sebesar  $K$ , dan membayar  $n$  kali cicilan sebesar  $R$  (dimulai setelah tahun pertama) sehingga menyisakan utang residual 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 lateks:  $i = \frac{P}{100}$

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

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

Kita dapat mencari laju  $R$  secara simbolis.

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

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

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

Tentu saja, kita memiliki limit 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 suku bunga sebesar 500.

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

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

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

```
>a =1  
1.00  
>$& a
```

*a*

# EMT\_Plot2D\_Adiyatma\_23030630062\_MatB

---

Adiyatma  
23030630062  
Matematika B

## Menggambar Grafik 2D dengan EMT

---

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

### Plot Dasar

---

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

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

```
>clg; // clear screen
>window(0,0,1024,1024); // use all of the window
>setplot(0,1,0,1); // set plot coordinates
>hold on; // start overwrite mode
>n=100; X=random(n,2); Y=random(n,2); // get random points
>colors=rgb(random(n),random(n),random(n)); // get random colors
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
>hold off; // end overwrite mode
>insimg; // insert to notebook
>reset;
```

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

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

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

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

```

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

```

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

## Aspek Plot

---

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

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

```

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

```

Fungsi `reset()` mengembalikan default plot termasuk rasio aspek.

## Plot 2D di Euler

---

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

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

Euler dapat membuat plot 2D

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

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

## Plot Ekspresi atau Variabel

---

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

Berikut adalah contoh paling dasar, yang menggunakan rentang default

dan menetapkan rentang  $y$  yang tepat agar sesuai dengan plot fungsinya.

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

```
> plot2d("x^2"):  
>aspect(1.5); plot2d("x^3-x"):  
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil  
plot setinggi 25 baris
```

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

Rentang plot diatur dengan parameter yang ditetapkan sebagai berikut

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

```
>plot2d("x^3-x",-1,2):  
>plot2d("sin(x)",-2*pi,2*pi); // plot sin(x) pada interval [-2pi, 2pi]  
>plot2d("cos(x)","sin(3*x)",xmin=0,xmax=2pi);
```

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

Dalam opsi, plot dapat diatur agar muncul

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

Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.

Bagaimanapun, tekan tombol tabulator untuk melihat plotnya, jika tersembunyi.

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

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

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

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

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

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

Pada contoh berikut, kita mengubah gaya kisi, menambahkan label, dan menggunakan label vertikal untuk sumbu y.

```
>aspect(1.5); plot2d("sin(x)",0,2pi,-1.2,1.2,grid=3,xl="x",yl="sin(x)":  
>plot2d("sin(x)+cos(2*x)",0,4pi):
```

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

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

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

```
>plot2d("sign(x)*exp(-x^2)",-1,1,<adaptive,n=10000):  
>plot2d("x^x",r=1.2,cx=1,cy=1):
```

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

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

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

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

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

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

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

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

```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):  
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression  
>plot2d(expr,-2,2): // plot from -2 to 2  
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)  
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot  
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle  
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square  
>plot2d("x^2",0,1,steps=1,color=red,n=10):  
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```

## Sisipan Soal Individu

```
>function h(x) := 3*x+6;  
>aspect(1); plot2d("h",-3,3,0,7):
```

Grafik tersebut menunjukkan grafik fungsi yang fungsinya disimpan terlebih dahulu pada variabel "h".

$h(x) = 3x+6$   
Memotong sumbu x ( $y = 0$ )

$$y=3x+6$$

$$0=3x+6$$

$$x=-6$$

$x=-2$

Memotong sumbu y ( $x=0$ )

$$y=3(0)+6 \quad y=3(0)+6$$

$$y=6$$

Rentang sumbu x pada grafik tersebut diubah menjadi  $(-3, 3)$  agar titik potong x pada  $(-2)$  dapat terlihat. Begitu pula rentang sumbu y yang diubah menjadi  $(0, 7)$  agar titik potong y di titik 6 dapat terlihat.

```
> plot2d("cos(3*x)");
```

## Fungsi dalam satu Parameter

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

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

```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a); // plot with a=0.3
>plot2d("f",0,1;0.4); // plot with a=0.4
>plot2d({{"f",0.2}},0,1); // plot with a=0.2
>plot2d({{"f(x,b)"},b=0.1}},0,1); // plot with 0.1
>function f(x) := x^3-x; ...
plot2d("f",r=1);
```

Berikut ini ringkasan fungsi yang diterima

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

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

```
>function f(x) &= diff(x^x,x)
           x
           x  (log(x) + 1)
```

```
>plot2d(f,0,2);
Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel sudah cukup untuk memplotnya.
```

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

```
>plot2d(expr,0,3pi);
>function f(x) &= x^x;
>plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
>plot2d(&diff(f(x),x),>add,color=red,style="-.-");
Untuk gaya garis ada berbagai pilihan.
```

- gaya="...". Pilih dari "-", "--", "-.", ".", ".-", "-.-".
- Warna: Lihat di bawah untuk warna.

- ketebalan: Defaultnya adalah 1.

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

- 0..15: indeks warna default.

- konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu muda, abu-abu, abu-abu tua, oranye, hijau muda, pirus, biru muda, oranye muda, kuning

- `rgb(merah,hijau,biru)`: parameteranya real di [0,1].

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

Berikut adalah tampilan warna EMT yang telah ditentukan sebelumnya.

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

Tapi anda bisa menggunakan warna lain.

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

## Sisipan soal Fungsi dalam 1 Parameter

```
>function f(x,a):= x^2+n*x+2;  
>n=3; plot2d("f",-6,3,-3,4;n):  
>plot2d("f",-4,2,-3,4;4):
```

Soal di atas adalah kondisi ketika mengerjakan grafik fungsi di 1 parameter dengan nilai koefisien yang berbeda namun dalam satu fungsi yang sama. Bentuk grafik tersebut serupa, hanya berbeda di letak titik potongnya saja.

Pertama, simpan fungsi dalam satu variabel (fungsi di atas disimpan dalam variabel f). Kemudian, pada baris perintah selanjutnya, masukkan

nilai koefisien yang didefinisikan dengan satu variabel yang berbeda dengan variabel fungsi (di atas menggunakan variabel n). Letakkan nilai koefisien di sebelum formula `plot2d("")` kemudian batasi dengan titik koma();. Disana saya juga mengubah rentang sumbu x dan y agar titik perpotongan dan bentuk parabolanya terlihat lebih jelas dan sesuai dengan perhitungan titik potong.

## 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 metodenya adalah menggunakan `>add` untuk beberapa panggilan ke `plot2d` secara keseluruhan, kecuali panggilan pertama. Kami telah menggunakan fitur ini pada contoh di atas.

```
>aspect(); plot2d("cos(x)",r=2,grid=6); plot2d("x",style=".",>add):  
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```

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

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

Kita tambahkan titik perpotongan dengan label (pada posisi "cl" untuk kiri tengah), dan masukkan hasilnya ke dalam buku catatan. Kami juga menambahkan judul pada plot.

```
>plot2d(["cos(x)","x"],r=1.1,cx=0.5,cy=0.5, ...  
color=[black,blue],style=["-","."], ...  
grid=1);  
>x0=solve("cos(x)-x",1); ...
```

```
plot2d(x0,x0,>points,>add,title="Intersection Demo"); ...
label("cos(x) = x",x0,x0,pos="cl",offset=20);
```

Dalam demo berikut, kita memplot fungsi  $\sin(x)/x$  dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung perluasan ini menggunakan Maxima melalui ekspresi simbolik.

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

Kami menambahkan kotak label yang menjelaskan fungsinya.

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

$$x^{120} - x^{26+1}x^{4120-x^{26+1}}$$

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

Dalam contoh berikut, kami menghasilkan Polinomial Bernstein.

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

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

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

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

```
>x=linspace(0,1,500);
>n=10; k=(0:n)'; // n is row vector, k is column vector
>y=bin(n,k)*x^k*(1-x)^(n-k); // y is a matrix then
>plot2d(x,y):
```

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

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

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

```
>plot2d(["sin(x)","cos(x)"],0,2pi,color=4:5):
>plot2d(["sin(x)","cos(x)"],0,2pi): // plot vector of expressions
Kita bisa mendapatkan vektor seperti itu dari Maxima menggunakan makelist() dan mxm2str().
```

```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
          10           9           8   2           7   3
          [(1 - x) , 10 (1 - x) x, 45 (1 - x) x , 120 (1 - x) x ,
           6   4           5   5           4   6           3   7
210 (1 - x) x , 252 (1 - x) x , 210 (1 - x) x , 120 (1 - x) x ,
           2   8           9   10
45 (1 - x) x , 10 (1 - x) x , x ]
```

```
>mxm2str(v) // get a vector of strings from the symbolic vector
```

$$\begin{aligned}
&(1-x)^{10} \\
&10*(1-x)^9*x \\
&45*(1-x)^8*x^2 \\
&120*(1-x)^7*x^3 \\
&210*(1-x)^6*x^4 \\
&252*(1-x)^5*x^5 \\
&210*(1-x)^4*x^6
\end{aligned}$$

```
120*(1-x)^3*x^7  
45*(1-x)^2*x^8  
10*(1-x)*x^9  
x^10
```

```
>plot2d(mxm2str(v),0,1); // plot functions
```

Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

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

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

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

Ekspresi dan fungsi satu baris dapat melihat variabel global.

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

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

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

Alternatifnya, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut kumpulan panggilan, dan ini adalah

cara yang lebih disukai untuk meneruskan argumen ke suatu fungsi yang kemudian diteruskan sebagai argumen ke fungsi lain.

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

```
>plot2d({{"f",1}},-10,10); ...
```

```
for a=2:10; plot2d({{"f",a}},>add); end;
```

Kita dapat mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris matriks f(x,a) merupakan satu fungsi.

Selain itu, kita dapat mengatur warna untuk setiap baris matriks.

Klik

dua kali pada fungsi getspectral() untuk penjelasannya.

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

## Sisipan Soal Menggambar Beberapa Kurva dg Koordinat Sama

```
>aspect(1); plot2d("sin(2*x)",-2,2,-2,4); plot2d("x+2",-2,2,-2,4,color=blue,>add);
```

Kita dapat menggambar lebih dari satu ekspresi matematika dalam satu bidang koordinat yang sama. Dengan rumus yang sama yaitu menggunakan plot2d("") lalu dibatasi dengan titik koma();ndi antara kedua ekspresi atau fungsi yang ingin digambar. Kemudian pada fungsi kedua, tambahkan color / style agar dapat membedakan kedua kurva yang dihasilkan.

## Label Teks

---

Dekorasi sederhana pun bisa

- judul dengan judul = "..."
- label x dan y dengan xl="...", yl="..."
- label teks lain dengan label("...",x,y)

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

```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x"):  
>expr := "log(x)/x"; ...  
    plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...  
    label("(1,0)",1,0); label("Max",E,expr(E),pos="lc");
```

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

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

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

Untuk plot titik, kotak label juga berfungsi. Tambahkan parameter >points, atau vektor bendera, satu untuk setiap label.

Pada contoh berikut, hanya ada satu fungsi. Jadi kita bisa menggunakan

string sebagai pengganti vektor string. Kami mengatur warna teks menjadi hitam untuk contoh ini.

```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...  
labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...  
tcolor=black,>left);
```

Gaya plot ini juga tersedia di statplot(). Seperti di plot2d() warna dapat diatur untuk setiap baris plot. Masih banyak lagi plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

```
>statplot(1:10,random(2,10),color=[red,blue]):  
Fitur serupa adalah fungsi textbox().
```

Lebarnya secara default adalah lebar maksimal baris teks. Tapi itu bisa diatur oleh pengguna juga.

```
>function f(x) &= exp(-x)*sin(2*pi*x); ...  
    plot2d("f(x)",0,2pi); ...  
    textbox(latex("\text{Example of a damped oscillation}\backslash f(x)=e^{-x}\sin(2\pi x)"),w=0.85);
```

Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

```
>plot2d("x^3-x",title=u"x &rarr; x^3 - x");  
Label pada sumbu x dan y bisa vertikal, begitu juga dengan sumbunya.  
>plot2d("sinc(x)",0,2pi,xl="x",yl=u"x &rarr; sinc(x)",>vertical);
```

## LaTeX

---

Anda juga dapat memplot rumus LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke biner "lateks" dan "dvipng" harus berada di jalur sistem, atau Anda harus mengatur LaTeX di menu opsi.

Perhatikan, penguraian LaTeX lambat. Jika Anda ingin menggunakan LaTeX

dalam plot animasi, Anda harus memanggil `latex()` sebelum loop satu kali dan menggunakan hasilnya (gambar dalam matriks RGB).

Pada plot berikut, kami menggunakan LaTeX untuk label x dan y, label, kotak label, dan judul plot.

```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
    title=latex("\text{Function } \Phi"), ...
    xl=latex("\phi"),yl=latex("\Phi(\phi)"); ...
    textbox( ...
        latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5);
...
label(latex("\Phi",color=blue),1,0.4);
```

Seringkali, kita menginginkan spasi dan label teks yang tidak konformal pada sumbu x. Kita bisa menggunakan `xaxis()` dan `yaxis()` seperti yang akan kita tunjukkan nanti.

Cara termudah adalah membuat plot kosong dengan bingkai menggunakan `grid=4`, lalu menambahkan grid dengan `ygrid()` dan `xgrid()`. Pada contoh berikut, kami menggunakan tiga string LaTeX untuk label pada sumbu x dengan `xtick()`.

```
>plot2d("sinc(x)",0,2pi,grid=4,<ticks); ...
ygrid(-2:0.5:2,grid=6); ...
xgrid([0:2]*pi,<ticks,grid=6); ...
xtick([0,pi,2pi],["0","\pi","2\pi"],>latex);
    Tentu saja fungsinya juga bisa digunakan.
```

```
>function map f(x) ...
if x>0 then return x^4
else return x^2
endif
endfunction
```

Parameter "peta" membantu menggunakan fungsi untuk vektor. Untuk plot, itu tidak perlu. Tapi untuk menunjukkan vektorisasi itu berguna, kita menambahkan beberapa poin penting ke plot di  $x=-1$ ,  $x=0$  dan  $x=1$ .

Pada plot berikut, kami juga memasukkan beberapa kode LaTeX. Kami menggunakananya untuk dua label dan kotak teks. Tentu saja, Anda hanya bisa menggunakananya LaTeX jika Anda telah menginstal LaTeX dengan benar.

```
>plot2d("f",-1,1,xl="x",yl="f(x)",grid=6); ...
plot2d([-1,0,1],f([-1,0,1]),>points,>add); ...
label(latex("x^3"),0.72,f(0.72)); ...
label(latex("x^2"),-0.52,f(-0.52),pos="ll"); ...
textbox( ...
    latex("f(x)=\begin{cases} x^3 & x>0 \\ x^2 & x \leq 0 \end{cases}"), ...
    x=0.7,y=0.2);
```

## Interaksi pengguna

---

Saat memplot suatu fungsi atau ekspresi, parameter `>pengguna` memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna bisa

- perbesar dengan + atau -
- pindahkan plot dengan tombol kursor
- pilih jendela plot dengan mouse
- atur ulang tampilan dengan spasi
- keluar dengan kembali

Tombol spasi akan mengatur ulang plot ke jendela plot aslinya.

Saat memplot data, flag >user hanya akan menunggu penekanan tombol.

```
>plot2d({{"x^3-a*x",a=1}},>user,title="Press any key!"):
>plot2d("exp(x)*sin(x)",user=true, ...
        title="+/- or cursor keys (return to exit)":
```

Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi bawaan mousedrag() menunggu aktivitas mouse atau keyboard. Ini melaporkan mouse ke bawah, gerakan mouse atau mouse ke atas, dan penekanan tombol. Fungsi dragpoints() memanfaatkan ini, dan memungkinkan pengguna menyeret titik mana pun dalam plot.

Kita membutuhkan fungsi plot terlebih dahulu. Misalnya, kita melakukan interpolasi pada 5 titik dengan polinomial. Fungsi tersebut harus diplot ke dalam area plot yang tetap.

```
>function plotf(xp,yp,select) ...
  d=interp(xp,yp);
  plot2d("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 nilainya secara global.

Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret titiknya.

```
>t=-1:0.5:1; dragpoints("plotf",t,random(size(t))-0.5):
```

Ada juga fungsi yang memplot fungsi lain bergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

Pertama kita membutuhkan fungsi plot.

```
>function plotf([a,b]) := plot2d("exp(a*x)*cos(2pi*b*x)",0,2pi;a,b);
Kemudian kita memerlukan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional garis judul.
```

Ada penggeser interaktif, yang dapat menetapkan nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

```
>dragvalues("plotf",["a","b"],[-1,2],[-2,2];[1,10]), ...
  heading="Drag these values:",hcolor=black):
```

Dimungkinkan untuk membatasi nilai yang diseret menjadi bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor berderajat n ke fungsi kosinus.

```
>function plotf(n) ...
  plot2d("cos(x)",0,2pi,>square,grid=6);
  plot2d("&taylor(cos(x),x,0,@n)",color=blue,>add);
  textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",>left);
endfunction
```

Sekarang kita izinkan derajat n bervariasi dari 0 hingga 20 dalam 20

```

perhentian. Hasil dragvalues() digunakan untuk memplot sketsa dengan
n
ini, dan untuk memasukkan plot ke dalam buku catatan.
>nd=dragvalues("plotf","degree",2,[0,20],20,y=0.8, ...
    heading="Drag the value:"); ...
plotf(nd):
    Berikut ini adalah demonstrasi sederhana dari fungsinya. Pengguna
    dapat menggambar jendela plot, meninggalkan jejak titik.
>function dragtest ...
    plot2d(None,r=1,title="Drag with the mouse, or press any key!");
    start=0;
    repeat
        {flag,m,time}=mousedrag();
        if flag==0 then return; endif;
        if flag==2 then
            hold on; mark(m[1],m[2]); hold off;
        endif;
    end
endfunction
>dragtest // lihat hasilnya dan cobalah lakukan!

```

## Gaya Plot 2D

---

Secara default, EMT menghitung tick sumbu otomatis dan menambahkan label ke setiap tick. Ini dapat diubah dengan parameter grid. Gaya default sumbu dan label dapat diubah. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk menyetel ulang ke gaya default, gunakan reset().

```

>aspect();
>figure(3,4); ...
    figure(1); plot2d("x^3-x",grid=0); ... // no grid, frame or axis
> figure(2); plot2d("x^3-x",grid=1); ... // x-y-axis
> figure(3); plot2d("x^3-x",grid=2); ... // default ticks
> figure(4); plot2d("x^3-x",grid=3); ... // x-y- axis with labels inside
> figure(5); plot2d("x^3-x",grid=4); ... // no ticks, only labels
> figure(6); plot2d("x^3-x",grid=5); ... // default, but no margin
> figure(7); plot2d("x^3-x",grid=6); ... // axes only
> figure(8); plot2d("x^3-x",grid=7); ... // axes only, ticks at axis
> figure(9); plot2d("x^3-x",grid=8); ... // axes only, finer ticks at axis
> figure(10); plot2d("x^3-x",grid=9); ... // default, small ticks inside
> figure(11); plot2d("x^3-x",grid=10); ...// no ticks, axes only
> figure(0):
    Parameter <frame mematikan frame, dan framecolor=blue mengatur frame
    menjadi warna biru.

```

Jika Anda menginginkan tanda centang Anda sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

```

>aspect(1.5);
>plot2d("x^3-x",grid=0); // plot
>frame; xgrid([-1,0,1]); ygrid(0): // add frame and grid
    Untuk judul plot dan label sumbu, lihat contoh berikut.
>plot2d("exp(x)",-1,1);
>textcolor(black); // set the text color to black
>title(latex("y=e^x")); // title above the plot
>xlabel(latex("x")); // "x" for x-axis
>ylabel(latex("y"),>vertical); // vertical "y" for y-axis
>label(latex("(0,1)'),0,1,color=blue): // label a point
    Sumbu dapat digambar secara terpisah dengan xaxis() dan yaxis().
>plot2d("x^3-x",<grid,<frame);

```

```

>xaxis(0,xx=-2:1,style="->"); yaxis(0,yy=-5:5,style="->"):
Teks pada plot dapat diatur dengan label(). Dalam contoh berikut,
"lc"
berarti bagian tengah bawah. Ini menetapkan posisi label relatif
terhadap koordinat plot.
>function f(x) &= x^3-x
            3
            x  - x

>plot2d(f,-1,1,>square);
>x0=fmin(f,0,1); // compute point of minimum
>label("Rel. Min.",x0,f(x0),pos="lc"); // add a label there
Ada juga kotak teks.
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]); // label box
>plot2d(["exp(x)","1+x"],color=[black,blue],style=["-","-.-"]):
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray); ...
plot2d("x^3-x",grid=1); ...
settitle("y=x^3-x",color=black); ...
label("x",2,0,pos="bc",color=gray); ...
label("y",0,6,pos="cl",color=gray); ...
reset():
Untuk kontrol lebih lanjut, sumbu x dan sumbu y dapat dilakukan
secara
manual.
```

Perintah fullwindow() memperluas jendela plot karena kita tidak lagi memerlukan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk menyetel ulang ke default.

```

>fullwindow; ...
gridstyle(color=darkgray,textcolor=darkgray); ...
plot2d(["2^x","1","2^(-x)"],a=-2,b=2,c=0,d=4,<grid,color=4:6,<frame); ...
xaxis(0,-2:1,style="->"); xaxis(0,2,"x",<axis); ...
yaxis(0,4,"y",style="->"); ...
yaxis(-2,1:4,>left); ...
yaxis(2,2^(-2:2),style=".",<left); ...
labelbox(["2^x","1","2^-x"],colors=4:6,x=0.8,y=0.2); ...
reset:
```

Berikut adalah contoh lain, di mana string Unicode digunakan dan sumbunya berada di luar area plot.

```

>aspect(1.5);
>plot2d(["sin(x)","cos(x)"],0,2pi,color=[red,green],<grid,<frame); ...
  xaxis(-1.1,(0:2)*pi,xt=["0",u"\u03c0",u"2\u03c0"],style="-",>ticks,>zero);
...
  xgrid((0:0.5:2)*pi,<ticks); ...
  yaxis(-0.1*pi,-1:0.2:1,style="-",>zero,>grid); ...
  labelbox(["sin","cos"],colors=[red,green],x=0.5,y=0.2,>left); ...
  xlabel(u"\u03c6"); ylabel(u"f(\u03c6)"):
```

## Merencanakan Data 2D

Jika x dan y adalah vektor data, maka data tersebut akan digunakan sebagai koordinat x dan y pada suatu kurva. Dalam hal ini, a, b, c, dan d, atau radius r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Alternatifnya, >persegi dapat diatur untuk mempertahankan rasio aspek persegi.

Merencanakan ekspresi hanyalah singkatan dari plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai x, dan satu atau

beberapa baris nilai y. Dari rentang dan nilai x, fungsi plot2d akan menghitung data yang akan diplot, secara default dengan evaluasi fungsi yang adaptif. Untuk plot titik gunakan ">titik", untuk garis dan titik campuran gunakan ">addpoints".

Tapi Anda bisa memasukkan data secara langsung.

- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y diplot baris demi baris.

Berikut adalah contoh dengan satu baris untuk x dan y.

```
>x=-10:0.1:10; y=exp(-x^2)*x; plot2d(x,y);
```

Data juga dapat diplot sebagai poin. Gunakan points=true untuk ini. Plotnya berfungsi seperti poligon, tetapi hanya menggambar sudutnya saja.

- style="...": Pilih dari "[]", "<>", "o", ".", "..", "+", "\*", "[ ]#", "< >#", "o#", "...#", "#", "|".

Untuk memplot kumpulan titik, gunakan >titik. Jika warna merupakan vektor warna, masing-masing titik mendapat warna berbeda. Untuk matriks koordinat dan vektor kolom, warna diterapkan pada baris matriks.

Parameter >addpoints menambahkan titik ke segmen garis untuk plot data.

```
>xdata=[1,1.5,2.5,3,4]; ydata=[3,3.1,2.8,2.9,2.7]; // data  
>plot2d(xdata,ydata,a=0.5,b=4.5,c=2.5,d=3.5,style="."); // lines  
>plot2d(xdata,ydata,>points,>add,style="o"); // add points  
>p=polyfit(xdata,ydata,1); // get regression line  
>plot2d("polyval(p,x)",>add,color=red); // add plot of line
```

## Menggambar Daerah Yang Dibatasi Kurva

Plot data sebenarnya berbentuk poligon. Kita juga dapat memplot kurva atau kurva terisi.

- filled=true mengisi plot.
- style="...": Pilih dari "#", "/", "\", "\\".
- fillcolor : Lihat di atas untuk mengetahui warna yang tersedia.

Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional, mencegah menggambar batas untuk semua gaya kecuali gaya default.

```
>t=linspace(0,2pi,1000); // parameter for curve  
>x=sin(t)*exp(t/pi); y=cos(t)*exp(t/pi); // x(t) and y(t)  
>figure(1,2); aspect(16/9)  
>figure(1); plot2d(x,y,r=10); // plot curve  
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // fill curve  
>figure(0):
```

Dalam contoh berikut kita memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian berbeda.

```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/");  
>t=linspace(0,2pi,6); ...  
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.2);  
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#" );
```

Contoh lainnya adalah septagon yang kita buat dengan 7 titik pada lingkaran satuan.

```
>t=linspace(0,2pi,7); ...
```

```

plot2d(cos(t),sin(t),r=1,>filled,style="/",fillcolor=red):
Berikut adalah himpunan nilai maksimal dari empat kondisi linier yang
kurang dari atau sama dengan 3. Ini adalah A[k].v<=3 untuk semua
baris
A. Untuk mendapatkan sudut yang bagus, kita menggunakan n yang
relatif
besar.

>A=[2,1;1,2;-1,0;0,-1];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=4,level=[0;3],color=green,n=111):
Poin utama dari bahasa matriks adalah memungkinkan pembuatan tabel
fungsi dengan mudah.

>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);
Kami sekarang memiliki nilai vektor x dan y. plot2d() dapat memplot
nilai-nilai ini
sebagai kurva yang menghubungkan titik-titik tersebut. Plotnya bisa
diisi. Pada kasus ini, ini memberikan hasil yang bagus karena aturan
belitan, yang digunakan untuk isi.

>plot2d(x,y,<grid,<frame,>filled):
Vektor interval diplot terhadap nilai x sebagai wilayah terisi
antara nilai interval yang lebih rendah dan lebih tinggi.

Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu
bisa
juga dapat digunakan untuk memplot kesalahan statistik.

>t=0:0.1:1; ...
plot2d(t,interval(t-random(size(t)),t+random(size(t))),style="|");
plot2d(t,t,add=true):
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval,
maka plot2d akan memplot rentang interval yang terisi pada bidang.
Gaya isiannya sama dengan gaya poligon.

>t=-1:0.01:1; x=~t-0.01,t+0.01~; y=x^3-x;
>plot2d(t,y):
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk
ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah
dan baris kedua berisi batas atas.

>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
Kita juga dapat mengisi rentang nilai seperti
-1≤(x2+y2)2-x2+y2≤0.-1≤(x2+y2)2-x2+y2≤0.

>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/"):
>plot2d("cos(x)","sin(x)^3",xmin=0,xmax=2pi,>filled,style="/"):
```

## Grafik Fungsi Parametrik

Nilai x tidak perlu diurutkan. (x,y) hanya menggambarkan sebuah kurva.

Jika x diurutkan, kurva tersebut merupakan grafik suatu fungsi.

Dalam contoh berikut, kita memplot spiral

$$\gamma(t)=t \cdot (\cos(2\pi t), \sin(2\pi t))$$

Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptif() untuk mengevaluasi ekspresi (lihat fungsi adaptif() untuk lebih jelasnya).

```

>t=linspace(0,1,1000); ...
plot2d(t*cos(2*pi*t),t*sin(2*pi*t),r=1):
Sebagai alternatif, dimungkinkan untuk menggunakan dua ekspresi untuk
kurva. Berikut ini plot kurva yang sama seperti di atas.
```

```
>plot2d("x*cos(2*pi*x)","x*sin(2*pi*x)",xmin=0,xmax=1,r=1):
>t=linspace(0,1,1000); r=exp(-t); x=r*cos(2pi*t); y=r*sin(2pi*t);
>plot2d(x,y,r=1):
```

Pada contoh berikutnya, kita memplot kurvanya

$$\gamma(t) = (r(t)\cos(t), r(t)\sin(t)) \quad \gamma(t) = (r(t)\cos(\frac{2\pi}{r}t), r(t)\sin(\frac{2\pi}{r}t))$$

dengan

$$r(t) = 1 + \sin(3t)$$

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5):
```

## Menggambar Grafik Bilangan Kompleks

Serangkaian bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan dihubungkan. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi  $1 \times 2$ ) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.

Matriks bilangan kompleks secara otomatis akan diplot sebagai kisi-kisi pada bidang kompleks.

Pada contoh berikut, kita memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
>aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=r*exp(I*a);
>plot2d(exp(z),cgrid=[40,10]):
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```

Vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian nyata dan bagian imajiner.

Dalam contoh, kita memplot lingkaran satuan dengan

$$\gamma(t) = e^{it}$$

```
>t=linspace(0,2pi,1000); ...
plot2d(exp(I*t)+exp(4*I*t),r=2):
```

## Plot Statistik

Ada banyak fungsi yang diperuntukkan pada plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

Jumlah kumulatif dari nilai terdistribusi normal 0-1 menghasilkan jalan acak.

```
>plot2d(cumsum(randnormal(1,1000))):
Penggunaan dua baris menunjukkan jalan dalam dua dimensi.
```

```
>X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
>columnspplot(cumsum(random(10)),style="/",color=blue):
```

Itu juga dapat menampilkan string sebagai label.

```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
"Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnspplot(values,lab=months,color=red,style="-");
>title("Temperature"):
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```

```

>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
>plot2d(normal(1,1000),>distribution,style="O"):
>plot2d("qnormal",0,5;2.5,0.5,>filled):
    Untuk memplot distribusi statistik eksperimental, Anda dapat
    menggunakan distribution=n dengan plot2d.

>w=randexponential(1,1000); // exponential distribution
>plot2d(w,>distribution): // or distribution=n with n intervals
    Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya
    dengan >bar di plot3d, atau dengan plot kolom.

>w=normal(1000); // 0-1-normal distribution
>{x,y}=histo(w,10,v=[-6,-4,-2,-1,0,1,2,4,6]); // interval bounds v
>plot2d(x,y,>bar):
    Fungsi statplot() mengatur gaya dengan string sederhana.

>statplot(1:10,cumsum(random(10)), "b"):
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
    Selain itu, data dapat diplot sebagai batang. Dalam hal ini, x harus
    diurutkan dan satu elemen lebih panjang dari y. Batangnya akan
    memanjang dari x[i] hingga x[i+1] dengan nilai y[i]. Jika x berukuran
    sama dengan y, maka x akan diperpanjang satu elemen dengan spasi
    terakhir.

    Gaya isian dapat digunakan seperti di atas.

>n=10; k=bin(n,0:n); ...
>plot2d(-0.5:n+0.5,k,bar=true,fillcolor=lightgray):
    Data untuk plot batang (batang=1) dan histogram (histogram=1) dapat
    diberikan secara eksplisit dalam xv dan yv, atau dapat dihitung dari
    distribusi empiris dalam xv dengan >distribusi (atau distribusi=n).
    Histogram nilai xv akan dihitung secara otomatis dengan >histogram.
    Jika >even ditentukan, nilai xv akan dihitung dalam interval bilangan
    bulat.

>plot2d(normal(10000),distribution=50):
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
>columnsplot(m,k):
>plot2d(random(600)*6,histogram=6):
    Untuk distribusi, terdapat parameter distribution=n, yang menghitung
    nilai secara otomatis dan mencetak distribusi relatif dengan n
    sub-interval.

>plot2d(normal(1,1000),distribution=10,style="\\"/"):
    Dengan parameter even=true, ini akan menggunakan interval bilangan
    bulat.

>plot2d(intrandom(1,1000,10),distribution=10,even=true):
    Perhatikan bahwa ada banyak plot statistik yang mungkin berguna.
    Silahkan lihat tutorial tentang statistik.

>columnsplot(getmultiplicities(1:6,intrandom(1,6000,6))):
>plot2d(normal(1,1000),>distribution); ...
>plot2d("qnormal(x)",color=red,thickness=2,>add):
    Ada juga banyak plot khusus untuk statistik. Plot kotak menunjukkan
    kuartil distribusi ini dan banyak outlier. Menurut definisinya,
    outlier dalam plot kotak adalah data yang melebihi 1,5 kali rentang
    50% tengah plot.

>M=normal(5,1000); boxplot(quartiles(M)):

```

## Sisipan Soal Plot Statistik

Gambarkan diagram batang yang berisi data-data berikut.  
Lian menjual keripik singkong kepada teman kelasnya selama 5 hari.  
Pda

```

hari Senin terjual 5 bungkus, Selasa terjual 3 bungkus, Rabu dan
Kamis
terjual 8 bungkus, dan Jumat terjual 10 bungkus.
>day=["Senin","Selasa","Rabu","Kamis","Jumat"];
>values=[5,3,8,8,10];
>columnsplot(values,lab=day,color=red,style="-");
>title("Penjualan Keripik"):

Untuk membuat diagram pada EMT dapat menggunakan columnsplot()

```

## Fungsi Implisit

Plot implisit menunjukkan penyelesaian garis level  $f(x,y)=\text{level}$ , dengan "level" dapat berupa nilai tunggal atau vektor nilai. Jika `level = "auto"`, akan ada garis level nc, yang akan tersebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan `>hue` untuk menunjukkan nilai fungsi. Untuk fungsi implisit, xv harus berupa fungsi atau ekspresi parameter x dan y, atau alternatifnya, xv dapat berupa matriks nilai.

Euler dapat menandai garis level

$$f(x,y)=c \quad f(x,y)=c$$

dari fungsi apa pun.

Untuk menggambar himpunan  $f(x,y)=c$  untuk satu atau lebih konstanta c, Anda dapat menggunakan `plot2d()` dengan plot implisitnya pada bidang. Parameter c adalah `level=c`, dimana c dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi setiap titik dalam plot. Parameter "n" menentukan kehalusan plot.

```

>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
>plot2d(expr,level=0:0.5:20,>hue,contourcolor=white,n=200): // nice
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
    Ini juga berfungsi untuk plot data. Namun Anda harus menentukan
    rentangnya untuk label sumbu.
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
>plot2d("x^3-y^2",>contour,>hue,>spectral):
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
>plot2d("x^2+y^3+x*y",level=1,r=4,n=100):
>plot2d("x^2+2*y^2-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):
    Dimungkinkan juga untuk mengisi set
        a≤f(x,y)≤b a≤f(x,y)≤b

```

dengan rentang level.

Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

```

>plot2d(expr,level=[0:1],style="-",color=blue): // 0 <= f(x,y) <= 1
    Plot implisit juga dapat menunjukkan rentang level. Maka level harus
    berupa matriks interval level 2xn, di mana baris pertama berisi awal
    dan baris kedua berisi akhir setiap interval. Alternatifnya, vektor

```

baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level ke interval.

```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
>plot2d("x^2+y^3+x*y",level=-10:20,r=2,style="-",dl=0.1,n=100):
>plot2d("sin(x)*cos(y)",r=pi,>hue,>levels,n=100):
```

Dimungkinkan juga untuk menandai suatu wilayah

$$a \leq f(x,y) \leq b \quad a \leq f(x,y) \leq b.$$

Hal ini dilakukan dengan menambahkan level dengan dua baris.

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
style="#",color=red,<outline, ...
level=[-2;0],n=100):
```

Dimungkinkan untuk menentukan level tertentu. Misalnya, kita dapat memplot solusi persamaan seperti

$$x_3 - xy + x_2 y_2 = 6x_3 - xy + x_2 y_2 = 6$$

```
>plot2d("x^3-x*y+x^2*y^2",r=6,level=1,n=100):
>function starplot1 (v, style="/", color=green, lab=none) ...
  if !holding() then clg; endif;
  w=window(); window(0,0,1024,1024);
  h=holding(1);
  r=max(abs(v))*1.2;
  setplot(-r,r,-r,r);
  n=cols(v); t=linspace(0,2pi,n);
  v=v|v[1]; c=v*cos(t); s=v*sin(t);
  cl=barcolor(color); st=barstyle(style);
  loop 1 to n
    polygon([0,c[#],c[#+1]],[0,s[#],s[#+1]],1);
    if lab!=none then
      rlab=v[#]+r*0.1;
      {col,row}=toscreen(cos(t[#])*rlab,sin(t[#])*rlab);
      ctext(""+lab#[#],col,row-textheight()/2);
    endif;
  end;
  barcolor(cl); barstyle(st);
  holding(h);
  window(w);
endfunction
```

Tidak ada tanda centang kotak atau sumbu di sini. Selain itu, kami menggunakan jendela penuh untuk plotnya.

Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Ini tidak perlu dilakukan jika Anda yakin plot Anda berhasil.

```
>reset; starplot1(normal(1,10)+5,color=red,lab=1:10):
```

Terkadang, Anda mungkin ingin merencanakan sesuatu yang plot2d tidak bisa lakukan, tapi hampir.

Dalam fungsi berikut, kita membuat plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

```
>function logimpulseplot1 (x,y) ...
  {x0,y0}=makeimpulse(x,log(y)/log(10));
  plot2d(x0,y0,>bar,grid=0);
  h=holding(1);
  frame();
  xgrid(ticks(x));
  p=plot();
  for i=-10 to 10;
    if i<=p[4] and i>=p[3] then
```

```

        ygrid(i,yt="10^"+i);
    endif;
end;
holding(h);
endfunction
Mari kita uji dengan nilai yang terdistribusi secara eksponensial.
>aspect(1.5); x=1:10; y=-log(random(size(x)))*200; ...
logimpulseplot1(x,y):
Mari kita menganimasikan kurva 2D menggunakan plot langsung. Perintah
plot(x,y) hanya memplot kurva ke dalam jendela plot. setplot(a,b,c,d)
menyetel jendela ini.

Fungsi wait(0) memaksa plot muncul di jendela grafis. Jika tidak,
pengundian ulang akan dilakukan dalam interval waktu yang jarang.
>function animliss (n,m) ...
t=linspace(0,2pi,500);
f=0;
c=framecolor(0);
l=linewidth(2);
setplot(-1,1,-1,1);
repeat
  clg;
  plot(sin(n*t),cos(m*t+f));
  wait(0);
  if testkey() then break; endif;
  f=f+0.02;
end;
framecolor(c);
linewidth(1);
endfunction
Tekan tombol apa saja untuk menghentikan animasi ini.
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER

```

## Plot Logaritmik

EMT menggunakan parameter "logplot" untuk skala logaritmik.  
 Plot logaritma dapat diplot menggunakan skala logaritma di y dengan logplot=1, atau menggunakan skala logaritma di x dan y dengan logplot=2, atau di x dengan logplot=3.

- logplot=1: y-logaritma
- logplot=2: x-y-logaritma
- logplot=3: x-logaritma

```

>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
>plot2d("exp(x+sin(x))",0,100,logplot=1):
>plot2d("exp(x+sin(x))",10,100,logplot=2):
>plot2d("gamma(x)",1,10,logplot=1):
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
Ini juga berfungsi dengan plot data.
>x=10^(1:20); y=x^2-x;
>plot2d(x,y,logplot=2):

```

## Sisipan Soal Plot Logaritmik

```
>plot2d("exp(x^2+x)",0,5,logplot=1)
```

## Rujukan Lengkap Fungsi plot2d()

```

function plot2d (xv, yv, btest, a, b, c, d, xmin, xmax, r, n, ...
logplot, grid, frame, framecolor, square, color, thickness, style,
..
.. auto, add, user, delta, points, addpoints, pointstyle, bar,
histogram, ..
distribution, even, steps, own, adaptive, hue, level, contour, ..
nc, filled, fillcolor, outline, title, xl, yl, maps, contourcolor,
..
.. contourwidth, ticks, margin, clipping, cx, cy, insimg, spectral,
..
cgrid, vertical, smaller, dl, niveau, levels)

Multipurpose plot function for plots in the plane (2D plots). This
function can do
plots of functions of one variables, data plots, curves in the plane,
bar plots, grids
of complex numbers, and implicit plots of functions of two variables.

```

#### Parameters

```

x,y      : equations, functions or data vectors
a,b,c,d  : Plot area (default a=-2,b=2)
r        : if r is set, then a=cx-r, b=cx+r, c=cy-r, d=cy+r
           r can be a vector [rx,ry] or a vector [rx1,rx2,ry1,ry2].
xmin,xmax : range of the parameter for curves
auto      : Determine y-range automatically (default)
square    : if true, try to keep square x-y-ranges
n         : number of intervals (default is adaptive)
grid      : 0 = no grid and labels,
           1 = axis only,
           2 = normal grid (see below for the number of grid lines)
           3 = inside axis
           4 = no grid
           5 = full grid including margin
           6 = ticks at the frame
           7 = axis only
           8 = axis only, sub-ticks
frame     : 0 = no frame
framecolor: color of the frame and the grid
margin    : number between 0 and 0.4 for the margin around the plot
color     : Color of curves. If this is a vector of colors,
           it will be used for each row of a matrix of plots. In the
case of
           point plots, it should be a column vector. If a row
vector or a
           full matrix of colors is used for point plots, it will be
used for
           each data point.
thickness : line thickness for curves
           This value can be smaller than 1 for very thin lines.
style     : Plot style for lines, markers, and fills.
           For points use
           "[", "<>", ".", "..", "...",
           "*", "+", "|", "-", "o"
           "[#]", "<>#", "o#" (filled shapes)
           "[w]", "<>w", "ow" (non-transparent)
           For lines use
           "-", "--", "-.", ".", "-.-", "-.-", "->"
           For filled polygons or bar plots use
           "#", "#O", "O", "/", "\\", "\\\",
           "+", "|", "-", "t"

```

```

points      : plot single points instead of line segments
addpoints   : if true, plots line segments and points
add         : add the plot to the existing plot
user        : enable user interaction for functions
delta       : step size for user interaction
bar         : bar plot (x are the interval bounds, y the interval
values)
histogram   : plots the frequencies of x in n subintervals
distribution=n : plots the distribution of x with n subintervals
even        : use inter values for automatic histograms.
steps       : plots the function as a step function (steps=1,2)
adaptive    : use adaptive plots (n is the minimal number of steps)
level       : plot level lines of an implicit function of two variables
outline     : draws boundary of level ranges.

```

If the level value is a 2xn matrix, ranges of levels will be drawn in the color using the given fill style. If outline is true, it will be drawn in the contour color. Using this feature, regions of  $f(x,y)$  between limits can be marked.

```

hue        : add hue color to the level plot to indicate the function
value
contour   : Use level plot with automatic levels
nc        : number of automatic level lines
title     : plot title (default "")
xl, yl    : labels for the x- and y-axis
smaller   : if >0, there will be more space to the left for labels.
vertical  :
    Turns vertical labels on or off. This changes the global variable
    verticallabels locally for one plot. The value 1 sets only vertical
    text, the value 2 uses vertical numerical labels on the y axis.
filled    : fill the plot of a curve
fillcolor  : fill color for bar and filled curves
outline   : boundary for filled polygons
logplot   : set logarithmic plots
    1 = logplot in y,
    2 = logplot in xy,
    3 = logplot in x
own       :
    A string, which points to an own plot routine. With >user, you get
    the same user interaction as in plot2d. The range will be set
    before each call to your function.
maps      : map expressions (0 is faster), functions are always
mapped.
contourcolor : color of contour lines
contourwidth : width of contour lines
clipping   : toggles the clipping (default is true)
title     :
    This can be used to describe the plot. The title will appear above
    the plot. Moreover, a label for the x and y axis can be added with
    xl="string" or yl="string". Other labels can be added with the
    functions label() or labelbox(). The title can be a unicode
    string or an image of a Latex formula.
cgrid     :
    Determines the number of grid lines for plots of complex grids.
    Should be a divisor of the the matrix size minus 1 (number of
    subintervals). cgrid can be a vector [cx,cy].

```

## Overview

The function can plot

- expressions, call collections or functions of one variable,
- parametric curves,
- x data against y data,
- implicit functions,
- bar plots,
- complex grids,
- polygons.

If a function or expression for `xv` is given, `plot2d()` will compute values in the given range using the function or expression. The expression must be an expression in the variable `x`. The range must be defined in the parameters `a` and `b` unless the default range `[-2,2]` should be used. The y-range will be computed automatically, unless `c` and `d` are specified, or a radius `r`, which yields the range `[-r,r]` for `x` and `y`. For plots of functions, `plot2d` will use an adaptive evaluation of the function by default. To speed up the plot for complicated functions, switch this off with `<adaptive`, and optionally decrease the number of intervals `n`. Moreover, `plot2d()` will by default use mapping. I.e., it will compute the plot element for element. If your expression or your functions can handle a vector `x`, you can switch that off with `<maps` for faster evaluation.

Note that adaptive plots are always computed element for element. If functions or expressions for both `xv` and for `yv` are specified, `plot2d()` will compute a curve with the `xv` values as x-coordinates and the `yv` values as y-coordinates. In this case, a range should be defined for the parameter using `xmin`, `xmax`. Expressions contained in strings must always be expressions in the parameter variable `x`.

# Adiyatma\_23030630062\_EMT4Plot3D (2)

---

Nama : Adiyatma  
NIM : 23030630062  
Kelas: Matematika B 2023

## 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 40x40 untuk membuat permukaan. Ini bisa diubah.

- $n = 40, n = [40,40]$ : jumlah garis grid di setiap arah
- $grid= 10, grid = [10,10]$ : jumlah garis kisi di setiap arah.

Kami menggunakan default  $n = 40$  dan  $grid = 10$ .

```
>plot3d("x^2+y^2");
```

Interaksi pengguna 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)":
```

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

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,>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)":
```

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

dapat divisualisasikan dalam potongan sejajar dengan bidang x-y-, x-z-, dan y-z.

- implicit = 1: potong sejajar bidang y-z
- implicit = 2: potong sejajar dengan bidang x-z
- implicit = 4: potong sejajar bidang x-y

Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh yang kami plot

```
>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 fx (x, y), fy (x, y), fz (x, y).

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

```
>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

dengan

Kami menyimpangkan ini dengan sebuah faktor

```
>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=blue,<outline, ...
  level=[-2;0],n=100):
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

Kami ingin memutar kurva jantung di sekitar sumbu y. Inilah ungkapan yang mendefinisikan hati:

Selanjutnya kita atur

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce;
$fr(r,a)
```

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=red,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);  
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);
```

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);
```

## 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]);
```

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);
```

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
      2   3
      x   y
```

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

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);
```

Berikut ini adalah simpul Trefoil yang dilakukan oleh A. Busser di Povray. Ada versi perbaikannya dalam contoh.

### 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));
```

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();
```

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();
```

## 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 }
}
```

Seperi 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;
```

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();
```

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));
```

Sekarang tambahkan tiga sumbu.

```
>writeAxis(-1.2,1.2,axis=1); ...
writeAxis(-1.2,1.2,axis=2); ...
writeAxis(-1.2,1.2,axis=4); ...
povend();
```

## Fungsi Implisit

Povray dapat memplot himpunan di mana  $f(x, y, z) = 0$ , seperti parameter implisit 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();
```

## 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();
```

## 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);
```

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);
  clx=povobject(cl,rotate=xrotate(90°));
  cly=povobject(cl,rotate=yrotate(90°));
  c=povbox([-1,-1,0],1);
  un=povunion([cl,clx,cly,c]);
  obj=povdifference(s,un,povlook(red));
  writeln(obj);
  writeAxes();
endfunction
```

Fungsi povanaglyph () melakukan semua ini. Parameternya seperti di povstart () dan povend () digabungkan.

```
>povanaglyph ("myscene", zoom=4.5);
```

## 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 parameteranya. 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 menyelesaiakannya

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

```
>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[#]],r);
    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();
```

## Latihan Soal

---

1. Buatkan grafik

```
>plot3d("x^3-4*y^2"):
```

2. Buatkan grafik

dengan  $a=-2$ ,  $b=2$ , dan  $\text{grid}=18$

```
>plot3d("x^3-3*x+4",a=-2,b=2,rotate=true,grid=18):
```

3. Buatlah grafik plot kontur

```
>plot3d("exp(3*x*y)",angle=150°,>contour,color=yellow):
```

4. Buat grafik plot implisit

```
>plot3d("x^2+y^2+5*x*z+3*z^3",>implicit,r=2,zoom=2.2):
```

# **EMT\_Kalkulus\_Adiyatma\_23030630062**

## **\_MatB**

---

Adiyatma  
23030630062  
Matematika B

### **Kalkulus dengan EMT**

---

Materi Kalkulus mencakup di antaranya:

- Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi)
- Limit Fungsi,
- Turunan Fungsi,
- Integral Tak Tentu,
- Integral Tentu dan Aplikasinya,
- Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

## **Mendefinisikan Fungsi**

---

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:

- Menggunakan format `nama_fungsi := rumus_fungsi` (untuk fungsi numerik),
- Menggunakan format `nama_fungsi &= rumus_fungsi` (untuk fungsi simbolik, namun dapat dihitung secara numerik),
- Menggunakan format `nama_fungsi &&= rumus_fungsi` (untuk fungsi simbolik murni, tidak dapat dihitung langsung),
- Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah `function` (bukan sebagai ekspresi).

Berikut adalah beberapa contoh cara mendefinisikan fungsi:

$$f(x)=2x^2+e\sin(x). f(x)=2x^2+e\sin(\frac{\pi}{2}x).$$

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
1
4.31977682472
20.7392088022
>f(a) // tidak dapat dihitung nilainya
Variable or function a not found.
Error in:
f(a) // tidak dapat dihitung nilainya ...
^
```

Silakan Anda plot kurva fungsi di atas!

Berikutnya kita definisikan fungsi:

```

g(x)=x^2-3x-----sqrt(x+1).g(x)=x^2-3xx+1.

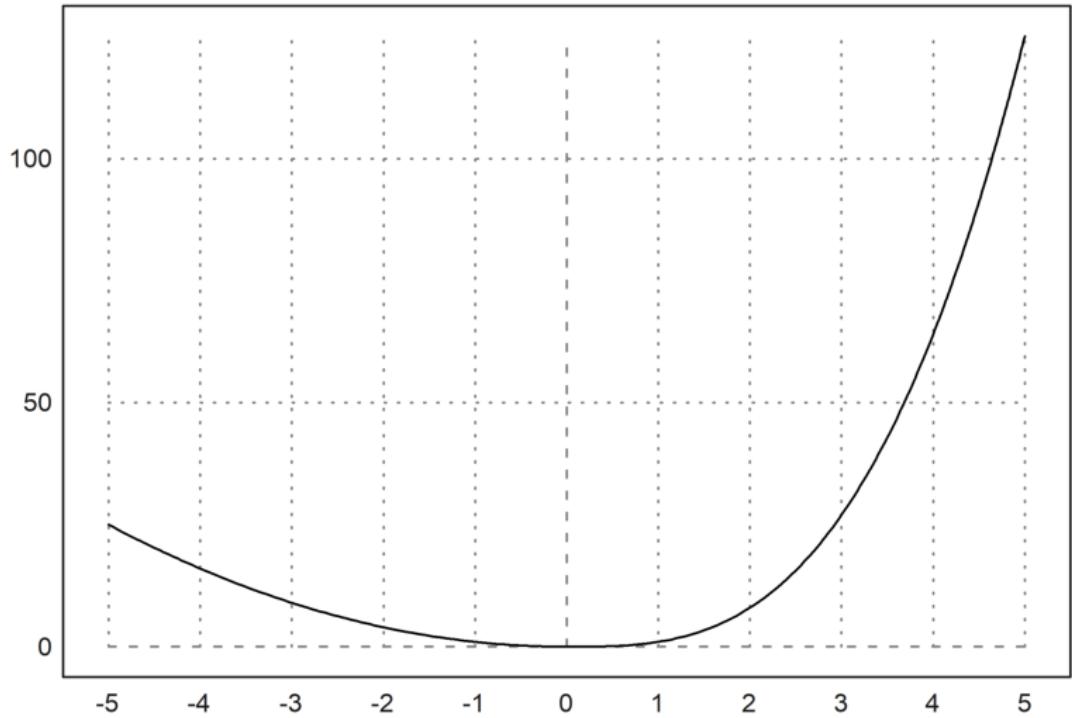
>function g(x) := sqrt(x^2-3*x) / (x+1)
>g(3)
0
>g(0)
0
>g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik
Floating point error!
Error in sqrt
Try "trace errors" to inspect local variables after errors.
g:
    useglobal; return sqrt(x^2-3*x) / (x+1)
Error in:
g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik ...
^
Silakan Anda plot kurva fungsi di atas!
>f(g(5)) // komposisi fungsi
2.20920171961
>g(f(5))
0.950898070639
>function h(x) := f(g(x)) // definisi komposisi fungsi
>h(5) // sama dengan f(g(5))
2.20920171961
Silakan Anda plot kurva fungsi komposisi fungsi f dan g:
h(x)=f(g(x))h(x)=f(g(x))

dan
u(x)=g(f(x))u(x)=g(f(x))

bersama-sama kurva fungsi f dan g dalam satu bidang koordinat.
>f(0:10) // nilai-nilai f(0), f(1), f(2), ..., f(10)
[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,
99.929, 130.69, 163.51, 200.58]
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,
99.929, 130.69, 163.51, 200.58]
>gmap(200:210)
[0.987534, 0.987596, 0.987657, 0.987718, 0.987778, 0.987837,
0.987896, 0.987954, 0.988012, 0.988069, 0.988126]
Misalkan kita akan mendefinisikan fungsi
f(x)={x^3|x>0}x^2|x≤0}.f(x)={x^3|x>0}x^2|x≤0}.

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik
secara "inline" menggunakan format :=, melainkan didefinisikan
sebagai
program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima
vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata
"map" fungsinya hanya dapat menerima input satu nilai.
>function map f(x) ...
  if x>0 then return x^3
  else return x^2
  endif;
endfunction
>f(1)
1
>f(-2)
4
>f(-5:5)
[25, 16, 9, 4, 1, 0, 1, 8, 27, 64, 125]
>aspect(1.5); plot2d("f(x)", -5, 5):

```



```
>function f(x) &= 2*x // fungsi simbolik
          x
          2 E

>$f(a) // nilai fungsi secara simbolik
2ea2ea

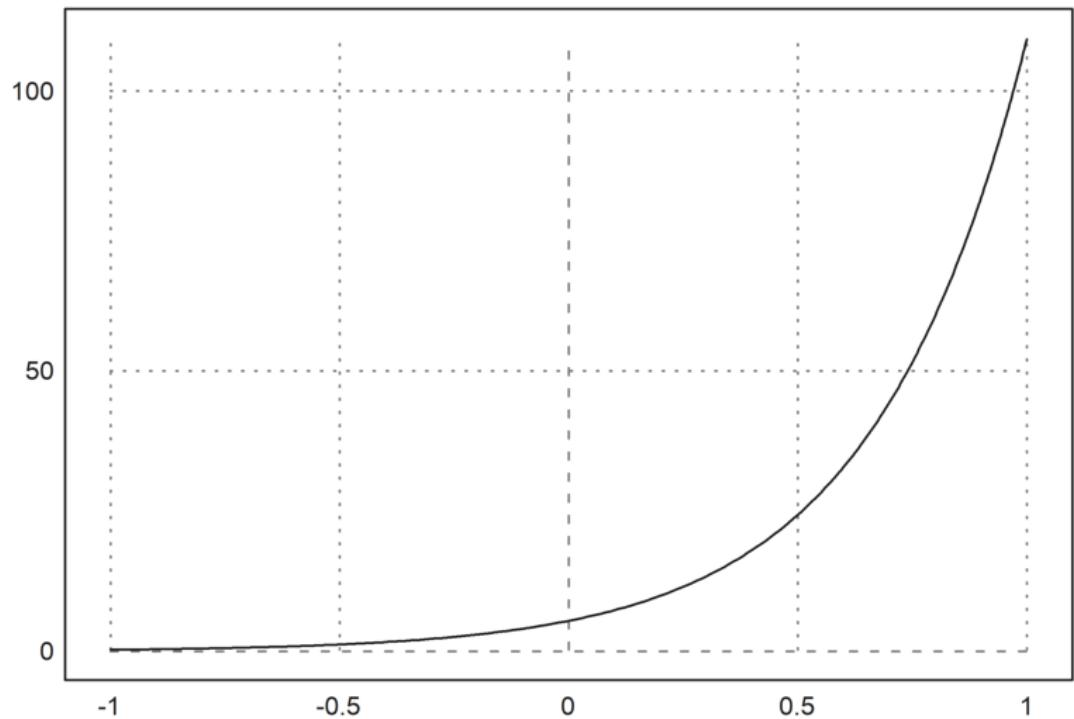
>f(E) // nilai fungsi berupa bilangan desimal
30.308524483
>$f(E), $float(%)
30.3085244829585230.30852448295852
```

**30.30852448295852**

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

>function h(x) &= f(g(x)) // komposisi fungsi
          3 x + 1
          2 E

>plot2d("h(x)", -1, 1):
```



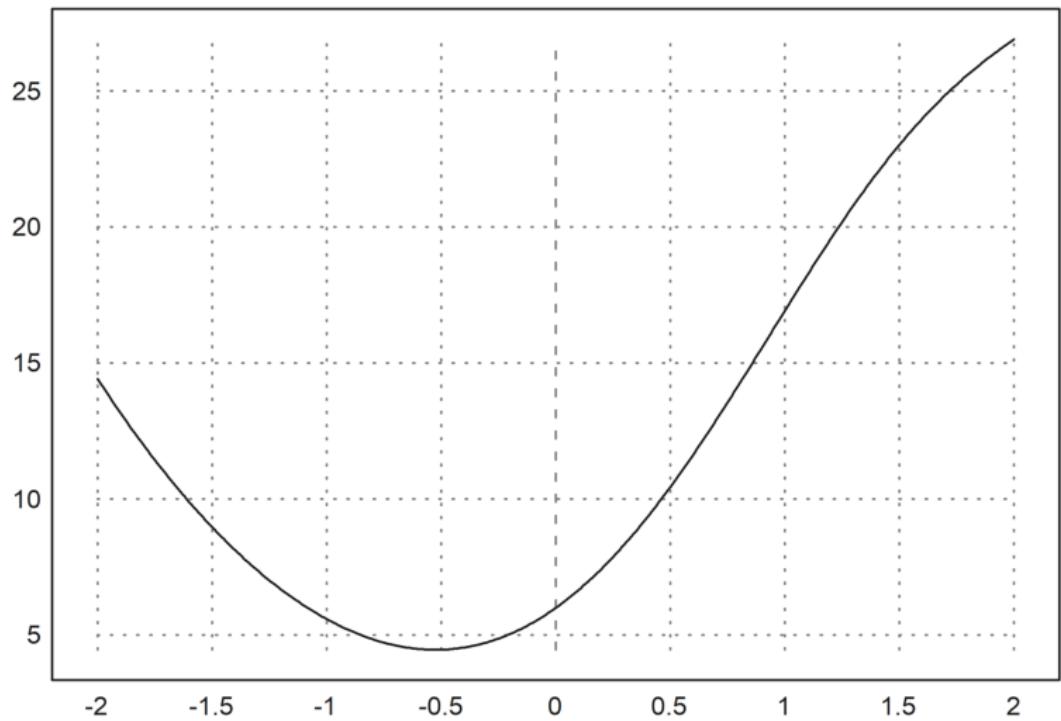
## Latihan

---

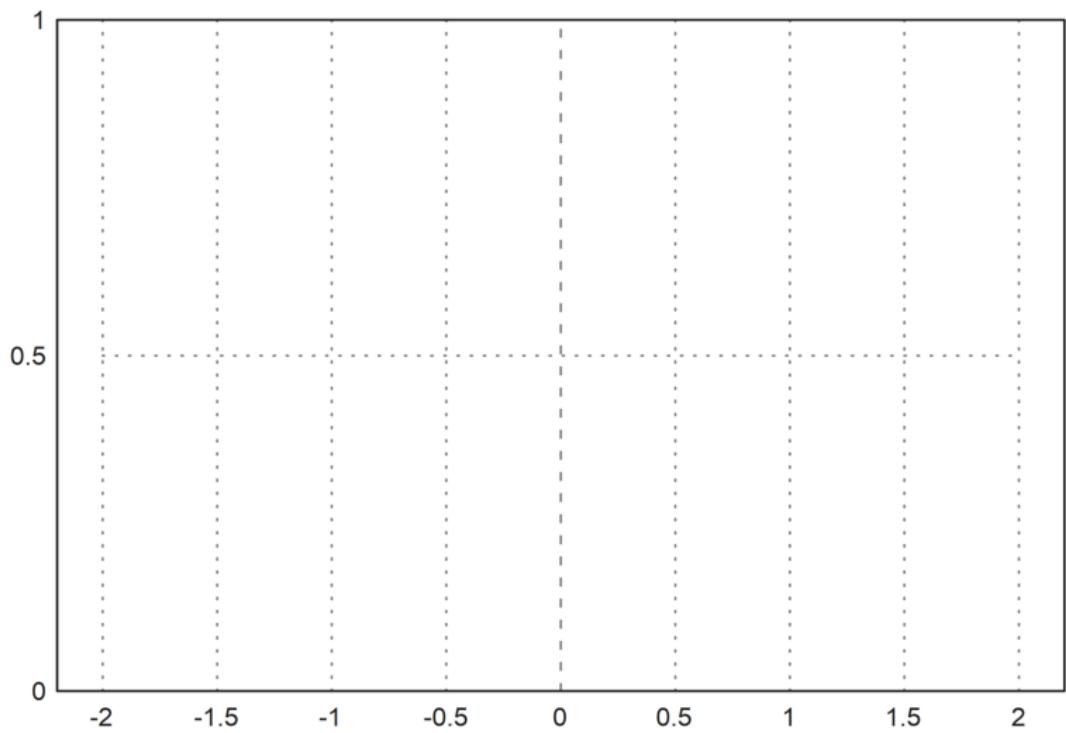
Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan fungsi-fungsi tersebut dan komposisinya di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik fungsi-fungsi tersebut dan komposisi-komposisi 2 fungsi.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

```
>function g(x) := 3*x^2+6*exp(sin(x))
>g(0), g(3)
6
33.9093770191
>plot2d("g(x)":
```



```
>function a(x) := sqrt(2*x^2-8)/(x+2)
>a(4)
0.816496580928
>plot2d("a(x)":
```



```

>function f(x,y) := x^2+y^2
>f(2,3)
      13
>function f(x) &= 3*E^2*x
              2
            3 E   x

```

## Menghitung Limit

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit".

Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi "plus" atau "minus". Hasil limit dapat berupa nilai, "und" (tak definisi), "ind" (tak tentu namun terbatas), "infinity" (kompleks tak hingga).

Perhatikan beberapa contoh berikut. Perhatikan cara menampilkan perhitungan secara lengkap, tidak hanya menampilkan hasilnya saja.

```

>$showev('limit(sqrt(x^2-3*x)/(x+1),x,inf))
      lim_{x \rightarrow \infty} \frac{\sqrt{x^2-3x}}{x+1} = \lim_{x \rightarrow \infty} \frac{x\sqrt{1-\frac{3}{x}}}{x+1} = \lim_{x \rightarrow \infty} \frac{x\sqrt{1-\frac{3}{x}}}{x(1+\frac{1}{x})} = \lim_{x \rightarrow \infty} \frac{\sqrt{1-\frac{3}{x}}}{1+\frac{1}{x}} = \frac{\sqrt{1-0}}{1+0} = 1

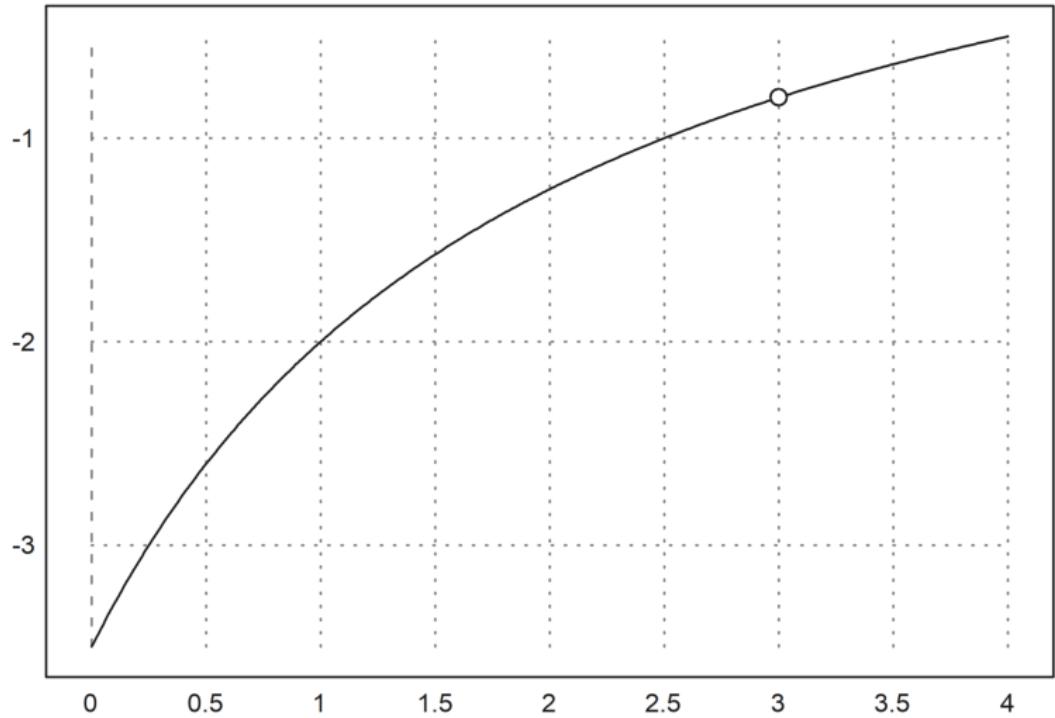
>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
      -45-45

maxima: 'limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)=limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

Fungsi tersebut diskontinu di titik  $x=3$ . Berikut adalah grafik fungsinya.

```

>aspect(1.5); plot2d("(x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18)",0,4);
plot2d(3,-4/5,>points,style="ow",>add):
```



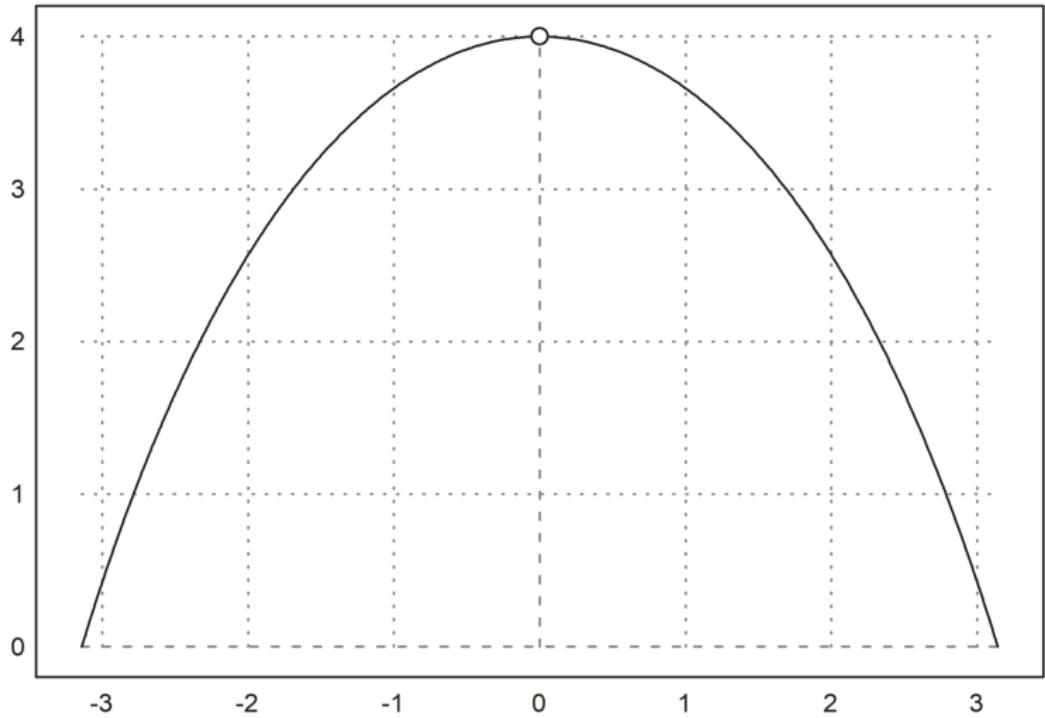
```
>$limit(2*x*sin(x)/(1-cos(x)),x,0)
```

44

```
maxima: 'limit(2*x*sin(x)/(1-cos(x)),x,0)=limit(2*x*sin(x)/(1-
cos(x)),x,0)
```

Fungsi tersebut diskontinu di titik  $x=0$ . Berikut adalah grafik fungsinya.

```
>plot2d("2*x*sin(x)/(1-cos(x))",-pi,pi);
plot2d(0,4,>points,style="ow",>add):
```



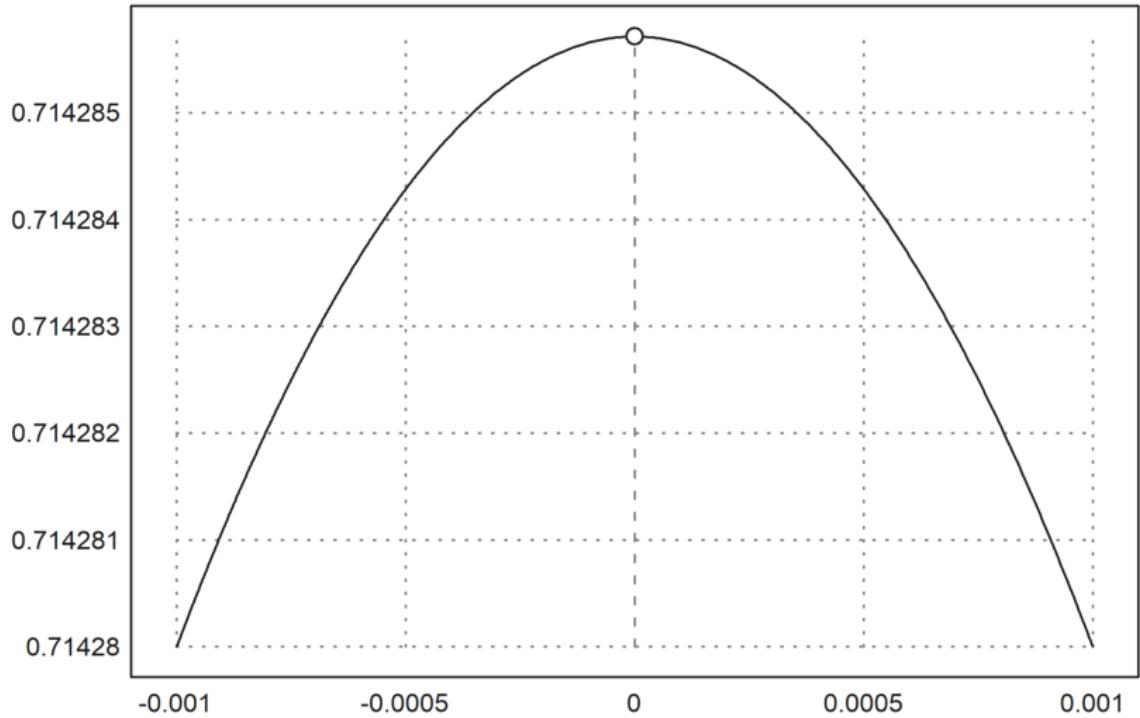
```
>$limit(cot(7*h)/cot(5*h),h,0)
```

5757

```
maxima: showev('limit(cot(7*h)/cot(5*h),h,0))
```

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di  $x=0$ .  
Berikut adalah grafiknya.

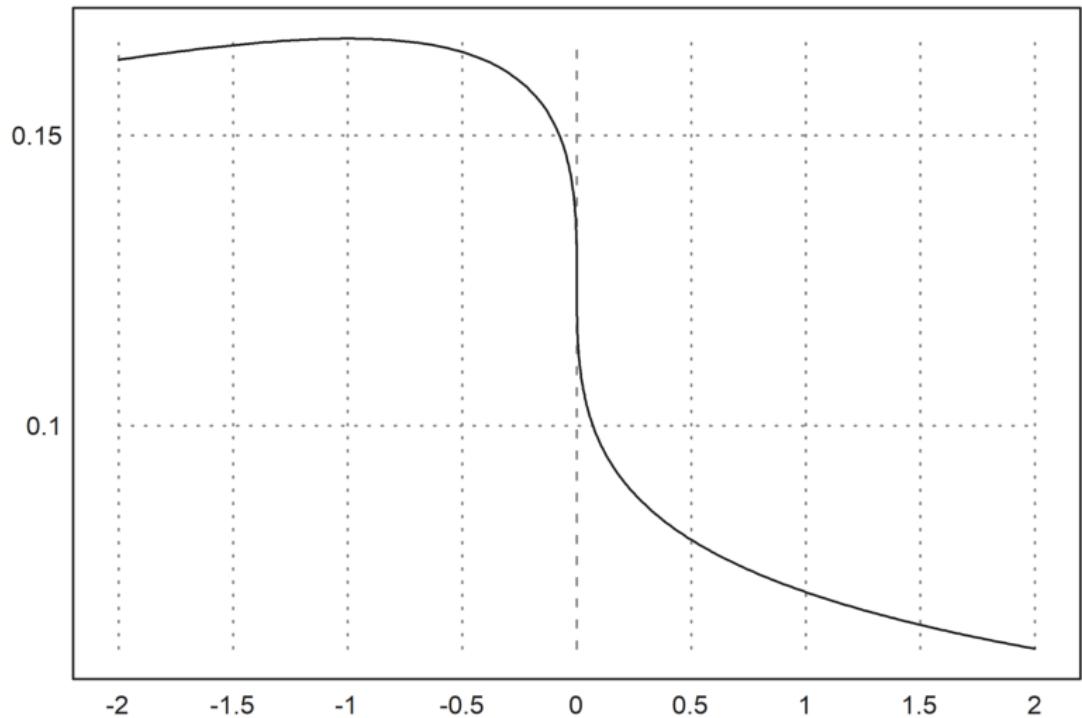
```
>plot2d("cot(7*x)/cot(5*x)",-0.001,0.001);
plot2d(0,5/7,>points,style="ow",>add):
```



```
>$showev('limit(((x/8)^(1/3)-1)/(x-8),x,8))
lim_{x \rightarrow 8} \frac{x^{1/3}-1}{x-8}=124 lim_{x \rightarrow 8} x^{1/3}-1=124
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

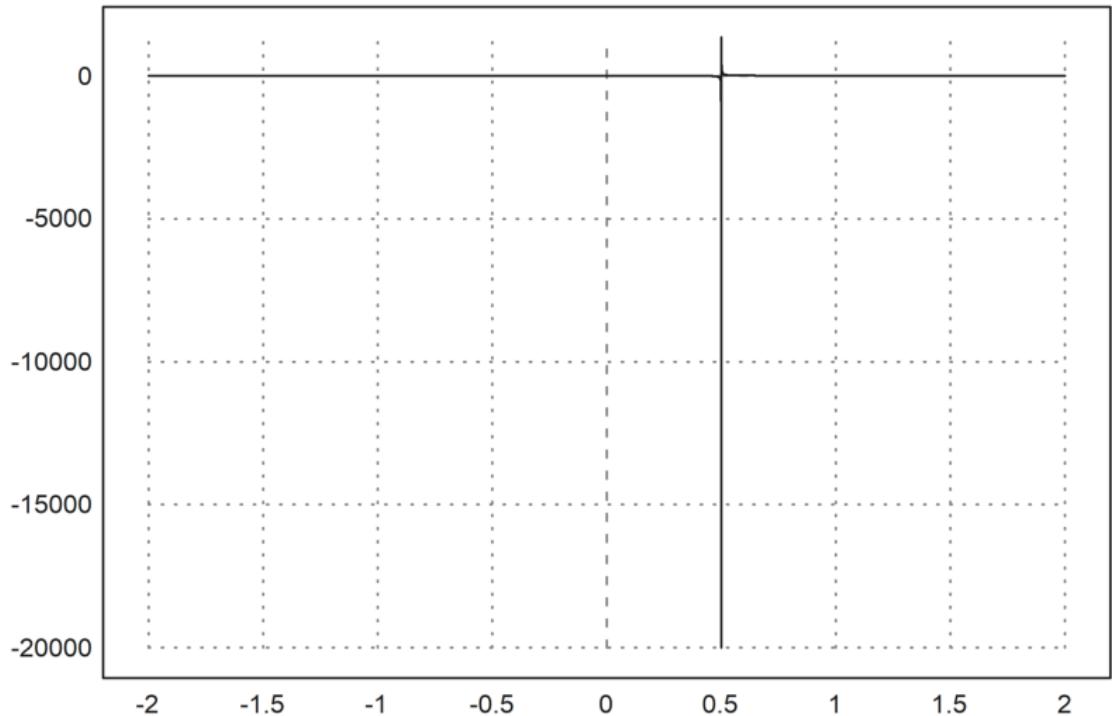
```
>plot2d("(((x/8)^(1/3)-1)/(x-8))":
```



```
>$showev('limit(1/(2*x-1),x,0))  
lim_{x \rightarrow 0} 1/(2x - 1) = -1
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("1/(2*x-1)":
```



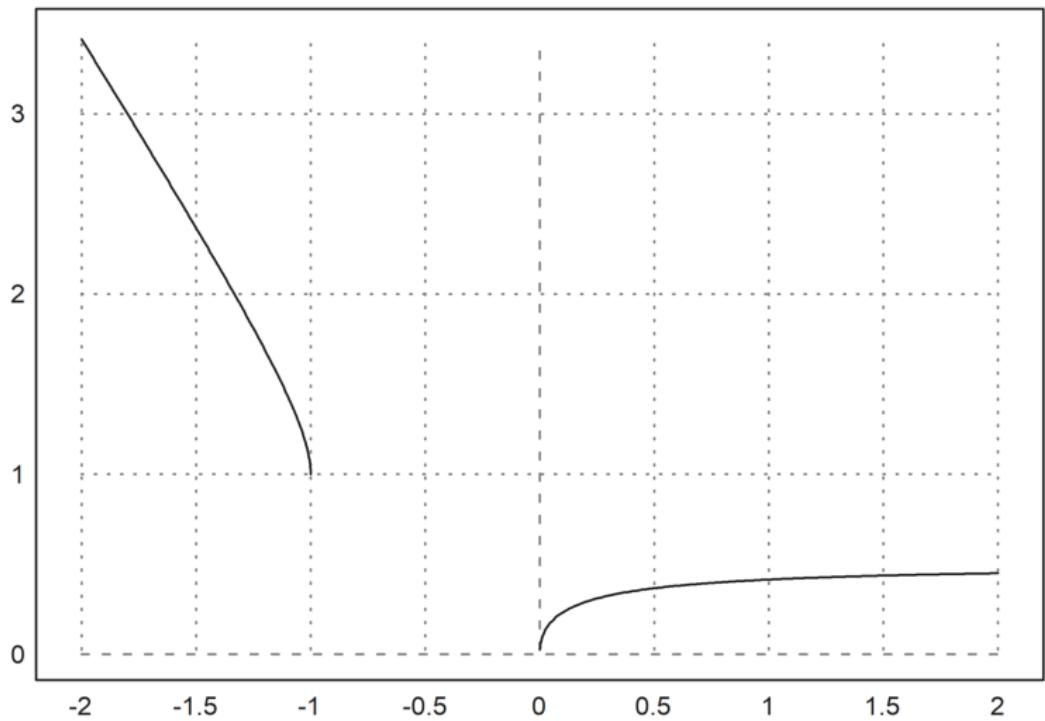
```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))  
lim_{x \rightarrow 5} (x^2 - 3x - 10)/(x - 5) = 7
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(sqrt(x^2+x)-x,x,inf))  
lim_{x \rightarrow \infty} \sqrt{x^2 + x} - x = 12
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(sqrt(x^2+x))-x":
```

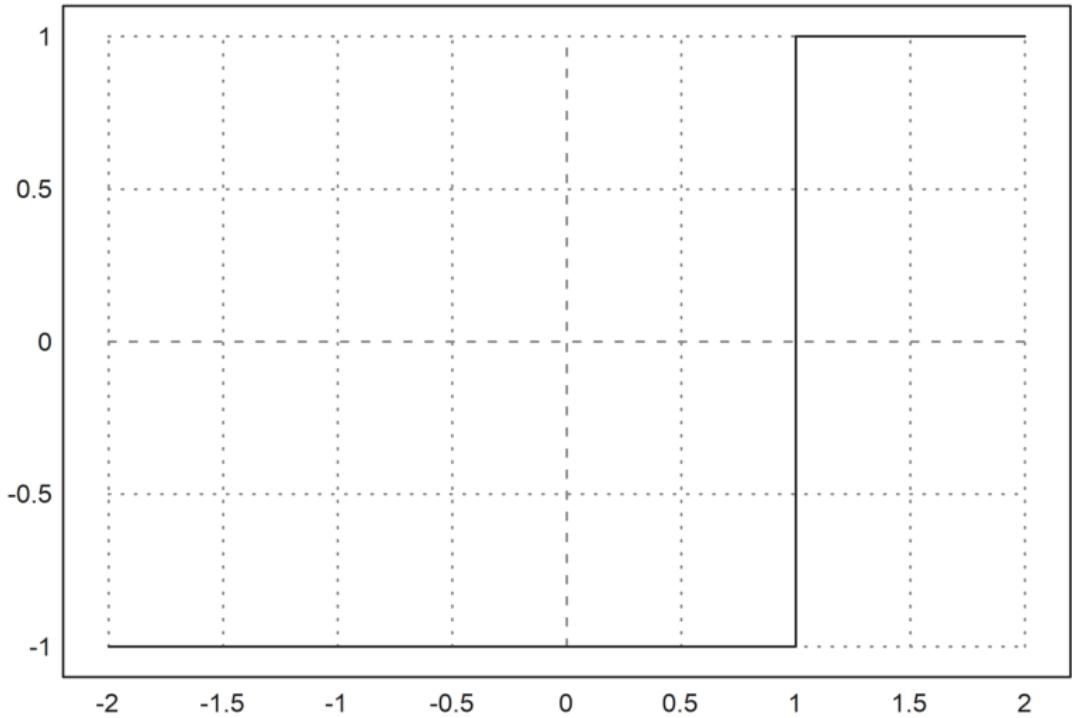


```
>$showev('limit(abs(x-1)/(x-1),x,1,minus))  
lim_{x \rightarrow 1^-} \frac{|x-1|}{x-1}
```

Hitung limit di atas untuk  $x$  menuju 1 dari kanan.

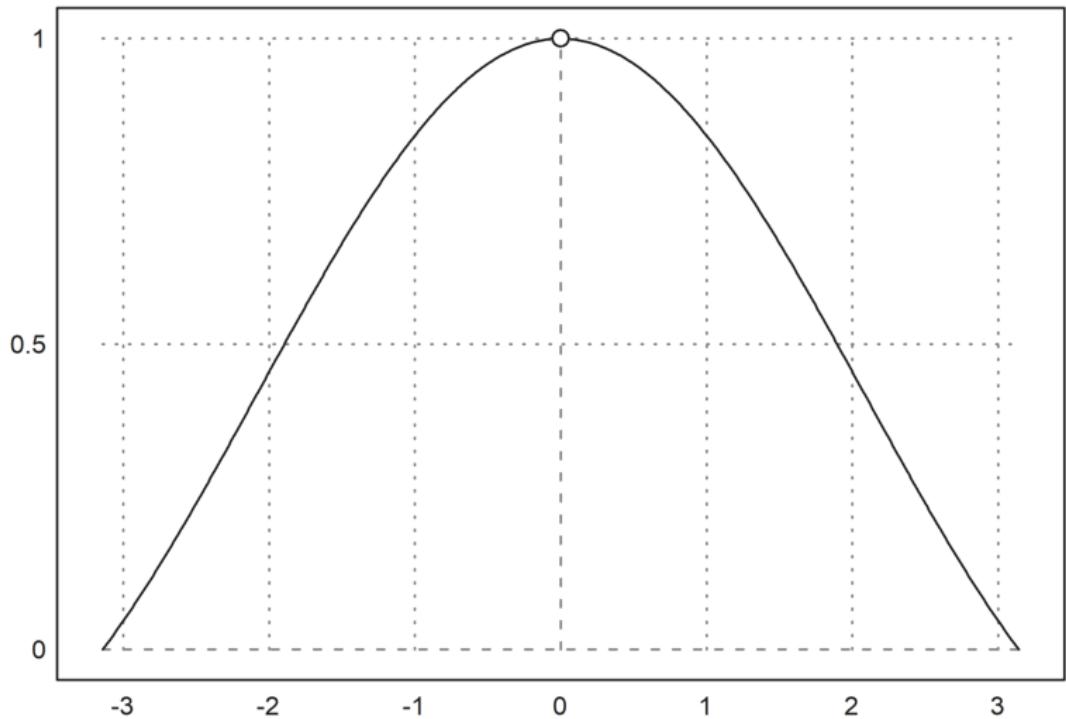
Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(abs(x-1)/(x-1))":
```



```
>$showev('limit(sin(x)/x,x,0))
lim_{x \rightarrow 0} \sin x / x = 1
lim_{x \rightarrow 0} \sin[f_0] x / x = 1

>plot2d("sin(x)/x", -pi, pi); plot2d(0, 1, >points, style="ow", >add) :
```

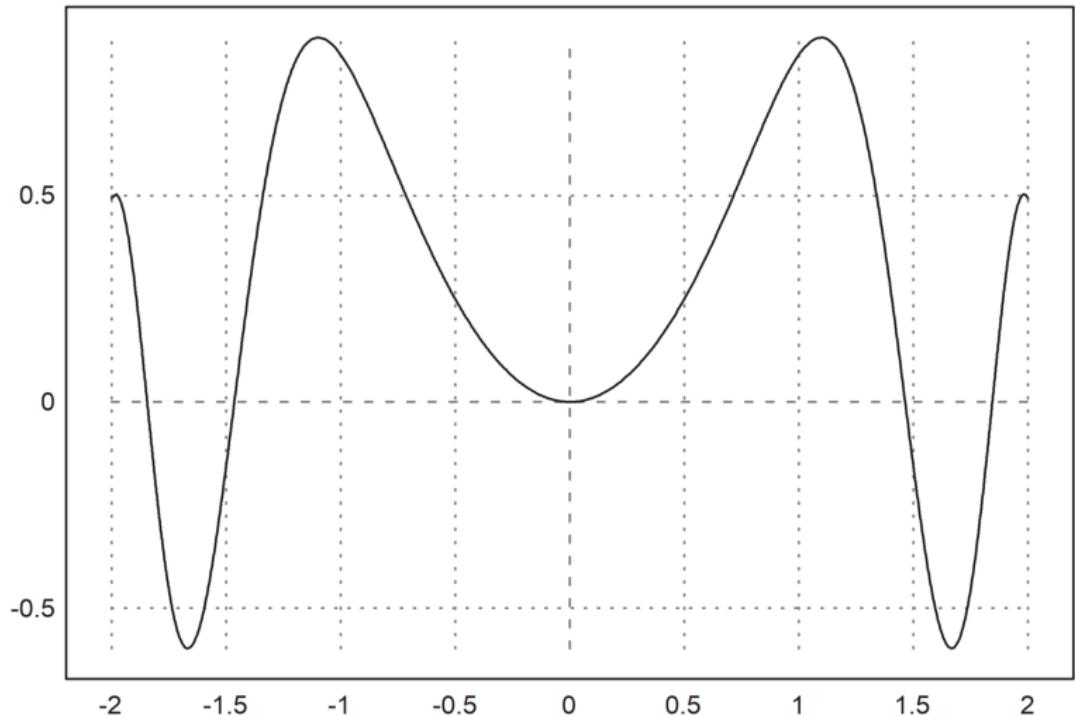


```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x \rightarrow 0} \sin x / 3x = 0$$

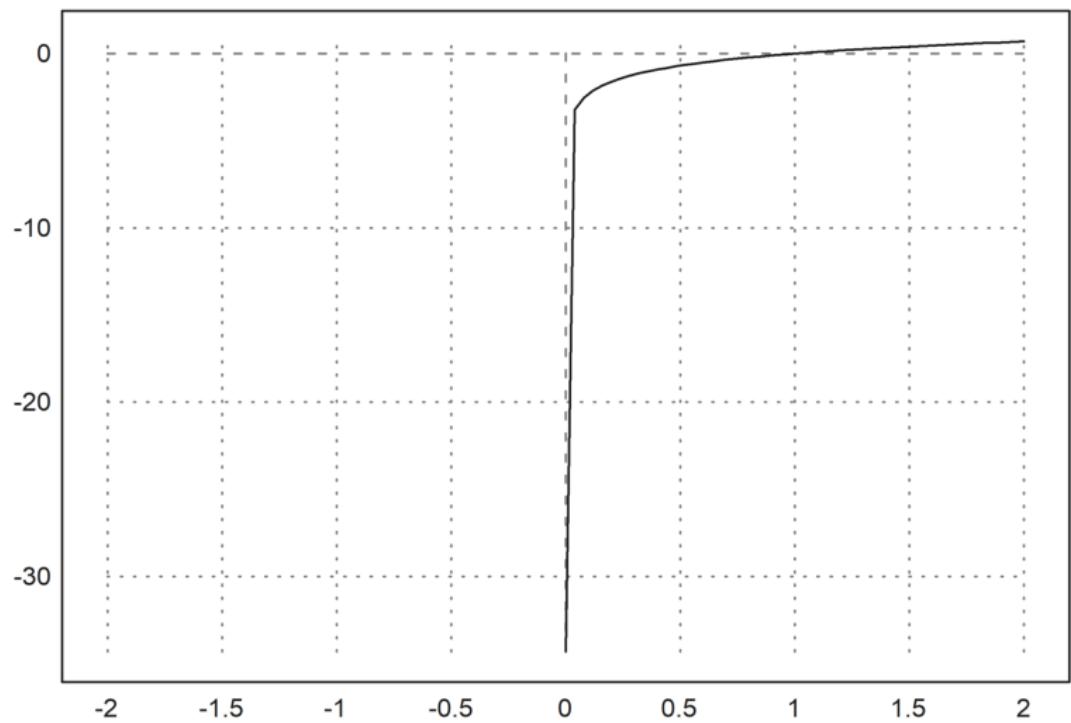
Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(sin(x^3))/x"):
```



