

# Proyek EMT Mata Kuliah Aplikasi dan Komputer

Hamemayu Hayuningrat

November 29, 2024

# Contents

1 Aljabar dengan EMT	2
2 Plot 2D dengan EMT	177
3 Plot 3D dengan EMT	361
4 Menggunakan EMT untuk Kalkulus	465
5 Penggunaan EMT untuk Geometri	637
6 Statistika pada EMT	773

# Chapter 1

## Aljabar dengan EMT

## **EMT untuk Perhitungan Aljabar**

---

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

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

### **Contoh pertama**

---

Menyederhanakan bentuk aljabar:

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

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

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

Menjabarkan:

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

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

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

---

**Baris Perintah**

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

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

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

16.7551608191

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

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

50.2654824574  
100.530964915

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

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

0.540302305868

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

0.857553215846

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

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

1.41666666667  
1.41421568627  
1.41421356237

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

Untuk melipat semua garis yang ada tekan Ctrl+L. Maka garis-garis berikutnya hanya akan terlihat jika salah satu dari garis-garis tersebut menjadi fokus. Untuk melipat satu baris ganda, mulailah baris pertama dengan "%+".

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

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  
1.416666666667  
1.41421568627  
1.41421356237  
1.41421356237
```

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

```
>x := 1.5; // komentar pergi ke sini sebelum ...
>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 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 mengeklik bagian komentar di atas perintah untuk membuka perintah tersebut.

Saat 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 yang ingin 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 expand di bawah pada baris perintah.

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

2.5

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

---

## 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 dalam 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 ";". Tanda titik koma menghilangkan keluaran perintah. Di akhir baris perintah, tanda "," 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 mendapatkan bantuan. Perhatikan bahwa konstanta Euler e diberi nama E dalam EMT.

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

8.77908249441

Untuk menghitung ekspresi rumit seperti

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

Anda perlu memasukkannya dalam formulir 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. Angka ini dicetak dengan akurasi sekitar 12 digit secara default.

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

dan masih banyak lagi.

Berikut 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).

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

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

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

```
>longest 1/3
```

```
0.3333333333333333
```

Representasi ganda internal membutuhkan 8 byte.

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

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

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

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

---

String

Suatu string dalam Euler didefinisikan dengan "...".

```
>"Suatu string dapat berisi apa saja."
```

Suatu string dapat berisi apa saja.

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<sup>2</sup>.

Fungsi cetak juga mengonversi angka menjadi string. Fungsi ini dapat memuat 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
```

```
= 45°
```

I

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

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

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

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

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

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

Ü is a German letter

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

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

We have =.

Dimungkinkan juga untuk menggunakan entitas numerik.

```
>u"\u00d6hnliches"
```

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

"and" adalah operator "`&&`" dan "or" adalah operator "`||`", seperti dalam bahasa C. (Kata "and" dan "or" hanya dapat digunakan dalam kondisi "if".)

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

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 defaulnya adalah (12).

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

```
0.333333333333
```

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

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

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

Format default untuk skalar adalah format(12). 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 keluaran yang paling penting.

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

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

Namun format keluaran EMT dapat diatur secara fleksibel.

```
>longestformat; pi,
```

```
3.141592653589793
```

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

```
3.14159
```

Standarnya adalah deformat().

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

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

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

```
4.934802200544679
```

Ada juga operator 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

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.

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

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

12

Berdasarkan konvensi, ekspresi simbolik atau numerik harus diberi nama fx, fxy, 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 mendasar 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 ditegaskan.

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

-0.425

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

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

Matematika simbolik terintegrasi dengan lancar ke dalam Euler dengan &. Ekspresi apa pun yang dimulai dengan & adalah ekspresi simbolik. Ekspresi ini dievaluasi dan dicetak oleh Maxima.

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

```
>$&44!
```

```
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" pada 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 suatu persamaan dalam persamaan lainnya.

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

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

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

Sekali lagi, % mengacu pada hasil sebelumnya.

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

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

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

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

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

2432902008176640000

```
>::: 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
```

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

$$x^3 e^x$$

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

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

$$\begin{matrix} 5 \\ 125 E \end{matrix}$$

$$125 e^5$$

$$18551.64488782208$$

```
>fx(5)
```

$$18551.6448878$$

Untuk mengevaluasi suatu 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(%)
```

$$1000 \text{ E}^{10} - 125 \text{ E}^5$$

$$2.20079141499189\text{e}+7$$

```
>$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)
```

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

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  $x = \text{dst}$ . Jika Anda tidak memerlukan hasil numerik untuk perhitungan lebih lanjut, Anda juga dapat membiarkan Maxima menemukan nilai numeriknya.

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

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

[-3.23607, 1.23607]

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

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

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

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

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

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

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

2

Ekspresi 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 ditentukan dengan perintah “function”. Fungsi dapat berupa fungsi satu baris atau fungsi multibaris. Fungsi satu baris dapat berupa numerik atau simbolik. Fungsi satu baris numerik didefinisikan dengan “:=”.

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

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

```
>f(2)
```

4.472135955

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

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

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

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

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

0.786151377757

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

Anda masih dapat memanggil fungsi bawaan sebagai “...”, jika fungsi tersebut merupakan fungsi dalam inti Euler.

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

0.707106781187

Sebaiknya kita hilangkan definisi ulang tentang dosa ini.

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

0.707106781187

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

16

Menetapkannya akan menimpa nilai default.

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

80

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

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

16

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

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

24

Tetapi parameter yang ditetapkan akan menggantikan 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-fungsi ini didefinisikan dalam Euler dan Maxima, dan dapat digunakan di kedua bahasa tersebut. Ekspresi pendefinisian dijalankan melalui Maxima sebelum definisi.

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

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

Fungsi simbolis dapat digunakan dalam ekspresi simbolis.

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

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

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

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

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

$$178.635099908$$

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

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

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

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

0.703467422498

Hal berikut ini juga dapat digunakan, 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(%)
```

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

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

625

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

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

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

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

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

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

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

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

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

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

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

$$x^3 - x$$

Dengan `&=`, fungsi ini bersifat simbolis, dan dapat digunakan dalam ekspresi simbolis lainnya.

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

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

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

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

yang tidak dapat dievaluasi secara simbolik.

Jika kita mendefinisikan ulang fungsi tersebut dengan kata kunci “map”, maka fungsi tersebut dapat digunakan untuk vektor x.

Secara internal, fungsi tersebut dipanggil untuk semua nilai x satu kali, dan hasilnya disimpan dalam sebuah vektor.

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

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

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

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

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

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

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

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

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

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

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

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

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

0

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

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

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

Untuk meringkas

- $\&=$  mendefinisikan fungsi simbolik,
- $:=$  mendefinisikan fungsi numerik,
- $\&\&=$  mendefinisikan fungsi simbolik murni.

## Memecahkan Ekspresi

---

Ekspresi dapat diselesaikan secara numerik dan simbolik.

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

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

1.41421356237

Hal ini juga bisa digunakan untuk ekspresi simbolis. Perhatikan fungsi berikut ini.

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

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

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

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

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

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

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

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

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

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

Sekarang kita mencari titik, di mana polinomialnya adalah 2. Dalam solve(), nilai target default y=0 dapat diubah dengan variabel yang ditetapkan.  
ami menggunakan y=2 dan mengeceknya dengan mengevaluasi polinomial pada hasil sebelumnya.

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

0.966715594851

2

Memecahkan sebuah ekspresi simbolik dalam bentuk simbolik mengembalikan sebuah 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 “with”.

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

0

Menyelesaikan sistem persamaan secara simbolik dapat dilakukan dengan vektor persamaan dan pemecah simbolik `solve()`. Jawabannya adalah sebuah daftar daftar persamaan.

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

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

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

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

dengan `a = 3`.

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

Salah satu cara untuk mengoper parameter tambahan ke `f()` adalah dengan menggunakan sebuah daftar yang berisi nama fungsi dan parameternya (cara lainnya adalah dengan menggunakan parameter titik koma).

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

2.54116291558

Hal ini juga dapat dilakukan dengan ekspresi. Namun, elemen daftar bernama harus digunakan. (Lebih lanjut tentang daftar dalam tutorial tentang sintaks EMT).