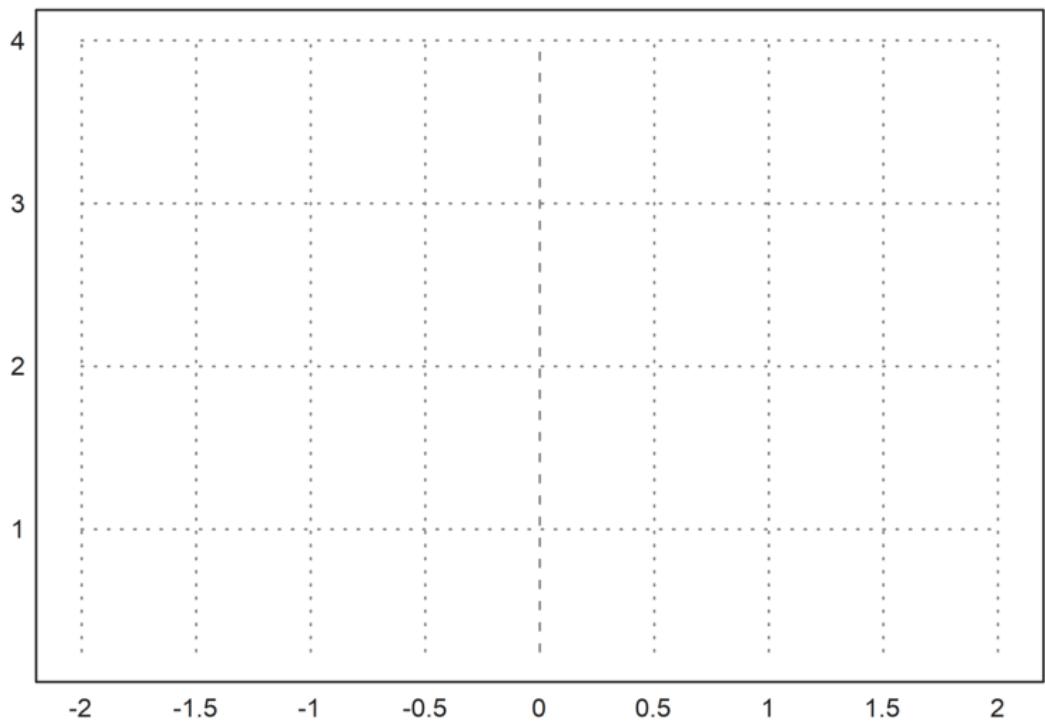
```
>$showev('limit(log(x), x, minf)')  
lim_{x \rightarrow -\infty} \log x = -\infty
```

```
>plot2d("log(x)"):
```



```
>$showev('limit((-2)^x,x, inf))
lim_{x \rightarrow \infty} (-2)^x = \infty
```

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

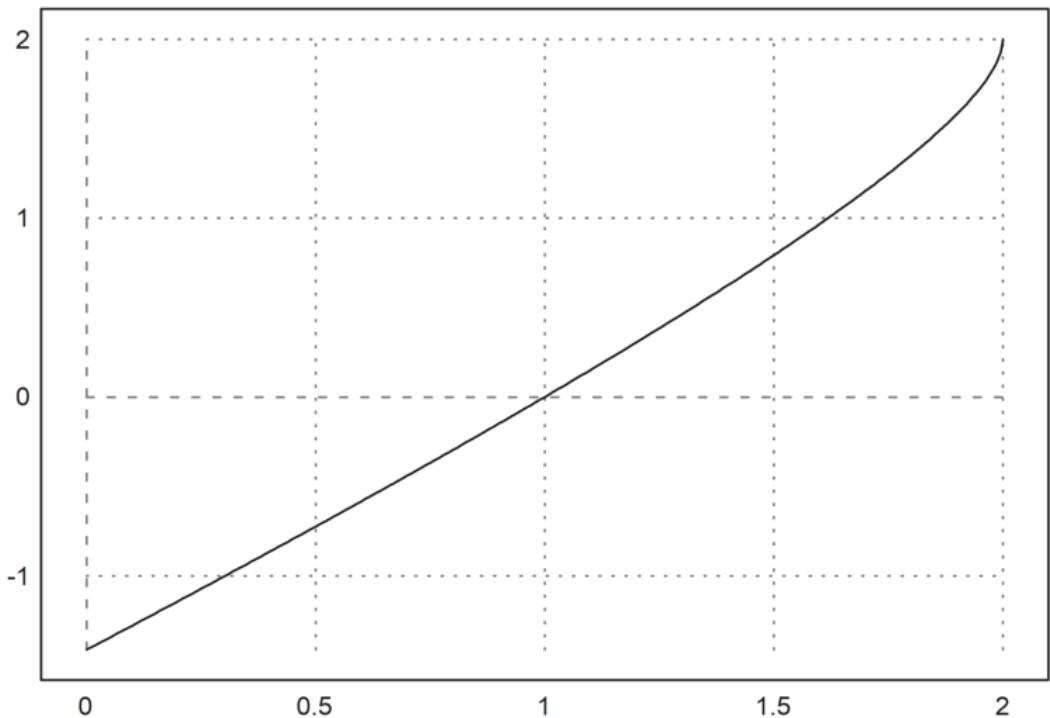


```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

```

limt→2 $\sqrt{2-t}$  = 2
>$showev('limit(t-sqrt(2-t),t,2,plus))
limt→2 $\sqrt{2-t}$  = 2
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
limt→5 $\sqrt{2-t}$  = 5 - 3 - sqrt(5 - 2) = 5 - 3i
>plot2d("x-sqrt(2-x)",0,2):

```

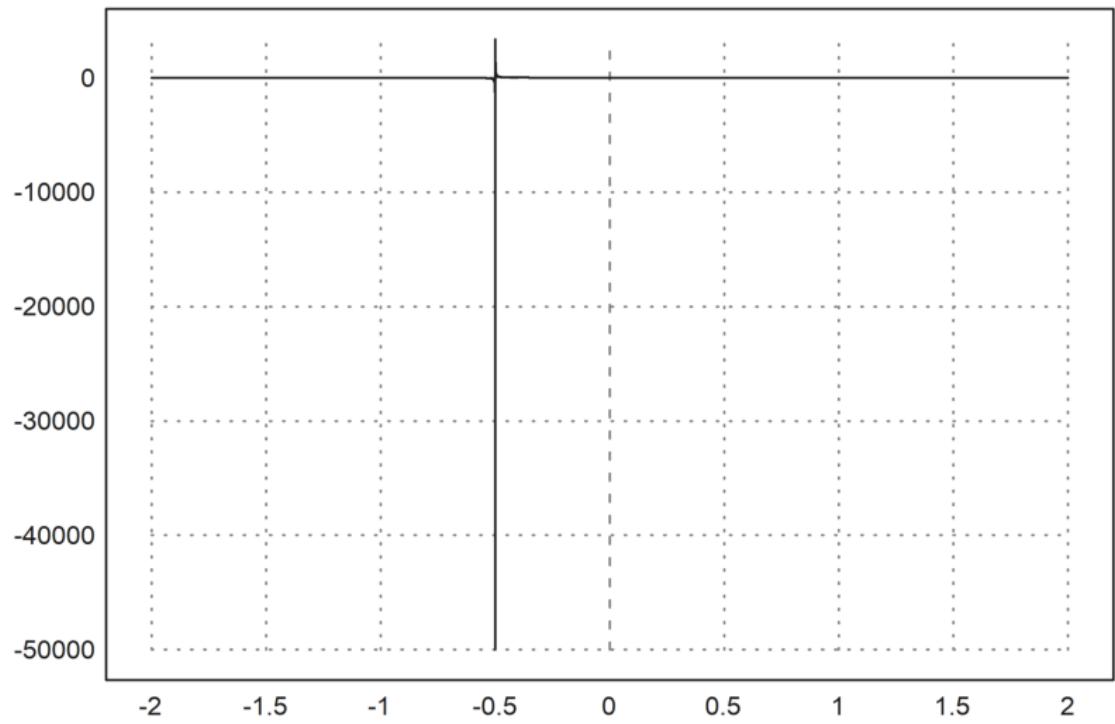


```

>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
limx→3 $\frac{x^2-9}{2x^2-5x-3} = 67$ 
limx→3 $\frac{x^2-9}{2x^2-5x-3} = 67$ 
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

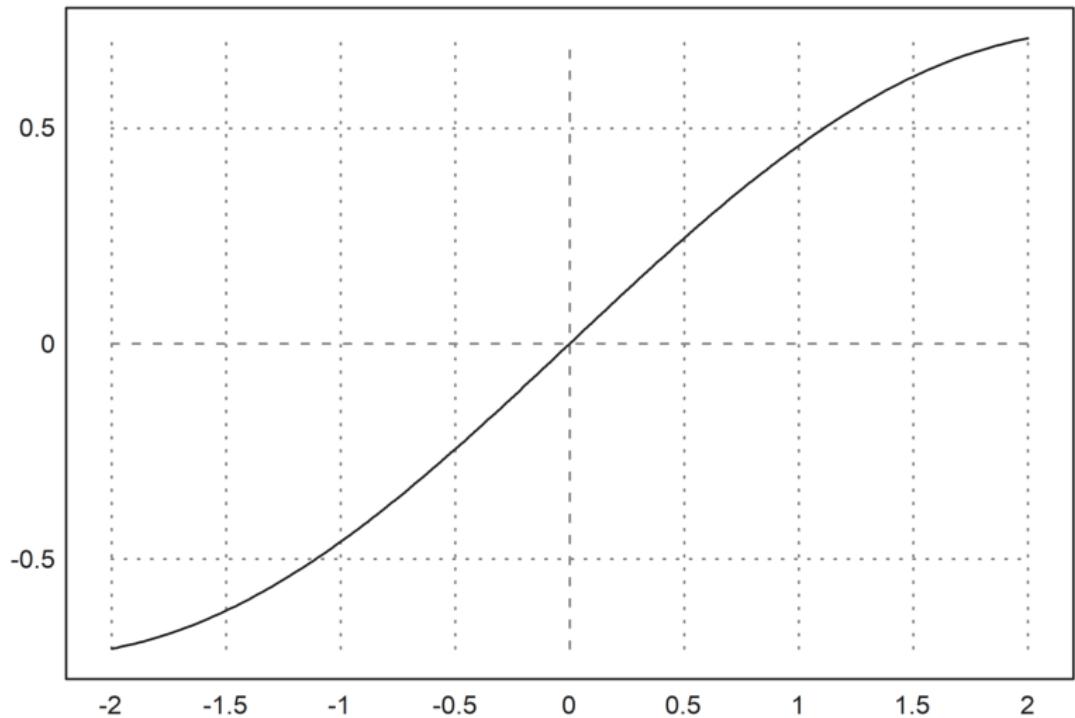
```
>plot2d("(x^2-9)/(2*x^2-5*x-3)":
```



```
>$showev('limit((1-cos(x))/x,x,0))  
lim_{x \rightarrow 0} \frac{1-\cos x}{x}=0
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1-cos(x))/x"):
```

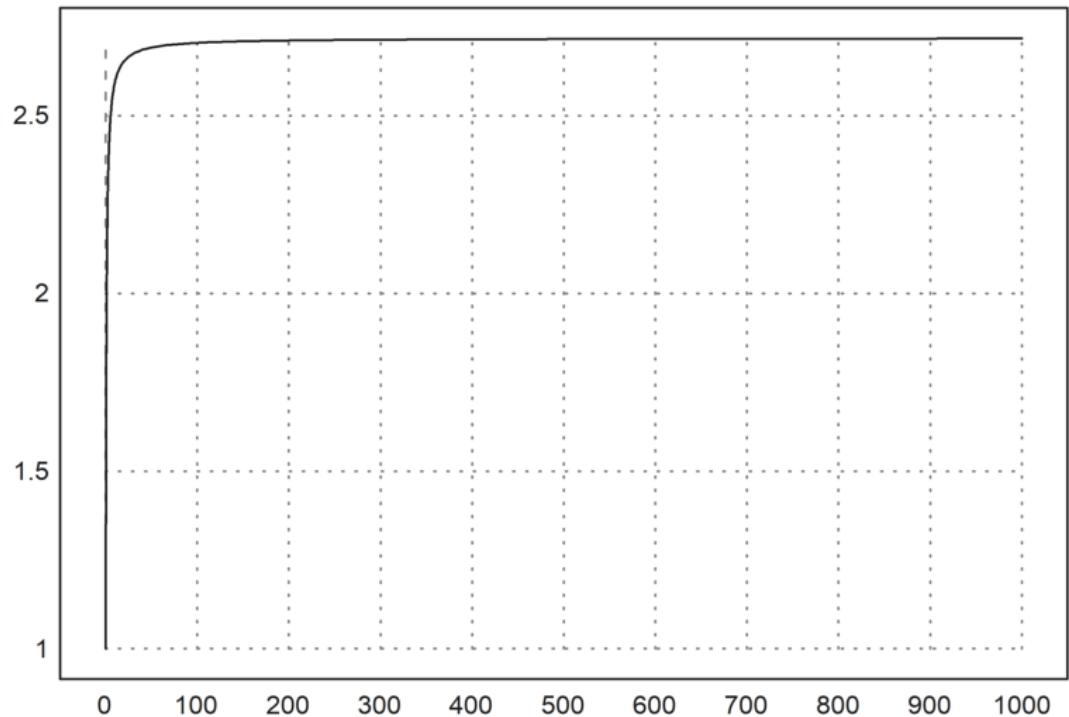


```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
lim_{x \rightarrow 0} |x| + x^2 - |x| = -1 lim_{x \rightarrow 0} |x| + x^2 - |x| = -1
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit((1+1/x)^x,x,inf))
lim_{x \rightarrow \infty} (1+x) = e lim_{x \rightarrow \infty} (1+x) = e
```

```
>plot2d("(1+1/x)^x",0,1000):
```

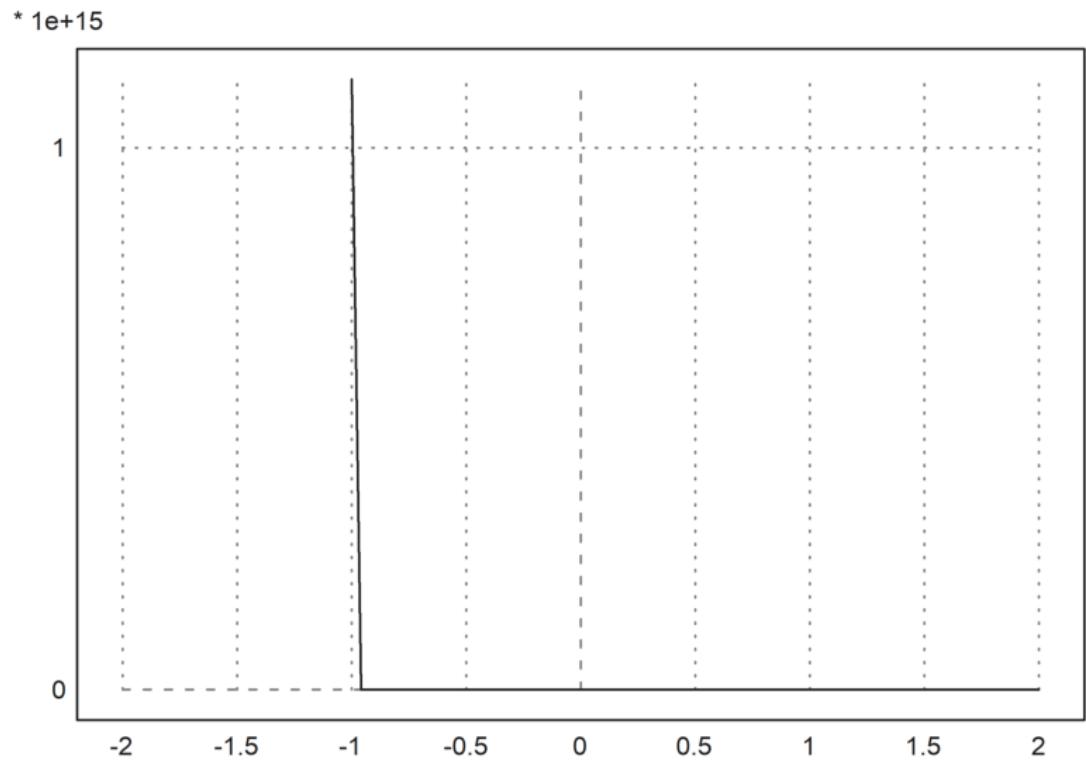


```
>$showev('limit((1+k/x)^x,x,inf))
lim_{x \rightarrow \infty} (kx+1) = e k lim_{x \rightarrow \infty} (kx+1) = e k
```

```
>$showev('limit((1+x)^(1/x),x,0))
lim_{x \rightarrow 0} (x+1) = e lim_{x \rightarrow 0} (x+1) = e
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((1+x)^(1/x))":
```



```
>showev('limit((x/(x+k))^x,x,inf))
lim_{x \rightarrow \infty} (x/x+k)^x = e^{-k} lim_{x \rightarrow \infty} (x+k)x = e^{-k}
```

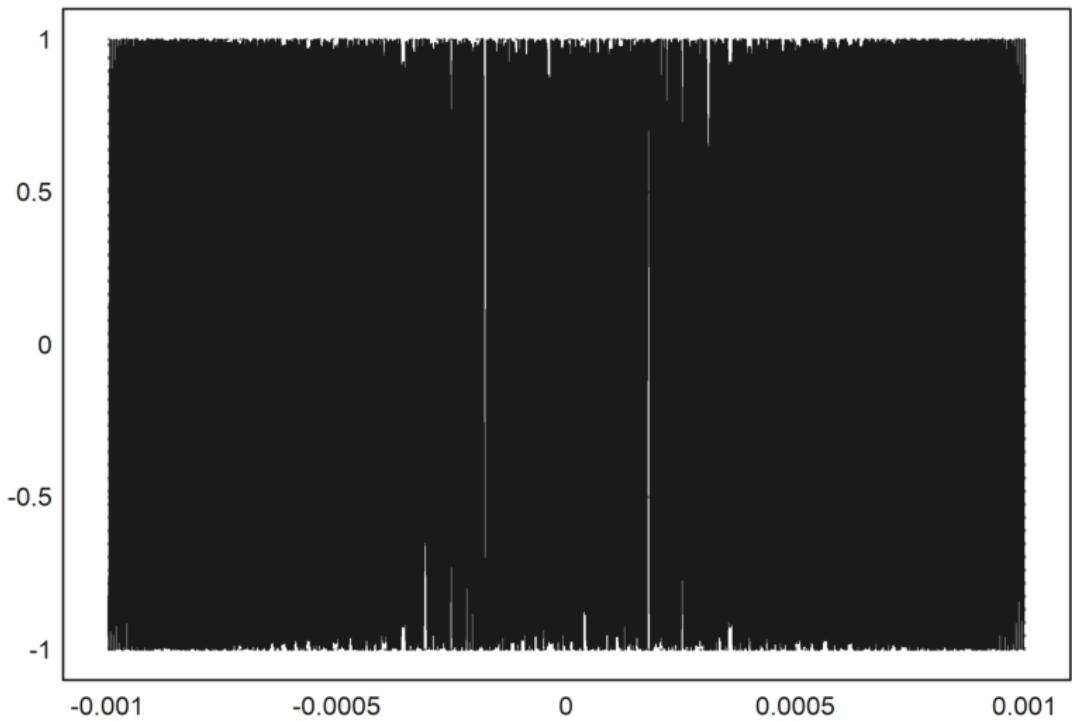
```
>showev('limit((E^x-E^2)/(x-2),x,2))
lim_{x \rightarrow 2} E^x - E^2 x - 2 = E^2 lim_{x \rightarrow 2} E^x - E^2 x - 2 = E^2
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>showev('limit(sin(1/x),x,0))
lim_{x \rightarrow 0} sin(1/x) = ind lim_{x \rightarrow 0} sin[0](1/x) = ind
```

```
>showev('limit(sin(1/x),x,inf))
lim_{x \rightarrow \infty} sin(1/x) = 0 lim_{x \rightarrow \infty} sin[0](1/x) = 0
```

```
>plot2d("sin(1/x)",-0.001,0.001):
```



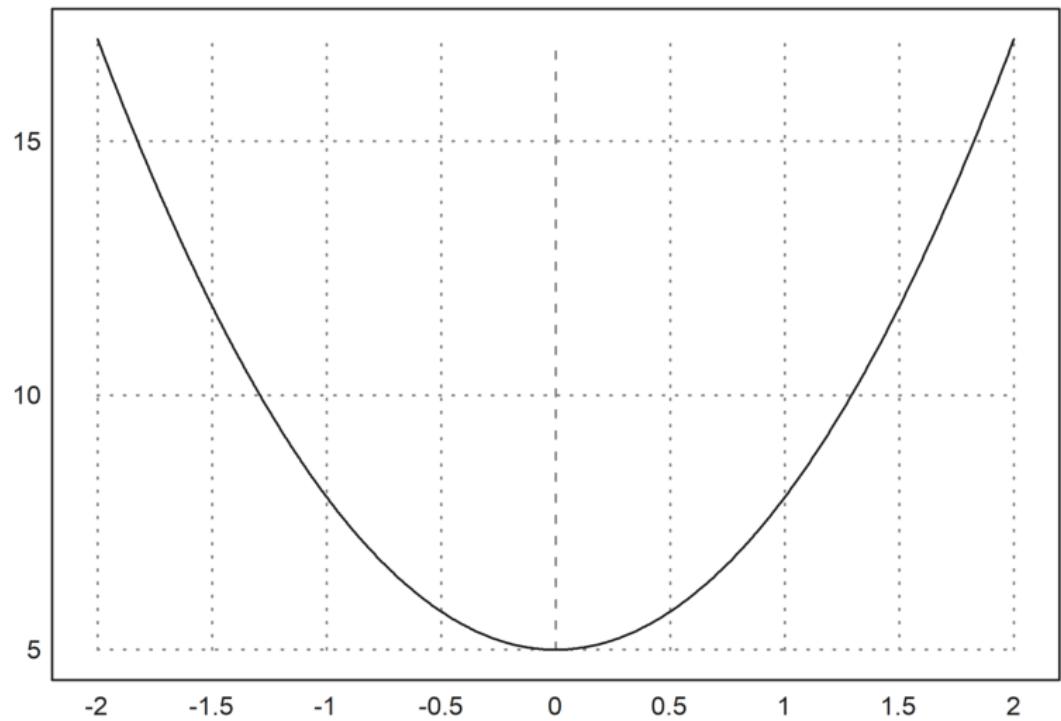
## Latihan

---

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

```
>$showev('limit((3*x^2+5),x,3))  
lim_{x \rightarrow 3} 3x^2+5=32 lim_{x \rightarrow 3} x^2+5=32
```

```
>plot2d(" (3*x^2+5) "):
```



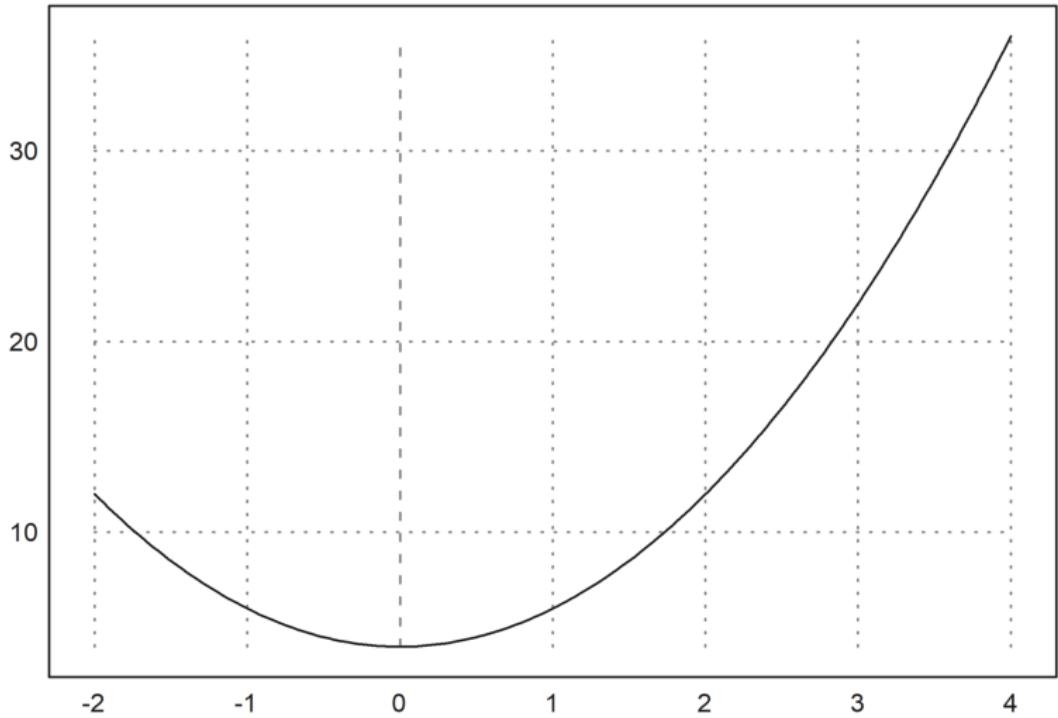
```
>&showev('limit((x^2-9),x,3))
```

$$\lim_{\substack{x \rightarrow 3}} (x^2 - 9) = 0$$

```
>&showev('limit((2*x^2+4),x,2))
```

$$\lim_{\substack{x \rightarrow 2}} (2x^2 + 4) = 12$$

```
>plot2d("2*x^2+4", -2, 4):
```



```
>&showev('limit((x^2-9),x,3))
          2
      limit  (x  - 9) = 0
      x -> 3
```

## Turunan Fungsi

Definisi turunan:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>&showev('limit(((x+h)^2-x^2)/h,h,0)) // turunan x^2
          lim_{h \rightarrow 0} (x+h)^2 - x^2/h = 2x \lim_{h \rightarrow 0} (x+h) - 2x = 2x
```

```
>p &= expand((x+h)^2-x^2)|simplify; $p // pembilang dijabarkan dan
disederhanakan
          2hx + h^2
```

```
>q &= ratsimp(p/h); $q // ekspresi yang akan dihitung limitnya
disederhanakan
          2x + h^2/x + h
```

```
>$limit(q,h,0) // nilai limit sebagai turunan
          2x^2
```

```
>&showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
          lim_{h \rightarrow 0} (x+h)^n - x^n/h = nx^{n-1} \lim_{h \rightarrow 0} (x+h)^{n-1} - x^{n-1} = nx^{n-1}
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut

benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan  $(x+h)^n$  dengan menggunakan teorema binomial.

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
lim_{h \rightarrow 0} \sin(x+h) - \sin(x) = \cos(x)
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan  $\sin(x+h)$  dengan menggunakan rumus jumlah dua sudut.

```
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
lim_{h \rightarrow 0} \log(x+h) - \log(x) = 1/x
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, gunakan sifat-sifat logaritma dan hasil limit pada bagian sebelumnya di atas.

```
>$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x
lim_{h \rightarrow 0} \frac{1}{x+h} - \frac{1}{x} = -\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:

$$\lim_{h \rightarrow 0} e^{x+h} - e^x$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
lim_{h \rightarrow 0} e^h - 1 = 1
```

```
>$showev('factor(E^(x+h)-E^x))
factor(e^{x+h} - e^x) = (e^h - 1)e^x
```

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
(lim_{h \rightarrow 0} e^h - 1)e^x = e^x (lim_{h \rightarrow 0} e^h - 1)e^x = e^x
```

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

$$\frac{x}{x}$$

```

>$showev('limit(f(x),x,0))
lim_{x \rightarrow 0} x = 1 lim_{x \rightarrow 0} x = 1

Silakan Anda gambar kurva
y=x.x.y=xx.

>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
lim_{h \rightarrow 0} (x+h)^{x+h} - x^x h = infinity lim_{h \rightarrow 0} (x+h)^{x+h} - x^x h = infinity

Di sini Maxima juga bermasalah terkait limit:
lim_{h \rightarrow 0} (x+h)^{x+h} - x^x h. lim_{h \rightarrow 0} (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} (x+h)^{x+h} - x^x h = x x (\log x + 1) lim_{h \rightarrow 0} (x+h)^{x+h} - x^x h = x x (\log x + 1)

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil
limit tersebut benar, sehingga benar turunan fungsinya benar.
Tulis penjelasan Anda di komentar ini.

>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
[x > 0]

>&forget(x<0)
[x < 0]

>&facts()
[]

>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
lim_{h \rightarrow 0} \arcsin(x+h) - \arcsin x h = 1 - x^2 ----- \sqrt{lim_{h \rightarrow 0} \arcsin(x+h) - \arcsin x h} = 1 - x^2

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil
limit tersebut benar, sehingga
benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
lim_{h \rightarrow 0} \tan(x+h) - \tan x h = 1 \cos^2 x lim_{h \rightarrow 0} \tan(x+h) - \tan x h = 1 \cos^2 x

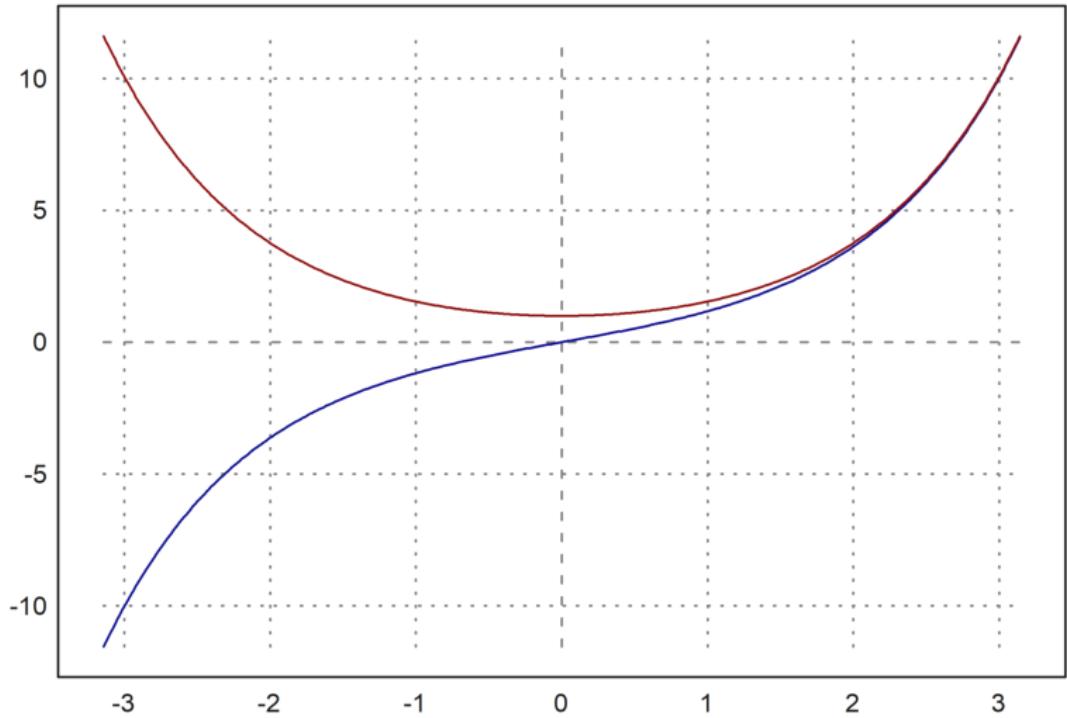
Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil
limit tersebut benar, sehingga
benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
                           sinh(x)

>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
                           e^{-x}(e^{2x}+1)2e^{-x}(e^{2x}+1)2

Hasilnya adalah cosh(x), karena
ex+e^{-x}2=cosh(x).ex+e^{-x}2=cosh(x).
```

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```



```

>function f(x) &= sin(3*x^5+7)^2
          2      5
          sin (3 x  + 7)

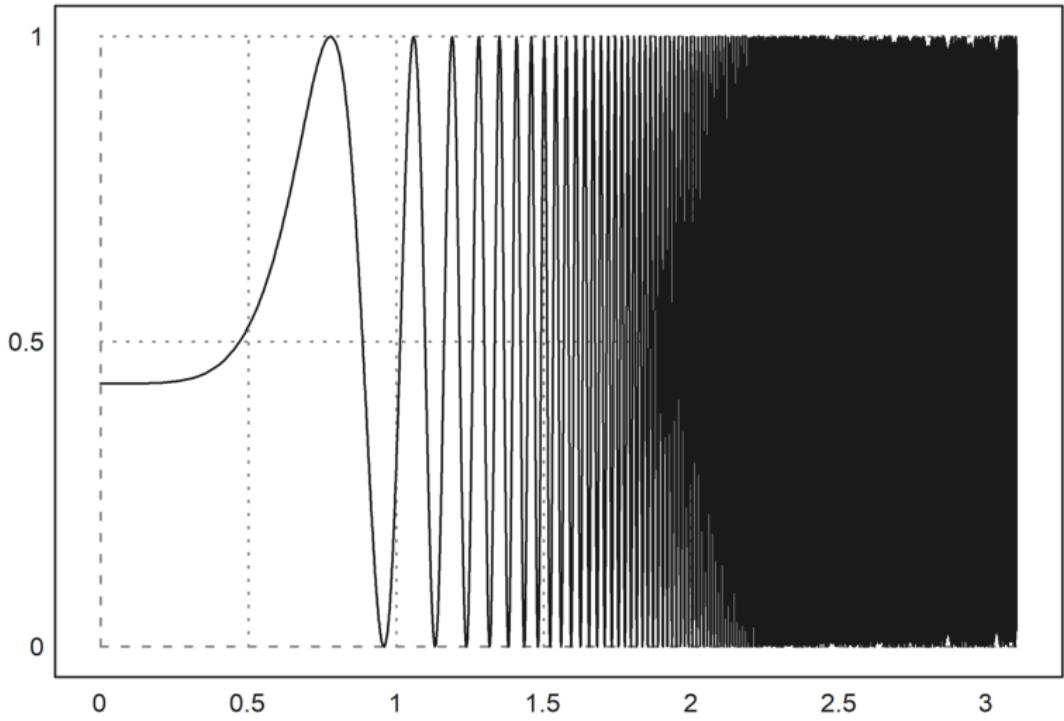
>diff(f,3), diffc(f,3)
1198.32948904
1198.72863721
Apakah perbedaan diff dan diffc?
>$showev('diff(f(x),x)) 
ddxsin2(3x5+7)=30x4cos(3x5+7)sin(3x5+7)ddxsin2[f](3x5+7)=30x4cos[f](3x5+7)sin[f](3x5+7)

>% with x=3
%at(ddxsin2(3x5+7),x=3)=2430cos736sin736%at(ddxsin2[f](3x5+7),x=3)=2430cos[f]736sin[f]736

>$float(%)
%at(d1.0dx1.0sin2(3.0x5+7.0),x=3.0)=1198.728637211748%at(d1.0dx1.0sin2[f](3.0x5+7.0),x=3.0)=1198.72
8637211748

>plot2d(f,0,3.1):

```



```
>function f(x) &=5*cos(2*x)-2*x*sin(2*x) // mendefinisikan fungsi f
      5 cos(2 x) - 2 x sin(2 x)
```

```
>function df(x) &=diff(f(x),x) // fd(x) = f'(x)
      - 12 sin(2 x) - 4 x cos(2 x)
```

```
>${'f(1)=f(1), $float(f(1)), '$f(2)=f(2), $float(f(2))} // nilai f(1) dan
f(2)
-0.2410081230863468-0.2410081230863468
```

$$-3.899329036387075$$

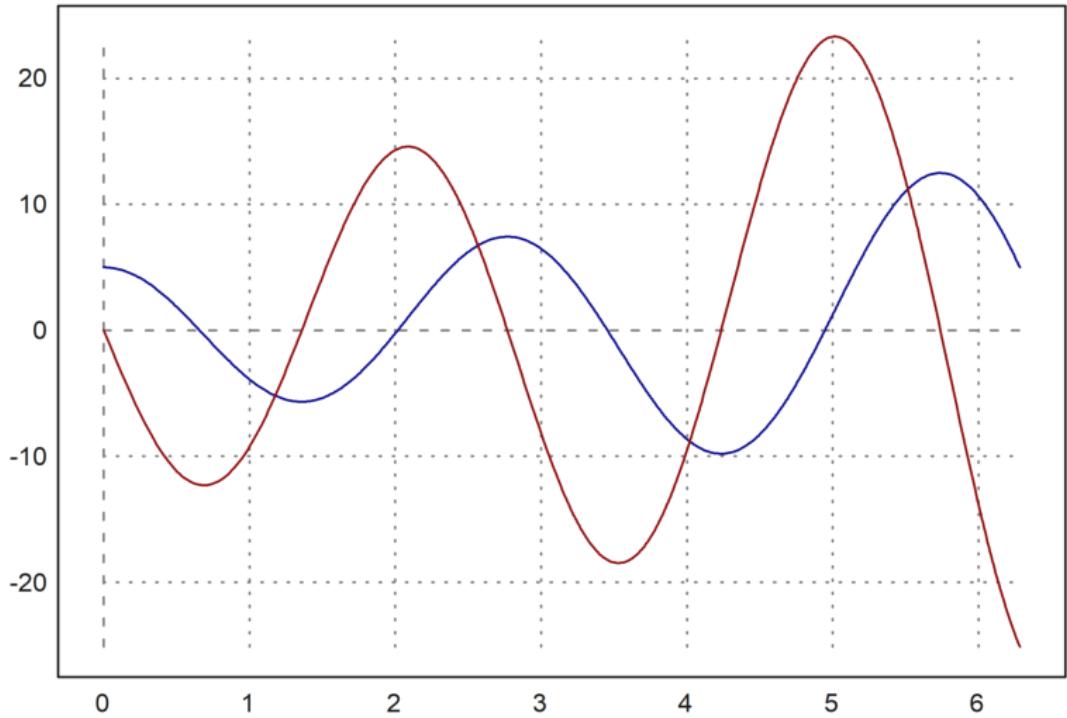
$$f(2) = 5 \cos 4 - 4 \sin 4$$

$$-0.2410081230863468$$

```
>xp=solve("df(x)",1,2,0) // solusi f'(x)=0 pada interval [1, 2]
1.35822987384
```

```
>df(xp), f(xp) // cek bahwa f'(xp)=0 dan nilai ekstrim di titik tersebut
0
-5.67530133759
```

```
>plot2d(["f(x)", "df(x)"], 0, 2*pi, color=[blue, red]): //grafik fungsi dan
turunannya
```



Perhatikan titik-titik "puncak" grafik  $y=f(x)$  dan nilai turunan pada saat grafik fungsinya mencapai titik "puncak" tersebut.

```
>&showev('limit(((1-cos*x)/x),x,0))
          1 - x cos
      limit  ----- = infinity
           x -> 0      x
```

## Latihan

---

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), menggunakan perintah diff, dan secara manual (langkah demi langkah yang dihitung dengan Maxima) seperti contoh-contoh di atas. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

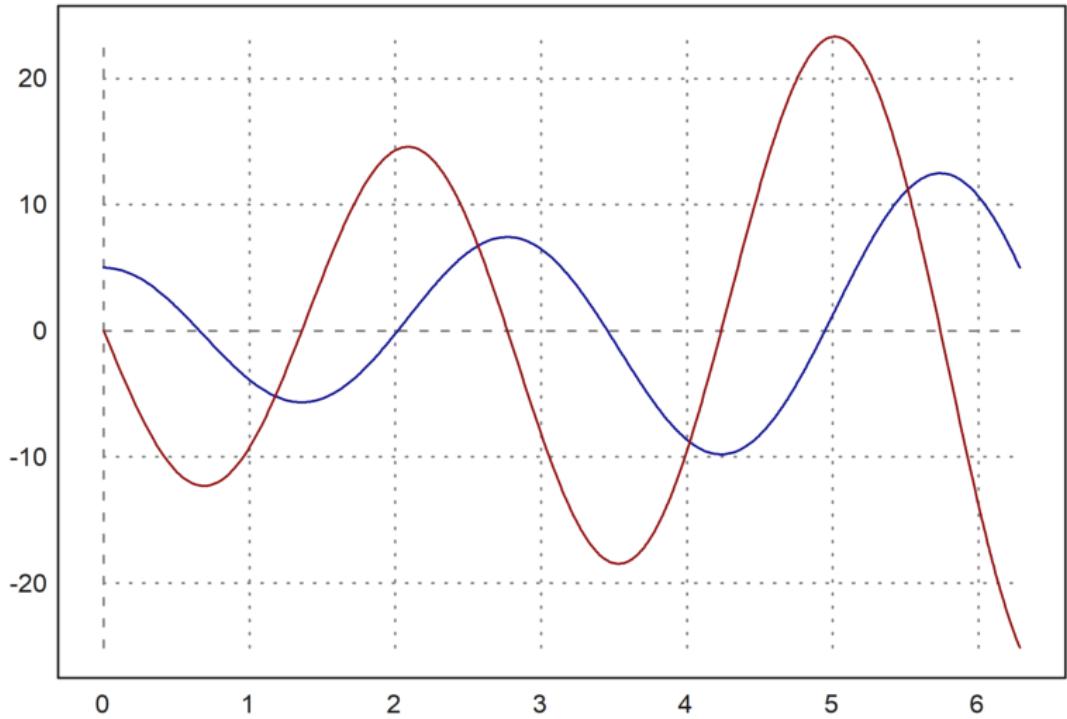
```
>&showev('limit(((1-cos*x)/x),x,0))
          1 - x cos
      limit  ----- = infinity
           x -> 0      x

>&showev('limit(((1-cos*2*x)/(1-cos*x)),x,0)):
>$showev('limit((tan(x+h)-tan(x))/h,h,0))
limh->0tan(x+h)-tanxh=1cos2xlimh->0tan[0](x+h)-tan[0]xh=1cos2[0]x

>$showev('limit(((x+h)^2+4*x^2)/h,h,2))
limh->2(x+h)2+4x2h=5x2+4x+42limh->2(x+h)2+4x2h=5x2+4x+42

>$showev('limit((1/(x+h)-1/x^2)/h,h,1))
limh->11x+h-1x2h=x2-x-1x3+x2limh->11x+h-1x2h=x2-x-1x3+x2
```

```
>plot2d(["f(x)", "df(x)"], 0, 2*pi, color=[blue, red]):
```



## 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:

$$\int_a^b f(x) dx = F(b) - F(a), \text{ dengan } F'(x) = f(x).$$

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"

$$\int x^n dx = x^{n+1} \frac{n+1}{n+1}$$

```
>$showev('integrate(1/(1+x),x))
```

$$\int \frac{1}{x+1} dx = \log|x+1|$$

```
>$showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2+1} dx = \arctan x$$

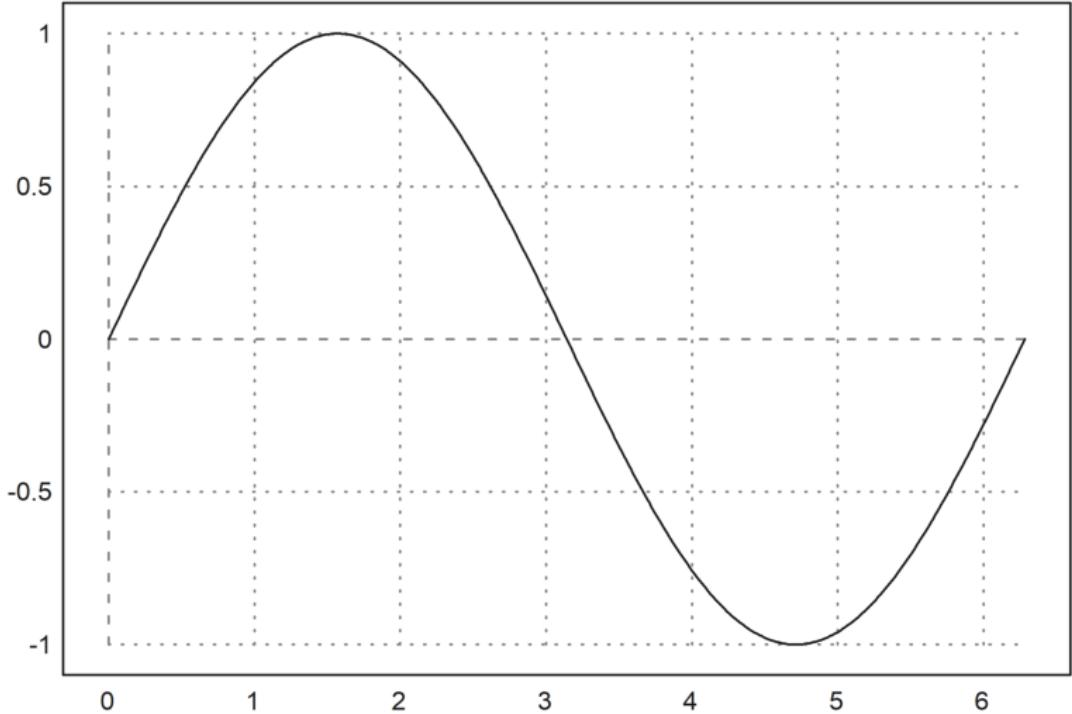
```

>$showev('integrate(1/sqrt(1-x^2),x))
          ∫1-x2----√dx=arcsinx[1-x2dx=arcsin[f0]x

>$showev('integrate(sin(x),x,0,pi))
          ∫π0sinxdx=2∫0πsin[f0]xdx=2

>plot2d("sin(x)",0,2*pi):

```



```

>$showev('integrate(sin(x),x,a,b))
          ∫basinxdx=cosa-cosb[absin[f0]xdx=cos[f0]a-cos[f0]b

>$showev('integrate(x^n,x,a,b))
      Answering "Is n positive, negative or zero?" with "positive"
          ∫baXndx=b[n+1]n+1-a[n+1]n+1[abxndx=bn+1n+1-an+1n+1

>$showev('integrate(x^2*sqrt(2*x+1),x))
          ∫x22x+1----√dx=(2x+1)7228-(2x+1)5210+(2x+1)3212∫x22x+1dx=(2x+1)7228-(2x+1)5210+(2x+1)3212

>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
          ∫20x22x+1----√dx=255221-2105∫02x22x+1dx=255221-2105

>$ratsimp(%)
          ∫20x22x+1----√dx=2572-2105∫02x22x+1dx=2572-2105

>$showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
          ∫π20sin(x--√+a)ex√x--√dx=(-eπ-1)sina+(eπ+1)cosa[0π2sin[f0](x+a)exxdx=(-eπ-1)sin[f0]a+(eπ+1)cos[f0]a

>$factor(%)
          ∫π20sin(x--√+a)ex√x--√dx=(-eπ-1)(sina-cosa)[0π2sin[f0](x+a)exxdx=(-eπ-1)(sin[f0]a-cos[f0]a)

>function map f(x) &= E^(-x^2)
          2
          - x
          E

```

```
>$showev('integrate(f(x),x))
 $\int e^{-x^2} dx = \pi - \sqrt{\pi} \operatorname{erf}(x) / 2$ 
```

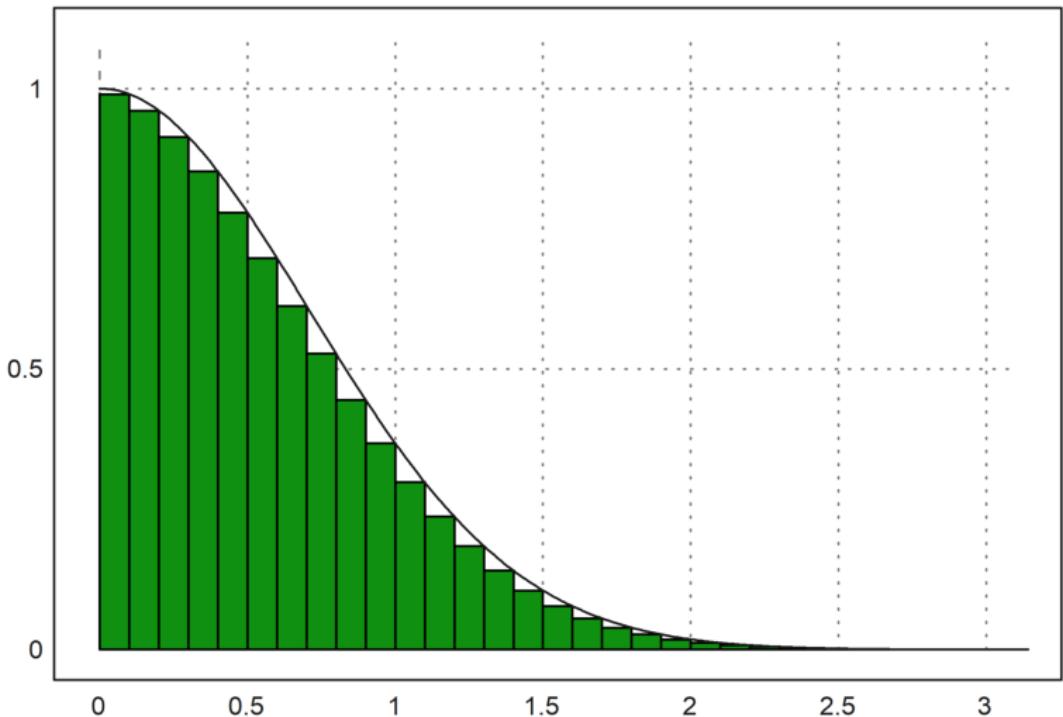
Fungsi  $f$  tidak memiliki antiturunan, integralnya masih memuat integral lain.

$$\operatorname{erf}(x) = \int e^{-x^2} \pi - \sqrt{\pi} dx. \operatorname{erf}(x) = \int e^{-x^2} \pi dx.$$

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

```
maxima: 'integrate(f(x),x,0,pi)
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```



Integral tentu

```
maxima: 'integrate(f(x),x,0,pi)
```

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva  $y=f(x)$  tersebut. Langkah-langkahnya adalah sebagai berikut.

```
>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-nilai x
>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)
// jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!
```

Hasilnya adalah:

```
maxima: 'integrate(f(x),x,0,pi) = 0.1*sum(fx[i],i,1,length(fx))
```

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1)

dan jumlah nilai-nilai  $f(x)$  untuk  $x = 0.1, 0.2, 0.3, \dots, 3.2$ .

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

0.836219610253

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \pi - \sqrt{2} \int_0^{\infty} e^{-x^2} dx = \pi/2$$

Tunjukkan kebenaran hasil di atas!

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

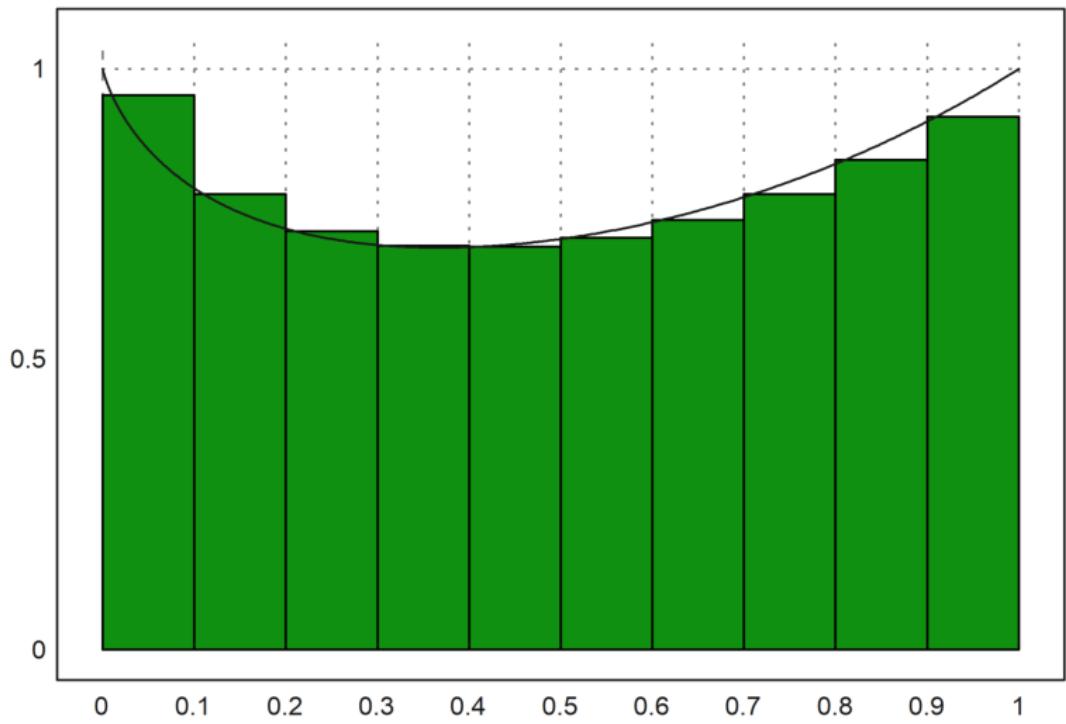
```
>function f(x) &= x^x
```

$$\frac{x}{x}$$

```
>$showev('integrate(f(x),x,0,1))
```

$$\int_0^1 x^x dx = \int_0^1 x^x dx \int_0^1 x^x dx = \int_0^1 x^x dx$$

```
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```



Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
maxima: 'integrate(f(x),x,0,1) = 0.01*sum(fx[i],i,1,length(fx))
```

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
>function f(x) &= sin(3*x^5+7)^2
                           2      5
                           sin (3 x  + 7)

>integrate(f,0,1)
0.542581176074
>&showev('integrate(f(x),x,0,1))
1
/
[      2      5           1          pi
I   sin (3 x  + 7) dx = -----
]                               5
/
0
4/5           1           4/5           1
- ((6      gamma_incomplete(-, 6 I) + 6      gamma_incomplete(-, - 6 I))
5
sin(14) + (6      I gamma_incomplete(-, 6 I)
5
4/5           1           pi
- 6      I gamma_incomplete(-, - 6 I)) cos(14) sin(--)- 60)/120
```

```

>&float(%)
      1.0
      /
      [
          2      5
          I sin (3.0 x + 7.0) dx =
      ]
      /
      0.0
0.09820784258795788 - 0.00833333333333333
(0.3090169943749474 (0.1367372182078336
(4.192962712629476 I gamma_incomplete(0.2, 6.0 I)
- 4.192962712629476 I gamma_incomplete(0.2, - 6.0 I))
+ 0.9906073556948704 (4.192962712629476 gamma_incomplete(0.2, 6.0 I)
+ 4.192962712629476 gamma_incomplete(0.2, - 6.0 I))) - 60.0

>$showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)
 $\int_0^1 xe^{-x} dx = 1 - 2e^{-1}$ 

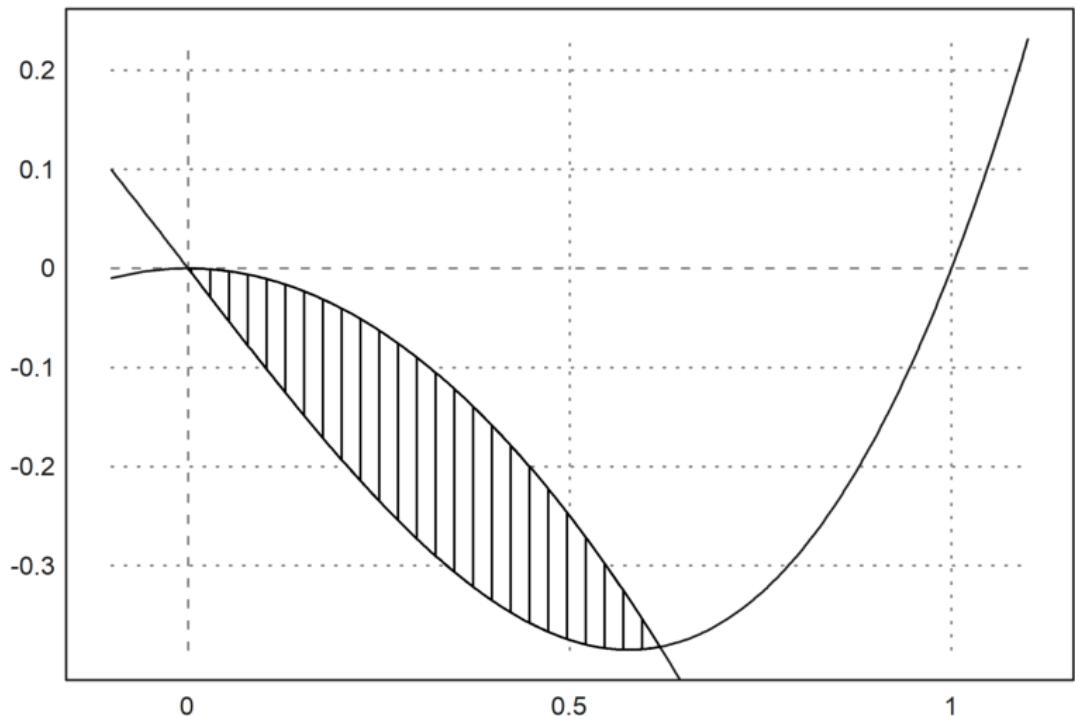
```

## Aplikasi Integral Tentu

```

>plot2d("x^3-x", -0.1, 1.1); plot2d("-x^2", >add); ...
b=solve("x^3-x+x^2", 0.5); x=linspace(0,b,200); xi=flipx(x); ...
plot2d(x|xi, x^3-x|-xi^2, >filled, style="|", fillcolor=1, >add); // Plot
daerah antara 2 kurva

```



```

>a=solve("x^3-x+x^2", 0), b=solve("x^3-x+x^2", 1) // absis titik-titik potong
kedua kurva
0
0.61803398875
>integrate("(-x^2)-(x^3-x)", a, b) // luas daerah yang diarsir
0.0758191713542

```

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```
>a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik potong kedua kurva secara eksak
```

$$[x=-5-\sqrt{-12}, x=5-\sqrt{-12}, x=0] [x=-5-12, x=5-12, x=0]$$

```
>$showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara eksak
```

$$\int_{5\sqrt{-12}}^{0} -x^3 - x^2 + x \, dx = 13 - 53224 \int_{0}^{5\sqrt{-12}} -x^3 - x^2 + x \, dx = 13 - 53224$$

```
>$float(%)
```

$$\int_{0.61803398874989490.0}^{1.0x3-1.0x2+x} 0.07581917135421037 \int_{0.0}^{0.6180339887498949-1.0x3-1.0x2+x} 0.07581917135421037$$

## Panjang Kurva

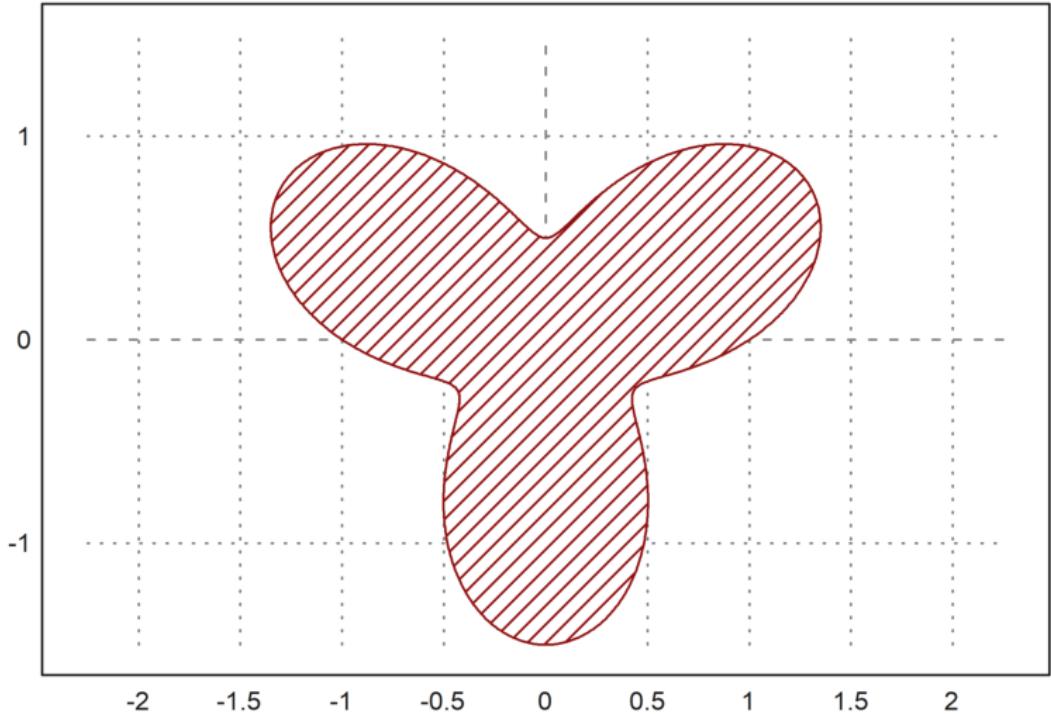
Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

$$\gamma(t) = [r(t)\cos(t), r(t)\sin(t)] \quad \gamma(t) = (r(t)\cos(t), r(t)\sin(t))$$

dengan

$$r(t) = 1 + \sin(3t) \quad 2,0 \leq t \leq 2\pi. \quad r(t) = 1 + \sin(3t) \quad 2,0 \leq t \leq 2\pi.$$

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5); // Kita gambar kurvanya terlebih dahulu
```



```
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
r([0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19,
0.2,0.21,0.2200000000000001,0.2300000000000001,0.2400000000000001,0.250000000000000001,0.2600000000000001,0.2700000000000001,0.2800000000000001,0.2900000000000001,0.3000000000000001,0.3100000000000001,0.3200000000000001,0.3300000000000001,0.3400000000000001,0.3500000000000001,0.3600000000000001,0.3700000000000001,0.3800000000000001,0.3900000000000001,0.4000000000000001,0.4100000000000001,0.4200000000000001,0.4300000000000001,0.4400000000000001,0.4500000000000001,0.4600000000000001,0.4700000000000001,0.4800000000000001,0.4900000000000001,0.5000000000000001,0.5100000000000001,0.5200000000000001,0.5300000000000001,0.5400000000000001,0.5500000000000001,0.5600000000000001,0.5700000000000001,0.5800000000000001,0.5900000000000001,0.6000000000000001,0.6100000000000001,0.6200000000000001,0.6300000000000001,0.6400000000000001,0.6500000000000001,0.6600000000000001,0.6700000000000001,0.6800000000000001,0.6900000000000001,0.7000000000000001,0.7100000000000001,0.7200000000000001,0.7300000000000001,0.7400000000000001,0.7500000000000001,0.7600000000000001,0.7700000000000001,0.7800000000000001,0.7900000000000001,0.8000000000000001,0.8100000000000001,0.8200000000000001,0.8300000000000001,0.8400000000000001,0.8500000000000001,0.8600000000000001,0.8700000000000001,0.8800000000000001,0.8900000000000001,0.9000000000000001,0.9100000000000001,0.9200000000000001,0.9300000000000001,0.9400000000000001,0.9500000000000001,0.9600000000000001,0.9700000000000001,0.9800000000000001,0.9900000000000001,1.0000000000000001])
```

4000000000000001, 0.3500000000000001, 0.3600000000000002, 0.3700000000000002, 0.3800  
 000000000002, 0.3900000000000002, 0.4000000000000002, 0.4100000000000002, 0.4200000  
 000000002, 0.4300000000000002, 0.4400000000000002, 0.4500000000000002, 0.460000000  
 000002, 0.4700000000000003, 0.4800000000000003, 0.4900000000000003, 0.5000000000000  
 002, 0.5100000000000002, 0.5200000000000002, 0.5300000000000002, 0.5400000000000003  
 , 0.5500000000000003, 0.5600000000000003, 0.5700000000000003, 0.5800000000000003, 0.5  
 90000000000003, 0.6000000000000003, 0.6100000000000003, 0.6200000000000003, 0.6300  
 000000000003, 0.6400000000000003, 0.6500000000000004, 0.6600000000000004, 0.6700000  
 00000004, 0.6800000000000004, 0.6900000000000004, 0.7000000000000004, 0.710000000  
 000004, 0.7200000000000004, 0.7300000000000004, 0.7400000000000004, 0.7500000000000  
 004, 0.7600000000000005, 0.7700000000000005, 0.7800000000000005, 0.7900000000000005  
 , 0.8000000000000005, 0.8100000000000005, 0.8200000000000005, 0.8300000000000005, 0.8  
 400000000000005, 0.8500000000000005, 0.8600000000000005, 0.8700000000000006, 0.8800  
 000000000006, 0.8900000000000006, 0.9000000000000006, 0.9100000000000006, 0.9200000  
 000000006, 0.9300000000000006, 0.9400000000000006, 0.9500000000000006, 0.96000000000  
 000006, 0.9700000000000006, 0.9800000000000006, 0.9900000000000007] = [1, 1.014997750  
 101248, 1.029982003239722, 1.044939274599006, 1.05985610364446, 1.0747190662368, 1.08  
 9514786712912, 1.10422994992305, 1.118851313213567, 1.133365718344415, 1.1477601033  
 3067, 1.162021514197434, 1.176137116637545, 1.190094207561581, 1.203880226529785, 1.2  
 17482767055615, 1.230889587770742, 1.244088623441454, 1.257067995826556, 1.26981602  
 4366985, 1.282321236697518, 1.294572378971135, 1.306558425986717, 1.318268591110984  
 , 1.329692335985737, 1.340819380011667, 1.351639709600205, 1.362143587185071, 1.37232  
 155998543, 1.382164468512753, 1.391663454813742, 1.400809970441889, 1.4095957841504  
 99, 1.41801298930026, 1.426054010974682, 1.433711612797009, 1.440978903442474, 1.4478  
 49342840024, 1.454316748057942, 1.460375298868068, 1.466019542983613, 1.47124440096  
 5849, 1.476045170795258, 1.480417532103036, 1.484357550059133, 1.48786167891333, 1.49  
 092676518618, 1.493550050506925, 1.495729174095843, 1.49746217488879, 1.49874749330  
 2027, 1.499583972635738, 1.499970860114983, 1.49997087567145, 1.499394871735262, 1.4  
 98432514226959, 1.497021601099038, 1.495163402078079, 1.492859589417777, 1.49011223  
 6394023, 1.486923815439098, 1.483297195916649, 1.479235641539457, 1.474742807432315  
 , 1.469822736842662, 1.464479857501934, 1.458718977640905, 1.4525452816626, 1.4459643  
 2547669, 1.438982031499539, 1.431604683324436, 1.423838920066784, 1.415691730389341  
 , 1.407170446212898, 1.398282736118043, 1.38903659844396, 1.379440354090461, 1.369502  
 639029735, 1.359232396534563, 1.348638869129968, 1.337731590275575, 1.3265203757861  
 32, 1.315015314997945, 1.303226761689157, 1.29116532476204, 1.278841858695708, 1.2662  
 6745377781, 1.253453426124026, 1.240411307494323, 1.227152834915152, 1.213689940116  
 914, 1.200034738796209, 1.186199519712527, 1.172196733629194, 1.158038982108526, 1.14  
 3739006171271, 1.129309674830555, 1.114763973510631, 1.100114992360884, 1.085375914  
 475572]  $r([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.22])$   
 000000000001, 0.2300000000000001, 0.2400000000000001, 0.2500000000000001, 0.2600000000000001, 0.270000000000000  
 01, 0.2800000000000001, 0.2900000000000001, 0.3000000000000001, 0.3100000000000001, 0.3200000000000001, 0.330000  
 0000000001, 0.3400000000000001, 0.3500000000000001, 0.3600000000000002, 0.3700000000000002, 0.3800000000000002  
 , 0.3900000000000002, 0.4000000000000002, 0.4100000000000002, 0.4200000000000002, 0.4300000000000002, 0.44000000  
 00000002, 0.4500000000000002, 0.4600000000000002, 0.4700000000000003, 0.4800000000000003, 0.4900000000000003, 0.  
 5000000000000002, 0.5100000000000002, 0.5200000000000002, 0.5300000000000002, 0.5400000000000003, 0.55000000000  
 000003, 0.5600000000000003, 0.5700000000000003, 0.5800000000000003, 0.5900000000000003, 0.6000000000000003, 0.61  
 00000000000003, 0.6200000000000003, 0.6300000000000003, 0.6400000000000003, 0.6500000000000004, 0.660000000000000  
 0004, 0.6700000000000004, 0.6800000000000004, 0.6900000000000004, 0.7000000000000004, 0.7100000000000004, 0.7200  
 000000000004, 0.7300000000000004, 0.7400000000000004, 0.7500000000000004, 0.7600000000000005, 0.770000000000000  
 05, 0.7800000000000005, 0.7900000000000005, 0.8000000000000005, 0.8100000000000005, 0.8200000000000005, 0.830000  
 0000000005, 0.8400000000000005, 0.8500000000000005, 0.8600000000000005, 0.8700000000000006, 0.8800000000000006  
 , 0.8900000000000006, 0.9000000000000006, 0.9100000000000006, 0.9200000000000006, 0.9300000000000006, 0.94000000  
 00000006, 0.9500000000000006, 0.9600000000000006, 0.9700000000000006, 0.9800000000000006, 0.9900000000000007] = [1, 1.014997750101248, 1.029982003239722, 1.044939274599006, 1.05985610364446, 1.0747190662368, 1.0895147867129  
 12, 1.10422994992305, 1.118851313213567, 1.133365718344415, 1.14776010333067, 1.162021514197434, 1.1761371166375  
 45, 1.190094207561581, 1.203880226529785, 1.217482767055615, 1.230889587770742, 1.244088623441454, 1.25706799582  
 6556, 1.269816024366985, 1.282321236697518, 1.294572378971135, 1.306558425986717, 1.318268591110984, 1.329692335

985737,1.340819380011667,1.351639709600205,1.362143587185071,1.37232155998543,1.382164468512753,1.39166345  
 4813742,1.400809970441889,1.409595784150499,1.4180129893026,1.426054010974682,1.433711612797009,1.4409789  
 03442474,1.447849342840024,1.454316748057942,1.460375298868068,1.466019542983613,1.471244400965849,1.47604  
 5170795258,1.480417532103036,1.484357550059133,1.48786167891333,1.49092676518618,1.493550050506925,1.49572  
 9174095843,1.49746217488879,1.498747493302027,1.499583972635738,1.499970860114983,1.499907807567145,1.4993  
 94871735262,1.498432514226959,1.497021601099038,1.495163402078079,1.492859589417777,1.490112236394023,1.48  
 6923815439098,1.483297195916649,1.479235641539457,1.474742807432315,1.469822736842662,1.464479857501934,1.  
 458718977640905,1.4525452816626,1.44596432547669,1.438982031499539,1.431604683324436,1.423838920066784,1.4  
 15691730389341,1.407170446212898,1.398282736118043,1.38903659844396,1.379440354090461,1.369502639029735,1.  
 359232396534563,1.348638869129968,1.337731590275575,1.326520375786132,1.315015314997945,1.303226761689157  
 ,1.29116532476204,1.278841858695708,1.26626745377781,1.253453426124026,1.240411307494323,1.227152834915152  
 ,1.213689940116914,1.200034738796209,1.186199519712527,1.172196733629194,1.158038982108526,1.1437390061712  
 71,1.129309674830555,1.114763973510631,1.100114992360884,1.085375914475572]

```

>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
fx([0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19
,0.2,0.21,0.22000000000000001,0.23000000000000001,0.24000000000000001,0.2500000000000
0001,0.26000000000000001,0.27000000000000001,0.28000000000000001,0.2900000000000000
1,0.30000000000000001,0.31000000000000001,0.32000000000000001,0.33000000000000001,0.
34000000000000001,0.35000000000000001,0.36000000000000002,0.37000000000000002,0.380
00000000000002,0.39000000000000002,0.40000000000000002,0.41000000000000002,0.420000
0000000002,0.43000000000000002,0.44000000000000002,0.45000000000000002,0.460000000
0000002,0.47000000000000003,0.48000000000000003,0.49000000000000003,0.500000000000000
0002,0.51000000000000002,0.52000000000000002,0.53000000000000002,0.5400000000000000
3,0.55000000000000003,0.56000000000000003,0.57000000000000003,0.58000000000000003,0.
59000000000000003,0.60000000000000003,0.61000000000000003,0.62000000000000003,0.630
0000000000003,0.64000000000000003,0.65000000000000004,0.66000000000000004,0.670000
000000004,0.68000000000000004,0.69000000000000004,0.70000000000000004,0.710000000
0000004,0.72000000000000004,0.73000000000000004,0.74000000000000004,0.750000000000000
0004,0.76000000000000005,0.77000000000000005,0.78000000000000005,0.7900000000000000
5,0.80000000000000005,0.81000000000000005,0.82000000000000005,0.83000000000000005,0.
84000000000000005,0.85000000000000005,0.86000000000000005,0.87000000000000006,0.880
0000000000006,0.89000000000000006,0.90000000000000006,0.91000000000000006,0.920000
0000000006,0.93000000000000006,0.94000000000000006,0.95000000000000006,0.960000000
0000006,0.97000000000000006,0.98000000000000006,0.99000000000000007])=[1,1.01494700
0636657,1.029776013705529,1.044469087191079,1.059008331806833,1.073375947255439
,1.087554248364218,1.10152569105367,1.11527289811021,1.128778684687222,1.142026
083553954,1.154998369993414,1.16767908634602,1.180052066148761,1.19210145783388
6,1.203811747950136,1.215167783870255,1.226154795949382,1.236758419099762,1.2469
64713748154,1.256760186143285,1.266131807981756,1.275067035321848,1.28355382675
5846,1.29158066081265,1.29913655256367,1.306211069406282,1.312794346000405,1.318
877098335118,1.324450636903608,1.329506878966172,1.334038359882425,1.3380382434
95345,1.341500331551311,1.344419072141793,1.346789567153917,1.348607578718725,1.
349869534647481,1.350572532848044,1.350714344714907,1.350293417488142,1.3493088
75578123,1.347760520854542,1.345648831899879,1.342974962229111,1.33974073747909
7,1.335948651572729,1.331601861864506,1.326704183275865,1.321260081430156,1.3152
74664798767,1.308753675871437,1.301703481365363,1.294131061489226,1.28604399827
9732,1.277450463029762,1.268359202828647,1.25877952623647,1.248721288115691,1.23
8194873644713,1.227211181539273,1.215781606508839,1.203918020976346,1.191632756
090801,1.17893858206338,1.165848687858719,1.152376660274093,1.138536462440146,1.
124342411777761,1.10980915744646,1.094951657320579,1.079785154530145,1.06432515
3604093,1.04858739625406,1.032587836837555,1.0163426175398,0.999868043313951,0.9
831805566197906,0.9662967120012925,0.9492331505436565,0.932006574250646,0.91463
37203831,0.897131335799599,0.8795161513401855,0.8618048562939812,0.844014072991
3906,0.8261603315613344,0.8082600448937051,0.7903294838468643,0.772384752739602
5,0.754441765166499,0.7365162201750889,0.7186235788426429,0.7007790412897039,0.6
829975241668103,0.6652936386500562,0.6476816689803099,0.6301755515800127,0.6127

```

888547805567,0.595534759192214]fx([0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1,0.11,0.12,0.13,0.14,0  
 .15,0.16,0.17,0.18,0.19,0.2,0.21,0.2200000000000001,0.2300000000000001,0.2400000000000001,0.2500000000000001,0.  
 2600000000000001,0.2700000000000001,0.2800000000000001,0.2900000000000001,0.3000000000000001,0.3100000000000001,0.  
 000001,0.3200000000000001,0.3300000000000001,0.3400000000000001,0.3500000000000001,0.3600000000000002,0.37  
 00000000000002,0.3800000000000002,0.3900000000000002,0.4000000000000002,0.4100000000000002,0.4200000000000000  
 0002,0.4300000000000002,0.4400000000000002,0.4500000000000002,0.4600000000000002,0.4700000000000003,0.4800  
 00000000000003,0.4900000000000003,0.5000000000000002,0.5100000000000002,0.5200000000000002,0.5300000000000000  
 02,0.5400000000000003,0.5500000000000003,0.5600000000000003,0.5700000000000003,0.5800000000000003,0.590000  
 0000000003,0.6000000000000003,0.6100000000000003,0.6200000000000003,0.6300000000000003,0.6400000000000003  
 ,0.6500000000000004,0.6600000000000004,0.6700000000000004,0.6800000000000004,0.6900000000000004,0.70000000  
 00000004,0.7100000000000004,0.7200000000000004,0.7300000000000004,0.7400000000000004,0.7500000000000004,0.  
 7600000000000005,0.7700000000000005,0.7800000000000005,0.7900000000000005,0.8000000000000005,0.8100000000000000  
 000005,0.8200000000000005,0.8300000000000005,0.8400000000000005,0.8500000000000005,0.8600000000000005,0.87  
 00000000000006,0.8800000000000006,0.8900000000000006,0.9000000000000006,0.9100000000000006,0.9200000000000000  
 0006,0.9300000000000006,0.9400000000000006,0.9500000000000006,0.9600000000000006,0.9700000000000006,0.9800  
 000000000006,0.9900000000000007])=[1,1.014947000636657,1.029776013705529,1.044469087191079,1.0590083318068  
 33,1.073375947255439,1.087554248364218,1.10152569105367,1.11527289811021,1.12877864687222,1.142026083553  
 954,1.154998369993414,1.16767908634602,1.180052066148761,1.192101457833886,1.203811747950136,1.21516778387  
 0255,1.226154795949382,1.236758419099762,1.246964713748154,1.256760186143285,1.266131807981756,1.275067035  
 321848,1.283553826755846,1.29158066081265,1.29913655256367,1.306211069406282,1.312794346000405,1.318877098  
 335118,1.324450636903608,1.329506878966172,1.334038359882425,1.338038243495345,1.341500331551311,1.3444190  
 72141793,1.346789567153917,1.348607578718725,1.349869534647481,1.350572532848044,1.350714344714907,1.35029  
 3417488142,1.349308875578123,1.347760520854542,1.345648831899879,1.342974962229111,1.339740737479097,1.335  
 948651572729,1.331601861864506,1.326704183275865,1.321260081430156,1.315274664798767,1.308753675871437,1.3  
 01703481365363,1.294131061489226,1.286043998279732,1.277450463029762,1.268359202828647,1.25877952623647,1.  
 248721288115691,1.238194873644713,1.227211181539273,1.215781606508839,1.203918020976346,1.191632756090801  
 ,1.17893858206338,1.165848687858719,1.152376660274093,1.138536462440146,1.124342411777761,1.10980915744646  
 ,1.094951657320579,1.079785154530145,1.064325153604093,1.04858739625406,1.032587836837555,1.0163426175398,  
 0.999868043313951,0.9831805566197906,0.9662967120012925,0.9492331505436565,0.932006574250646,0.9146337203  
 831,0.897131335799599,0.8795161513401855,0.8618048562939812,0.8440140729913906,0.8261603315613344,0.808260  
 0448937051,0.7903294838468643,0.7723847527396025,0.754441765166499,0.7365162201750889,0.7186235788426429,  
 0.7007790412897039,0.6829975241668103,0.6652936386500562,0.6476816689803099,0.6301755515800127,0.61278885  
 47805567,0.595534759192214]

```

>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
fy([0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19
,0.2,0.21,0.2200000000000001,0.2300000000000001,0.2400000000000001,0.2500000000000000
0001,0.2600000000000001,0.2700000000000001,0.2800000000000001,0.2900000000000000
1,0.3000000000000001,0.3100000000000001,0.3200000000000001,0.3300000000000001,0.
3400000000000001,0.3500000000000001,0.3600000000000002,0.3700000000000002,0.380
000000000002,0.3900000000000002,0.4000000000000002,0.4100000000000002,0.420000
0000000002,0.4300000000000002,0.4400000000000002,0.4500000000000002,0.460000000000000
000002,0.4700000000000003,0.4800000000000003,0.4900000000000003,0.5000000000000000
0002,0.5100000000000002,0.5200000000000002,0.5300000000000002,0.5400000000000000
3,0.5500000000000003,0.5600000000000003,0.5700000000000003,0.5800000000000003,0.
5900000000000003,0.6000000000000003,0.6100000000000003,0.6200000000000003,0.630
000000000003,0.6400000000000003,0.6500000000000004,0.6600000000000004,0.670000000000000
0004,0.6800000000000004,0.6900000000000004,0.7000000000000004,0.710000000000000
0004,0.7200000000000004,0.7300000000000004,0.7400000000000004,0.750000000000000
0004,0.7600000000000005,0.7700000000000005,0.7800000000000005,0.790000000000000
5,0.8000000000000005,0.8100000000000005,0.8200000000000005,0.8300000000000005,0.
8400000000000005,0.8500000000000005,0.8600000000000005,0.8700000000000006,0.880
000000000006,0.8900000000000006,0.9000000000000006,0.9100000000000006,0.920000
0000000006,0.9300000000000006,0.9400000000000006,0.9500000000000006,0.960000000000000
0006,0.9700000000000006,0.9800000000000006,0.9900000000000007])=[0,0.01014980
833556662,0.02059826678292271,0.03134347622283015,0.04238293991838228,0.0537135
6612987439,0.06533167172990376,0.07723298681299934,0.08941266029246918,0.101865
2664755576,0.1145848126064173,0.1275647473648353,0.1407979703071057,0.154276842
2339107,0.1679931964685752,0.1819383510275811,0.1961031216637831,0.210477835761
3507,0.2250523470600841,0.2398160511854019,0.2547579019589912,0.269866428463849

```

7,0.2851297528362152,0.3005356087557041,0.3160713606038417,0.3317240232600813,0.  
 3474802825033731,0.3633265159863522,0.3792488147482899,0.3952330052320643,0.411  
 264671769591,0.4273291794993832,0.4434116976792021,0.4594972233561165,0.4755706  
 053556919,0.4916165685515136,0.5076197383757777,0.5235646655312819,0.5394358508  
 648145,0.5552177703616642,0.5708949002207642,0.5864517419698421,0.6018728475798  
 654,0.6171428445380648,0.6322464608388652,0.6471685498521687,0.6618941150286309  
 ,0.6764083344018014,0.6906965848473219,0.704744466059751,0.7185378242080237,0.73  
 20627752310482,0.7453057277355214,0.7582534054586558,0.7708928692592016,0.78321  
 15386008901,0.7951972124932317,0.8068380898554457,0.8181227892702304,0.82904036  
 80950348,0.8395803408995157,0.8497326971989371,0.8594879184543822,0.86883699431  
 18147,0.877771438053233,0.8862833012344233,0.894365187485098,0.9020102654485477  
 ,0.9092122808393135,0.91596556759876,0.9222650581299157,0.9281062925943645,0.933  
 4854272555032,0.9383992418539865,0.9428451460027243,0.9468211845903713,0.950326  
 0421838114,0.9533590464217597,0.9559201703932094,0.9580100339960551,0.959629904  
 2728891,0.9607816947225576,0.9614679635877484,0.9616919111204768,0.961457375828  
 9937,0.9607688297112769,0.9596313724818526,0.9580507248003547,0.956033220511779  
 6,0.9535857979100135,0.950715990037748,0.9474319140374602,0.9437422595696462,0.9  
 396562763159917,0.9351837605866338,0.9303350410521015,0.9251209636219332,0.9195  
 528754933222,0.9136426083945087,0.9074024610488752]fy([0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.0  
 8,0.09,0.1,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19,0.2,0.21,0.2200000000000001,0.2300000000000001,0.24000000000  
 00001,0.2500000000000001,0.2600000000000001,0.2700000000000001,0.2800000000000001,0.2900000000000001,0.300  
 000000000001,0.3100000000000001,0.3200000000000001,0.3300000000000001,0.3400000000000001,0.350000000000000  
 001,0.3600000000000002,0.3700000000000002,0.3800000000000002,0.3900000000000002,0.4000000000000002,0.41000  
 0000000002,0.4200000000000002,0.4300000000000002,0.4400000000000002,0.4500000000000002,0.4600000000000000  
 2,0.4700000000000003,0.4800000000000003,0.4900000000000003,0.5000000000000002,0.5100000000000002,0.5200000  
 000000002,0.5300000000000002,0.5400000000000003,0.5500000000000003,0.5600000000000003,0.5700000000000003,  
 0.5800000000000003,0.5900000000000003,0.6000000000000003,0.6100000000000003,0.6200000000000003,0.63000000  
 00000003,0.6400000000000003,0.6500000000000004,0.6600000000000004,0.6700000000000004,0.6800000000000004,0.  
 6900000000000004,0.7000000000000004,0.7100000000000004,0.7200000000000004,0.7300000000000004,0.74000000000  
 000004,0.7500000000000004,0.7600000000000005,0.7700000000000005,0.7800000000000005,0.7900000000000005,0.80  
 00000000000005,0.8100000000000005,0.8200000000000005,0.8300000000000005,0.8400000000000005,0.8500000000000  
 0005,0.8600000000000005,0.8700000000000006,0.8800000000000006,0.8900000000000006,0.9000000000000006,0.9100  
 000000000006,0.9200000000000006,0.9300000000000006,0.9400000000000006,0.9500000000000006,0.960000000000000  
 06,0.9700000000000006,0.9800000000000006,0.9900000000000007]=-0,0.0101498033556662,0.02059826678292271,0.  
 03134347622283015,0.04238293991838228,0.05371356612987439,0.06533167172990376,0.07723298681299934,0.08941  
 266029246918,0.1018652664755576,0.1145848126064173,0.1275647473648353,0.1407979703071057,0.15427684223391  
 07,0.1679931964685752,0.1819383510275811,0.1961031216637831,0.2104778357613507,0.2250523470600841,0.239816  
 0511854019,0.2547579019589912,0.2698664284638497,0.2851297528362152,0.3005356087557041,0.3160713606038417  
 ,0.3317240232600813,0.3474802825033731,0.3633265159863522,0.3792488147482899,0.3952330052320643,0.41126467  
 1769591,0.4273291794993832,0.4434116976792021,0.4594972233561165,0.4755706053556919,0.4916165685515136,0.5  
 07619738375777,0.5235646655312819,0.5394358508648145,0.5552177703616642,0.5708949002207642,0.58645174196  
 98421,0.6018728475798654,0.6171428445380648,0.6322464608388652,0.6471685498521687,0.6618941150286309,0.676  
 4083344018014,0.6906965848473219,0.70474446059751,0.7185378242080237,0.7320627752310482,0.74530572773552  
 14,0.7582534054586558,0.7708928692592016,0.7832115386008901,0.7951972124932317,0.8068380898554457,0.818122  
 7892702304,0.8290403680950348,0.8395803408995157,0.8497326971989371,0.8594879184543822,0.8688369943118147  
 ,0.877771438053233,0.8862833012344233,0.894365187485098,0.9020102654485477,0.9092122808393135,0.9159655675  
 9876,0.9222650581299157,0.9281062925943645,0.9334854272555032,0.9383992418539865,0.9428451460027243,0.9468  
 211845903713,0.9503260421838114,0.9533590464217597,0.9559201703932094,0.9580100339960551,0.95962990427288  
 91,0.9607816947225576,0.9614679635877484,0.9616919111204768,0.9614573758289937,0.9607688297112769,0.959631  
 3724818526,0.9580507248003547,0.9560332205117796,0.9535857979100135,0.950715990037748,0.9474319140374602,  
 0.9437422595696462,0.9396562763159917,0.9351837605866338,0.9303350410521015,0.9251209636219332,0.91955287  
 54933222,0.9136426083945087,0.9074024610488752]

```

>function ds(t) &=
trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t)
  Maxima said:
  diff: second argument must be a variable; found errexpl
  -- an error. To debug this try: debugmode(true);

  Error in:
  ... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t ...
  ^

```

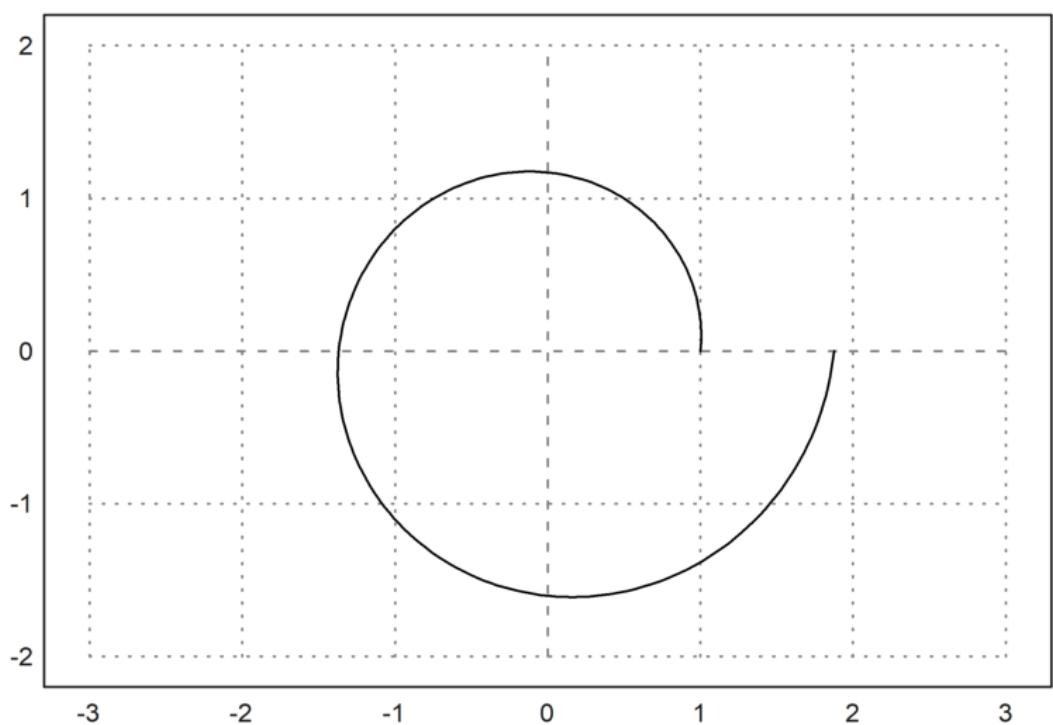
```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
       $\int_{2\pi}^0 ds(x) dx / 2\pi ds(x) dx$ 
```

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara numerik dengan perintah EMT.

```
>integrate("ds(x)",0,2*pi)
Function ds not found.
Try list ... to find functions!
Error in expression: ds(x)
%mapexpression1:
    return expr(x,args());
Error in map.
%evalexpression:
    if maps then return %mapexpression1(x,f$;args());
gauss:
    if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());
adaptivegauss:
    t1=gauss(f$,c,c+h;args(),=maps);
Try "trace errors" to inspect local variables after errors.
integrate:
    return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
Spiral Logaritmik
x=eaxcosx, y=eaxsinx.x=eaxcos[0]x, y=eaxsin[0]x.
```

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi):
```



```
>&kill(a) // hapus expresi a
```

done

```
>function fx(t) &= exp(a*t)*cos(t); '$fx(t)=fx(t)
fx([0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19
,0.2,0.21,0.2200000000000001,0.23000000000000001,0.24000000000000001,0.25000000000000001]
```

0001,0.2600000000000001,0.2700000000000001,0.2800000000000001,0.290000000000000  
1,0.3000000000000001,0.3100000000000001,0.3200000000000001,0.3300000000000001,0.  
3400000000000001,0.3500000000000001,0.3600000000000002,0.3700000000000002,0.380  
00000000000002,0.3900000000000002,0.4000000000000002,0.4100000000000002,0.420000  
000000000002,0.4300000000000002,0.4400000000000002,0.4500000000000002,0.46000000  
000000000002,0.4700000000000003,0.4800000000000003,0.4900000000000003,0.500000000000  
0002,0.5100000000000002,0.5200000000000002,0.5300000000000002,0.5400000000000000  
3,0.5500000000000003,0.5600000000000003,0.5700000000000003,0.5800000000000003,0.  
5900000000000003,0.6000000000000003,0.6100000000000003,0.6200000000000003,0.630  
00000000000003,0.6400000000000003,0.6500000000000004,0.6600000000000004,0.670000  
000000000004,0.6800000000000004,0.6900000000000004,0.7000000000000004,0.710000000  
00000004,0.7200000000000004,0.7300000000000004,0.7400000000000004,0.7500000000000  
0004,0.7600000000000005,0.7700000000000005,0.7800000000000005,0.7900000000000000  
5,0.8000000000000005,0.8100000000000005,0.8200000000000005,0.8300000000000005,0.  
8400000000000005,0.8500000000000005,0.8600000000000005,0.8700000000000006,0.880  
0000000000006,0.8900000000000006,0.9000000000000006,0.9100000000000006,0.920000  
0000000006,0.9300000000000006,0.9400000000000006,0.9500000000000006,0.96000000  
0000006,0.9700000000000006,0.9800000000000006,0.9900000000000007])=[1,0.99995000  
04166653e0.01a,0.9998000066665778e0.02a,0.9995500337489875e0.03a,0.9992001066609779e  
0.04a,0.9987502603949663e0.05a,0.9982005399352042e0.06a,0.9975510002532796e0.07a,0.9968  
017063026194e0.08a,0.9959527330119943e0.09a,0.9950041652780258e0.1a,0.99395609795669  
68e0.11a,0.9928086358538663e0.12a,0.9915618937147881e0.13a,0.9902159962126372e0.14a,0.98  
87710779360422e0.15a,0.9872272833756269e0.16a,0.9855847669095608e0.17a,0.98384369278  
81214e0.18a,0.9820042351172703e0.19a,0.9800665778412416e0.2a,0.9780309147241483e0.21a,0  
.9758974493306055e0.2200000000000001a,0.9736663950053748e0.2300000000000001a,0.97133797485  
20296e0.2400000000000001a,0.9689124217106447e0.2500000000000001a,0.9663899781345132e0.2600000  
000000001a,0.9637708963658905e0.2700000000000001a,0.9610554383107709e0.280000000000001a,0.958  
2438755126972e0.2900000000000001a,0.955336489125606e0.3000000000000001a,0.952333569885713  
4e0.3100000000000001a,0.9492354180824408e0.3200000000000001a,0.9460423435283869e0.3300000000000  
001a,0.9427546655283462e0.3400000000000001a,0.9393727128473789e0.3500000000000001a,0.9358968  
236779348e0.3600000000000002a,0.9323273456060344e0.3700000000000002a,0.9286646355765101e0.3  
800000000000002a,0.924909059857313e0.3900000000000002a,0.92106094002885e0.400000000000002a,0.9  
17120822816605e0.4100000000000002a,0.9130889403123081e0.4200000000000002a,0.9089657496748  
851e0.4300000000000002a,0.9047516632199634e0.4400000000000002a,0.9004471023526768e0.45000000000  
000002a,0.8960524975255252e0.4600000000000002a,0.8915682881953289e0.4700000000000003a,0.88699  
4922779284e0.4800000000000003a,0.8823328586101213e0.4900000000000003a,0.8775825618903726e0.  
5000000000000002a,0.8727445076457512e0.5100000000000002a,0.8678191796776498e0.5200000000000002a,  
0.8628070705147609e0.5300000000000002a,0.857708681363824e0.5400000000000003a,0.85252452205  
95056e0.5500000000000003a,0.847255111013416e0.5600000000000003a,0.8419009751622686e0.570000000  
0000003a,0.8364626499151868e0.5800000000000003a,0.8309406791001633e0.5900000000000003a,0.8253  
356149096781e0.6000000000000003a,0.8196480178454794e0.6100000000000003a,0.813878456662533  
8e0.6200000000000003a,0.8080275083121516e0.6300000000000003a,0.8020957578842924e0.6400000000000  
003a,0.7960837985490556e0.6500000000000004a,0.7899922314973649e0.6600000000000004a,0.7838216  
65880849e0.6700000000000004a,0.7775727187509277e0.6800000000000004a,0.7712460149971063e0.690  
000000000004a,0.7648421872844882e0.7000000000000004a,0.7583618759905079e0.7100000000000004a,0.  
7518057291408947e0.7200000000000004a,0.7451744023448701e0.7300000000000004a,0.73846855872  
95876e0.7400000000000004a,0.7316888688738206e0.7500000000000004a,0.7248360107409049e0.7600000  
000000005a,0.7179106696109431e0.7700000000000005a,0.7109135380122771e0.7800000000000005a,0.703  
8453156522357e0.7900000000000005a,0.696706709347165e0.8000000000000005a,0.689498432951746  
6e0.8100000000000005a,0.6822212072876132e0.8200000000000005a,0.6748757600712667e0.8300000000000  
005a,0.6674628258413078e0.8400000000000005a,0.6599831458849817e0.8500000000000005a,0.6524374  
681640515e0.8600000000000005a,0.6448265472400008e0.8700000000000006a,0.6371511441985798e0.8  
80000000000006a,0.6294120265736964e0.8900000000000006a,0.6216099682706641e0.9000000000000006a,  
0.613745749488811e0.9100000000000006a,0.6058201566434623e0.9200000000000006a,0.5978339822

$872978e0.9300000000000006a, 0.5897880250310977e0.9400000000000006a, 0.581683089463883e0.9500000$   
 $000000006a, 0.5735199860724561e0.9600000000000006a, 0.5652995311603538e0.9700000000000006a, 0.557$   
 $0225467662168e0.9800000000000006a, 0.548689860581587e0.9900000000000007a]fx([0,0.01,0.02,0.03,0.04,0.0$   
 $5,0.06,0.07,0.08,0.09,0.1,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19,0.2,0.21,0.2200000000000001,0.2300000000000001,$   
 $0.2400000000000001,0.2500000000000001,0.2600000000000001,0.2700000000000001,0.2800000000000001,0.2900000000000001,$   
 $0.3000000000000001,0.3100000000000001,0.3200000000000001,0.3300000000000001,0.3400000000000001,0.$   
 $3500000000000001,0.3600000000000002,0.3700000000000002,0.3800000000000002,0.3900000000000002,0.4000000000000000$   
 $000002,0.4100000000000002,0.4200000000000002,0.4300000000000002,0.4400000000000002,0.4500000000000002,0.46$   
 $00000000000002,0.4700000000000003,0.4800000000000003,0.4900000000000003,0.5000000000000002,0.5100000000000000$   
 $0002,0.5200000000000002,0.5300000000000002,0.5400000000000003,0.5500000000000003,0.5600000000000003,0.5700$   
 $000000000003,0.5800000000000003,0.5900000000000003,0.6000000000000003,0.6100000000000003,0.6200000000000000$   
 $03,0.6300000000000003,0.6400000000000003,0.6500000000000004,0.6600000000000004,0.6700000000000004,0.680000$   
 $0000000004,0.6900000000000004,0.7000000000000004,0.7100000000000004,0.7200000000000004,0.7300000000000004$   
 $,0.7400000000000004,0.7500000000000004,0.7600000000000005,0.7700000000000005,0.7800000000000005,0.7900000000000005,0.$   
 $8000000000000005,0.8100000000000005,0.8200000000000005,0.8300000000000005,0.8400000000000005,0.$   
 $8500000000000005,0.8600000000000005,0.8700000000000006,0.8800000000000006,0.8900000000000006,0.9000000000000000$   
 $000006,0.9100000000000006,0.9200000000000006,0.9300000000000006,0.9400000000000006,0.9500000000000006,0.96$   
 $00000000000006,0.9700000000000006,0.9800000000000006,0.9900000000000007]= [1,0.9999500004166653e0.01a,0.99$   
 $98000066665778e0.02a,0.9995500337489875e0.03a,0.9992001066609779e0.04a,0.9987502603949663e0.05a,0.998200539$   
 $9352042e0.06a,0.997551002532796e0.07a,0.9968017063026194e0.08a,0.9959527330119943e0.09a,0.9950041652780258$   
 $e0.1a,0.9939560979566968e0.11a,0.9928086358538663e0.12a,0.9915618937147881e0.13a,0.9902159962126372e0.14a,0.$   
 $9887710779360422e0.15a,0.9872272833756269e0.16a,0.9855847669095608e0.17a,0.9838436927881214e0.18a,0.9820042$   
 $351172703e0.19a,0.9800665778412416e0.2a,0.9780309147241483e0.21a,0.9758974493306055e0.2200000000000001a,0.9$   
 $736663950053748e0.2300000000000001a,0.9713379748520296e0.2400000000000001a,0.9689124217106447e0.25000000$   
 $00000001a,0.9663899781345132e0.2600000000000001a,0.9637708963658905e0.2700000000000001a,0.96105543831077$   
 $09e0.2800000000000001a,0.9582438755126972e0.2900000000000001a,0.955336489125606e0.3000000000000001a,0.952$   
 $3335698857134e0.3100000000000001a,0.9492354180824408e0.3200000000000001a,0.9460423435283869e0.3300000000000000$   
 $000001a,0.9427546655283462e0.3400000000000001a,0.9393727128473789e0.3500000000000001a,0.9358968236779348$   
 $e0.3600000000000002a,0.9323273456060344e0.3700000000000002a,0.9286646355765101e0.3800000000000002a,0.9249$   
 $09059857313e0.3900000000000002a,0.921060994002885e0.4000000000000002a,0.917120822816605e0.4100000000000000$   
 $02a,0.9130889403123081e0.4200000000000002a,0.9089657496748851e0.4300000000000002a,0.9047516632199634e0.44$   
 $000000000000002a,0.9004471023526768e0.4500000000000002a,0.8960524975255252e0.4600000000000002a,0.89156828$   
 $81953289e0.4700000000000003a,0.886994922779284e0.4800000000000003a,0.8823328586101213e0.4900000000000003$   
 $a,0.8775825618903726e0.5000000000000002a,0.8727445076457512e0.5100000000000002a,0.8678191796776498e0.5200$   
 $0000000000002a,0.8628070705147609e0.5300000000000002a,0.857708681363824e0.5400000000000003a,0.85252452205$   
 $95056e0.5500000000000003a,0.84725511013416e0.5600000000000003a,0.8419009751622686e0.5700000000000003a,0.$   
 $8364626499151868e0.5800000000000003a,0.8309406791001633e0.5900000000000003a,0.8253356149096781e0.6000000$   
 $00000003a,0.8196480178454794e0.6100000000000003a,0.8138784566625338e0.6200000000000003a,0.8080275083121$   
 $516e0.6300000000000003a,0.8020957578842924e0.6400000000000003a,0.7960837985490556e0.6500000000000004a,0.7$   
 $899922314973649e0.6600000000000004a,0.783821665880849e0.6700000000000004a,0.7775727187509277e0.68000000$   
 $0000004a,0.7712460149971063e0.6900000000000004a,0.7648421872844882e0.7000000000000004a,0.758361875990507$   
 $9e0.7100000000000004a,0.7518057291408947e0.7200000000000004a,0.7451744023448701e0.7300000000000004a,0.738$   
 $4685587295876e0.7400000000000004a,0.7316888688738206e0.7500000000000004a,0.7248360107409049e0.760000000000000$   
 $000005a,0.7179106696109431e0.7700000000000005a,0.7109135380122771e0.7800000000000005a,0.7038453156522357$   
 $e0.7900000000000005a,0.696706709347165e0.8000000000000005a,0.6894984329517466e0.8100000000000005a,0.6822$   
 $12072876132e0.8200000000000005a,0.674875760712667e0.8300000000000005a,0.6674628258413078e0.840000000000000$   
 $0005a,0.6599831458849817e0.8500000000000005a,0.6524374681640515e0.8600000000000005a,0.644826547240008e0.$   
 $8700000000000006a,0.6371511441985798e0.8800000000000006a,0.6294120265736964e0.8900000000000006a,0.621609$   
 $9682706641e0.9000000000000006a,0.6137457494888111e0.9100000000000006a,0.6058201566434623e0.920000000000000$   
 $006a,0.5978339822872978e0.9300000000000006a,0.5897880250310977e0.9400000000000006a,0.581683089463883e0.95$   
 $00000000000006a,0.5735199860724561e0.9600000000000006a,0.5652995311603538e0.9700000000000006a,0.55702254$   
 $67662168e0.9800000000000006a,0.548689860581587e0.9900000000000007a]$

```

>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)

fy([0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19
,0.2,0.21,0.2200000000000001,0.2300000000000001,0.2400000000000001,0.2500000000000001
0001,0.2600000000000001,0.2700000000000001,0.2800000000000001,0.2900000000000001
1,0.3000000000000001,0.3100000000000001,0.3200000000000001,0.3300000000000001,0.
3400000000000001,0.3500000000000001,0.3600000000000001,0.3700000000000001,0.380
0000000000002,0.3900000000000002,0.4000000000000002,0.4100000000000002,0.4200000
000000002,0.4300000000000002,0.4400000000000002,0.4500000000000002,0.460000000000000
0002,0.4700000000000003,0.4800000000000003,0.4900000000000003,0.5000000000000003
0002,0.5100000000000002,0.5200000000000002,0.5300000000000002,0.5400000000000002
3,0.5500000000000003,0.5600000000000003,0.5700000000000003,0.5800000000000003.

```

5900000000000003, 0.6000000000000003, 0.6100000000000003, 0.6200000000000003, 0.630  
 00000000000003, 0.6400000000000003, 0.6500000000000004, 0.6600000000000004, 0.670000  
 0000000004, 0.6800000000000004, 0.6900000000000004, 0.7000000000000004, 0.71000000  
 00000004, 0.7200000000000004, 0.7300000000000004, 0.7400000000000004, 0.750000000000  
 0004, 0.7600000000000005, 0.7700000000000005, 0.7800000000000005, 0.7900000000000005,  
 5, 0.8000000000000005, 0.8100000000000005, 0.8200000000000005, 0.8300000000000005, 0.  
 8400000000000005, 0.8500000000000005, 0.8600000000000005, 0.8700000000000006, 0.880  
 00000000000006, 0.8900000000000006, 0.9000000000000006, 0.9100000000000006, 0.920000  
 000000000006, 0.9300000000000006, 0.9400000000000006, 0.9500000000000006, 0.960000000  
 0000006, 0.9700000000000006, 0.9800000000000006, 0.9900000000000007])=[0,0.0099983  
 3334166664e0.01a, 0.0199986666933308e0.02a, 0.02999550020249566e0.03a, 0.0399893341866  
 3416e0.04a, 0.04997916927067833e0.05a, 0.0599640064794446e0.06a, 0.0699428473753277e0.07  
 a, 0.0799146939691727e0.08a, 0.08987854919801104e0.09a, 0.09983341664682814e0.1a, 0.10977  
 83008371748e0.11a, 0.1197122072889193e0.12a, 0.1296341426196948e0.13a, 0.13954311464423  
 65e0.14a, 0.1494381324735992e0.15a, 0.159318206614246e0.16a, 0.169182349066996e0.17a, 0.1790  
 295734258242e0.18a, 0.1888588949765006e0.19a, 0.1986693307950612e0.2a, 0.20845989984609  
 96e0.21a, 0.2182296230808694e0.2200000000000001a, 0.2279775235351885e0.2300000000000001a, 0.2377  
 026264271347e0.2400000000000001a, 0.247403959254523e0.2500000000000001a, 0.2570805518921552e  
 0.2600000000000001a, 0.2667314366888312e0.2700000000000001a, 0.2763556485641138e0.2800000000000001  
 a, 0.2859522251048356e0.2900000000000001a, 0.2955202066613397e0.3000000000000001a, 0.305058636  
 4434436e0.3100000000000001a, 0.3145665606161179e0.3200000000000001a, 0.3240430283948685e0.3300  
 00000000000001a, 0.3334870921408145e0.3400000000000001a, 0.3428978074554515e0.3500000000000001a, 0.3  
 522742332750901e0.3600000000000002a, 0.3616154319649622e0.3700000000000002a, 0.370920469412  
 9828e0.3800000000000002a, 0.3801884151231616e0.3900000000000002a, 0.3894183423086507e0.400000000  
 0000002a, 0.3986093279844231e0.4100000000000002a, 0.4077604530595704e0.4200000000000002a, 0.4168  
 70802429211e0.4300000000000002a, 0.4259394650659998e0.4400000000000002a, 0.434965534112304e  
 0.4500000000000002a, 0.44394810696552e0.4600000000000002a, 0.4528862853790685e0.4700000000000003a, 0  
 .4617791755414831e0.4800000000000003a, 0.4706258881711582e0.4900000000000003a, 0.47942553860  
 42032e0.5000000000000002a, 0.4881772468829077e0.5100000000000002a, 0.4968801378437369e0.5200000  
 000000002a, 0.5055333412048472e0.5300000000000002a, 0.5141359916531133e0.5400000000000003a, 0.522  
 6872289306594e0.5500000000000003a, 0.5311861979208836e0.5600000000000003a, 0.53963204873396  
 95e0.5700000000000003a, 0.5480239367918738e0.5800000000000003a, 0.556361022912784e0.590000000000000  
 003a, 0.5646424733950356e0.6000000000000003a, 0.5728674601004815e0.6100000000000003a, 0.5810351  
 605373053e0.6200000000000003a, 0.5891447579422698e0.6300000000000003a, 0.5971954413623923e0.6  
 400000000000003a, 0.6051864057360399e0.6500000000000004a, 0.6131168519734341e0.6600000000000004a,  
 0.6209859870365599e0.6700000000000004a, 0.6287930240184688e0.6800000000000004a, 0.6365371822  
 219682e0.6900000000000004a, 0.6442176872376913e0.7000000000000004a, 0.651833771021537e0.7100000  
 000000004a, 0.6593846719714734e0.7200000000000004a, 0.6668696350036982e0.7300000000000004a, 0.674  
 2879116281454e0.7400000000000004a, 0.6816387600233345e0.7500000000000004a, 0.68892144511055  
 16e0.7600000000000005a, 0.696135238627357e0.7700000000000005a, 0.7032794192004105e0.780000000000000  
 005a, 0.7103532724176082e0.7900000000000005a, 0.7173560908995231e0.8000000000000005a, 0.7242871  
 743701429e0.8100000000000005a, 0.7311458297268962e0.8200000000000005a, 0.7379313711099631e0.8  
 3000000000000005a, 0.7446431199708596e0.8400000000000005a, 0.751280405140293e0.8500000000000005a, 0  
 .7578425628952773e0.8600000000000005a, 0.7643289370255054e0.8700000000000006a, 0.77073887889  
 89696e0.8800000000000006a, 0.7770717475268242e0.8900000000000006a, 0.7833269096274837e0.9000000  
 000000006a, 0.7895037396899508e0.9100000000000006a, 0.7956016200363664e0.9200000000000006a, 0.801  
 6199408837775e0.9300000000000006a, 0.8075581004051147e0.9400000000000006a, 0.81341550478937  
 41e0.9500000000000006a, 0.8191915683009986e0.9600000000000006a, 0.8248857133384504e0.97000000000  
 00006a, 0.8304973704919708e0.9800000000000006a, 0.8360259786005209e0.9900000000000007a]fy([0,0.01,0  
 .02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19,0.2,0.21,0.2200000000000001,0.230  
 00000000000001,0.2400000000000001,0.2500000000000001,0.2600000000000001,0.2700000000000001,0.280000000000000  
 001,0.2900000000000001,0.3000000000000001,0.3100000000000001,0.3200000000000001,0.3300000000000001,0.34000  
 000000000001,0.3500000000000001,0.3600000000000002,0.3700000000000002,0.3800000000000002,0.390000000000000  
 2,0.4000000000000002,0.4100000000000002,0.4200000000000002,0.4300000000000002,0.4400000000000002,0.4500000  
 000000002,0.4600000000000002,0.4700000000000003,0.4800000000000003,0.4900000000000003,0.5000000000000002,

0.5100000000000002, 0.5200000000000002, 0.5300000000000002, 0.5400000000000003, 0.5500000000000003, 0.56000000  
 00000003, 0.5700000000000003, 0.5800000000000003, 0.5900000000000003, 0.6000000000000003, 0.6100000000000003, 0.  
 6200000000000003, 0.6300000000000003, 0.6400000000000003, 0.6500000000000004, 0.6600000000000004, 0.6700000000  
 000004, 0.6800000000000004, 0.6900000000000004, 0.7000000000000004, 0.7100000000000004, 0.7200000000000004, 0.73  
 00000000000004, 0.7400000000000004, 0.7500000000000004, 0.7600000000000005, 0.7700000000000005, 0.780000000000  
 0005, 0.7900000000000005, 0.8000000000000005, 0.8100000000000005, 0.8200000000000005, 0.8300000000000005, 0.8400  
 000000000005, 0.8500000000000005, 0.8600000000000005, 0.8700000000000006, 0.8800000000000006, 0.8900000000000006  
 06, 0.9000000000000006, 0.9100000000000006, 0.9200000000000006, 0.9300000000000006, 0.9400000000000006, 0.950000  
 0000000006, 0.9600000000000006, 0.9700000000000006, 0.9800000000000006, 0.9900000000000007] = [0, 0.00999833334  
 166664e0.01a, 0.019998666933308e0.02a, 0.02999550020249566e0.03a, 0.03998933418663416e0.04a, 0.04997916927067  
 833e0.05a, 0.0599640064794446e0.06a, 0.06994284733753277e0.07a, 0.0799146939691727e0.08a, 0.08987854919801104e0  
 .09a, 0.09983341664682814e0.1a, 0.1097783008371748e0.11a, 0.1197122072889193e0.12a, 0.1296341426196948e0.13a, 0.1  
 395431146442365e0.14a, 0.1494381324735992e0.15a, 0.159318206614246e0.16a, 0.169182349066996e0.17a, 0.1790295734  
 258242e0.18a, 0.1888588949765006e0.19a, 0.1986693307950612e0.2a, 0.2084598998460996e0.21a, 0.2182296230808694e0  
 .2200000000000001a, 0.2279775235351885e0.2300000000000001a, 0.2377026264271347e0.2400000000000001a, 0.247403  
 959254523e0.2500000000000001a, 0.2570805518921552e0.2600000000000001a, 0.2667314366888312e0.2700000000000000  
 01a, 0.2763556485641138e0.2800000000000001a, 0.2859522251048356e0.2900000000000001a, 0.2955202066613397e0.30  
 00000000000001a, 0.3050586364434436e0.3100000000000001a, 0.314566560616179e0.3200000000000001a, 0.32404302  
 83948685e0.3300000000000001a, 0.3334870921408145e0.3400000000000001a, 0.3428978074554515e0.3500000000000000  
 1a, 0.3522742332750901e0.3600000000000002a, 0.3616154319649622e0.3700000000000002a, 0.370920469412982e0.380  
 00000000000002a, 0.3801884151231616e0.3900000000000002a, 0.3894183423086507e0.4000000000000002a, 0.398609327  
 9844231e0.4100000000000002a, 0.4077604530595704e0.4200000000000002a, 0.416870802429211e0.4300000000000002a,  
 0.4259394650659998e0.4400000000000002a, 0.4349655341112304e0.4500000000000002a, 0.44394810696552e0.4600000  
 00000002a, 0.4528862853790685e0.4700000000000003a, 0.4617791755414831e0.4800000000000003a, 0.4706258881711  
 582e0.4900000000000003a, 0.4794255386042032e0.5000000000000002a, 0.488177246882907e0.5100000000000002a, 0.4  
 968801378437369e0.5200000000000002a, 0.5055333412048472e0.5300000000000002a, 0.5141359916531133e0.5400000  
 00000003a, 0.5226872289306594e0.5500000000000003a, 0.5311861979208836e0.5600000000000003a, 0.53963204873396  
 95e0.5700000000000003a, 0.5480239367918738e0.5800000000000003a, 0.556361022912784e0.5900000000000003a, 0.564  
 6424733950356e0.6000000000000003a, 0.5728674601004815e0.6100000000000003a, 0.5810351605373053e0.6200000000  
 000003a, 0.5891447579422698e0.6300000000000003a, 0.5971954413623923e0.6400000000000003a, 0.6051864057360399  
 e0.6500000000000004a, 0.6131168519734341e0.6600000000000004a, 0.6209859870365599e0.6700000000000004a, 0.6287  
 930240184688e0.6800000000000004a, 0.6365371822219682e0.6900000000000004a, 0.6442176872376913e0.700000000000000  
 00004a, 0.651833771021537e0.7100000000000004a, 0.6593846719714734e0.7200000000000004a, 0.6668696350036982e0.  
 7300000000000004a, 0.6742879116281454e0.7400000000000004a, 0.6816387600233345e0.7500000000000004a, 0.688921  
 4451105516e0.7600000000000005a, 0.696135238627357e0.7700000000000005a, 0.7032794192004105e0.780000000000000  
 05a, 0.7103532724176082e0.7900000000000005a, 0.7173560908995231e0.8000000000000005a, 0.7242871743701429e0.81  
 00000000000005a, 0.7311458297268962e0.8200000000000005a, 0.7379313711099631e0.8300000000000005a, 0.74464311  
 99708596e0.8400000000000005a, 0.751280405140293e0.8500000000000005a, 0.7578425628952773e0.8600000000000005  
 a, 0.7643289370255054e0.8700000000000006a, 0.7707388788989696e0.8800000000000006a, 0.7770717475268242e0.8900  
 000000000006a, 0.7833269096274837e0.9000000000000006a, 0.7895037396899508e0.9100000000000006a, 0.7956016200  
 363664e0.9200000000000006a, 0.8016199408837775e0.9300000000000006a, 0.8075581004051147e0.9400000000000006a  
 , 0.8134155047893741e0.9500000000000006a, 0.8191915683009986e0.9600000000000006a, 0.8248857133384504e0.97000  
 0000000006a, 0.8304973704919708e0.9800000000000006a, 0.8360259786005209e0.9900000000000007a]

```

>function df(t) &=
trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t)
Maxima said:
diff: second argument must be a variable; found errexpl
-- an error. To debug this try: debugmode(true);

Error in:
...
e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t ...
^

>S &=integrate(df(t),t,0,2*pi); $S // panjang kurva (spiral)
Maxima said:
defint: variable of integration cannot be a constant; found errexpl
-- an error. To debug this try: debugmode(true);

Error in:
S &=integrate(df(t),t,0,2*pi); $S // panjang kurva (spiral) ...
^

>S(a=0.1) // Panjang kurva untuk a=0.1
Function S not found.
Try list ... to find functions!
Error in:

```

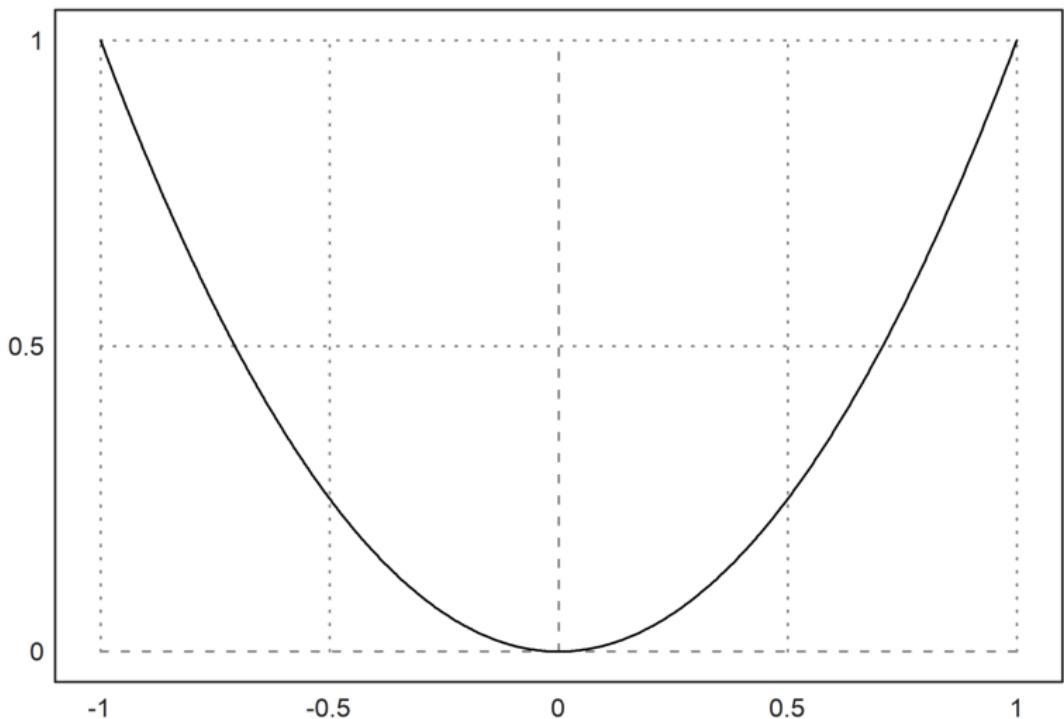
```
S(a=0.1) // Panjang kurva untuk a=0.1 ...  
^
```

Soal:

Tunjukkan bahwa keliling lingkaran dengan jari-jari r adalah  $K=2\pi r$ .

Berikut adalah contoh menghitung panjang parabola.

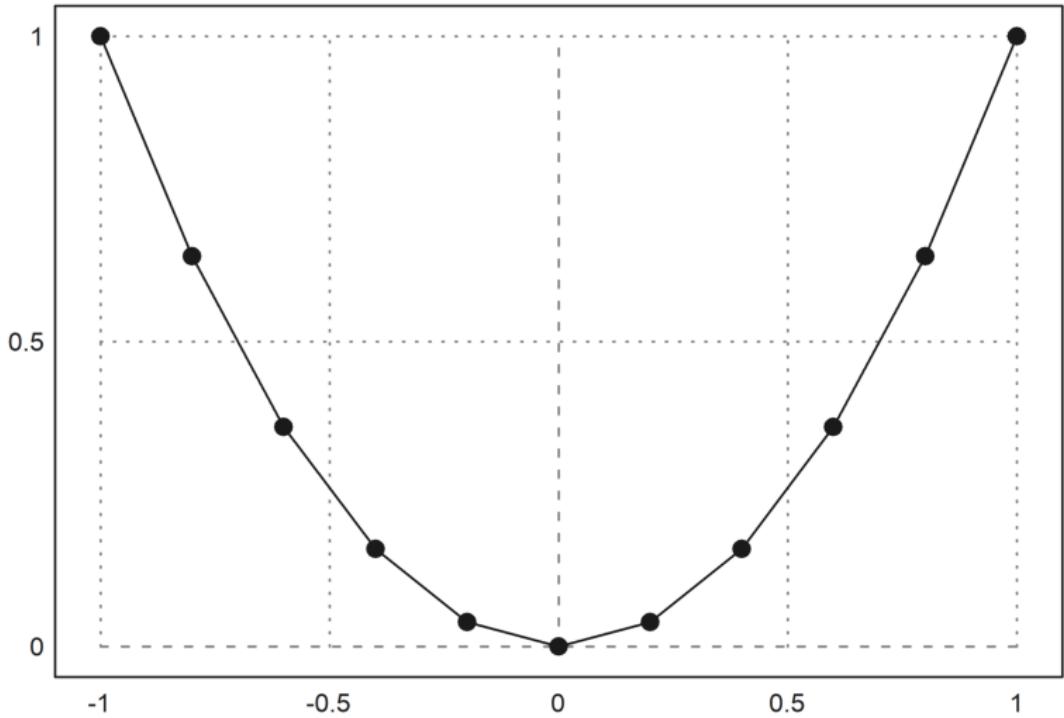
```
>plot2d("x^2",xmin=-1,xmax=1):
```



```
>$showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))  
ʃ1-14x2+1-----√dx=asinh2+25-√2ʃ-114x2+1dx=asinh2+252
```

```
>$float(%)  
ʃ1.0-1.04.0x2+1.0-----√dx=2.957885715089195ʃ-1.01.04.0x2+1.0dx=2.957885715089195
```

```
>x=-1:0.2:1; y=x^2; plot2d(x,y); ...  
plot2d(x,y,points=1,style="o#",add=1):
```



Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```
>i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])^2+(y[i+1]-y[i])^2))
2.95191957027
```

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

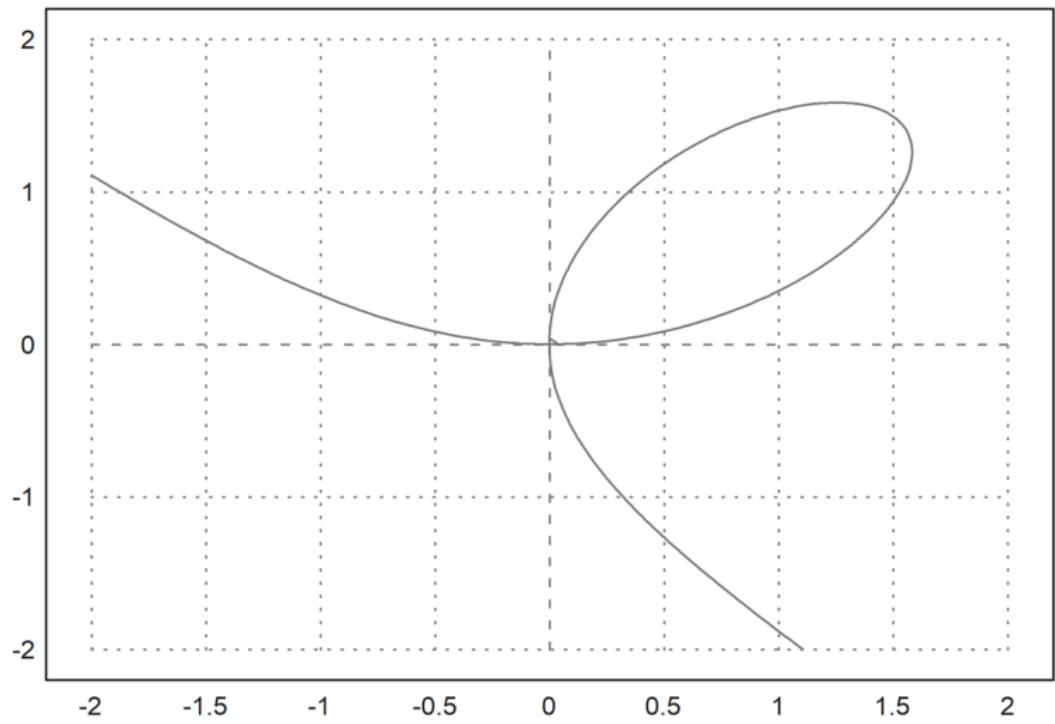
## Koordinat Kartesius

---

Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

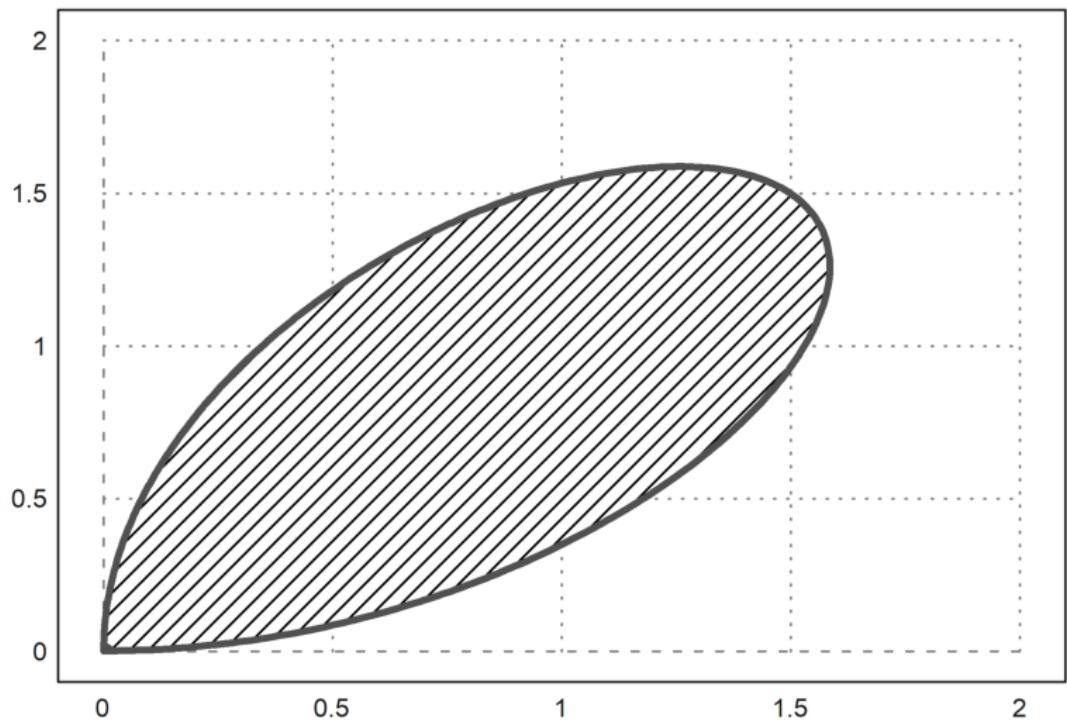
$$x^3 + y^3 - 3xy = 0 \quad x^3 + y^3 - 3xy + x^3 = 0$$

```
>z &= x^3+y^3-3*x*y; $z
y^3-3xy+x^3y^3-3xy+x^3
>plot2d(z,r=2,level=0,n=100);
```



Kita tertarik pada kurva di kuadran pertama.

```
>plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="/"):
```



Kita selesaikan persamaannya untuk x.

```
>z with y=1*x, sol &= solve(% ,x); $sol
```

$$[x=3l^3+1, x=0] [x=3l^3+1, x=0]$$

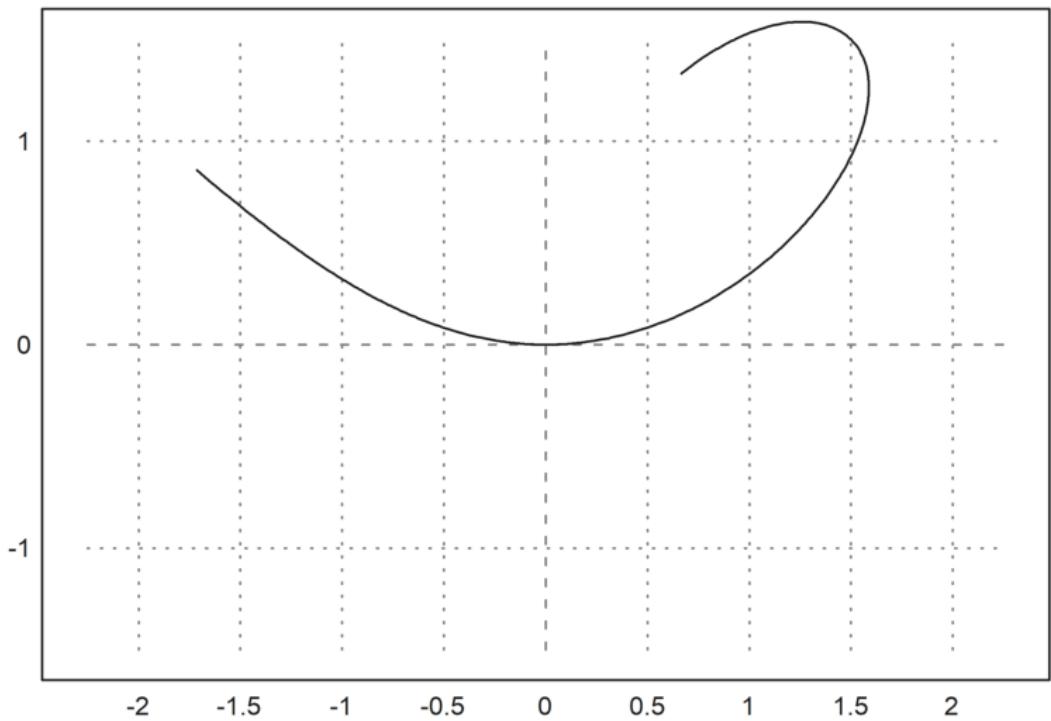
$$\left[ x = \frac{3l}{l^3 + 1}, x = 0 \right]$$

Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); $'f(l)=f(l)
f(l)=3l^3+1f(l)=3l^3+1
```

Fungsi tersebut juga dapat digunakan untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai  $x$  dan nilai  $y=l^3x$ , yakni  $x=f(l)$  dan  $y=l^3f(l)$ .

```
>plot2d(&f(x), &x*f(x), xmin=-0.5, xmax=2, a=0, b=2, c=0, d=2, r=1.5);
```



Elemen panjang kurva adalah:

$$ds = \sqrt{f'(l)^2 + (lf'(l) + f(l))^2}$$

```
>function ds(l) &= ratsimp(sqrt(diff(f(l), l)^2 + diff(l*f(l), l)^2));
$'ds(l)=ds(l)
ds(l)=9l^8+36l^6-36l^5-36l^3+36l^2+9-----\sqrt{l^2+4l^9+6l^6+4l^3+1}-----
-----\sqrt{ds(l)}=9l^8+36l^6-36l^5-36l^3+36l^2+9l^2+9l^9+6l^6+4l^3+1
-----\sqrt{ds(l)}
```

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral tersebut secara numerik dengan Euler.

Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
>2*integrate("ds(x)",0,1)  
4.91748872168
```

```
>2*romberg(&ds(x),0,1)// perintah Euler lain untuk menghitung nilai  
hampiran integral yang sama
```

```
4.91748872168
```

Perhitungan di atas dapat dilakukan untuk sebarang fungsi x dan y dengan mendefinisikan fungsi EMT, misalnya kita beri nama panjangkurva. Fungsi ini selalu memanggil Maxima untuk menurunkan fungsi yang diberikan.

```
>function panjangkurva(fx,fy,a,b) ...  
ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");  
return romberg(ds,a,b);  
endfunction
```

```
>panjangkurva("x","x^2",-1,1) // cek untuk menghitung panjang kurva  
parabola sebelumnya
```

```
2.95788571509
```

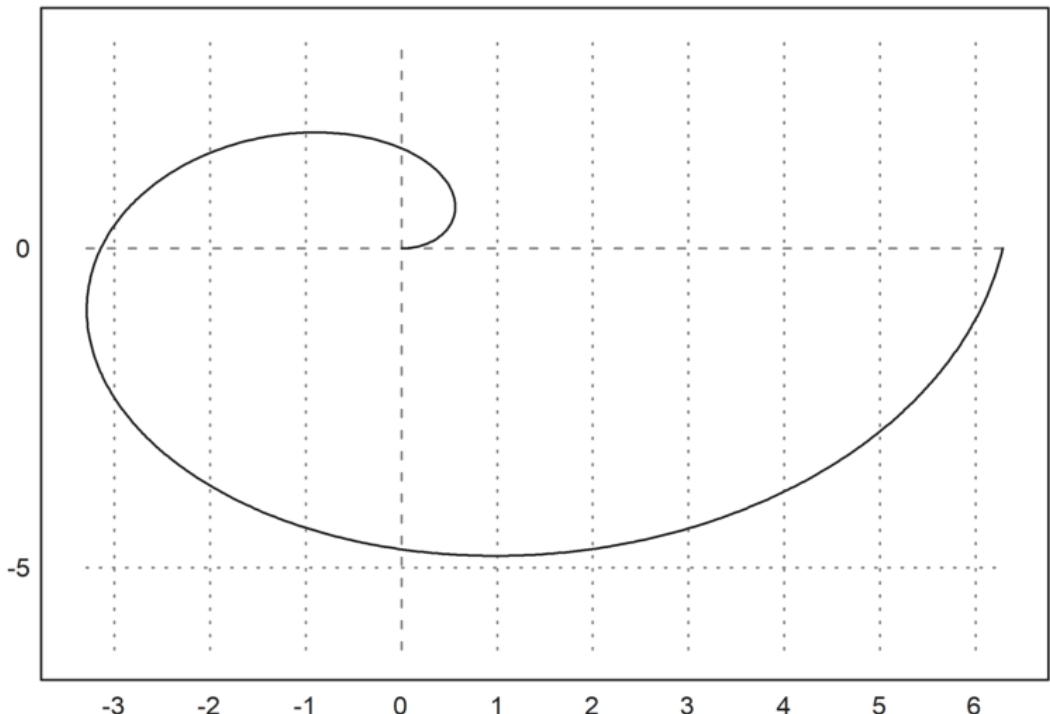
Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)"),mxm("x*f(x)'),0,1) // cek contoh terakhir,  
bandingkan hasilnya!
```

```
4.91748872168
```

Kita hitung panjang spiral Archimedes berikut ini dengan fungsi tersebut.

```
>plot2d("x*cos(x)","x*sin(x)",xmin=0,xmax=2*pi,square=1):
```



```
>panjangkurva("x*cos(x)","x*sin(x)",0,2*pi)
```

```
21.2562941482
```

Berikut kita definisikan fungsi yang sama namun dengan Maxima, untuk perhitungan eksak.

```
>&kill(ds,x,fx,fy)
```

```
done
```

```

>function ds(fx,fy) &=& sqrt(diff(fx,x)^2+diff(fy,x)^2)
              2           2
              sqrt(diff (fy, x) + diff (fx, x))

>sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk menghitung
panjang kurva terakhir di atas
(cosx-xsinx)2+(sinx+x*cosx)2-----sqrt((cos[f0]x-xsin[f0]x)2+(sin[f0]x+x*cos[f0]
x)2)

>$sol | trigreduce | expand, $integrate(% ,x,0,2*pi), %()
asinh(2pi)+2pi4pi2+1-----sqrt(2asinh(2pi)+2pi4pi2+12

```

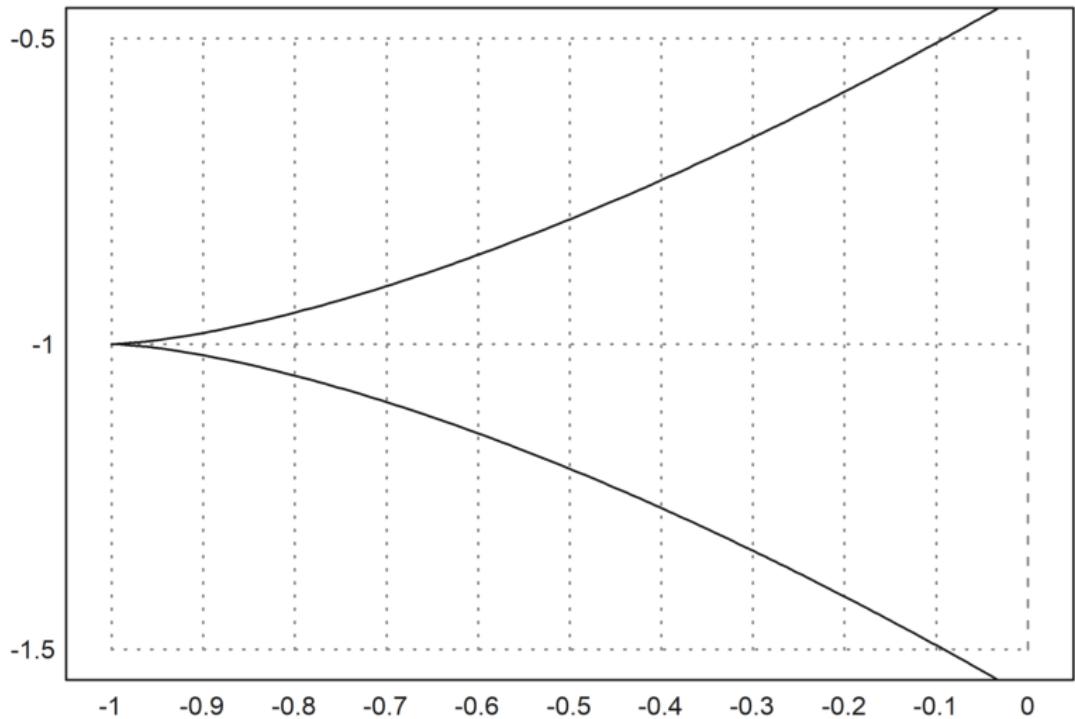
$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$

21.2562941482

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.

Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
>plot2d("3*x^2-1","3*x^3-1",xmin=-1/sqrt(3),xmax=1/sqrt(3),square=1):
```



```

>sol &= radcan(ds(3*x^2-1,3*x^3-1)); $sol
3x9x2+4-----sqrt(3x9x2+4

>$showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%) // panjang kurva di
atas
6.0J0.57735026918962580.0x9.0x2+4.0-----sqrt(dx=2.3378353727671416.0|0.005773502691896258x9.0
x2+4.0dx=2.337835372767141

```

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0 x^2 + 4.0} dx = 2.337835372767141$$

## Sikloid

---

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar ke kanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah  $r$ . Posisi titik pusat lingkaran pada saat  $t$  adalah:

$$(rt, r).$$

Misalkan posisi titik pada lingkaran tersebut mula-mula  $(0, 0)$  dan posisinya pada saat  $t$  adalah:

$$(r(t-\sin(t)), r(1-\cos(t))).$$

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika  $t=0$ ,  $t=\pi/2$ ,  $t=r\pi$ .

```
>x &= r*(t-sin(t))
      r (t - sin(t))

>y &= r*(1-cos(t))
      r (1 - cos(t))
```

Berikut kita gambar sikloid untuk  $r=1$ .

```
>ex &= x-sin(x); ey &= 1-cos(x); aspect(1);
>plot2d(ex,ey,xmin=0,xmax=4pi,square=1); ...
  plot2d("2+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
  plot2d([2,ex(2)], [1,ey(2)],color=red,>add); ...
  plot2d(ex(2),ey(2),>points,>add,color=red); ...
  plot2d("2pi+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
  plot2d([2pi,ex(2pi)], [1,ey(2pi)],color=red,>add); ...
  plot2d(ex(2pi),ey(2pi),>points,>add,color=red):
Variable or function t not found.
Error in expression: r*(t-sin(t))-sin(r*(t-sin(t)))
adaptiveeval:
sx=f$(t;args());
Try "trace errors" to inspect local variables after errors.
plot2d:
dw/n,dw/n^2,dw/n;args());
Berikut dihitung panjang lintasan untuk 1 putaran penuh. (Jangan salah menduga bahwa panjang lintasan 1 putaran penuh sama dengan keliling lingkaran!)
>ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds); // elemen
panjang kurva sikloid
Maxima said:
diff: second argument must be a variable; found r*(t-sin(t))
-- an error. To debug this try: debugmode(true);

Error in:
ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds ...
^
>ds &= trigsimp(ds); ds
      ds

>$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu putaran
penuh
Maxima said:
```

```

defint: variable of integration must be a simple or subscripted
variable.
defint: found r*(t-sin(t))
#0: showev(f='integrate(ds,r*(t-sin(t)),0,2*pi))
-- an error. To debug this try: debugmode(true);

Error in:
$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid sat ...
^

>integrate(mxm("ds"),0,2*pi) // hitung secara numerik
Illegal function result in map.
%evalexpression:
    if maps then return %mapexpression1(x,f$;args());
gauss:
    if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());
adaptivegauss:
    t1=gauss(f$,c,c+h;args(),=maps);
Try "trace errors" to inspect local variables after errors.
integrate:
    return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
>romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik
Wrong argument!

Cannot combine a symbolic expression here.
Did you want to create a symbolic expression?
Then start with &.

Try "trace errors" to inspect local variables after errors.
romberg:
    if cols(y)==1 then return y*(b-a); endif;
Error in:
romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik ...
^
Perhatikan, seperti terlihat pada gambar, panjang sikloid lebih besar
daripada keliling lingkarannya, yakni:
 $2\pi \cdot 2\pi$ .

```

## Kurvatur (Kelengkungan) Kurva

---

image: Osculating.png

Aslinya, kelengkungan kurva diferensiabel (yakni, kurva mulus yang tidak lancip) di titik P didefinisikan melalui lingkaran oskulasi (yaitu, lingkaran yang melalui titik P dan terbaik memperkirakan, paling banyak menyinggung kurva di sekitar P). Pusat dan radius kelengkungan kurva di P adalah pusat dan radius lingkaran oskulasi. Kelengkungan adalah kebalikan dari radius kelengkungan:

$$\kappa = \frac{1}{R}$$

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar  $2\pi$  sejauh  $2\pi R$ .) Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekivalen.

## Definisi Kurvatur dengan Fungsi Parametrik Panjang Kurva

---

Setiap kurva diferensiabel dapat dinyatakan dengan persamaan parametrik terhadap panjang kurva s:

$$\gamma(s) = [x(s), y(s)], \gamma(s) = (x(s), y(s)),$$

dengan x dan y adalah fungsi riil yang diferensiabel, yang memenuhi:  
 $\|\gamma'(s)\|^2 = x'(s)^2 + y'(s)^2 = 1$ .

Ini berarti bahwa vektor singgung

$$T(s) = (x'(s), y'(s))$$

memiliki norm 1 dan merupakan vektor singgung satuan.

Apabila kurvanya memiliki turunan kedua, artinya turunan kedua x dan y ada, maka  $T'(s)$  ada. Vektor ini merupakan normal kurva yang arahnya menuju pusat kurvatur, norm-nya merupakan nilai kurvatur (kelengkungan):

$$T(s) T_2(s) \kappa(s) = \gamma'(s) = 1 \text{ (konstanta)} \Rightarrow T(s) \cdot T(s) = 0 = \|T'(s)\|^2 = \|\gamma''(s)\|^2 = x''(s)^2 + y''(s)^2 = 1$$

$$T(s) = \gamma'(s), T_2(s) = 1 \text{ (konstanta)} \Rightarrow T'(s) \cdot T(s) = 0 \kappa(s) = \|T'(s)\| = \|\gamma''(s)\| = x''(s)^2 + y''(s)^2 = 1$$

Nilai

$$R(s) = 1/\kappa(s)$$

disebut jari-jari (radius) kelengkungan kurva.

Bilangan riil

$$k(s) = \pm \kappa(s)$$

disebut nilai kelengkungan bertanda.

Contoh:

Akan ditentukan kurvatur lingkaran

$$x = r \cos t, y = r \sin t$$

```
>fx := r*cos(t); fy := r*sin(t);
>&assume(t>0, r>0); s := integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2), t, 0, t); s
// elemen panjang kurva, panjang busur lingkaran (s)
r t

>&kill(s); fx := r*cos(s/r); fy := r*sin(s/r); // definisi ulang persamaan
parametrik terhadap s dengan substitusi t=s/r
>k := trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur
lingkaran dengan menggunakan definisi di atas
```

1/r^2

Untuk representasi parametrik umum, misalkan

$$x = x(t), y = y(t)$$

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

$\kappa = d\phi/ds = d\phi/dt ds/dt$  ( $\phi$  adalah sudut kemiringan garis singgung dan s adalah panjang kurva) =  $d\phi/dt (dx/dt)^2 + (dy/dt)^2 = 1$ .  $\kappa = d\phi/ds = d\phi/dt ds/dt$  ( $\phi$  adalah sudut kemiringan garis singgung dan s adalah panjang kurva) =  $d\phi/dt (dx/dt)^2 + (dy/dt)^2 = d\phi/dt (x'(t)^2 + y'(t)^2)$ .

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

$$\sec^2 \phi d\phi dt ds/dt = ddt(\tan \phi) = ddt(dy/dx) = ddt(dy/dt dx/dt) = ddt(y'(t)x'(t)) = x'(t)y''(t) - x''(t)y'(t)x'(t)^2 = 1$$

$$\sec^2 \phi x'(t)y''(t) - x''(t)y'(t)x'(t)^2 = 1 + \tan^2 \phi x'(t)y''(t) - x''(t)y'(t)x'(t)^2 = 1 + (y'(t)x'(t))^2 x'(t)y''(t) - x''(t)y'(t)x'(t)^2 = x'(t)y''(t) - x''(t)y'(t)x'(t)^2 + y'(t)x'(t)^2 = x'(t)y''(t) - x''(t)y'(t)x'(t)^2 + y'(t)x'(t)^2 = \sec^2 \phi d\phi dt = ddt(\tan \phi) = ddt(dy/dx) = ddt(dy/dt dx/dt) = ddt(y'(t)x'(t)) =$$

$x'(t)y''(t) - x''(t)y'(t)x'(t)^2 \cdot d\phi dt = 1 \sec^2(\theta) \phi x'(t)y''(t) - x''(t)y'(t)x'(t)^2 = 1 + \tan^2(\theta) \phi x'(t)y''(t) - x''(t)y'(t)x'(t)^2 = 1 + (y'(t)x'(t))^2$

Jadi, rumus kurvatur untuk kurva parametrik  
 $x=x(t), y=y(t)$

adalah

$$\kappa(t) = x'(t)y''(t) - x''(t)y'(t)(x'(t)^2 + y'(t)^2)^{3/2}.$$

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kutub

$$x=r(\theta)\cos\theta, y=r(\theta)\sin\theta, x=r(\theta)\cos(\theta), y=r(\theta)\sin(\theta),$$

maka rumus kurvaturnya adalah

$$\kappa(\theta) = r(\theta)^2 + 2r'(\theta)^2 - r(\theta)r''(\theta)(r'(\theta)^2 + r''(\theta)^2)^{3/2}.$$

(Silakan Anda turunkan rumus tersebut!)

Contoh:

Lingkaran dengan pusat  $(0, 0)$  dan jari-jari  $r$  dapat dinyatakan dengan persamaan parametrik

$$x=rcost, y=rsint, x=r\cos(\theta), y=r\sin(\theta).$$

Nilai kelengkungan lingkaran tersebut adalah

$$\kappa(t) = x'(t)y''(t) - x''(t)y'(t)(x'(t)^2 + y'(t)^2)^{3/2} = r^2r^3 = 1r.$$

Hasil cocok dengan definisi kurvatur suatu kelengkungan.  
Kurva

$$y=f(x)$$

dapat dinyatakan ke dalam persamaan parametrik

$$x=t, y=f(t), \text{ dengan } x'(t)=1, x''(t)=0, x=t, y=f(t), \text{ dengan } x'(t)=1, x''(t)=0,$$

sehingga kurvaturnya adalah

$$\kappa(t) = y''(t)(1+y'(t)^2)^{3/2}.$$

Contoh:

Akan ditentukan kurvatur parabola

$$y=ax^2+bx+c, y=ax^2+bx+c.$$

```
>function f(x) &= a*x^2+b*x+c; $y=f(x)
r(1-cost)=br(t-sint)+ar2(t-sint)^2+cr(1-cos(t))=br(t-sin(t))+ar2(t-sin(t))^2+2c
```

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); '$k(x)=k(x) //'
kelengkungan parabola
```

Maxima said:

diff: second argument must be a variable; found r\*(t-sin(t))
-- an error. To debug this try: debugmode(true);

Error in:

```
... (x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); '$k(x)=k(x) ...'
```

```
>function f(x) &= x^2+x+1; $y=f(x) // akan kita plot kelengkungan parabola
untuk a=b=c=1
```

$$r(1-cost)=r(t-sint)+r2(t-sint)^2+1r(1-cos(t))=r(t-sin(t))+r2(t-sin(t))^2+1$$

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); '$k(x)=k(x) //'
kelengkungan parabola
```

Maxima said:

diff: second argument must be a variable; found r\*(t-sin(t))
-- an error. To debug this try: debugmode(true);

Error in:

```
... (x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) ...
```

Berikut kita gambar parabola tersebut beserta kurva kelengkungan, kurva jari-jari kelengkungan dan salah satu lingkaran oskulasi di titik puncak parabola. Perhatikan, puncak parabola dan jari-jari lingkaran oskulasi di puncak parabola adalah

$$(-1/2, 3/4), 1/k(2)=1/2, (-1/2, 3/4), 1/k(2)=1/2,$$

sehingga pusat lingkaran oskulasi adalah  $(-1/2, 5/4)$ .

```
>plot2d(["f(x)", "k(x)"], -2, 1, color=[blue, red]); plot2d("1/k(x)", -1.5, 1, color=green, >add); ...
plot2d("-1/2+1/k(-1/2)*cos(x)", "5/4+1/k(-1/2)*sin(x)", xmin=0, xmax=2pi, >add, color=blue):
```

Variable or function t not found.

f:

```
useglobal; return r*(t-sin(t))+r^2*(t-sin(t))^2+1
```

Error in expression: f(x)

%ploteval:

```
y0=f$(x[1],args());
```

adaptiveevalone:

```
s=%ploteval(g$,t,args());
```

Try "trace errors" to inspect local variables after errors.

plot2d:

```
dw/n, dw/n^2, dw/n, auto; args());
```

Untuk kurva yang dinyatakan dengan fungsi implisit

$$F(x,y)=0$$

dengan turunan-turunan parsial

$F_x = \partial F / \partial x$ ,  $F_y = \partial F / \partial y$ ,  $F_{xy} = \partial^2 F / \partial x \partial y$ ,  $F_{xx} = \partial^2 F / \partial x^2$ ,  $F_{yy} = \partial^2 F / \partial y^2$ ,  $F_x = \partial F / \partial x$ ,  $F_y = \partial F / \partial y$ ,  $F_{xy} = \partial^2 F / \partial x \partial y$ ,  $F_x = \partial^2 F / \partial x^2$ ,  $F_y = \partial^2 F / \partial y^2$ ,

berlaku

$$F_x dx + F_y dy = 0 \text{ atau } dy/dx = -F_x/F_y, F_x dx + F_y dy = 0 \text{ atau } dx/dy = -F_y/F_x,$$

sehingga kurvaturnya adalah

$$\kappa = F_{2y}F_{xx} - 2F_xF_yF_{xy} + F_{2x}F_{yy} (F_{2x} + F_{2y})^{3/2}. \kappa = F_y^2 F_{xx} - 2F_x F_y F_{xy} + F_x^2 F_{yy} (F_x^2 + F_y^2)^{3/2}.$$

(Silakan Anda turunkan sendiri!)

Contoh 1:

Parabola

$$y = ax^2 + bx + c$$

dapat dinyatakan ke dalam persamaan implisit

$$ax^2 + bx + c - y = 0. ax^2 + bx + c - y = 0.$$

```
>function F(x, y) &= a*x^2+b*x+c-y; $F(x, y)
    br(t-sint)+ar2(t-sint)^2-r(1-cost)+cbr(t-sin[t])^2+ar2(t-sin[t])^2-r(1-cos[t])+c
```

```
>Fx &= diff(F(x, y), x), Fxx &= diff(F(x, y), x, 2), Fy &= diff(F(x, y), y), Fxy
&= diff(diff(F(x, y), x), y), Fyy &= diff(F(x, y), y, 2)
```

Maxima said:

```
diff: second argument must be a variable; found r*(t-sin(t))
-- an error. To debug this try: debugmode(true);
```

Error in:

```
Fx &= diff(F(x, y), x), Fxx &= diff(F(x, y), x, 2), Fy &= diff(F(x, y) ...
```

```
>function k(x) &= (Fy^2*Fxx-2*Fx*Fy*Fxy+Fx^2*Fyy)/(Fx^2+Fy^2)^(3/2);
$'k(x)=k(x) // kurvatur parabola tersebut
```

$$k(r(t-sint)) = Fx^2 Fyy + Fxx Fy^2 - 2 Fx Fxy Fy (Fy^2 + Fx^2)^{3/2} k(r(t-sin[t])) = Fx^2 Fyy + Fxx Fy^2 - 2 Fx Fxy Fy (Fy^2 + Fx^2)^{3/2}$$

Hasilnya sama dengan sebelumnya yang menggunakan persamaan parabola biasa.

## Latihan

- Bukalah buku Kalkulus.
- Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi).
- Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah integral tentu dengan batas-batas yang menarik (Anda tentukan sendiri), seperti contoh-contoh tersebut.
- Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat diintegralkan (cari sedikitnya 3 fungsi).
- Gambar grafik fungsi dan daerah integrasinya pada sumbu koordinat yang sama.
- Gunakan integral tentu untuk mencari luas daerah yang dibatasi oleh dua kurva yang berpotongan di dua titik. (Cari dan gambar kedua kurva dan arsir (warnai) daerah yang dibatasi oleh keduanya.)
- Gunakan integral tentu untuk menghitung volume benda putar kurva  $y = f(x)$  yang diputar mengelilingi sumbu  $x$  dari  $x=a$  sampai  $x=b$ , yakni

$$V = \int_a^b \pi(f(x))^2 dx. V = \int_a^b \pi(f(x))^2 dx.$$

(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:

$$S = \int_a^b \sqrt{1+(f'(x))^2} dx. S = \int_a^b \sqrt{1+(f'(x))^2} dx.$$

(Pilih fungsi dan gambar kurvanya.)

- Apabila fungsi dinyatakan dalam koordinat kutub  $x=f(r,t)$ ,  $y=g(r,t)$ ,  $r=h(t)$ ,  $x=a$  ber sesuaian dengan  $t=t_0$  dan  $x=b$  ber sesuaian dengan  $t=t_1$ , maka rumus di atas akan menjadi:

$$S = \int_{t_0}^{t_1} \sqrt{x'(t)^2 + y'(t)^2} dt. S = \int_{t_0}^{t_1} \sqrt{x'(t)^2 + y'(t)^2} dt.$$

- Pilih beberapa kurva menarik (selain lingkaran dan parabola) dari buku kalkulus. Nyatakan setiap kurva tersebut dalam bentuk:

- a. koordinat Kartesius (persamaan  $y=f(x)$ )
  - b. koordinat kutub ( $r=r(\theta)$ )
  - c. persamaan parametrik  $x=x(t)$ ,  $y=y(t)$
  - d. persamaan implisit  $F(x,y)=0$
- Tentukan kurvatur masing-masing kurva dengan menggunakan keempat representasi tersebut (hasilnya harus sama).
  - Gambarlah kurva asli, kurva kurvatur, kurva jari-jari lingkaran oskulasi, dan salah satu lingkaran oskulasinya.

```
>function g(x) &= 6*x^2;$g(x)
               6r^2(t-sint)^2
               6r^2(t-sin[t])^2

>&showev('integrate(g(x),x))
      /
      [ 2
        6 I r  (t - sin(t))   dr (t - sin(t)) =
      ]
      /
      /
      [ 2
        6 I r  (t - sin(t))   dr (t - sin(t))
      ]
      /
```

```

>$showev('integrate(g(x),x,2,3))
Maxima said:
defint: variable of integration must be a simple or subscripted
variable.
defint: found r*(t-sin(t))
#0: showev(f=6*'integrate(r^2*(t-sin(t))^2,r*(t-sin(t)),2,3))
-- an error. To debug this try: debugmode(true);

Error in:
$showev('integrate(g(x),x,2,3)) ...
^

>x=0.01:0.03:4; plot2d(x,g(x+0.01),>bar); plot2d("g(x)",3,3,>add):
Variable r not found!
Use global or local variables defined in function g.
Error in ^
Try "trace errors" to inspect local variables after errors.
g:
useglobal; return 6*r^2*(t-sin(t))^2
Error in:
x=0.01:0.03:4; plot2d(x,g(x+0.01),>bar); plot2d("g(x)",3,3,>ad ...
^

>function f(x) &= (sin(2*x)) ; $f(x)
sin(2r(t-sint))sin([f0](2r(t-sin[f0]t)))

>$showev('integrate(f(x),x))
∫sin(2r(t-sint))dr(t-sint)=∫sin(2r(t-sint))dr(t-sint)[sin([f0](2r(t-sin[f0]t)))dr(t-sin[f0]t)=∫sin([f0](2r(t-sin[f0]t)))dr(t-sin[f0]t)

>plot2d("f(x)",-2,2):
Variable r not found!
Use global or local variables defined in function f.
f:
useglobal; return sin(2*r*(t-sin(t)))
Error in expression: f(x)
%ploteval:
y0=f$(x[1],args());
adaptiveevalone:
s=%ploteval(g$,t,args());
Try "trace errors" to inspect local variables after errors.
plot2d:
dw/n,dw/n^2,dw/n,auto;args());
>$showev('integrate(x^2*sqrt(3*x+2),x,0,2))
Maxima said:
defint: variable of integration must be a simple or subscripted
variable.
defint: found r*(t-sin(t))
#0: showev(f='integrate(r^2*sqrt(3*r*(t-sin(t))+2)*(t-sin(t))^2,r*(t-sin(t)),0,2))
-- an error. To debug this try: debugmode(true);

Error in:
$showev('integrate(x^2*sqrt(3*x+2),x,0,2)) ...
^

>function f(x) &= 5*x^2+1 ; $f(x)
5r2(t-sint)2+15r2(t-sin[f0]t)2+1

>&showev('integrate(f(x),x))
/
[      2           2
I ( 5 r   (t - sin(t))   + 1) dr (t - sin(t)) =
]

```

```

/
/
[      2           2
I (5 r   (t - sin(t))   + 1) dr (t - sin(t))
]
/

```

```

>plot2d("f(x)",-2,2):
Variable r not found!
Use global or local variables defined in function f.
Error in ^
f:
    useglobal; return 5*r^2*(t-sin(t))^2+1
Error in expression: f(x)
%ploteval:
y0=f$(x[1],args());
adaptiveevalone:
s=%ploteval(g$,t,args());
Try "trace errors" to inspect local variables after errors.
plot2d:
dw/n,dw/n^2,dw/n,auto;args();
showev('integrate(sin(x),x,0,pi))

```

## Barisan dan Deret

---

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan (menggunakan titik dua ":");
- menggunakan perintah "sequence" dan rumus barisan (suku ke  $-n$ );
- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create\_list" atau "makelist" untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
>1:2:30  
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
```

## Iterasi dan Barisan

EMT menyediakan fungsi `iterate("g(x)", x0, n)` untuk melakukan iterasi  $x_{k+1}=g(x_k)$ ,  $x_0=x_0, k=1,2,3,\dots,n$ .  $x_{k+1}=g(x_k)$ ,  $x_0=x_0, k=1,2,3,\dots,n$ .

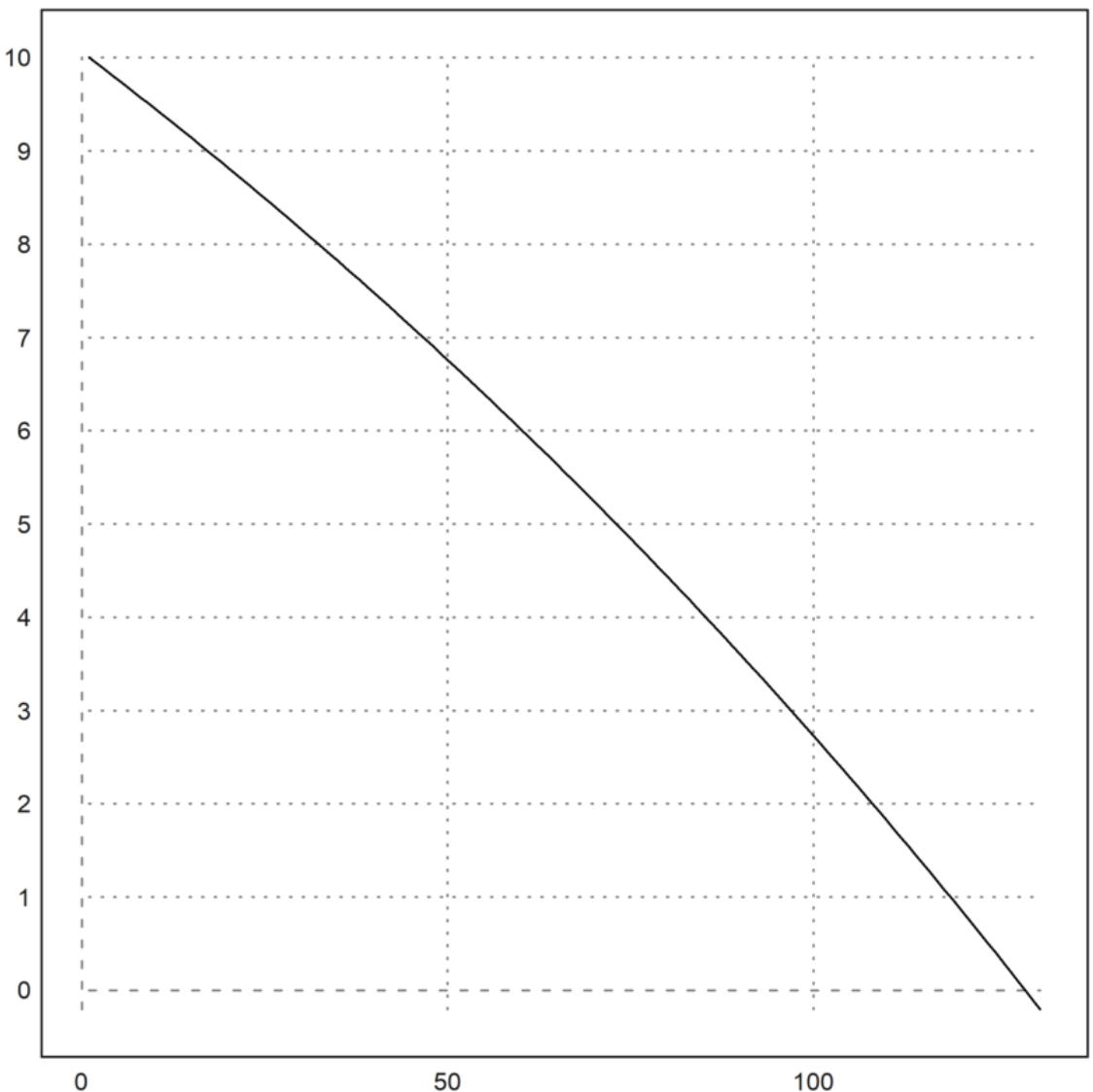
Berikut ini disajikan contoh-contoh penggunaan iterasi dan rekursi dengan EMT. Contoh pertama menunjukkan pertumbuhan dari nilai awal 1000 dengan laju pertambahan 5%, selama 10 periode.

```
>q=1.05; iterate("x*q",1000,n=10)'  
1000  
1050  
1102.5  
1157.63  
1215.51  
1276.28  
1340.1  
1407.1  
1477.46  
1551.33  
1628.89
```

Contoh berikutnya memperlihatkan bahaya menabung di bank pada masa sekarang! Dengan bunga tabungan sebesar 6% per tahun atau 0.5% per bulan dipotong pajak 20%, dan biaya administrasi 10000 per bulan, tabungan sebesar 1 juta tanpa diambil selama sekitar 10 tahunan akan habis diambil oleh bank!

```
>r=0.005; plot2d(iterate("(1+0.8*r)*x-10000",1000000,n=130)):
```

\* 1e+05



Silakan Anda coba-coba, dengan tabungan minimal berapa agar tidak akan habis diambil oleh bank dengan ketentuan bunga dan biaya administrasi seperti di atas.

Berikut adalah perhitungan minimal tabungan agar aman di bank dengan bunga sebesar  $r$  dan biaya administrasi  $a$ , pajak bunga 20%.

```
>$solve(0.8*r*A-a,A), $% with [r=0.005, a=10]
[A=2500.0][A=2500.0]
```

$$[A = 2500.0]$$

Berikut didefinisikan fungsi untuk menghitung saldo tabungan, kemudian dilakukan iterasi.

```
>function saldo(x,r,a) := round((1+0.8*r)*x-a,2);
>iterate({{"saldo",0.005,10}},1000,n=6)
[1000, 994, 987.98, 981.93, 975.86, 969.76, 963.64]
>iterate({{"saldo",0.005,10}},2000,n=6)
[2000, 1998, 1995.99, 1993.97, 1991.95, 1989.92, 1987.88]
```

```
>iterate({{"saldo",0.005,10}},2500,n=6)
[2500, 2500, 2500, 2500, 2500, 2500, 2500]
Tabungan senilai 2,5 juta akan aman dan tidak akan berubah nilai
(jika tidak ada penarikan),
sedangkan jika tabungan awal kurang dari 2,5 juta, lama kelamaan akan
berkurang meskipun
tidak pernah dilakukan penarikan uang tabungan.
```

```
>iterate({{"saldo",0.005,10}},3000,n=6)
[3000, 3002, 3004.01, 3006.03, 3008.05, 3010.08, 3012.12]
Tabungan yang lebih dari 2,5 juta baru akan bertambah jika tidak ada
penarikan.
```

Untuk barisan yang lebih kompleks dapat digunakan fungsi "sequence()".

Fungsi ini menghitung nilai-nilai  $x[n]$  dari semua nilai sebelumnya,  $x[1], \dots, x[n-1]$  yang diketahui.

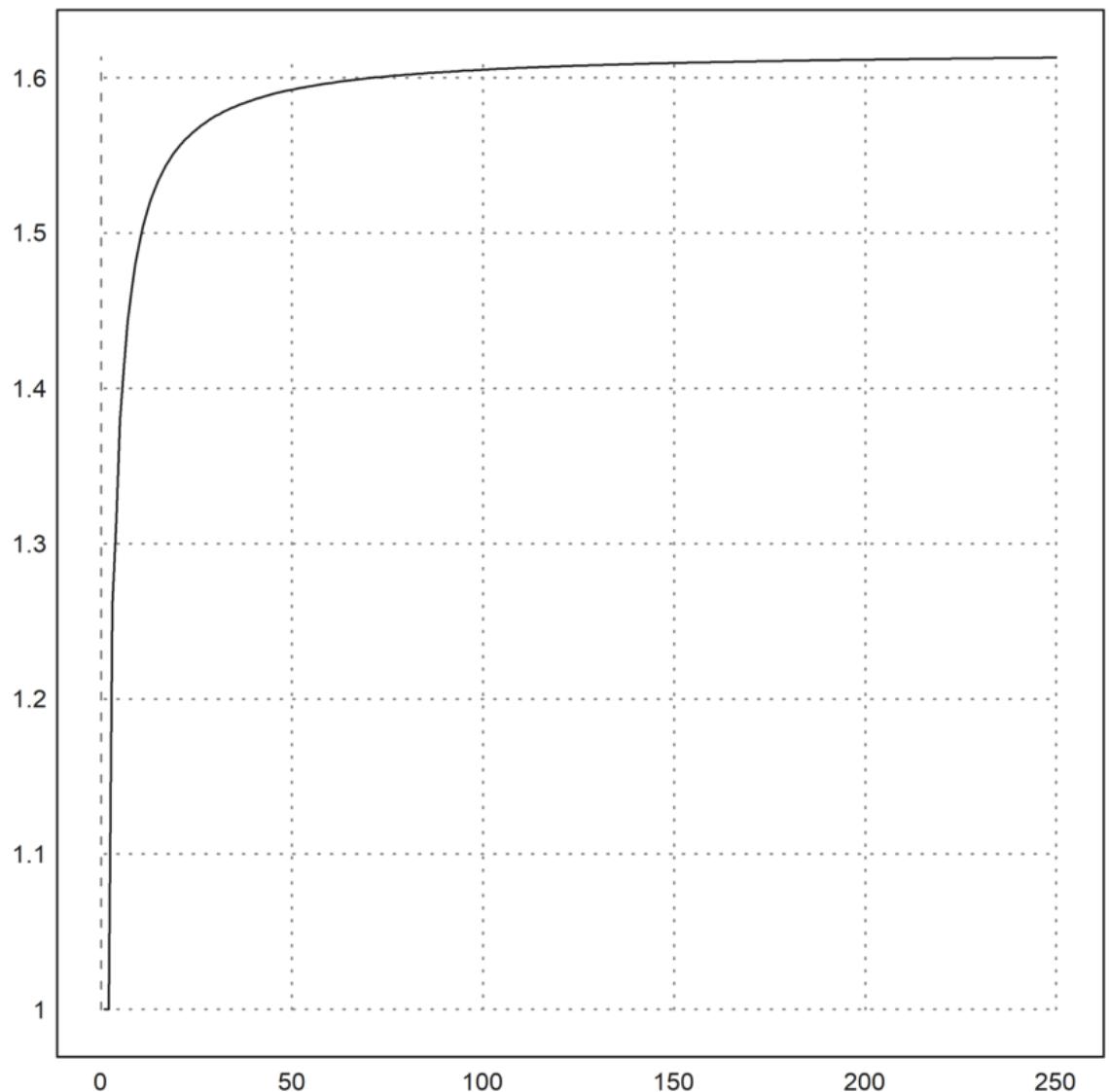
Berikut adalah contoh barisan Fibonacci.

$$x_n = x_{n-1} + x_{n-2}, x_1 = 1, x_2 = 1$$

```
>sequence("x[n-1]+x[n-2]",[1,1],15)
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377,
610]
Barisan Fibonacci memiliki banyak sifat menarik, salah satunya adalah
akar pangkat ke-n suku
ke-n akan konvergen ke pecahan emas:
```

```
>$(1+sqrt(5))/2=float((1+sqrt(5))/2)
5-sqrt(5)=1.6180339887498955+12=1.618033988749895
```

```
>plot2d(sequence("x[n-1]+x[n-2]",[1,1],250)^(1/(1:250))):
```



Barisan yang sama juga dapat dihasilkan dengan menggunakan loop.

```
>x=ones(500); for k=3 to 500; x[k]=x[k-1]+x[k-2]; end;
```

Rekursi dapat dilakukan dengan menggunakan rumus yang tergantung pada semua elemen

sebelumnya. Pada contoh berikut, elemen ke- $n$  merupakan jumlah (n-1) elemen sebelumnya,

dimulai dengan 1 (elemen ke-1). Jelas, nilai elemen ke- $n$  adalah  $2^{(n-2)}$ , untuk n=2, 4, 5,

....

```
>sequence("sum(x)",1,10)
```

```
[1, 1, 2, 4, 8, 16, 32, 64, 128, 256]
```

Selain menggunakan ekspresi dalam x dan n, kita juga dapat menggunakan fungsi.

Pada contoh berikut, digunakan iterasi

$$x_n = A \cdot x_{n-1}, x_n = A \cdot x_{n-1},$$

dengan A suatu matriks  $2 \times 2$ , dan setiap  $x[n]$  merupakan matriks/vektor  $2 \times 1$ .

```
>A=[1,1;1,2]; function suku(x,n) := A.x[,n-1]
>sequence("suku",[1;1],6)
Real 2 x 6 matrix
```

1	2	5	13	...
1	3	8	21	...

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "matrixpower()". Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak  $\log_2(n)$ .

$$x_n = A \cdot x_{n-1} = A_2 \cdot x_{n-2} = A_3 \cdot x_{n-3} = \dots = A_{n-1} \cdot x_1 \cdot x_n = A \cdot x_{n-1} = A_2 \cdot x_{n-2} = A_3 \cdot x_{n-3} = \dots = A_{n-1} \cdot x_1.$$

```
>sequence("matrixpower(A,n).[1;1]",1,6)
Real 2 x 6 matrix
```

1	5	13	34	...
1	8	21	55	...