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

2.54116291558

## Menyelesaikan Pertidaksamaan

---

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

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

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

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

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

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

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

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

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

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

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

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

$$\emptyset$$

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

$$\text{universal set}$$

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

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

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

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

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

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

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

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

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

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

```
>${&fourier_elim}(((x + y < 5) \text{ and } x < 1) \text{ 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 diskusi terperinci tentang bahasa matriks Euler. Vektor dan matriks dimasukkan dengan tanda kurung siku, elemen dipisahkan dengan koma, baris dipisahkan dengan titik koma.

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

1	2
3	4

Hasil kali matriks dilambangkan dengan sebuah titik.

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

3
4

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

[3, 4]

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

```
-2 1  
1.5 -0.5
```

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

```
11  
25
```

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

```
1 0  
0 1
```

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

```
>A.A
```

```
7 10  
15 22
```

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

1	4
9	16

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

37	54
81	118

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

37	54
81	118

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

1	1
1	1

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

```
0.333333      0.666667  
0.75          1
```

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

```
-2  
2.5
```

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

```
-2  
2.5
```

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

```
1      0  
0      1
```

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

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

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

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

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

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

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

ika salah satu operan adalah vektor atau skalar, maka operan tersebut akan diperluas dengan cara alami.

```
>2*A
```

$$\begin{array}{ccc} 2 & & 4 \\ & 6 & 8 \end{array}$$

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

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

$$\begin{array}{ccc} 1 & & 4 \\ & 3 & 8 \end{array}$$

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

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

$$\begin{array}{ccc} 2 & & 6 \\ & 6 & 12 \end{array}$$

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

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

1	2
1	2

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

1	4
3	8

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

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

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

Sekali lagi, ingatlah bahwa ini bukan produk matriks!

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

55

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

55

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

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

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

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

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

5

Euler memiliki operator perbandingan, seperti “==”, yang memeriksa kesetaraan. Kita mendapatkan vektor 0 dan 1, di mana 1 berarti benar.

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

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

Dari vektor seperti itu, “nonzeros” memilih elemen bukan nol.

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

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

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

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

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

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

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

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

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

EMT tidak sepenuhnya efektif untuk komputasi bilangan bulat. EMT menggunakan floating point presisi ganda secara internal. Akan tetapi, hal ini sering kali sangat berguna.

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

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

112

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

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

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

Ini mengembalikan indeks elemen, yang bukan nol.

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

1	4
2	1
2	2
3	2

Indeks ini dapat digunakan untuk menetapkan elemen ke suatu nilai.

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

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

Fungsi mset() juga dapat mengatur elemen-elemen pada indeks ke entri-entri matriks lain.

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

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

Dan dimungkinkan untuk mendapatkan elemen-elemen dalam vektor.

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

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

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

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

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

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

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

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

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

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

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

Tetapi dengan mget(), kita dapat mengekstrak indeks dan menggunakan informasi ini untuk mengekstrak elemen-elemen pada posisi yang sama dari matriks yang 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, kolom yang lebih pendek akan diisi dengan 0.

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

1	2	3
1	2	3

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

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

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

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

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

```
>A|1
```

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

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

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

1	2	3
1	2	3

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

1	1
2	2
3	3

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

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

1	1
2	4
3	9

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut ini.

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

2	
4	
[2,	4]
4	

Untuk vektor, ada length().

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

Ada banyak fungsi lain yang menghasilkan matriks.

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

1	1
1	1

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

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

[6, 6, 6, 6, 6]

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

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

0.66566	0.831835
0.977	0.544258

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

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

1	2	3
4	5	6
7	8	9

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

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

Mari kita uji.

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

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

Fungsi multdup() menduplikasi elemen-elemen sebuah 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() membalik urutan baris atau kolom dari sebuah matriks. Misalnya, fungsi flipx() membalik secara horizontal.

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

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

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

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

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

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

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

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

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

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

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
[0, 5, 0, 6, 0, 0, 0, 7, 0, 8, 0]
[2, 3, 5, 7, 23, 29, 31, 37, 41, 43, 47]
```

Seperti yang Anda lihat, tidak ada salahnya menyertakan indeks di luar jangkauan (seperti 0), indeks ganda, atau indeks yang tidak diurutkan.

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

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

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

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

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

Kemudian, kami menetapkan diagonal bawah (-1) ke 1:4.

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

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

Perhatikan bahwa kita tidak mengubah matriks A. Kita mendapatkan sebuah matriks baru sebagai hasil dari setdiag().

Berikut adalah sebuah fungsi yang mengembalikan sebuah matriks tri-diagonal.

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

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

Diagonal sebuah matriks juga dapat diekstrak dari matriks. Untuk mendemonstrasikan hal ini, kami merestrukturisasi vektor 1:9 menjadi matriks 3x3.

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

1	2	3
4	5	6
7	8	9

Sekarang kita bisa mengekstrak diagonal.

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

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

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

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

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

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

Sebagai contoh, fungsi `sqrt()` menghitung akar kuadrat dari semua elemen vektor atau matriks.

```
>sqrt(1:3)
```

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

Jadi, Anda dapat dengan mudah membuat tabel nilai. Ini adalah salah satu cara untuk memplot sebuah fungsi (alternatif lainnya menggunakan ekspresi).

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

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

Pada contoh berikut, kita membuat vektor nilai `t[i]` dengan jarak 0.1 dari -1 hingga 1. Kemudian kita membuat vektor nilai dari fungsi  
lateks:

```
s = t^3-t
```

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

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

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

Misalnya, vektor kolom dikalikan vektor baris diperluas 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 dengan hasil kali matriks. Hasil kali matriks dilambangkan dengan sebuah titik “.” dalam EMT.

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

Secara default, vektor baris dicetak dalam format ringkas.

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

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

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

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

```
5  
25
```

Untuk mentransposisikan matriks, kita menggunakan apostrof.

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

```
1  
2  
3  
4
```

Jadi kita dapat menghitung matriks A dikali vektor b.

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

```
30  
70
```

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

```
>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 1x1, yang berfungsi seperti bilangan real.

```
>v.v'
```

30

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

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

30

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ini adalah ringkasan aturannya.

- Sebuah 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 dua matriks memiliki dimensi yang berbeda, keduanya diperluas dengan cara yang masuk akal, sehingga memiliki ukuran yang sama.

Misalnya, nilai skalar dikalikan vektor mengalikan nilai tersebut dengan setiap elemen vektor. Atau matriks dikali vektor (dengan \*, bukan .) memperluas vektor ke ukuran matriks dengan menduplikasinya. Berikut ini adalah kasus sederhana dengan operator ^.

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

[1, 4, 9]

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

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

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

Perhatikan bahwa hasil kali skalar menggunakan hasil kali matriks, bukan tanda \*!

```
>v.v'
```

14

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

sum,prod menghitung jumlah dan hasil kali dari 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]
```

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

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

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

Untuk menggabungkan matriks dan vektor, terdapat operator “|” dan “\_”.

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

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

Elemen-elemen dari sebuah matriks disebut dengan “A[i,j]”.

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

6

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

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

6  
[7, 8, 9]

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

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

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

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

4

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

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

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

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

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

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

Sekarang kita menambahkan kolom ke matriks.

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

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

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

Pada contoh berikut, kita menghitung  $t[j]^i$  untuk  $i$  dari 1 hingga  $n$ . Kita mendapatkan sebuah matriks, di mana setiap baris adalah tabel  $t^i$  untuk satu  $i$ . Dengan kata lain, matriks tersebut memiliki elemen-elemen lateks:  $a_{\{i, j\}} = t_j^i$ ,  $\forall i \leq 1 \leq n$ ,  $\forall j \leq 1 \leq 101$ .

Sebuah fungsi yang tidak bekerja untuk input vektor harus “divektorkan”. Hal ini dapat dicapai dengan kata kunci “map” dalam definisi fungsi. Kemudian fungsi akan dievaluasi untuk setiap elemen parameter vektor.

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

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

Kata kunci “map” membuat vektor fungsi. Fungsi ini sekarang akan bekerja ntuk vektor angka.

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

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

## Sub-Matriks dan Elemen Matriks

---

Untuk mengakses elemen mmatriks, gunakan notasi kurung.

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

1	2	3
4	5	6
7	8	9
5		

Kita dapat mengakses baris lengkap dari sebuah matriks.

```
>A[2]
```

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

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

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

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

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

[4, 5, 6]

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

Di sini kita menginginkan baris pertama dan kedua dari A.

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

1	2	3
4	5	6

Kita bahkan dapat menyusun ulang A menggunakan vektor indeks. Tepatnya, kita tidak menubah A di sini tetapi menghitung versi susunan ulng dari A.

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

7	8	9
4	5	6
1	2	3

Trik indeks juga dapat digunakan pada kolom.

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

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

2	3
5	6
8	9

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

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

3
6
9

Sebagai artenatif, biarkan indeks pertama kosong.

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

2	3
5	6
8	9

Kita juga bisa mendapatkan baris terakhir A.

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

[7, 8, 9]

Sekarang mari kita ubah elemen-elemen dari A dengan memberikan sebuah submatriks dari A kesuatu nilai.

Hal ini sebenarnya mengubah matriks A yang tersimpan.

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

4	2	3
4	5	6
7	8	9

Kita juga dapat menetapkan nilai pada deretan A.

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

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

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

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

5	6	-1
7	8	6
7	8	9

Selain itu, beberapa jalan pintas diperbolehkan.

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

0	0	-1
0	0	6
7	8	9

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

```
>A[4]
```

```
Row index 4 out of bounds!
```

```
Error in:
```

```
A[4] ...
```

```
^
```

## Menyortir dan Mengacak

---

Fungsi mengurutkan () mengurutkan vektor baris.

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

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

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

Mari kita acak sebuah vektor.

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

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

Indeks berisi urutan v yang tepat.

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

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

Hal ini juga berlaku untuk vektor string.

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

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

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

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

Seperti yang anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

Fungsi unique mengembalikan daftar terurut dari elemen unik vektor.

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

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

Hal ini juga berlaku untuk vektor string.

```
>unique(s)
```

```
a  
aa  
d  
e
```

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

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

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

```
-4  
4.5
```

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

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

```
8.790745908981989e-13
```

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

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

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

Determinan dari matris ini adalah 0.

```
>det(A)
```

0

## Matriks Simbolik

---

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Untuk mengekstrak vektor eigen tertentu, diperlukan pengindeksan yang cermat.

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

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

$$[1, - 1]$$

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

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

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

Dalam ekspresi simbolik, gunakan dengan.

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

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

Akses ke baris matriks simbolik bekerja seperti halnya matriks numerik.

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

$$[1, a]$$

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

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

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

Terdapat fungsi-fungsi simbolik dalam Maxima untuk membuat vektor dan matriks. Untuk hal ini, lihat dokumentasi Maxima atau tutorial tentang Maxima di EMT.

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

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

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

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

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

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

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

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

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

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

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

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

Misalnya, nilai eigen dari A dapat dihitung secara numerik.

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

```
[16.1168, -1.11684, 0]
```

Atau secara simbolik. Lihat tutorial tentang Maxima untuk mengetahui detailnya.

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

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

Nilai Numerik dalam Ekspresi simbolik

---

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

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

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

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

```
>$&A
```

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

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

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

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

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

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

```
1.414213562373095
```

Ketepatan angka floating point yang besar dapat diubah.

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

```
3.14159265358979323846264338327950288419716939937510582097494\  
4592307816406286208998628034825342117068b0
```

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

Perhatikan bahwa hal ini hanya diperlukan, jika variabel telah didefinisikan dengan ":=" atau "=" sebuah variabel numerik.

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

```
-5.424777960769379
```

## Demo - Suku Bunga

---

Di bawah ini, kami menggunakan Euler Matlab Toolbox (EMT) untuk menghitung suku bunga. Kami melakukan secara numerik dan simbolik untuk menunjukkan kepada anda bagaimana euler dapat digunakan untuk memecahkan masalah dikehidupan nyata.

Asumsikan anda memiliki modal awal sebesar modal awal sebesar 5000(katakan dalam dolar).

```
>K=5000
```

5000

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

```
>K*1.03
```

5150

Euler juga akan memahami sintaks berikut ini.

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

5150

Tetapi akan lebih mudah untuk menggunakan faktor.

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

```
1.03  
5150
```

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

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

```
6719.58189672
```

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

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

```
6719.58
```

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

```
>"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 ke-1 hingga tahun ke-9? Untuk hal ini, bahasa matriks euler sangat membantu. Anda tidak perlu menulis perulangan, tetapi cukup memasukkan.

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

Real 1 x 11 matrix

5000.00      5150.00      5304.50      5463.64      ...

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

```
>short 0:10
```

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

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

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

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

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

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

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

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

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

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

```
1271.61  
1271.6071
```

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

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

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

```
Real 1 x 11 matrix
```

```
5000.00      5150.00      5304.50      5463.64      ...
```

Kita bisa mencetaknya dengan cara yang bersahabat, menggunakan format kita dengan angka desimal yang tetap.

```
>VKr'
```

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

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

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

```
5150.00  
5000.00      5150.00      5304.50
```

Yang mengejutkan, kita juga dapat menggunakan vektor indeks. Ingatlah bahwa 1:3 menghasilkan vektor [1,2,3].

Mari kita bandingkan elemen terakhir dari nilai yang dibulatkan dengan nilai penuh.

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

```
6719.60  
6719.58
```

Perbedaannya sangat kecil.

## Menyelesaikan Persamaan

---

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

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

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

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

Real 1 x 11 matrix

5000.00	5350.00	5710.50	6081.82	...
---------	---------	---------	---------	-----

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

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

Real 1 x 11 matrix

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

Kami melihat bahwa uangnya berkurang. Jelas, jika kita hanya mendapatkan 150 bunga di tahun pertama, tetapi menghapus 200, kita kehilangan uang setiap tahun.

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

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

Real 1 x 51 matrix

```
5000.00      4950.00      4898.50      4845.45      ...
```

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

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

48.00

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

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

Fungsi iterate() memiliki satu trik lagi. Fungsi ini dapat menerima 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. Anggaplah kita tahu bahwa nilainya adalah 0 setelah 50 tahun. Berapakah tingkat suku bunganya?

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

Langkah pertama adalah mendefinisikan sebuah fungsi yang melakukan iterasi sebanyak n kali. Kita tambahkan semua parameter ke fungsi ini.

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

Perulangannya sama seperti di atas

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

Tetapi kita tidak lagi menggunakan nilai global R dalam ekspresi kita. Fungsi-fungsi seperti iterate() memiliki trik khusus dalam Euler. Anda bisa mengoper nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R.

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

Mari kita coba sebuah tes.

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

-19.83

Sekarang kita bisa menyelesaikan masalah kita.

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

3.15

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

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

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

-336.08

Perhatikan bahwa Anda tidak dapat menyelesaikan jumlah tahun, karena fungsi kita mengasumsikan n sebagai nilai bilangan bulat.

## Solusi Simbolik untuk Masalah Suku Bunga

---

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

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

$$R + q K$$

Sekarang kita bisa mengulangi hal ini.

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

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

Kita melihat sebuah pola. Setelah n periode, kita memiliki

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

Rumus tersebut adalah rumus untuk jumlah geometris, yang dikenal dengan Maxima.

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

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

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

Mari kita buat sebuah fungsi untuk ini.

```
>function fs(K,R,P,n) &= (1+P/100)^n*K + ((1+P/100)^n-1)/(P/100)*R; $&fs(K,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 kita sebelumnya. Tetapi fungsi ini lebih efektif.

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

```
-19.82504734650985  
-19.82504734652684
```

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

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

```
20.51
```

Jawaban ini mengatakan bahwa nilai tersebut akan menjadi negatif setelah 21 tahun.

Kita juga bisa menggunakan sisi simbolis dari Euler untuk menghitung rumus pembayaran.

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

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

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

Biasanya rumus ini diberikan dalam bentuk

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

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

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

Kita dapat menyelesaikan laju R secara simbolis.

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

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

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

Tentu saja, kita memiliki batas berikut.

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

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

Jelasnya, tanpa bunga kita harus membayar kembali 10 suku bunga 500.