## Spiral Theodorus

---

image: Spiral\_of\_Theodorus.png

Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

$$x_n = (1+i^{n-1}) \sqrt{x_{n-1}}, x_1 = 1, x_n = (1+i^{n-1})x_{n-1}, x_1 = 1,$$

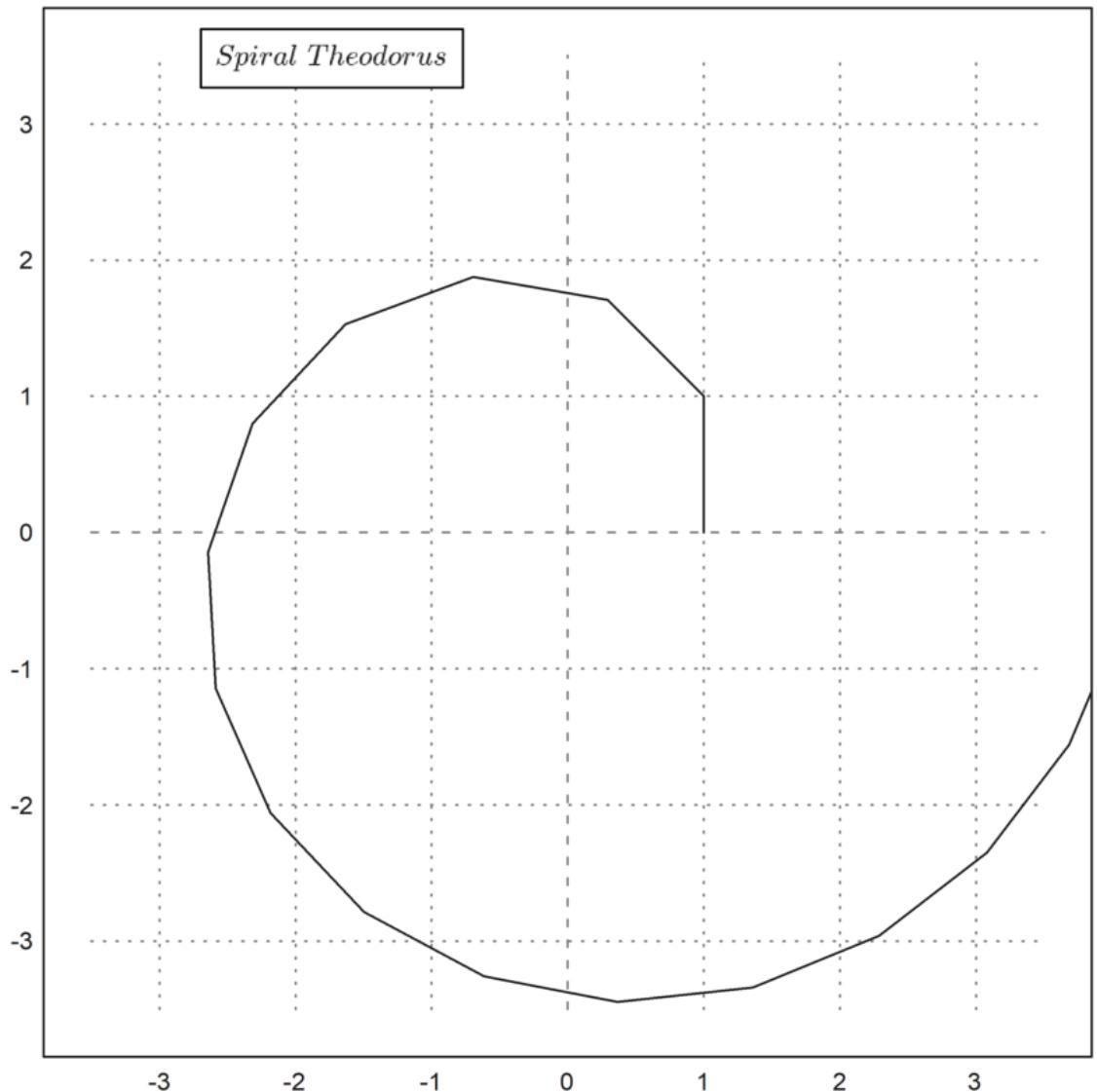
yang menghasilkan barisan bilangan kompleks.

```
>function g(n) := 1+I/sqrt(n)
```

Rekursinya dapat dijalankan sebanyak 17 untuk menghasilkan barisan 17 bilangan kompleks,

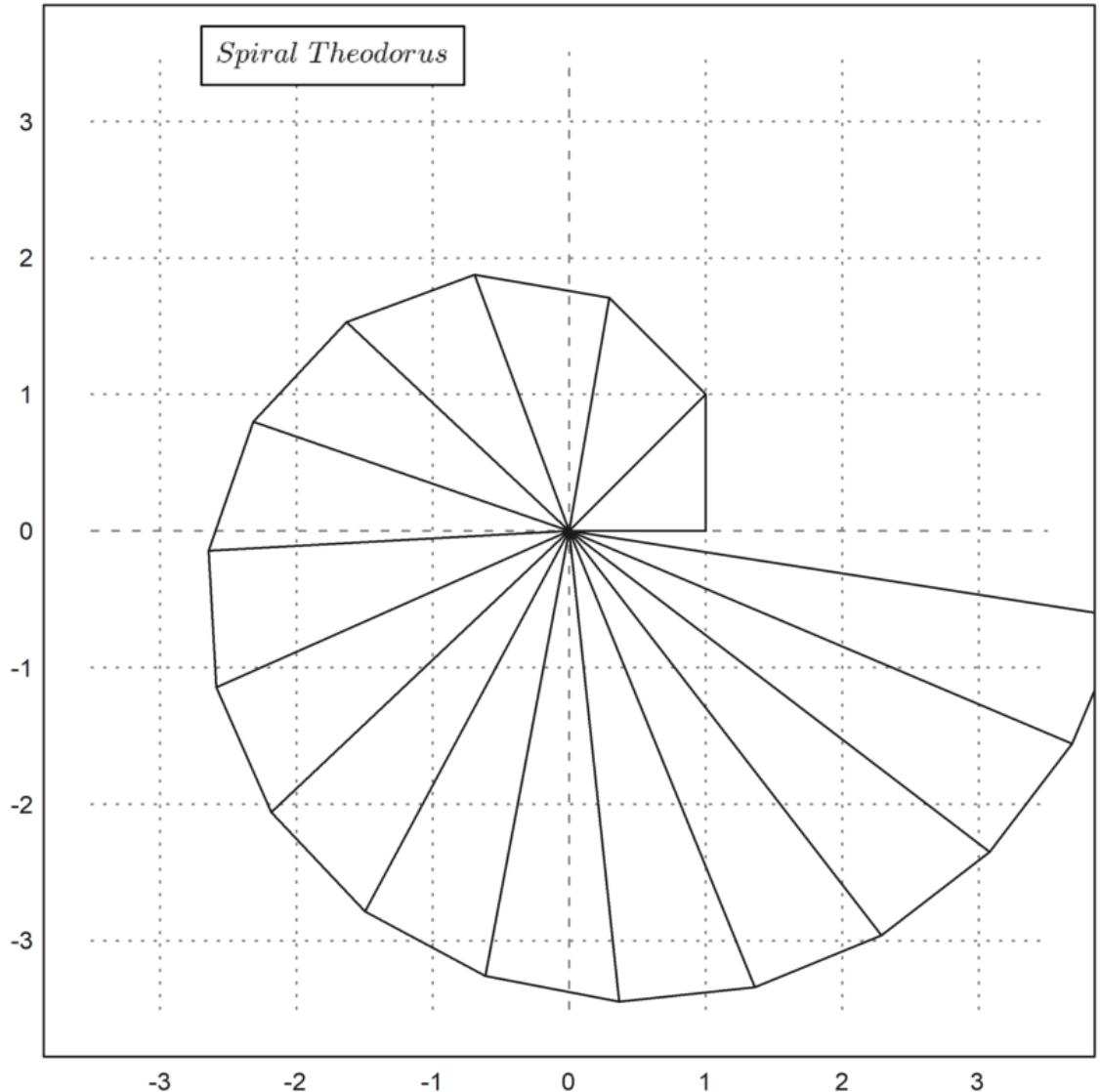
kemudian digambar bilangan-bilangan kompleksnya.

```
>x=sequence("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox(latex("Spiral\Theodorus"),0.4):
```



Selanjutnya dihubungkan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end;
```



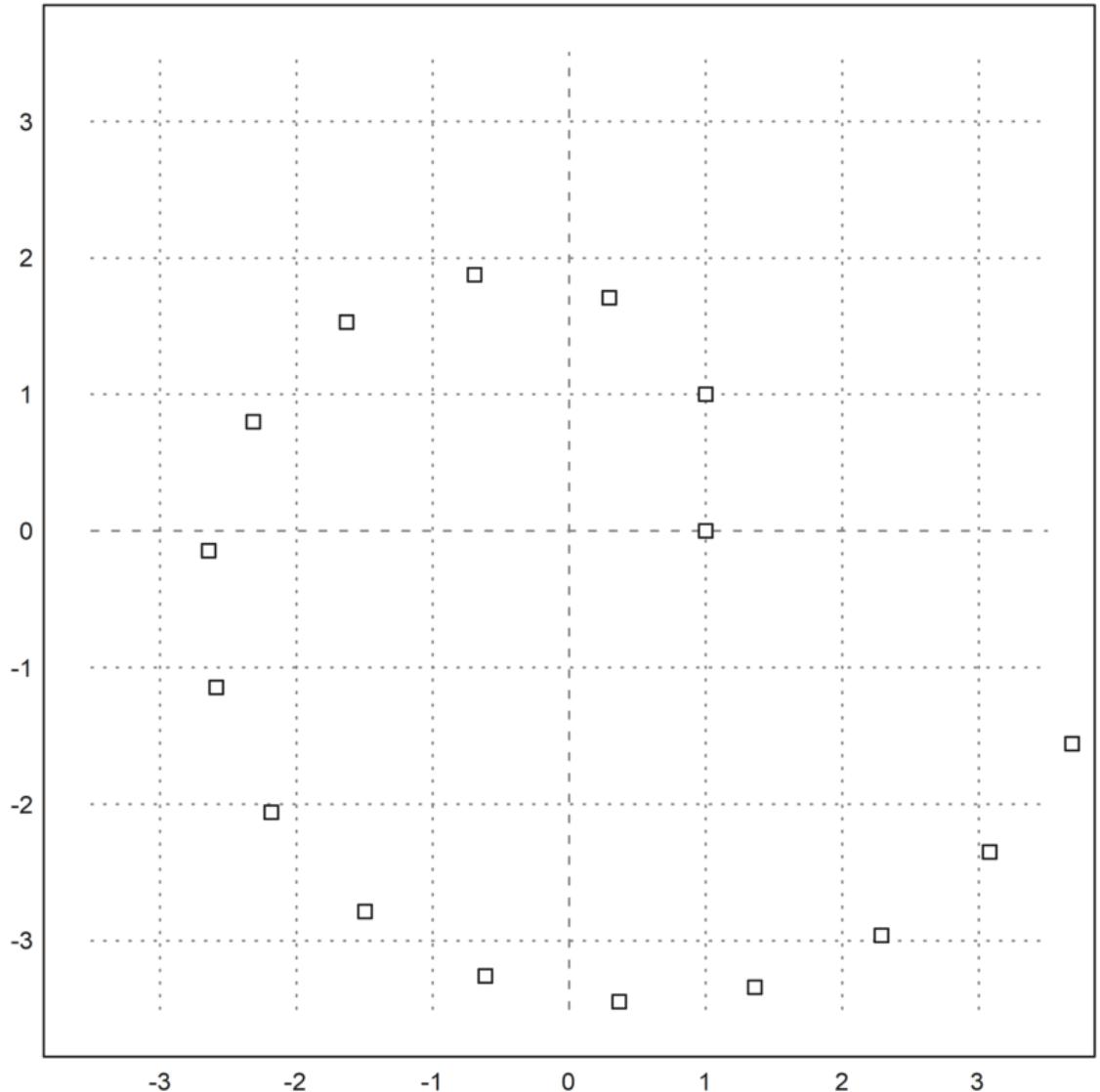
>

Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memerlukan indeks dan bilangan kompleks. Dalam hal ini digunakan vektor kolom pada bidang.

```
>function gstep (v) ...
w=[-v[2];v[1]];
return v+w/norm(w);
endfunction
```

Jika dilakukan iterasi 16 kali dimulai dari  $[1;0]$  akan didapatkan matriks yang memuat vektor-vektor dari setiap iterasi.

```
>x=iterate("gstep",[1;0],16); plot2d(x[1],x[2],r=3.5,>points):
```



## Kekonvergenan

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```
>iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1.
0.739085133216
Iterasi tersebut konvergen ke penyelesaian persamaan
x=cos(x).x=cos[0](x).
```

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.

```
>hasil := iterate("cos(x)",~1,2) //iterasi x(n+1)=cos(x(n)), dengan
interval awal (1, 2)
~0.739085133211,0.7390851332133~
Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa
interval tersebut
memuat akar persamaan x=cos(x).
```

```

>h=expand(hasil,100), cos(h) << h
~0.73908513309,0.73908513333~
1
Iterasi juga dapat digunakan pada fungsi yang didefinisikan.
>function f(x) := (x+2/x)/2
Iterasi x(n+1)=f(x(n)) akan konvergen ke akar kuadrat 2.
>iterate("f",2), sqrt(2)
1.41421356237
1.41421356237
Jika pada perintah iterate diberikan tambahan parameter n, maka hasil
iterasinya akan
ditampilkan mulai dari iterasi pertama sampai ke-n.
>iterate("f",2,5)
[2, 1.5, 1.41667, 1.41422, 1.41421, 1.41421]
Untuk iterasi ini tidak dapat dilakukan terhadap interval.
>niterate("f",~1,2~,5)
[ ~1,2~, ~1,2~, ~1,2~, ~1,2~, ~1,2~, ~1,2~ ]
Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya
adalah perhitungan dengan
interval bersifat terlalu longgar. Untuk meningkatkan perhitungan pada
ekspresi dapat
digunakan pembagian intervalnya, menggunakan fungsi ieval().
>function s(x) := ieval("(x+2/x)/2",x,10)
Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal,
dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang
memuat akar persamaan:

$$x=12(x+2x).x=12(x+2x).$$

Satu-satunya solusi adalah

$$x=2-\sqrt{x=2}.$$


> iterate("s",~1,2~)
~1.41421356236,1.41421356239~
Fungsi "iterate()" juga dapat bekerja pada vektor. Berikut adalah
contoh fungsi vektor, yang menghasilkan rata-rata aritmetika dan
rata-rata geometri.

$$(a_{n+1},b_{n+1})=(a_n+b_n/2,a_nb_n---\sqrt{a_n a_{n+1}})=(a_n+b_n/2,a_nb_n)$$

Iterasi ke-n disimpan pada vektor kolom x[n].
>function g(x) := [(x[1]+x[2])/2,sqrt(x[1]*x[2])]
Iterasi dengan menggunakan fungsi tersebut akan konvergen ke rata-
rata aritmetika dan
geometri dari nilai-nilai awal.
>iterate("g",[1;5])
2.60401
2.60401
Hasil tersebut konvergen agak cepat, seperti kita cek sebagai
berikut.
>iterate("g",[1;5],4)
1 3 2.61803 2.60403 2.60401
5 2.23607 2.59002 2.60399 2.60401
Iterasi pada interval dapat dilakukan dan stabil, namun tidak
menunjukkan bahwa limitnya
pada batas-batas yang dihitung.
>iterate("g",[~1~;~5~],4)
Interval 2 x 5 matrix

$$\begin{matrix} \sim 0.99999999999999778, 1.0000000000000022 & \dots \\ \sim 4.9999999999999911, 5.0000000000000089 & \dots \end{matrix}$$

Iterasi berikut konvergen sangat lambat.

$$x_{n+1}=x_n-\sqrt{x_{n+1}-x_n}.$$


```

```

>iterate("sqrt(x)",2,10)
[2, 1.41421, 1.18921, 1.09051, 1.04427, 1.0219, 1.01089,
1.00543, 1.00271, 1.00135, 1.00068]
Kekonvergenan iterasi tersebut dapat dipercepat dengan percepatan
Steffenson:
>steffenson("sqrt(x)",2,10)
[1.04888, 1.00028, 1, 1]

```

## Iterasi menggunakan Loop yang ditulis Langsung

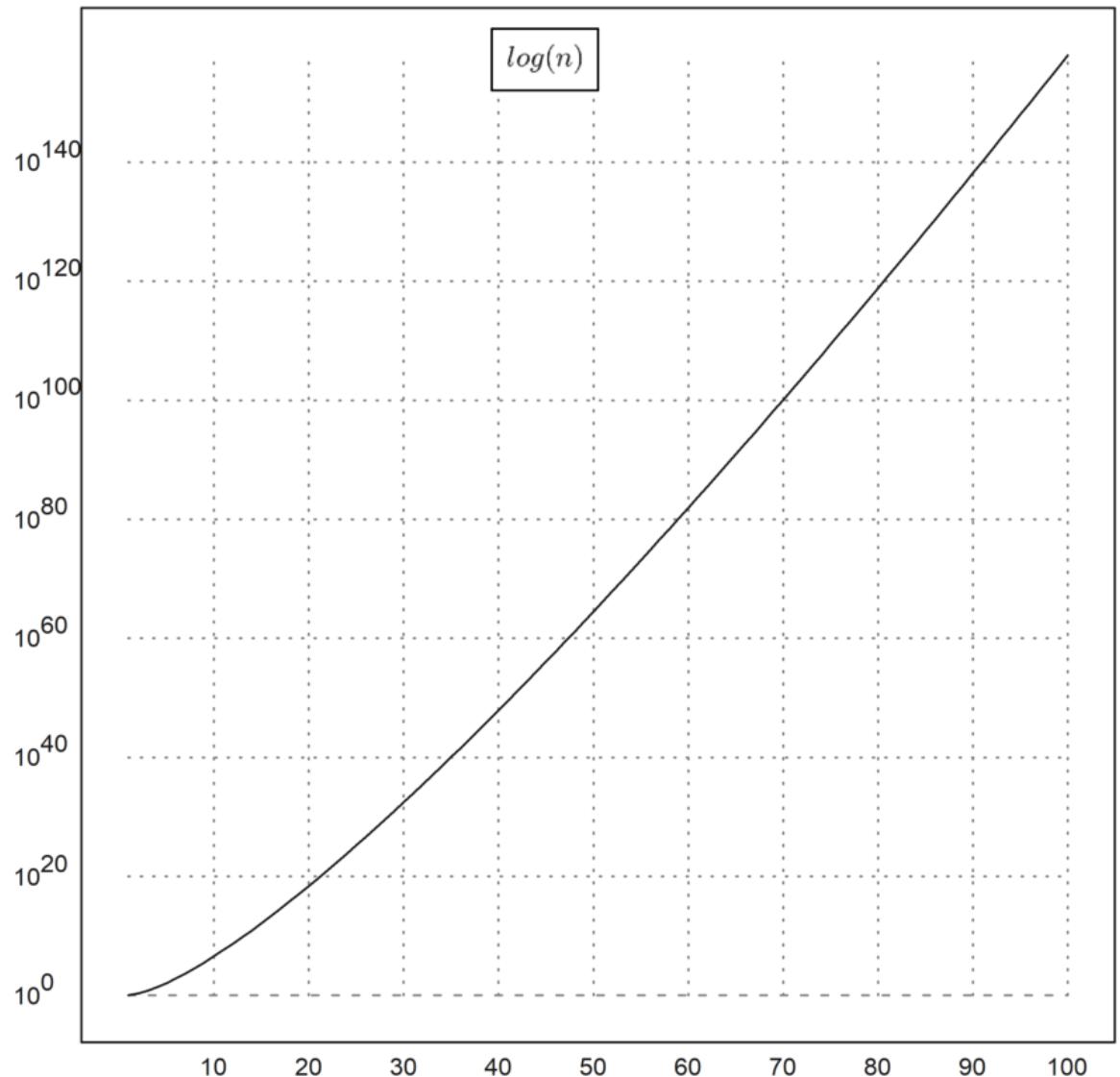
---

Berikut adalah beberapa contoh penggunaan loop untuk melakukan iterasi yang ditulis langsung pada baris perintah.

```

>x=2; repeat x=(x+2/x)/2; until x^2~=2; end; x,
1.41421356237
Penggabungan matriks menggunakan tanda "|" dapat digunakan untuk menyimpan semua hasil
iterasi.
>v=[1]; for i=2 to 8; v=v|(v[i-1]*i); end; v,
[1, 2, 6, 24, 120, 720, 5040, 40320]
hasil iterasi juga dapat disimpan pada vektor yang sudah ada.
>v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]*i; end; ...
plot2d(v,logplot=1); textbox(latex(&log(n)),x=0.5):

```



```
>A =[0.5,0.2;0.7,0.1]; b=[2;2]; ...
x=[1;1]; repeat xnew=A.x-b; until all(xnew~≈x); x=xnew; end; ...
x,
-7.09677
-7.74194
```

## Iterasi di dalam Fungsi

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.

Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

```
>function map hiter(f$,x0) ...
x=x0;
maxiter=0;
repeat
```

```

xnew=f$(x);
maxiter=maxiter+1;
until xnew~≈x;
x=xnew;
end;
return maxiter;
endfunction

```

Misalnya, berikut adalah iterasi untuk mendapatkan hampiran akar kuadrat 2, cukup cepat, konvergen pada iterasi ke-5, jika dimulai dari hampiran awal 2.

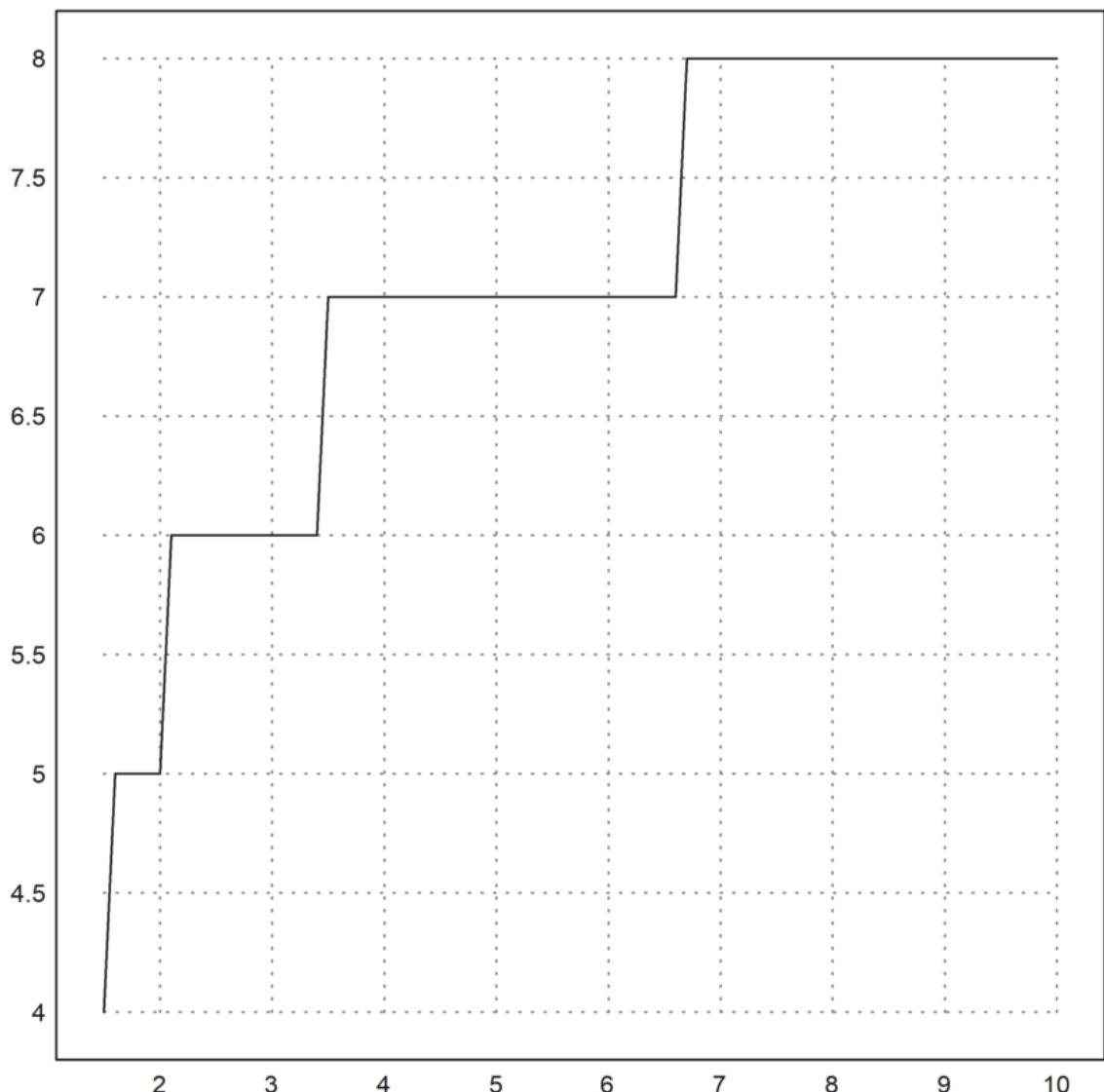
```
>hiter("(x+2/x)/2",2)
```

5

Karena fungsinya didefinisikan menggunakan "map". maka nilai awalnya dapat berupa vektor.

```
>x=1.5:0.1:10; hasil=hiter("(x+2/x)/2",x); ...
```

```
plot2d(x,hasil):
```



Dari gambar di atas terlihat bahwa kekonvergenan iterasinya semakin lambat, untuk nilai awal

```

semakin besar, namun penambahannya tidak kontinu. Kita dapat menemukan
kapan maksimum
iterasinya bertambah.
>hasil[1:10]
[4, 5, 5, 5, 5, 5, 6, 6, 6, 6]
>x[nonzeros(differences(hasil))]
[1.5, 2, 3.4, 6.6]
maksimum iterasi sampai konvergen meningkat pada saat nilai awalnya
1.5, 2, 3.4, dan 6.6.

Contoh berikutnya adalah metode Newton pada polinomial kompleks
berderajat 3.
>p &= x^3-1; newton &= x-p/diff(p,x); $newton
Maxima said:
diff: second argument must be a variable; found r*(t-sin(t))
-- an error. To debug this try: debugmode(true);

Error in:
p &= x^3-1; newton &= x-p/diff(p,x); $newton ...
^

Selanjutnya didefinisikan fungsi untuk melakukan iterasi (aslinya 10
kali).
>function iterasi(f$,x,n=10) ...
loop 1 to n; x=f$(x); end;
return x;
endfunction
Kita mulai dengan menentukan titik-titik grid pada bidang
kompleksnya.
>r=1.5; x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z);
Function newton needs at least 3 arguments!
Use: newton (f$: call, df$: call, x: scalar complex {}, y: number, eps:
none{})
Error in:
... x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z); ...
^

Berikut adalah akar-akar polinomial di atas.
>z=&solve(p)()
Maxima said:
solve: more unknowns than equations.
Unknowns given :
[t,r]
Equations given:
[r^3*(t-sin(t))^3-1]
-- an error. To debug this try: debugmode(true);

Error in:
z=&solve(p)() ...
^

Untuk menggambar hasil iterasinya, dihitung jarak dari hasil iterasi
ke-10 ke masing-masing
akar, kemudian digunakan untuk menghitung warna yang akan digambar,
yang menunjukkan limit
untuk masing-masing nilai awal.

Fungsi plotrgb() menggunakan jendela gambar terkini untuk menggambar
warna RGB sebagai
matriks.
>C=rgb(max(abs(W-z[1])),1),max(abs(W-z[2])),1),max(abs(W-z[3]),1)); ...
plot2d(None,-r,r,-r,r); plotrgb(C):
z is not a variable!
Error in:

```

```
C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1) ...  
^
```

## Iterasi Simbolik

Seperti sudah dibahas sebelumnya, untuk menghasilkan barisan ekspresi simbolik dengan Maxima dapat digunakan fungsi makelist().

```
>&powerdisp:true // untuk menampilkan deret pangkat mulai dari suku berpangkat terkecil
```

```
true
```

```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Taylor untuk e^x
```

```
Maxima said:
```

```
taylor: r*(t-sin(t)) cannot be a variable.  
-- an error. To debug this try: debugmode(true);
```

```
Error in:
```

```
deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // baris ...  
^
```

Untuk mengubah barisan deret tersebut menjadi vektor string di EMT digunakan fungsi mxm2str(). Selanjutnya, vektor string/ekspresi hasilnya dapat digambar seperti menggambar vektor ekspresi pada EMT.

```
>plot2d("exp(x)",0,3); // plot fungsi aslinya, e^x  
>plot2d(mxm2str("deret"),>add,color=4:6): // plot ketiga deret taylor hampiran fungsi tersebut
```

```
Maxima said:
```

```
length: argument cannot be a symbol; found deret  
-- an error. To debug this try: debugmode(true);
```

```
mxmeval:
```

```
    return evaluate(mxm(s));
```

```
Try "trace errors" to inspect local variables after errors.
```

```
mxm2str:
```

```
    n=mxmeval("length(VVV)");
```

Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

```
>$deret[3]
```

```
deret3deret3
```

```
>plot2d(["exp(x)",&deret[1],&deret[2],&deret[3]],0,3,color=1:4):
```

```
deret is not a variable!
```

```
Error in expression: deret[1]
```

```
%ploteval:
```

```
    y0=f$(x[1],args());
```

```
Try "trace errors" to inspect local variables after errors.
```

```
plot2d:
```

```
    u=u_(%ploteval(xx[],t,args()));
```

```
>$sum(sin(k*x)/k,k,1,5)
```

```
sin(r(t-sint))+sin(2r(t-sint))2+sin(3r(t-sint))3+sin(4r(t-sint))4+sin(5r(t-sint))5sin[f0](r(t-sin[f0]t))  
+sin[f0](2r(t-sin[f0]t))2+sin[f0](3r(t-sin[f0]t))3+sin[f0](4r(t-sin[f0]t))4+sin[f0](5r(t-sin[f0]t))5
```

Berikut adalah cara menggambar kurva

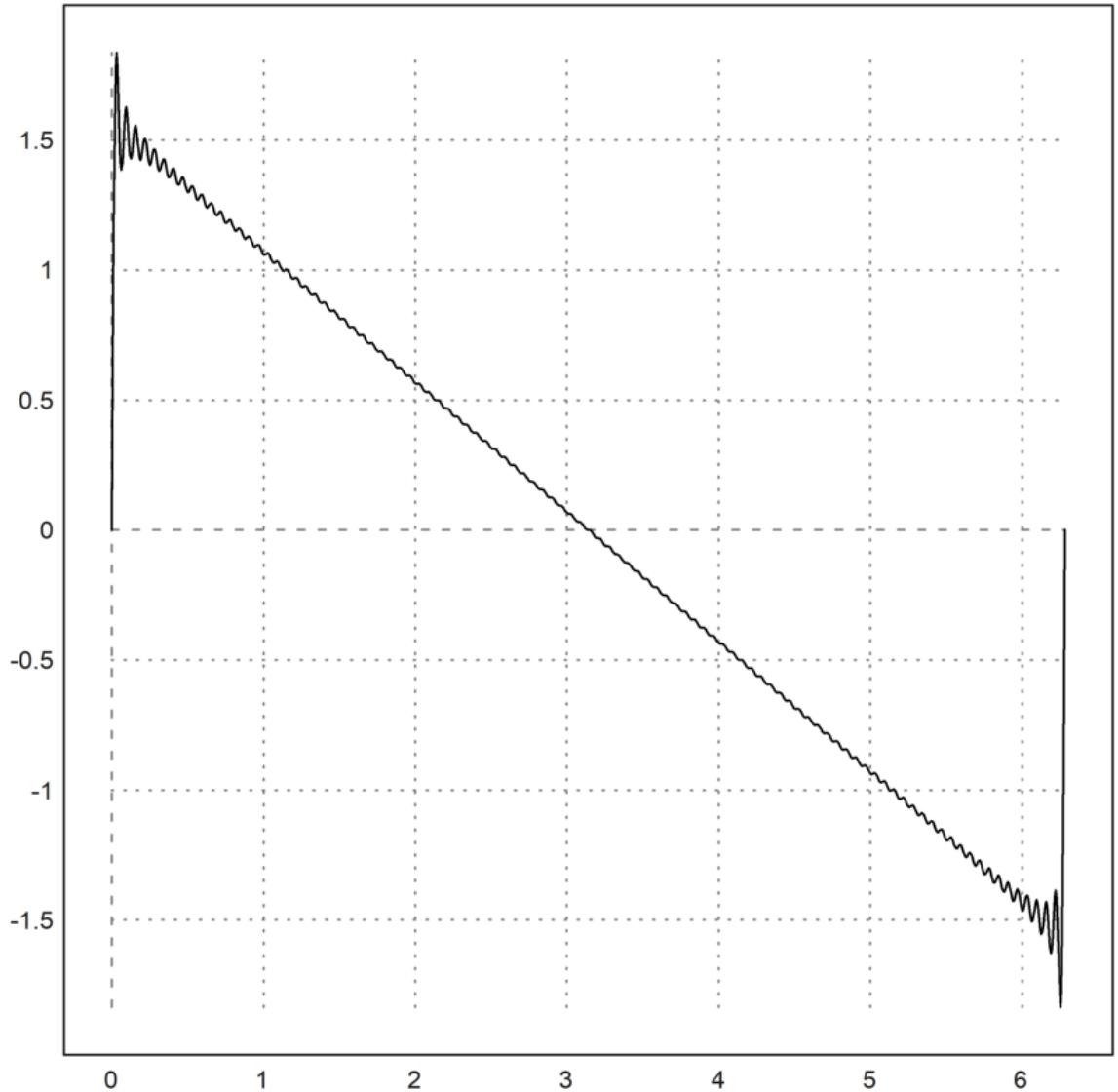
```
y=sin(x)+sin3x3+sin5x5+....y=sin[f0](x)+sin[f0]3x3+sin[f0]5x5+....
```

```
>plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi):
```

```

Variable or function t not found.
Error in expression: sin(r*(t-sin(t)))+sin(3*r*(t-
sin(t))/3+sin(5*r*(t-sin(t)))/5+sin(7*r*(t-sin(t)))/7+sin(9*r*(t-
sin(t))/9+sin(11*r*(t-sin(t)))/11+sin(13*r*(t-
sin(t))/13+sin(15*r*(t-sin(t)))/15+sin(17*r*(t-
sin(t))/17+sin(19*r*(t-sin(t)))/19+sin(21*r*(t-
sin(t))/21+sin(23*r*(t-sin(t)))/23+sin(25*r*(t-
sin(t))/25+sin(27*r*(t-sin(t)))/27+sin(29*r*(t-
sin(t))/29+sin(31*r*(t-sin(t)))/31+sin(33*r*(t-
sin(t))/33+sin(35*r*(t-sin(t)))/35+sin(37*r*(t-
sin(t))/37+sin(39*r*(t-sin(t)))/39+sin(41*r*(t-sin(t)))/41
%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());
Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan
kita akan menggambar kurva
y=Σk=1100sin(kx)k,0≤x≤2π.y=Σk=1100sin([/](kx)k,0≤x≤2π.
```

```
>x=linspace(0,2pi,1000); k=1:100; y=sum(sin(k*x')/k)'; plot2d(x,y):
```



## Tabel Fungsi

Terdapat cara menarik untuk menghasilkan barisan dengan ekspresi Maxima. Perintah `mxmtable()` berguna untuk menampilkan dan menggambarkan barisan dan menghasilkan barisan sebagai vektor kolom.

Sebagai contoh berikut adalah barisan turunan ke- $n$   $x^n$  di  $x=1$ .

```
>mxmtable("diffat(x^x,x=1,n)","n",1,8,frac=1);
Maxima said:
diff: second argument must be a variable; found r*(t-sin(t))
#0: diffat(expr=r^(r*(t-sin(t)))*(t-sin(t))^(r*(t-sin(t))),x=[r*(t-
sin(t)) = 1,1])
-- an error. To debug this try: debugmode(true);

%mxmtable:
return mxm("@expr,@var=@value")();
Try "trace errors" to inspect local variables after errors.
```

```

mxmtable:
y[#,1]=%mxmevtable(expr,var,x[#]);
>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // 
simpsum:menghitung deret secara simbolik

$$\sum_{k=1}^n k = n(1+n)2 \sum_{k=1}^n k = n(1+n)2$$


>$'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0,
inf),simpsum=true))

$$\sum_{k=0}^{\infty} 1k+3k = \sum_{k=0}^{\infty} 1k+3k \sum_{k=0}^{\infty} 1k+3k = \sum_{k=0}^{\infty} 1k+3k$$


Di sini masih gagal, hasilnya tidak dihitung.
>$'sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev:
menghitung nilai ekspresi

$$\sum_{x=1}^{\infty} 1x^2 = \pi^2 6 \sum_{x=1}^{\infty} 1x^2 = \pi^2 6$$


>$'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1,
inf),simpsum=true))

$$\sum_{k=1}^{\infty} (-1)^{-1+k}k = -\sum_{x=1}^{\infty} (-1)x \sum_{k=1}^{\infty} (-1)^{-1+k}k = -\sum_{x=1}^{\infty} (-1)xx$$


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))

$$\sum_{k=1}^{\infty} (-1)^{k-1+2k} = \sum_{k=1}^{\infty} (-1)^{k-1+2k} \sum_{k=1}^{\infty} (-1)^{k-1+2k} = \sum_{k=1}^{\infty} (-1)^{k-1+2k}$$


>$ev(sum(1/n!, n, 0, inf),simpsum=true)

$$\sum_{n=0}^{\infty} 1n! \sum_{n=0}^{\infty} 1n!$$


Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.
>&assume(abs(x)<1); $'sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0,
inf),simpsum=true), &forget(abs(x)<1);
Answering "Is -1+abs(-r*t+r*sin(t)) positive, negative or zero?" with
"positive"
Maxima said:
sum: sum is divergent.
-- an error. To debug this try: debugmode(true);

Error in:
... k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forget(abs ...

$$\sum_{k=0}^{\infty} rk(t-sint)kk! = \sum_{k=0}^{\infty} rk(t-sint)kk! \sum_{k=0}^{\infty} rk(t-\sin[t])kk! = \sum_{k=0}^{\infty} rk(t-\sin[t])kk!$$


Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1.
>$'sum(x^k/k!,k,0,inf)=ev(sum(x^k/k!,k,0,inf),simpsum=true)

$$\sum_{k=0}^{\infty} rk(t-sint)kk! \lim_{n \rightarrow \infty} \sum_{k=0}^n rk(t-\sin[t])kk!$$


>$limit(sum(x^k/k!,k,0,n),n,inf)

$$\lim_{n \rightarrow \infty} \sum_{k=0}^n rk(t-sint)kk! \lim_{n \rightarrow \infty} \sum_{k=0}^n rk(t-\sin[t])kk!$$


>function d(n) &= sum(1/(k^2-k),k,2,n); $'d(n)=d(n)

$$d(n) = \sum_{k=2}^n 1-k + k2d(n) = \sum_{k=2}^n 1-k + k2$$


>$d(10)=ev(d(10),simpsum=true)

$$\sum_{k=2}^{10} 1-k + k2 = 910 \sum_{k=2}^{10} 1-k + k2 = 910$$


>$d(100)=ev(d(100),simpsum=true)

$$\sum_{k=2}^{100} 1-k + k2 = 99100 \sum_{k=2}^{100} 1-k + k2 = 99100$$


```

## Deret Taylor

Deret Taylor suatu fungsi  $f$  yang diferensiabel sampai tak hingga di sekitar  $x=a$  adalah:

$$f(x) = \sum_{k=0}^{\infty} (x-a)^k f^{(k)}(a) k! . f(x) = \sum_{k=0}^{\infty} (x-a)^k f^{(k)}(a) k! .$$

```
>$'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai suku ke-11
```

```
Maxima said:  
taylor: r*(t-sin(t)) cannot be a variable.  
-- an error. To debug this try: debugmode(true);
```

```
Error in:  
$'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x= ...  
^
```

```
>$'log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1
```

```
Maxima said:  
taylor: r*(t-sin(t)) cannot be a variable.  
-- an error. To debug this try: debugmode(true);
```

```
Error in:  
$'log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1 ...  
^
```

# EMT\_Geometry\_Adiyatma\_2303063006

## 2\_MatB

---

Adiyatma  
23030630062  
Matematika B

### Visualisasi dan Perhitungan Geometri dengan EMT

---

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan geometri tersebut disimpan di dalam file program "geometry.e", sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

```
>load geometry
```

Numerical and symbolic geometry.

### Fungsi-fungsi Geometri

---

Fungsi-fungsi untuk Menggambar Objek Geometri:

```
defaultd:=textheight()*1.5: nilai asli untuk parameter d
setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang koordinat
setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas sumbu-x dan y adalah -r sd r
plotPoint (P, "P"): menggambar titik P dan diberi label "P"
plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label "AB" sejauh d
plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d
plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"
plotLabel (label, P, V, d): menuliskan label pada posisi P
```

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

```
turn(v, phi): memutar vektor v sejauh phi
turnLeft(v): memutar vektor v ke kiri
turnRight(v): memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang vektorv dan w.
lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh.
ax+by=c.
lineWithDirection(A,v): garis melalui A searah vektor v
getLineDirection(g): vektor arah (gradien) garis g
getNormal(g): vektor normal (tegak lurus) garis g
getPointOnLine(g): titik pada garis g
perpendicular(A, g): garis melalui A tegak lurus garis g
parallel (A, g): garis melalui A sejajar garis g
lineIntersection(g, h): titik potong garis g dan h
projectToLine(A, g): proyeksi titik A pada garis g
distance(A, B): jarak titik A dan B
distanceSquared(A, B): kuadrat jarak A dan B
quadrance(A, B): kuadrat jarak A dan B
areaTriangle(A, B, C): luas segitiga ABC
```

```

computeAngle(A, B, C): besar sudut <ABC
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/atas) garis
quad(A,B): kuadrat jarak AB
spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c,
yakni  $\sin(\alpha)^2$  dengan
alpha sudut yang menghadap sisi a.
crosslaw(a,b,c,sa): persamaan 3 quads dan 1 spread pada segitiga
dengan panjang sisi a, b,
c.
triplespread(sa,sb,sc): persamaan 3 spread sa,sb,sc yang memebntuk
suatu segitiga
doublespread(sa): Spread sudut rangkap Spread  $2\phi$ , 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 tetapkan tiga poin dan plot mereka.
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
>B=[0,1]; plotPoint(B,"B");
>C=[2,2]; plotPoint(C,"C");
Kemudian tiga segmen.
>plotSegment(A,B,"c"); // c=AB
>plotSegment(B,C,"a"); // a=BC
>plotSegment(A,C,"b"); // b=AC
Fungsi geometri meliputi fungsi untuk membuat garis dan lingkaran.
Format garis adalah [a,b,c], yang mewakili garis dengan persamaan
ax+by=c.
>lineThrough(B,C) // garis yang melalui B dan C
[-1, 2, 2]
Hitunglah garis tegak lurus yang melalui A pada BC.
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
Dan persimpangannya dengan BC.
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
Plot itu.
>plotPoint(D,value=1); // koordinat D ditampilkan
>aspect(1); plotSegment(A,D); // tampilkan semua gambar hasil plot...()
```

Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2} AD \cdot BC$$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
1.5
Bandingkan dengan rumus determinan.
>areaTriangle(A,B,C) // hitung luas segitiga langsung dengan fungsi
1.5
Cara lain menghitung luas segitiga ABC:
>distance(A,D)*distance(B,C)/2
1.5
Sudut di C
>degsprint(computeAngle(B,C,A))
36°52'11.63"
Sekarang lingkaran luar segitiga.
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
>R=getCircleRadius(c); // jari-jari lingkaran luar
>O=getCircleCenter(c); // titik pusat lingkaran c
>plotPoint(O,"O"); // gambar titik "O"
>plotCircle(c,"Lingkaran luar segitiga ABC"):
    Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.
>O, R
[1.16667, 1.16667]
1.17851130198
Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat
lingkaran dalam adalah
titik potong garis-garis bagi sudut.
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
[0.86038, 0.86038]
Tambahkan semuanya ke plot.
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi
sudut
>plotPoint(P,"P"); // gambar titik potongnya
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
0.509653732104
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"): //
gambar lingkaran dalam
```

## Latihan

---

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

```
>setPlotRange(-2.5,4.5,-2.5,4.5);
>A=[-2,1]; plotPoint(A,"A");
>B=[1,-2]; plotPoint(B,"B");
>C=[4,4]; plotPoint(C,"C");
2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.
>plotSegment(A,B,"c")
>plotSegment(B,C,"a")
>plotSegment(A,C,"b")
>aspect(1):
3. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.
>l=angleBisector(A,C,B);
>g=angleBisector(C,A,B);
>P=lineIntersection(l,g)
[0.581139, 0.581139]
```

```

>color(5); plotLine(l); plotLine(g); color(1);
>plotPoint(P,"P");
>r=norm(P-projectToLine(P,lineThrough(A,B)))
1.52896119631
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):
Jadi, terbukti bahwa garis bagi sudut yang ketiga juga melalui titik
pusat lingkaran dalam.

```

4. Gambar jari-jari lingkaran dalam.

```

>r=norm(P-projectToLine(P,lineThrough(A,B)))
1.52896119631
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):

```

## Contoh 2: Geometri Simbolik

---

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometri.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, kita dapat menggunakan perhitungan simbolis sekarang.

```

>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi
memberikan perhitungan simbolis.
>c &= lineThrough(B,C) // c=BC
[- 1, 2, 2]

```

Kita bisa mendapatkan persamaan garis dengan mudah.

```

>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
2y-x=22y-x=2
[y=x2+1][y=x2+1]

```

```

>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A
dan (x1, y1)
(x1-1)y-xy1=-y1(x1-1)y-xy1=-y1

```

```

>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
[2, 1, 2]

```

```

>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
2   6
[-, -]
5   5

```

```

>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
[25,65][25,65]

```

```

>$distance(A,Q) // jarak AQ
35-√35

```

```

>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran
melalui A, B, C
[76,76,532-√][76,76,532]

```

```

>r=>getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
532-√532

```

1.1785113019775791.178511301977579

```

>$computeAngle(A,C,B) // nilai <ACB

```

```

arccos(45)arccos[45](45)

>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis
bagi <ACB
y=x

>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P //
titik potong 2 garis bagi sudut
[2-√5-√+26,2-√5-√+26][25+26,25+26]

>P() // hasilnya sama dengan perhitungan sebelumnya
[0.86038, 0.86038]

```

## Garis dan Lingkaran yang Berpotongan

---

Tentu saja, kita juga dapat memotong garis dengan lingkaran, dan lingkaran dengan lingkaran.

```

>A &:= [1,0]; c=circleWithCenter(A,4);
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l);
Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah
titik potong.
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2,
[4.64575, -1.64575]
[-0.645751, 3.64575]
>plotPoint(P1); plotPoint(P2):
Begitu pula 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
[[7-√+2,1-7-√],[2-7-√,7-√+1]][[7+2,1-7],[2-7,7+1]]

```

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);
>degprint(computeAngle(P1,C,P2))
69°17'42.68"
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C);
plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
69°17'42.68"
>insimg;

```

## Garis Sumbu

---

Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B.
  2. Gambar lingkaran dengan pusat B melalui A.
  3. Tarik garis melalui kedua titik potong kedua lingkaran tersebut.
- Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

```

>A=[2,2]; B=[-1,-2];
>c1=circleWithCenter(A,distance(A,B));
>c2=circleWithCenter(B,distance(A,B));
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(5); plotCircle(c1); plotCircle(c2);
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):
Selanjutnya, kami melakukan hal yang sama di Maxima dengan koordinat umum.

>A &= [a1,a2]; B &= [b1,b2];
>c1 &= circleWithCenter(A,distance(A,B));
>c2 &= circleWithCenter(B,distance(A,B));
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];
Persamaan untuk persimpangan cukup terlibat. Tetapi kita dapat menyederhanakannya, jika kita memecahkan y.
>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)
[y=-(2b1-2a1)x+b22+b12-a22-a122b2-2a2][y=-(2b1-2a1)x+b22+b12-a22-a122b2-2a2]

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sama sekali berbeda.

>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
[y=-(2b1-2a1)x+b22+b12-a22-a122b2-2a2][y=-(2b1-2a1)x+b22+b12-a22-a122b2-2a2]

>h &=getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
[y=(b2-a2)x-a1b2+a2b1b1-a1][y=(b2-a2)x-a1b2+a2b1b1-a1]

Perhatikan hasil kali gradien garis g dan h adalah:
-(b1-a1)(b2-a2)×(b2-a2)(b1-a1)=-1.-(b1-a1)(b2-a2)×(b2-a2)(b1-a1)=-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:

$$L=s(s-a)(s-b)(s-c)\sqrt{\quad}$$
 dengan  $s=(a+b+c)/2$ .  $L=s(s-a)(s-b)(s-c)$  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

$$L_{\Delta ABC} = \frac{1}{2}ax \cdot y.$$

Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2+y^2=b^2, (x-a)^2+y^2=c^2, x^2+y^2=b^2, (x-a)^2+y^2=c^2.$$

```
>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 ...
^

Kami mendapatkan rumus Heron.
```

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H(a,b,[1,0,4]) = |a||ysol|/2$$

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 juga jelas, bahwa ini adalah segitiga dengan luas maksimal dan dua sisi 3 dan 4.

```
>aspect(1.5); plot2d(&H(3,4,x),1,7); // Kurva luas segitiga sengang panjang sisi 3, 4, x (1 <= x <= 7)
```

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 ellips.

```
>s1 &= subst(d-c,b,sol[2]); $s1
```

Maxima said:

part: invalid index of list or matrix.

-- an error. To debug this try: debugmode(true);

Error in:

```
s1 &= subst(d-c,b,sol[2]); $s1 ...  
^
```

Dan buat fungsi ini.

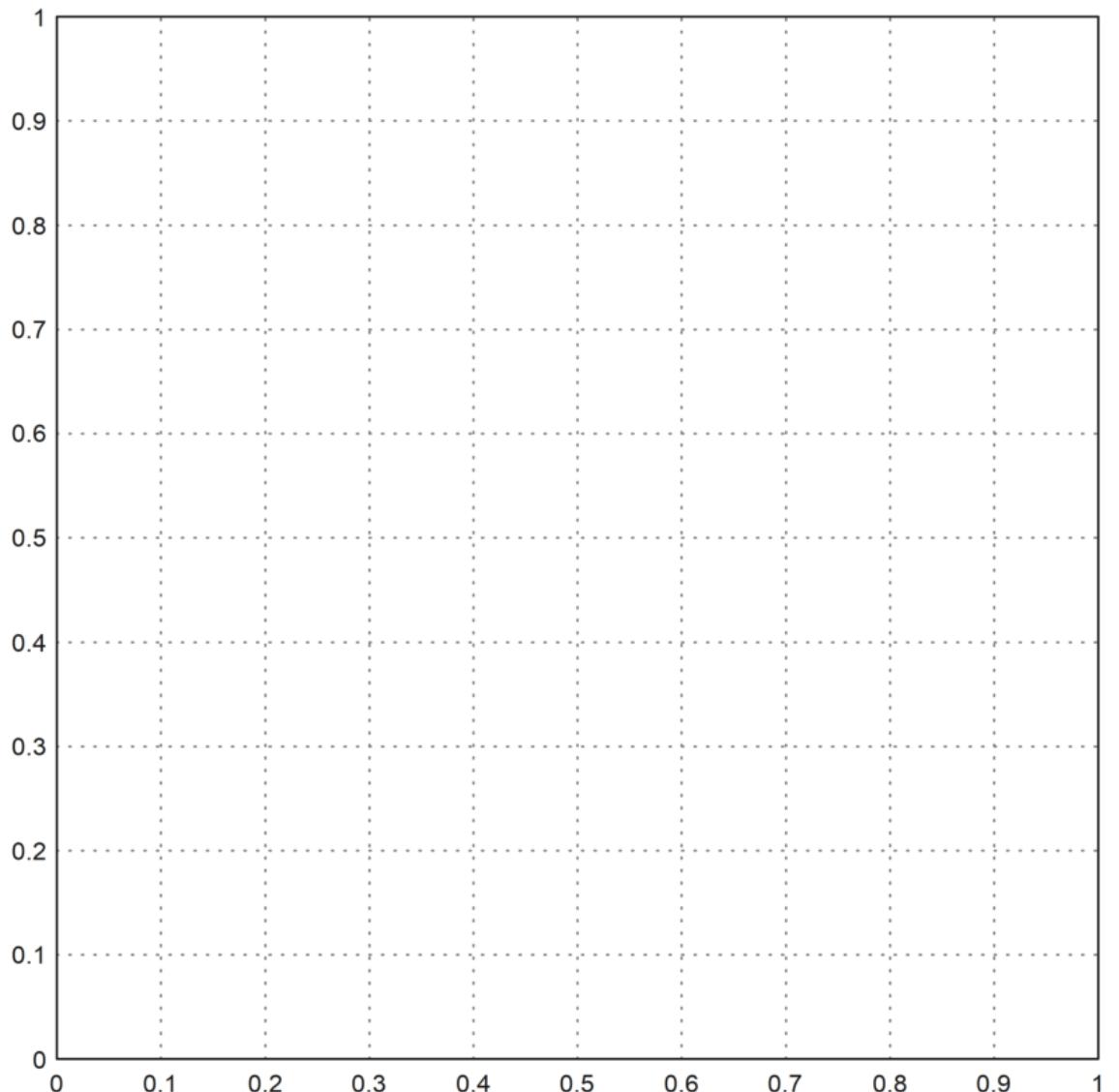
```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &=  
rhs(s1[2]); $fy(a,c,d)
```

00

0

Sekarang kita bisa menggambar setnya. Sisi b bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan ellips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```



Kita dapat memeriksa persamaan umum untuk elips ini, yaitu.

$$(x-x_m)^2u^2 + (y-y_m)^2v^2 = 1, \quad (x-x_m)^2u^2 + (y-y_m)v^2 = 1,$$

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])
```

$$a^2d^2a^2d^2$$

Kita lihat bahwa tinggi dan luas segitiga adalah maksimal untuk  $x=0$ . Jadi luas segitiga dengan  $a+b+c=d$  maksimal jika segitiga sama sisi. Kami ingin menurunkan ini secara analitis.

```
> eqns &= [diff(H(a,b,d-(a+b))^2,a)=0, diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
      [aysol2=0,0=0][aysol22=0,0=0]
```

Kami mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusinya  $a=b=c=d/3$ .

```
> $solve(eqns,[a,b])
[[a=0,b=%r1]][[a=0,b=%r1]]
```

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)=lb, ...]
```

```

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-tama atur poin di Maxima.

>A &= at([x,y],sol[2]); $A
Maxima said:
part: invalid index of list or matrix.
-- an error. To debug this try: debugmode(true);

Error in:
A &= at([x,y],sol[2]); $A ...
^

>B &= [0,0]; $B, C &= [a,0]; $C
[a,0][a,0]

```

$$[a, 0]$$

Kemudian atur rentang plot, dan plot titik-titiknya.

```

>setPlotRange(0,5,-2,3); ...
a=4; b=3; c=2; ...
plotPoint(mxmeval("B"),"B"); plotPoint(mxmeval("C"),"C"); ...
plotPoint(mxmeval("A"),"A"):

Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
    return evaluate(mxm(s));
Error in:
... otPoint(mxmeval("C"),"C"); plotPoint(mxmeval("A"),"A")): ...
^

Plot segmen.

```

```

>plotSegment(mxmeval("A"),mxmeval("C")); ...
plotSegment(mxmeval("B"),mxmeval("C")); ...
plotSegment(mxmeval("B"),mxmeval("A")):

Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
    return evaluate(mxm(s));
Error in:
plotSegment(mxmeval("A"),mxmeval("C")); plotSegment(mxmeval("B") ...
^

Hitung tegak lurus tengah di Maxima.

```

```

>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
Dan pusat lingkaran.
>U &= lineIntersection(h,g);
Kami mendapatkan rumus untuk jari-jari lingkaran.
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
a22+a12-----sqrt(a22+a12-2aa1+a2)-----sqrt(2|a2|a22+a12a22+a12-2aa1+a22|a2|)

Mari kita tambahkan ini ke plot.
>plotPoint(U()); ...

```

```

plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
Variable a2 not found!
Use global variables or parameters for string evaluation.
Error in ^
Error in expression: [a/2, (a2^2+a1^2-a*a1)/(2*a2)]
Error in:
plotPoint(U()); plotCircle(circleWithCenter(mxmeval("U"),mxmev ...
^

Menggunakan geometri, kami memperoleh rumus sederhana
 $\sin(\alpha) = 2r \sin(\frac{\alpha}{2})$  atau  $\sin(\alpha) = 2r$ 

untuk radiusnya. Kami dapat memeriksa, apakah ini benar dengan
Maxima.
Maxima akan memfaktorkan ini hanya jika kita kuadratkan.
>$c^2/sin(computeAngle(A,B,C))^2 | factor
[a22+a12a22, 0, 16(a22+a12)a22][a22+a12a22, 0, 16(a22+a12)a22]

```

## Contoh 4: Garis Euler dan Parabola

---

Garis Euler adalah garis yang ditentukan dari sembarang segitiga yang tidak sama sisi. Ini adalah garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk orthocenter, circumcenter, centroid, titik Exeter dan pusat lingkaran sembilan titik segitiga.

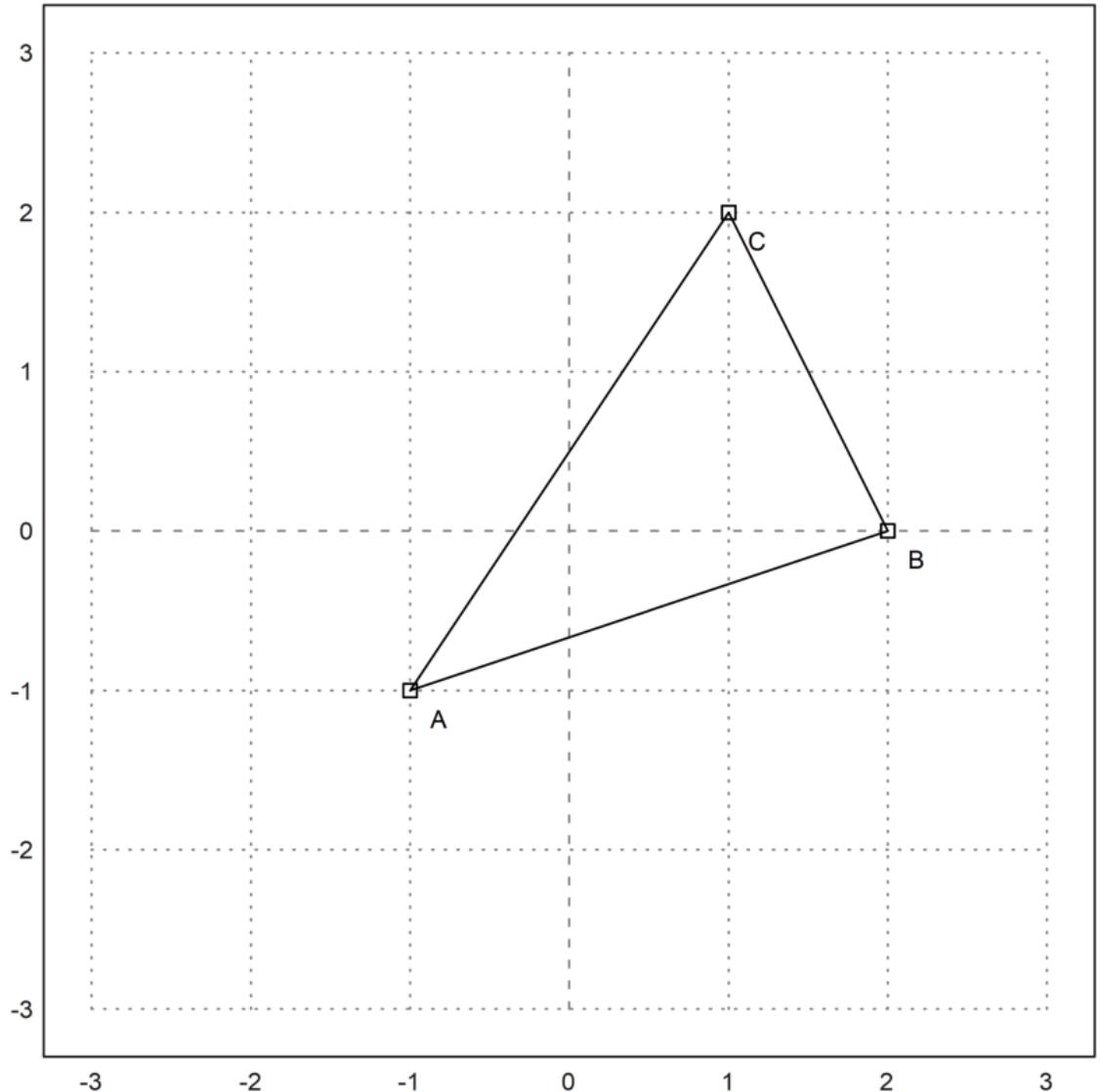
Untuk demonstrasi, kami menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kita mendefinisikan sudut-sudut segitiga di Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolis.

```

>A:=[-1,-1]; B:=[2,0]; C:=[1,2];
Untuk memplot objek geometris, kami menyiapkan area plot, dan
menambahkan titik ke sana. Semua plot objek geometris ditambahkan ke
plot saat ini.
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
Kita juga bisa menambahkan sisi segitiga.
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,"");

```



Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

```
>$areaTriangle(A,B,C)
```

-72-72

Kita dapat menghitung koefisien 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 sebuah titik, sehingga titik

tersebut berada di sisi positif dari bentuk Hesse. Memasukkan titik menghasilkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$710-\sqrt{710}$

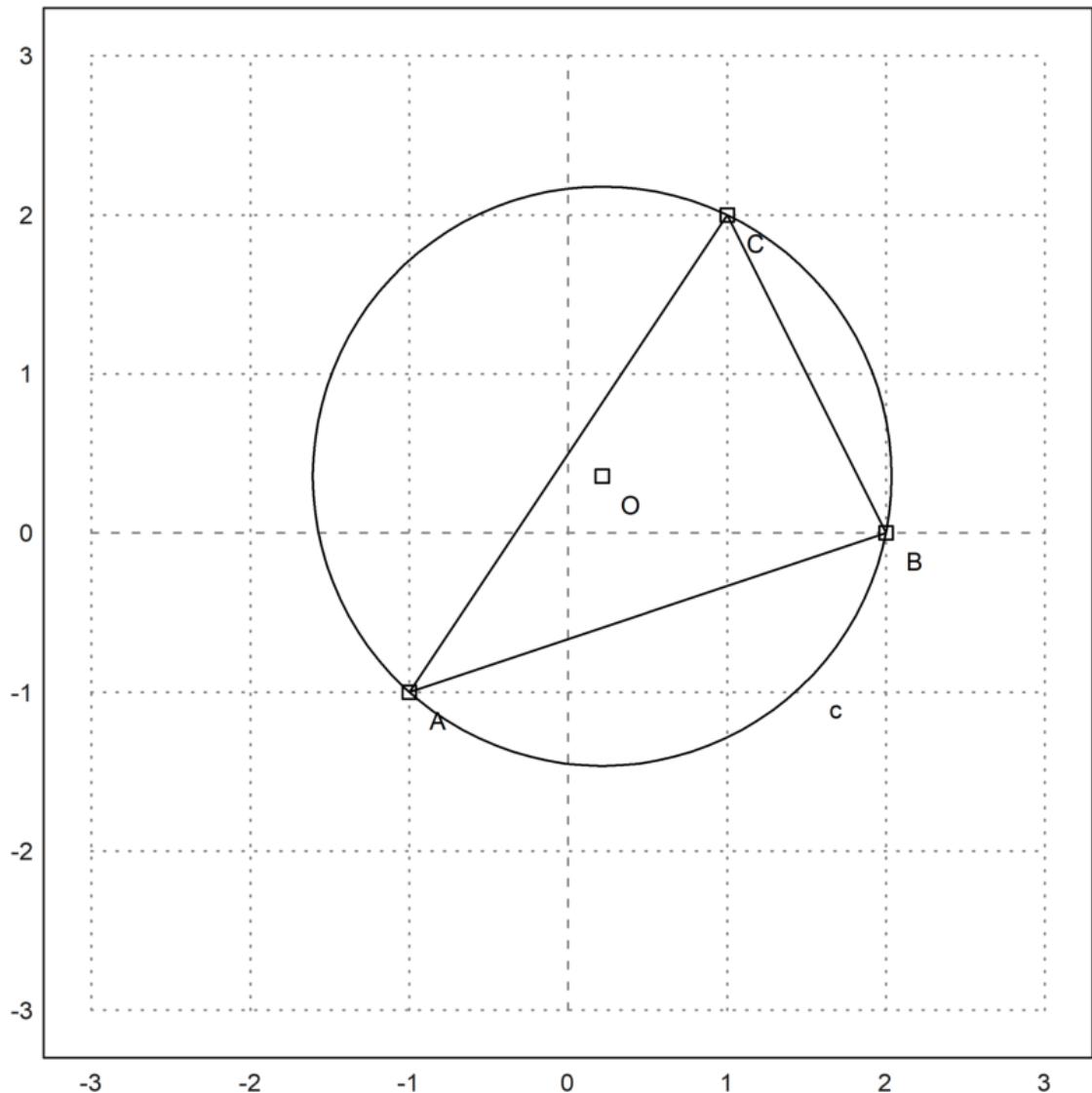
$$\frac{7}{\sqrt{10}}$$

Sekarang kita hitung lingkaran luar ABC.

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
(y-514)^2+(x-314)^2=32598(y-514)^2+(x-314)^2=32598
>O &= getCircleCenter(LL); $O
[314,514][314,514]
```

Gambarkan lingkaran dan pusatnya. Cu dan U adalah simbolis. Kami mengevaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O");
```



Kita dapat menghitung perpotongan ketinggian di ABC (orthocenter) secara numerik dengan perintah berikut.

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...
```

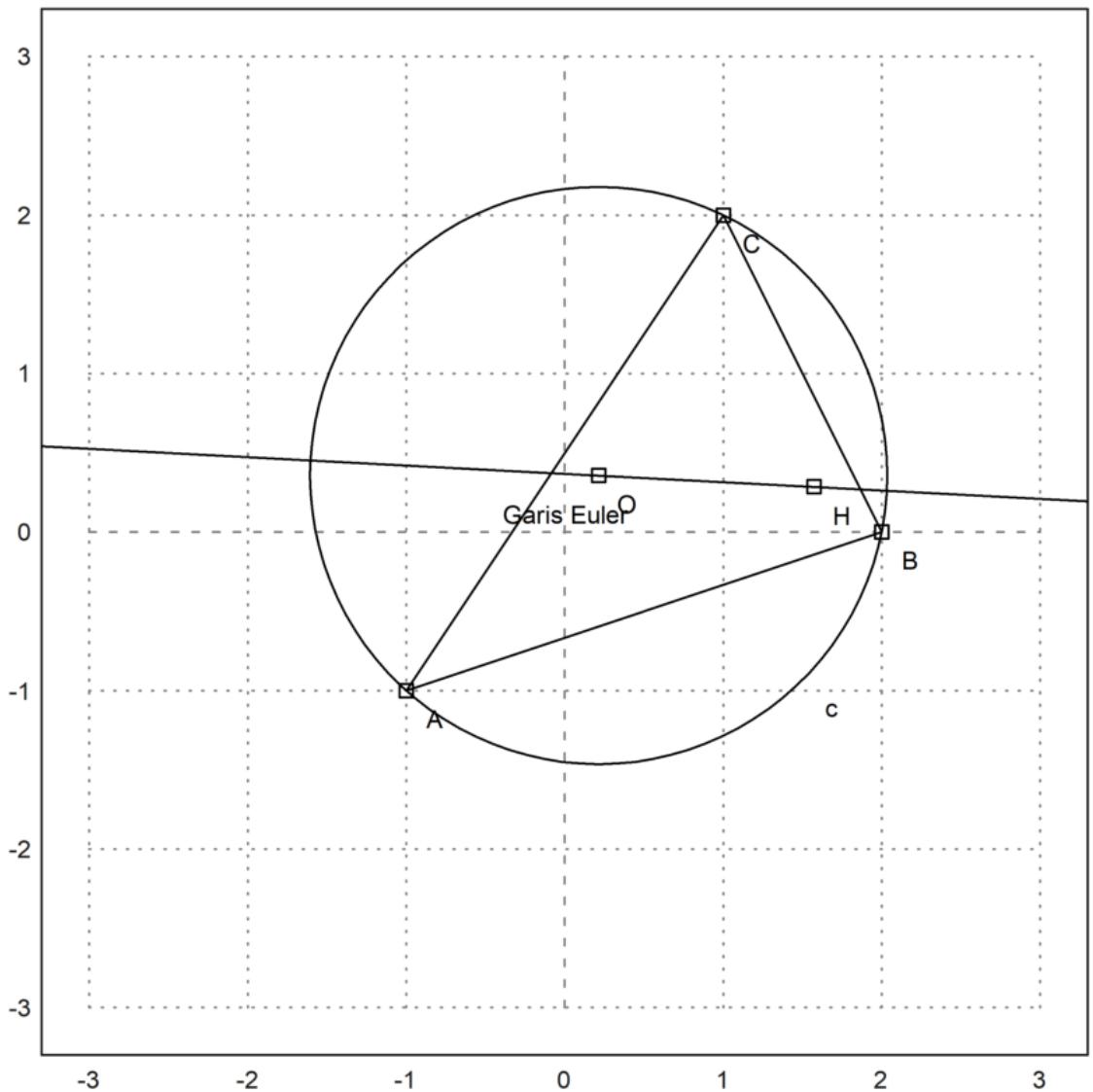
```
perpendicular(B,lineThrough(A,C))); $H
[117,27][117,27]
```

Sekarang kita dapat menghitung garis Euler dari segitiga.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
-19y14-x14=-12-19y14-x14=-12
```

Tambahkan ke plot kami.

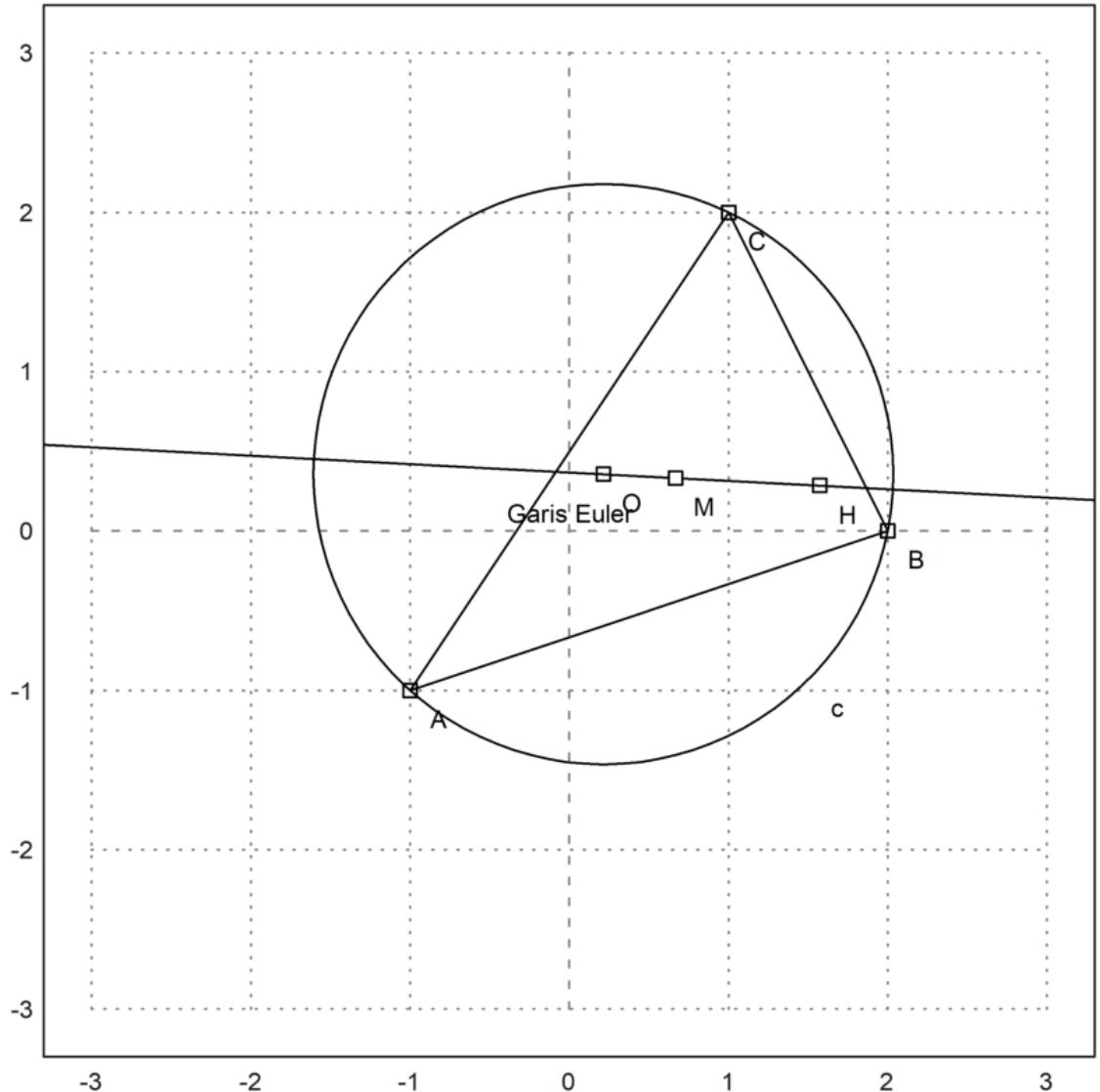
```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```



Pusat gravitasi harus berada di garis ini.

```
>M &= (A+B+C)/3; $getLineEquation(el,x,y) with [x=M[1],y=M[2]]
-12=-12-12=-12
```

```
>plotPoint(M(),"M"); // titik berat
```



Teorinya memberitahu kita  $MH=2*MO$ . Kita perlu menyederhanakan dengan radcan untuk mencapai ini.

```
> $distance(M, H) / $distance(M, O) | radcan
```

22

Fungsi termasuk fungsi untuk sudut juga.

```
> $computeAngle(A, C, B), degprint(%())
```

$$\arccos(45 - \sqrt{13} - \sqrt{5})$$

$60^\circ 15' 18.43''$

Persamaan untuk pusat incircle tidak terlalu bagus.

```
> Q &= lineIntersection(angleBisector(A, C, B), angleBisector(C, B, A)) | radcan;
```

\$Q

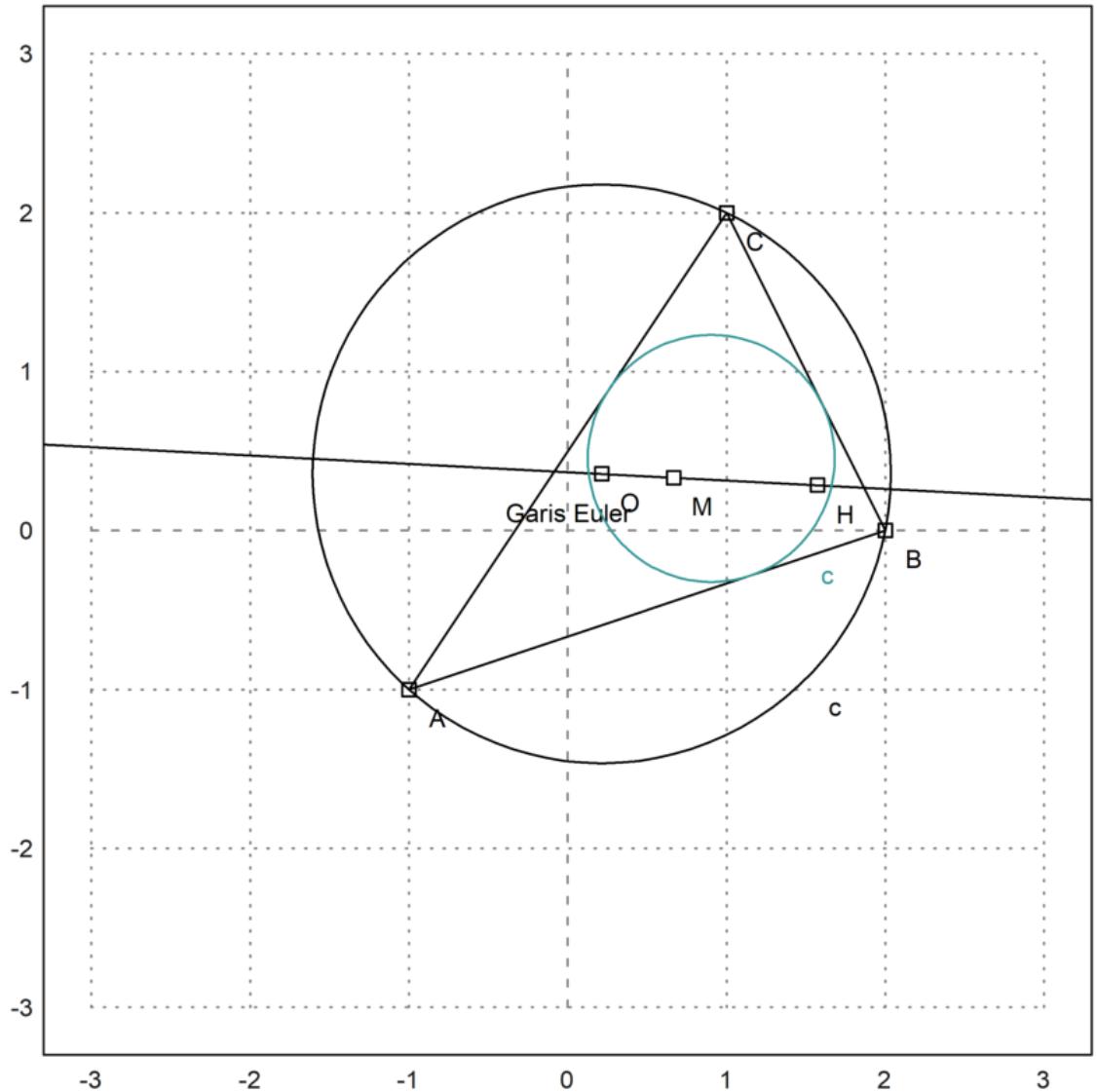
$$\frac{[(2\sqrt{3}+1)\sqrt{5}-\sqrt{13}-\sqrt{-152}-\sqrt{314}, (2-\sqrt{-3})\sqrt{5}-\sqrt{13}-\sqrt{+52}\sqrt{3}+514]}{[(2\sqrt{3}+1)\sqrt{5}\sqrt{13}-152+\sqrt{314}, (2-\sqrt{-3})\sqrt{5}\sqrt{13}+52\sqrt{3}+514]}$$

Mari kita hitung juga ekspresi untuk jari-jari lingkaran yang tertulis.

```
> r &= distance(Q, projectToLine(Q, lineThrough(A, B))) | ratsimp; $r
```

$$\frac{(-412-\sqrt{-31})5-\sqrt{13}-\sqrt{+1152-\sqrt{+614}}}{\sqrt{(-412-31)513+1152+61472}} - \sqrt{72-}$$

```
>LD &= circleWithCenter(Q,r); // Lingkaran dalam
   Mari kita tambahkan ini ke plot.
>color(5); plotCircle(LD());
```



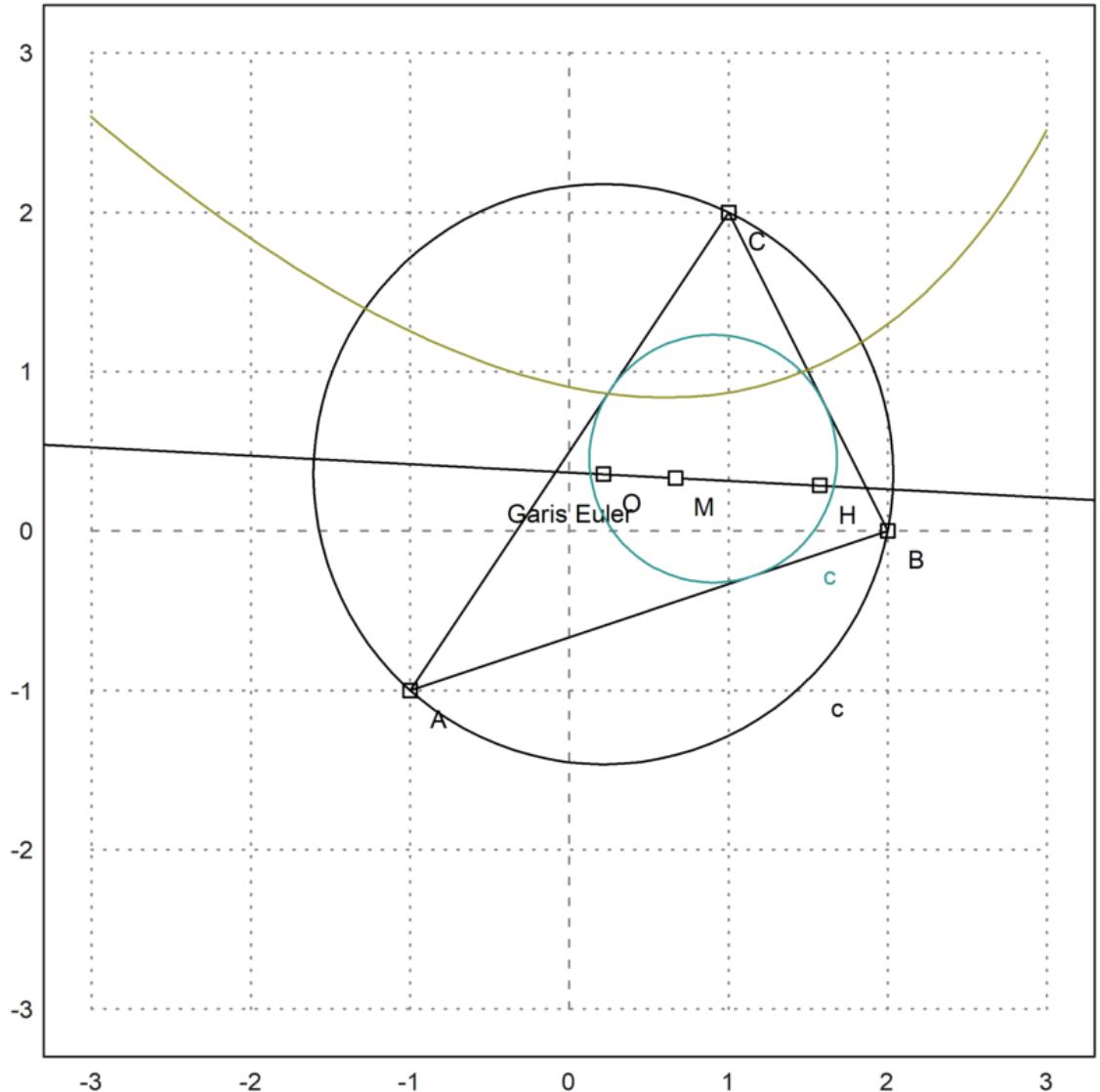
## Parabola

Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
   3y-x+210--\sqrt{-(2-y)^2+(1-x)^2}-----\sqrt{=03y-x+210-(2-y)^2+(1-x)^2=0}
```

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```



Ini seharusnya menjadi beberapa fungsi, tetapi pemecah default Maxima hanya dapat menemukan solusinya, jika kita kuadratkan persamaannya. Akibatnya, kami mendapatkan solusi palsu.

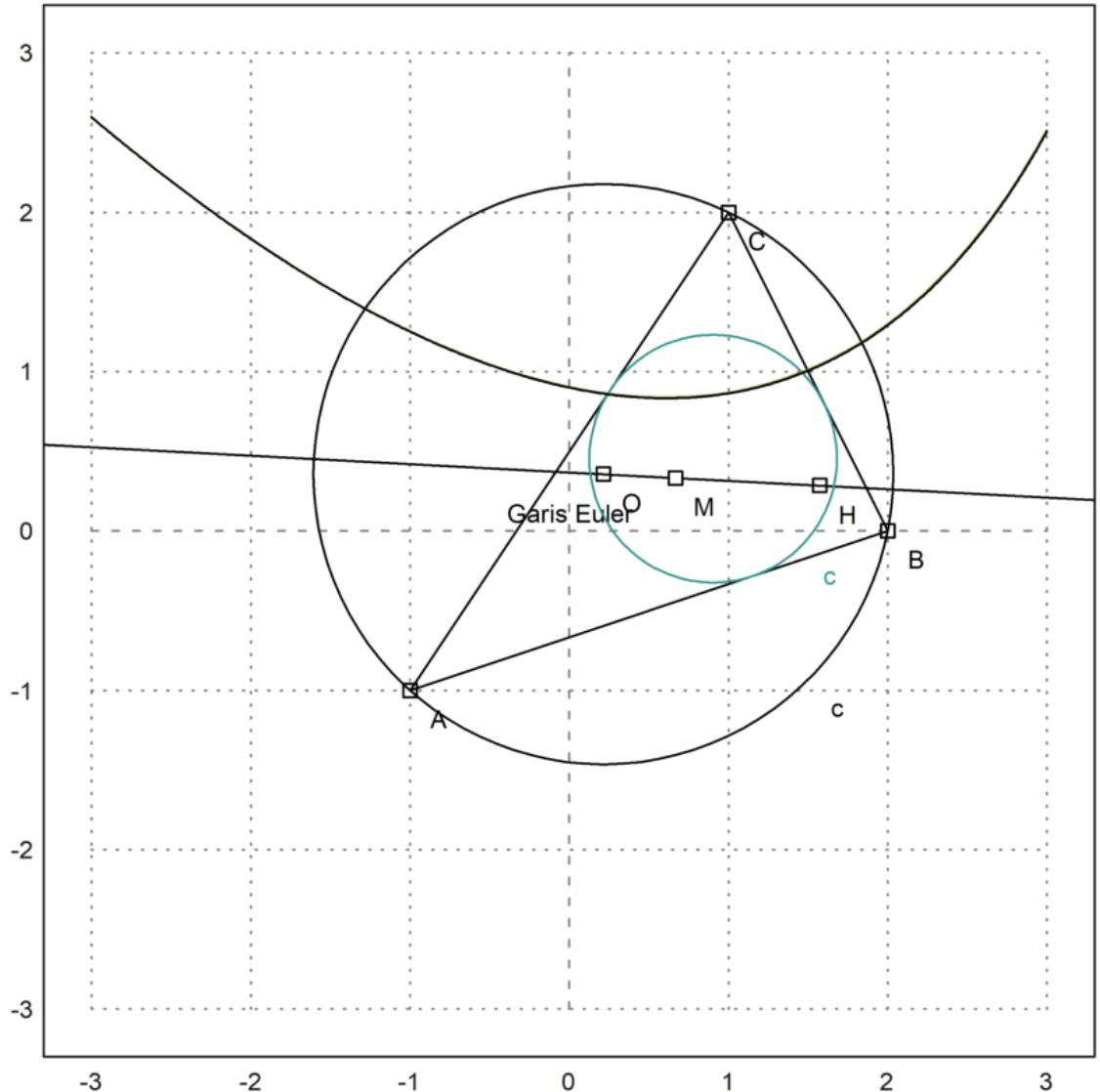
```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-
distance([x,y],C)^2,y)
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26,
y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 26]
```

Solusi pertama adalah

```
maxima: akar[1]
```

Menambahkan solusi pertama ke plot menunjukkan, bahwa itu memang jalan yang kita cari. Teorinya memberi tahu kita bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1):
```



```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan
kurva di atas
g(x)=-3x-70--sqrt(9-2x)-----sqrt(26)g(x)=-3x-709-2x+26
```

```
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
2.1356057793390612.135605779339061
```

$$2.135605779339061$$

```
>U &= projectToLine(T, lineThrough(A,B)); $U // proyeksi T pada garis AB
[80-311--sqrt(70)--sqrt(10), 20-11--sqrt(70)--sqrt(10)][80-3117010, 20-117010]
```

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jarak T ke AB
2.1356057793390612.135605779339061
```

$$2.135605779339061$$

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

## Contoh 5: Trigonometri Rasional

---

Ini terinspirasi dari ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti pengertian klasik tentang jarak dan sudut dengan kuadrat dan penyebaran. Dengan menggunakan ini, memang mungkin untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

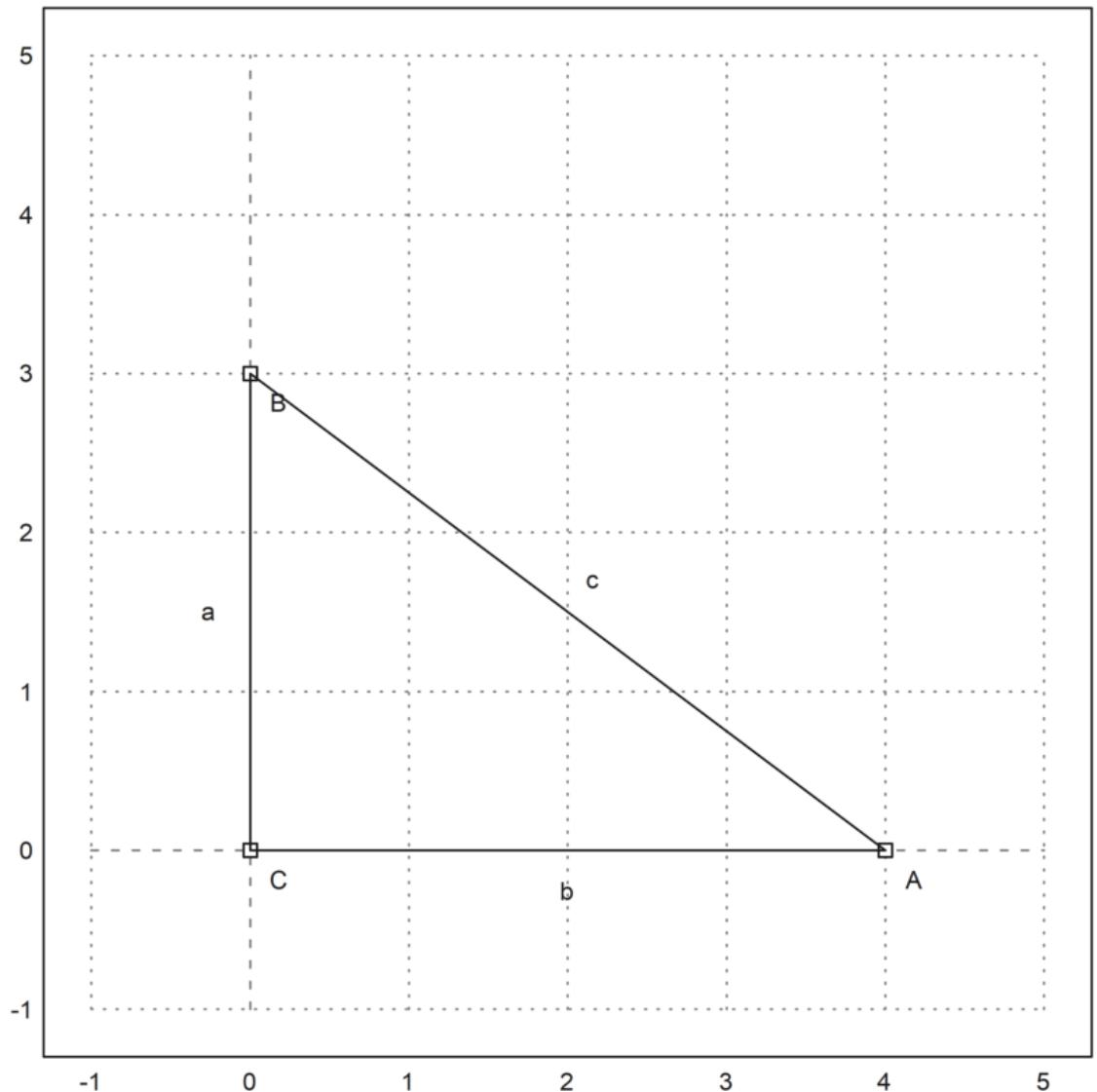
Berikut ini, saya memperkenalkan konsep, dan memecahkan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keuntungan utama dari trigonometri rasional bahwa perhitungan hanya dapat dilakukan dengan kertas dan pensil. Anda diundang untuk memeriksa hasil tanpa komputer.

Intinya adalah bahwa perhitungan rasional simbolis sering kali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

```
>load geometry;
```

Untuk pengenalan pertama, kami menggunakan segitiga persegi panjang dengan proporsi Mesir terkenal 3, 4 dan 5. Perintah berikut adalah perintah Euler untuk merencanakan geometri bidang yang terdapat dalam file Euler "geometry.e".

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
setPlotRange(-1,5,-1,5); ...
plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
insimg(30);
```



Tentu saja,

$$\sin(w_a) = \frac{a}{c}, \sin(\text{arcsin}(w_a)) = \frac{a}{c},$$

di mana  $w_a$  adalah sudut di A. Cara yang biasa untuk menghitung sudut ini, adalah dengan mengambil invers dari fungsi sinus. Hasilnya adalah

sudut yang tidak dapat dicerna, yang hanya dapat dicetak kira-kira.

```
>wa := arcsin(3/5); degprint(wa)
```

$36^{\circ}52'11.63''$

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadran, yang menggantikan jarak. Sebenarnya, itu hanya jarak kuadrat. Berikut ini,  $a$ ,  $b$ , dan  $c$  menunjukkan kuadrat dari sisi-sisinya.

Teorema Pythagoras menjadi  $a+b=c$ .

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
25 = 25
```

Pengertian kedua dari trigonometri rasional adalah penyebaran. Spread

mengukur pembukaan antar baris. Ini adalah 0, jika garis-garisnya sejajar, dan 1, jika garis-garisnya persegi panjang. Ini adalah kuadrat sinus sudut antara dua garis.

Penyebaran garis AB dan AC pada gambar di atas didefinisikan sebagai:

$$sa = \sin(\alpha) = ac, sa = \sin(\beta) = bc,$$

di mana a dan c adalah kuadrat dari sembarang segitiga siku-siku dengan salah satu sudut di A.

```
>sa &= a/c; $sa
```

$$925925$$

Ini lebih mudah dihitung daripada sudut, tentu saja. Tetapi Anda kehilangan properti bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengonversi nilai perkiraan untuk sudut wa menjadi sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

$$9/25$$

Hukum kosinus trigonometri klasik diterjemahkan menjadi "hukum silang" berikut.

$$(c+b-a)^2 = 4bc(1-sa) \quad (c+b-a)^2 = 4bc(1-sa)$$

Di sini a, b, dan c adalah kuadrat dari sisi-sisi segitiga, dan sa adalah penyebaran sudut A. Sisi a, seperti biasa, berhadapan dengan sudut A.

Hukum ini diimplementasikan dalam file geometri.e yang kami muat ke Euler.

```
>$crosslaw(aa,bb,cc,sa)
```

$$\sqrt{[bb-aa+76]^2, [bb-aa+76]^2, [bb-aa+532-aa]^2] = [14bb(1-sa)^3, 14bb(1-sa)^3, 5232bb(1-sa)^3]} \quad [(bb-aa+76)^2, (bb-aa+76)^2, (bb-aa+532)^2] = [14bb(1-sa)^3, 14bb(1-sa)^3, 5232bb(1-sa)^3]$$

Dalam kasus kami, kami mendapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024 = 1024 \quad 1024 = 1024$$

Mari kita gunakan crosslaw ini untuk mencari spread di A. Untuk melakukan ini, kita buat crosslaw untuk kuadran a, b, dan c, dan selesaikan untuk spread yang tidak diketahui sa.

Anda dapat melakukannya dengan tangan dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasilnya, kami sudah memilikinya.

```
>$crosslaw(a,b,c,x), $solve(% ,x)
```

$$[x=925] \quad [x=925]$$

$$x = \frac{9}{25}$$

Kita sudah tahu ini. Definisi spread adalah kasus khusus dari crosslaw.

Kita juga dapat menyelesaikan ini untuk umum a,b,c. Hasilnya adalah rumus yang menghitung penyebaran sudut segitiga yang diberikan kuadrat dari ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$[[168bbx+36bb^2+(-72aa-84)bb+36aa^2-84aa+4936, 168bbx+36bb^2+(-72aa-84)bb+36aa^2-84aa+4936, 152_{52}bbx+18bb^2+(-36aa-152_{32})bb+18aa^2-152_{32}aa+2518] = 0]]$$

$$[[168bbx+36bb^2+(-72aa-84)bb+36aa^2-84aa+4936, 168bbx+36bb^2+(-72aa-84)bb+36aa^2-84aa+4936, 15252bbx+18bb^2+(-36aa-15232)bb+18aa^2-15232aa+2518] = 0]$$

Kita bisa membuat fungsi dari hasilnya. Fungsi seperti itu sudah didefinisikan dalam file geometri.e dari Euler.

```
>$spread(a,b,c)
```

925925

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi

a,a,4a7a,a,4a7

Hasilnya rasional, yang tidak begitu mudah didapat jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

6767

Ini adalah sudut dalam derajat.

```
>deprint(arcsin(sqrt(6/7)))
```

67° 47' 32.44'

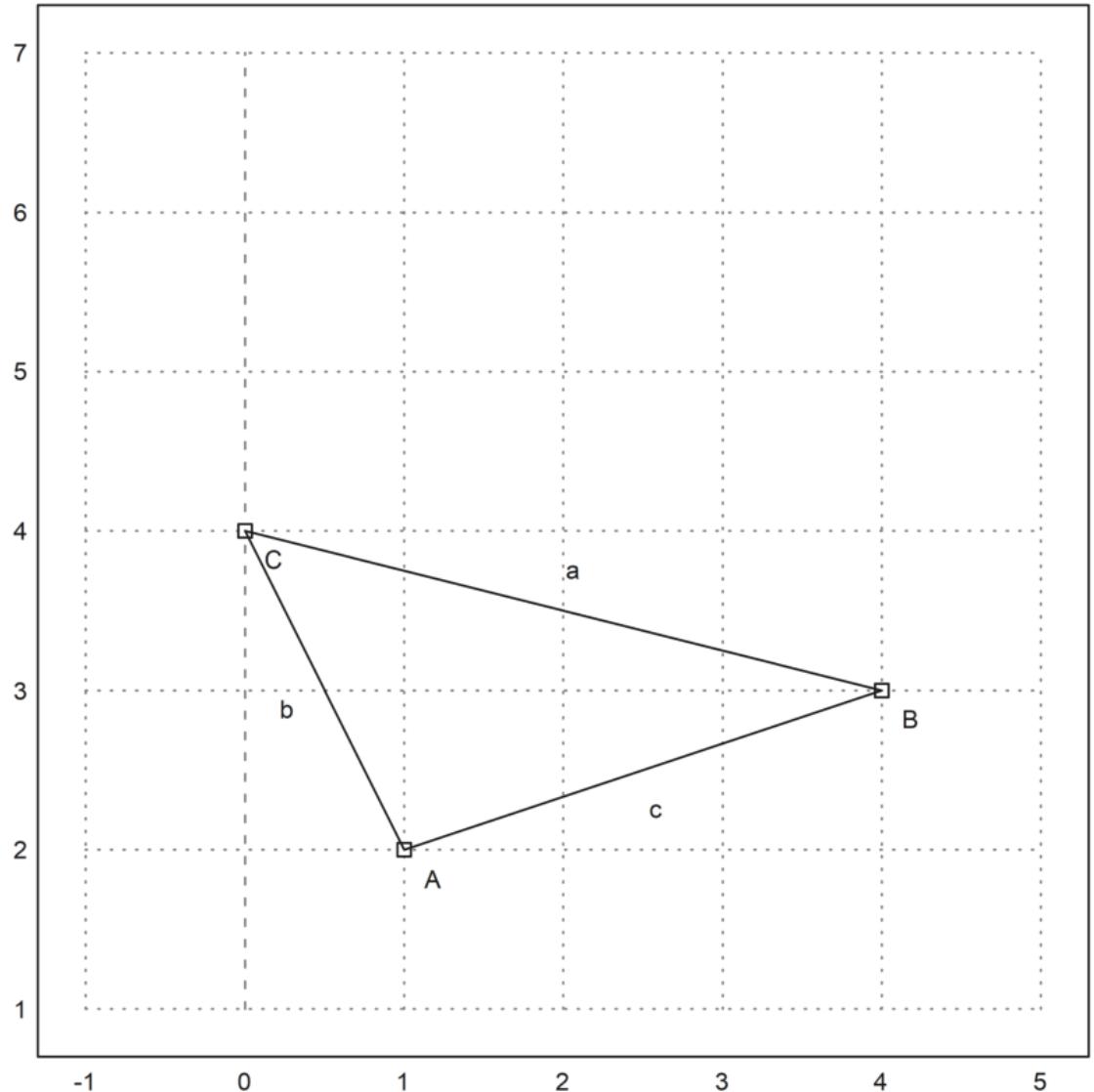
## Contoh lain

---

Sekarang, mari kita coba contoh yang lebih maju.

Kami mengatur tiga sudut segitiga sebagai berikut.

```
>A&:=[1,2]; B&:=[4,3]; C&:=[0,4]; ...
setPlotRange(-1,5,1,7); ...
plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
insimg;
```



Menggunakan 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)
```

$$10\sqrt{10}$$

Euler juga mengandung fungsi untuk kuadran antara dua titik.

Dalam contoh berikut, karena  $c+b$  bukan  $a$ , maka segitiga itu bukan persegi panjang.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

1717

5

17

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle

menggunakan metode biasa berdasarkan hasil kali titik dua vektor.  
Hasilnya adalah beberapa pendekatan floating point.

$$A = \langle 1, 2 \rangle, B = \langle 4, 3 \rangle, C = \langle 0, 4 \rangle$$

$$a = C - B = \langle -4, 1 \rangle, c = A - B = \langle -3, -1 \rangle, \beta = \angle ABC$$

$$a \cdot c = |a| \cdot |c| \cos \beta$$

$$\cos \angle ABC = \cos \beta = a \cdot c / |a| \cdot |c| = 12 - 117 - \sqrt{10} - \sqrt{1117} - \sqrt{10} - \sqrt{11710} = 1$$

```
> wb &= computeAngle(A, B, C); $wb, $(wb/pi*180)()  
arccos(1110 - sqrt(17) - sqrt(1117)) * 180 / pi
```

32.4711922908

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kami memasukkan kuadran a, b, dan c ke dalam

hukum silang dan menyelesaikan x.

```
> $crosslaw(a, b, c, x), $solve(% , x),  
[x=4950][x=4950]
```

$$x = \frac{49}{50}$$

Yaitu, apa yang dilakukan oleh penyebaran fungsi yang didefinisikan dalam "geometry.e".

```
> sb &= spread(b, a, c); $sb  
4917049170
```

Maxima mendapatkan hasil yang sama menggunakan trigonometri biasa, jika kita memaksanya. Itu menyelesaikan istilah  $\sin(\arccos(...))$  menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

```
> $sin(computeAngle(A, B, C))^2  
4917049170
```

Setelah kita memiliki spread di B, kita dapat menghitung tinggi ha di sisi a. Ingat bahwa

$$sb = ha \cdot csb = hac$$

Menurut definisi.

```
> ha &= c * sb; $ha  
49174917
```

Gambar berikut telah dihasilkan dengan program geometri C.a.R., yang dapat menggambar kuadrat dan menyebar.

image: (20) Rational\_Geometry\_CaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadratnya.

```
> $sqrt(ha)  
717 - sqrt(717)
```

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita berhadapan dengan kuadrat!

```
> $sqrt(ha) * sqrt(a) / 2  
7272
```

Rumus determinan biasa menghasilkan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

7272

## Rumus Heron

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue(a,b,c,sb,ha);
```

Pertama kita hitung spread di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat ("quadrea"?), faktorkan dengan Maxima, dan kita mendapatkan rumus Heron yang terkenal.

Memang, ini sulit dilakukan dengan pensil dan kertas.

```
>$spread(b^2,c^2,a^2), $factor(%*c^2*a^2/4)
```

$$(-c+b+a)(c-b+a)(c+b+a)16(-c+b+a)(c-b+a)(c+b-a)(c+b+a)16$$

$$\frac{(-c+b+a)(c-b+a)(c+b-a)(c+b+a)}{16}$$

## Aturan Triple Spread

Kerugian dari spread adalah mereka tidak lagi hanya menambahkan sudut yang sama.

Namun, tiga spread dari sebuah segitiga memenuhi aturan "triple spread" berikut.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

$$(sc+sb+sa)_2=2(sc_2+sb_2+sa_2)+4sasc(sc+sb+sa)_2=2(sc_2+sb_2+sa_2)+4sasc$$

Aturan ini berlaku untuk setiap tiga sudut yang menambah 180 °.

$$\alpha+\beta+\gamma=\pi \alpha+\beta+\gamma=\pi$$

Sejak menyebar

$$\alpha, \pi-\alpha, \alpha, \pi-\alpha$$

sama, aturan triple spread juga benar, jika

$$\alpha+\beta=\gamma \alpha+\beta=\gamma$$

Karena penyebaran sudut negatif adalah sama, aturan penyebaran rangkap

tiga juga berlaku, jika

$$\alpha+\beta+\gamma=0 \alpha+\beta+\gamma=0$$

Misalnya, kita dapat menghitung penyebaran sudut 60°. Ini 3/4. Persamaan memiliki solusi kedua, bagaimanapun, di mana semua spread adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

$$[x=34, x=0][x=34, x=0]$$

Sebaran 90° jelas 1. Jika dua sudut dijumlahkan menjadi 90°, sebarannya menyelesaikan persamaan sebaran rangkap tiga dengan a,b,1. Dengan perhitungan berikut kita mendapatkan a+b=1.

```
>$triplespread(x,y,1), $solve(% ,x)
```

$$[x=1-y][x=1-y]$$

$$[x = 1 - y]$$

Karena sebaran  $180^\circ - t$  sama dengan sebaran  $t$ , rumus sebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau selisih dua sudut lainnya.

Jadi kita dapat menemukan penyebaran sudut berlipat ganda. Perhatikan bahwa ada dua solusi lagi. Kami membuat ini fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &=  
factor(rhs(%[1]))
```

$$[x=4a-4a^2, x=0][x=4a-4a^2, x=0]$$

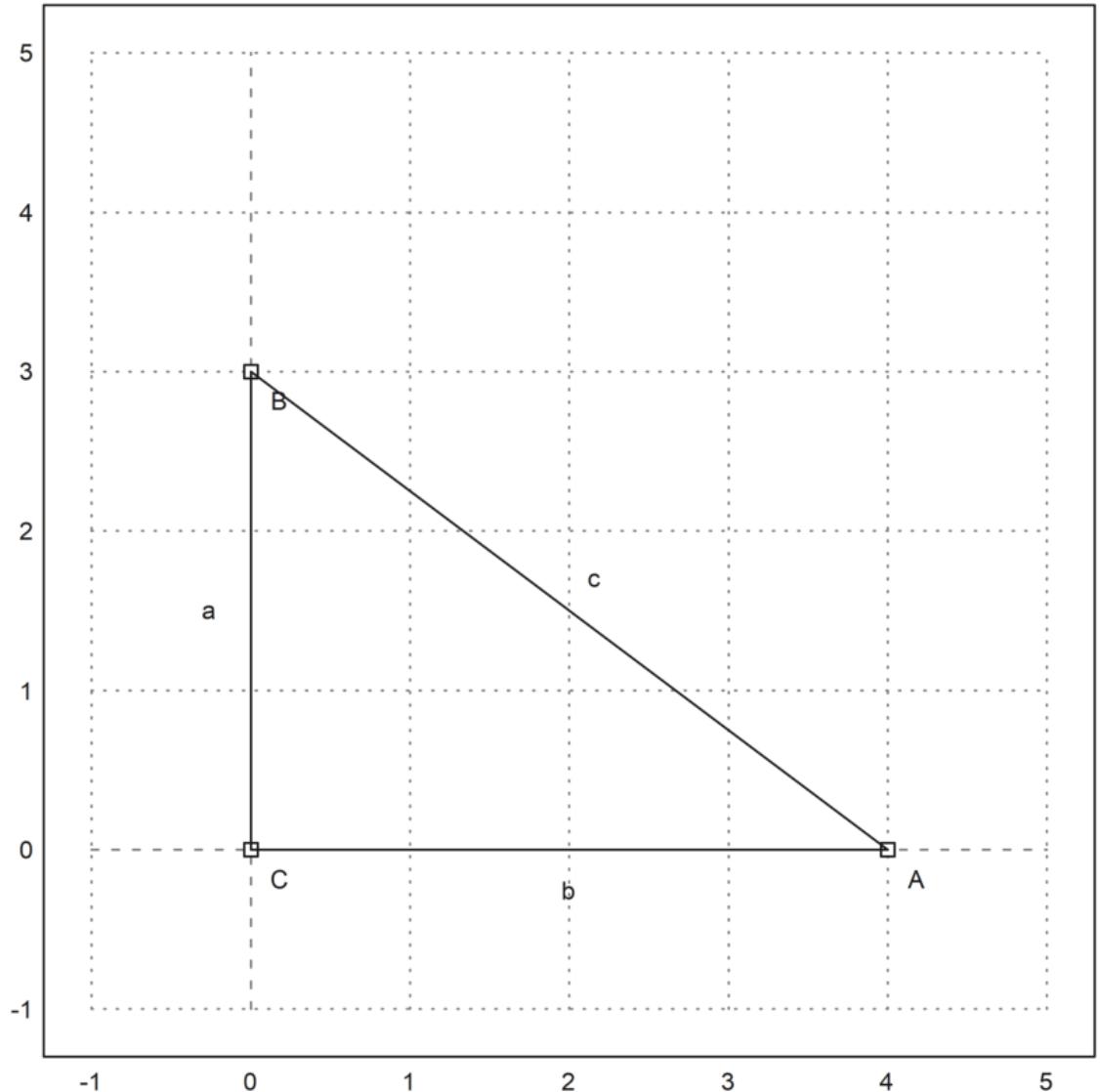
$$- 4 (a - 1) a$$

## Sudut Pembagi

---

Ini situasinya, kita sudah tahu.

```
>C:=[0,0]; A:=[4,0]; B:=[0,3]; ...  
setPlotRange(-1,5,-1,5); ...  
plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...  
plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...  
insimg;
```



Mari kita hitung panjang garis bagi sudut di A. Tetapi kita ingin menyelesaiakannya untuk umum  $a, b, c$ .

```
>&remvalue(a,b,c);
```

Jadi pertama-tama kita hitung penyebaran sudut yang dibagi dua di A, dengan menggunakan rumus sebaran rangkap tiga.

Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut terbelah  $180^\circ$ -wa.

```
>$triplespread(x,x,a/(a+b)), $solve(% ,x), sa2 &= rhs(%[1]); $sa2  
-b\sqrt{b+a}----\sqrt{b+a}2b+2a-bb+a+b+a2b+2a
```

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

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

Mari kita periksa persegi panjang Mesir.

```
>sa2 with [a=3^2,b=4^2]
```

```
110110
```

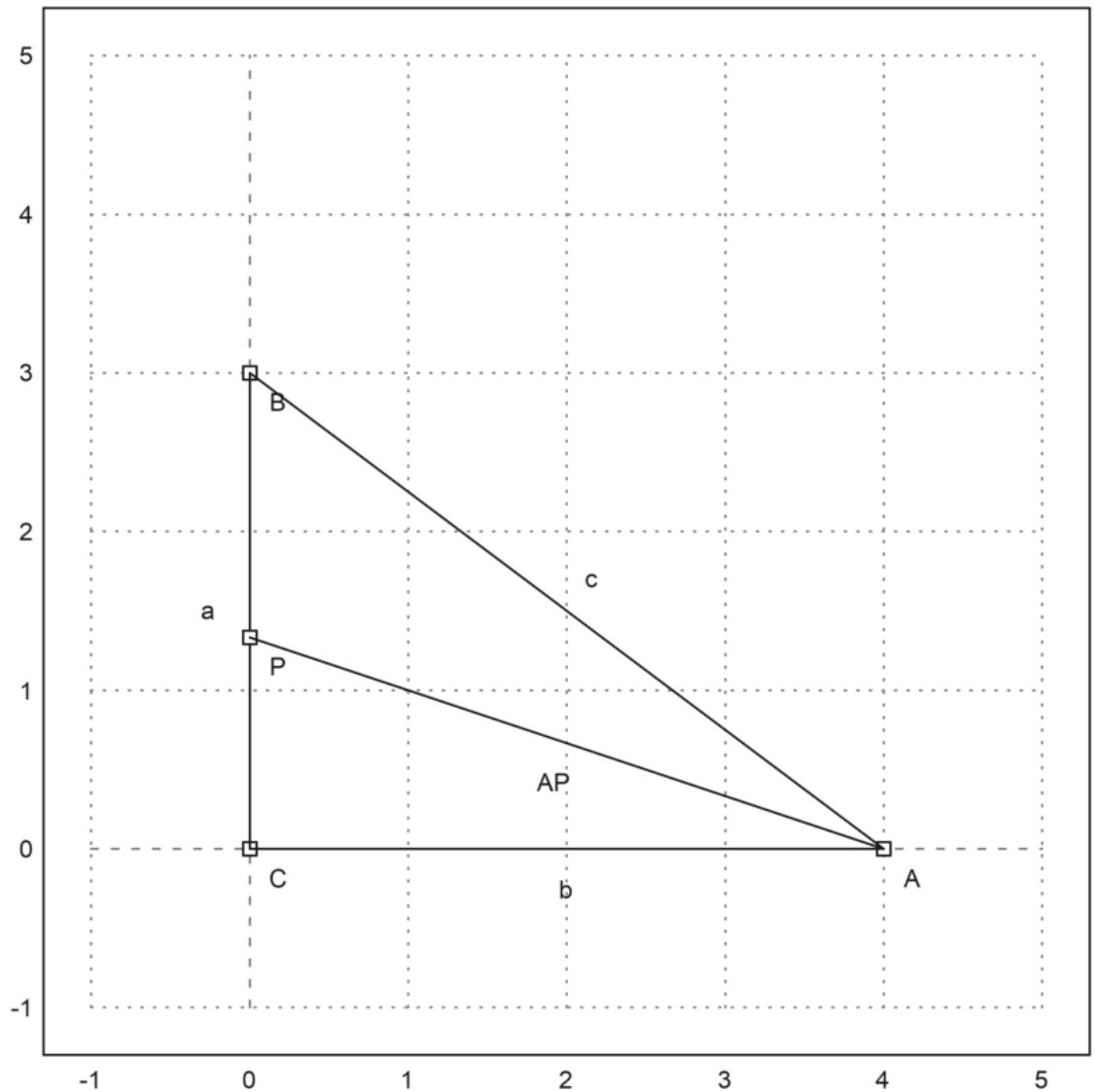
Kami dapat mencetak sudut dalam Euler, setelah mentransfer penyebaran ke radian.

```
>wa2 := arccsin(sqrt(1/10)); degprint(wa2)
18°26'5.82''
```

Titik P adalah perpotongan garis bagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
[0, 1.33333]
```

```
>plotPoint(P,"P"); plotSegment(A,P):
```



Mari kita periksa sudut dalam contoh spesifik kita.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

```
0.321750554397
```

```
0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.

Kami menggunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

$$BC\sin(w_a) = AC\sin(w_b) = AB\sin(w_c) \quad BC\sin(\frac{\pi}{2})(w_a) = AC\sin(\frac{\pi}{2})(w_b) = AB\sin(\frac{\pi}{2})(w_c)$$

berlaku dalam segitiga apa pun. Kuadratkan, itu diterjemahkan ke dalam

apa yang disebut "hukum penyebaran"

$$a^2 = b^2 + c^2 - 2bc \cos A$$

di mana  $a, b, c$  menunjukkan qudrances.

Karena spread CPA adalah  $1 - \sin^2 A$ , kita dapatkan darinya  $b^2 = a^2 + c^2 - 2ac \cos A$

dan dapat menghitung  $\cos A$  (kuadran dari garis-bagi sudut).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$2b(b+a)b\sqrt{b^2 + a^2 - 2ab}$$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa, [a=3^2, b=4^2]))); distance(A,P)
```

```
4.21637021356
```

```
4.21637021356
```

Kita juga dapat menghitung  $P$  menggunakan rumus spread.

```
>py&=factor(ratsimp(sa2*bisa)); $py
```

$$-b(b\sqrt{b^2 + a^2 - 2ab})b\sqrt{b^2 + a^2 - 2ab - b(b^2 + a^2 - 2ab)}$$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

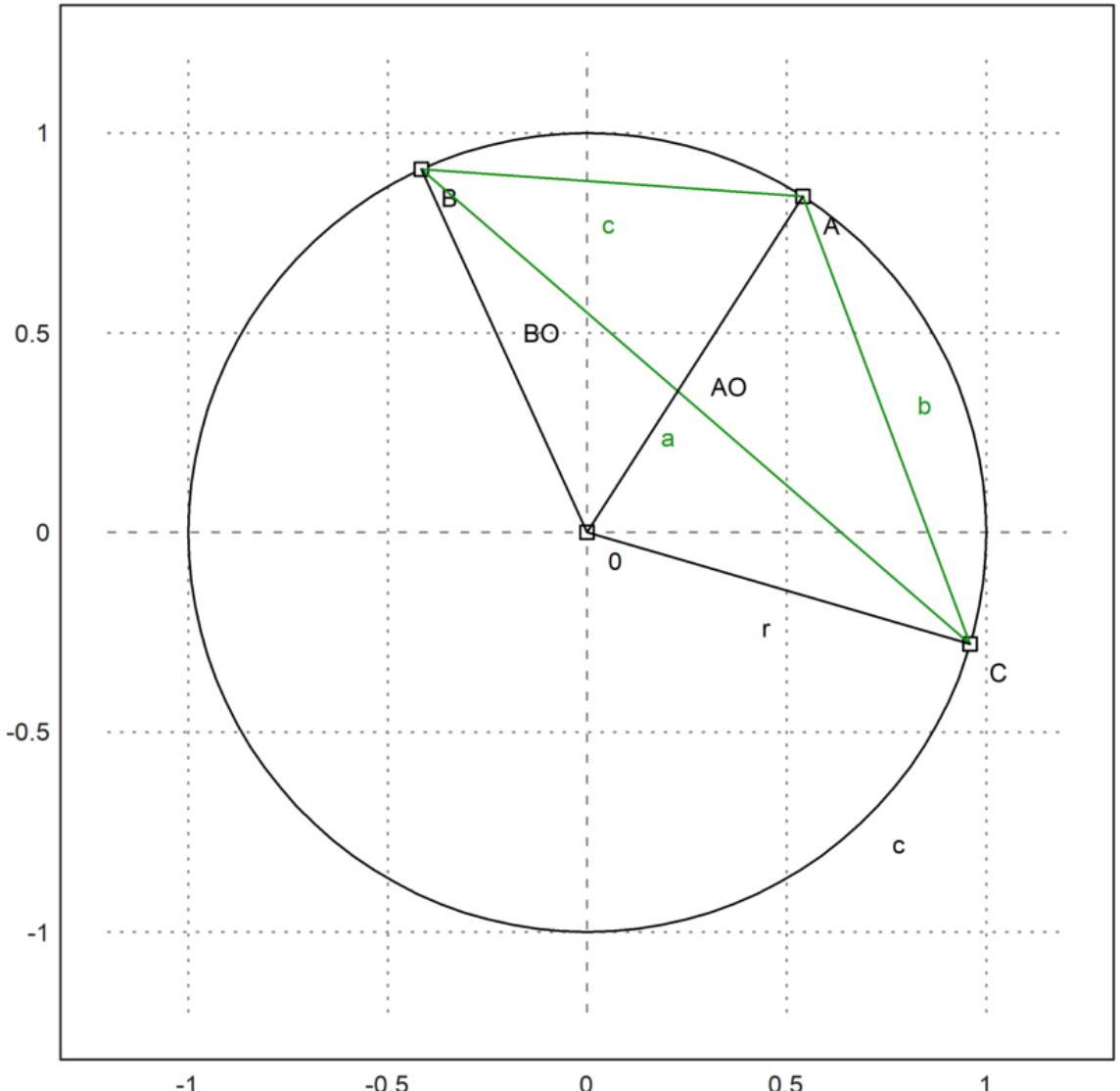
```
>sqrt(mxmeval("at(py, [a=3^2, b=4^2]))
```

```
1.333333333333
```

## Sudut Akord

Perhatikan situasi berikut.

```
>setPlotRange(1.2); ...
color(1); plotCircle(circleWithCenter([0,0],1)); ...
A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...
plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b");
plotSegment(C,B,"a"); ...
color(1); O:=[0,0]; plotPoint(O,"O"); ...
plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
insimg;
```



Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran rangkap tiga untuk sudut-sudut di pusat O untuk  $r$ . Jadi kita mendapatkan rumus untuk jari-jari kuadrat dari pericircle dalam hal kuadrat dari sisi.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &=
rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]);
$rabc
-abcc2-2bc+a(-2c-2b)+b2+a2-abcc2-2bc+a(-2c-2b)+b2+a2
```

Kita dapat menjadikannya sebagai fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
Mari kita periksa hasilnya untuk poin A,B,C.
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
Jari-jarinya memang 1.
>periradius(a,b,c)
```

Faktanya, spread CBA hanya bergantung pada b dan c. Ini adalah teorema sudut chord.

```
>$spread(b,a,c)*rabc | ratsimp
```

b4b4

Sebenarnya spreadnya adalah  $b/(4r)$ , dan kita melihat bahwa sudut chord dari chord b adalah setengah dari sudut pusat.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

00

## Contoh 6: Jarak Minimal pada Bidang

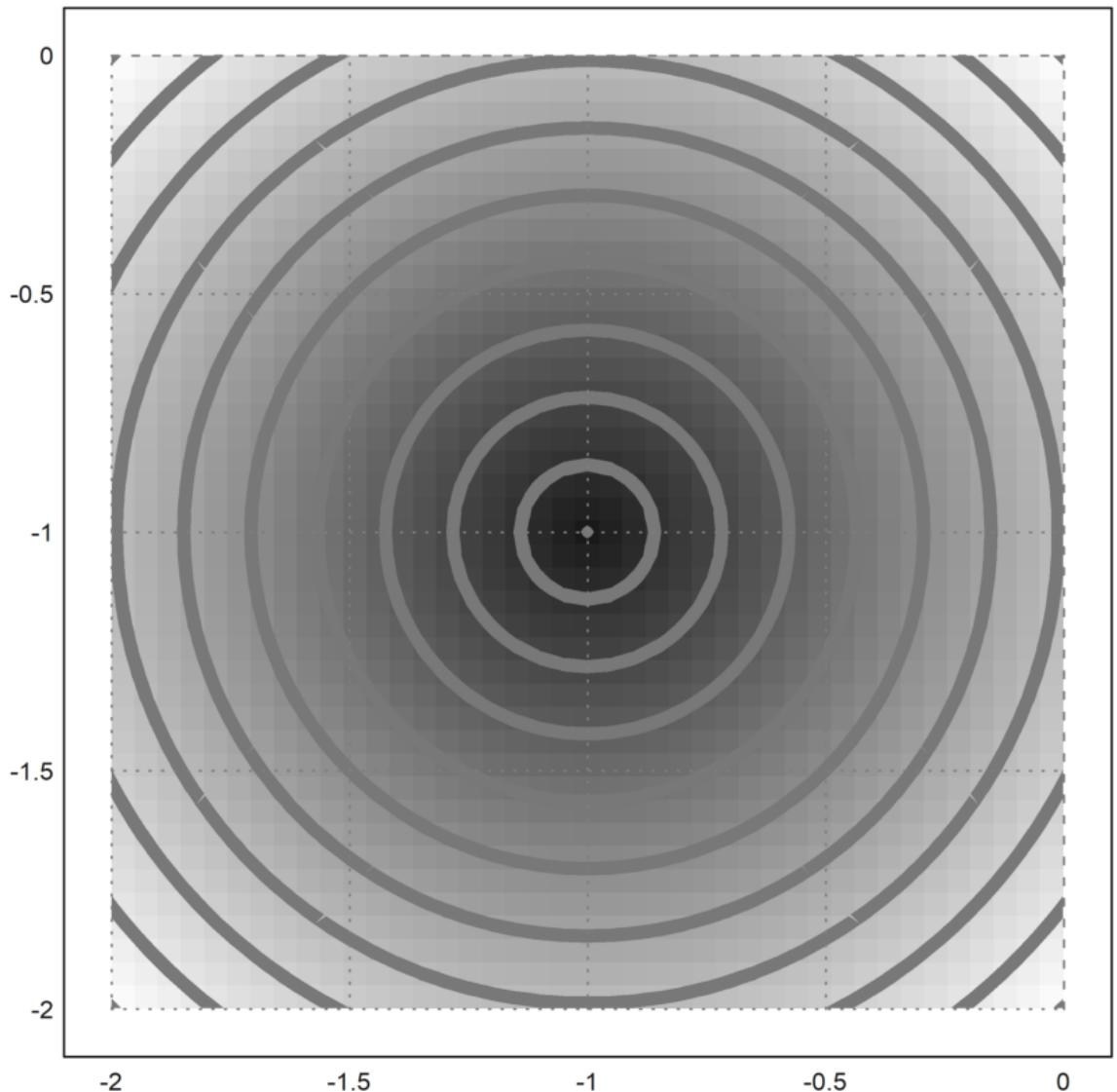
### Catatan awal

---

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.

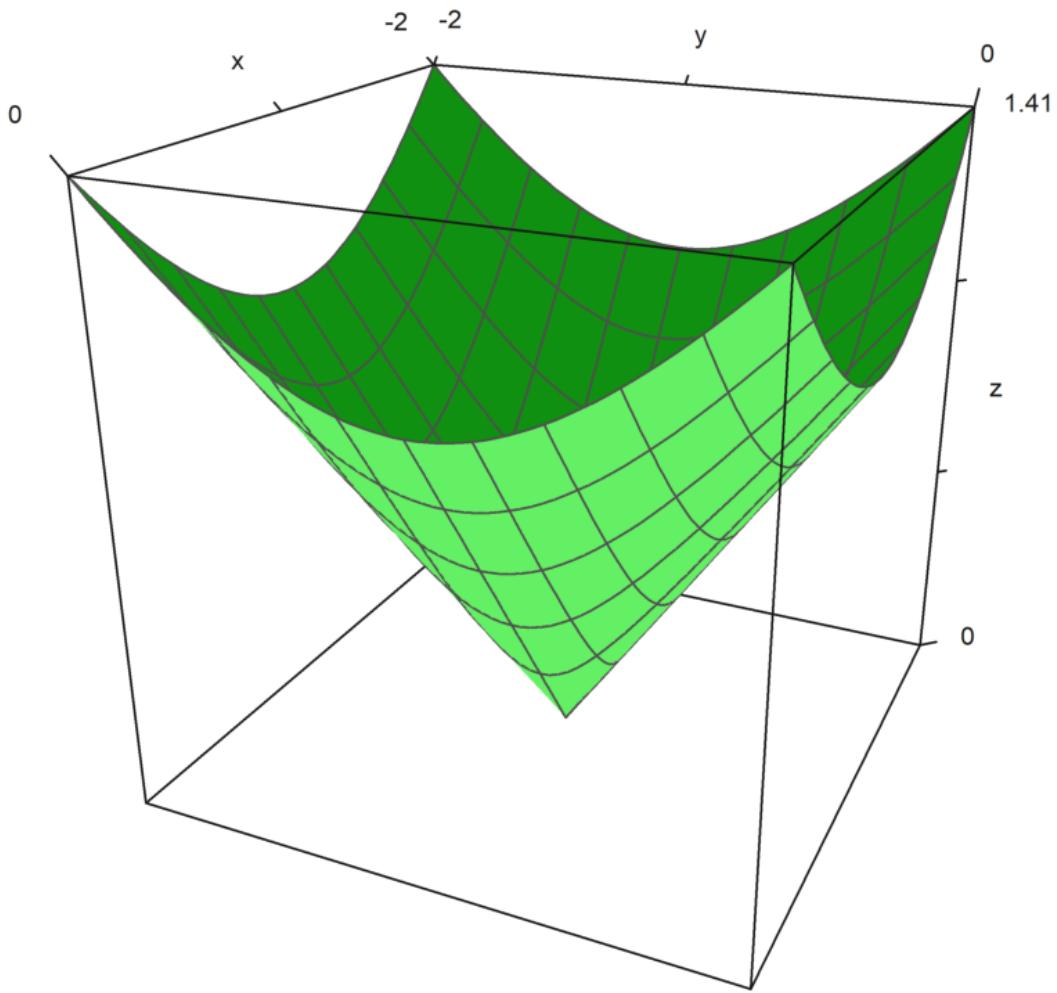
```
>&remvalue();  
>A=[-1,-1];  
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)  
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...  
title="If you see ellipses, please set your window square"):
```

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):
```



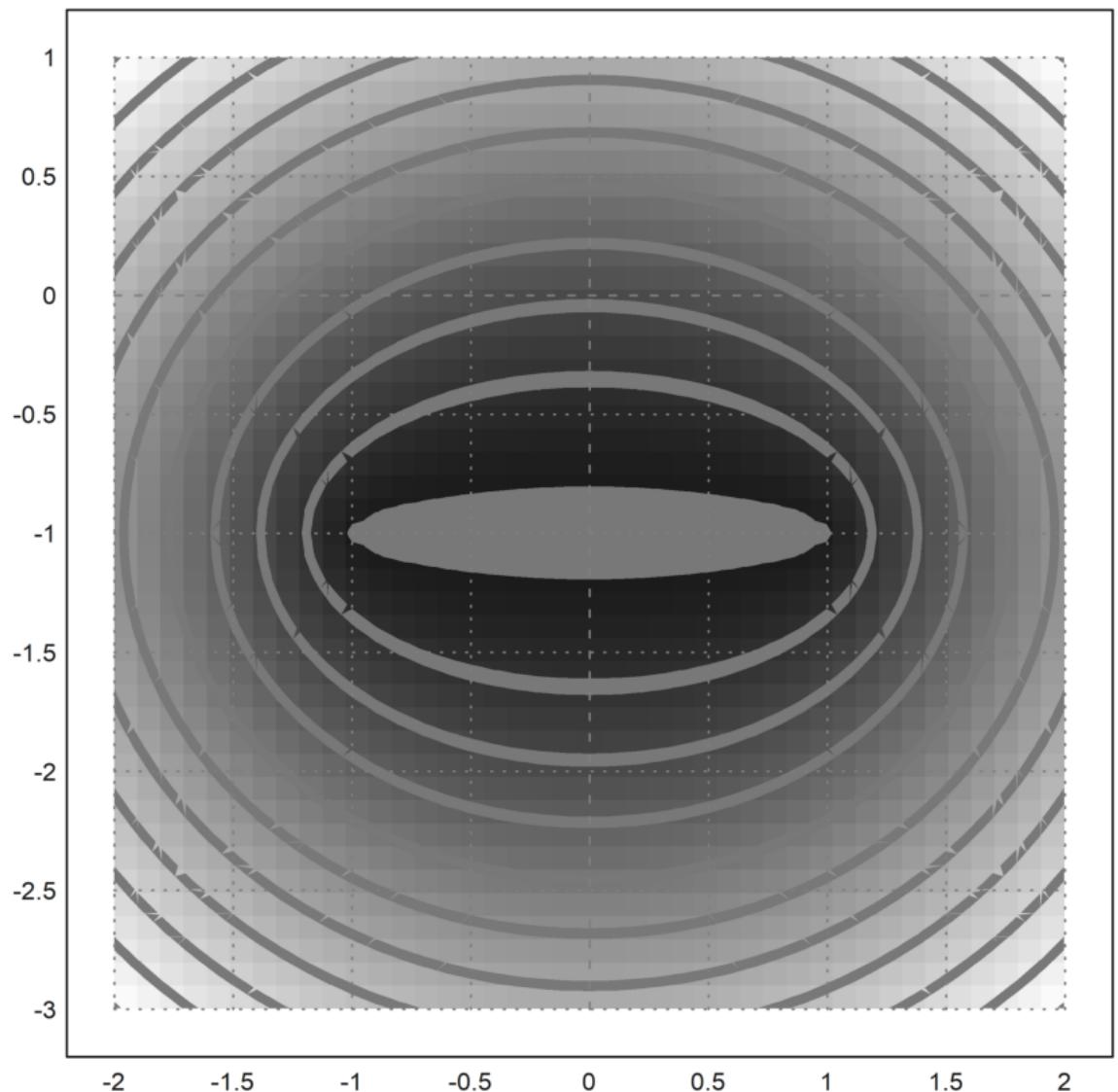
Tentu saja minimal 0 dicapai di A.

## Poin Dua

---

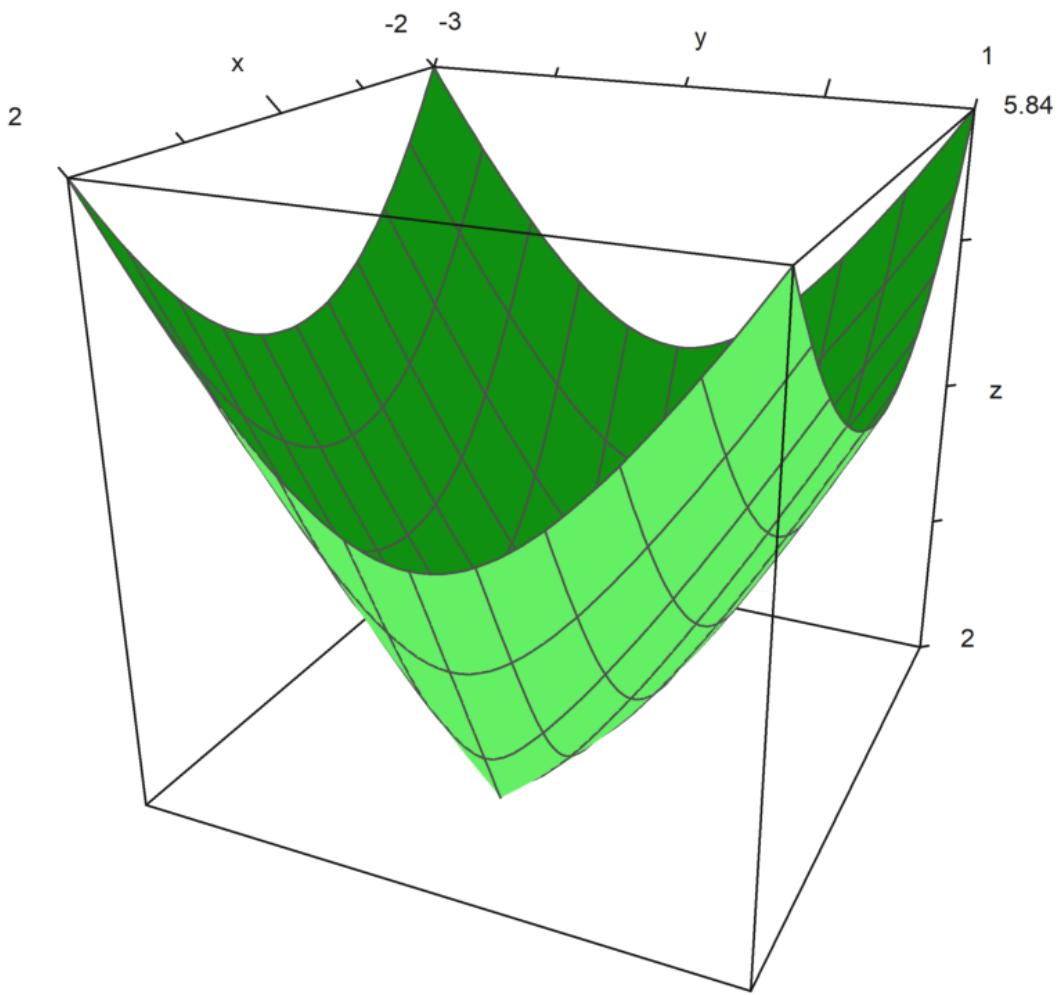
Sekarang kita lihat fungsi MA+MB dimana A dan B adalah dua titik (tetap). Ini adalah "fakta yang diketahui" bahwa kurva level adalah ellips, titik fokusnya adalah A dan B; kecuali untuk AB minimum yang konstan pada segmen [AB]:

```
>B=[1,-1];
>function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2+(y-B[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



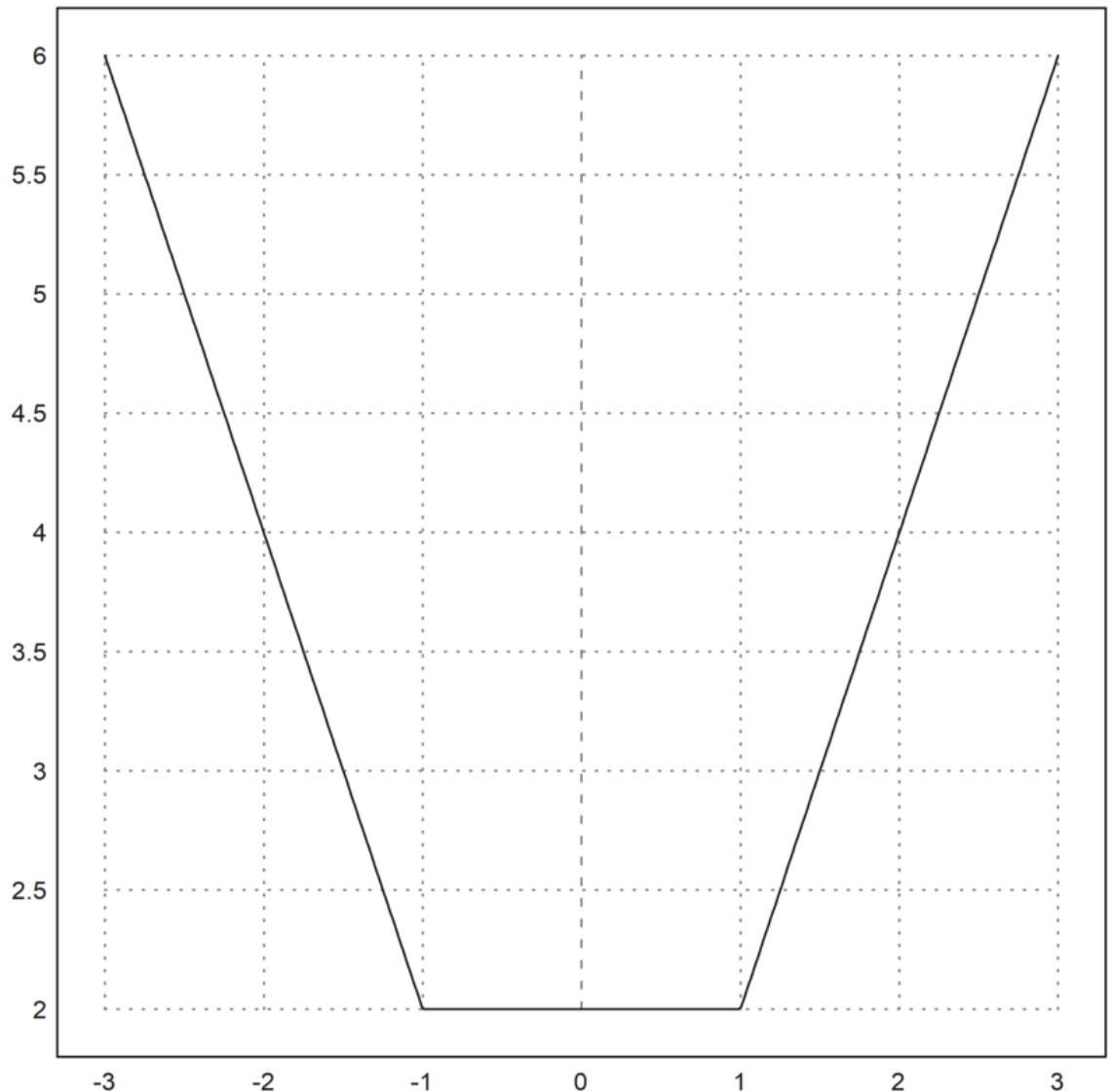
Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Pembatasan garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)", xmin=-3, xmax=3):
```



## Poin Ketiga

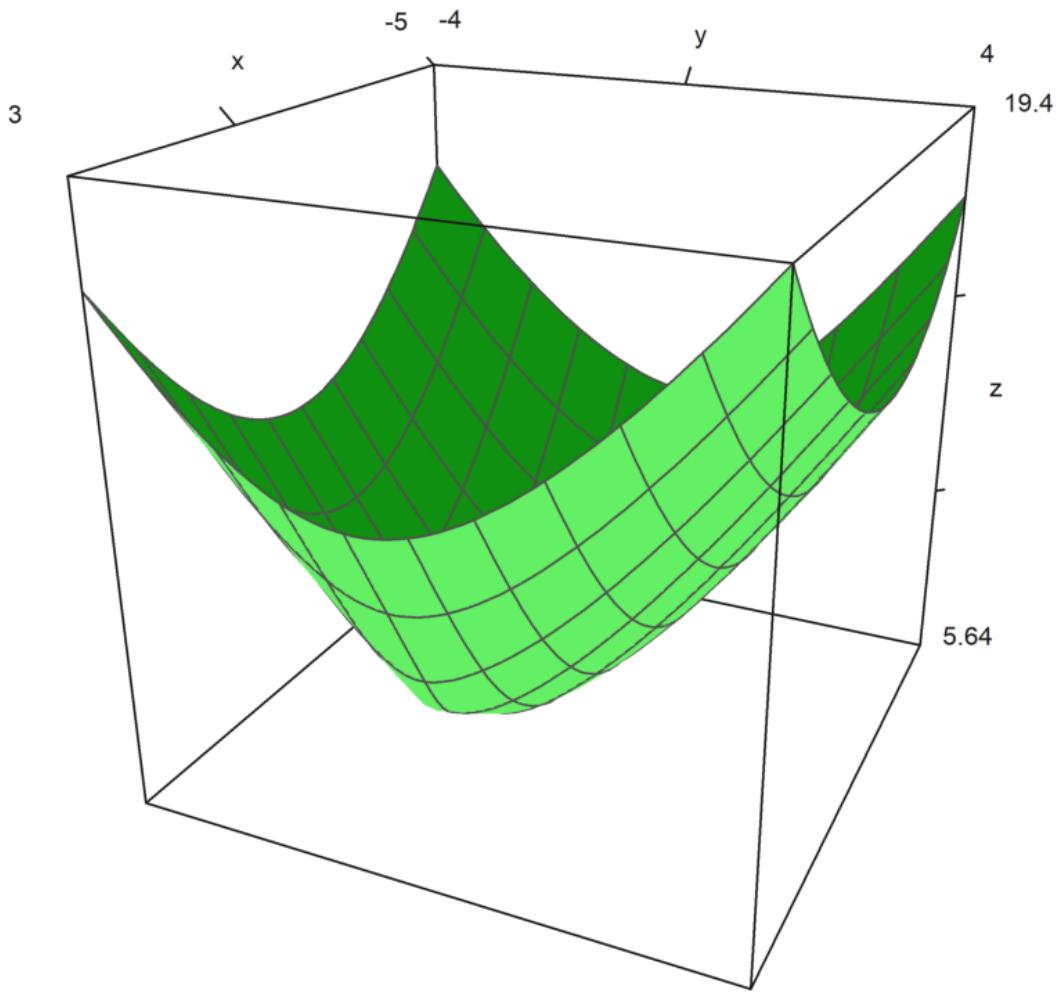
---

Sekarang hal-hal yang kurang sederhana: Ini sedikit kurang terkenal bahwa  $MA+MB+MC$  mencapai minimum pada satu titik pesawat tetapi untuk menentukan itu kurang sederhana:

- 1) Jika salah satu sudut segitiga ABC lebih dari  $120^\circ$  (katakanlah di A), maka minimum dicapai pada titik ini (misalnya  $AB+AC$ ).

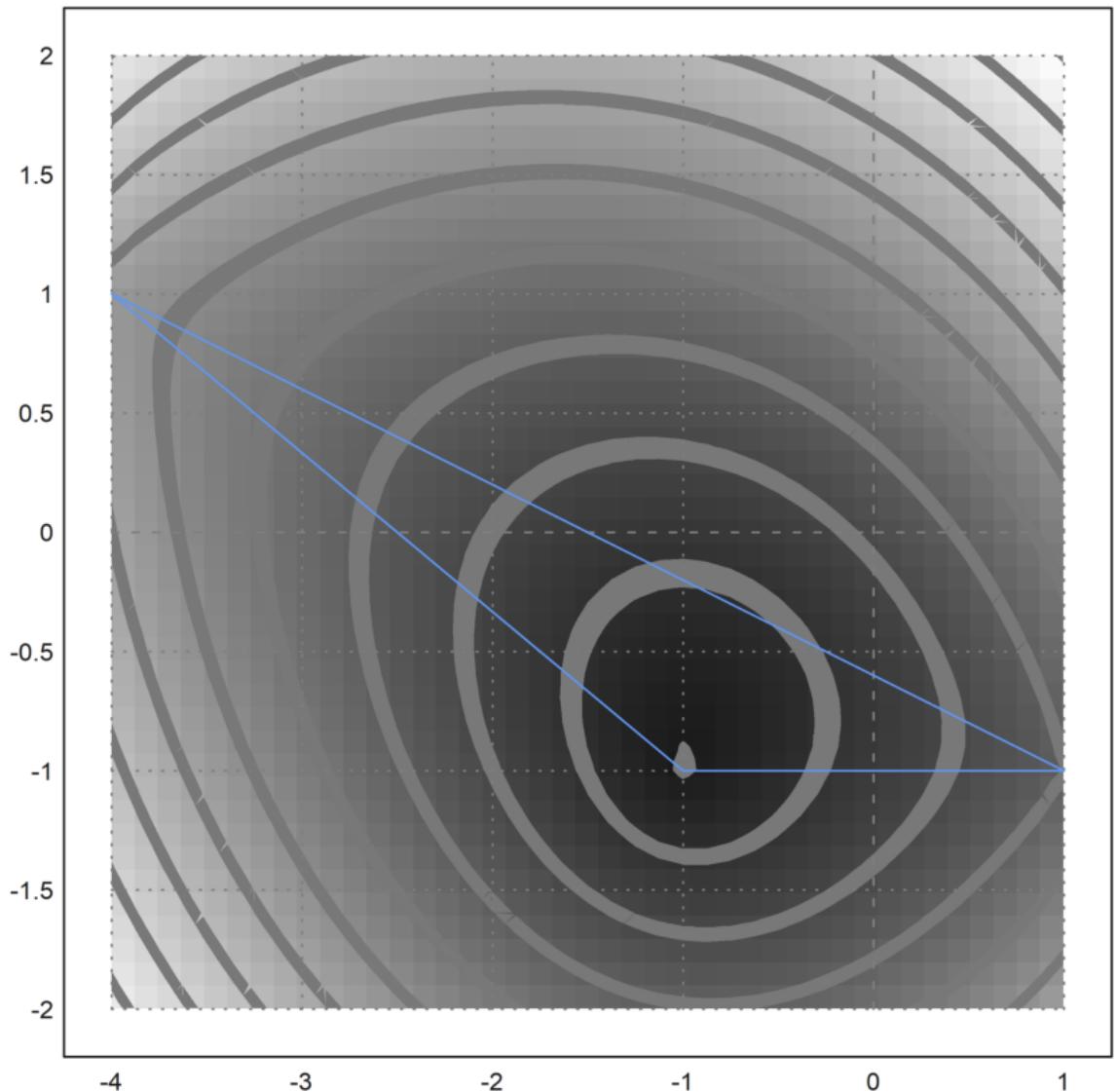
Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
>insimg;
```



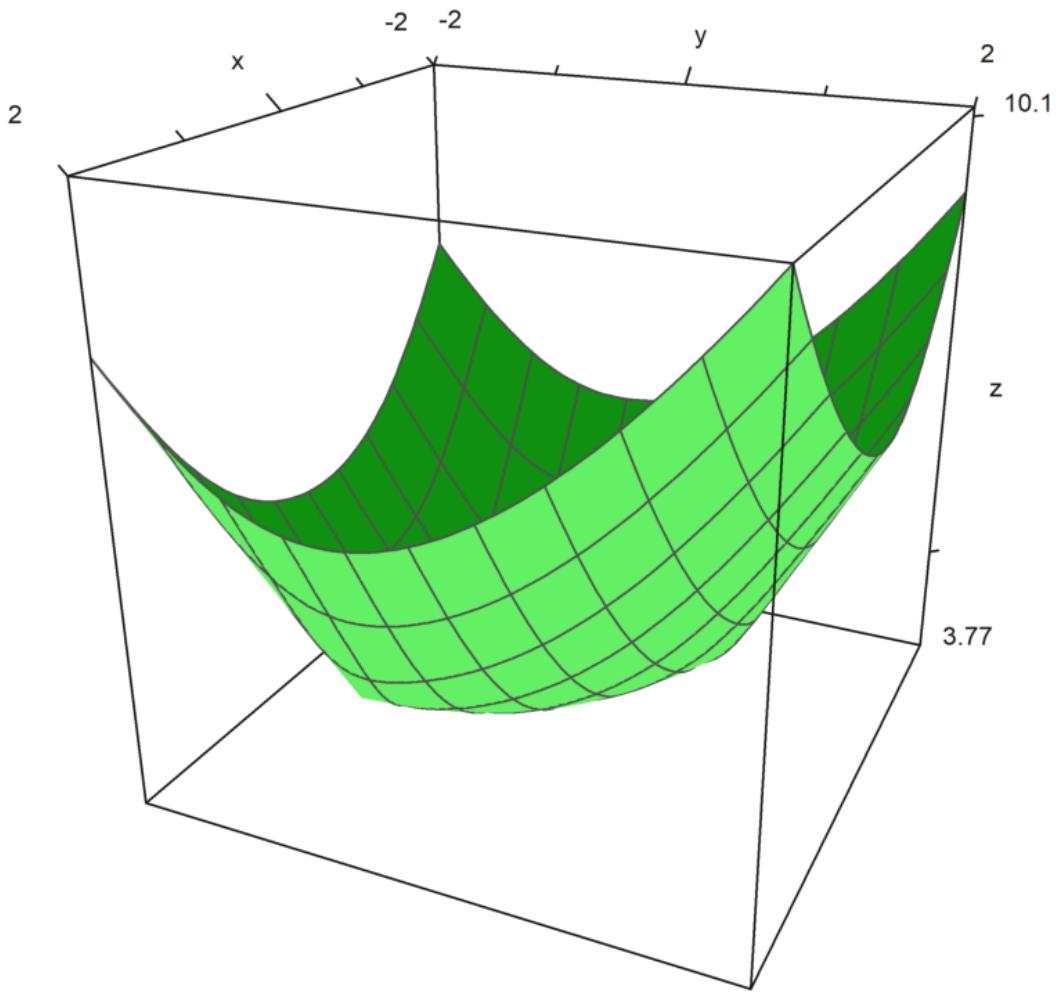
```
>fcontour ("d3", xmin=-4, xmax=1, ymin=-2, ymax=2, hue=1, title="The minimum is on  
A");  
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);  
>insimg;
```

The minimum is on A



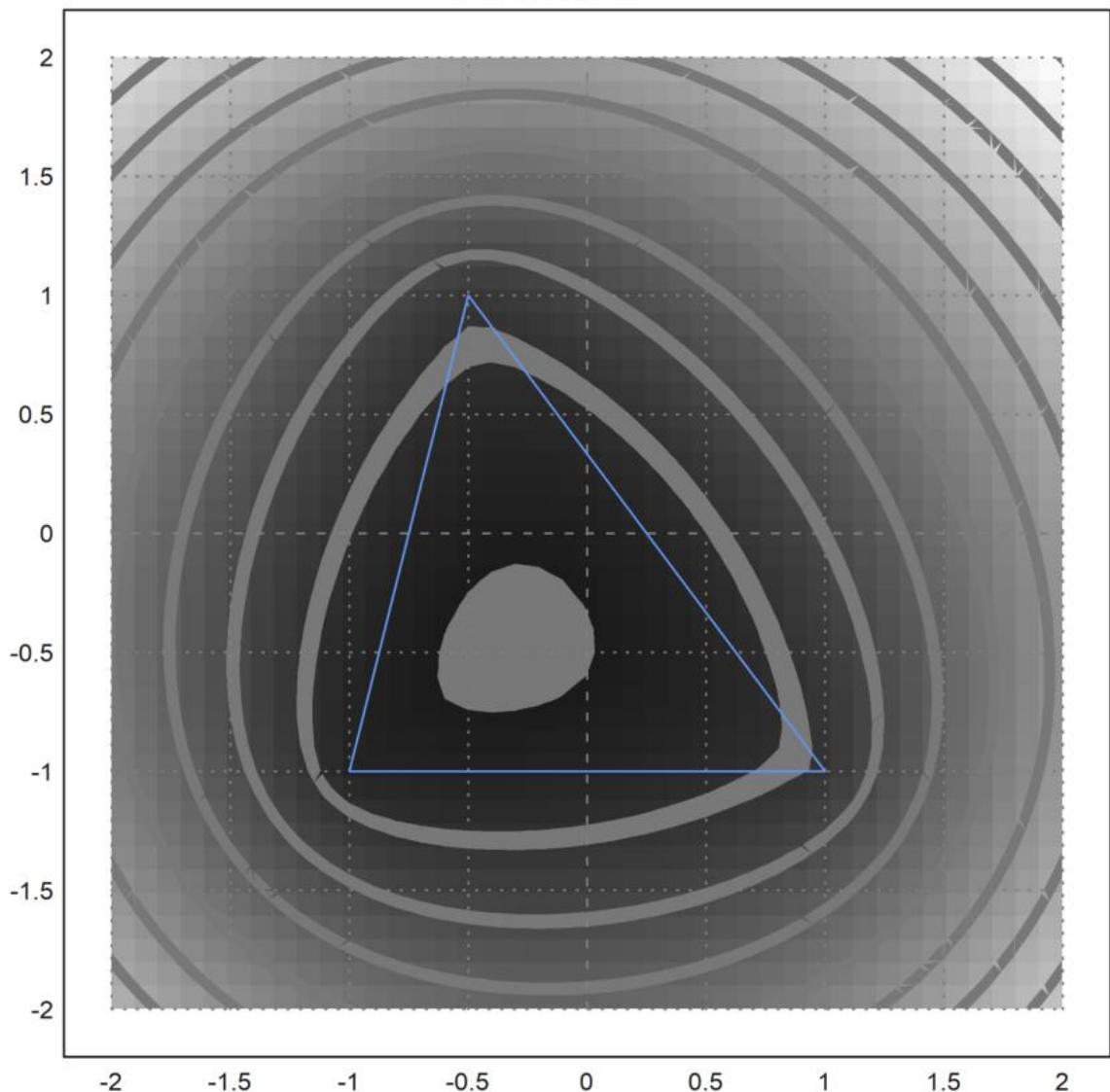
- 2) Tetapi jika semua sudut segitiga ABC kurang dari  $120^\circ$ , minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (maka masing-masing  $120^\circ$ ):

```
>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;
```

The Fermat point



Merupakan kegiatan yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya tahu soft yang ditulis di Jawa yang memiliki instruksi "garis kontur" ...

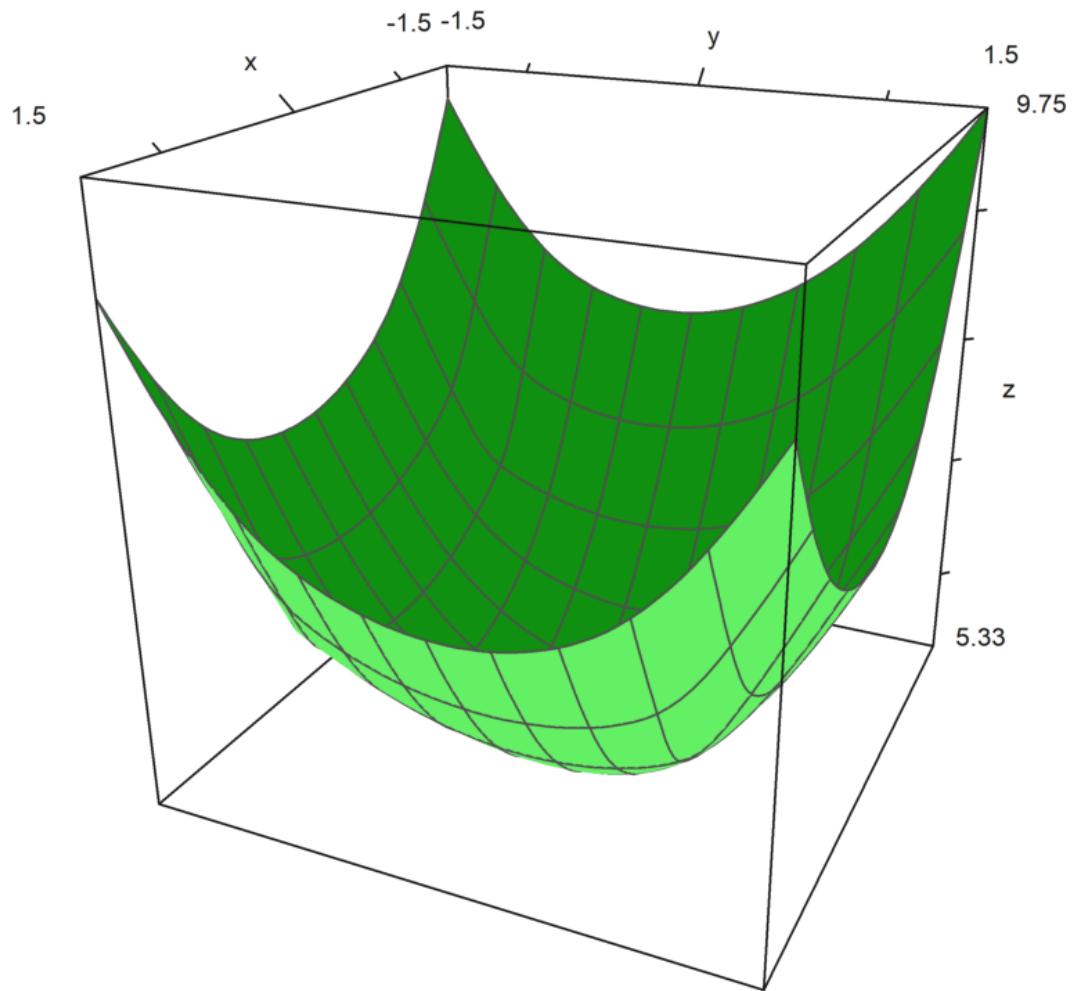
Semua ini di atas telah ditemukan oleh seorang hakim Perancis bernama Pierre de Fermat; dia menulis surat kepada dilettants lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di pajak penghasilan. Jadi titik unik F sedemikian rupa sehingga  $FA+FB+FC$  minimal, disebut titik Fermat segitiga. Tetapi tampaknya beberapa tahun sebelumnya, Torriccelli Italia telah menemukan titik ini sebelum Fermat melakukannya! Bagaimanapun tradisinya adalah mencatat poin ini F...

## Poin Empat

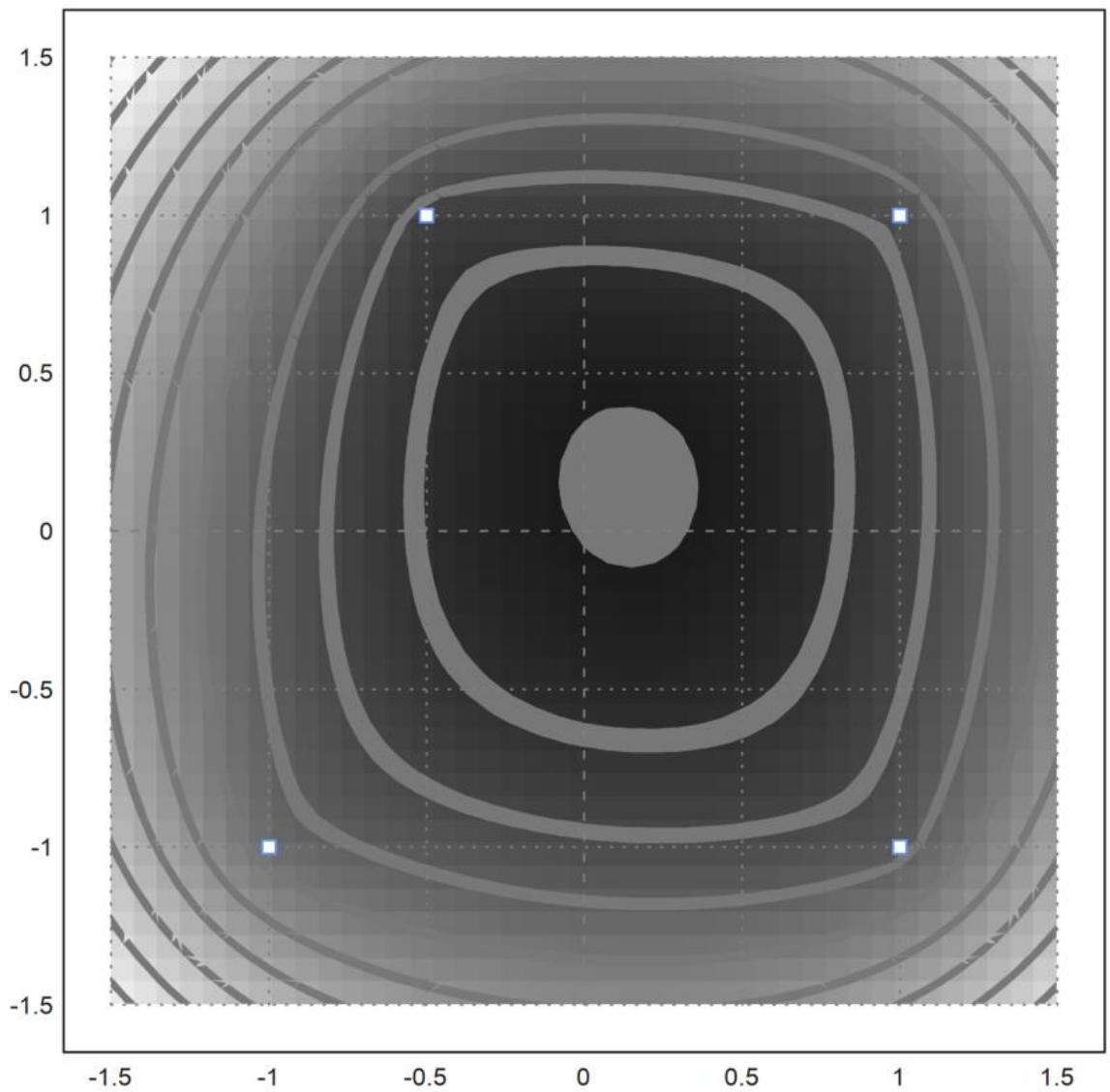
Langkah selanjutnya adalah menambahkan 4 titik D dan mencoba meminimalkan  $MA+MB+MC+MD$ ; katakan bahwa Anda adalah operator TV kabel

dan ingin mencari di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
```



```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
>insimg;
```



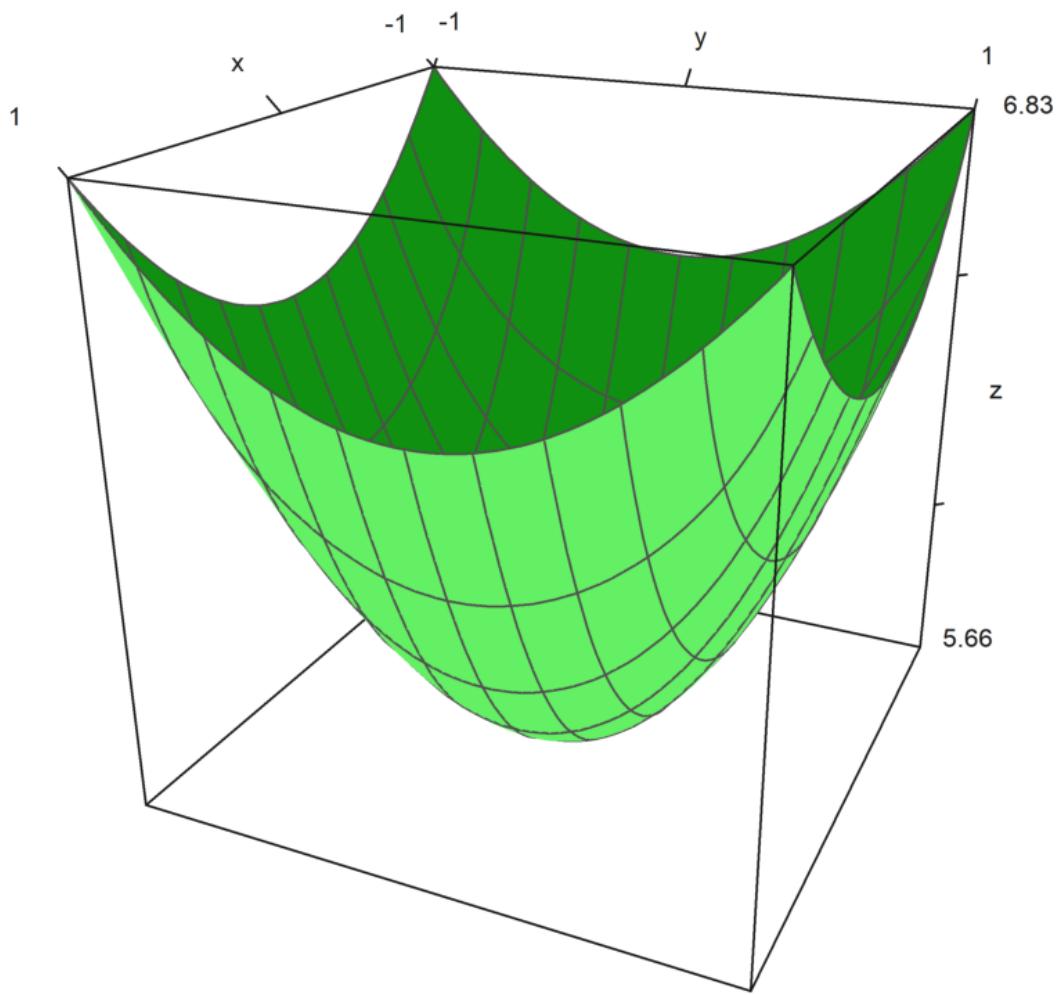
Masih ada minimum dan tidak tercapai di salah satu simpul A, B, C atau

D:

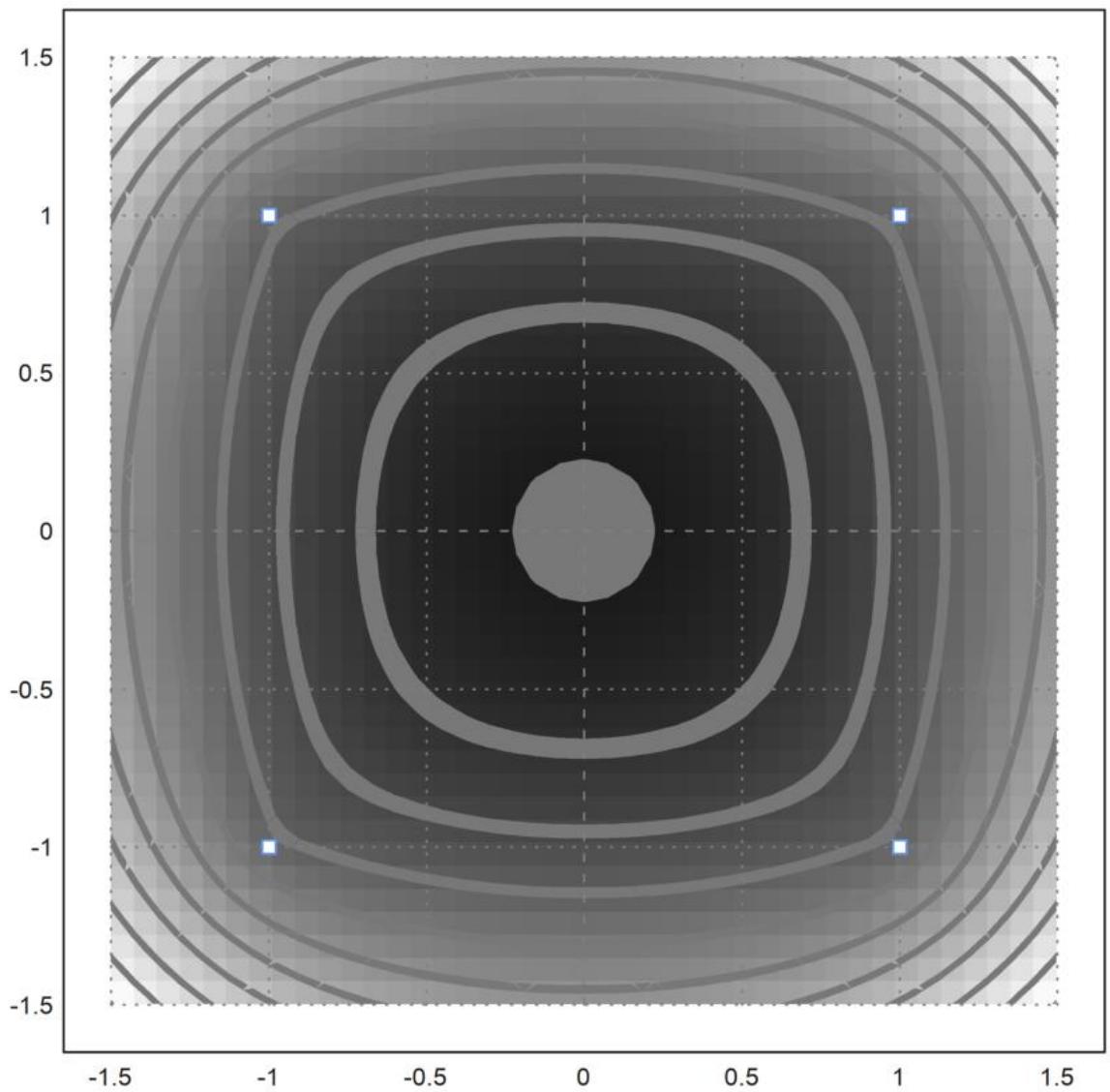
```
>function f(x):=d4(x[1],x[2])
>neldermin("f", [0.2,0.2])
[0.142858, 0.142857]
Tampaknya dalam kasus ini, koordinat titik optimal adalah rasional atau mendekati rasional...
```

Sekarang ABCD adalah persegi, kami berharap bahwa titik optimal akan menjadi pusat ABCD:

```
>C=[-1,1];
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1);
```



```
>fcontour ("d4", xmin=-1.5, xmax=1.5, ymin=-1.5, ymax=1.5, hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);
>insimg;
```



## Contoh 7: Bola Dandelin dengan Povray

---

Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan `pvengine.exe` di jalur program.

Pertama kita hitung jari-jari bola.

Jika Anda melihat gambar di bawah, Anda melihat bahwa kita membutuhkan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file `geometri.e` dari Euler untuk ini.

```
>load geometry;
```

Pertama dua garis yang membentuk kerucut.

```
>g1 &= lineThrough([0,0],[1,a])
      [- a, 1, 0]
```

```
>g2 &= lineThrough([0,0],[-1,a])
```

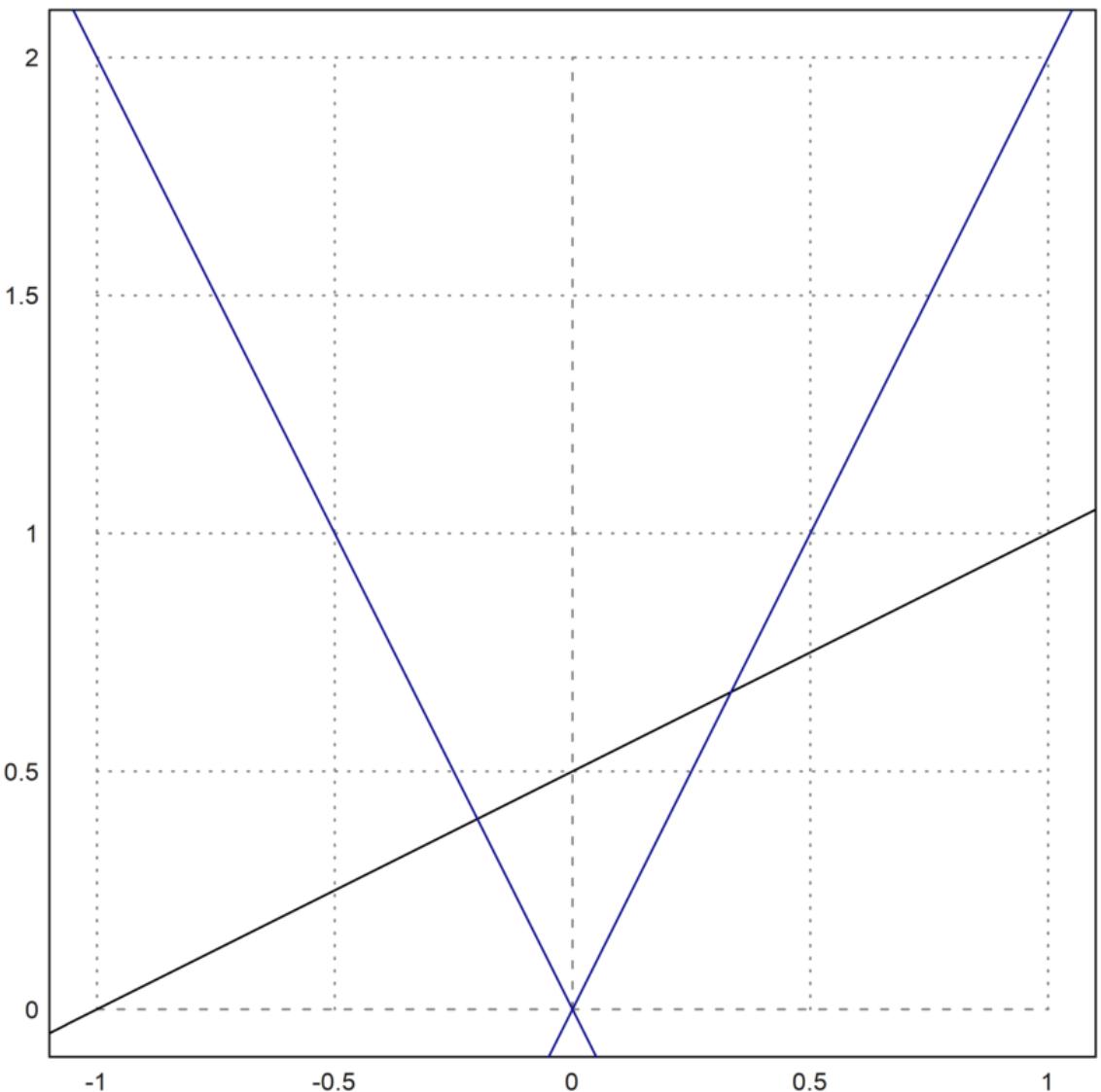
```
[- a, - 1, 0]
```

Kemudian baris ketiga.

```
>g &= lineThrough([-1,0],[1,1])  
[- 1, 2, 1]
```

Kami merencanakan semuanya sejauh ini.

```
>setPlotRange(-1,1,0,2);  
>color(black); plotLine(g(),"")  
>a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```



Sekarang kita ambil titik umum pada sumbu y.

```
>P &= [0,u]  
[0, u]
```

Hitung jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1  
(a2ua2+1-u)2+a2u2(a2+1)2-----\sqrt{(a2ua2+1-u)2+a2u2(a2+1)2}
```

Hitung jarak ke g.

```
>d &= distance(P,projectToLine(P,g)); $d  
 (u+25-u)2+(2u-1)225-----sqrt((u+25-u)2+(2u-1)225
```

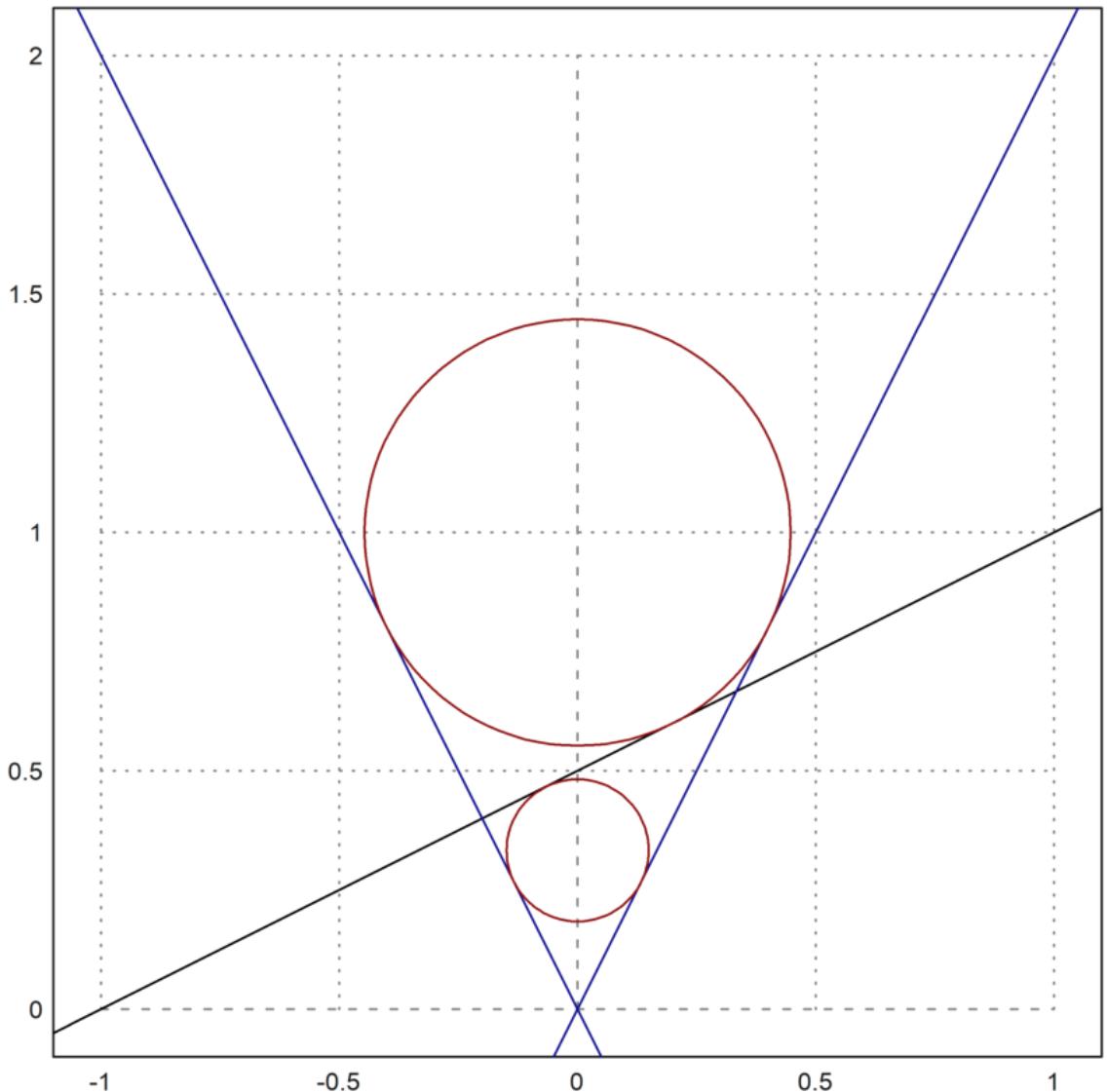
Dan temukan pusat kedua lingkaran yang jaraknya sama.

```
>sol &= solve(d1^2=d^2,u); $sol  
 [u=-5-sqrt(a2+1)-----sqrt(+2a2+24a2-1,u=5-  
 sqrt(a2+1)-----sqrt(+2a2+24a2-1)[u=-5a2+1+2a2+24a2-1,u=5a2+1+2a2+24a2-1]
```

Ada dua solusi.

Kami mengevaluasi solusi simbolis, dan menemukan kedua pusat, dan kedua jarak.

```
>u := sol()  
 [0.333333, 1]  
>dd := d()  
 [0.149071, 0.447214]  
 Plot lingkaran ke dalam gambar.  
>color(red);  
>plotCircle(circleWithCenter([0,u[1]],dd[1]),"");  
>plotCircle(circleWithCenter([0,u[2]],dd[2]),"");  
>insimg;
```



## Plot dengan Povray

---

Selanjutnya kami merencanakan semuanya dengan Povray. Perhatikan bahwa

Anda mengubah perintah apa pun dalam urutan perintah Povray berikut, dan menjalankan kembali semua perintah dengan Shift-Return.