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

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

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

### Contoh Penyelesaian Soal-Soal Aljabar

---

## R.2 Exercise Set

---

Write an equivalent expression without negative exponents

$$49. \left( \frac{24a^{10}b^{-8}c^7}{12a^6b^{-3}c^5} \right)^{-5}$$

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

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

Calculate.

$$90. (2^6 \times 2^{-3} \div 2^{10} \div 2^{-8})$$

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

2

Syntesis

Saving plan. The formula

$$S = P \left( \frac{(1 + \frac{r}{12})^{12t} - 1}{\frac{r}{12}} \right)$$

gives the amount S accumulated in a savings plan when a deposit of P dollars is made each month for t years in an account with interest rate r, compounded monthly.

```
>function savingPlan(deposit, years, interestRate) ...
```

```
    return deposit * ((1 + (years / 12) ^ (12 * years)) - 1 / (interestRate / 12))
  endfunction
```

97. James deposits \$250 in a retirement account each month beginning at age 40. If the investment earns 5% interest, compounded monthly, how much will have accumulated in the account when he retires 27 years later?

```
>result := savingPlan(250, 27, 0.05)
```

31994540460589789358826662710200054727006612566029477411600158184622621262418936313231860997654656

```
>&round(result)
```

```
round(result)
```

98. Kayla deposits \$100 in a retirement account each month beginning at age 25. If the investment earns 4% interest, compounded monthly, how much will have accumulated in the account when she retires at age 65?

```
>savingPlan(100, 65-25, 0.04)
```

95895391064934425636904631683687745180441308685905596046473504061898973164769632734224476061086759

100. Lamont wants to have \$200,000 accumulated in a retirement account by age 70. If he starts making monthly deposits to the plan at age 30 and can count on an interest rate of 4.5%, compounded monthly, how much should he deposit each month in order to accomplish this?

```
>savingPlan(200000, 70 - 30, 0.045)
```

19179078212986883147814141575530811418632261366807710681944573874844559515314888679396340235107830

Simplify. Assume that all exponents are integers, all denominators are nonzero, and zero is not raised to a nonpositive power

$$104. ((m^{x-b} n^{x+b})^x (m^b n^{-b})^x)$$

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

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

### R.3 Exercise Set

---

Determine the terms and the degree of the polynomials

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

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

$$-3z + 2y + 4x - 3$$

$$8.(2x^2 + 12xy - 11) + (6x^2 - 2x + 4) + (-x^2 - y - 2)$$

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

$$12xy - y + 7x^2 - 2x - 9$$

```
>
```

$$9.(3x^2 - 2x - x^3 + 2) - (5x^2 - 8x - x^3 + 4)$$

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

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

```
>
```

31.  $(5x - 3)^2$

```
>${&expand((5*x - 3)^2)}
```

$$25x^2 - 30x + 9$$

```
>
```

Systhesis

Multiply. Assume that all exponents are natural numbers.

55.  $(a + b + c)^2$

```
>${&expand((a + b + c)^2)}
```

$$c^2 + 2bc + 2ac + b^2 + 2ab + a^2$$

---

#### R.4 Exercise Set

Factor the difference of squares

$$47.z^2 - 81$$

```
>${&factor(z^2 - 81)}
```

$$(z - 9) (z + 9)$$

Factor the square of a binomial.

$$66. (5a^2 - 10ab + 5b^2)$$

```
>${&factor(5*a^2 - 10*a*b + 5*b^2)}
```

$$5 (b - a)^2$$

Factor the sum or difference of cubes.

$$75. (t^6 + 1)$$

```
>${&factor(t^6 + 1)}
```

$$(t^2 + 1) (t^4 - t^2 + 1)$$

Factor completely.

$$78. (4x^2 + 12xy^2)$$

```
>${&factor(4*x^2 + 12*x*y^2)}
```

$$4x (3y^2 + x)$$

Synthesis  
Factor.

$$130. (x + 0.01)^2 - x^2$$

```
>${&factor((x + 0.01)^2 - x^2)}
```

$$\frac{200x + 1}{10000}$$

Factor. Assume that variables in exponents represent natural numbers.

$$139. (y - 1)^4 - (y - 1)^2$$

```
>${&factor((y - 1)^4 - (y - 1)^2)}
```

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

---

## R.5 Exercise Set

$$31.7(3x + 6) = 11 - (x + 2)$$

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

$$\begin{aligned}x &= -\frac{3}{2} \\36.y^2 - 4y - 45 &= 0\end{aligned}$$

```
>$&solve(y^2 - 4*y - 45 = 0, y)
```

$$[y = 9, y = -5]$$

$$65.A = \frac{1}{2}h(b_1 + b_2)$$

for: h (Area of a trapezoid)

```
>$&(solve(A = (1/2)* h * (b1 + b2), h))
```

$$\begin{aligned}\left[ \begin{pmatrix} \frac{2-(b_2+b_1)h}{2} & \frac{80143857-(12755291b_2+12755291b_1)h}{\frac{25510582}{10-(b_2+b_1)}h} \end{pmatrix} = 0 \right] \\71.Ax + By = C\end{aligned}$$

for A

```
>${&}solve(A*x + By = C, A)
```

□

Synthesis  
Solve

$$81.3[5 - 3(4 - t)] - 2 = 5[3(5t - 4) + 8] - 26$$

```
>${&}solve(3*(5 - 3*(4 - t)) - 2 = 5*(3*(5 * t - 4) + 8) - 26, t)
```

$$\left[ t = \frac{23}{66} \right]$$

$$87.3x^3 + 6x^2 - 27x - 54 = 0$$

```
>${&}solve(3*x^3 + 6*x^2 - 27*x - 54 = 0, x)
```

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

## R.6 Exercise Set

---

$$11). \frac{x^3 - 6x^2 + 9x}{x^3 - 3x^2}$$

Penyelesaian :  
mencari faktor pembilang

```
> $&factor(x^3-6*x^2+9*x)
```

$$(x - 3)^2 \ x$$

mencari faktor penyebut

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

$$(x - 3) \ x^2$$

jadi tugas kita menyelesaikan :

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

```
>${&((x-3)^2 * x)/((x-3) * x^2)}
```

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

atau dengan perintah EMT untuk menyelesaikan soal ini adalah

```
>${ratsimp((x^3-6*x^2+9*x)/(x^3-3*x^2))} ...
>/menyederhanakan pecahan dengan perintah ratsimp
```

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

17. Multiply or divide and, if possible, simplify

$$\frac{r-s}{r+s} \cdot \frac{r^2 - s^2}{(r-s)^2}$$

Penyelesaian:

```
>${ratsimp((r-s)/(r+s))*((r^2-s^2)/((r-s)^2)) ...
>/menyederhanakan pecahan
```

$$\frac{r^2 - s^2}{(r - s) (s + r)}$$

```
>${(r^2-s^2)/(r-s)*(s+r), ${ratsimp(%)} ...
>/menyederhanakan pecahan lalu hasilnya disederhanakan lagi pakai %
```

$$s^2 + 2 r s + r^2$$

$$28 \cdot \frac{(c^3 + 8)}{(c^2 - 4)} : \frac{(c^2 - 2c + 4)}{(c^2 - 4c + 4)}$$

Penyelesaian:

```
>${ratsimp((c^3+8)/(c^2-4))/((c^2-2*c+4)/(c^2-4*c+4)), ratsimp(%)} ...
>/menyederhanakan pecahan lalu hasilnya disederhanakan lagi pakai %
```

$$c - 2$$

40. add or subtract and, if possible, simplify

$$\frac{6}{y^2 + 6y + 9} - \frac{5}{y - 3}$$

Penyelesaian:

```
>${expand(6/y^2+6*y+9)-(5/y-3), ratsimp(%)} ...
>/menyederhanakan pecahan lalu melakukan operasi pada hasilnya
```

$$\frac{6y^3 + 12y^2 - 5y + 6}{y^2}$$

$$\frac{6y^3 + 12y^2 - 5y + 6}{y^2}$$

62. Simplify

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

Penyelesaian:

```
>${&factor((a^2/b)+(b^2/a))/(a^2-a*b+b^2) ...  
>/menyederhanakan bentuk paling sederhana dengan perintah factor
```

$$\frac{b+a}{ab}$$

## R.7 Exercise Set

---

53. Simplify. Assume that no radicands were formed by raising negative quantities to even powers.

$$(2\sqrt{3} + \sqrt{5}) \cdot (\sqrt{3} - 3\sqrt{5})$$

Penyelesaian:

$$> \$\&expand(2*(3)^{1/2} + (5)^{1/2}*((3)^{1/2} - 3*(5)^{1/2})$$

-33

## Review Exercise

---

66. the cost of a house is \$98,000. The down payment is %16,000, the interest rate is 6 1/2%, and the loan period is 25 years. What is the monthly mortgage payment?