Pertama kita memuat fungsi povray.

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
Kami mengatur adegan dengan tepat.
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
Selanjutnya kita menulis dua bidang ke file Povray.
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
Dan kerucutnya, transparan.
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
Kami menghasilkan bidang terbatas pada kerucut.
```

```

>gp=g();
>pc=povcone([0,0,0],0,[0,0,a],1,"");
>vp=[gp[1],0,gp[2]]; dp=gp[3];
>writeln(povplane(vp,dp,povlook(blue,0.5),pc));
    Sekarang kita menghasilkan dua titik pada lingkaran, di mana bola
    menyentuh kerucut.

>function turnz(v) := return [-v[2],v[1],v[3]]
>P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
>writeln(povpoint(P1,povlook(yellow)));
>P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
>writeln(povpoint(P2,povlook(yellow)));
    Kemudian kami menghasilkan dua titik di mana bola menyentuh bidang.
    Ini adalah fokus dari elips.

>P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
>writeln(povpoint(P3,povlook(yellow)));
>P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
>writeln(povpoint(P4,povlook(yellow)));
    Selanjutnya kita hitung perpotongan P1P2 dengan bidang.

>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
>writeln(povpoint(P5,povlook(yellow)));
    Kami menghubungkan titik-titik dengan segmen garis.

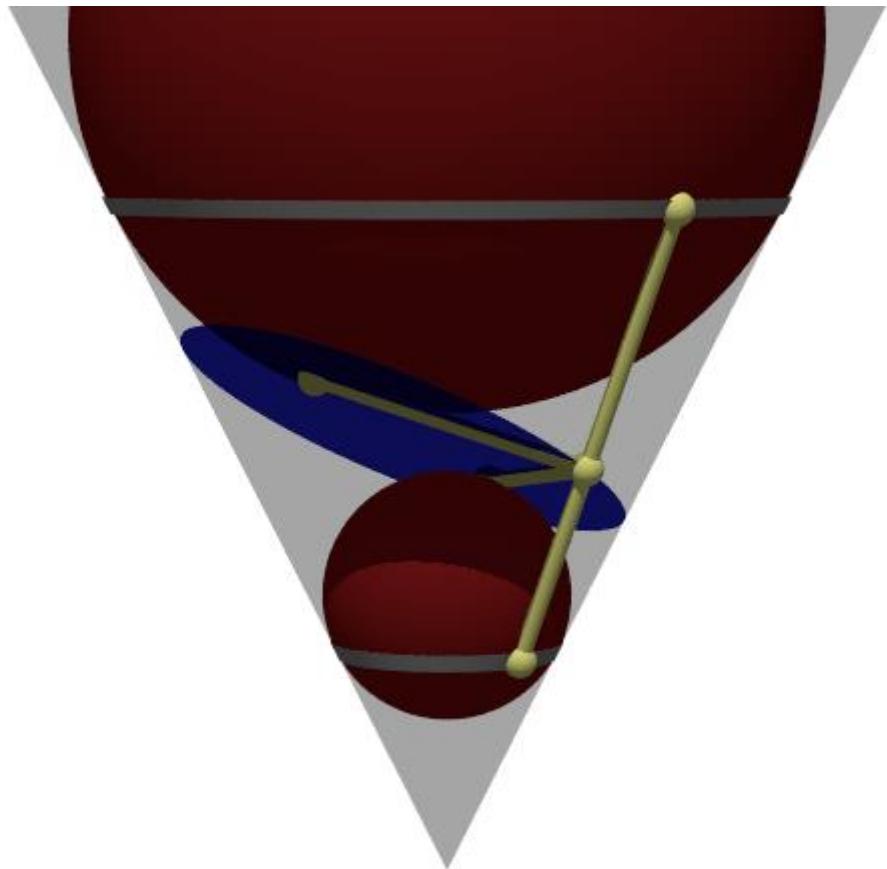
>writeln(povsegment(P1,P2,povlook(yellow)));
>writeln(povsegment(P5,P3,povlook(yellow)));
>writeln(povsegment(P5,P4,povlook(yellow)));
    Sekarang kita menghasilkan pita abu-abu, di mana bola menyentuh
    kerucut.

>pcw=povcone([0,0,0],0,[0,0,a],1.01);
>pc1=povcylinder([0,0,P1[3]-
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 kemudian.

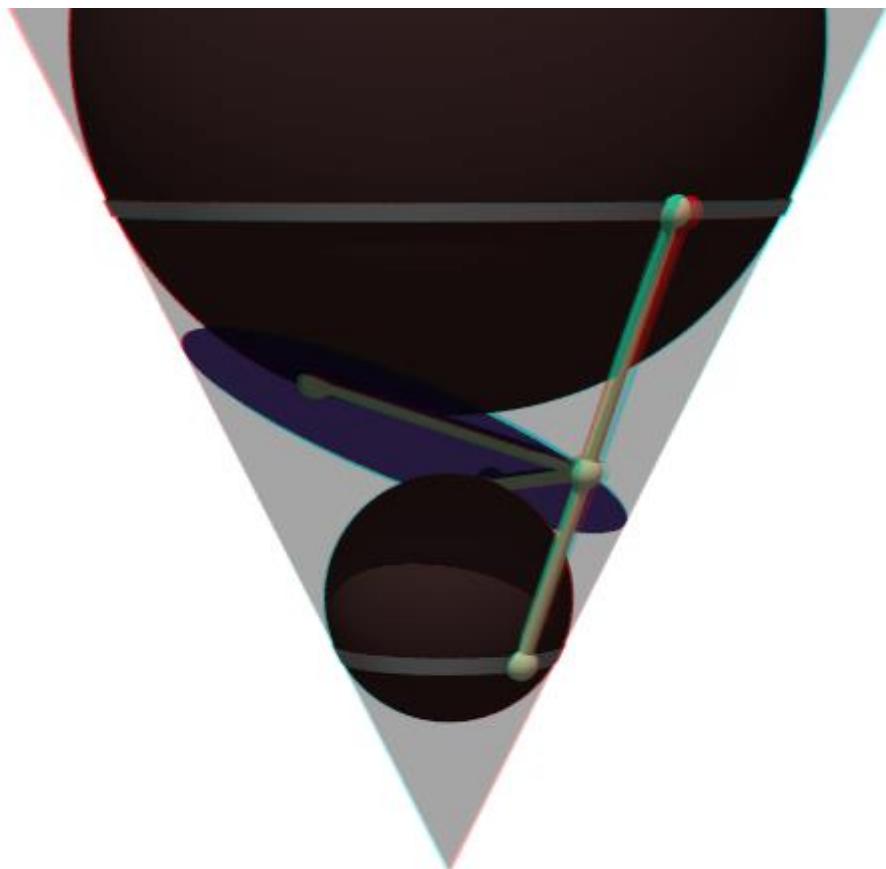
```
>function scene () ...
global a,u,dd,g,g1,defaultpointsize;
writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
gp=g();
pc=povcone([0,0,0],0,[0,0,a],1,"");
vp=[gp[1],0,gp[2]]; dp=gp[3];
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow)));
writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow)));
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-
defaultpointsize/2],[0,0,P1[3]+defaultpointsize/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)));
endfunction

```

Anda membutuhkan kacamata merah/sian untuk menghargai efek berikut.  
>povanaglyph("scene", zoom=11, center=[0,0,0.5], height=10°, angle=140°);



## Contoh 8: Geometri Bumi

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan sferis. Fungsi-fungsi tersebut terdapat dalam file "spherical.e" di folder contoh. Kita perlu memuat file itu terlebih dahulu.

```
>load "spherical.e";
```

Untuk memasukkan posisi geografis, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut koordinat Kampus FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569, 1.92657]
```

Anda dapat mencetak posisi ini dengan sposprint (cetak posisi spherical).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
S 7°46.467' E 110°23.050'
```

Mari kita tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-
59.05),rad(110,24.533)];
```

```
>sposprint(Solo), sposprint(Semarang),
```

```
S 7°34.333' E 110°49.683'
```

```
S 6°59.050' E 110°24.533'
```

Pertama kita menghitung vektor dari satu ke yang lain pada bola ideal.

Vektor ini [pos,jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang  $7^\circ$ .

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan  
jarak FMIPA-Solo
```

65°20'26.60''

53.8945384608

Ini adalah perkiraan yang baik. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak yang begitu pendek hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->" km", // perkiraan jarak FMIPA-Semarang
```

88.0114026318 km

Ada fungsi untuk heading, dengan mempertimbangkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

65.34°

Sudut segitiga melebihi  $180^\circ$  pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,  
Semarang,Solo); degprint(asum)
```

180°0'10.77''

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, ini tidak akurat karena kesalahan pengurangan dalam asum-pi.

```
>(asum-pi)*rearth(48°)^2->" km^2", // perkiraan luas segitiga FMIPA-Solo-  
Semarang,
```

2116.02948749 km<sup>2</sup>

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>

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi heading dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kami menggunakan vektor. Untuk menambahkan vektor ke posisi, kami menggunakan vektor saddr.

```
>v=svector(FMIPA,Solo); sposprint(saddrvector(FMIPA,v)), sposprint(Solo),
```

S 7°34.333' E 110°49.683'

S 7°34.333' E 110°49.683'

Fungsi-fungsi ini mengasumsikan bola yang ideal. Hal yang sama di bumi.

```
>sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))),  
sposprint(Solo),
```

S 7°34.333' E 110°49.683'

S 7°34.333' E 110°49.683'

Mari kita beralih ke contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];
```

```
>sposprint(Tugu), sposprint(Monas)
```

S 7°46.998' E 110°21.966'

S 6°10.500' E 106°48.717'

Menurut Google Earth, jaraknya adalah 429,66 km. Kami mendapatkan pendekatan yang baik.

```
>esdist(Tugu,Monas)->" km", // perkiraan jarak Tugu Jogja - Monas Jakarta
```

431.565659488 km

Judulnya sama dengan judul yang dihitung di Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

294°17'2.85''

Namun, kita tidak lagi mendapatkan posisi target yang tepat, jika kita

menambahkan heading dan jarak ke posisi semula. Hal ini terjadi, karena kita tidak menghitung fungsi invers secara tepat, tetapi mengambil perkiraan jari-jari bumi di sepanjang jalan.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

S 6°10.500' E 106°48.717'

Namun, kesalahannya tidak besar.

```
>sposprint(Monas),
```

S 6°10.500' E 106°48.717'

Tentu kita tidak bisa berlayar dengan tujuan yang sama dari satu tujuan ke tujuan lainnya, jika kita ingin menempuh jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana pun di bumi.

Kemudian

Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti heading yang konstan!

Perhitungan berikut menunjukkan bahwa kami jauh dari tujuan yang benar, jika kami menggunakan pos yang sama selama perjalanan kami.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh dari jarak, menggunakan pos ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya jauh.

```
>sposprint(p), skmprint(esdist(p,Monas))
```

S 6°11.250' E 106°48.372'

1.529km

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran garis lintang 30°, melainkan jalur terpendek yang dimulai 10° lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

79.69°

Tapi, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kami menyesuaikannya pada 1/10 dari total jarak.

```
>p=P1; dist=esdist(P1,P2); ...
```

```
loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10);
```

end;

79.69°

81.67°

83.71°

85.78°

87.89°

90.00°

92.12°

94.22°

96.29°

98.33°

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti heading yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

0.203km

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan pos setelah setiap 1/100 dari total jarak dari Tugu ke Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...
```

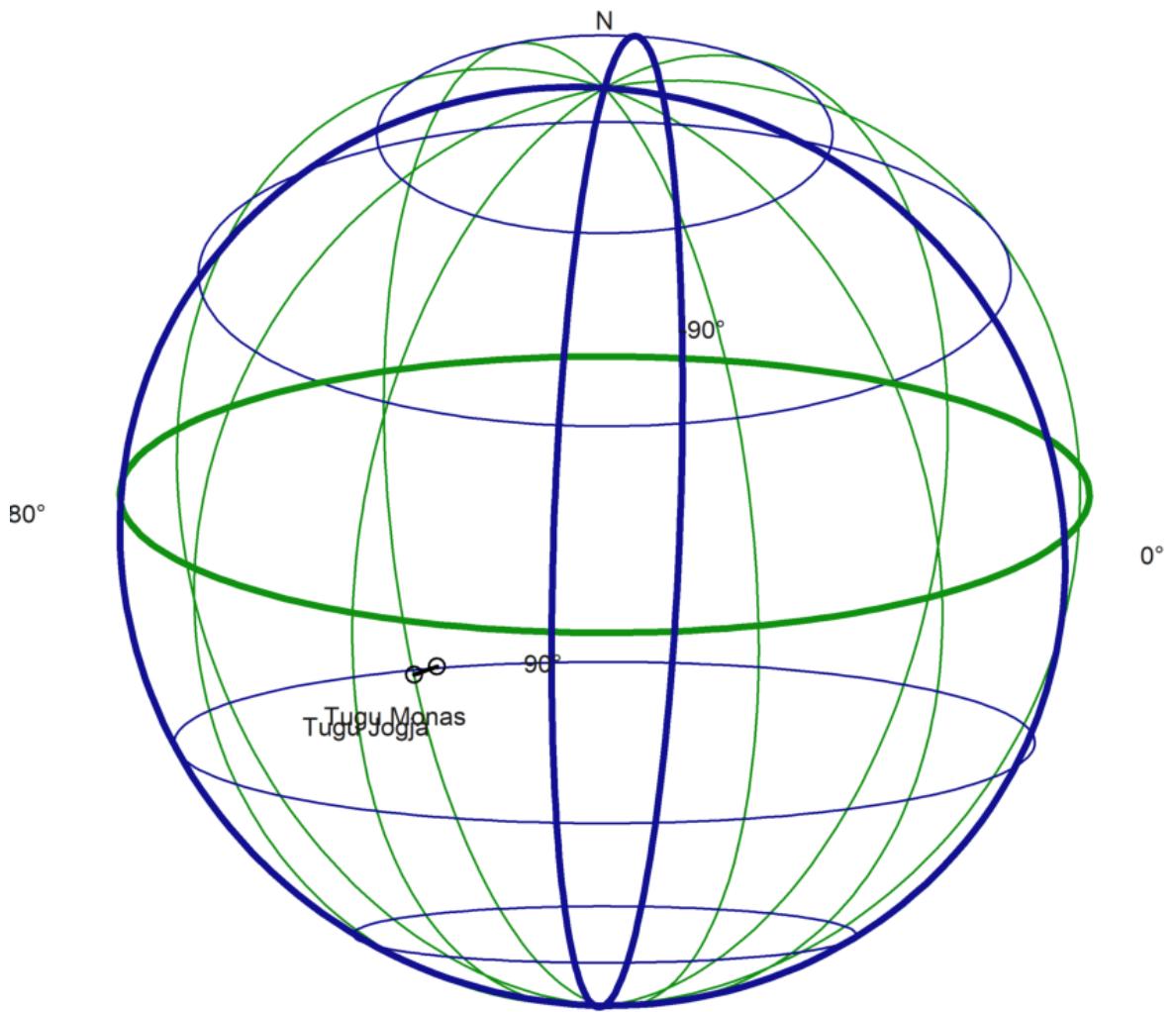
```
loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;
```

```
>skmprint(esdist(p,Monas))
```

0.000km

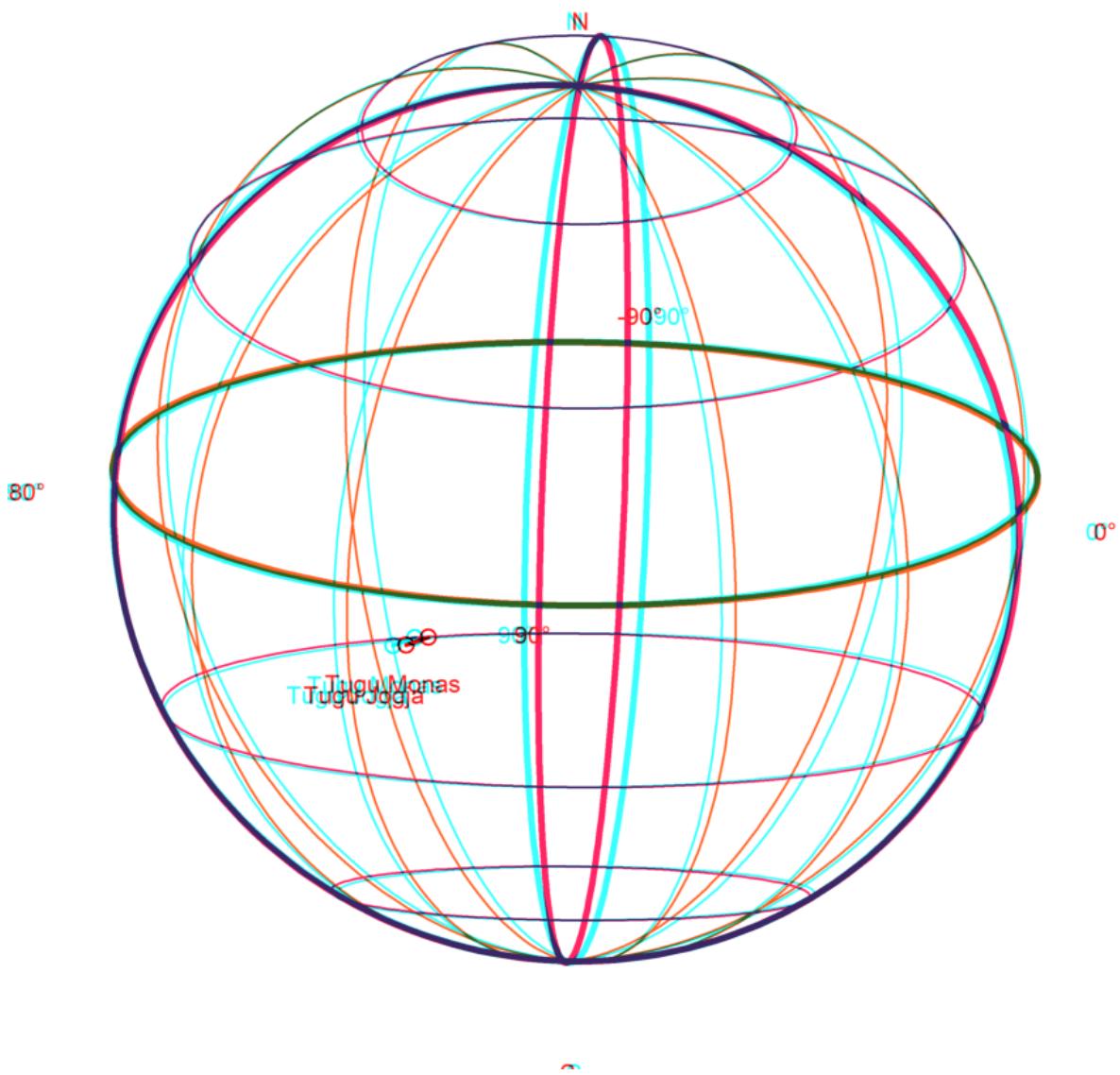
Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS di sepanjang lingkaran besar menuju Monas dengan fungsi navigasi.

```
>load spherical; v=navigate(Tugu,Monas,10); ...
  loop 1 to rows(v); sposprint(v[#]), end;
S 7°46.998' E 110°21.966'
S 7°37.422' E 110°0.573'
S 7°27.829' E 109°39.196'
S 7°18.219' E 109°17.834'
S 7°8.592' E 108°56.488'
S 6°58.948' E 108°35.157'
S 6°49.289' E 108°13.841'
S 6°39.614' E 107°52.539'
S 6°29.924' E 107°31.251'
S 6°20.219' E 107°9.977'
S 6°10.500' E 106°48.717'
Kami menulis sebuah fungsi, yang memplot bumi, dua posisi, dan posisi
di antaranya.
>function testplot ...
useglobal;
plotearth;
plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");
plotposline(v);
endfunction
Sekarang rencanakan semuanya.
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```



Atau gunakan `plot3d` untuk mendapatkan tampilan anaglyph. Ini terlihat sangat bagus dengan kacamata merah/sian.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4) :
```



## Latihan

1. Gambarlah segi- $n$  beraturan jika diketahui titik pusat  $O$ ,  $n$ , dan jarak titik pusat ke titik-titik sudut segi- $n$  tersebut (jari-jari lingkaran luar segi- $n$ ),  $r$ .

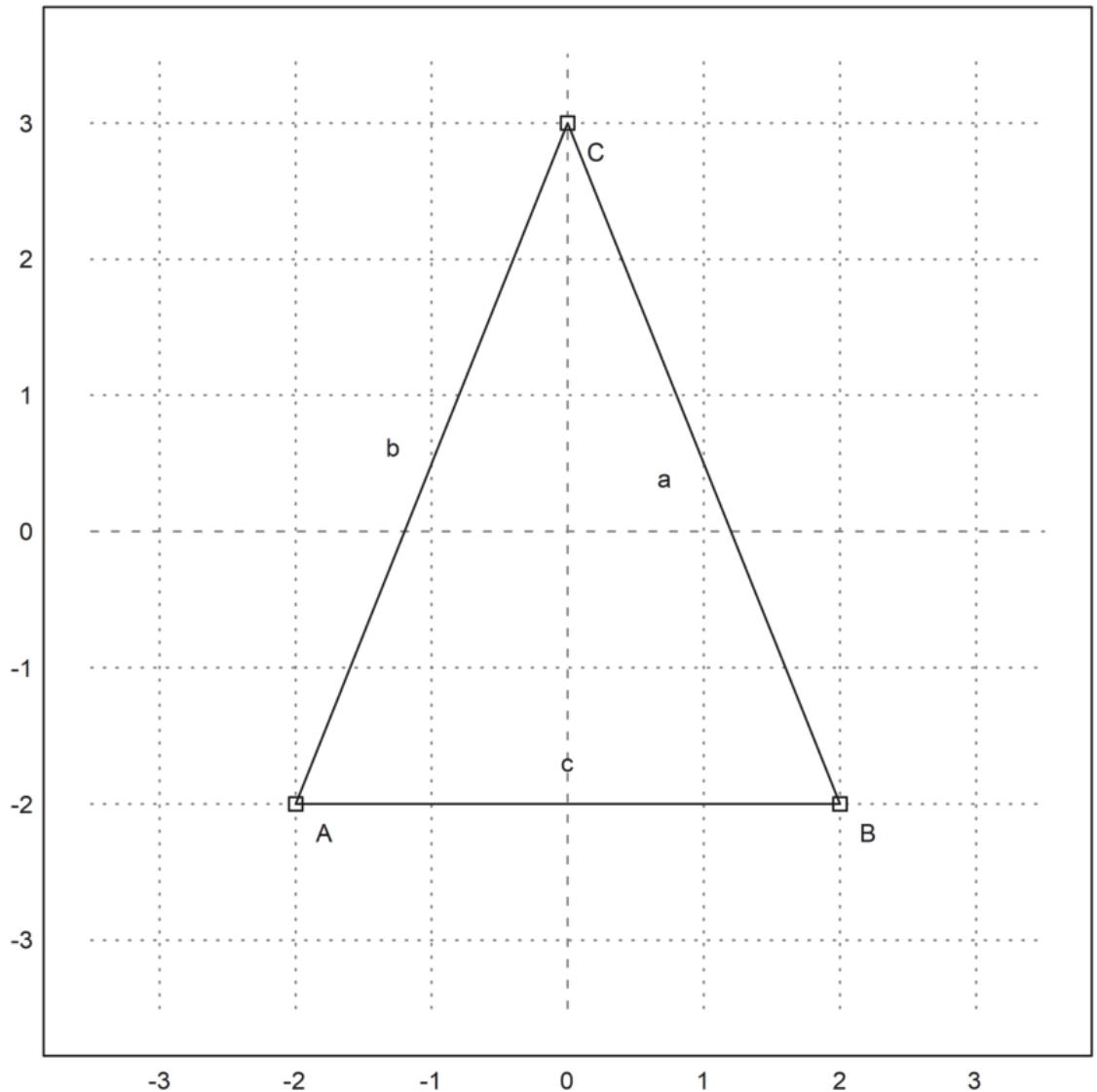
Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi- $n$  adalah  $(360/n)$ .
- Titik-titik sudut segi- $n$  merupakan perpotongan lingkaran luar segi- $n$  dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan  $(360/n)$ .
- Untuk  $n$  ganjil, pilih salah satu titik sudut adalah di atas.
- Untuk  $n$  genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat.
- Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

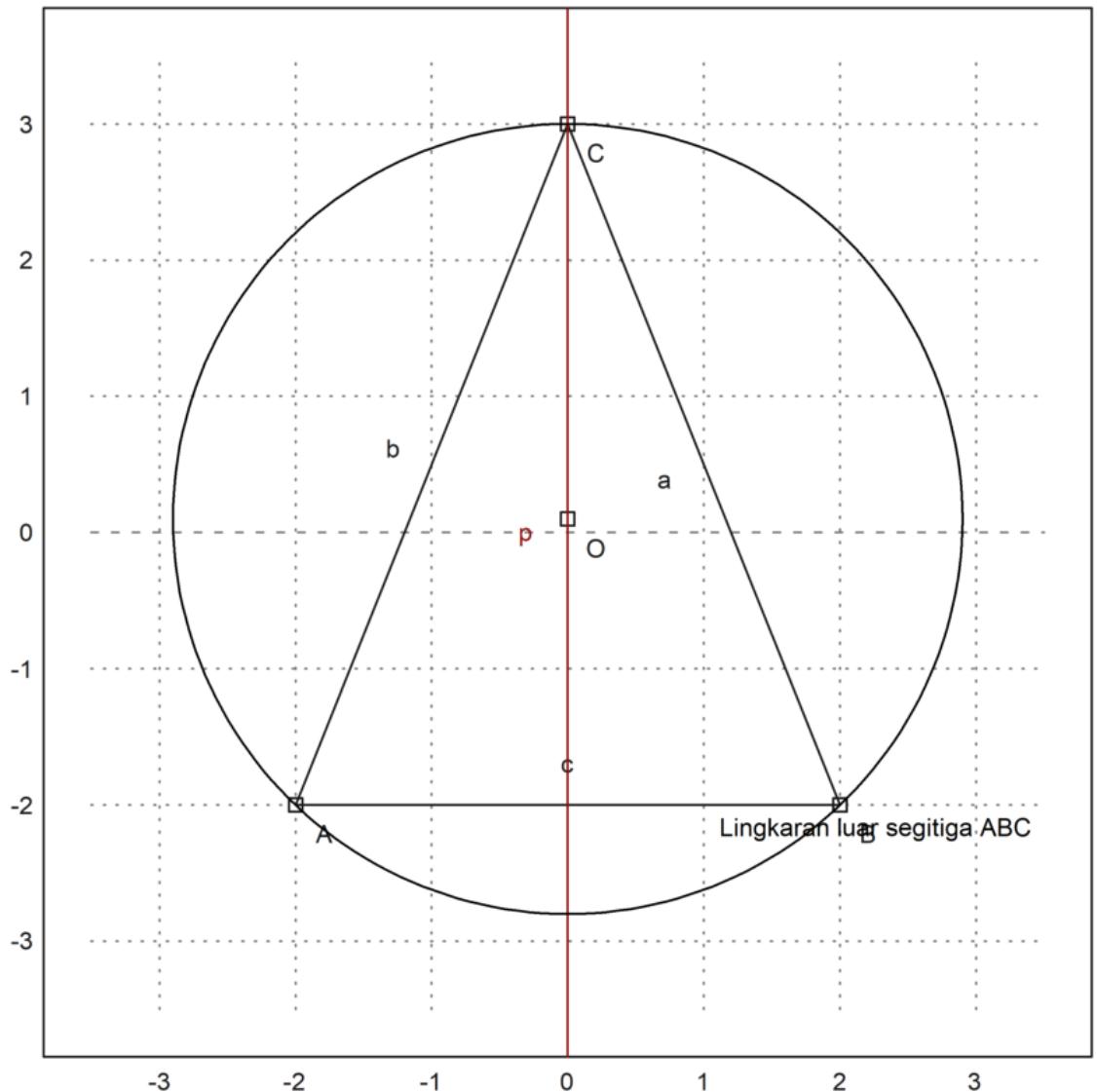
>load geometry

Numerical and symbolic geometry.

```
>setPlotRange(-3.5,3.5,-3.5,3.5);
>A=[-2,-2]; plotPoint(A,"A");
>B=[2,-2]; plotPoint(B,"B");
>C=[0,3]; plotPoint(C,"C");
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
>plotSegment(A,C,"b");
>aspect(1):
```



```
>c=circleThrough(A,B,C);
>R=getCircleRadius(c);
>O=getCircleCenter(c);
>plotPoint(O,"O");
>l=angleBisector(A,C,B);
>color(2); plotLine(l); color(1);
>plotCircle(c,"Lingkaran luar segitiga ABC"):
```

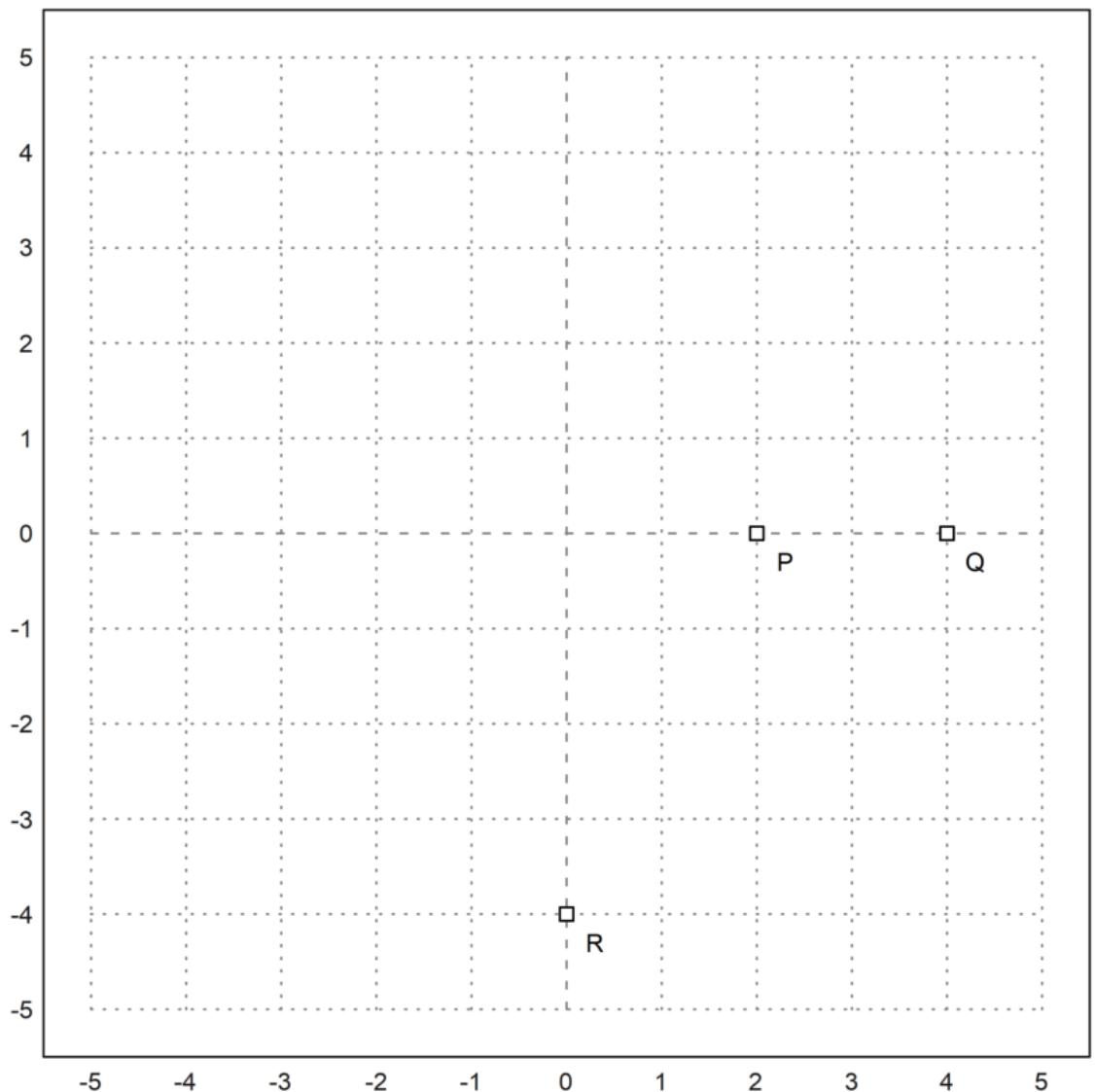


2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:

- Misalkan persamaan parabolanya  $y = ax^2 + bx + c$ .
- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai  $a$ ,  $b$ ,  $c$ .

```
>load geometry;
>setPlotRange(5); P=[2,0]; Q=[4,0]; R=[0,-4];
>plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R");
```



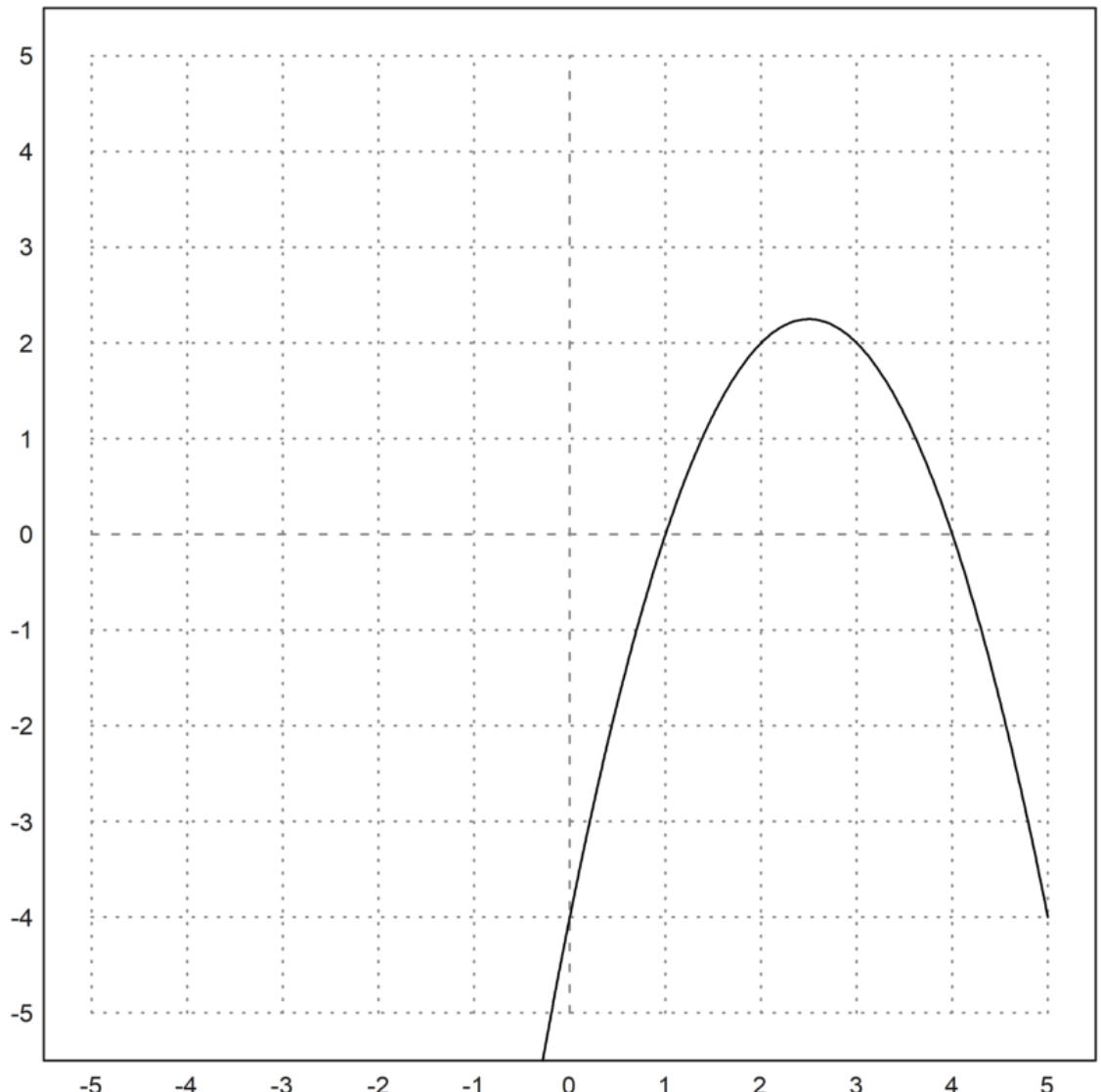
```

>sol &= solve([a+b=-c,16*a+4*b=-c,c=-4],[a,b,c])
      [[a = - 1, b = 5, c = - 4]]

>function y&=-x^2+5*x-4
      2
      - x  + 5 x - 4

>plot2d("-x^2+5*x-4", -5, 5, -5, 5):

```

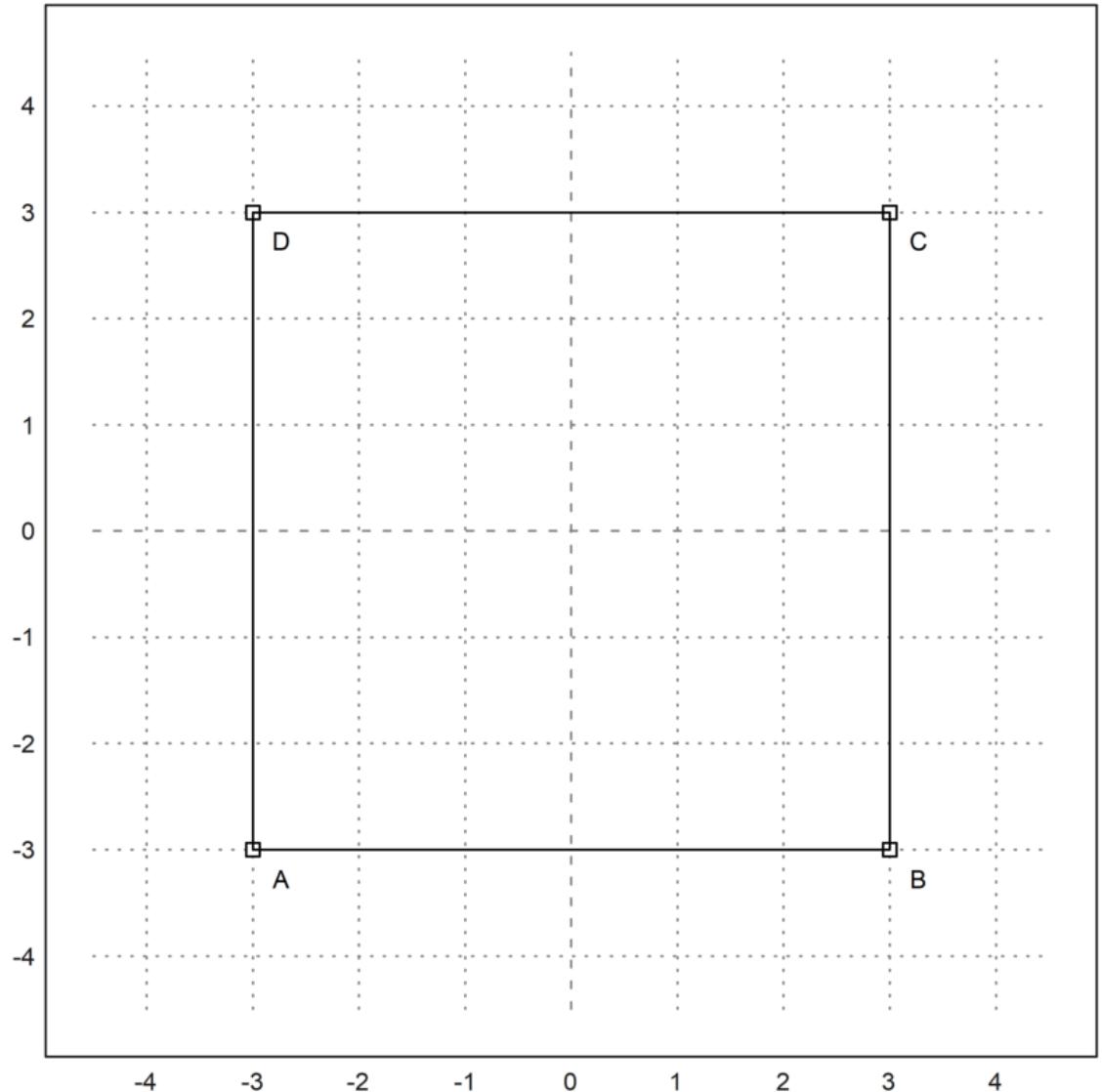


3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.

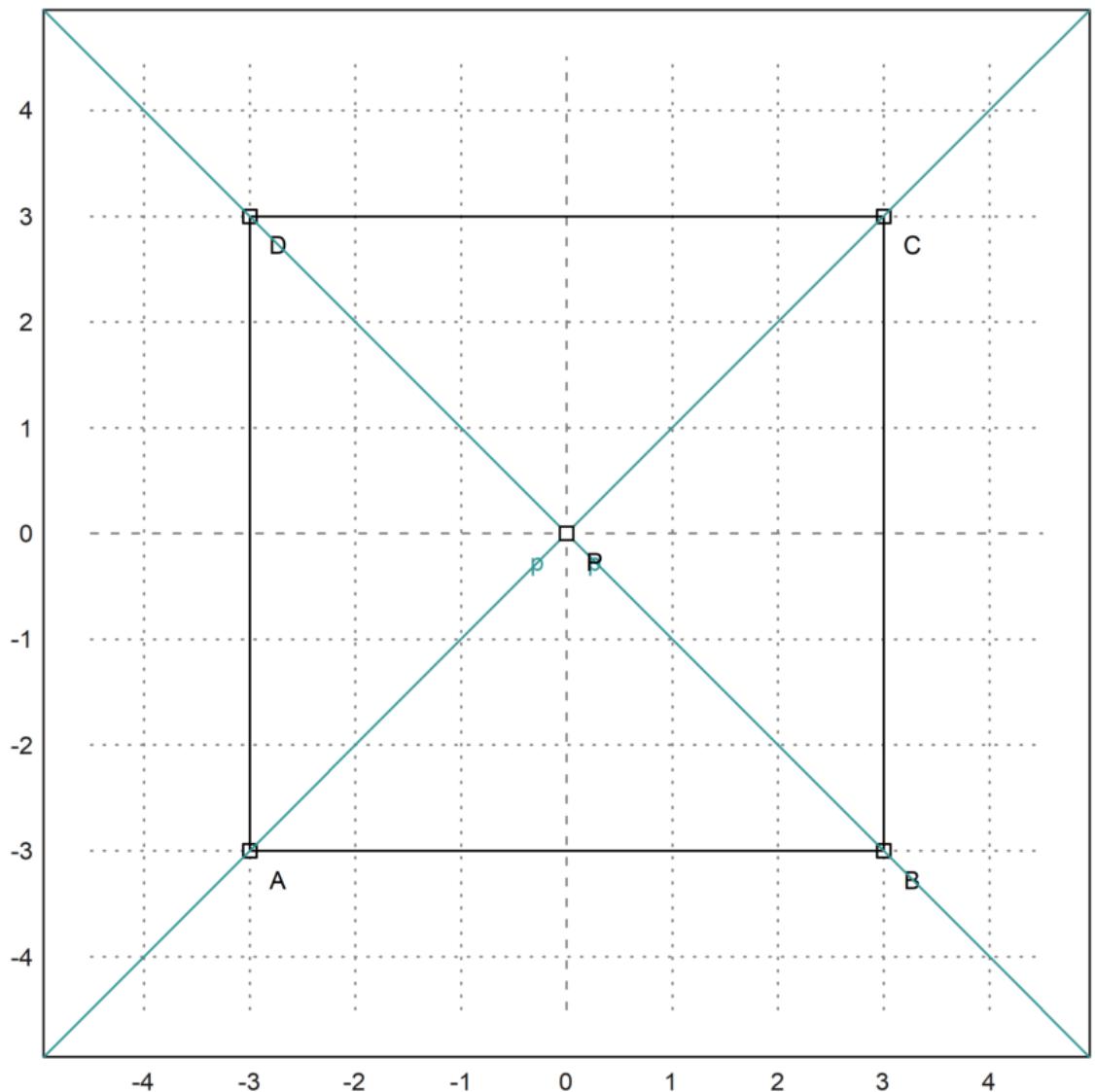
- Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung (sisinya-sisinya merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).
- Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya bertemu di satu titik.
- Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya.
- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

```
>load geometry
Numerical and symbolic geometry.
>setPlotRange(-4.5,4.5,-4.5,4.5);
>A=[-3,-3]; plotPoint(A,"A");
>B=[3,-3]; plotPoint(B,"B");
>C=[3,3]; plotPoint(C,"C");
>D=[-3,3]; plotPoint(D,"D");
>plotSegment(A,B,"");
>plotSegment(B,C,"");
```

```
>plotSegment(C,D,"");
>plotSegment(A,D,"");
>aspect(1):
```

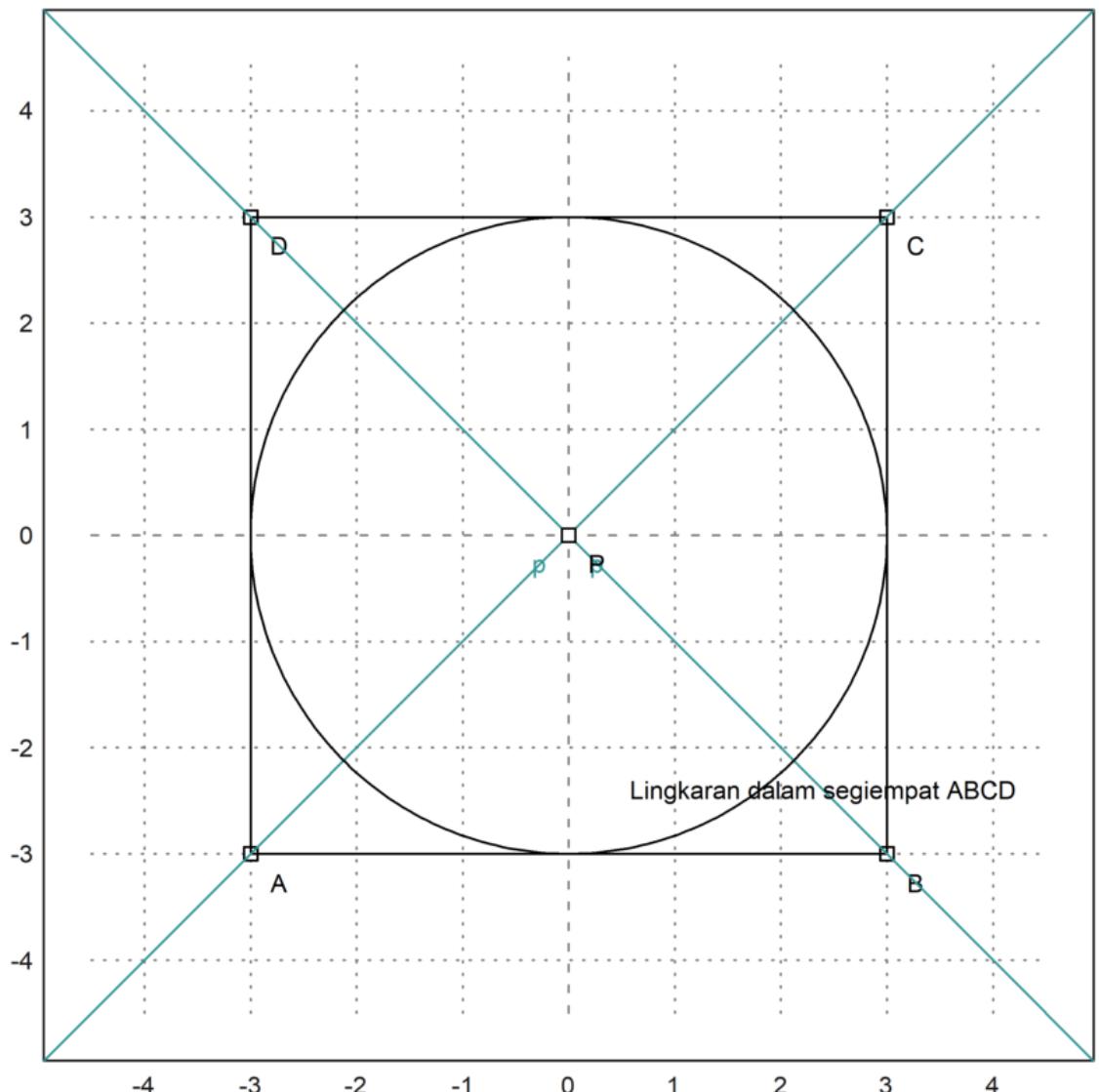


```
>l=angleBisector(A,B,C);
>m=angleBisector(B,C,D);
>P=lineIntersection(l,m);
>color(5); plotLine(l); plotLine(m); color(1);
>plotPoint(P,"P"):
```



Dari gambar diatas terlihat bahwa keempat garis bagi sudutnya bertemu di satu titik yaitu titik P.

```
>r=norm(P-projectToLine(P,lineThrough(A,B)));
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat ABCD");
```



Dari gambar diatas, terlihat bahwa sisi-sisinya merupakan garis singgung lingkaran yang sama yaitu lingkaran dalam segiempat. Akan ditunjukkan bahwa hasil kali panjang sisi-sisi yang berhadapan sama.

```

>AB=norm(A-B) //panjang sisi AB
6
>CD=norm(C-D) //panjang sisi CD
6
>AD=norm(A-D) //panjang sisi AD
6
>BC=norm(B-C) //panjang sisi BC
6
>AB.CD
36
>AD.BC
36

```

Terbukti bahwa hasil kali panjang sisi-sisi yang berhadapan sama yaitu 36. Jadi dapat dipastikan bahwa segiempat tersebut merupakan segiempat

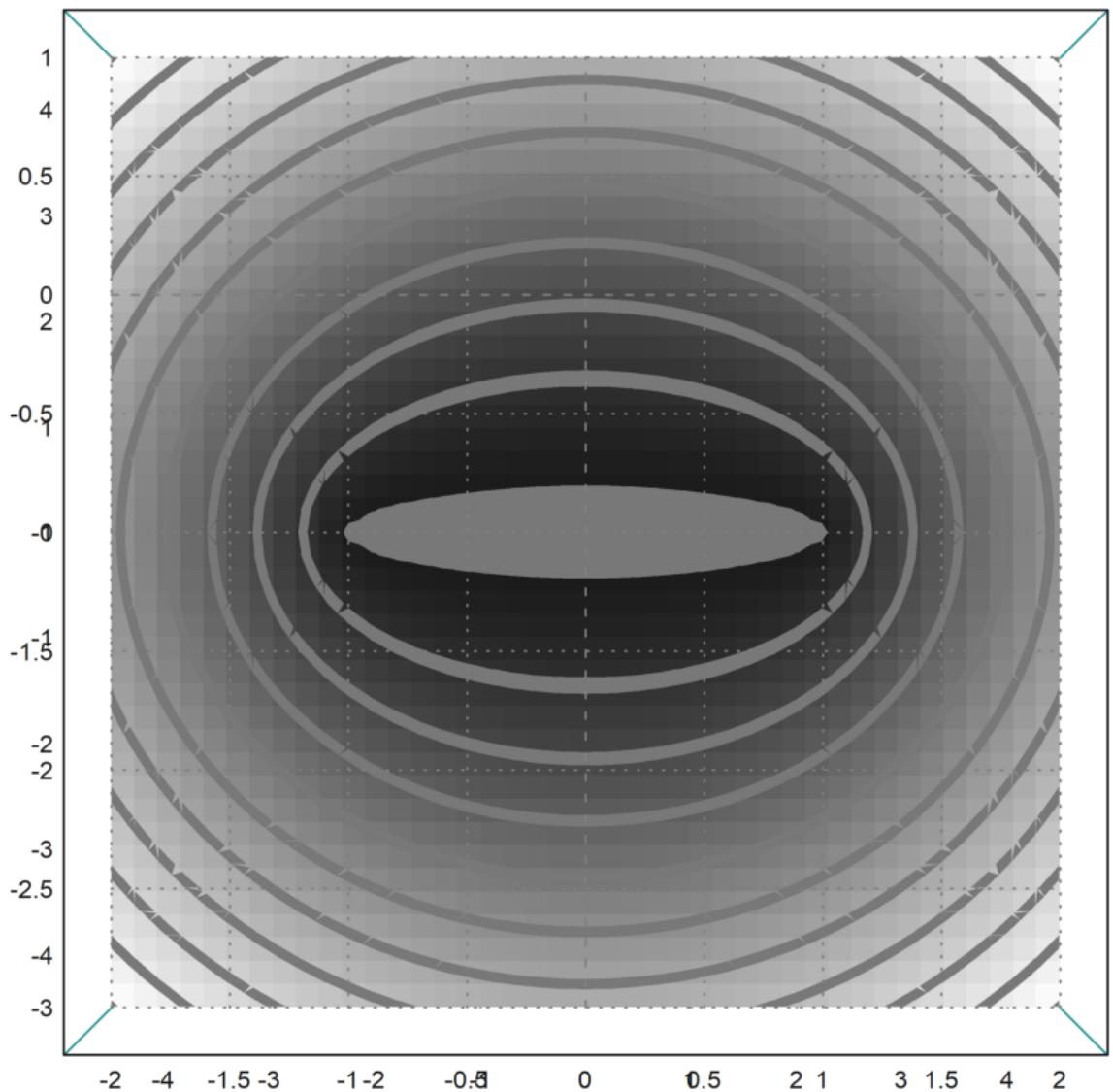
garis singgung.

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).

Penyelesaian :

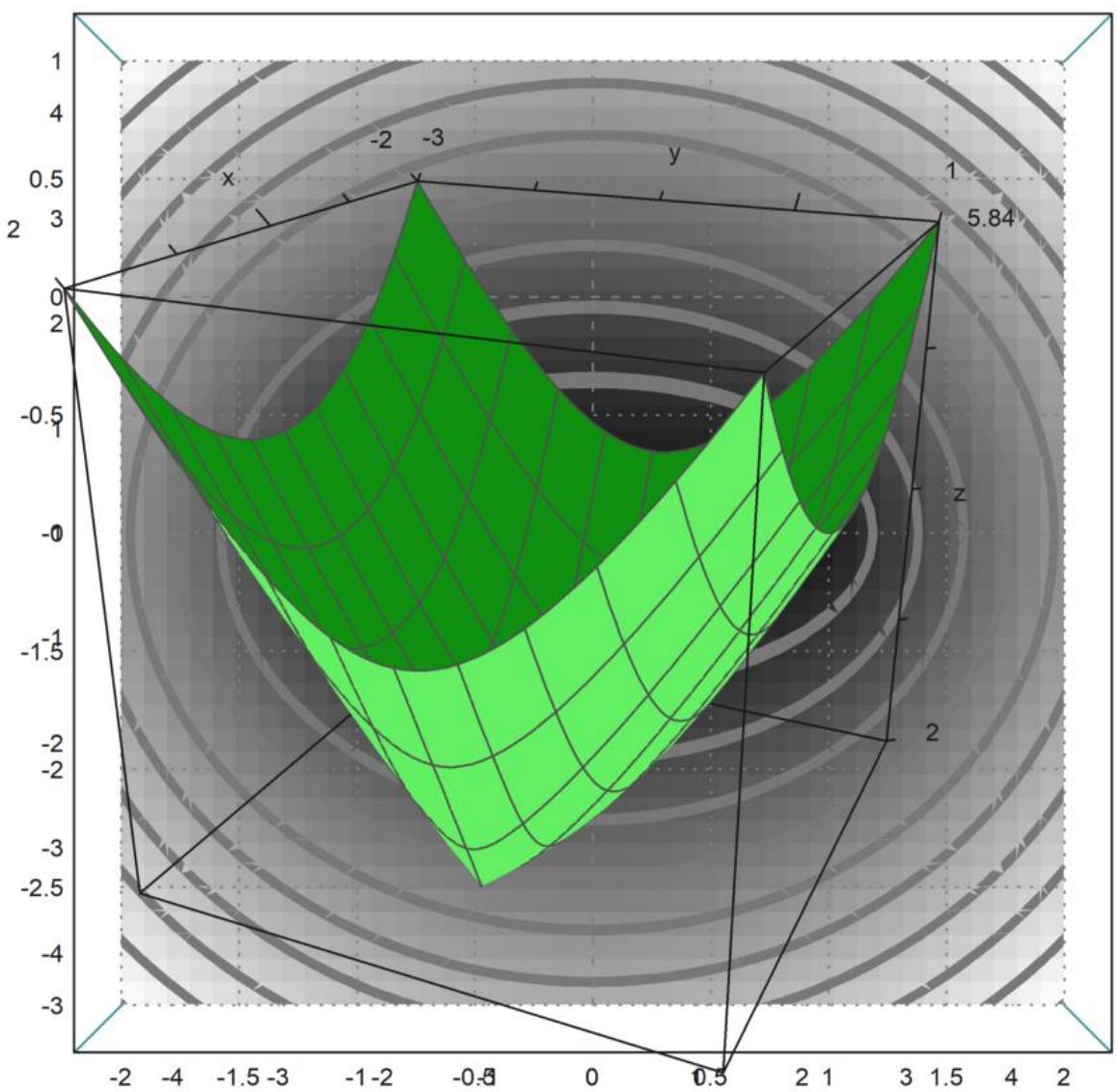
Diketahui kedua titik fokus  $P = [-1, -1]$  dan  $Q = [1, -1]$

```
>P=[-1,-1]; Q=[1,-1];
>function d1(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)
>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)+sqrt((x-
Q[1])^2+(y-Q[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



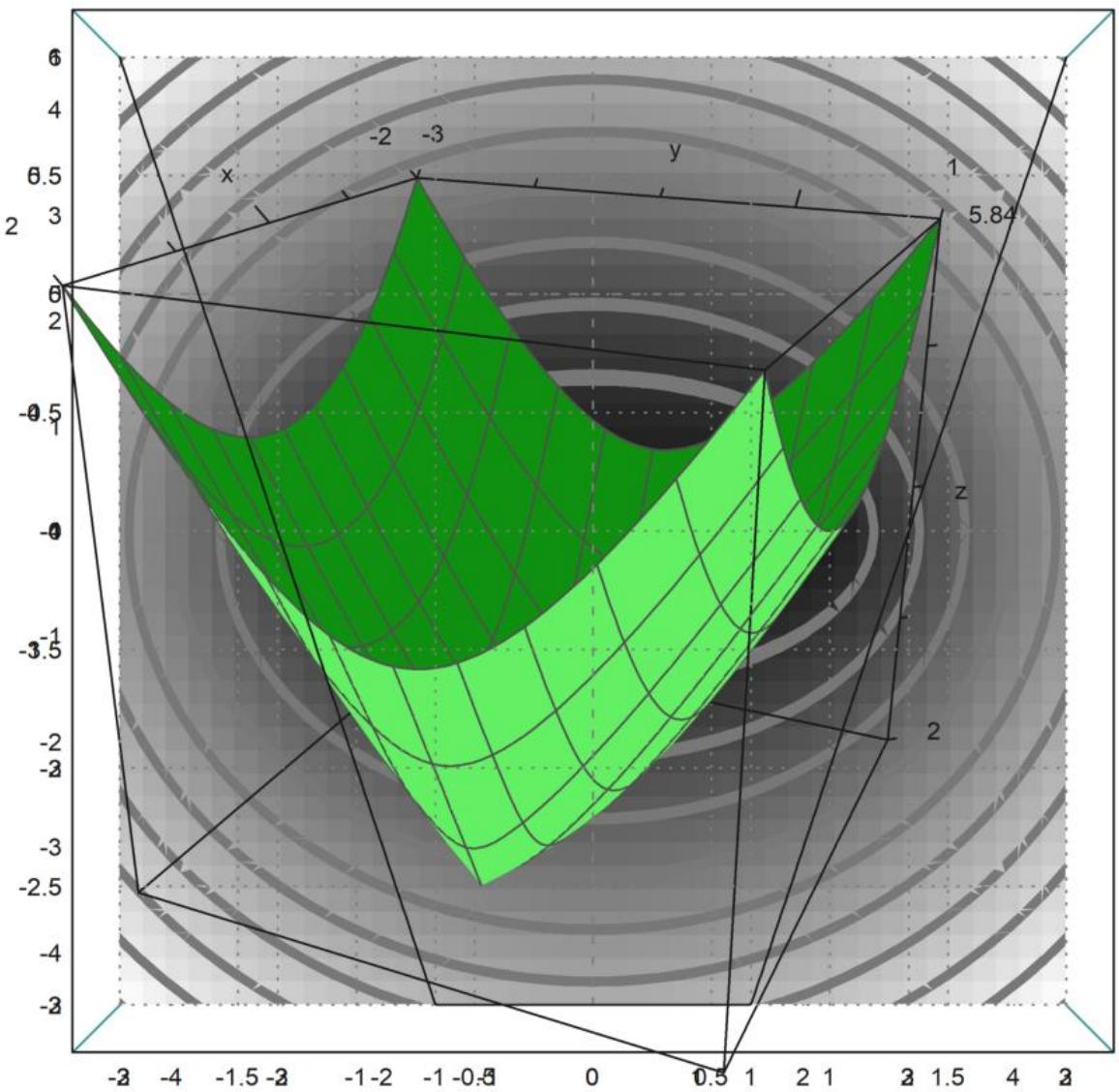
Grafik yang lebih menarik

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



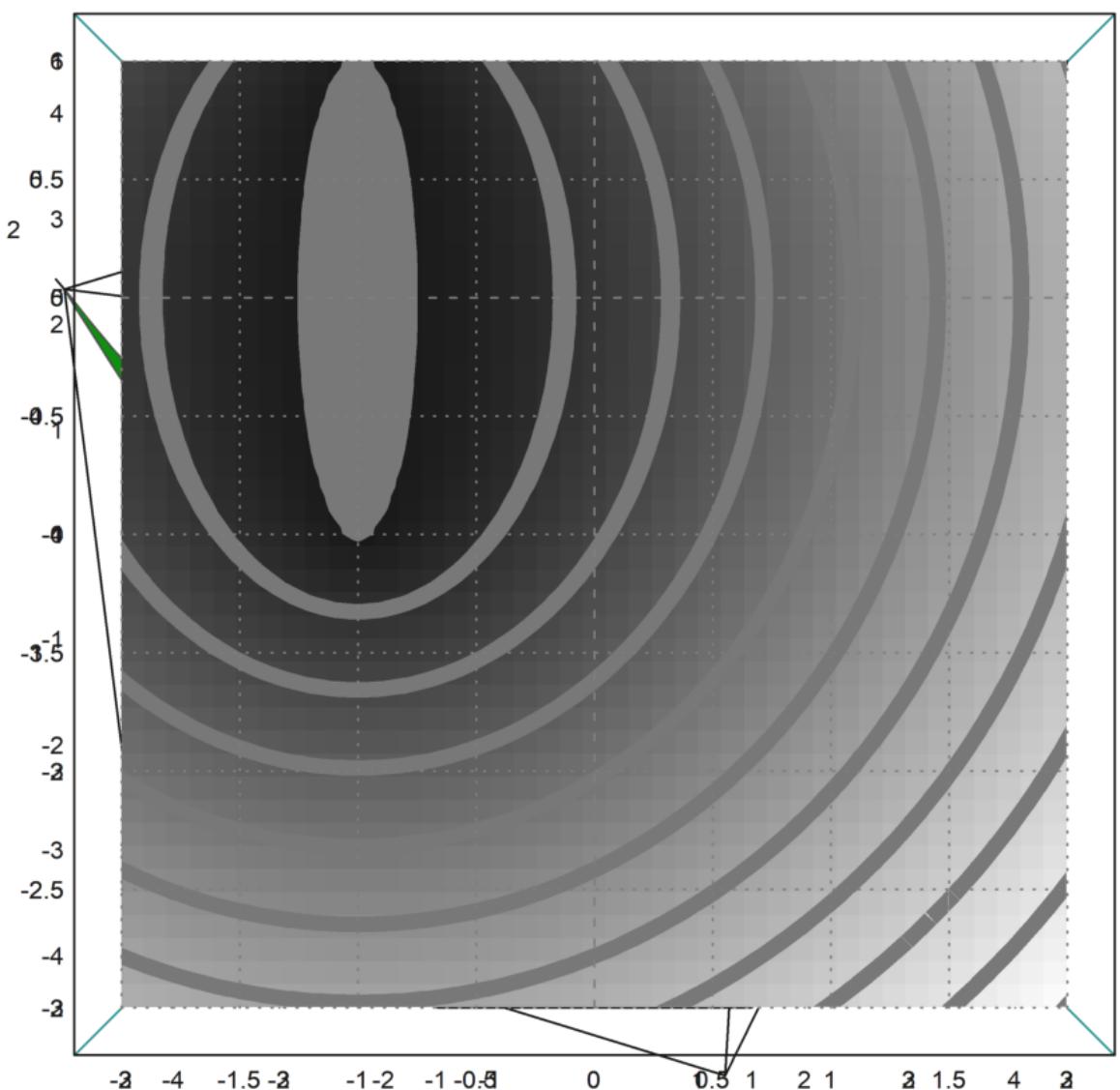
Batasan ke garis PQ

```
>plot2d("abs(x+1)+abs(x-1)", xmin=-3, xmax=3) :
```

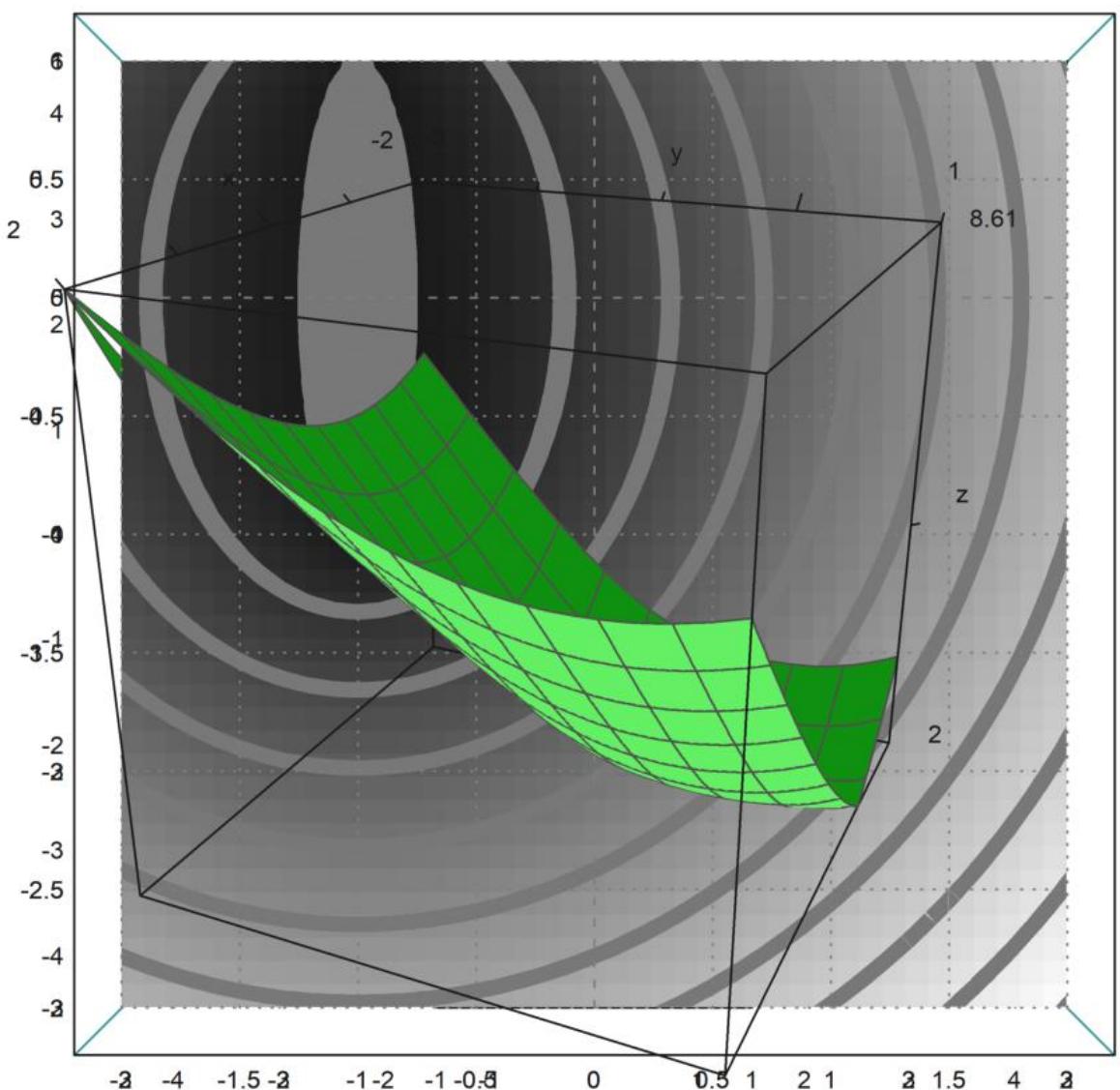


5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

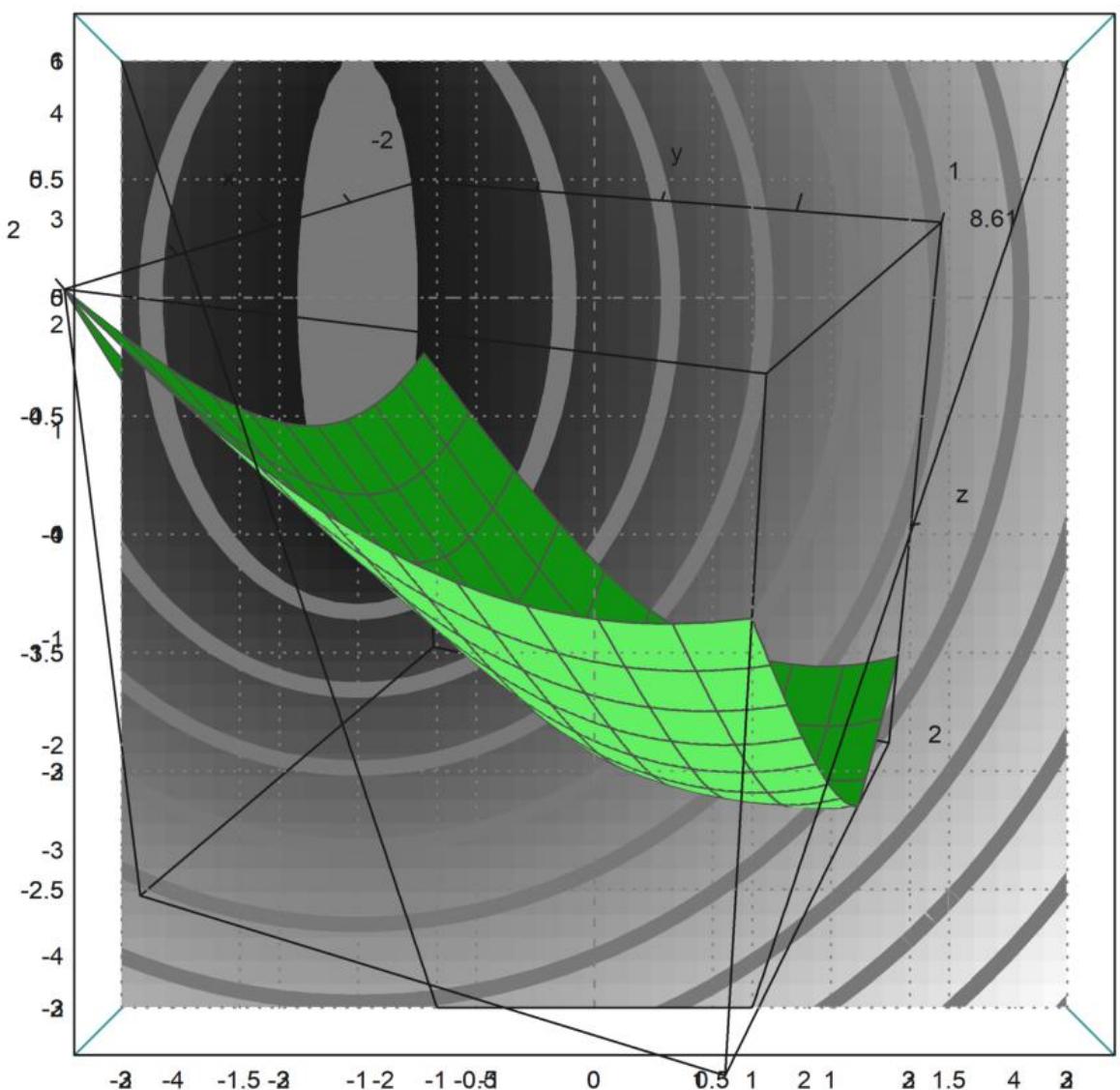
```
>P=[-1,-1]; Q=[1,-1];
>function d1(x,y):=sqrt((x-p[1])^2+(y-p[2])^2)
>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+(y-
P[2])^2)+sqrt((x+Q[1])^2+(y+Q[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



```
>plot2d("abs (x+1)+abs (x-1)",xmin=-3,xmax=3):
```



# EMT\_Statistika\_Adiyatma\_2303063006

## 2\_MatB

Adiyatma  
(23030630062)  
Matematika B

### EMT untuk Statistika

Di buku catatan ini, kami mendemonstrasikan plot statistik utama, pengujian, dan distribusi di Euler.

Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar statistik. Jadi, Anda mungkin memerlukan latar belakang untuk memahami detailnya.

Asumsikan pengukuran berikut. Kami ingin menghitung nilai rata-rata dan deviasi standar yang diukur.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...  
median(M), mean(M), dev(M),
```

```
999  
999.9  
2.72641400622
```

Kita dapat memplot plot kotak-dan-kumis untuk datanya. Dalam kasus kami, tidak ada outlier.

```
>aspect(1.75); boxplot(M);  
aspect (1.75) digunakan untuk mengatur rasio aspek dari plot  
(perbandingan antara lebar dan tinggi).  
boxplot(M) digunakan untuk membuat boxplot atau diagram kotak dari  
data di dalam variabel M. Boxplot adalah visualisasi statistik yang  
menunjukkan persebaran data, termasuk nilai minimum, median, dan  
nilai maksimum.
```

Contoh, kita asumsikan jumlah pria berikut dalam rentang ukuran tertentu.

```
>r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Berikut adalah alur pendistribusinya.

```
>plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="\"):
```

Kita bisa memasukkan data mentah tersebut ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal jangkauan, akhir jangkauan, jumlah pria dalam jangkauan.

Tabel dapat dicetak dengan header. Kami menggunakan vektor string untuk mengatur header.

```
>T:=r[1:8]' | r[2:9]' | v'; writetable(T,labc=["BB","BA","Frek"])
```

BB	BA	Frek
155.5	159.5	22
159.5	163.5	71
163.5	167.5	136
167.5	171.5	169
171.5	175.5	139
175.5	179.5	71
179.5	183.5	32

183.5        187.5        8  
Jika kita memerlukan nilai rata-rata dan statistik ukuran lainnya, kita perlu menghitung titik tengah rentang tersebut. Kita bisa menggunakan dua kolom pertama tabel kita untuk ini.

Simbol "|" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan opsi "labc" untuk menentukan header kolom.

```
>(T[,1]+T[,2])/2
 157.5
 161.5
 165.5
 169.5
 173.5
 177.5
 181.5
 185.5
>M=fold(r,[0.5,0.5])
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
Sekarang kita dapat menghitung mean dan deviasi sampel dengan frekuensi tertentu.
>{m,d}=meandev(M,v); m, d,
169.901234568
5.98912964449
Mari kita tambahkan distribusi nilai normal ke diagram batang di atas.
Rumus distribusi normal dengan mean m dan simpangan baku d adalah:
y=1d2π--√e-((x-m)2/2d2).y=1d2πe-(x-m)2/2d2.
```

Karena nilainya antara 0 dan 1, maka untuk memplotnya pada bar plot harus dikalikan dengan 4 kali jumlah data.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...
xmin=min(r),xmax=max(r),thickness=3,add=1):
```

## Tabel

---

Di direktori buku catatan ini Anda menemukan file dengan tabel. Data tersebut merupakan hasil survei. Berikut adalah empat baris pertama file tersebut. Datanya berasal dari buku online Jerman "Einführung in die Statistik mit R" oleh A. Handl.

```
>printfile("table.dat",4);
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
  open(filename,"r");
Tabel berisi 7 kolom angka atau token (string). Kami ingin membaca tabel dari file. Pertama, kami menggunakan terjemahan kami sendiri untuk tokennya.
```

Untuk ini, kami mendefinisikan kumpulan token. Fungsi strtokens() mendapatkan vektor string token dari string tertentu.

```
>mf:=["m","f"]; yn:=["y","n"]; ev:=strtokens("g vg m b vb");
Sekarang kita membaca tabel dengan terjemahan ini.
```

Argumen tok2, tok4 dll. adalah terjemahan dari kolom tabel. Argumen ini tidak ada dalam daftar parameter readtable(), jadi Anda perlu menyediakannya dengan ":".

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
>load over statistics;
>writetable(MT[1:10],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n

Titik "." mewakili nilai-nilai, yang tidak tersedia.

Jika kita tidak ingin menentukan token yang akan diterjemahkan terlebih dahulu, kita hanya perlu menentukan, kolom mana yang berisi token dan bukan angka.

```
>ctok=[2,4,5,7]; {MT,hd,tok}=readtable("table.dat",ctok=ctok);
```

ctok=[2,4,5,7]: Ini adalah untuk menentukan kolom yang akan diambil yaitu kolom ke-2, ke-4, ke-5, dan ke-7.

```
>tok
```

```
m
n
f
y
g
vg
```

Tabel berisi entri dari file dengan token yang diterjemahkan ke dalam angka.

String khusus NA = "." diartikan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) di tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAval.

```
>MT[1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Berikut isi tabel dengan nomor yang belum diterjemahkan.

```
>writetable(MT,wc=5)
```

1	1	30	2	.	1.8	2
2	3	23	4	5	1.8	2
3	3	26	4	5	1.8	4
4	1	33	2	.	2.8	2
5	1	37	2	.	1.8	2
6	1	28	4	5	2.8	4
7	3	31	4	6	2.8	2
8	1	23	2	.	0.8	2
9	3	24	4	6	1.8	4
10	1	26	2	.	1.8	2
11	3	23	4	6	1.8	4
12	1	32	4	5	1.8	2
13	1	29	4	6	1.8	4
14	3	25	4	5	1.8	4
15	3	31	4	5	0.8	2
16	1	26	4	5	2.8	2
17	1	37	2	.	3.8	2
18	1	38	4	5	.	2
19	3	29	2	.	3.8	2
20	3	28	4	6	1.8	2
21	3	28	4	1	2.8	4
22	3	28	4	6	1.8	4
23	3	38	4	5	2.8	2

```

24      3    27      4      1   1.8      4
25      1    27      2      .   2.8      4

```

Untuk kenyamanan, Anda dapat memasukkan keluaran `readtable()` ke dalam daftar.

```
>Table={{readtable("table.dat",ctok=ctok)}},
```

Dengan menggunakan kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan `ctok`, `tok`, dll.

atau menggunakan tabel daftar.

```
>writetable(Table,ctok=ctok,wc=5);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n
11	f	23	y	vg	1.8	y
12	m	32	y	g	1.8	n
13	m	29	y	vg	1.8	y
14	f	25	y	g	1.8	y
15	f	31	y	g	0.8	n
16	m	26	y	g	2.8	n
17	m	37	n	.	3.8	n
18	m	38	y	g	.	n
19	f	29	n	.	3.8	n
20	f	28	y	vg	1.8	n
21	f	28	y	m	2.8	y
22	f	28	y	vg	1.8	y
23	f	38	y	g	2.8	n
24	f	27	y	m	1.8	y
25	m	27	n	.	2.8	y

Fungsi `tablecol()` mengembalikan nilai kolom tabel, melewatkkan baris apa pun dengan nilai NAN ("." dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

```
>{c,i}=tablecol(MT,[5,6]);
```

Kita bisa menggunakan ini untuk mengekstrak kolom dari tabel untuk tabel baru.

```
>j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok)
```

Person	Evaluation	Tip
2	g	1.8
3	g	1.8
6	g	2.8
7	vg	2.8
9	vg	1.8
11	vg	1.8
12	g	1.8
13	vg	1.8
14	g	1.8
15	g	0.8
16	g	2.8
20	vg	1.8
21	m	2.8
22	vg	1.8
23	g	2.8
24	m	1.8

```

Tentu saja, kita perlu mengekstrak tabel itu sendiri dari daftar
Tabel
dalam kasus ini.
>MT=Table[1];
Tentu saja, kita juga dapat menggunakan untuk menentukan nilai
rata-rata suatu kolom atau nilai statistik lainnya.
>mean(tablecol(MT,6))
2.175
Fungsi getstatistics() mengembalikan elemen dalam vektor, dan
jumlahnya. Kami menerapkannya pada nilai "m" dan "f" di kolom kedua
tabel kami.
>{xu,count}=getstatistics(tablecol(MT,2)); xu, count,
[1, 3]
[12, 13]
Kita bisa mencetak hasilnya di tabel baru.
>writetable(count',labr=tok[xu])
      m      12
      f      13
Fungsi selecttable() mengembalikan tabel baru dengan nilai dalam satu
kolom yang dipilih dari vektor indeks. Pertama kita mencari indeks
dari dua nilai kita di tabel token.
>v:=indexof(tok,[ "g", "vg" ])
[5, 6]
Sekarang kita dapat memilih baris tabel, yang memiliki salah satu
nilai v pada baris ke-5.
>MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5);
Sekarang kita dapat mencetak tabel, dengan nilai yang diekstraksi dan
diurutkan di kolom ke-5.
>writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7);


| Person | Sex | Age | Titanic | Evaluation | Tip | Problem |
|--------|-----|-----|---------|------------|-----|---------|
| 2      | f   | 23  | y       | g          | 1.8 | n       |
| 3      | f   | 26  | y       | g          | 1.8 | y       |
| 6      | m   | 28  | y       | g          | 2.8 | y       |
| 18     | m   | 38  | y       | g          | .   | n       |
| 16     | m   | 26  | y       | g          | 2.8 | n       |
| 15     | f   | 31  | y       | g          | 0.8 | n       |
| 12     | m   | 32  | y       | g          | 1.8 | n       |
| 23     | f   | 38  | y       | g          | 2.8 | n       |
| 14     | f   | 25  | y       | g          | 1.8 | y       |
| 9      | f   | 24  | y       | vg         | 1.8 | y       |
| 7      | f   | 31  | y       | vg         | 2.8 | n       |
| 20     | f   | 28  | y       | vg         | 1.8 | n       |
| 22     | f   | 28  | y       | vg         | 1.8 | y       |
| 13     | m   | 29  | y       | vg         | 1.8 | y       |
| 11     | f   | 23  | y       | vg         | 1.8 | y       |


Untuk statistik selanjutnya, kami ingin menghubungkan dua kolom
tabel.
Jadi kita ekstrak kolom 2 dan 4 dan urutkan tabelnya.
>i=sortedrows(MT,[2,4]); ...
writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)
      m      n
      m      n
      m      n
      m      n
      m      n
      m      n
      m      n
      m      y
      m      y
      m      y
      m      y

```

m		y
f		n
f		y
f		y
f		y
f		y
f		y
f		y
f		y
f		y
f		y
f		y
f		y
f		y
f		y

Dengan `getstatistics()`, kita juga bisa menghubungkan jumlah dalam dua kolom tabel satu sama lain.

```
>MT24=tablecol(MT,[2,4]); ...
{xu1,xu2,count}=getstatistics(MT24[1],MT24[2]); ...
writetable(count,labr=tok[xu1],labc=tok[xu2])
```

	n	y
m	7	5
f	1	12

Sebuah tabel dapat ditulis ke file.

```
>filename="test.dat"; ...
writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
Kemudian kita bisa membaca tabel dari file tersebut.
```

```
>{MT2,hd,tok2,hdr}=readtable(filename,>clabs,>rlabs); ...
writetable(MT2,labr=hdr,labc=hd)
```

	n	y
m	7	5
f	1	12

Dan hapus filenya.

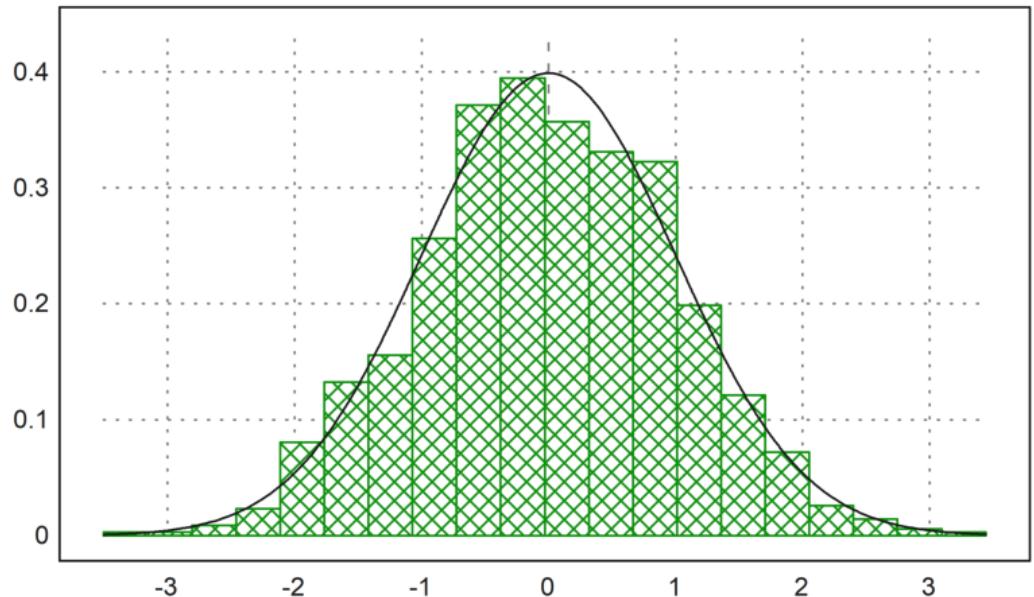
```
>fileremove(filename);
```

## Distribusi

---

Dengan `plot2d`, ada metode yang sangat mudah untuk memplot sebaran data eksperimen.

```
>p=normal(1,1000);
>plot2d(p,distribution=20,style="\/");
>plot2d("qnormal(x,0,1)",add=1):
```

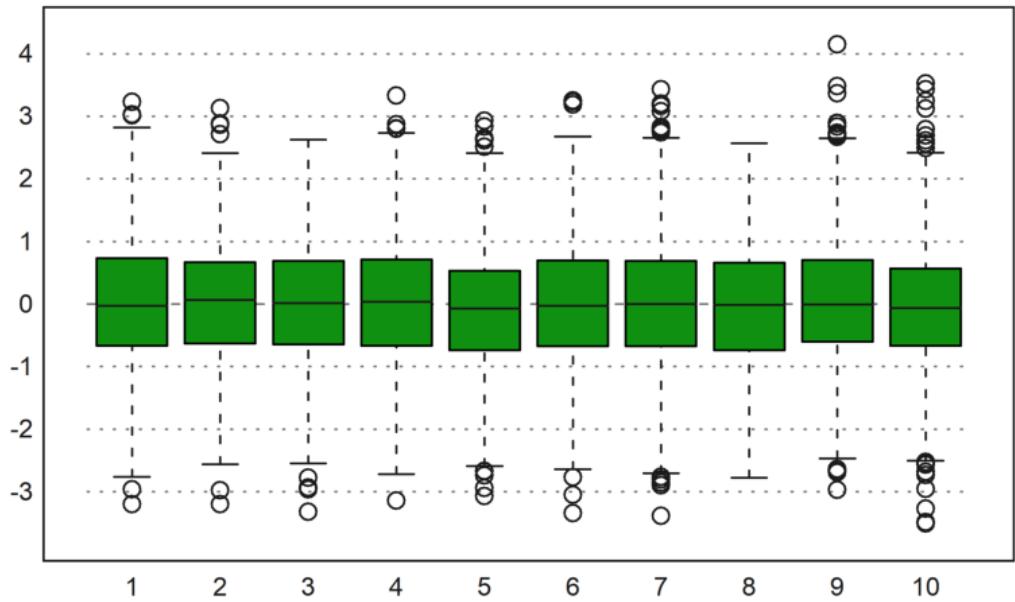


`p=normal(1,1000);` digunakan untuk menciptakan 1000 sampel acak yang terdistribusi normal dengan mean (rata-rata) 1 dan standar deviasi 1000.

`plot2d("qnormal(x,0,1)",add=1);`  
digunakan untuk menambahkan plot dari distribusi normal standar (dengan mean 0 dan standar deviasi 1) ke grafik yang sama. Fungsi `qnormal(x,0,1)` mengacu pada distribusi kumulatif dari variabel acak normal standar. `add=1` menunjukkan bahwa grafik ini harus ditambahkan ke grafik yang sudah ada, bukan dibuat baru.

Perlu diperhatikan perbedaan antara bar plot (sampel) dan kurva normal (distribusi sebenarnya). Masukkan kembali ketiga perintah untuk melihat hasil pengambilan sampel lainnya.  
Berikut adalah perbandingan 10 simulasi dari 1000 nilai terdistribusi normal menggunakan apa yang disebut plot kotak. Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, serta outlier.

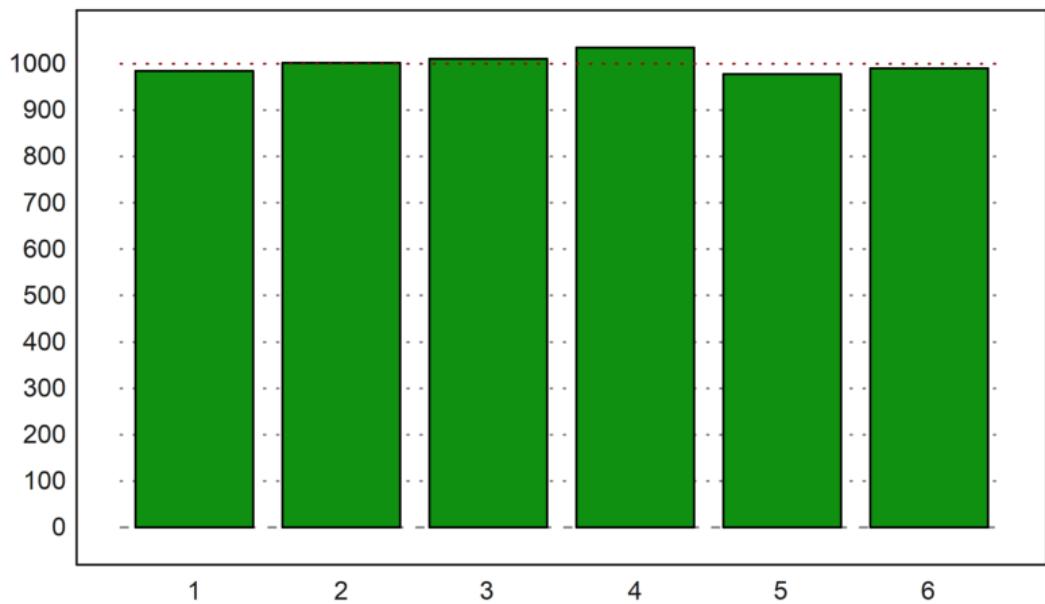
```
>p=normal(10,1000); boxplot(p);
```



Untuk menghasilkan bilangan bulat acak, Euler memiliki `intrandom`.  
Mari kita simulasi lemparan dadu dan plot distribusinya.

Kita menggunakan fungsi `getmultiplicities(v,x)`, yang menghitung seberapa sering elemen  $v$  muncul di  $x$ . Kemudian kita plot hasilnya menggunakan `kolomplot()`.

```
>k=intrandom(1,6000,6); ...
columnsplot(getmultiplicities(1:6,k)); ...
ygrid(1000,color=red);
```



Meskipun `intrandom(n,m,k)` mengembalikan bilangan bulat yang terdistribusi secara seragam dari 1 hingga  $k$ , distribusi bilangan

bulat lainnya dapat digunakan dengan `randpint()`.

Dalam contoh berikut, probabilitas untuk 1,2,3 masing-masing adalah 0,4,0.1,0.5.

```
>randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,%)  
[378, 102, 520]
```

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Lihat referensinya.

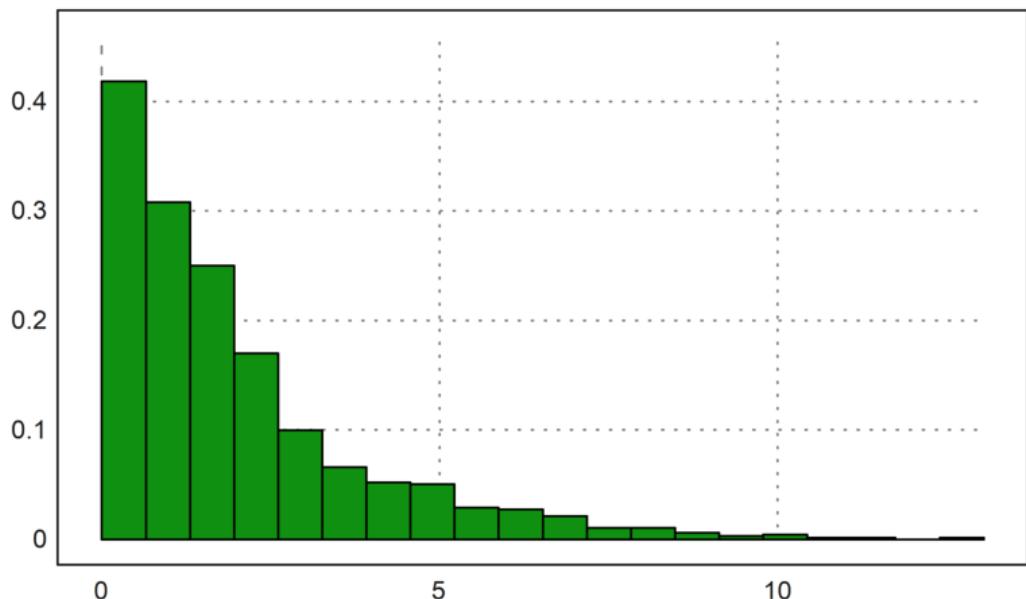
Misalnya, kita mencoba distribusi eksponensial. Variabel acak kontinu X dikatakan berdistribusi eksponensial, jika PDF-nya diberikan oleh

$$f(x) = \lambda e^{-\lambda x}, x > 0, \lambda > 0, f_X(x) = \lambda e^{-\lambda x}, x > 0, \lambda > 0,$$

with parameter

$\lambda=1/\mu$ ,  $\mu$  is the mean, and denoted by  $X \sim \text{Exponential}(\lambda)$ .  $\lambda=1/\mu$ ,  $\mu$  is the mean, and denoted by  $X \sim \text{Exponential}(\lambda)$ .

```
>plot2d(randexponential(1,1000,2),>distribution):
```

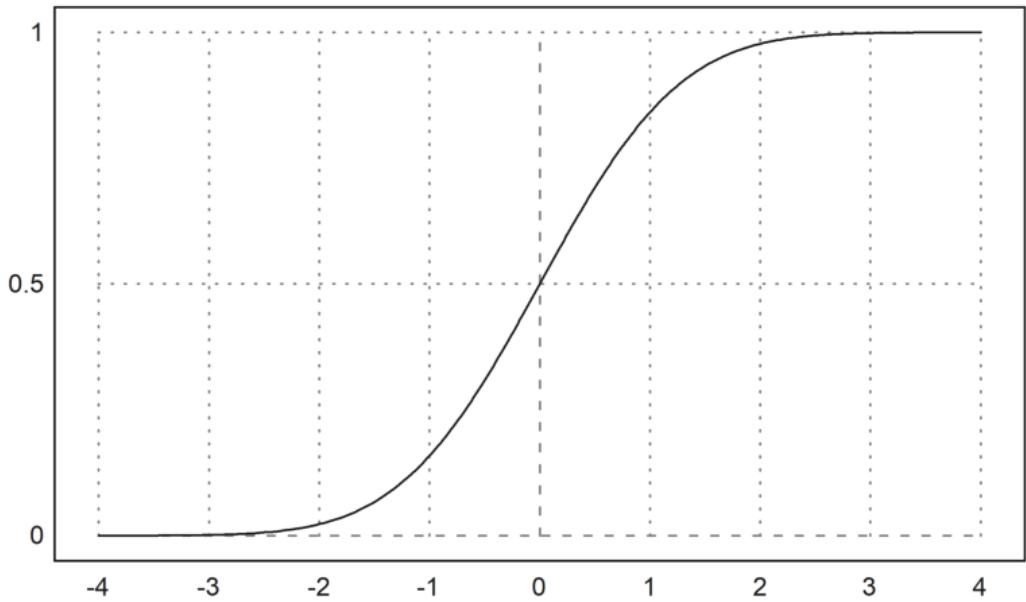


Parameter pertama (1) adalah lambda, yang merupakan parameter distribusi eksponensial.

Parameter kedua (1000) menunjukkan jumlah angka acak yang dihasilkan. Parameter ketiga (2) bisa menunjukkan dimensi atau bentuk output.

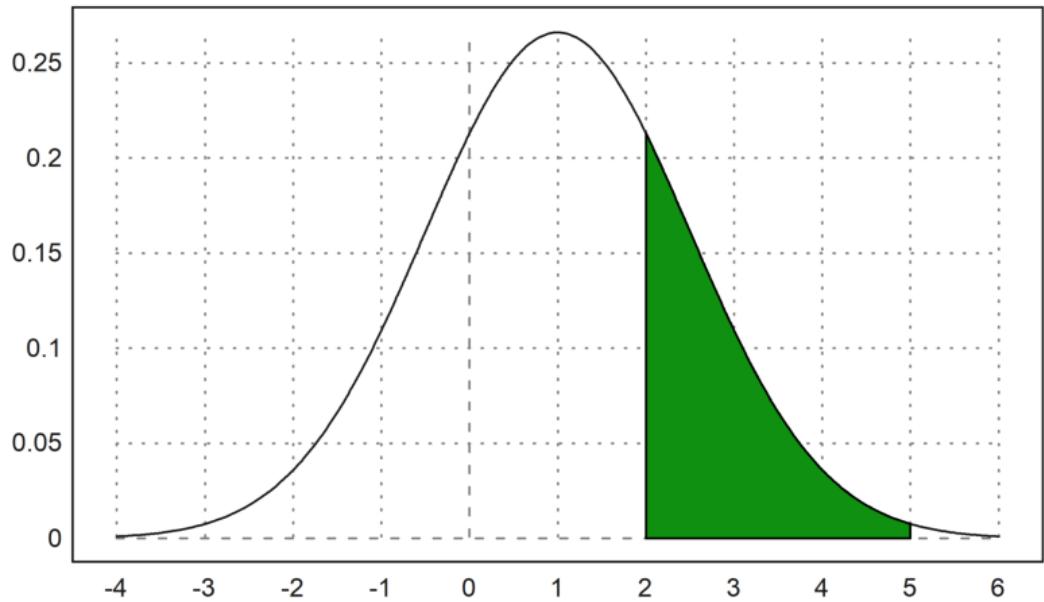
Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversnya.

```
>plot2d("normaldis",-4,4):
```



Berikut ini adalah salah satu cara untuk memplot kuantil.

```
>plot2d("qnormal(x,1,1.5)",-4,6); ...
plot2d("qnormal(x,1,1.5)",a=2,b=5,>add,>filled):
```



$$\text{normaldis}(x,m,d) = \int_{-\infty}^x 1/2\pi e^{-\frac{1}{2}(t-m)^2} dt. \quad \text{normaldis}(x,m,d) = \int_{-\infty}^x 1/2\pi e^{-\frac{1}{2}(t-m)^2} dt.$$

Peluang berada di kawasan hijau adalah sebagai berikut.

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
0.248662156979
```

Ini dapat dihitung secara numerik dengan integral berikut.

$$\int_{2}^{5} 1/2\pi e^{-\frac{1}{2}(x-1.5)^2} dx. \quad \int_{2}^{5} 1/2\pi e^{-\frac{1}{2}(x-1.5)^2} dx.$$

```
>gauss("qnormal(x,1,1.5)",2,5)
```

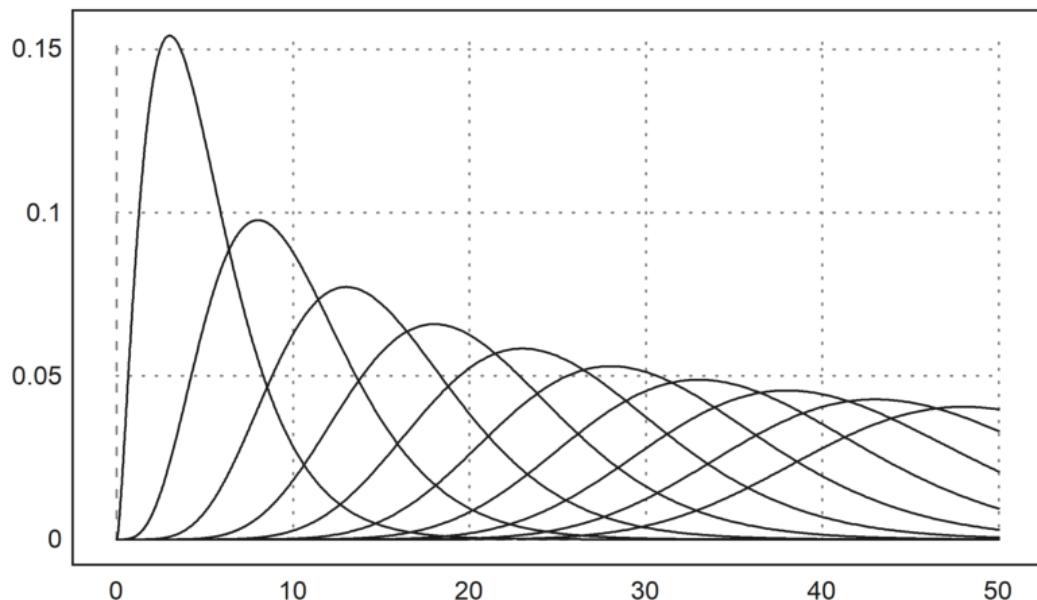
```
0.248662156979  
Mari kita bandingkan distribusi binomial dengan distribusi normal  
yang  
mean dan deviasinya sama. Fungsi invbindis() menyelesaikan  
interpolasi  
linier antara nilai integer.
```

```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

```
525.516721219  
526.007419394
```

Fungsi qdis() adalah kepadatan distribusi chi-kuadrat. Seperti biasa, Euler memetakan vektor ke fungsi ini. Dengan demikian kita mendapatkan plot semua distribusi chi-kuadrat dengan derajat 5 sampai 30 dengan mudah dengan cara berikut.

```
>plot2d("qchidis(x,(5:5:50))",0,50):
```



Euler memiliki fungsi akurat untuk mengevaluasi distribusi. Mari kita periksa chidis() dengan integral.

Penamaannya mencoba untuk konsisten. Misalnya.,

- distribusi chi-kuadratnya adalah chidis(),
- fungsi kebalikannya adalah invchidis(),
- kepadatannya adalah qchidis().

Pelengkap distribusi (ekor atas) adalah chicdis().

```
>chidis(1.5,2), integrate("qchidis(x,2)",0,1.5)
```

```
0.527633447259  
0.527633447259
```

## Distribusi Diskrit

Distribusi diskret adalah jenis distribusi probabilitas yang digunakan untuk variabel acak diskret, yaitu variabel yang hanya dapat memiliki nilai tertentu, biasanya dalam bentuk bilangan bulat.

Untuk menentukan distribusi diskrit Anda sendiri, Anda dapat menggunakan metode berikut.

Pertama kita atur fungsi distribusinya.

```
>wd = 0|((1:6)+[-0.01,0.01,0,0,0,0])/6  
[0, 0.165, 0.335, 0.5, 0.666667, 0.833333, 1]  
Perintah ini menggunakan operator | dan + untuk membuat nilai dalam variabel wd.
```

1:6 Ini menghasilkan vektor [1, 2, 3, 4, 5, 6].

(1:6) + [-0.01, 0.01, 0, 0, 0, 0]: Operasi ini menambahkan kedua vektor elemen per elemen.

Hasilnya:

```
[1-0.01,2+0.01,3,4,5,6]=[0.99,2.01,3,4,5,6][1-0.01,2+0.01,3,4,5,6]=[0.99,2.01,3,4,5,6]
```

[1-0.01,2+0.01,3,4,5,6]=[0.99,2.01,3,4,5,6]/6 Membagi setiap elemen hasil penjumlahan tadi dengan 6.

Hasilnya:

```
[0.996,2.016,36,46,56,66]=[0.165,0.335,0.5,0.6667,0.8333,1][0.996,2.016,36,46,56,66]=[0.165,0.335,0.5,0  
.6667,0.8333,1]
```

Artinya dengan probabilitas  $wd[i+1] - wd[i]$  kita menghasilkan nilai acak

i.

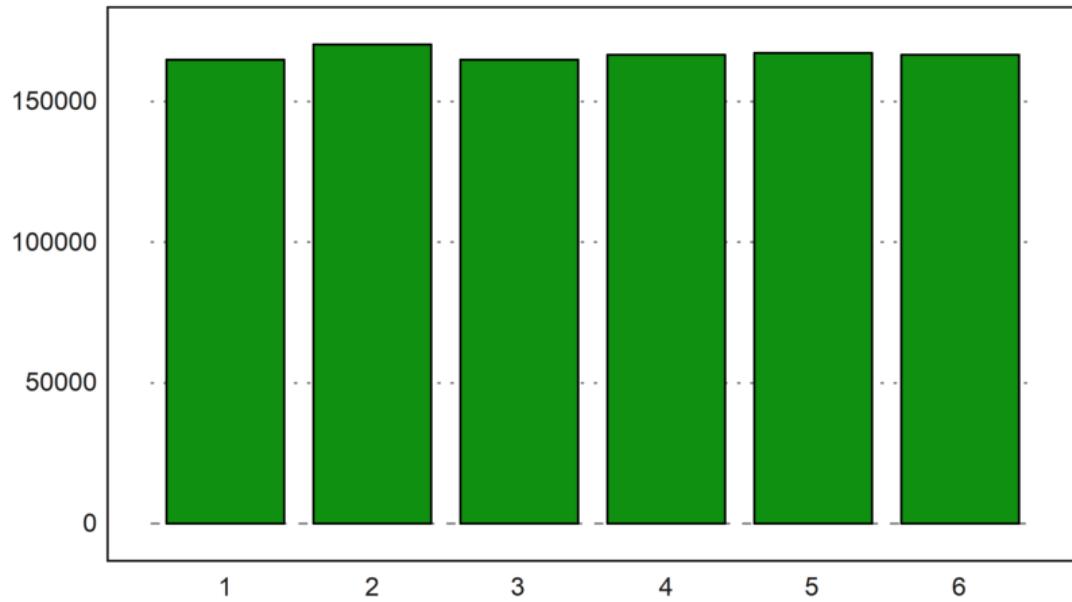
Ini hampir merupakan distribusi yang seragam. Mari kita tentukan generator nomor acak untuk ini. Fungsi `find(v,x)` mencari nilai x pada vektor v. Fungsi ini juga berfungsi untuk vektor x.

```
>function wrongdice (n,m) := find(wd,random(n,m))  
Kesalahannya sangat halus sehingga kita hanya melihatnya dengan  
banyak  
iterasi.
```

Fungsi `wrongdice` mengembalikan sebuah matriks berukuran n x m, di mana

setiap elemen dari matriks ini adalah indeks posisi dari elemen wd yang paling sesuai (atau mendekati) nilai acak dari `random(n, m)`.

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```



Hasil columnsplot akan menunjukkan frekuensi relatif dari setiap angka (1 hingga 6), yang memungkinkan Anda untuk melihat apakah distribusi itu merata atau tidak.

Berikut adalah fungsi sederhana untuk memeriksa keseragaman distribusi

nilai 1...K dalam v. Kita menerima hasilnya, jika untuk semua frekuensi

$$|||f_i - 1/K||| < \delta n \quad \text{atau} \quad |f_i - 1/K| < \delta n.$$

Metode tersebut merupakan metode statistik untuk menguji keseragaman distribusi. Distribusi dianggap seragam jika frekuensi setiap nilai dalam v mendekati frekuensi ideal  $1/K$ , dengan deviasi yang tidak melebihi batas toleransi.

```
>function checkrandom (v, delta=1) ...
  K=max(v); n=cols(v);
  fr=getfrequencies(v,1:K);
  return max(fr/n-1/K)<delta/sqrt(n);
endfunction
Memang fungsinya menolak distribusi seragam.
```

```
>checkrandom(wrongdice(1,1000000))
```

0

Dan ia menerima generator acak bawaan.

Manual:

- Asumsi dadu, maka peluang setiap sisi =  $1/6$
- Dalam 1 juta lemparan maka

$$1000000 \times 16 \approx 166667$$

- Frekuensi setiap sisi fr. Proporsi tiap sisi =  $fr/n$
- Misalkan frekuensi munculnya angka adalah

$[160000, 170000, 180000, 150000, 170000, 170000]$   $[160000, 170000, 180000, 150000, 170000, 170000]$

Maka proporsi setiap angka:

$[160000, 170000, 180000, 150000, 170000, 170000]$   $[1000000]$   $[160000, 170000, 180000, 150000, 170000, 170000]$

[0.16,0.17,0.18,0.15,0.17,0.17][0.16,0.17,0.18,0.15,0.17,0.17]

- Deviasi maksimum  $f_n/n - 1/K$   
 $1K=16=0.1667$   $1K=16=0.1667$

$$([0.16,0.17,0.18,0.15,0.17,0.17]-0.1667)([0.16,0.17,0.18,0.15,0.17,0.17]-0.1667)$$

$$\max(-0.0067,0.0033,0.0133,-0.0167,0.0033,0.0033)=0.0133 \max(-0.0067,0.0033,0.0133,-0.0167,0.0033,0.0033)=0.0133$$

- Bandingkan dengan batas toleransi.  
Batas=deltan-- $\sqrt{11000000}$ ----- $\sqrt{11000}=0.001$  Batas=deltan=11000000=11000=0.001

$$0.0133 > 0.001 \quad 0.0133 > 0.001$$

Hasil 0 di sini mengindikasikan bahwa fungsi checkrandom telah menentukan bahwa distribusi tidak seragam.

>checkrandom(intrandom(1,1000000,6))

1  
checkrandom mengembalikan 1 atau true yang berarti bahwa distribusi dari 1 juta bilangan acak rentang 1 sampai 6 dianggap cukup seragam dalam batas toleransi yang ditetapkan.

Kita dapat menghitung distribusi binomial. Pertama ada binomials() , yang mengembalikan probabilitas i atau kurang hit dari n percobaan.

Misal kita akan menghitung probabilitas dari distribusi binomial di mana terdapat 1000 percobaan (misalnya, 1000 kali pelemparan koin), dengan probabilitas sukses pada setiap percobaan sebesar 0.4, dan kita

ingin mengetahui probabilitas mendapatkan tepat 410 sukses.

Secara matematis, ini dihitung dengan rumus:

$$P(X \leq 410) = (1000 \cdot 410) \cdot (0.4)^{410} \cdot (0.6)^{1000-410}$$

>bindis(410,1000,0.4)

$$0.751401349654$$

>bindis(4,10,0.6)

$$0.1662386176$$

Manual:

Secara matematis, ini dihitung dengan rumus:

$$P(X \leq 4) = (104) \cdot (0.6)^4 \cdot (0.4)^{10-4}$$

- Untuk  $k = 0$

$$P(X=0) = (100) \cdot (0.6)^0 \cdot (0.4)^{10} \approx 0.00010$$

- Untuk  $k = 1$

$$P(X=1) = (101) \cdot (0.6)^1 \cdot (0.4)^9 \approx 0.00157$$

- Untuk  $k = 2$

$$P(X=2) = (102) \cdot (0.6)^2 \cdot (0.4)^8 \approx 0.01061$$

- Untuk  $k = 3$

$$P(X=3) = (103) \cdot (0.6)^3 \cdot (0.4)^7 \approx 0.04246$$

- Untuk  $k = 4$

$$P(X=4) = (104) \cdot (0.6)^4 \cdot (0.4)^6 \approx 0.11147$$

Maka,

$$P(X \leq 4) = P(X=0) + P(X=1) + P(X=2) + P(X=3) + P(X=4)$$

$$P(X \leq 4) = 0.00010 + 0.00157 + 0.01061 + 0.04246 + 0.11147$$

$$P(X \leq 4) \approx 0.1662$$

Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p. Tingkat defaultnya adalah alfa.

Arti dari interval ini adalah jika p berada di luar interval, hasil pengamatan 410 dalam 1000 jarang terjadi.

```
>clopperpearson(410,1000)
```

```
[0.37932, 0.441212]
```

Perintah berikut adalah cara langsung untuk mendapatkan hasil di atas.

Namun untuk n yang besar, penjumlahan langsungnya tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

```
0.751401349655
```

Omong-omong, invbinsum() menghitung kebalikan dari binomialsum().

```
>invbindis(0.75,1000,0.4)
```

```
409.932733047
```

Di Bridge, kami mengasumsikan 5 kartu beredar (dari 52) di dua tangan (26 kartu). Mari kita hitung probabilitas distribusi yang lebih buruk dari 3:2 (misalnya 0:5, 1:4, 4:1, atau 5:0).

```
>2*hypergeomsum(1,5,13,26)
```

```
0.321739130435
```

Ada juga simulasi distribusi multinomial.

```
>randmultinomial(10,1000,[0.4,0.1,0.5])
```

381	100	519
376	91	533
417	80	503
440	94	466
406	112	482
408	94	498
395	107	498
399	96	505
428	87	485
400	99	501

## Merencanakan Data/ Plot Data

Untuk memetakan data, kami mencoba hasil pemilu Jerman sejak tahun 1990, diukur dalam jumlah kursi.

```
>BW := [ ...
```

```
1990,662,319,239,79,8,17; ...
1994,672,294,252,47,49,30; ...
1998,669,245,298,43,47,36; ...
2002,603,248,251,47,55,2; ...
2005,614,226,222,61,51,54; ...
2009,622,239,146,93,68,76; ...
2013,631,311,193,0,63,64];
```

Untuk beberapa bagian, kami menggunakan rangkaian nama.

```
>P := ["CDU/CSU", "SPD", "FDP", "Gr", "Li"];
```

Mari kita cetak persentasenya dengan baik.

Pertama kita mengekstrak kolom yang diperlukan. Kolom 3 sampai 7 adalah kursi masing-masing partai, dan kolom 2 adalah jumlah kursi seluruhnya. Kolom 1 adalah tahun pemilihan.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Kemudian statistiknya kita cetak dalam bentuk tabel. Kami menggunakan nama sebagai header kolom, dan tahun sebagai header untuk baris.

Lebar

default untuk kolom adalah wc=10, tetapi kami lebih memilih keluaran

yang lebih padat. Kolom akan diperluas untuk label kolom, jika perlu.

```
>writetable(BT*100,wc=6,dc=0,>fixed,labc=P,labr=YT)
```

	CDU/CSU	SPD	FDP	Gr	Li
1990	48	36	12	1	3
1994	44	38	7	7	4
1998	37	45	6	7	5
2002	41	42	8	9	0
2005	37	36	10	8	9
2009	38	23	15	11	12
2013	49	31	0	10	10

Perkalian matriks berikut ini menjumlahkan persentase dua partai besar

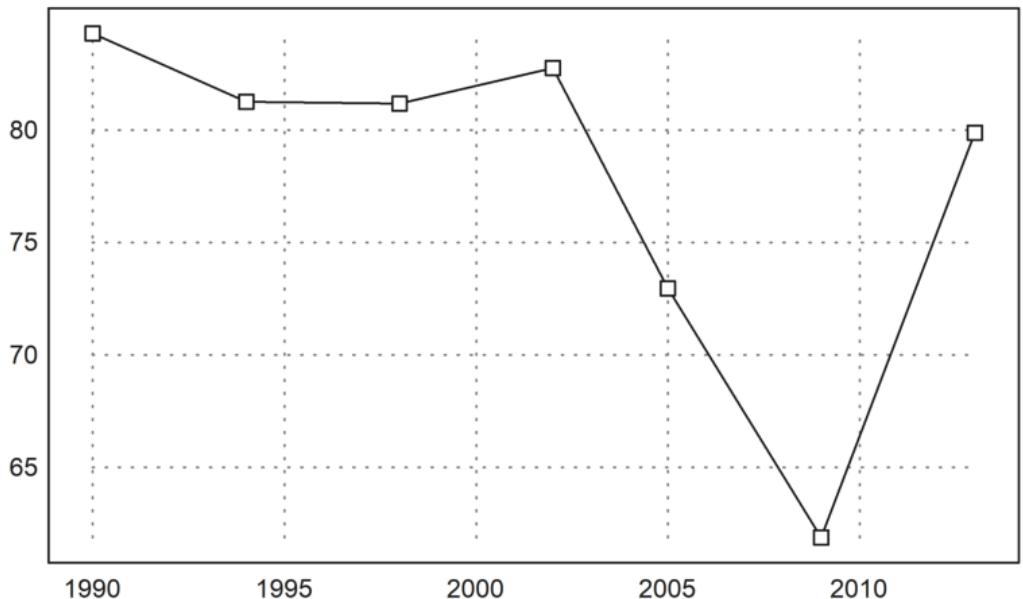
yang menunjukkan bahwa partai-partai kecil berhasil memperoleh suara di parlemen hingga tahun 2009.

```
>BT1:=(BT.[1;1;0;0;0])*100
```

```
[84.29, 81.25, 81.1659, 82.7529, 72.9642, 61.8971, 79.8732]
```

Ada juga plot statistik sederhana. Kami menggunakan untuk menampilkan garis dan titik secara bersamaan. Alternatifnya adalah memanggil plot2d dua kali dengan >add.

```
>statplot(YT,BT1,"b"):
```

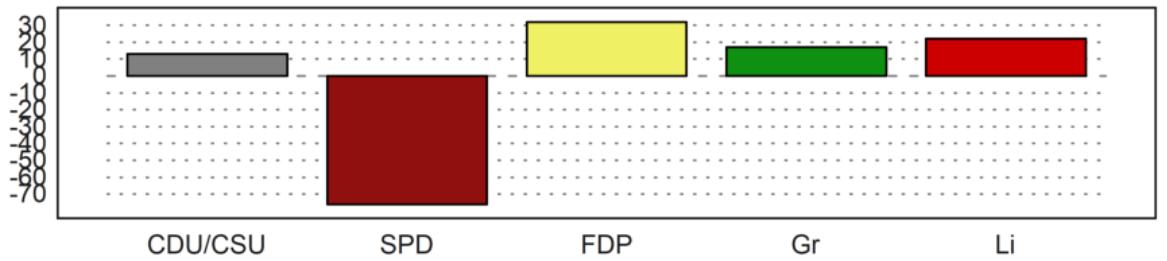
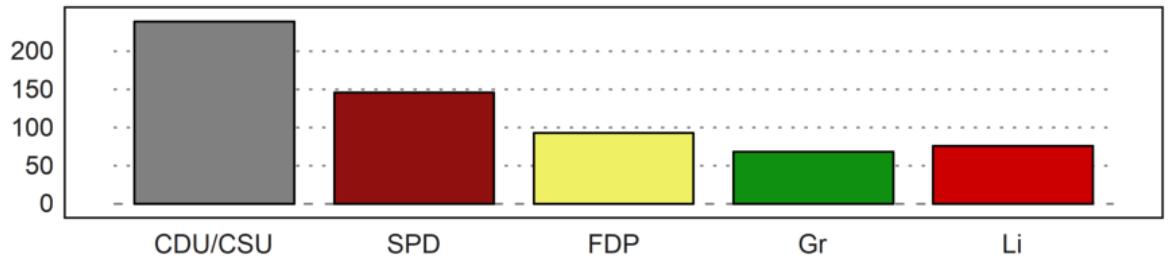


Tentukan beberapa warna untuk setiap pesta.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

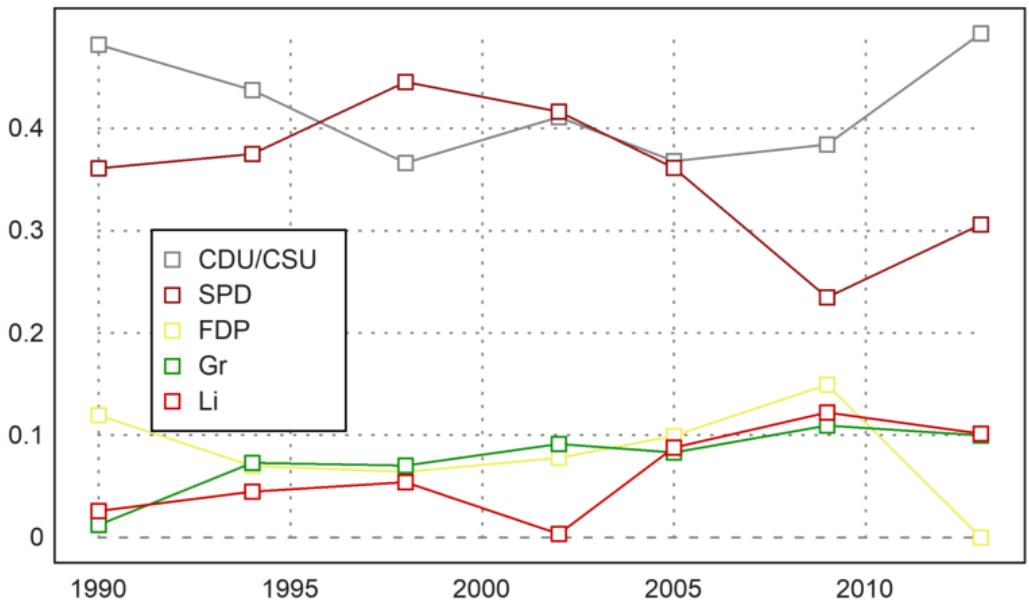
Sekarang kita bisa memplot hasil pemilu 2009 dan perubahannya menjadi satu plot dengan menggunakan gambar. Kita dapat menambahkan vektor kolom ke setiap plot.

```
>figure(2,1); ...
figure(1); columnsplot(BW[6,3:7],P,color=CP); ...
figure(2); columnsplot(BW[6,3:7]-BW[5,3:7],P,color=CP); ...
figure(0):
```



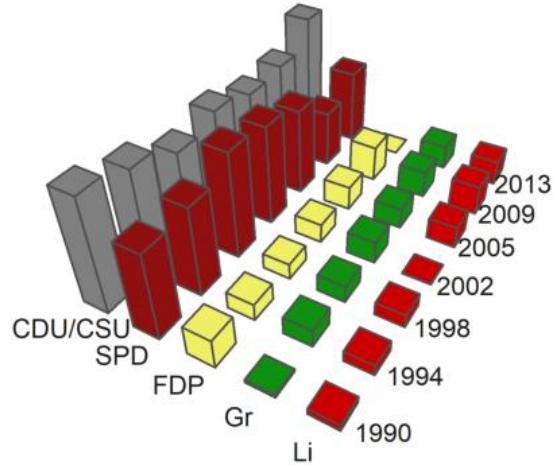
Plot data menggabungkan deretan data statistik dalam satu plot.

```
>J:=BW[,1]'; DP:=BW[,3:7]'; ...
dataplot(YT,BT',color=CP); ...
labelbox(P,colors=CP,styles="[]",>points,w=0.2,x=0.3,y=0.4):
```



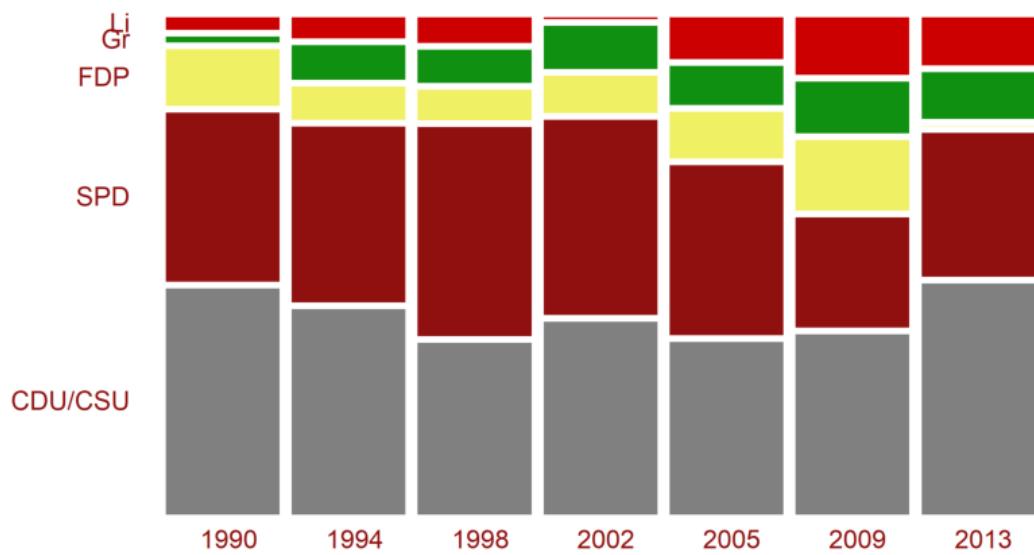
Plot kolom 3D memperlihatkan baris data statistik dalam bentuk kolom. Kami memberikan label untuk baris dan kolom. sudut adalah sudut pandang.

```
>columnspplot3d(BT,scols=P,srows=YT, ...
angle=30°,ccols=CP):
```



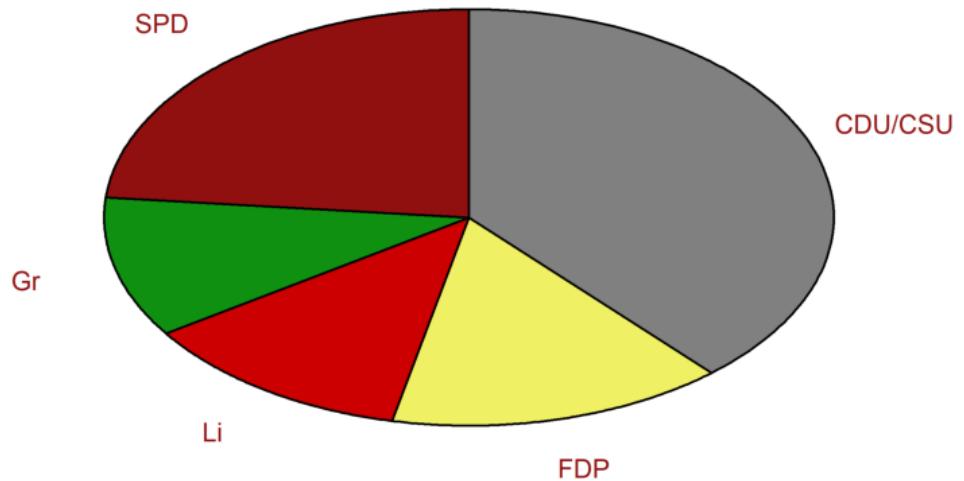
Representasi lainnya adalah plot mosaik. Perhatikan bahwa kolom plot mewakili kolom matriks di sini. Karena panjang label CDU/CSU, kami mengambil jendela yang lebih kecil dari biasanya.

```
>shrinkwindow(>smaller); ...
mosaicplot(BT',srows=YT,scols=P,color=CP,style="#" );
shrinkwindow():
```



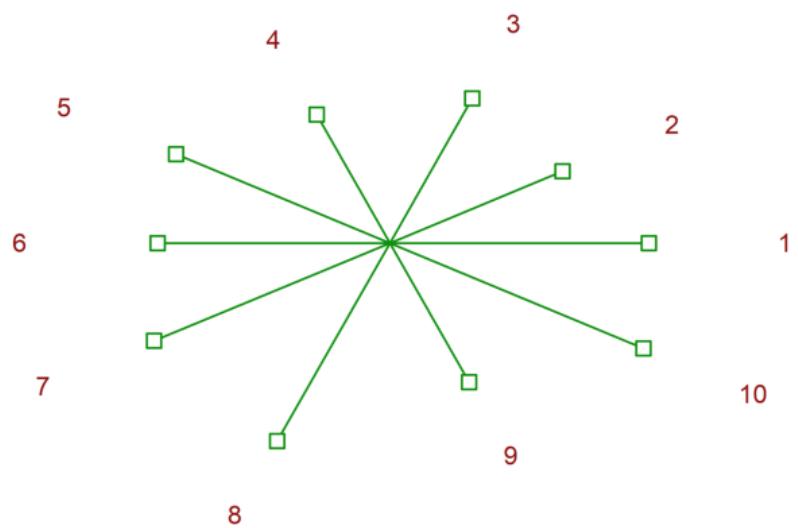
Kita juga bisa membuat diagram lingkaran. Karena hitam dan kuning membentuk koalisi, kami menyusun ulang elemen-elemennya.

```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```



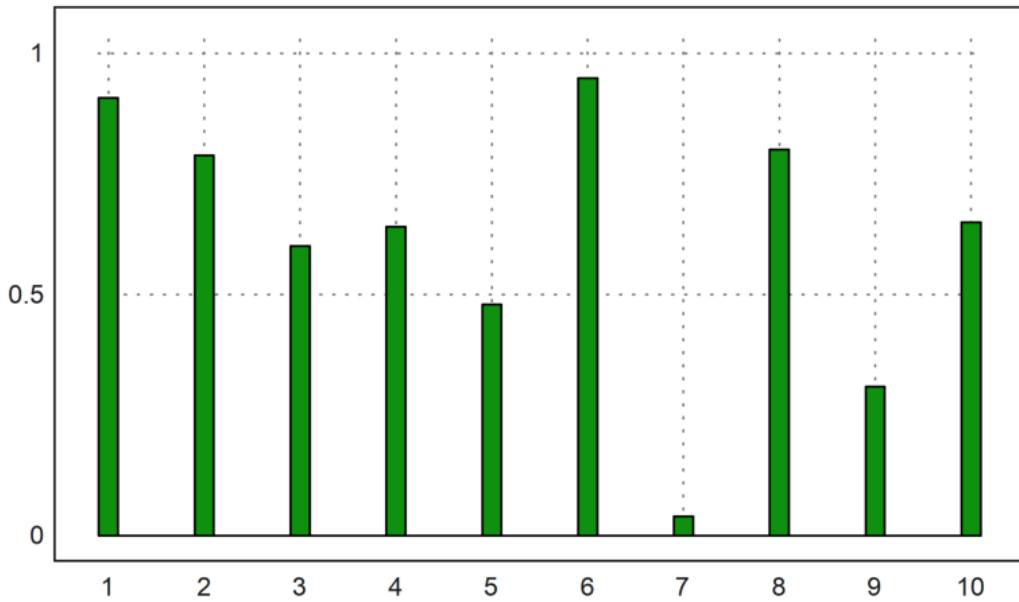
Ini adalah jenis plot lainnya.

```
>starplot(normal(1,10)+4,lab=1:10,>rays):
```



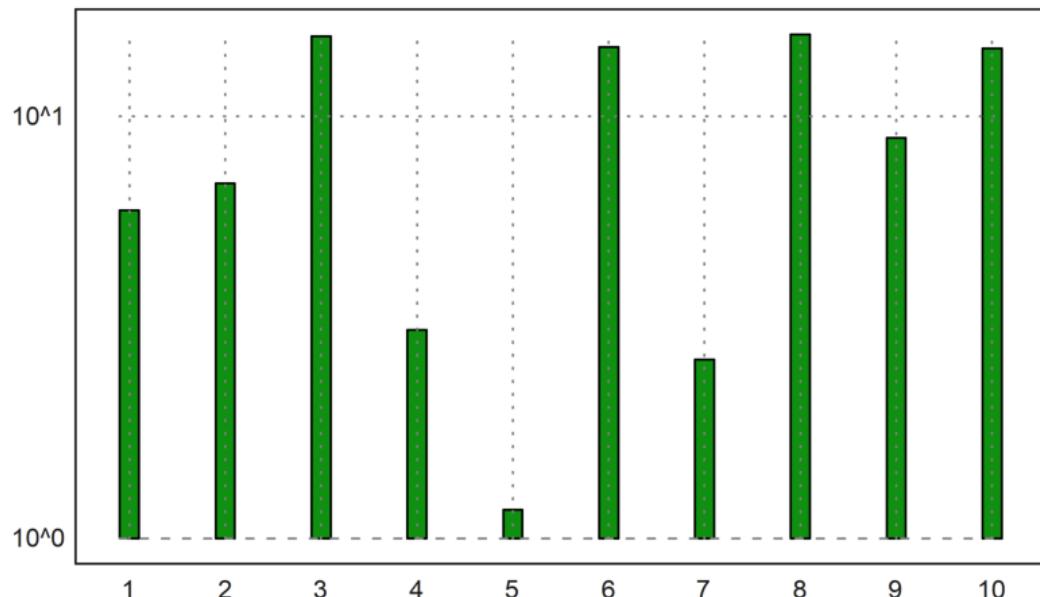
Beberapa plot di plot2d bagus untuk statika. Berikut adalah plot impuls dari data acak, terdistribusi secara seragam di [0,1].

```
>plot2d(makeimpulse(1:10,random(1,10)),>bar):
```



Namun untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

```
>logimpulseplot(1:10,-log(random(1,10))*10):
```

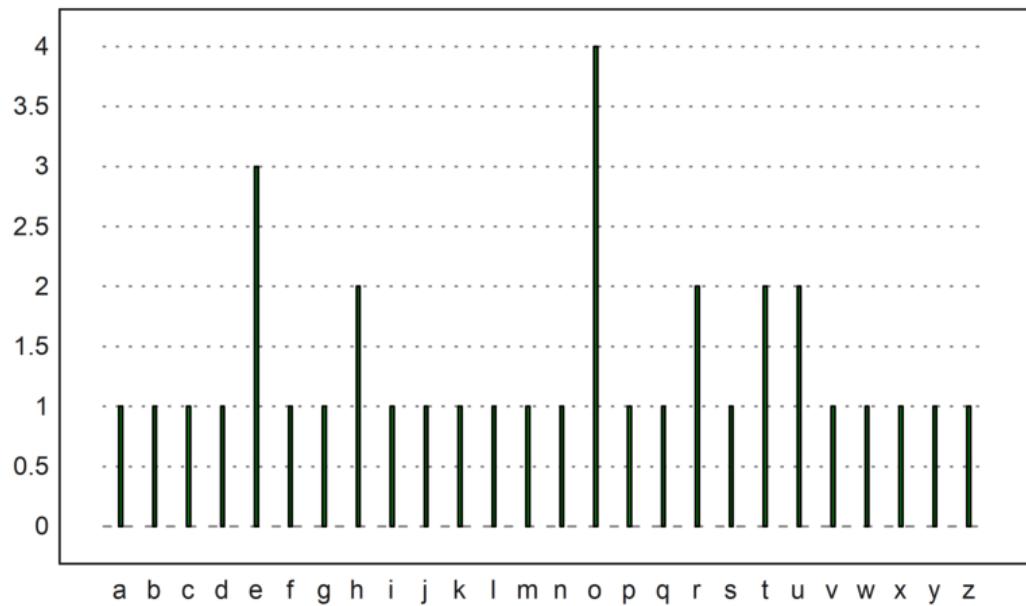


Fungsi Columnplot() lebih mudah digunakan, karena hanya memerlukan vektor nilai. Selain itu, ia dapat mengatur labelnya ke apa pun yang kita inginkan, kami telah mendemonstrasikannya di tutorial ini.

Ini adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan membuat statistik.

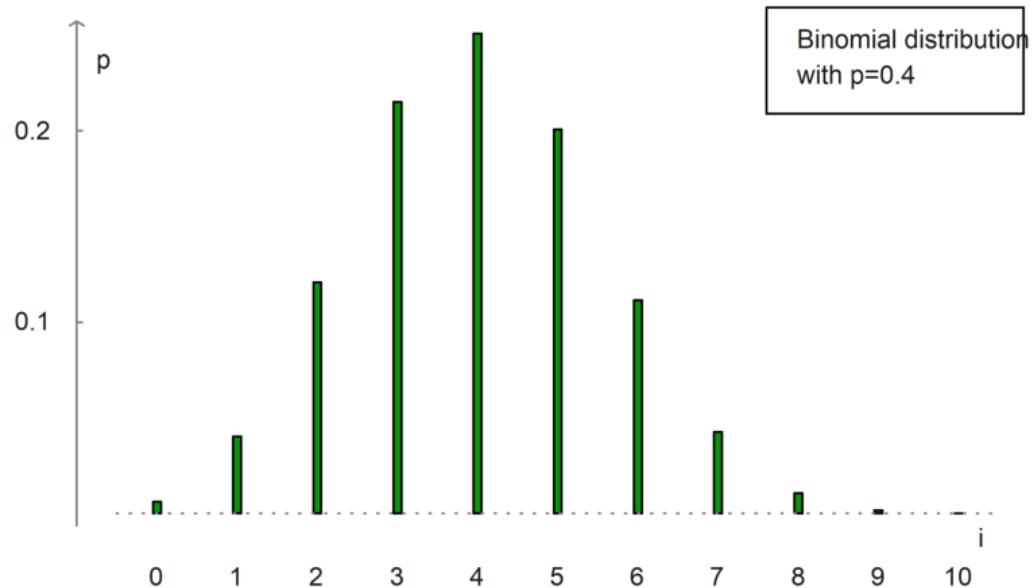
```
>v=strtochar("the quick brown fox jumps over the lazy dog"); ...
w=ascii("a"):ascii("z"); x=getmultiplicities(w,v); ...
cw=[]; for k=w; cw=cw|char(k); end; ...
```

```
columnsplot(x,lab=cw,width=0.05) :
```



Dimungkinkan juga untuk mengatur sumbu secara manual.

```
>n=10; p=0.4; i=0:n; x=bin(n,i)*p^i*(1-p)^(n-i); ...
columnsplot(x,lab=i,width=0.05,<frame,<grid); ...
yaxis(0,0:0.1:1,style="->",>left); xaxis(0,style="."); ...
label("p",0,0.25), label("i",11,0); ...
textbox(["Binomial distribution","with p=0.4"]);
```



Berikut ini cara memplot frekuensi bilangan dalam suatu vektor.

Kami membuat vektor bilangan acak bilangan bulat 1 hingga 6.

```
>v:=intrandom(1,10,10)
```

```
[8, 5, 8, 8, 6, 8, 8, 3, 5, 5]
```

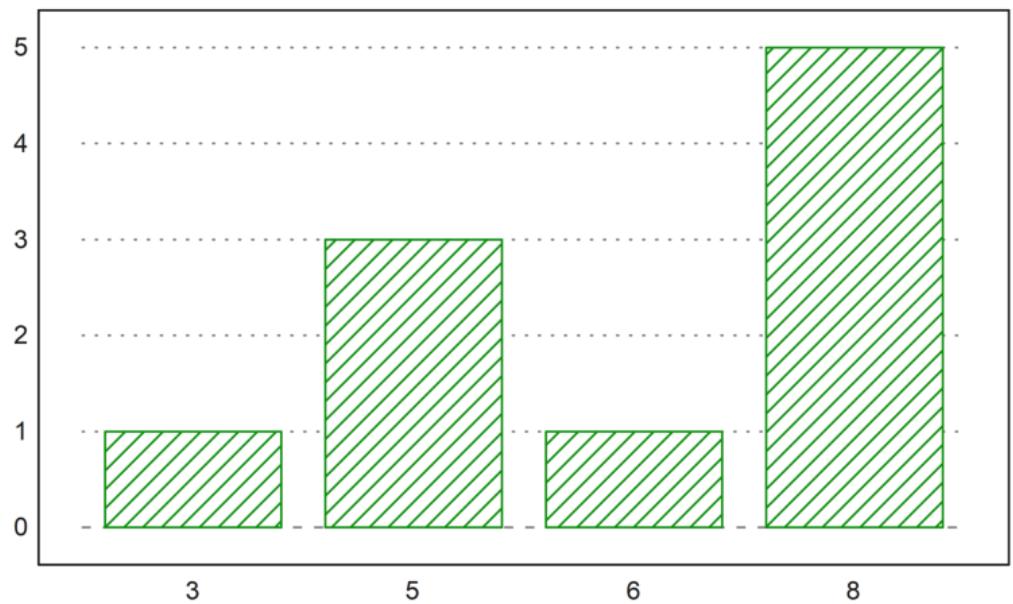
Kemudian ekstrak nomor unik di v.

```
>vu:=unique(v)
```

```
[3, 5, 6, 8]
```

Dan plot frekuensi dalam plot kolom.

```
>columnsplot(getmultiplicities(vu,v),lab=vu,style="/"):
```



Kami ingin mendemonstrasikan fungsi distribusi nilai empiris.

```
>x=normal(1,20);
```

Fungsi empdist(x,vs) memerlukan array nilai yang diurutkan. Jadi kita harus mengurutkan x sebelum kita dapat menggunakanya.

```
>xs=sort(x);
```

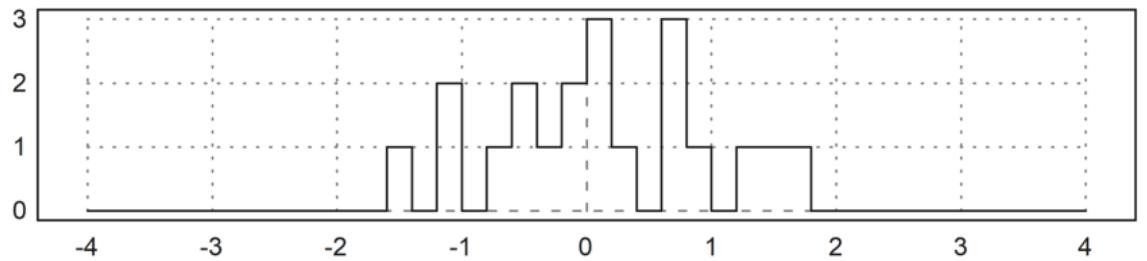
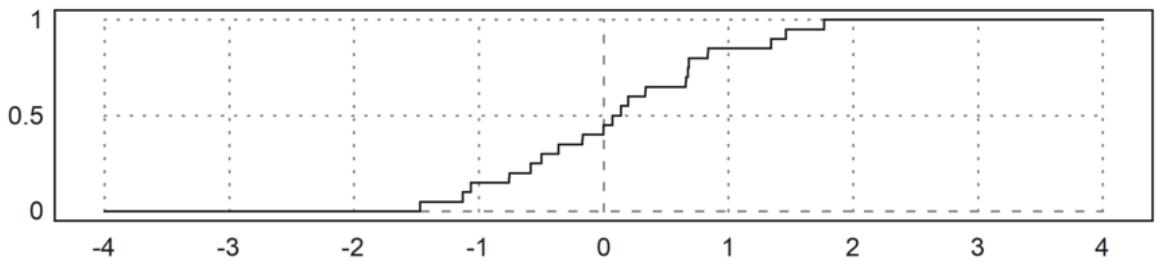
Kemudian kita plot distribusi empiris dan beberapa batang kepadatan ke dalam satu plot. Alih-alih plot batang untuk distribusi kali ini kami menggunakan plot gigi gergaji.

```
>figure(2,1); ...
```

```
figure(1); plot2d("empdist",-4,4;xs); ...
```

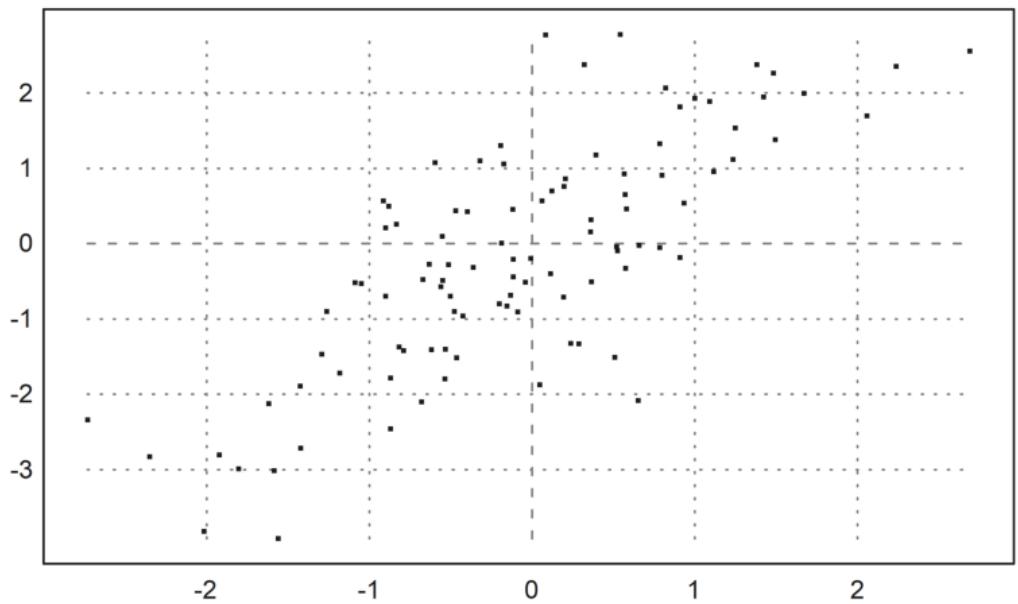
```
figure(2); plot2d(histo(x,v=-4:0.2:4,<bar>)); ...
```

```
figure(0):
```



Plot sebar mudah dilakukan di Euler dengan plot titik biasa. Grafik berikut menunjukkan bahwa  $X$  dan  $X+Y$  jelas berkorelasi positif.

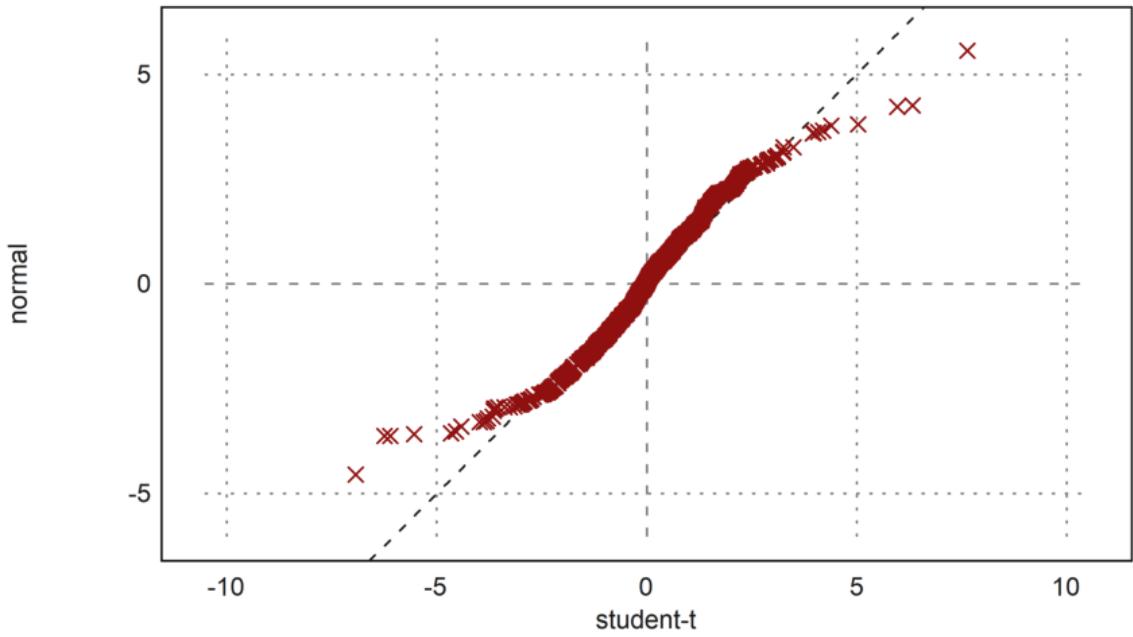
```
>x=normal(1,100); plot2d(x,x+rotrigh(x),>points,style=".."):
```



Seringkali kita ingin membandingkan dua sampel dengan distribusi yang berbeda. Hal ini dapat dilakukan dengan plot kuantil-kuantil.

Untuk pengujiannya, kami mencoba distribusi student-t dan distribusi eksponensial.

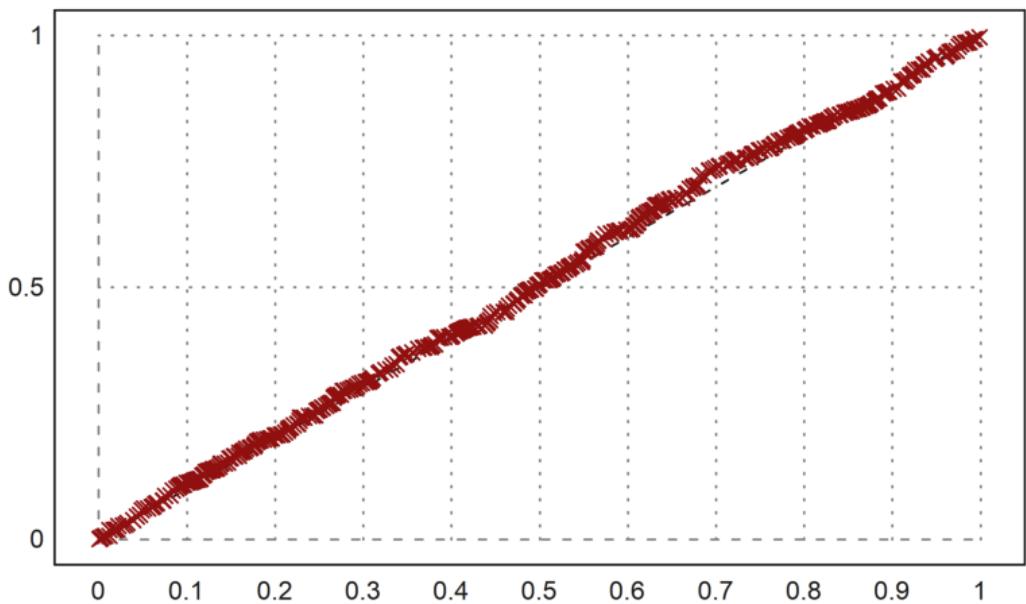
```
>x=randt(1,1000,5); y=randnormal(1,1000,mean(x),dev(x)); ...
plot2d("x",r=6,style="--",yl="normal",xl="student-t",>vertical); ...
plot2d(sort(x),sort(y),>points,color=red,style="x",>add):
```



Plot tersebut dengan jelas menunjukkan bahwa nilai terdistribusi normal cenderung lebih kecil di ujung ekstrim.

Jika kita mempunyai dua distribusi yang ukurannya berbeda, kita dapat memperluas distribusi yang lebih kecil atau mengecilkan distribusi yang lebih besar. Fungsi berikut ini baik untuk keduanya. Dibutuhkan nilai median dengan persentase antara 0 dan 1.

```
>function medianexpand (x,n) := median(x,p=linspace(0,1,n-1));
Mari kita bandingkan dua distribusi yang sama.
>x=random(1000); y=random(400); ...
plot2d("x",0,1,style="--");
plot2d(sort(medianexpand(x,400)),sort(y),>points,color=red,style="x",>add):
```



## Regresi dan Korelasi

Regresi linier dapat dilakukan dengan fungsi `polyfit()` atau berbagai fungsi `fit`.

Sebagai permulaan kita menemukan garis regresi untuk data univariat dengan `polyfit(x,y,1)`.

```
>x=1:10; y=[2,3,1,5,6,3,7,8,9,8]; writetable(x' | y', labc=["x", "y"])
    x      y
    1      2
    2      3
    3      1
    4      5
    5      6
    6      3
    7      7
    8      8
    9      9
    10     8
```

Kami ingin membandingkan kecocokan yang tidak berbobot dan berbobot. Pertama koefisien kecocokan linier.

```
>p=polyfit(x,y,1)
[0.733333,  0.812121]
Regresi linear dapat ditulis dalam bentuk:
```

$$y = mx + b = mx + b$$

dengan

$$m = \frac{\sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$$

$$b = \frac{\sum y - m(\sum x)}{n}$$

Kita hitung:

$n=10, x=[1,2,3,4,5,6,7,8,9,10], y=[2,3,1,5,6,3,7,8,9,8]$

$$\sum x = 55$$

$$\sum y = 52$$

$$\sum xy = 353$$

$$\sum x^2 = 385$$

Maka:

$$m = \frac{10(353) - (55)(52)}{10(385) - (55)^2} = \frac{3530 - 2860}{3850 - 3025} = 0.812121$$

$$b = \frac{52 - 0.812121(55)}{10} = 0.733333$$

Jadi,  $b, m = 0.733333, 0.812121$

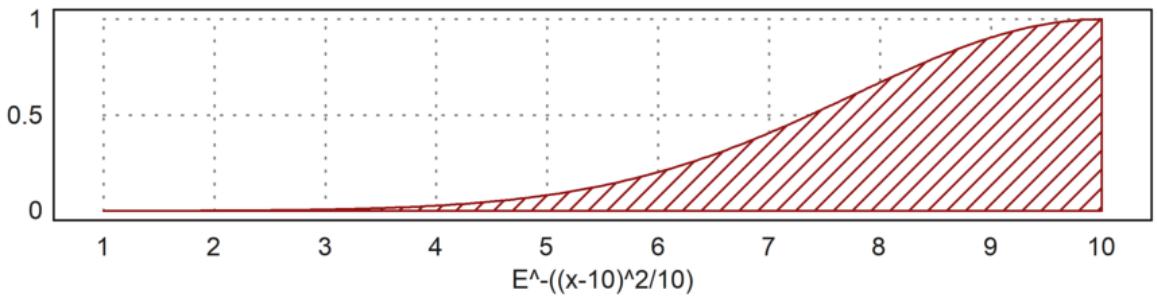
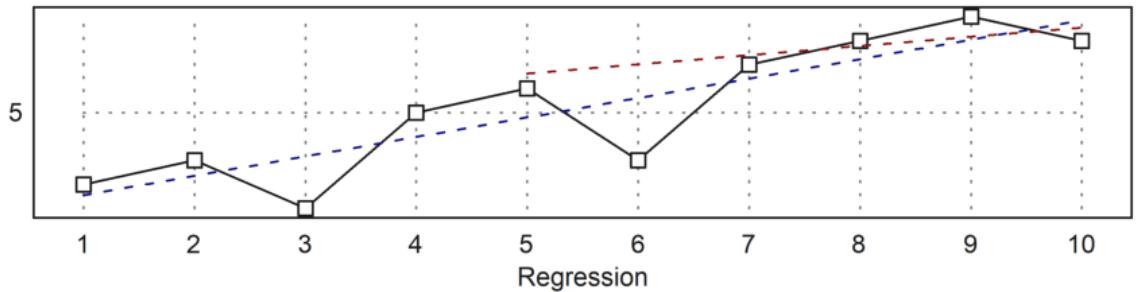
Sekarang koefisien dengan bobot yang menekankan nilai terakhir.

```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

$$[4.71566, 0.38319]$$

Kami memasukkan semuanya ke dalam satu plot untuk titik dan garis regresi, dan untuk bobot yang digunakan.

```
>figure(2,1); ...
figure(1); statplot(x,y,"b",xl="Regression"); ...
plot2d("evalpoly(x,p)",>add,color=blue,style="--"); ...
plot2d("evalpoly(x,pw)",5,10,>add,color=red,style="--"); ...
figure(2); plot2d(w,1,10,>filled,style="/",fillcolor=red,xl=w); ...
figure(0):
```



Contoh lain kita membaca survei siswa, usia mereka, usia orang tua mereka dan jumlah saudara kandung dari sebuah file.

Tabel ini berisi "m" dan "f" di kolom kedua. Kami menggunakan variabel

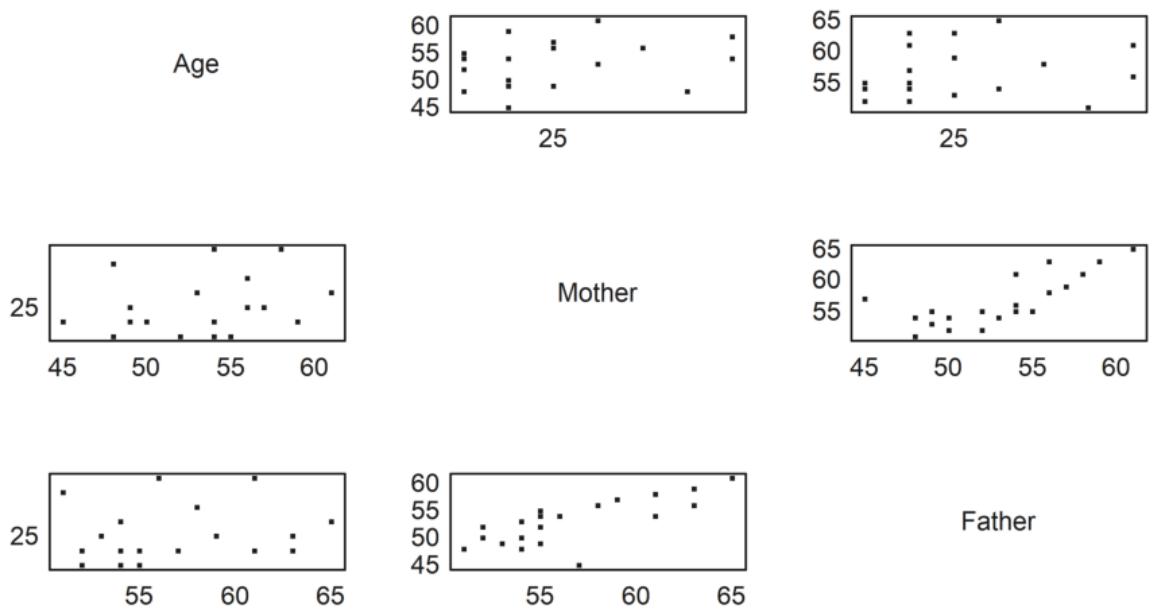
tok2 untuk mengatur terjemahan yang tepat alih-alih membiarkan readtable() mengumpulkan terjemahannya.

```
>{MS,hd}:=readtable("table1.dat",tok2:=[ "m", "f" ]); ...  
writetable(MS,labc=hd,tok2:=[ "m", "f" ]);
```

Person	Sex	Age	Mother	Father	Siblings
1	m	29	58	61	1
2	f	26	53	54	2
3	m	24	49	55	1
4	f	25	56	63	3
5	f	25	49	53	0
6	f	23	55	55	2
7	m	23	48	54	2
8	m	27	56	58	1
9	m	25	57	59	1
10	m	24	50	54	1
11	f	26	61	65	1
12	m	24	50	52	1
13	m	29	54	56	1
14	m	28	48	51	2
15	f	23	52	52	1
16	m	24	45	57	1
17	f	24	59	63	0
18	f	23	52	55	1
19	m	24	54	61	2
20	f	23	54	55	1

Bagaimana usia bergantung satu sama lain? Kesan pertama muncul dari plot sebar berpasangan.

```
>scatterplots(tablecol(MS,3:5),hd[3:5]):
```



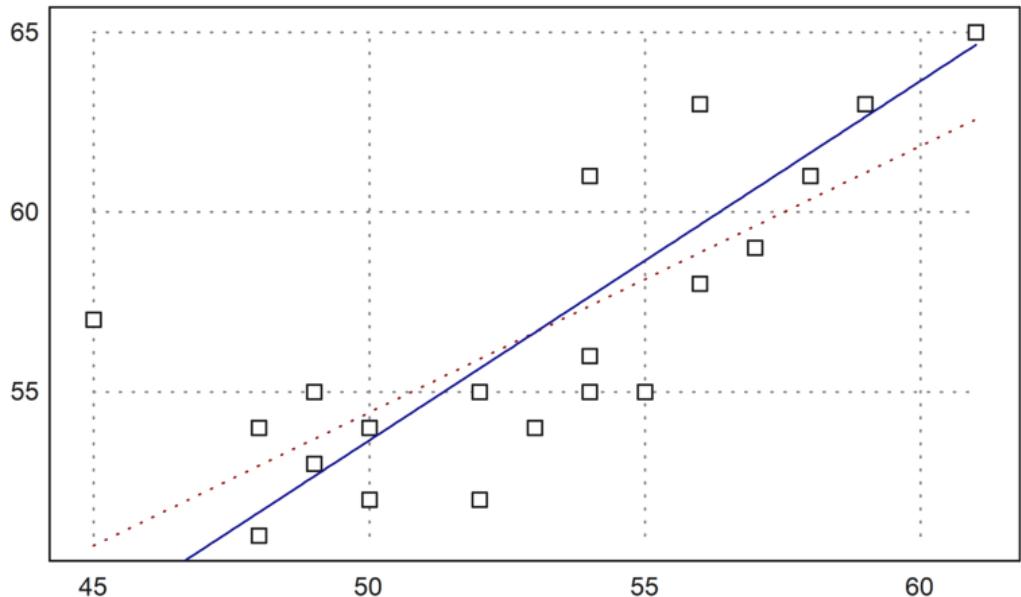
Jelas terlihat bahwa usia ayah dan ibu saling bergantung satu sama lain. Mari kita tentukan dan plot garis regresinya.

```
>cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1)
[17.3789, 0.740964]
```

Ini jelas merupakan model yang salah. Garis regresinya adalah  $s=17+0,74t$ , dengan  $t$  adalah umur ibu dan  $s$  adalah umur ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tapi tidak terlalu banyak.

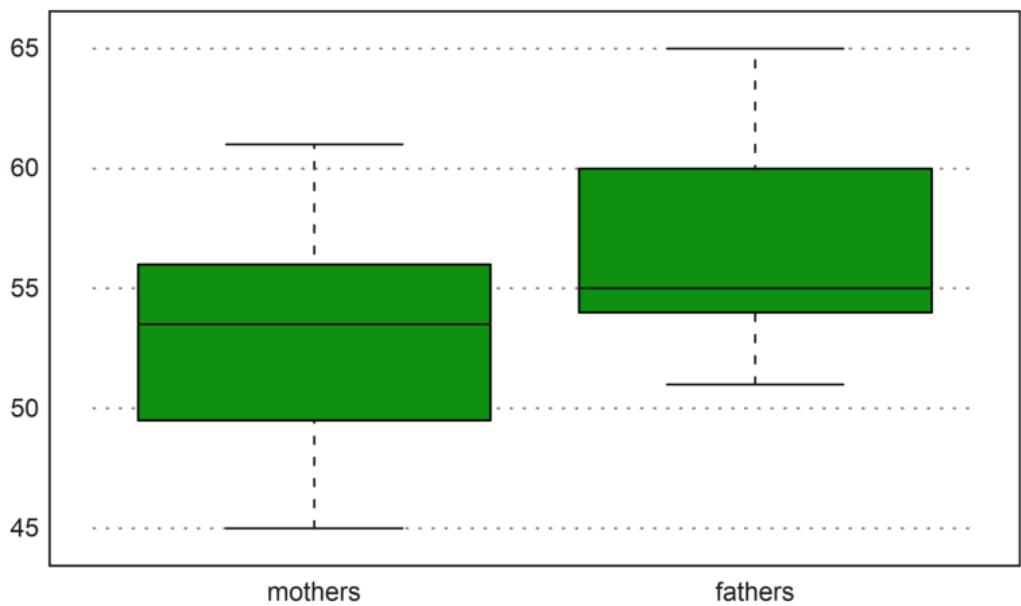
Sebaliknya, kami mencurigai fungsi seperti  $s=a+t$ . Maka  $a$  adalah mean dari  $s-t$ . Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

```
>da:=mean(cs[2]-cs[1])
3.65
Mari kita plot ini menjadi satu plot sebar.
>plot2d(cs[1],cs[2],>points); ...
plot2d("evalpoly(x,ps)",color=red,style=".",>add); ...
plot2d("x+da",color=blue,>add):
```



Berikut adalah plot kotak dari dua zaman tersebut. Ini hanya menunjukkan, bahwa usianya berbeda-beda.

```
>boxplot(cs, "mothers", "fathers"):
```



Menariknya, perbedaan median tidak sebesar perbedaan mean.

```
>median(cs[2]) - median(cs[1])
```

1.5

Koefisien korelasi menunjukkan korelasi positif.

```
>Koefisien korelasi menunjukkan korelasi positif.correl(cs[1],cs[2])
```

Variable Koefisien not found!

Error in:

Koefisien korelasi menunjukkan korelasi positif.correl(cs[1],c ...

^

Korelasi pangkat merupakan ukuran keteraturan yang sama pada kedua vektor. Hal ini juga cukup positif.

```
>rankcorrel(cs[1],cs[2])
```

0.758925292358

## Membuat Fungsi baru

Tentu saja, bahasa EMT dapat digunakan untuk memprogram fungsi-fungsi baru. Misalnya, kita mendefinisikan fungsi skewness.

$$sk(x) = \frac{n}{\sqrt{\sum_i (x_i - m)^3}} (\sum_i (x_i - m)^2)^{3/2}$$

$m$  adalah rata-rata dari  $x$ .

```
>function skew (x:vector) ...
m=mean(x);
return sqrt(cols(x))*sum((x-m)^3)/(sum((x-m)^2))^(3/2);
endfunction
```

Seperti yang Anda lihat, kita dapat dengan mudah menggunakan bahasa matriks untuk mendapatkan implementasi yang sangat singkat dan efisien. Mari kita coba fungsi ini.

```
>data=normal(20); skew(normal(10))
```

-0.198710316203

Berikut adalah fungsi lainnya, yang disebut koefisien skewness Pearson.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)
>skew1(data)
```

-0.0801873249135

## Simulasi Monte Carlo

Kita simulasikan variabel acak berdistribusi normal 1000-5 sebanyak sejuta kali. Untuk ini, kita gunakan fungsi normal(m,n), yang menghasilkan matriks nilai berdistribusi 0-1, atau normal(n) yang secara default bernilai  $m=1$ .

```
>n=1000000; x=normal(n)
```

```
[-1.9943, 1.1795, 1.18591, -0.441316, -0.488192, -0.675366,
-1.58021, 0.962882, -1.66963, -0.819503, -0.929235, 0.539573,
-0.606864, -0.56878, -0.087391, -0.404174, 0.0788819, 1.04282,
0.876296, -0.877162, 0.689615, 1.66396, 1.10287, 1.81334,
-0.679579, -0.739293, -0.0919214, -0.0516635, 1.68059, -0.8748,
-0.661822, 1.14053, -0.0725198, -0.653134, 0.338853, 0.158539,
-1.33566, -1.21098, 0.252185, 0.750436, -0.335005, 0.79995,
0.237049, 0.10056, -0.909771, -0.896389, 0.807295, 0.420089,
-2.60022, -0.334883, 0.563393, -1.51976, -0.618486, -0.757591,
-1.5655, -1.14804, -1.62101, -0.767098, 1.41355, 0.28399,
1.31699, 0.325665, 0.123923, -0.301708, 0.224152, 1.61033,
-0.928959, 0.112323, 2.06755, -0.415012, 1.227, -0.392912,
-0.82682, -1.37726, 1.09679, -2.83431, 0.310441, 1.333,
-0.774371, 1.51449, 0.0260218, 0.244161, -1.25954, 0.833535,
-1.05672, -3.99147, -1.70256, 0.620494, -0.98936, -0.868208,
0.344441, 0.926404, 0.00735312, -1.07317, 0.26722, 0.469612,
-1.44652, -0.0850004, -1.22639, -0.120585, -0.498201, 0.793201,
-2.71956, -0.832553, -1.50601, -0.473103, -0.857245, -0.397193,
-0.368307, -1.08578, -1.60682, -0.525828, 0.00660302, -2.44655,
0.761081, -0.0405162, -0.440874, -0.733893, 0.275494, 1.14203,
... ]
```

```
>n=1000000; x=normal(n)*5+1000
```

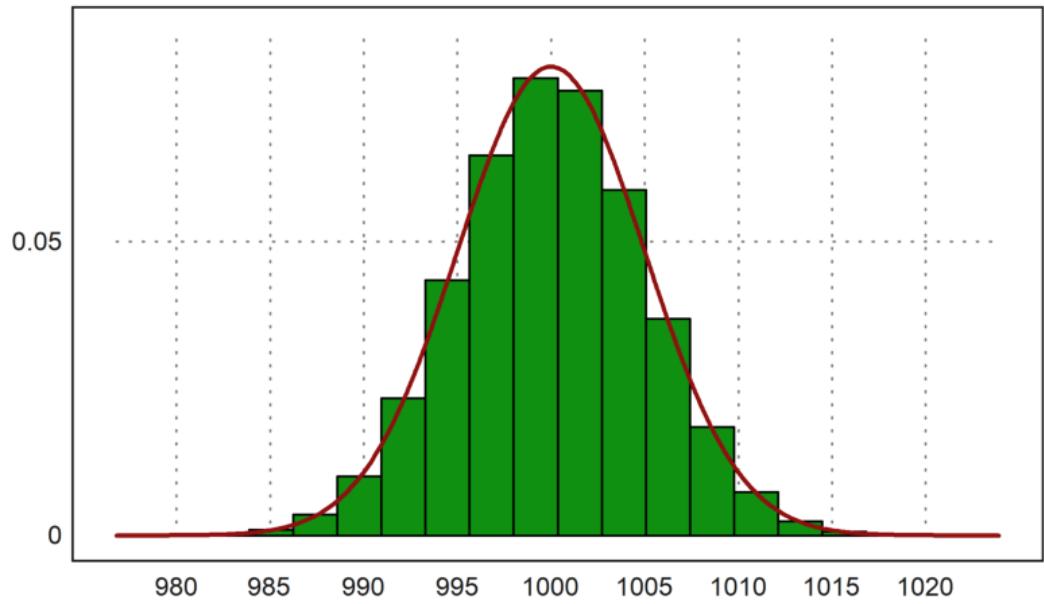
```
[1002.99, 1003.08, 996.596, 990.94, 998.608, 1005.05, 997.111,
1006.2, 1003.62, 988.073, 999.093, 1001.26, 999.921, 994.418,
995.893, 1008.1, 1000.85, 997.523, 997.748, 1000.46, 988.609,
```

```

1003.88, 996.944, 994.016, 1008.61, 997.575, 992.025, 993.416,
1003.18, 1003.39, 1001.73, 1005.57, 995.544, 1004.48, 996.802,
1004.15, 1005.39, 997.653, 1002.83, 1005.25, 998.235, 994.648,
998.584, 1000.88, 998.811, 994.607, 995.101, 995.2, 1002.16,
1009.75, 1000.77, 999.994, 1011.51, 1006.36, 990.598, 1007.2,
1000.18, 999.05, 1002.22, 1000.08, 993.234, 1000.81, 1004.78,
1002.42, 993.949, 1002.01, 995.443, 994.59, 1008.39, 1000.62,
995.42, 1000.23, 998.282, 998.84, 992.97, 1000.81, 993.34,
1003.17, 1001.89, 999.305, 1000.63, 992.78, 989.389, 998.162,
1001.79, 1008.18, 1003.34, 1001.35, 1001.08, 998.659, 998.586,
1000.31, 994.955, 994.737, 988.751, 991.572, 995.534, 999.851,
1004.17, 997.659, 993.513, 996.48, 999.944, 996.108, 998.717,
1002.66, 1001.39, 1003.17, 992.563, 998.983, 994.664, 997.407,
1001.87, 999.54, 993.469, 999.279, 1000.47, 996.123, 1000.8,
1003.22, 993.011, 998.241, 1000.07, 1001.6, 1002.9, 1004.22,
1002.22, 1005.54, 999.447, 1002.94, 1003.54, 999.995, 996.819,
1001.89, 1003.54, 1005.62, 1002.17, 998.897, 1005.83, 1006.59,
...
]
terdapat juga fungsi randnormal(n,m,mean,dev), yang dapat kita
gunakan. Fungsi ini mematuhi skema penamaan "rand..." untuk generator
acak.
```

```

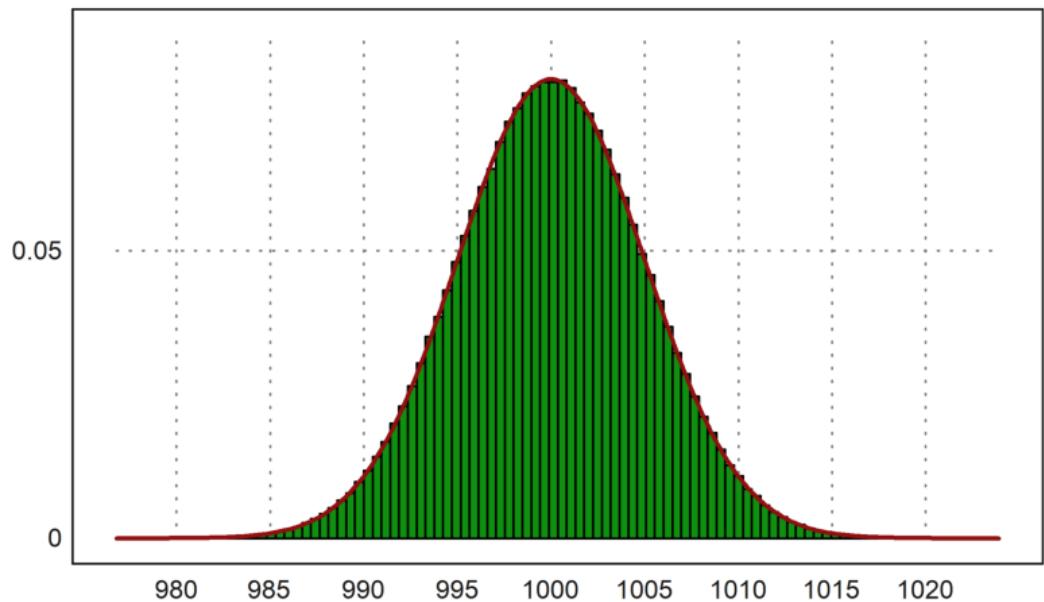
>n=1000000; x=randnormal(1,n,1000,5)
[1001.46, 1002.83, 995.424, 1010.42, 993.922, 995.65, 1002.73,
996.528, 998.885, 999.782, 999.165, 996.189, 1007.19, 1002.5,
1000.04, 1001.11, 995.786, 1000.87, 1007.36, 997.776, 989.569,
1000.06, 999.158, 996.89, 1000.07, 1000.75, 1003.66, 995.875,
1003.36, 992.127, 1003.03, 1006.05, 999.733, 999.09, 1005.97,
1003.15, 998.16, 989.205, 1002.45, 996.641, 1008.69, 1006.67,
992.93, 1003.96, 1002.56, 997.29, 995.598, 995.616, 1000.14,
1002.68, 984.552, 1004.04, 1008.83, 994.196, 991.508, 1003.67,
1000.44, 1001.38, 996.971, 1001.32, 998.596, 995.479, 1002.12,
990.516, 997.716, 1008.15, 1000.37, 1001.07, 996.014, 1001.07,
1003.76, 996.41, 992.867, 998.441, 993.741, 997.669, 1007.58,
1005.19, 997.211, 1000.75, 994.023, 994.398, 998.096, 994.683,
994.84, 1001.44, 1003.19, 987.72, 1003.07, 1008.01, 1009.52,
1007.39, 997.166, 1007.7, 987.807, 1002.73, 1003.18, 1002.86,
1002.75, 1001.91, 995.783, 998.36, 1002.57, 994.433, 1006.56,
1002.42, 997.431, 1005.47, 992.773, 1007.92, 1007.36, 999.482,
1003.88, 998.68, 1012.77, 1008.83, 1002.04, 1003.13, 1009.78,
997.305, 1003.33, 1001.45, 1004.12, 999.9, 1002.25, 996.054,
994.309, 1006.29, 999.89, 1003.39, 995.42, 990.872, 1000.26,
994.472, 997.125, 998.633, 994.745, 1001.02, 996.873, 1001.29,
...
]
10 nilai pertama x adalah
>x[1:10]
[1001.46, 1002.83, 995.424, 1010.42, 993.922, 995.65, 1002.73,
996.528, 998.885, 999.782]
Distribusi dapat kita plot dengan flag >distribution dari plot2d.
>plot2d(x,>distribution); ...
plot2d("qnormal(x,1000,5)",color=red,thickness=2,>add) :
```



kita juga dapat mengatur jumlah interval untuk distribusi menjadi 100.

Kemudian kita akan melihat seberapa dekat kecocokan distribusi yang diamati dan distribusi yang sebenarnya. Bagaimanapun, kita telah menghasilkan satu juta kejadian.

```
>plot2d(x,distribution=100); ...
plot2d("qnormal(x,1000,5)",color=red,thickness=2,>add):
```



kita dapat menghitung nilai rata-rata simulasi dan deviasinya harus sangat dekat dengan nilai yang diharapkan.

```
>mean(x), dev(x)
999.996936044
4.99872526095
```

```
rumus nilai rata rata
```

$$\text{mean} = \frac{\sum x_i}{n}$$

```
> xm = sum(x) / n
```

999.996936044

Rumus simpangan percobaannya (deviasi)

$$\text{deviasi} = \sqrt{\frac{\sum (x - xm)^2}{n}}$$

```
> sqrt(sum((x-xm)^2 / (n-1)))
```

4.99872526095

Perhatikan bahwa  $x - xm$  adalah vektor nilai yang dikoreksi, di mana  $xm$  dikurangi dari semua elemen vektor  $x$ .

Berikut adalah 10 nilai pertama  $x - xm$ .

```
> short(x-xm)[1:10]
```

[1.4664, 2.8367, -4.573, 10.419, -6.075, -4.3474, 2.7294, -3.4693, -1.1123, -0.21472]

Dengan menggunakan bahasa matriks, kita dapat dengan mudah menjawab pertanyaan lainnya. Misalnya, kita ingin menghitung proporsi  $x$  yang melebihi 1015.