Penyelesaian :

Rumus angsuran per bulan:

$$M = P \cdot \frac{\frac{r}{12} \cdot (1 + \frac{r}{12})^2}{(1 + \frac{r}{12})^2 - 1}$$

dengan M : angsuran per bulan

P : jumlah pinjaman

r : suku bunga bulanan

n : jumlah total pembayaran (dalam bulan)

diketahui:

Harga rumah : \$98,000

Uang muka : \$16,000

suku bunga tahunan:  $6 \frac{1}{2}\%$

$6 \frac{1}{2}\%$  diubah menjadi 6.5%

lama angsur : 25 tahun

ditanya: angsuran per bulan berapa

Variabel yang dibutuhkan pada rumus adalah :

$$>P = 98000 - 16000$$

$$82000.00$$

$$>r = 6.5 / 100$$

$$0.07$$

$$>n = 25 * 12$$

$$300.00$$

Substitusi semua variabel yang telah diperoleh ke rumus M

```
>M = P * (r/12 * (1 + r/12)^n)/((1+r/12)^n -1 )
```

$$553.67$$

Jadi, biaya angsuran per bulan selama 25 tahun dengan suku bunga sebesar 6.5% persen per tahun adalah \$53,669872305

$$70.(x^n + 10)(x^n - 4)$$

Penyelesaian:

```
>${\&expand((x^n + 10)*(x^n - 4))} ...
>/menjabarkan fungsi
```

$$x^{2n} + 6x^n - 40$$

$$71.(t^a + t^{-a})^2$$

Penyelesaian:

```
>${&expand((t^a + t^{(-a)})^2) ...  
>/menjabarkan fungsi
```

$$t^{2a} + \frac{1}{t^{2a}} + 2$$

73. $(a^n - b^n)^3$

```
>${&showev('expand((a^n - b^n)^3)), ${&expand((a^n - b^n)^3) ...  
>/menjabarkan fungsi dengan showev
```

$$-b^{3n} + 3a^n b^{2n} - 3a^{2n} b^n + a^{3n}$$

---

### 2.3 Exercise Set

Given that

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

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

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

Find each of the following

Menyimpan semua nilai fx, gx, dan hx.

```
>fx &= 3*x + 1; gx &= x^2 - 2*x - 6; hx &= x^3;
```

1. Mencari nilai

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

Penyelesaian:

fungsi komposisi di atas juga dapat ditulis sebagai :

$$(f(g(-1)))$$

Mencari nilai  $g(-1)$

```
>gx(-1) //cara manual
```

-3.00

Lalu mencari nilai  $f(-3)$

```
>fx(-3)
```

-8.00

Perintah fungsi komposisi di EMT

Akan menghasilkan hasil yang sama dengan cara manual di atas. Cara ini lebih efisien.

```
>fx(gx(-1)) //cara cepat
```

-8.00

4. Mencari nilai

$$(g \circ h)\left(\frac{1}{2}\right)$$

Penyelesaian:

Fungsi komposisi di atas juga dapat ditulis

$$g\left(h\left(\frac{1}{2}\right)\right)$$

```
>gx(hx(1/2))
```

-6.23

7. Mencari nilai

$$(f \circ h)(-3)$$

Penyelesaian:

Fungsi komposisi di atas bisa ditulis

$$f(g(-3))$$

```
>fx(gx(-3))
```

28.00

11. Mencari nilai

$$(h \circ h)(2)$$

Penyelesaian

Fungsi komposisi di atas dapat ditulis

$$(h(h(2)))$$

```
>hx(hx(2))
```

512.00

13. Mencari nilai

$$(f \circ f)(-4)$$

Penyelesaian:

Fungsi komposisi di atas dapat ditulis

$$f(f(-4))$$

```
>fx(fx(-4))
```

-32.00

### 3.1 Exercise Set

---

11. Simplify. Where answer in the form  $a + bi$ , where  $a$  and  $b$  are real numbers.

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

Penyelesaian:

```
>bilangan1 = -5 + 3*I, bilangan2 = 7 + 8*I, bilangan1 + bilangan2
```

$$\begin{aligned} &-5.00+3.00i \\ &7.00+8.00i \\ &2.00+11.00i \end{aligned}$$

$$21.(10 + 7i) - (5 + 3i)$$

Penyelesaian:

```
>bilangan1 = 10 + 7*I, bilangan2 = 5 + 3*I, bilangan1 - bilangan2
```

```
10.00+7.00i  
5.00+3.00i  
5.00+4.00i
```

$$35.7i \cdot (2 - 5i)$$

Penyelesaian:

```
>bilangan1 = 7*I, bilangan2 = 2-5*I, bilangan1*bilangan2
```

```
0.00+7.00i  
2.00-5.00i  
35.00+14.00i
```

$$36.3i \cdot (6 + 4i)$$

Penyelesaian:

```
>bilangan1 = 3*I, bilangan2 = 6 + 4*I, bilangan1*bilangan2
```

```
0.00+3.00i  
6.00+4.00i  
-12.00+18.00i
```

$$37. - 2i \cdot (-8 + 3i)$$

Penyelesaian:

```
>bilangan1 = -2*I, bilangan2 = -8 + 3*I, bilangan1*bilangan2
```

```
-0.00-2.00i  
-8.00+3.00i  
6.00+16.00i
```

---

### 3.4 Exercise Set

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

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

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

$$3).\frac{x+2}{4} - \frac{x-1}{5} = 15$$

```
> $&solve(((x+2)/4)-((x-1)/5)=15)
```

$$[x = 286]$$

$$4).\frac{t+1}{3} - \frac{t-1}{2} = 1$$

```
> $&solve(((t+1)/3)-((t-1)/2))
```

$$[t = 5]$$

$$7) \cdot \frac{5}{3x+2} = \frac{3}{2x}$$

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

$$\left[ x = \frac{-\sqrt{11} - 1}{3}, x = \frac{\sqrt{11} - 1}{3} \right]$$

$$29) \cdot \sqrt{3x - 4} = 1$$

```
> $&solve((3*x-4)^{1/2} = 1)
```

$$[x = 2]$$

---

### 3.5 Exercise Set

$$23). |x + 3| - 2 = 8$$

```
> $&solve(abs(x+3)-2=8, x)
```

$$[|x + 3| = 10]$$

```
> $&solve(abs(x+3))
```

$$[x = -3]$$

```
> $&solve(abs(x+3)-2=8)
```

$$[|x + 3| = 10]$$

```
> $&solve((abs(x+3))=10)
```

$$[|x + 3| = 10]$$

```
>${&expand(abs(x+3)=10)}
```

$$|x + 3| = 10$$

```
>${&solve((x+3)=10)}
```

$$[x = 7]$$

```
>${&solve((x+3)=-10)}
```

$$[x = -13]$$

$$28). |5x + 4| + 2 = 5$$

```
>${&solve(abs(5*x+4)+2=5, x)}
```

$$[|5x + 4| = 3]$$

```
>$&solve((5*x+4)=3)
```

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

```
>$&solve((5*x+4)=-3)
```

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

$$32).5 - |4x + 3| = 2$$

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

$$[|4x + 3| = 2]$$

```
>$&solve((4*x+3)=2)
```

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

```
>$$&solve((4*x+3)=-2)
```

$$\left[ x = -\frac{5}{4} \right]$$

43).  $|2x| \geq 6$

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

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

```
>$$&fourier_elim([abs(2*x)] >= 6, [x])
```

$[[2(|x| - 3)] = 0] \vee [[2(|x| - 3)] > 0]$

```
>$$&fourier_elim([abs(2*x)] >= 6, [x])
```

$[[2(|x| - 3)] = 0] \vee [[2(|x| - 3)] > 0]$

## Chapter 3 test

---

$$7).x + 5\sqrt{x} - 36 = 0$$

```
>$&solve(x+5*(x)^1/2-36=0)
```

$$\left[ x = \frac{72}{7} \right]$$

$$10).\sqrt{x+4} - \sqrt{x-4} = 2$$

```
>$&solve(((x+4)^1/2)-((x-4)^1/2=2))
```

$$[x = 8]$$

$$9).\sqrt{x+4}-2=1$$

```
>${&}solve((x+4)^{1/2}-2=1)
```

$$[x = 2]$$

$$13). |x + 3| \leq 4$$

```
>${&}fourier_elim([abs(x+3)<=4], [x])
```

$$[x = 1] \vee [x = -7] \vee [-7 < x, x < 1]$$

$$14). |2x - 1| < 5$$

```
>${&}fourier_elim([abs(2*x-1)<5], [x])
```

$$[-2 < x, x < 3]$$

## 4.1 Exercise Set

---

$$36). f(x) = (x^2 - 5x + 6)^2$$

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

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

$$37). f(x) = x^4 - 4x^2 + 3$$

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

$$\left[ x = -1, x = 1, x = -\sqrt{3}, x = \sqrt{3} \right]$$

$$40).f(x)=x^3-x^2-2x+2$$

```
>${&}solve(x^3-x^2-2*x+2)
```

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

$$61).2y-3\geq 1-y+5$$

```
>${&}fourier\_elim([2*y-3>=1-y+5],[x])
```

$$[y-3] \vee [y-3 > 0]$$

$$64).\left|x+\frac{1}{4}\right|\leq\frac{2}{3}$$

```
>${&}fourier\_elim([x+(1/4)<=(2/3)], [x])
```

$$\left[x=\frac{5}{12}\right]\vee\left[x<\frac{5}{12}\right]$$

## Mid-Chapter Mixed Review

---

$$18).g(x) = x^3 - 9x^2 + 4x - 10;$$

find g(-5)

```
>function g(x) := (x^3-9x^2+4x-10)
>g(-5)
```

-380.00

$$20).f(x) = 5x^4 + x^3 - x;$$

find

$$f(-\sqrt{-2})$$

```
>a = - sqrt(2)
```

$$-1.41$$

```
>fx &= 5*x^4 + x^3 - x; fx(a)
```

$$18.59$$

$$17) \cdot \frac{x^5 - 5}{x + 1}$$

```
>$&expand((x^5-5)/(x+1))
```

$$\frac{x^5}{x + 1} - \frac{5}{x + 1}$$

$$19). f(x) = 20x^2 - 40x;$$

find

$$f\left(\frac{1}{2}\right)$$

```
>function f(x) :=(20*x^2-40*x)
>f(1/2)
```

-15.00

$$23).h(x) = x^3 - 2x^2 - 55x + 56$$

Penyelesaian :

```
>${&factor(x^3-2*x^2-55*x+56)}
```

$$(x - 8) (x - 1) (x + 7)$$

```
>${&solve(x^3-2*x^2-55*x+56)}
```

$$[x = -7, x = 8, x = 1]$$

$$24).g(x) = x^4 - 2x^3 - 13x^2 + 14x + 24$$

Penyelesaian :

```
> $&factor(x^4-2*x^3-13*x^2+14*x+24)
```

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

```
> $&solve(x^4-2*x^3-13*x^2+14*x+24)
```

$$[x = -3, x = -1, x = 2, x = 4]$$

---

### 4.3 Exercise Set

$$f(x) = x^2 + 4x^2 + x - 6$$

penyelesaian :

```
>$&solve(x^3+4*x^2+x-6)
```

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

50). $f(x) = x^4 + x^3 - 3x^2 - 5x - 2$

Penyelesaian

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

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

51). $f(x) = x^3 - 7x + 6$

Penyelesaian

```
>$&solve(x^3-7*x+6)
```

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

$$52). f(x) = x^3 - 12x + 16$$

Penyelesaian :

```
>$&solve(x^3-12*x+16)
```

$$[x = -4, x = 2]$$

$$53). f(x) = -x^3 + 3x^2 + 6x - 8$$

```
>$&solve(-x^3+3*x^2+6*x-8)
```

$$[x = -2, x = 4, x = 1]$$

## Chapter 2

### Plot 2D dengan EMT

# 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).

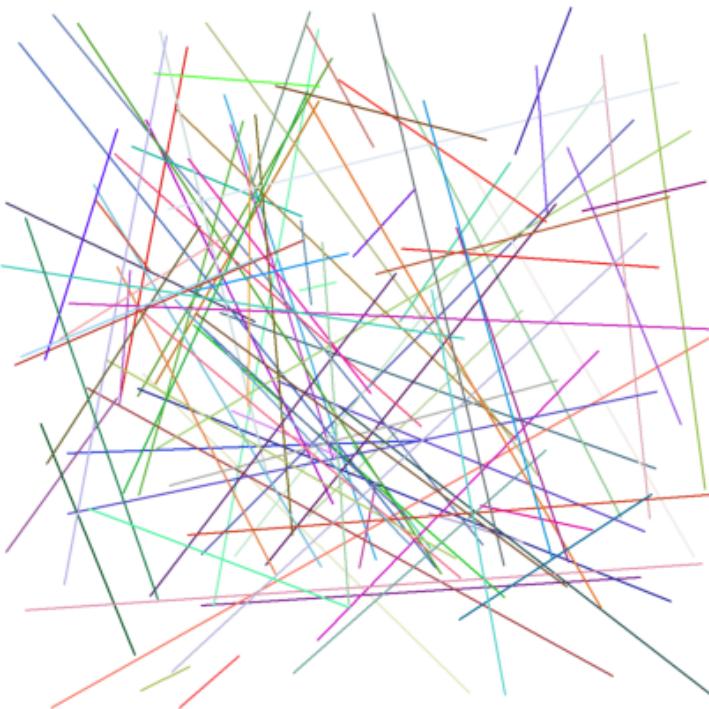
## Plot-Plot Dasar

---

Ini adalah fungsi-fungsi plot yang sangat umum. Terdapat koordinat-koordinat layar, yang mana selalu berupa selang dari 0 hingga 1024 dalam setiap sumbu, tidak peduli apakah layarnya yang berupa persegi atau tidak. Dan disini adalah koordinat-koordinat plot, yang mana dapat diatur dengan `setplot()`. Pemetaan antara koordinat-koordinat tergantung pada jendela layar. Sebagai contoh, bawaan `shirnkwindow()` akan mengatur jarak untuk sumbu label-label dan sebuah judul plot.

Dalam contoh, kami hanya akan menggambar beberapa garis-garis acak dalam warna-warna yang berbeda. Untuk lebih jelasnya dalam fungsi-fungsi tersebut, pelajari fungsi-fungsi utama dalam EMT.

```
>clc; // membersikan layar
>window(0, 0, 1024, 1024); // menggunakan semua rentang dalam jendela
>setplot(0, 1, 0, 1); // atur koordinat-koordinat plot
>hold on; // mulai dengan mode overwrite
>n = 100; X = random(n, 2); Y = random(n,2); // membuat titik-titik acak
>colors = rgb(random(n), random(n), random(n)); // mendapatkan warna-warna acak
>loop 1 to n; color(colors[#]); plot(X[#], Y[#]); end; // membuat plot
>hold off; // mengakhiri mode overwrite
>insimg; // menampilkan gambar ke notebook
```



```
>reset;
```

Ini diperlukan untuk menahan gambar-gambar, karena perintah plot() akan membersihkan layar plot.

Untuk memembersihkan semuanya, 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.

Untuk contoh lainnya, kami menggambar sebuah plot sebagai sisipan dalam plot lainnya. Ini dapat dilakukan dengan mendefinisikan sebuah layar plot yang lebih kecil. Catatan bahwa layar ini tidak menyediakan ruang untuk label-label sumbu yang diluar dari layar plot. Kami juga dapat menambahkan beberapa margin untuk ini apabila diperlukan. Catatan bahwa kami menyimpan dan mengembalikan layar secara penuh, dan menahan plot sekarang sementara kami mensisipkan plot.

```
>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);
```

Sebuah plot dengan gambar-gambar yang banyak dapat diperoleh dengan cara yang sama. Terdapat alat fungsi figure() untuk melakukan ini.

Plot bawaan menggunakan sebuah layar plot persegi. Kamu dapat mengubahnya dengan fungsi aspect(). Jangan lupa untuk mengatur ulang aspect nanti. Kamu juga dapat mengubah bawaan dalam menu dengan "Set Aspect" untuk sebuah rasio aspect secara spesific atau untuk ukuran sekarang dari layar gambar.

Tapi kamu dapat merubahnya juga untuk satu plot. Untuk melakukan ini, ukuran sekarang dari luas plot dapat berubah, dan layar diatur jadi label-label memiliki ruang yang cukup.

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

Fungsi reset() mengembalikan plot ke pengaturan bawaan termasuk rasio aspek.

## Plot-plot 2D dalam Euler

---

EMT Math Toolbox memiliki plot-plot dalam 2D, dua-duanya untuk data dan fungsi-fungsi. EMT menggunakan fungsi plot2d. Fungsi ini dapat mem-plotkan fungsi-fungsi dan data.

Sangat mungkin untuk mem-plotkan dalam Maxima menggunakan Gnuplot atau dalam Python menggunakan Math Plot Lib.