Ekspresi  $x >= 1015$  menghasilkan vektor 1 dan 0. Menjumlahkan vektor ini menghasilkan jumlah kali  $x[i] >= 1015$  terjadi.

```
> sum(x >= 1015) / n
```

0.001339

Probabilitas yang diharapkan dari hal ini dapat dihitung dengan fungsi

normaldis(x). sehingga,

$$\text{normaldis}(c, m, s) = P(X \leq c)$$

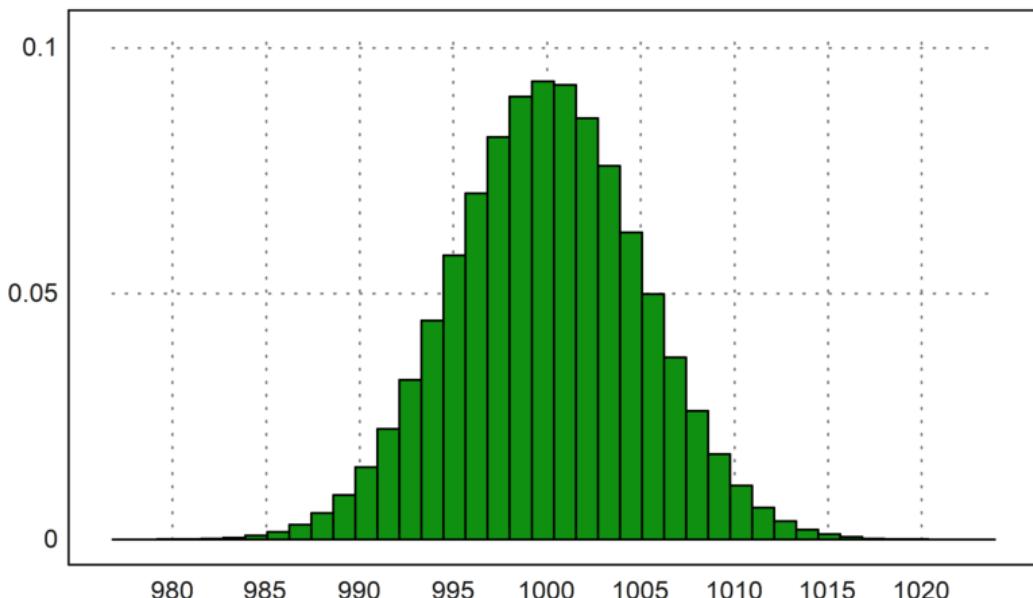
dimana  $X$  terdistribusi secara normal  $m-s$ .

```
> 1 - normaldis(1015, 1000, 5)
```

0.00134989803163

cara kerja >distribution flag dari plot2d adalah menggunakan fungsi histo(x), yang menghasilkan histogram frekuensi nilai dalam x. Fungsi ini mengembalikan batas interval dan jumlah dalam interval ini. Kami menormalkan jumlah untuk mendapatkan frekuensi.

```
> {t, s} = histo(x, 40); plot2d(t, s/n, >bar):
```



Fungsi `histo()` juga dapat menghitung frekuensi dalam interval yang diberikan.

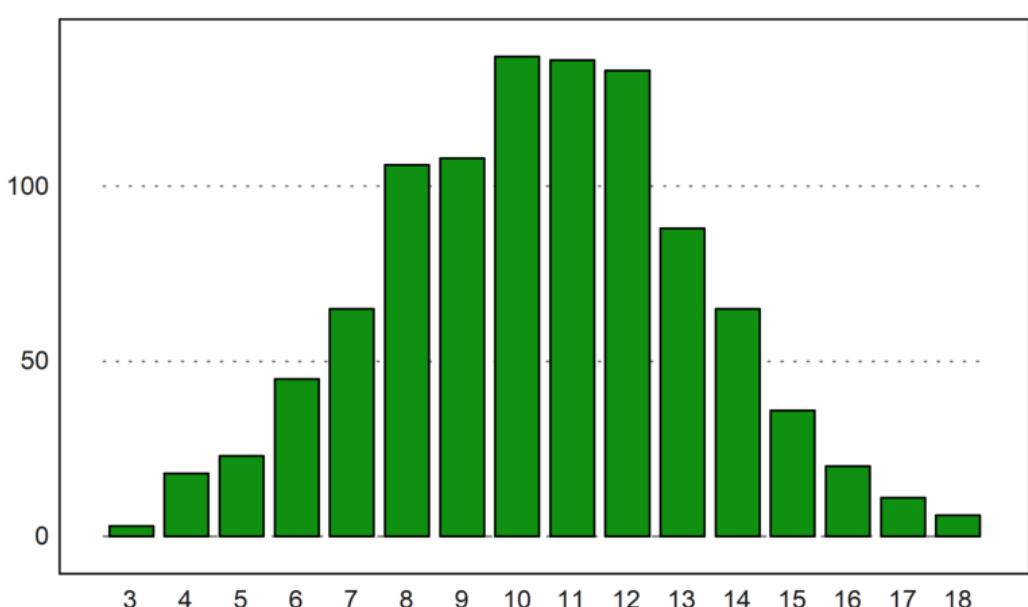
```
>{t,s}=histo(x,v=[950,980,990,1010,1020,1050]); t, s,
[950, 980, 990, 1010, 1020, 1050]
[25, 22795, 954608, 22545, 27]
hasil tersebut merupakan semua nilai acak yang berada antara 950 dan 1050.
```

menghitung total jumlah nilai dalam `s`, yang sama dengan total jumlah elemen dalam `x`

```
>sum(s)
1000000
kita akan mensimulasikan 1000 kali lemparan 3 dadu, dan menanyakan pembagian jumlahnya.
```

```
>ds:=sum(intrandom(1000,3,6)'); fs=getmultiplicities(3:18,ds)
[3, 18, 23, 45, 65, 106, 108, 137, 136, 133, 88, 65, 36,
20, 11, 6]
kita akan plot hasil tersebut
```

```
>columnspplot(fs,lab=3:18):
```



kita akan menggunakan rekursi tingkat lanjut.

Fungsi berikut menghitung banyaknya cara bilangan  $k$  dapat direpresentasikan sebagai jumlah dari  $n$  bilangan dalam rentang 1 sampai  $m$ .

```
>function map countways (k; n, m) ...
  if n==1 then return k>=1 && k<=m
  else
    sum=0;
    loop 1 to m; sum=sum+countways(k-#,n-1,m); end;
    return sum;
  end;
endfunction
```

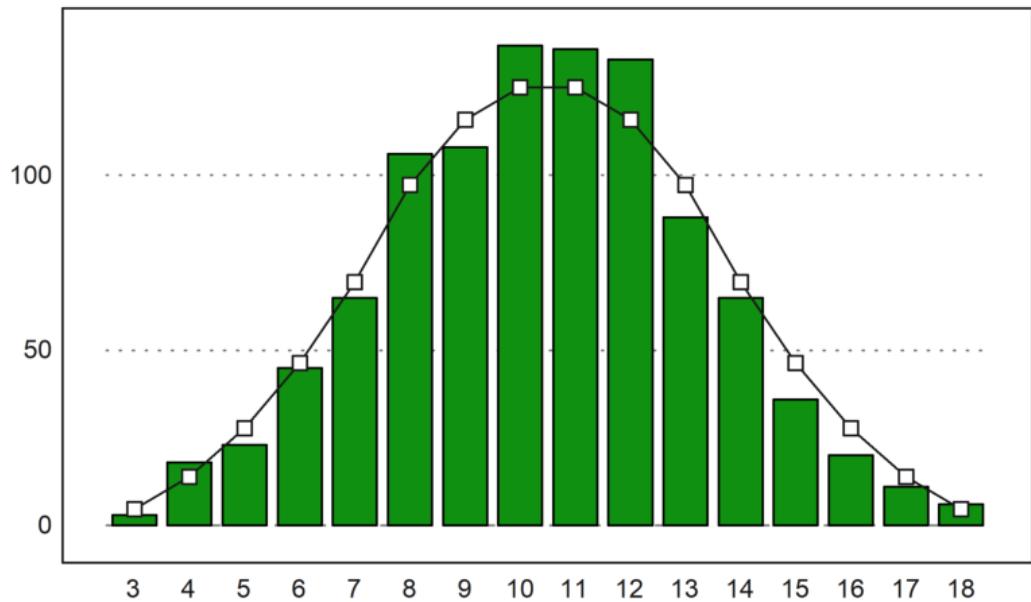
Berikut hasil pelemparan dadu sebanyak lima kali.

```
>countways(5:25,5,5)
[1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320,
255, 185, 121, 70, 35, 15, 5, 1]
>cw=countways(3:18,3,6)
```

```
[1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3,  
1]
```

Kita akan menambahkan nilai yang diharapkan ke plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```



Untuk simulasi lain, deviasi nilai rata-rata n 0-1-variabel acak terdistribusi normal adalah  $1/\sqrt{n}$ .

```
>longformat; 1/sqrt(10)
```

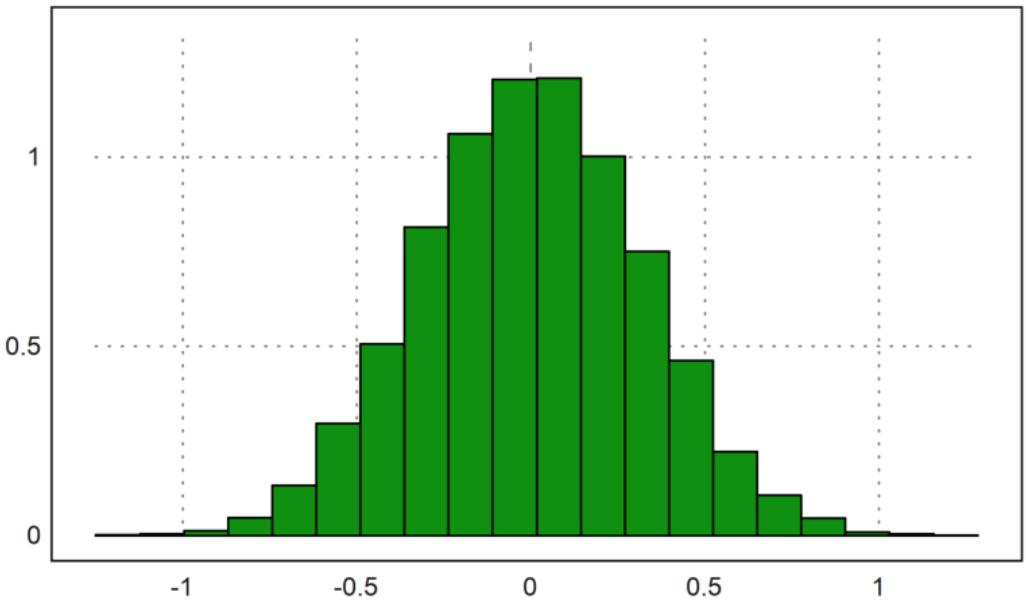
```
0.316227766017
```

Mari kita periksa ini dengan simulasi. Kami menghasilkan 10.000 kali 10 vektor acak.

```
>M=normal(10000,10); dev(mean(M)')
```

```
0.317942172219
```

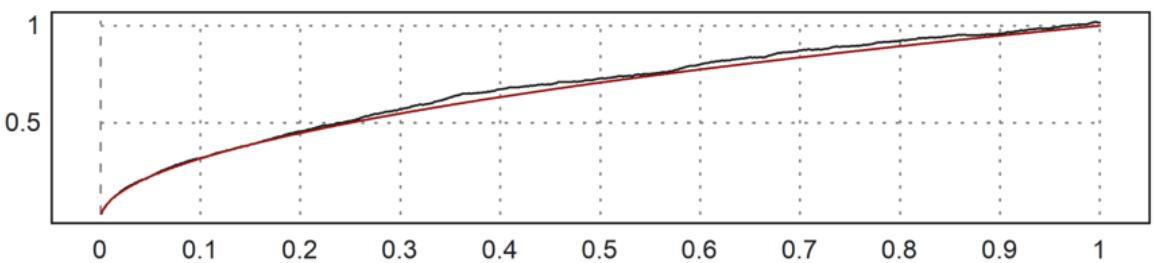
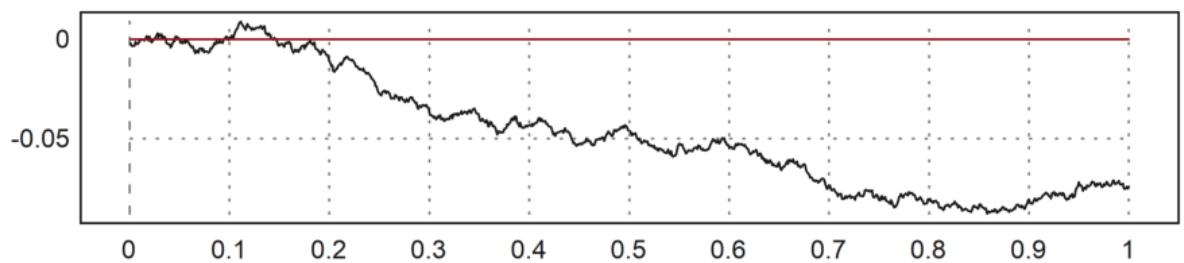
```
>plot2d(mean(M)',>distribution):
```



Median dari 10 bilangan acak berdistribusi normal 0-1 mempunyai deviasi yang lebih besar.

Karena kita dapat dengan mudah menghasilkan jalan acak, kita dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami kemudian memplot deviasi standar dan rata-rata langkah ke-n dari proses ini bersama dengan nilai yang diharapkan berwarna merah.

```
>n=1000; m=1000; M=cumsum(normal(n,m)/sqrt(m)); ...
t=(1:n)/n; figure(2,1); ...
figure(1); plot2d(t,mean(M)'), plot2d(t,0,color=red,>add); ...
figure(2); plot2d(t,dev(M)'), plot2d(t,sqrt(t),color=red,>add); ...
figure(0):
```



## uji chi-kuadrat

uji chi-kuadrat adalah alat penting dalam statistik. Di Euler, banyak tes yang diterapkan. Semua pengujian ini mengembalikan kesalahan yang kita terima jika kita menolak hipotesis nol.

Misalnya, kami menguji lemparan dadu untuk distribusi yang seragam. Pada 600 kali lemparan, kami mendapatkan nilai berikut, yang kami masukkan ke dalam uji chi-kuadrat.

```
>chitest([90,103,114,101,103,89],dup(100,6)')
```

0.498830517952

Ini adalah nilai p-value dari uji chi-kuadrat

Uji chi-kuadrat juga memiliki mode yang menggunakan simulasi Monte Carlo untuk menguji statistiknya, menggunakan Parameter `>p` menafsirkan vektor `y` sebagai vektor probabilitas.

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

0.505

Ini adalah p-value dari uji chi-kuadrat menggunakan pendekatan Monte Carlo. Dengan simulasi Monte Carlo, kita memperoleh p-value yang mirip dengan uji chi-kuadrat standar (0,4988 di uji pertama)

Selanjutnya kita menghasilkan 1000 lemparan dadu menggunakan generator angka acak, dan melakukan tes yang sama.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

0.558957010759

Mari kita uji nilai rata-rata 100 dengan uji-t.

```
>s=200+normal([1,100])*10; ...
```

```
ttest(mean(s),dev(s),100,200)
```

0.25577040307

Fungsi `ttest()` memerlukan nilai mean, deviasi, jumlah data, dan nilai mean yang akan diuji.

Sekarang mari kita periksa dua pengukuran untuk mean yang sama. Kami menolak hipotesis bahwa keduanya mempunyai mean yang sama, jika hasilnya  $<0,05$ .

```
>tcomparedata(normal(1,10),normal(1,10))
```

0.4848094938

Jika kita menambahkan bias pada satu distribusi, kita akan mendapatkan

lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

0.000205842407976

Menambah nilai 2 ke salah satu distribusi menyebabkan p-value menjadi sangat kecil.

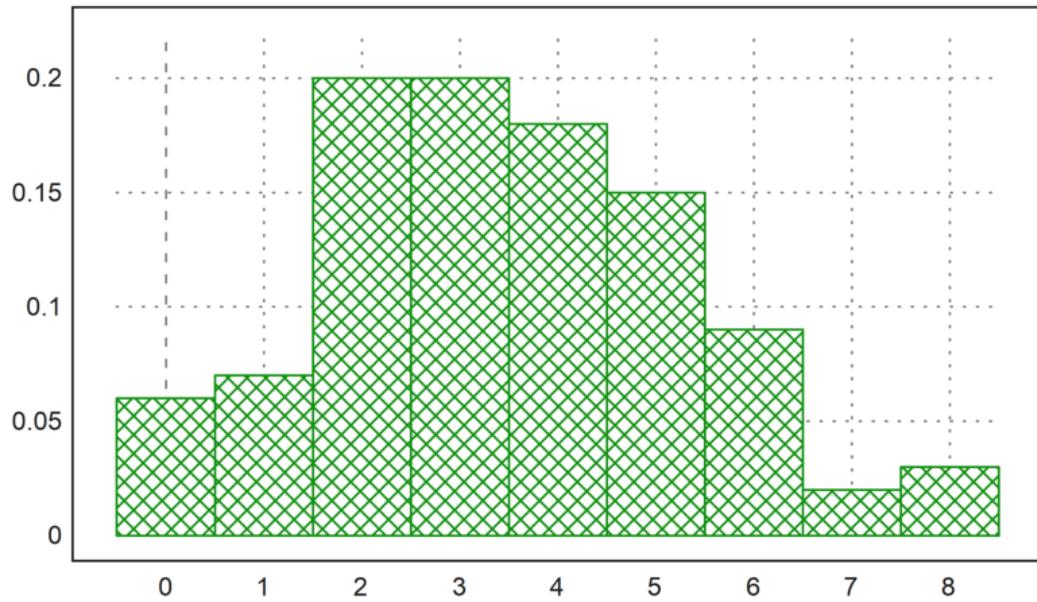
Pada contoh berikutnya, kita membuat 20 lemparan dadu acak sebanyak 100 kali dan menghitung yang ada di dalamnya. Rata-rata harus ada  $20/6=3,3$ .

```
>R=random(100,20); R=sum(R*6<=1)'; mean(R)
```

3.46

Sekarang kita bandingkan jumlah satuan dengan distribusi binomial. Pertama kita plot distribusinya.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\\"/"):
```



kita akan Menghitung frekuensi kemunculan setiap jumlah angka "1" dalam 20 lemparan dadu acak yang telah dilakukan 100 kali

```
>t=count(R,21);
```

Kemudian kami menghitung nilai yang diharapkan.

```
>n=0:20; b=bin(20,n)*(1/6)^n*(5/6)^(20-n)*100;
```

Kita harus mengumpulkan beberapa angka untuk mendapatkan kategori yang cukup besar.

```
>t1=sum(t[1:2])|t[3:7]|sum(t[8:21]); ...
b1=sum(b[1:2])|b[3:7]|sum(b[8:21]);
```

Uji chi-square menolak hipotesis bahwa distribusi kita merupakan distribusi binomial, jika hasilnya <0,05.

```
>chitest(t1,b1)
```

0.855843823605

Contoh berikut berisi hasil dua kelompok orang (misalnya laki-laki dan perempuan) yang memilih satu dari enam partai.

```
>A=[23,37,43,52,64,74;27,39,41,49,63,76]; ...
writetable(A,wc=6,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	23	37	43	52	64	74
f	27	39	41	49	63	76

Kita akan menguji independensi suara dari jenis kelamin.

```
>tabletest(A)
```

0.990701632326

Berikut ini adalah tabel yang diharapkan, jika kita mengasumsikan frekuensi pemungutan suara yang diamati.

```
>writetable(expectedtable(A),wc=6,dc=1,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	24.9	37.9	41.9	50.3	63.3	74.7
f	25.1	38.1	42.1	50.7	63.7	75.3

Kita dapat menghitung koefisien kontingensi yang dikoreksi. Karena sangat mendekati 0, kami menyimpulkan bahwa pemungutan suara tidak bergantung pada jenis kelamin.

```
>contingency(A)
```

0.0427225484717

## uji F

Selanjutnya kita menggunakan analisis varians (uji F) untuk menguji tiga sampel data yang berdistribusi normal untuk nilai mean yang sama.

Metode tersebut disebut ANOVA (analisis varians). Di Euler, fungsi 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

Dengan p-value sebesar 0.0138 (1,38%), kita bisa menolak hipotesis bahwa ketiga sampel memiliki mean yang sama pada tingkat signifikansi 5% (0.05) dan bahkan pada tingkat signifikansi 1% (0.01). Artinya, terdapat perbedaan yang signifikan antara mean dari setidaknya satu sampel.

Ada juga uji median, yang menolak sampel data dengan distribusi rata-rata yang berbeda, menguji median dari sampel yang disatukan.

```
>a=[56,66,68,49,61,53,45,58,54]  
[56, 66, 68, 49, 61, 53, 45, 58, 54]  
>b=[72,81,51,73,69,78,59,67,65,71,68,71]  
[72, 81, 51, 73, 69, 78, 59, 67, 65, 71, 68, 71]  
>mediantest(a,b)
```

0.0241724220052

Tes kesetaraan lainnya adalah tes peringkat. Ini jauh lebih tajam daripada tes median.

```
>ranktest(a,b)  
0.00199969612469  
Pada contoh berikut, kedua distribusi mempunyai mean yang sama.
```

```
>ranktest(random(1,100),random(1,50)*3-1)  
0.162380065554  
ini menunjukkan bahwa perbedaan tidak cukup signifikan pada tingkat signifikansi 5%, sehingga hipotesis bahwa kedua distribusi memiliki median yang sama tidak dapat ditolak.
```

Sekarang mari kita coba mensimulasikan dua perlakuan a dan b yang diterapkan pada orang yang berbeda.

```
>a=[8.0,7.4,5.9,9.4,8.6,8.2,7.6,8.1,6.2,8.9];  
>b=[6.8,7.1,6.8,8.3,7.9,7.2,7.4,6.8,6.8,8.1];  
Tes signum memutuskan, apakah a lebih baik dari b.
```

```
>signtest(a,b)  
0.0546875  
Ini kesalahan yang terlalu besar untuk menolak hipotesis. Kita tidak dapat menolak bahwa a sama baiknya dengan b, Karena p > 0.05.
```

Uji Wilcoxon lebih tajam dibandingkan uji ini, namun mengandalkan nilai kuantitatif perbedaannya.

```
>wilcoxon(a,b)  
0.0296680599405  
Mari kita coba dua tes lagi menggunakan rangkaian yang dihasilkan.  
>wilcoxon(normal(1,20),normal(1,20)-1)  
9.45060349657e-05  
ini menunjukkan bahwa ada perbedaan signifikan antara kedua sampel pada tingkat signifikansi 5%.
```

```
>wilcoxon(normal(1,20),normal(1,20))
```

```
0.559353645673
```

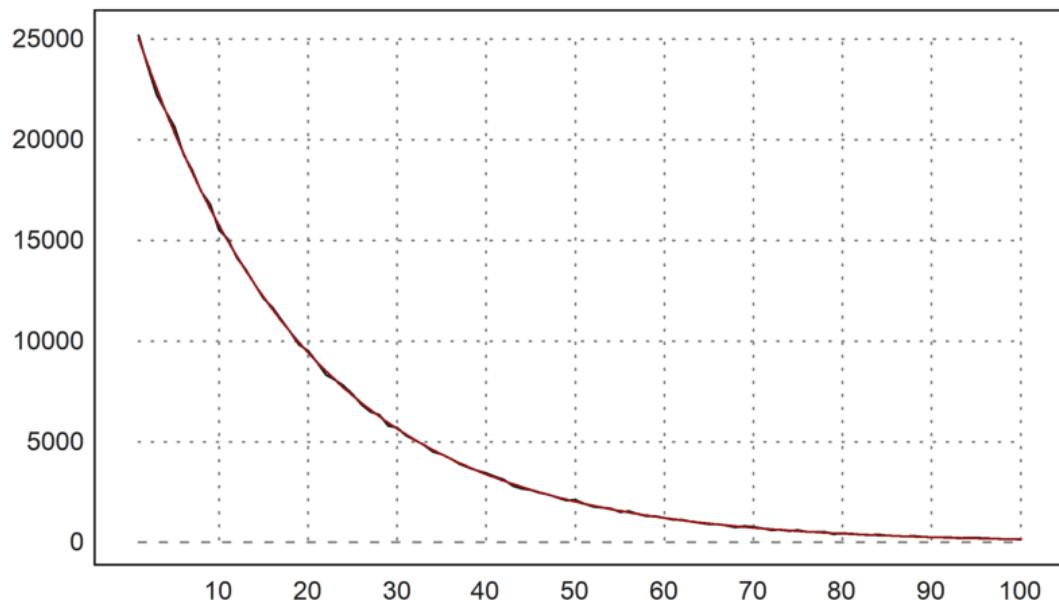
hasil ini jauh di atas 0.05, sehingga kita tidak bisa menolak hipotesis bahwa kedua sampel berasal dari distribusi yang sama.

## Angka Acak

Berikut ini adalah pengujian pembangkit bilangan acak. Euler menggunakan generator yang sangat bagus, jadi kita tidak perlu mengharapkan adanya masalah.

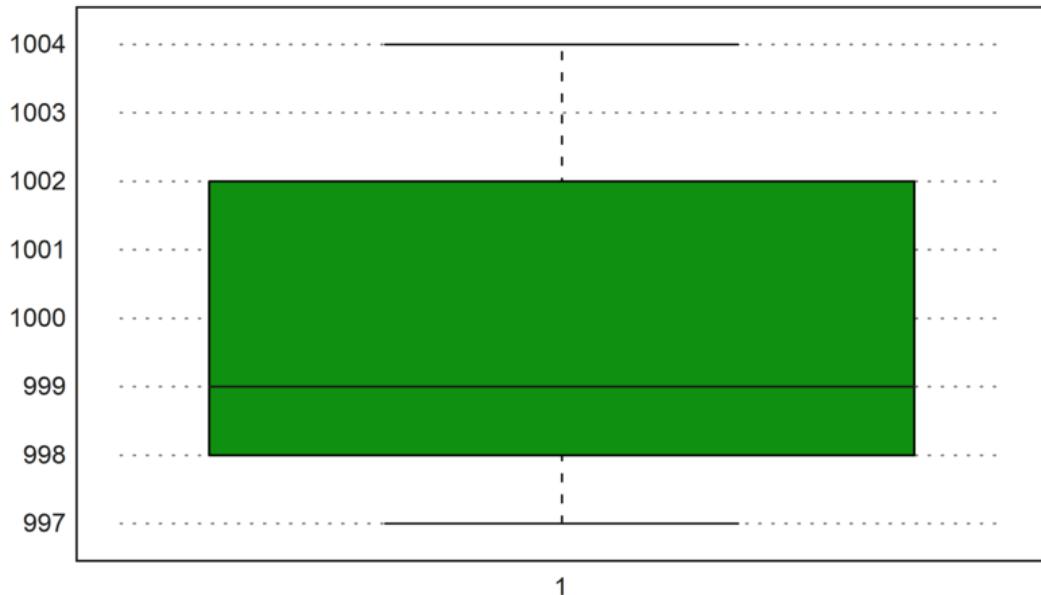
Pertama kita menghasilkan sepuluh juta angka acak di  $[0,1]$ .

```
>n:=10000000; r:=random(1,n);  
Selanjutnya kita hitung jarak antara dua angka yang kurang dari 0,05.  
>a:=0.05; d:=differences(nonzeros(r<a));  
Terakhir, kami memplot berapa kali, setiap jarak terjadi, dan  
membandingkannya dengan nilai yang diharapkan.  
>m=getmultiplicities(1:100,d); plot2d(m); ...  
plot2d("n*(1-a)^(x-1)*a^2",color=red,>add):
```



Hapus datanya.

```
>remvalue n;  
Kami ingin menghitung nilai rata-rata dan simpangan baku yang diukur.  
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...  
mean(M), dev(M),  
999.9  
2.72641400622  
Kita dapat membuat diagram kotak dan kumis untuk data tersebut. Dalam  
kasus kita, tidak ada outlier.  
>boxplot(M):
```



Kami menghitung probabilitas bahwa suatu nilai lebih besar dari 1005, dengan asumsi nilai terukur dan distribusi normal.

Semua fungsi untuk distribusi dalam Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

Kami mencetak hasilnya dalam % dengan akurasi 2 digit menggunakan fungsi cetak.

```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
```

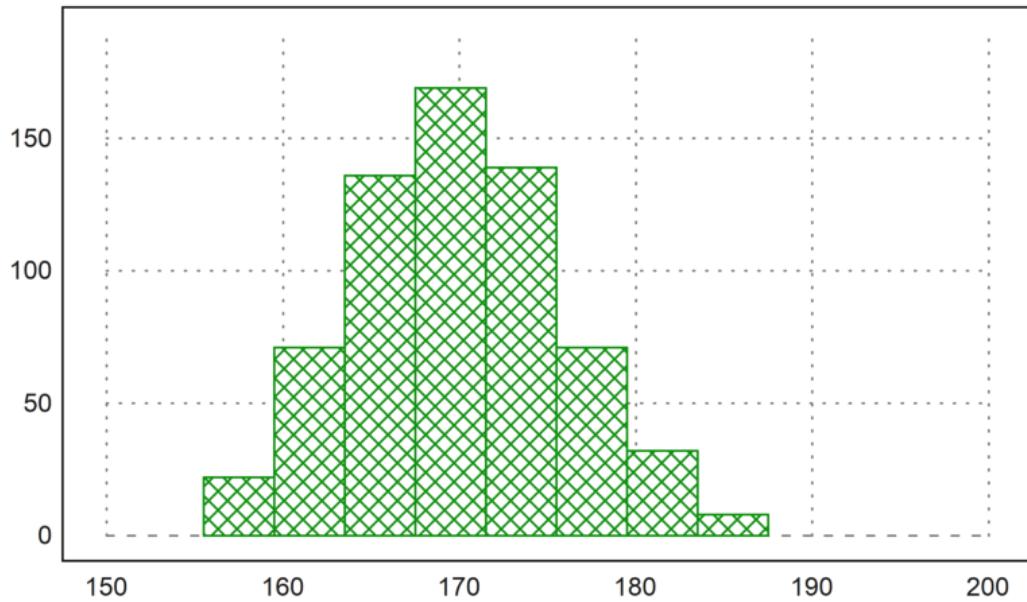
3.07 %

Untuk contoh berikutnya, kami mengasumsikan jumlah pria berikut dalam rentang ukuran tertentu.

```
>r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Berikut adalah plot distribusinya.

```
>plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="\"):
```



Kita dapat memasukkan data mentah tersebut ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal rentang, akhir rentang, jumlah orang dalam rentang.

Tabel dapat dicetak dengan tajuk. Kita menggunakan vektor string untuk mengatur tajuk.

```
>T:=r[1:8]' | r[2:9]' | v'; writetable(T,labc=["from","to","count"])
   from      to    count
155.5    159.5     22
159.5    163.5     71
163.5    167.5    136
167.5    171.5    169
171.5    175.5    139
175.5    179.5     71
179.5    183.5     32
183.5    187.5      8
```

Jika kita memerlukan nilai rata-rata dan statistik ukuran lainnya, kita perlu menghitung titik tengah rentang. Kita dapat menggunakan dua kolom pertama tabel kita untuk ini.

```
>(T[,1]+T[,2])/2
  157.5
  161.5
  165.5
  169.5
  173.5
  177.5
  181.5
  185.5
```

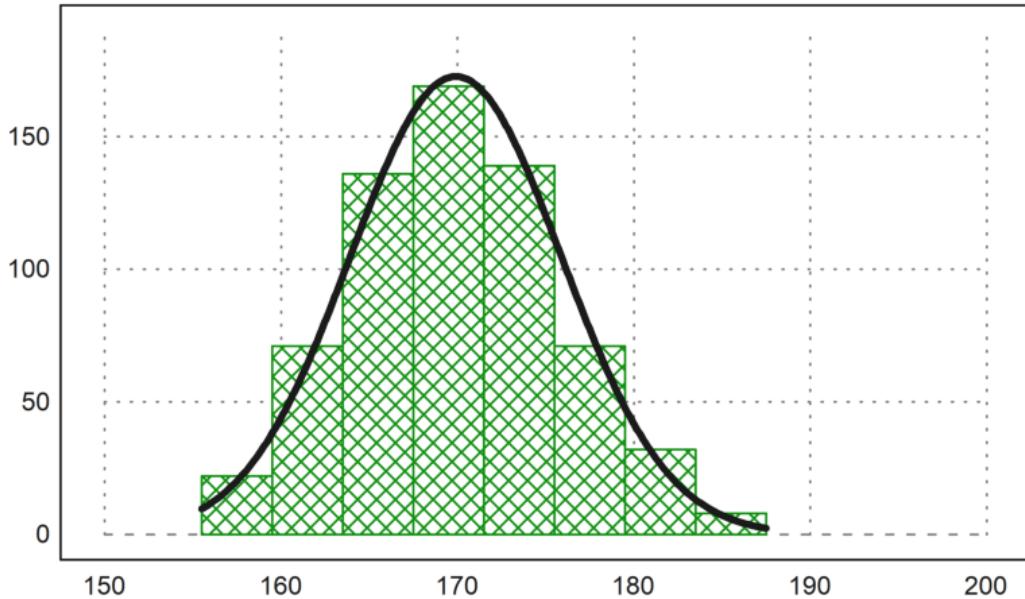
Namun lebih mudah untuk melipat rentang dengan vektor  $[1/2, 1/2]$ .

```
>l=fold(r,[0.5,0.5])
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
Sekarang kita dapat menghitung rata-rata dan deviasi sampel dengan frekuensi yang diberikan.
```

```

>{m,d}=meandev(l,v); m, d,
169.901234568
5.98912964449
Mari kita tambahkan distribusi normal nilai-nilai tersebut ke plot.
>plot2d("qnormal(x,m,d)*sum(v)^4", ...
xmin=min(r),xmax=max(r),thickness=3,add=1):

```



## Pengantar untuk Pengguna Proyek R

Jelasnya, EMT tidak bersaing dengan R sebagai paket statistik. Namun, ada banyak prosedur dan fungsi statistik yang tersedia di EMT juga. Jadi EMT dapat memenuhi kebutuhan dasar. Bagaimanapun, EMT hadir dengan paket numerik dan sistem aljabar komputer.

Notebook ini cocok untuk Anda yang sudah familiar dengan R, namun perlu mengetahui perbedaan sintaksis EMT dan R. Kami mencoba memberikan gambaran umum tentang hal-hal yang sudah jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami mencari cara untuk bertukar data antara kedua sistem.

Note that this is a work in progress.

## Sintaks Dasar

Hal pertama yang Anda pelajari di R adalah membuat vektor. Dalam EMT, perbedaan utamanya adalah operator : dapat mengambil ukuran langkah. Selain itu, ia mempunyai daya ikat yang rendah.

```

>n:=10; 0:n/20:n-1
[0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5,
7, 7.5, 8, 8.5, 9]
>x:=[10.4, 5.6, 3.1, 6.4, 21.7]; [x,0,x]
[10.4, 5.6, 3.1, 6.4, 21.7, 0, 10.4, 5.6, 3.1, 6.4, 21.7]

```

Operator titik dua dengan ukuran langkah EMT digantikan oleh fungsi seq() di R. Kita dapat menulis fungsi ini di EMT.

```

>function seq(a,b,c) := a:b:c; ...
  seq(0,-0.1,-1)
  [0, -0.1, -0.2, -0.3, -0.4, -0.5, -0.6, -0.7, -0.8, -0.9, -
  1]
>function seq(a,b,c) := a:b:c; ...
  seq(0,-0.5,-5)
  [0, -0.5, -1, -1.5, -2, -2.5, -3, -3.5, -4, -4.5, -5]
>function rep(x:vector,n:index) := flatten(dup(x,n)); ...
  rep(x,2)
  [10.4, 5.6, 3.1, 6.4, 21.7, 10.4, 5.6, 3.1, 6.4, 21.7]
  Fungsi rep() dari R tidak ada di EMT. Untuk masukan vektor dapat dituliskan sebagai berikut.

  Perhatikan bahwa "=" atau ":=" digunakan untuk tugas. Operator "->" digunakan untuk satuan dalam EMT.

>125km -> " miles"
77.6713990297 miles
  Operator "<-" untuk penugasan memang bukan ide yang baik untuk R.

  tetapi di EMT operator "<-" itu bukan penugasan melainkan perbandingan
  Berikut ini akan membandingkan a dan -4 di EMT.

>a:=2; a<-4
0
  EMT dan R memiliki vektor bertipe boolean. Namun dalam EMT, angka 0 dan 1 digunakan untuk mewakili salah dan benar. Di R, nilai benar dan salah tetap bisa digunakan dalam aritmatika biasa seperti di EMT.

>x<5, %*x
[0, 0, 1, 0, 0]
[0, 0, 3.1, 0, 0]
  EMT memunculkan kesalahan atau menghasilkan NAN tergantung pada tanda "kesalahan".
>errors off; 0/0, isNaN(sqrt(-1)), errors on;
NAN
1
  Stringnya sama di R dan EMT. Keduanya berada di lokal saat ini, bukan di Unicode.

  Di R ada paket untuk Unicode. Di EMT, string dapat berupa string Unicode. String unicode dapat diterjemahkan ke pengkodean lokal dan sebaliknya. Selain itu, u"..." dapat berisi entitas HTML.

>u"\u00a9; Ren\u00e9 Grothmann"
  © René Grothmann
  karakter khusus (hak cipta © dan karakter aksen é),
>chartoutf([480])
Ā
  Berikut ini mungkin tidak ditampilkan dengan benar pada sistem sebagai
  A dengan titik dan garis di atasnya. Itu tergantung pada font yang Anda gunakan.

  Penggabungan string dilakukan dengan "+" atau "|". Penggabungan ini akan menghasilkan string tunggal, dan angka yang digabungkan akan dikonversi otomatis ke format string. Ini dapat mencakup angka, yang akan dicetak dalam format saat ini.

>"pi = "+pi
pi = 3.14159265359

```

Seringkali, ini akan berfungsi seperti di R.

Namun EMT akan menafsirkan indeks negatif dari belakang vektor, sementara R menafsirkan  $x[n]$  sebagai  $x$  tanpa elemen ke- $n$ .

```
>x, x[1:3], x[-2]
[10.4, 5.6, 3.1, 6.4, 21.7]
[10.4, 5.6, 3.1]
6.4
>x, x[1:5], x[-3]
[10.4, 5.6, 3.1, 6.4, 21.7]
[10.4, 5.6, 3.1, 6.4, 21.7]
3.1
Untuk meniru perilaku R di EMT, kita dapat menggunakan fungsi
drop(x,n)
>drop(x,2)
[10.4, 3.1, 6.4, 21.7]
Vektor logika tidak diperlakukan berbeda sebagai indeks di EMT,
berbeda dengan R. Anda perlu mengekstrak elemen bukan nol terlebih
dahulu di EMT.
>x, x>5, x[nonzeros(x>5)]
[10.4, 5.6, 3.1, 6.4, 21.7]
[1, 1, 0, 1, 1]
[10.4, 5.6, 6.4, 21.7]
Sama seperti di R, vektor indeks dapat berisi pengulangan.
>x[[1,2,2,1]]
[10.4, 5.6, 5.6, 10.4]
```

## Tipe Data

EMT memiliki lebih banyak tipe data tetap daripada R. Jelasnya, di R terdapat vektor yang berkembang. Anda dapat mengatur vektor numerik kosong  $v$  dan memberikan nilai ke elemen  $v[17]$ . Hal ini tidak mungkin dilakukan di EMT.

Berikut ini agak tidak efisien.

```
>v=[]; for i=1 to 10000; v=v|i; end;
kenapa cara ini kurang efisien? karna setiap elemen baru di tambahkan
EMT harus menyalin seluruh isi  $v$  kembali ke variabel  $v$ .
```

Semakin efisien vektor telah ditentukan sebelumnya.

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
Untuk mengubah tipe data di EMT, Anda dapat menggunakan fungsi
seperti
kompleks().
>complex(1:4)
[ 1+0i , 2+0i , 3+0i , 4+0i ]
Konversi ke string hanya dimungkinkan untuk tipe data dasar. Format
saat ini digunakan untuk penggabungan string sederhana. Tapi ada
fungsi seperti print() atau frac().
```

Untuk vektor, Anda dapat dengan mudah menulis fungsi Anda sendiri.

```
>function tostr (v) ...
s="[";  
loop 1 to length(v);
  s=s+print(v[#],2,0);
  if #<length(v) then s=s+","; endif;
end;
return s+"]";
endfunction
```

- Variabel s diinisialisasi sebagai string "[ " untuk menyimpan hasil akhir. Awalnya, tanda kurung buka [ ditambahkan ke variabel s sebagai pembuka.
- loop 1 to length(v); menjalankan perulangan dari elemen pertama hingga elemen terakhir dalam v. Fungsi length(v) mengembalikan panjang atau jumlah elemen dalam vektor v.
- print(v[#, 2, 0]; adalah fungsi format yang mengonversi elemen vektor v pada posisi saat ini (v[#]) menjadi string. parameter 2 menunjukkan bahwa dua digit setelah titik desimal akan ditampilkan, sementara 0 memastikan bahwa angka ditampilkan tanpa tambahan simbol atau format lainnya.
- Bagian if #<length(v) memeriksa apakah elemen saat ini bukan elemen terakhir. Jika benar, maka koma , akan ditambahkan ke variabel s untuk memisahkan elemen.
- Setelah loop selesai, tanda kurung tutup ] ditambahkan ke string s, dan string ini kemudian dikembalikan sebagai output.

```
>tostr(linspace(0,1,10));
Untuk komunikasi dengan Maxima, terdapat fungsi convertmxm(), yang juga dapat digunakan untuk memformat vektor untuk keluaran.
```

```
>convertmxm(1:10);
Untuk Latex perintah tex dapat digunakan untuk mendapatkan perintah Latex.
```

```
>tex(&[1,2,3]);
[1,2,3][1,2,3]
```

## Faktor dan Tabel

---

Dalam pengantar R ada contoh yang disebut faktor.

Berikut ini adalah daftar wilayah 30 negara bagian.

```
>austates = ["tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa", ...
"qld", "vic", "nsw", "vic", "qld", "qld", "sa", "tas", ...
"sa", "nt", "wa", "vic", "qld", "nsw", "nsw", "wa", ...
"sa", "act", "nsw", "vic", "vic", "act"];
Perintah diatas digunakan untuk mendefinisikan sebuah array (array sendiri adalah sekumpulan variabel yang memiliki tipe data yang sama) karena pada data tersebut ada beberapa nama negara bagian yang terulang. Array ini berisi singkatan untuk negara bagian dan teritori di Australia.
```

Asumsikan, kita memiliki pendapatan yang sesuai di setiap negara bagian.

```
>incomes = [60, 49, 40, 61, 64, 60, 59, 54, 62, 69, 70, 42, 56, ...
61, 61, 61, 58, 51, 48, 65, 49, 49, 41, 48, 52, 46, ...
59, 46, 58, 43];
Sekarang mari kita coba mencari nilai mean dan median dari data pendapatan tersebut menggunakan perintah mean(incomes) dan median(incomes)
```

```
>mean(incomes)
54.73333333333333
>median(incomes)
57
```

Sekarang, kami ingin menghitung rata-rata pendapatan di suatu wilayah.

Menjadi program statistik, R memiliki faktor() dan tapply() untuk ini.

EMT dapat melakukan hal ini dengan menemukan indeks wilayah dalam daftar wilayah unik.

```
>auterr=sort(unique(austates)); f=indexofsorted(auterr,austates)
[6, 5, 4, 2, 2, 3, 8, 8, 4, 7, 2, 7, 4, 4, 5, 6, 5,
3,
8, 7, 4, 2, 2, 8, 5, 1, 2, 7, 7, 1]
```

Pada titik itu, kita dapat menulis fungsi perulangan kita sendiri untuk melakukan sesuatu hanya untuk satu faktor.

Atau kita bisa meniru fungsi tapply() dengan cara berikut.

```
>function map tappl (i; f$call, cat, x) ...
u=sort(unique(cat));
f=indexof(u,cat);
return f$(x[nonzeros(f==indexof(u,i))]);
endfunction
```

i: Parameter pertama biasanya adalah nilai yang digunakan untuk pencocokan atau pemetaan.

f\$call: Parameter kedua, yang kemungkinan besar adalah sebuah fungsi yang dipanggil dalam kode tersebut. f\$ di sini merujuk pada fungsi yang diterima sebagai input.

cat: Parameter ketiga adalah array atau vektor yang berisi kategori yang akan diproses.

x: Parameter keempat adalah array atau vektor yang akan diproses atau diubah berdasarkan pemetaan kategori yang dilakukan.

Ini agak tidak efisien, karena menghitung wilayah unik untuk setiap i,  
tetapi berhasil.

```
>tappl(auterr,"mean",austates,incomes)
```

```
[44.5, 57.3333333333, 55.5, 53.6, 55, 60.5, 56, 52.25]
```

Perhatikan bahwa ini berfungsi untuk setiap vektor wilayah.

```
>tappl(["act","nsw"],"mean",austates,incomes)
```

```
[44.5, 57.3333333333]
```

Sekarang, paket statistik EMT mendefinisikan tabel seperti di R.  
Fungsi readtable() dan writetable() dapat digunakan untuk input dan output.

Sehingga kita bisa mencetak rata-rata pendapatan negara di daerah secara bersahabat.

```
>writetable(tappl(auterr,"mean"),austates,incomes),labc=auterr,wc=7)
```

act	nsw	nt	qld	sa	tas	vic	wa
44.5	57.33	55.5	53.6	55	60.5	56	52.25

Fungsi writetable digunakan untuk menampilkan data dalam bentuk tabel yang terstruktur dengan label kolom dan lebar kolom yang dapat disesuaikan.

Dengan labc=auterr, berarti menetapkan label kolom untuk tabel tersebut berdasarkan kategori yang ada di auterr(yang sudah diurutkan sesuai abjad).

wc(width of columns)=7 berarti setiap kolom dalam tabel akan memiliki lebar minimal 7 karakter.

sebagai contoh 44.5 itu memiliki 4 karakter (termasuk titik desimal). karena data dalam kolom lebih pendek dari 7 karakter, kolom tersebut diberi ruang ekstra untuk tampilan yang rapi.

Kita juga bisa mencoba meniru perilaku R sepenuhnya.

Faktor-faktor tersebut harus disimpan dengan jelas dalam kumpulan beserta jenis dan kategorinya (negara bagian dan teritori dalam contoh

kita). Untuk EMT, kami menambahkan indeks yang telah dihitung sebelumnya.

```
>function makef (t) ...
  ## Factor data
```

```

## Returns a collection with data t, unique data, indices.
## See: tapply
u=sort(unique(t));
return {{t,u,indexofsorted(u,t)}};
endfunction
>statef=makef(austates);

```

Perintah statef = makef(austates); digunakan untuk mengolah data yang ada dalam variabel austates, dan mengidentifikasi elemen unik yang ada dalam data tersebut.

Sekarang elemen ketiga dari koleksi akan berisi indeks.

```

>statef[3]
[6, 5, 4, 2, 2, 3, 8, 8, 4, 7, 2, 7, 4, 4, 5, 6, 5,
3,
8, 7, 4, 2, 2, 8, 5, 1, 2, 7, 7, 1]

```

statef[3] adalah elemen ketiga dari koleksi yang dikembalikan oleh fungsi makef, yaitu indeks posisi dari elemen-elemen dalam austates yang sudah dipetakan ke urutan dalam u (data unik yang terurut). statef[3] akan mengembalikan indeks posisi dari setiap elemen dalam austates berdasarkan urutan yang ada di u.

Sekarang kita bisa meniru tapply() dengan cara berikut. Ini akan mengembalikan tabel sebagai kumpulan data tabel dan judul kolom.

```

>function tapply (t:vector,tf,f$:call) ...
## Makes a table of data and factors
## tf : output of makef()
## See: makef
uf=tf[2]; f=tf[3]; x=zeros(length(uf));
for i=1 to length(uf);
  ind=nonzeros(f==i);
  if length(ind)==0 then x[i]=NAN;
  else x[i]=f$(t[ind]);
  endif;
end;
return {{x,uf}};
endfunction

```

Kami tidak menambahkan banyak pengecekan tipe di sini. Satu-satunya tindakan pencegahan menyangkut kategori (faktor) yang tidak memiliki data. Tetapi kita harus memeriksa panjang t yang benar dan kebenaran pengumpulan tf.

Tabel ini dapat dicetak sebagai tabel dengan writetable().

```

>writetable(tapply(incomes,statef,"mean"),wc=7)
      act    nsw     nt     qld     sa     tas     vic     wa
  44.5   57.33   55.5   53.6    55   60.5    56   52.25

```

## Array

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Namun, akan mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau perpustakaan C untuk ini.

R memiliki lebih dari dua dimensi. Di R array adalah vektor dengan bidang dimensi.

Dalam EMT, vektor adalah matriks dengan satu baris. Itu dapat dibuat menjadi matriks dengan redim().

```

>shortformat; X=redim(1:20,4,5)
      1          2          3          4          5

```

```

6      7      8      9      10
11     12     13     14     15
16     17     18     19     20

```

Fungsi shortformat digunakan untuk mengatur format tampilan angka agar

lebih ringkas dan mudah dibaca.

Perintah diatas digunakan untuk membuat matrik X dari angka 1 sampai 20 dengan ketentuan matriks dengan 4 baris dan 5 kolom.

Ekstraksi baris dan kolom, atau sub-matriks, mirip dengan R.

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

```

2      3
7      8
12     13
17     18

```

Perintah diatas digunakan untuk menampilkan matriks X kolom kedua sampai ketiga.

```
>X[,3:5]
```

```

3      4      5
8      9      10
13     14     15
18     19     20

```

Namun, di R dimungkinkan untuk menyetel daftar indeks tertentu ke suatu nilai. Hal yang sama mungkin terjadi di EMT hanya dengan satu putaran.

```
>function setmatrixvalue (M, i, j, v) ...
loop 1 to max(length(i),length(j),length(v))
  M[i#{},j#{}] = v#;
end;
endfunction
```

Perintah setmatrixvalue(M, i, j, v) adalah fungsi yang digunakan untuk mengubah nilai elemen-elemen dalam matriks berdasarkan indeks tertentu.

M: Matriks yang akan dimodifikasi.

i: Indeks baris atau posisi baris dalam matriks M yang ingin diubah.

j: Indeks kolom atau posisi kolom dalam matriks M yang ingin diubah.

v: Nilai yang akan dimasukkan ke dalam elemen-elemen matriks M pada posisi yang ditentukan oleh indeks i dan j.

Kami mendemonstrasikan ini untuk menunjukkan bahwa matriks dilewatkan dengan referensi di EMT. Jika Anda tidak ingin mengubah matriks M asli, Anda perlu menyalinnya ke dalam fungsi.

```
>setmatrixvalue(X,1:3,3:-1:1,0); X,
```

```

1      2      0      4      5
6      0      8      9      10
0      12     13     14     15
16     17     18     19     20

```

Perkalian luar dalam EMT hanya dapat dilakukan antar vektor. Ini otomatis karena bahasa matriks. Satu vektor harus berupa vektor kolom dan vektor lainnya harus berupa vektor baris.

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

```

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

```

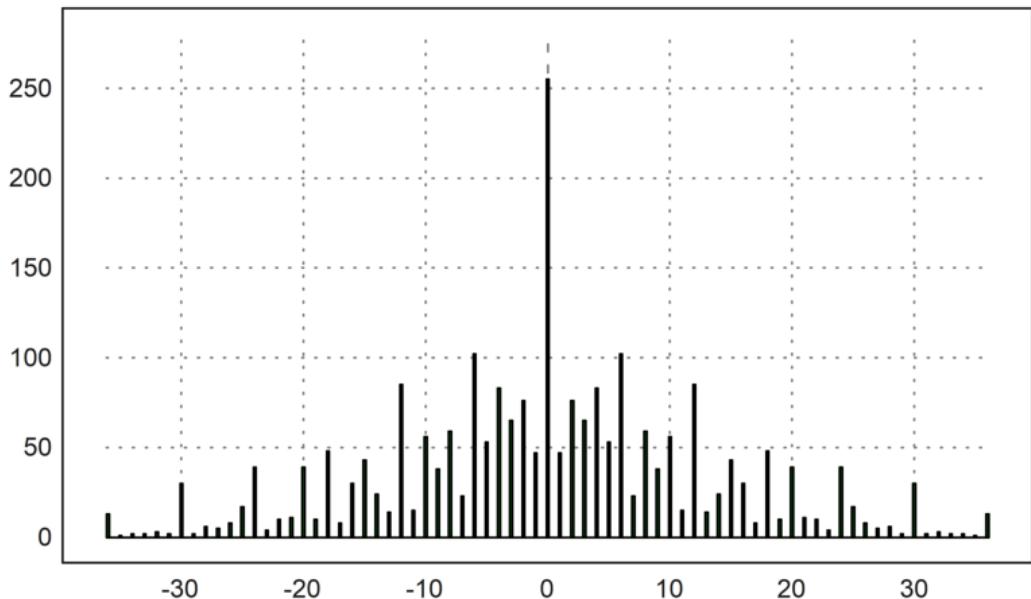
```
1:5: Ini adalah vektor baris yang berisi angka-angka dari 1 hingga 5  
(1:5)': Tanda ' di sini menunjukkan transposisi dari vektor baris  
1:5.  
Dengan kata lain, ini mengubah vektor baris menjadi vektor kolom.
```

Dalam PDF pendahuluan untuk R terdapat contoh yang menghitung distribusi  $ab - cd$  untuk  $a, b, c, d$  yang dipilih dari 0 hingga  $n$  secara acak. Solusi dalam R adalah membentuk matriks 4 dimensi dan menjalankan `table()` di atasnya.

Tentu saja, hal ini dapat dicapai dengan satu putaran. Tapi loop tidak efektif di EMT atau R. Di EMT, kita bisa menulis loop di C dan itu akan menjadi solusi tercepat.

Namun kita ingin meniru perilaku R. Untuk melakukannya, kita perlu meratakan perkalian  $ab$  dan membuat matriks  $ab - cd$ .

```
>a=0:6; b=a'; p=flatten(a*b); q=flatten(p-p'); ...  
u=sort(unique(q)); f=getmultiplicities(u,q); ...  
statplot(u,f,"h"):
```

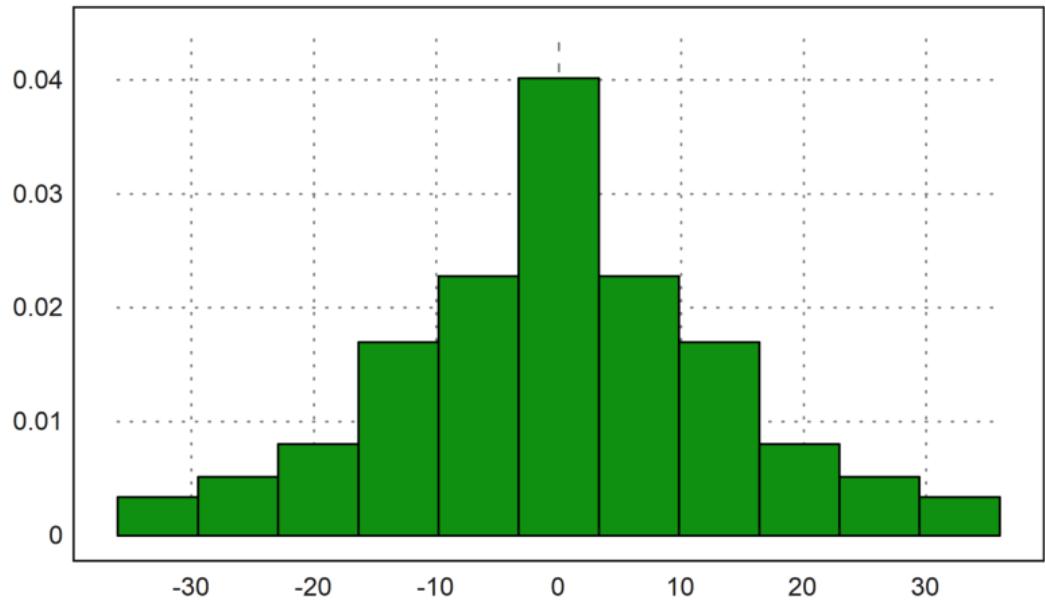


Selain multiplisitas eksak, EMT dapat menghitung frekuensi dalam vektor.

```
>getfrequencies(q,-50:10:50)  
[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]  
Perintah diatas digunakan untuk menghitung distribusi frekuensi nilai-nilai dalam vektor  $q$  dalam rentang dari -50 hingga 50, dengan interval 10. Fungsi ini menghitung berapa banyak nilai dalam  $q$  yang jatuh dalam setiap interval: [-50, -40), [-40, -30), ..., [40, 50).
```

Cara paling mudah untuk memplotnya sebagai distribusi adalah sebagai berikut.

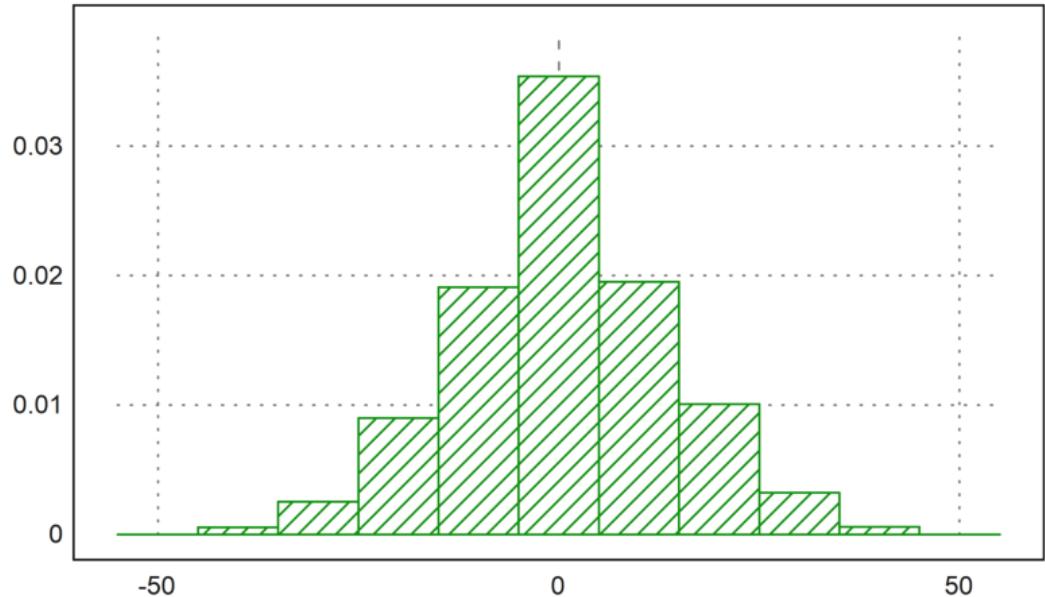
```
>plot2d(q,distribution=11):
```



Namun dimungkinkan juga untuk menghitung terlebih dahulu penghitungan dalam interval yang dipilih sebelumnya. Tentu saja, berikut ini menggunakan `getfrequencies()` secara internal.

Karena fungsi `histo()` mengembalikan frekuensi, kita perlu menskalakannya sehingga integral di bawah grafik batang adalah 1.

```
>{x,y}=histo(q,v=-55:10:55); y=y/sum(y)/differences(x); ...
plot2d(x,y,>bar,style="/"):
```



EMT memiliki dua jenis daftar. Salah satunya adalah daftar global yang bisa berubah, dan yang lainnya adalah tipe daftar yang tidak bisa diubah. Kami tidak peduli dengan daftar global di sini.

Tipe daftar yang tidak dapat diubah disebut koleksi di EMT. Ini berperilaku seperti struktur di C, tetapi elemennya hanya diberi nomor dan tidak diberi nama.

#### 1. Membuat list dan mengakses elemen dalam list

```
>L={"Fred","Flintstone",40,[1990,1992]}
```

```
Fred  
Flintstone  
40  
[1990, 1992]
```

Perintah diatas digunakan untuk membuat list L dengan nama depan Fred, nama belakang Flintstone, usia 40, dan tahun 1990, 1992. Namun untuk tahun tersebut tidak dapat dipastikan apa arti dari tahun-tahun tersebut, bisa saja tahun kelahiran dan kematian, tahun pendidikan, tahun pekerjaan, atau yang lainnya.

Saat ini unsur-unsur tersebut tidak memiliki nama, meskipun nama dapat ditetapkan untuk tujuan khusus. Mereka diakses dengan nomor.

```
>(L[4])[2]
```

```
1992
```

Perintah diatas digunakan untuk menampilkan list L keempat urutan kedua. Karena pada list L keempat berisi tahun yang dimana terdapat 2 tahun, tahun pertama adalah 1990 dan tahun kedua adalah 1992. Perintah tersebut ingin menampilkan tahun kedua, maka outputnya adalah 1992.

#### 2. Menggabungkan dua list

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

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

```
>B := [4,5,6]
```

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

```
>C := [A, B]
```

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

#### 3. Mengubah elemen dalam list

```
>D := [7,8,9,10]
```

```
[7, 8, 9, 10]
```

```
>D[3] := 99
```

```
[7, 8, 99, 10]
```

#### 4. menghitung panjang list

```
>E := [10,20,30,40,50,60,70]
```

```
[10, 20, 30, 40, 50, 60, 70]
```

```
>len := length(E)
```

```
7
```

## File Input dan Output (Membaca dan Menulis Data)

Anda sering kali ingin mengimpor matriks data dari sumber lain ke EMT.

Tutorial ini memberi tahu Anda tentang banyak cara untuk mencapai hal ini. Fungsi sederhananya adalah writematrix() dan readmatrix().

Mari kita tunjukkan cara membaca dan menulis vektor real ke file.

```

>a=random(1,100); mean(a), dev(a),
0.44779
0.29446
mean=1/nΣi=1nximean=1/nΣi=1nxi
dev=1/(n-1)Σi=1n(xi-x)2-----√dev=1/(n-1)Σi=1n(xi-x)2

Untuk menulis data ke file, kita menggunakan fungsi writematrix().

Karena pengenalan ini kemungkinan besar ada di direktori, di mana pengguna tidak memiliki akses tulis, kami menulis data ke direktori home pengguna. Untuk buku catatan sendiri, hal ini tidak diperlukan, karena file data akan ditulis ke dalam direktori yang sama.

>filename="test.dat";
Sekarang kita menulis vektor kolom a' ke file. Ini menghasilkan satu nomor di setiap baris file.

>writematrix(a',filename)
Untuk membaca data, kita menggunakan readmatrix()

>a=readmatrix(filename)'
[0.74157, 0.13184, 0.082656, 0.1975, 0.63019, 0.51127, 0.98091,
0.020501, 0.29063, 0.83916, 0.051539, 0.45459, 0.77198, 0.8754,
0.49257, 0.96765, 0.48685, 0.11645, 0.086971, 0.35536, 0.69515,
0.19161, 0.66654, 0.61295, 0.38779, 0.16748, 0.65044, 0.64784,
0.17712, 0.048322, 0.24795, 0.8682, 0.49634, 0.9125, 0.079817,
0.6544, 0.66279, 0.54949, 0.10807, 0.71727, 0.75473, 0.34844,
0.38634, 0.40978, 0.3234, 0.32117, 0.85977, 0.78211, 0.12933,
0.39591, 0.018822, 0.95697, 0.26985, 0.2409, 0.52222, 0.79655,
0.30484, 0.53489, 0.014971, 0.22246, 0.13021, 0.82292, 0.83054,
0.99006, 0.064585, 0.57221, 0.16539, 0.31818, 0.16037, 0.71343,
0.6188, 0.67365, 0.22367, 0.86965, 0.67439, 0.859, 0.85639,
0.07405, 0.65051, 0.79298, 0.039088, 0.4536, 0.24248, 0.10425,
0.36493, 0.084869, 0.37012, 0.045487, 0.66347, 0.69294, 0.46928,
0.27625, 0.20081, 0.069966, 0.2009, 0.52929, 0.97226, 0.52319,
0.11857, 0.075214]
Dan hapus file tersebut.

>fileremove(filename);
>mean(a), dev(a),
0.44779
0.29446
Fungsi writematrix() atau writetable() dapat dikonfigurasi untuk bahasa lain.

Misalnya, jika Anda memiliki sistem Indonesia (titik desimal dengan koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan dengan titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan koma). File berikut "test.csv" akan muncul di folder saat ini Anda.

>filename="test.csv"; ...
writematrix(random(5,3),file=filename,separator=",")
Anda sekarang dapat membuka file ini dengan Excel bahasa Indonesia secara langsung.

>fileremove(filename);
Terkadang kita memiliki string dengan token seperti berikut.

>s1:="f m m f m m m f f f m m f"; ...
s2:="f f f m m f f";
Untuk melakukan tokenisasi ini, kami mendefinisikan vektor token.

>tok:=["f","m"]
f
m
Kemudian kita dapat menghitung berapa kali setiap token muncul dalam

```

```

        string, dan memasukkan hasilnya ke dalam tabel.
>M:=getmultiplicities(tok,strtokens(s1))_ ...
    getmultiplicities(tok,strtokens(s2));
    Tulis tabel dengan header token.
>writetable(M,labc=tok,labr=1:2,wc=8)
      f      m
    1     6     7
    2     5     2
Untuk statika, EMT dapat membaca dan menulis tabel.
>file="test.dat"; open(file,"w"); ...
writeln("A,B,C"); writematrix(random(3,3)); ...
close();
The file looks like this.
>printfile(file)
A,B,C
0.5894214399628446,0.1500948494104958,0.4911562128682984
0.2817236329179764,0.3586251183800097,0.1159031925964036
0.2135992153342503,0.3457575959104093,0.3271359940090028

Fungsi readtable() dalam bentuknya yang paling sederhana dapat
membaca
ini dan mengembalikan kumpulan nilai dan baris judul.
>L=readtable(file,>list);
Koleksi ini dapat dicetak dengan writetable() ke buku catatan, atau
ke
file.
>writetable(L,wc=10,dc=5)
      A          B          C
    0.58942   0.15009   0.49116
    0.28172   0.35863   0.1159
    0.2136    0.34576   0.32714
Matriks nilai adalah elemen pertama dari L. Perhatikan bahwa mean()
di
EMT menghitung nilai rata-rata baris matriks.
>mean(L[1])
0.41022
0.25208
0.2955

```

## File CSV

---

Pertama, mari kita menulis matriks ke dalam file. Untuk outputnya, kami membuat file di direktori kerja saat ini.

```

>file="test.csv"; ...
M=random(3,3); writematrix(M,file);
Here is the content of this file.
>printfile(file)
0.5537619367566726,0.6738419762237779,0.04291882768650555
0.2401989793483695,0.0861139922960146,0.3678722550483149
0.1343166294586442,0.3616257867488041,0.3383621511798272

CSV ini dapat dibuka pada sistem berbahasa Inggris ke Excel dengan
klik dua kali. Jika Anda mendapatkan file seperti itu di sistem
Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik
desimal.

Namun titik desimal juga merupakan format default untuk EMT. Anda
dapat membaca matriks dari file dengan readmatrix().
>readmatrix(file)
0.55376   0.67384   0.042919

```

```
0.2402  0.086114  0.36787
0.13432 0.36163  0.33836
```

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
```

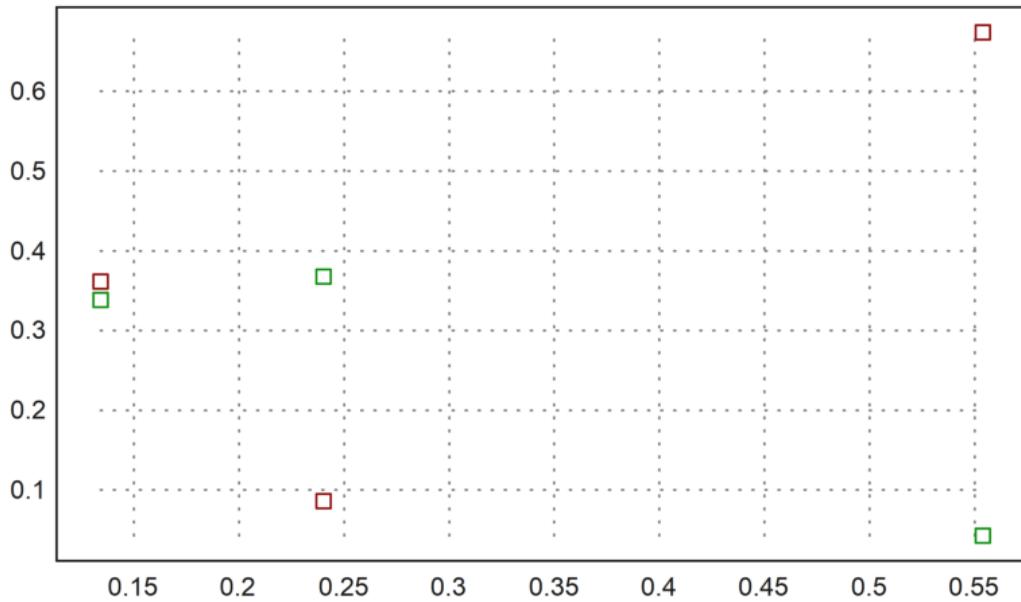
Di Excel atau spreadsheet serupa, Anda dapat mengekspor matriks sebagai CSV (nilai yang dipisahkan koma). Di Excel 2007, gunakan "save as" dan "other format", lalu pilih "CSV". Pastikan tabel saat ini hanya berisi data yang ingin Anda ekspor.

Ini sebuah contoh.

```
> printfile("excel-data.csv")
Could not open the file
excel-data.csv
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
    open(filename,"r");
Seperti yang Anda lihat, sistem bahasa Jerman saya menggunakan titik
koma sebagai pemisah dan koma desimal. Anda dapat mengubahnya di
pengaturan sistem atau di Excel, tetapi hal ini tidak diperlukan
untuk
membaca matriks menjadi EMT.
```

Cara termudah untuk membaca ini ke dalam Euler adalah `readmatrix()`. Semua koma diganti dengan titik dengan parameter `>koma`. Untuk CSV bahasa Inggris, hilangkan saja parameter ini.

```
>M=readmatrix("excel-data.csv",>koma)
Could not open the file
excel-data.csv
for reading!
Try "trace errors" to inspect local variables after errors.
readmatrix:
    if filename<>"" then open(filename,"r"); endif;
    Let us plot this.
>plot2d(M'[1],M'[2:3],>points,color=[red,green]'):
```



Ada cara yang lebih mendasar untuk membaca data dari suatu file. Anda dapat membuka file dan membaca angka baris demi baris. Fungsi `getvectorline()` akan membaca angka dari sebaris data. Secara default, ini mengharapkan titik desimal. Tapi bisa juga menggunakan koma desimal, jika Anda memanggil `setdecimaldot(",")` sebelum Anda menggunakan fungsi ini.

Fungsi berikut adalah contohnya. Itu akan berhenti di akhir file atau baris kosong.

```
>function myload (file) ...
open(file);
M=[];
repeat
    until eof();
    v=getvectorline(3);
    if length(v)>0 then M=M_v; else break; endif;
end;
return M;
close(file);
endfunction
>myload(file)
0.55376      0  0.67384      0  0.042919
0.2402       0  0.086114     0  0.36787
0.13432      0  0.36163      0  0.33836
Dimungkinkan juga untuk membaca semua angka dalam file itu dengan
getvector().
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
0.55376      0  0.67384
0  0.042919  0.2402
0  0.086114  0
```

Oleh karena itu sangat mudah untuk menyimpan suatu vektor nilai, satu nilai di setiap baris dan membaca kembali vektor ini.

```
>v=random(1000); mean(v)
0.493
>writematrix(v',file); mean(readmatrix(file)')
0.493
```

## Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik.

Misalnya, kita menulis tabel dengan header baris dan kolom ke sebuah file.

```
>file="test.tab"; M=random(3,3); ...
open(file,"w");
writetable(M,separator=",",labc=["one","two","three"]); ...
close(); ...
printfile(file)
one,two,three
0.56,      0.63,      0.59
0.45,      0.86,      0.44
0.72,      0.19,      0.91
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
0.56      0.63      0.59
0.45      0.86      0.44
0.72      0.19      0.91
```

## Menganalisis Garis

Pada subbab ini sering digunakan untuk memproses atau mengekstrak data

dari teks yang berformat khusus, seperti data tabel dalam HTML. Anda bahkan dapat mengevaluasi setiap baris dengan tangan. Misalkan, kita memiliki baris dengan format berikut.

2020-11-03,Tue,1'114.05 2020-11-03,Tue,1'114.05

```
>line="2020-11-03,Tue,1'114.05"
2020-11-03,Tue,1'114.05
Pertama, kita akan memisahkan string line menjadi bagian-bagian yang lebih kecil, yang dikenal sebagai "token".
>vt=strtoks(line)
2020-11-03
Tue
1'114.05
Kemudian kita dapat mengevaluasi setiap elemen garis menggunakan evaluasi yang sesuai.
>day(vt[1]); ...
indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])); ...
strrepl(vt[3],"'',''')();
Dengan menggunakan ekspresi reguler, dimungkinkan untuk mengekstrak hampir semua informasi dari sebaris data.
```

Selanjutnya, kita akan melihat bagaimana mengekstrak data string yang berisi markup HTML menggunakan ekspresi reguler.

```
>line("<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>");
Untuk mengekstraknya, kami menggunakan ekspresi reguler, yang mencari

- tanda kurung tutup >, untuk mengindikasikan bahwa kita akan mencari awal dari elemen yang ada di dalam tag.
- string apa pun yang tidak mengandung tanda kurung akan mencocokkan elemen di dalam tag <td>.
```

- braket pembuka dan penutup menggunakan solusi terpendek, dengan tag pembuka (`<td>`) dan penutup (`</td>`).  
 - sekali lagi string apa pun yang tidak mengandung tanda kurung, ini akan menjamin bahwa kita akan mengambil isi yang relevan di dalam tagnya.  
 - dan tanda kurung buka `<` menandai bahwa ini adalah akhir dari tag dan awal dari tag baru.

Mencari pola tertentu dalam string line yang menggunakan ekspresi reguler.

```
>{pos,s,vt}=strxfind(line,">([^\>]+)<.+?>([^\>]+)<" );
Hasilnya adalah posisi kecocokan, string yang cocok, dan vektor
string
untuk sub-kecocokan.
```

Kita akan mengeksekusi elemen-elemen di dalam array atau list vt satu per satu dalam sebuah perulangan.

```
>for k=1:length(vt); vt[k](), end;
1145.5
5.6
Berikut adalah fungsi yang membaca semua item numerik antara <td> dan </td>.
>function readtd (line) ...
v=[]; cp=0;
repeat
  {pos,s,vt}=strxfind(line,"<td.*?>(.+?)</td>",cp);
  until pos==0;
  if length(vt)>0 then v=v|vt[1]; endif;
  cp=cp+strlen(s);
end;
return v;
endfunction
Kita akan mengekstrak dan menampilkan semua nilai yang berada di
antara tag <td>...</td> dalam baris, dan mencari apakah nilai tersebut
numerik atau bukan.
>readtd(line+"<td>non-numerical</td>")
1145.45
5.6
-4.5
non-numerical
```

## Membaca dari Web

---

Situs web atau file dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris.

Dalam contoh, kita membaca versi terkini dari situs EMT. Kami menggunakan ekspresi reguler untuk memindai "Versi ..." dalam sebuah judul.

```
>function readversion () ...
urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html");
repeat
  until urleof();
  s=urlgetline();
  k=strfind(s,"Version ",1);
  if k>0 then substring(s,k,strfind(s,"<",k)-1), break; endif;
end;
urlclose();
endfunction
```

```

>readversion
Version 2024-01-12
Contoh lain membaca URL dengan EMT
"https://mywebsite.com/version.h"
>function readversionmywebsite () ...
urlopen("https://mywebsite.com/version.h");
repeat
    until urleof();
    s=urlgetline();
    k=strfind(s,"Release",1);
    if k>0 then substring(s,k,strfind(s,<,k)-1); break; endif;
end;
urlclose();
endfunction
>readversionmywebsite
Karena string "Release" tidak ada di dalam file version.h, maka
strfind(s, "Release", 1) akan mengembalikan nilai nol atau tidak
menghasilkan indeks yang diperlukan untuk proses pencarian.

```

## Input dan Output Variabel

---

Anda dapat menulis variabel dalam bentuk definisi Euler ke file atau ke baris perintah.

```

>writevar(pi,"mypi");
mypi = 3.141592653589793;
Untuk pengujian, kami membuat file Euler di direktori kerja EMT.
>file="tes.e"; ...
writevar(random(2,2),"M",file); ...
printfile(file,3)
M = [ ..
0.3888176479607355, 0.3870125417113612;
0.2576115973745406, 0.1750949536837037];
Sekarang kita dapat memuat file tersebut. Ini akan mendefinisikan
matriks M.
>load(file); show M,
M =
0.38882   0.38701
0.25761   0.17509
Omong-omong, jika writevar() digunakan pada suatu variabel, definisi
variabel dengan nama variabel tersebut akan dicetak.
>writevar(M); writevar(inch$)
M = [ ..
0.3888176479607355, 0.3870125417113612;
0.2576115973745406, 0.1750949536837037];
inch$ = 0.0254;
Kita juga bisa membuka file baru atau menambahkan file yang sudah
ada.
Dalam contoh kita menambahkan file yang dibuat sebelumnya.
>open(file,"a"); ...
writevar(random(2,2),"M1"); ...
writevar(random(3,1),"M2"); ...
close();
>load(file); show M1; show M2;
M1 =
0.17881   0.026345
0.63559   0.34332
M2 =
0.22274
0.91626
0.4616

```

```

Untuk menghapus file apa pun, gunakan fileremove().
>fileremove(file);
Vektor baris dalam suatu file tidak memerlukan koma, jika setiap
angka
berada pada baris baru. Mari kita buat file seperti itu, tulis setiap
baris satu per satu dengan writeln().

>open(file,"w"); writeln("M = [");
for i=1 to 5; writeln(""+random()); end; ...
writeln("]"); close(); ...
printfile(file)
M =
0.743886392671
0.92708388148
0.46506977432
0.0265574069961
0.147047179518
];
>load(file); M
[0.74389, 0.92708, 0.46507, 0.026557, 0.14705]

```

## LATIHAN

---

1. Misalkan anda memiliki vektor  $x=[2,4,6,8,10]$ 
  - a. buatkan vektor yang menggabungkan vektor  $x$ , angka 0 dan vektor  $x$  lagi
  - b. tentukan apakah setiap elemen vektor  $x$  lebih besar dari 5 (hasil logika 1 untuk benar dan 0 untuk salah)

```

>x:=[2,4,6,8,10]; [x,0,x]
[2, 4, 6, 8, 10, 0, 2, 4, 6, 8, 10]
>x>5, %*x
[0, 0, 1, 1, 1]
[0, 0, 6, 8, 10]

```

2. Tentukan matriks  $X$  dengan elemen-elemen yang berurutan dari 1 hingga 20 dan susunlah elemen tersebut menjadi matriks berukuran  $5 \times 4$ .

```

>shortformat; X=redim(1:20,5,4)
      1      2      3      4
      5      6      7      8
      9      10     11     12
     13     14     15     16
     17     18     19     20

```

3. Seorang analis memiliki data penjualan harian selama 5 hari (150, 200, 250, 300, 350) yang disimpan dalam bentuk vektor sebagai berikut:

- a. mean(rata-rata)
- b. deviasi standar

```

>penjualan=[150,200,250,300,350]
[150, 200, 250, 300, 350]
atau anda bisa memanggil data yang sudah dibuat

```

```

>filename="penjualan.dat";
>writematrix(penjualan,filename)
>penjualan=readmatrix(filename)'
[150, 200, 250, 300, 350]

```

```
>mean(penjualan)
```

```
250
```

```
>dev(penjualan)
```

```
79.057
```

4. Buat fungsi yang membuka URL "[https://en.wikipedia.org/wiki/Euler\\_\(software\)](https://en.wikipedia.org/wiki/Euler_(software))" dan mencari kata "Versi" di dalam URL tersebut, dan tampilkan hasilnya.

```
>function readversionwebsite () ...
```

```

urlopen("https://en.wikipedia.org/wiki/Euler_(software)");
repeat
  until urleof();
  s=urlgetline();
  k=strfind(s,"version",1);
  if k>0 then substring(s,k,strfind(s,<,k)-1), break; endif;
end;
urlclose();
endfunction
>readversion

```

Version 2024-01-12

5. Diberikan data pengukuran tinggi badan pada kelas matematika B adalah sebagai berikut:

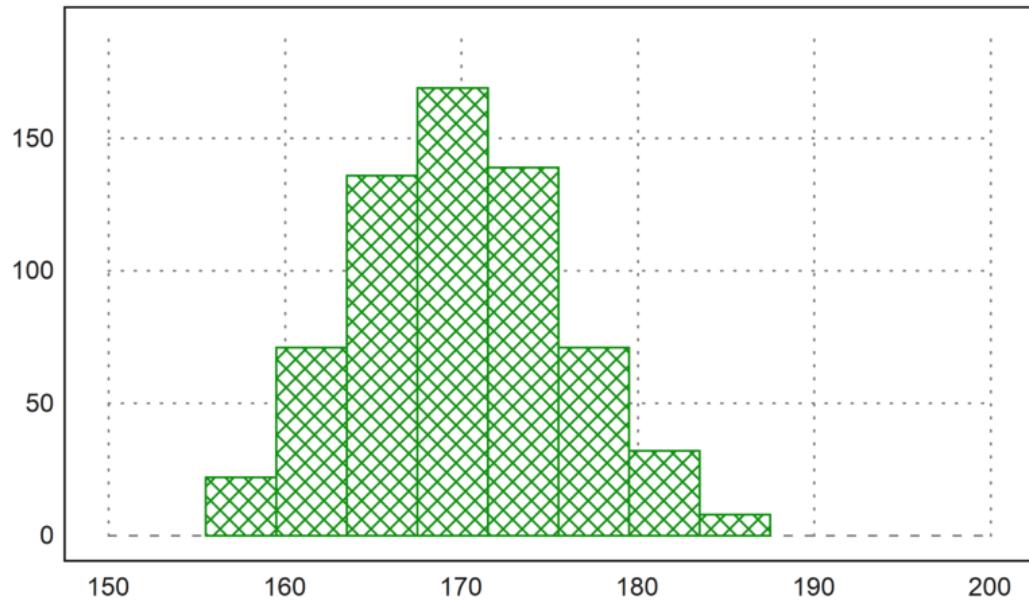
Rentang Tinggi (cm)	Jumlah Orang
155.5 - 159.5	22
159.5 - 163.5	71
163.5 - 167.5	136
167.5 - 171.5	169
171.5 - 175.5	139
175.5 - 179.5	71
179.5 - 183.5	32
183.5 - 187.5	8

- a.) Hitung rata-rata dan deviasi standar dari distribusi tinggi badan ini.
- b.) Plot distribusi frekuensi data (diagram batang).
- c.) Tambahkan kurva distribusi normal untuk dibandingkan dengan data.

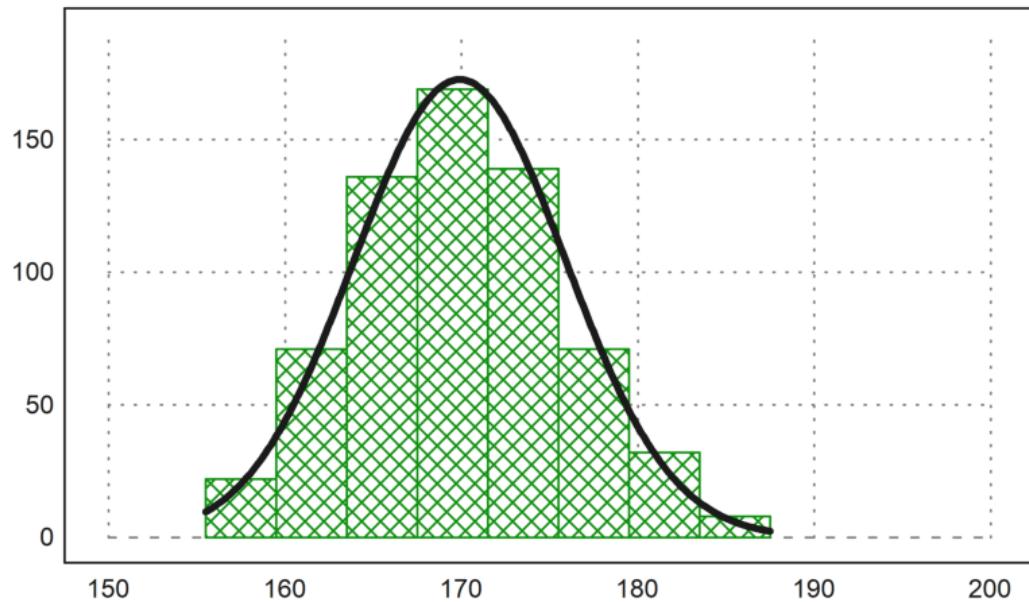
```

>r = 155.5:4:187.5 //Rentang ukuran tinggi badan
[155.5, 159.5, 163.5, 167.5, 171.5, 175.5, 179.5, 183.5,
187.5]
>v = [22, 71, 136, 169, 139, 71, 32, 8] //Jumlah orang dalam tiap rentang
[22, 71, 136, 169, 139, 71, 32, 8]
>l=fold(r,[0.5,0.5]) //Menghitung titik tengah dari setiap rentang tinggi
badan
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
>{m,d}=meandev(l,v); m, d, //Hitung rata-rata dan deviasi standar
169.9
5.9891
>plot2d(r, v, a=150, b=200, c=0, d=190, bar=1, style="\\"):

```



```
>plot2d("qnormal(x, m, d) * sum(v) * 4", ...
  xmin=min(r), xmax=max(r), thickness=3, add=1):
```



```
>&remvalue();
6. Sebuah survei dilakukan untuk mengetahui jumlah jam belajar siswa SMA dalam satu minggu. Berikut data jam belajar dari 10 siswa: 8, 10, 7, 6, 9, 10, 11, 9, 8, 12.
a) Hitung nilai rata-rata dari data di atas
b) Tentukan median dari data tersebut.
>M=[8,10,7,6,9,10,11,9,8,12];
>mean(M)
9
>median(M)
```

9

7. Anda diberikan data yang menunjukkan jumlah penjualan barang selama 12 bulan dalam satu tahun berturut-turut 120, 135, 150, 160, 170, 180, 190, 210, 200, 220, 230, 240.
- Buatlah plot garis dari data penjualan barang tersebut.
  - Hitung rata-rata penjualan perbulan.

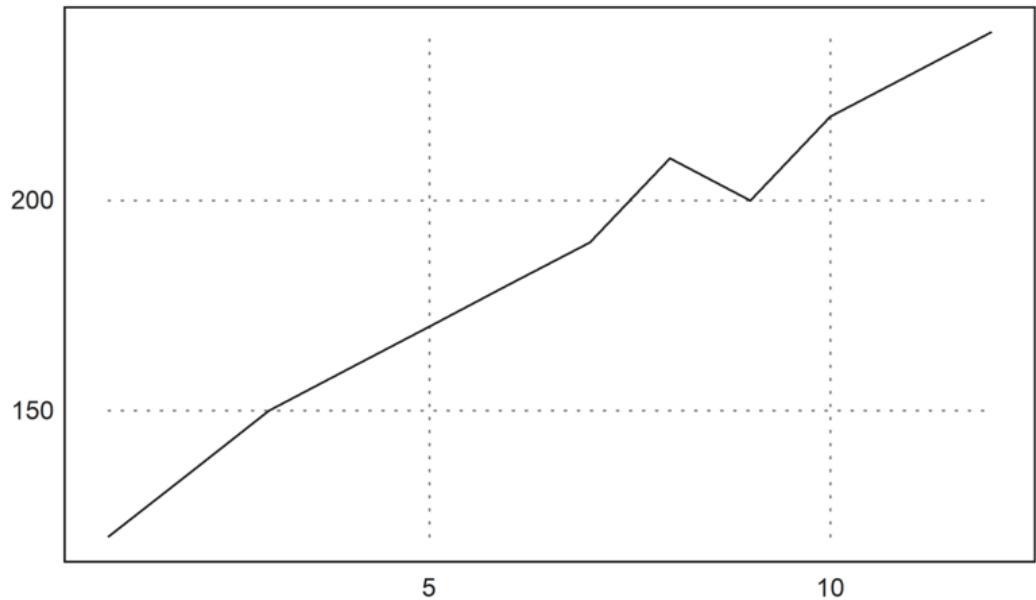
```
>X=[120,135,150,160,170,180,190,210,200,220,230,240]
```

```
[120, 135, 150, 160, 170, 180, 190, 210, 200, 220, 230,  
240]
```

```
>Y=[1,2,3,4,5,6,7,8,9,10,11,12]
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

```
>statplot(Y,X,"1"):
```



```
>mean(X)
```

```
183.75
```