Euler dapat mem-plot plot-plot 2D dari

- ekspresi,
- fungsi-fungsi, variabel-variabel, atau kurva-kurva yang terparameter.
- vektor-vektor dari nilai-nilai x-y.
- Titik-titik dalam bidang.
- Kurva-kurva implisit dengan level-level atau level regional.
- Fungsi kompleks

Gaya plot termasuk gaya-gaya yang bervariasi untuk garis-garis dan titik-titik, plot-plot batang dan plot-plot terasir.

## Plot Ekspresi atau Variabel

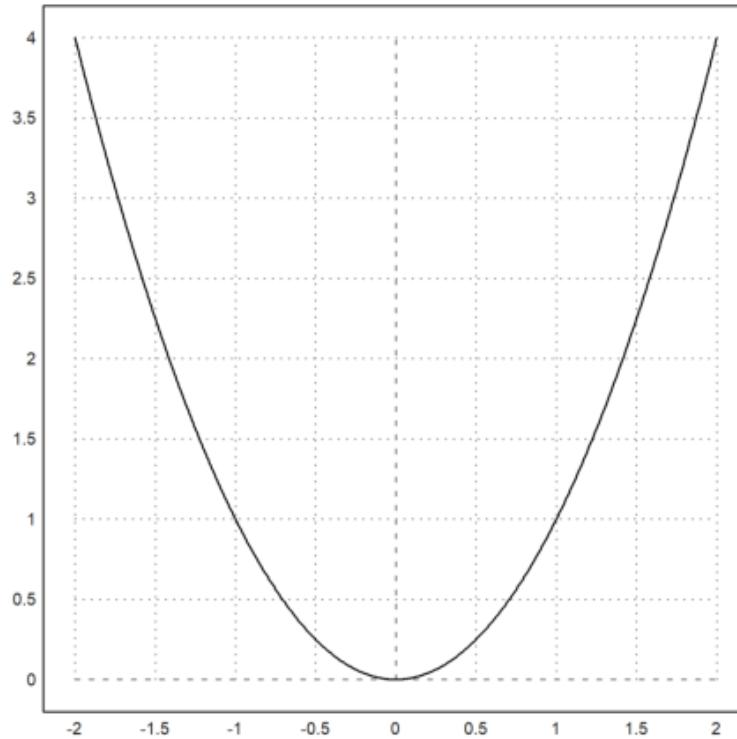
---

Ekspresi tunggal dalam "x" (seperti "4\*x^2") atau nama sebuah fungsi (seperti "f") menghasilkan sebuah grafik dari fungsi.

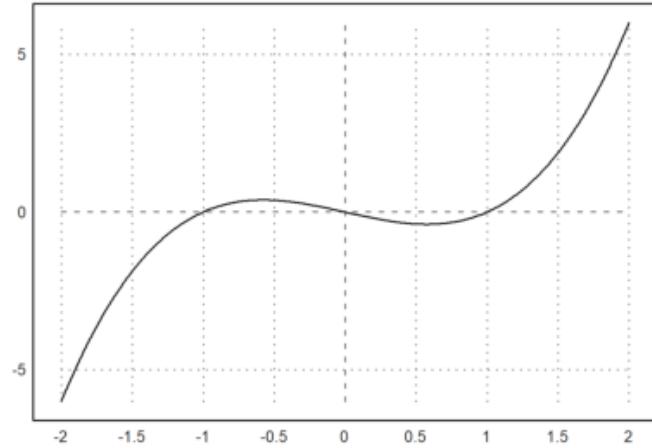
Ini adalah contoh umum, yang aman menggunakan selang bawaan dan himpunan-himpunan sesuai selang-y untuk menyesuaikan plot dari fungsi.

Catatan: Jika kamu mengakhiri sebuah baris perintah dengan sebuah titik dua ":", plot akan dimasukkan kedalam layar teks. Sebaliknya, tekan TAB untuk melihat plot jika layar plot tertutupi.

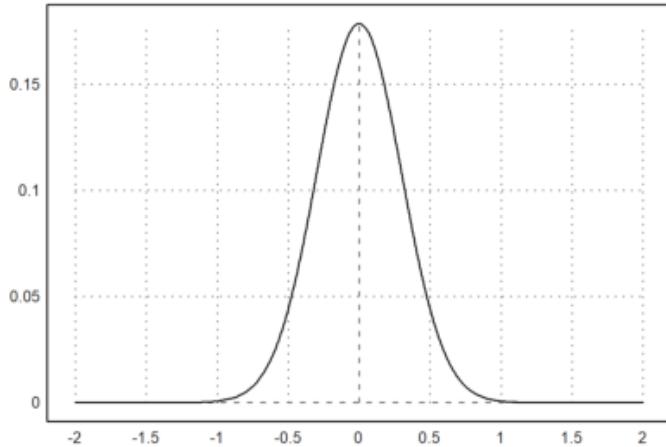
```
>plot2d("x^2"):
```



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



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

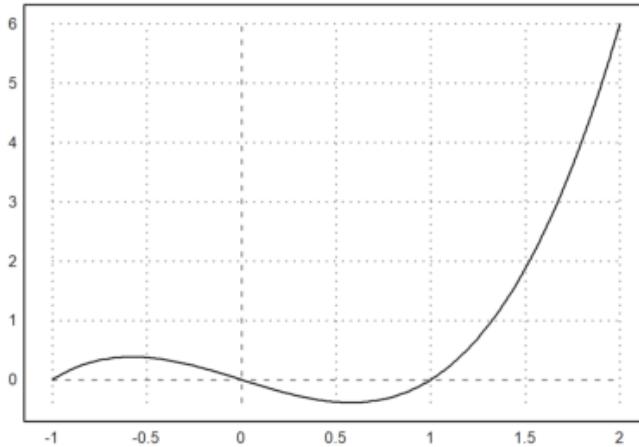


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

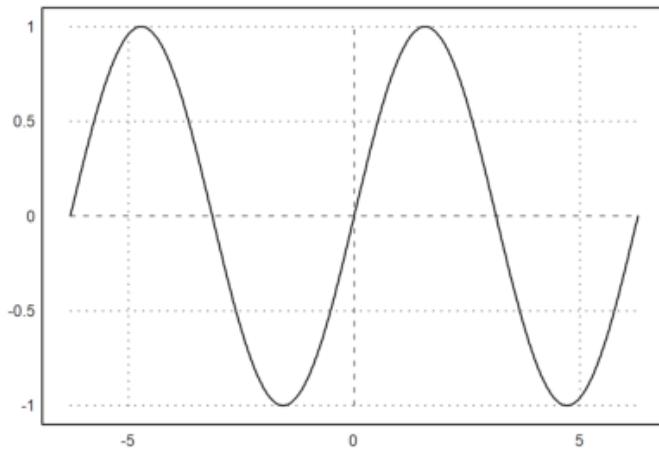
Selang dari plot diatur mengikuti parameter-parameter yang ditujukan

- a,b: x-range (bawaan -2,2)
- c,d: y-range (bawaan: skala dengan nilai-nilai)
- r: secara alternatif sebuah jari-jari mengelilingi plot lingkaran
- cx,cy: koordinat-koordinat dari tengah plot (bawaan 0,0)

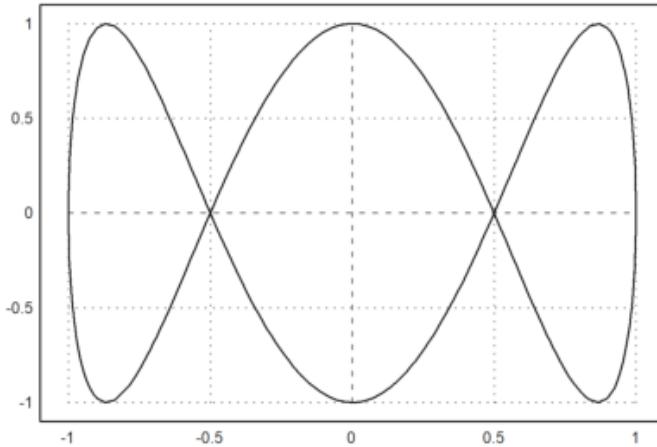
```
>plot2d("x^3-x",-1,2):
```



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



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



Sebuah alternatif untuk titik dua adalah perintah `insimg(lines)`, yang mana memasukkan plot-plot yang mempati sebuah angka yang spesifik dari baris-baris teks.

Dalam pilihan-pilihan, plot-plot dapat diatur untuk dimunculkan

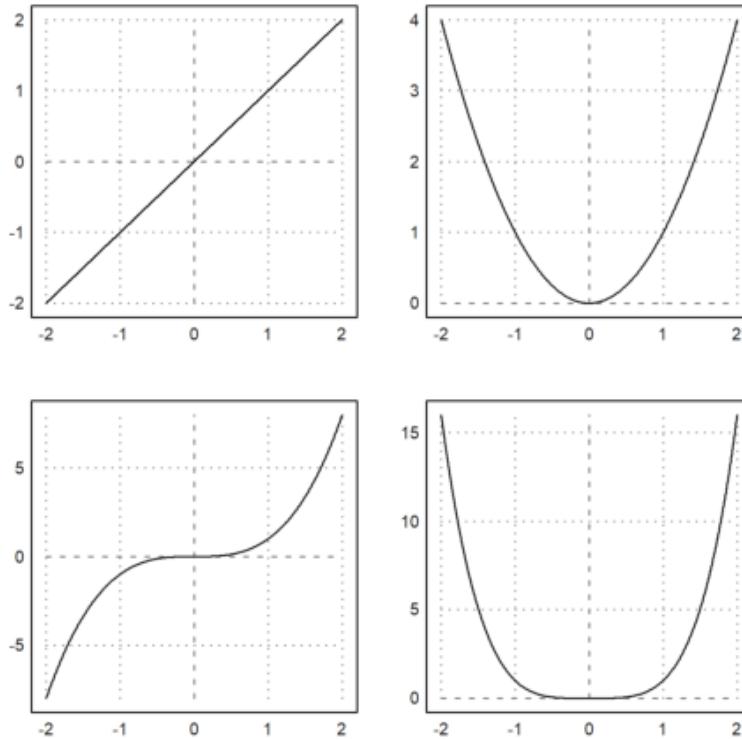
- dalam sebuah jendela yang dapat di ubah ukurannya secara terpisah,
- dalam jendela notebook.

Gaya-gaya lainnya yang dapat diperoleh dengan perintah-perintah plot yang spesifik

Dalam kasus yang lainnya, tekan tabulator untuk melihat plot, jika tersembunyi.

Untuk memisahkan jendela menjadi beberapa plot-plot, gunakan perintah `figure()`. Dalam contoh, kami membuat plot  $x^1$  ke  $x^4$  kedalam 4 bagian dari jendela. `figure(0)` mengatur ulang jendela bawaan.

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

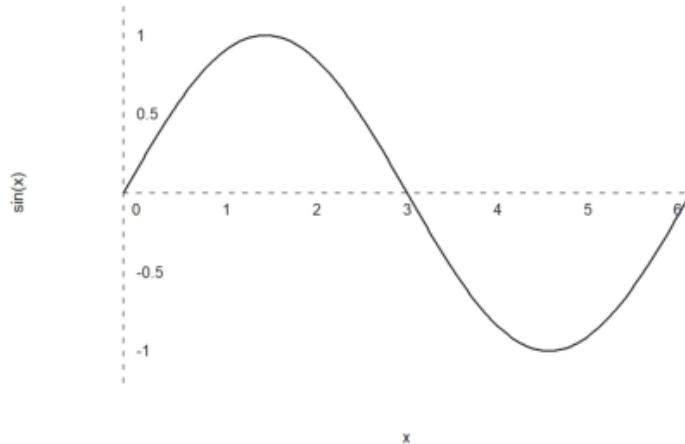


Jika dalam argumen-argumen untuk `plot2d()` adalah sebuah ekspresi yang diikuti oleh empat angka, ini angka-angka x- dan y-selang untuk plot.

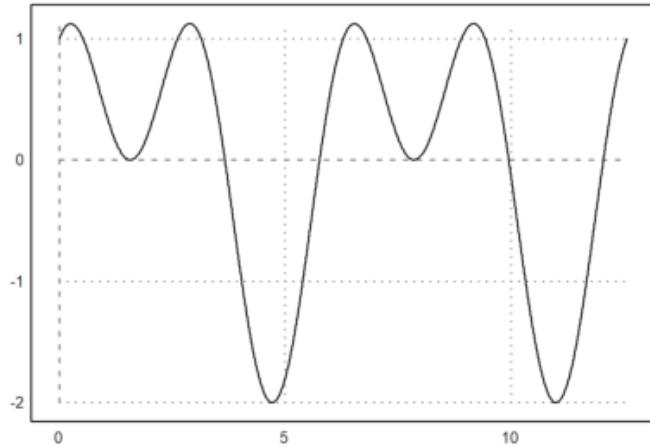
Sebagai alternatif, a, b, c, d dapat secara spesifik sebagai parameter-parameter yang ditujukan seperti `a = ...` dan lain-lain.

Seperti contoh berikut, kami mengubah gaya grid, menambah label-label, dan menggunakan label-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):
```

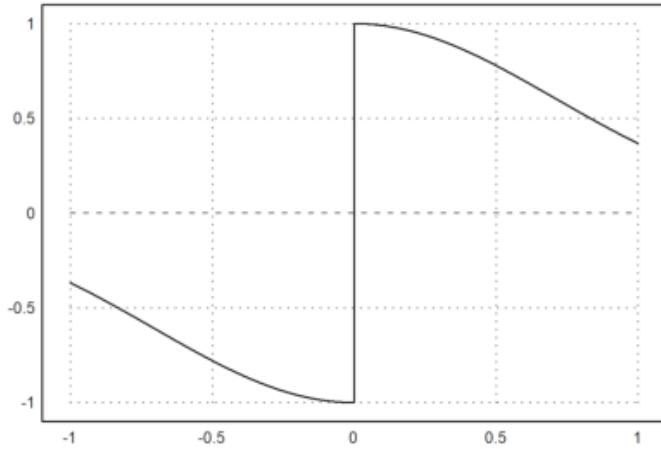


Gambar-gambar yang dihasilkan dengan memasukkan plot kedalam layar teks disimpan kedalam direktori seperti notebook, secara bawaaan dalam sebuah sub-direktori bernama "images". Mereka juga menggunakan dengan ekspor HTML.

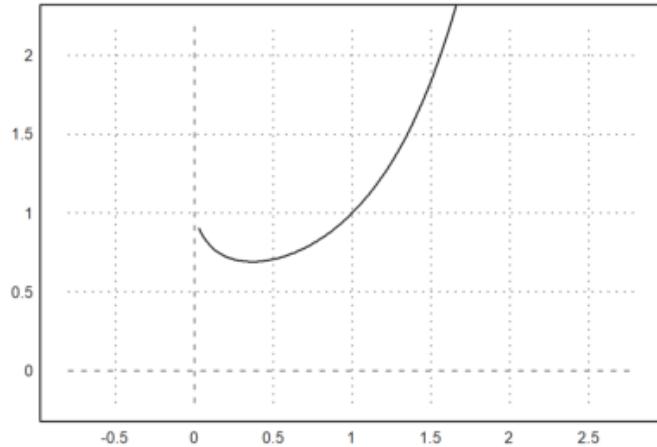
Kamu dapat secara mudah membuat tanda pada sebarang gambar dan mencetaknya kedalam clipboard dengan `Ctrl + c`. Tentu saja, kamu dapat juga mengekspor grafik-grafik sekarang dengan fungsi-fungsi dalam menu File.

Fungsi atau ekspresi dalam `plot2d` dapat dievaluasi secara adaptif. Untuk kecepatan yang lebih, matikan `plot2d` adaptif dengan `<adaptive` dan spesifik angka dalam sub-interval dengan `n = ...`. Ini seharusnya dibutuhkan dalam kasus yang sangat jarang.

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

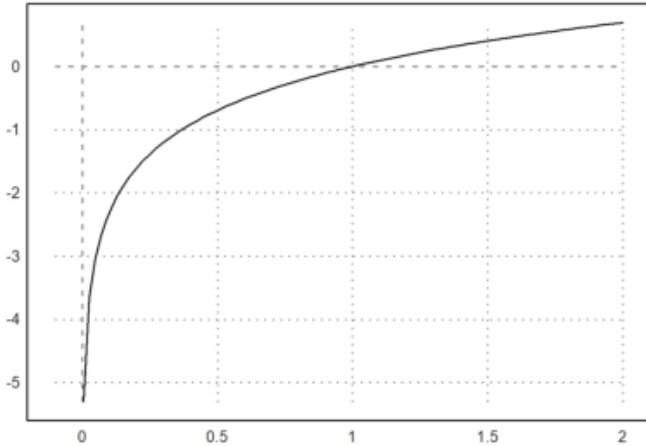


```
>plot2d("x^x", r = 1.2, cx = 1, cy = 1):
```



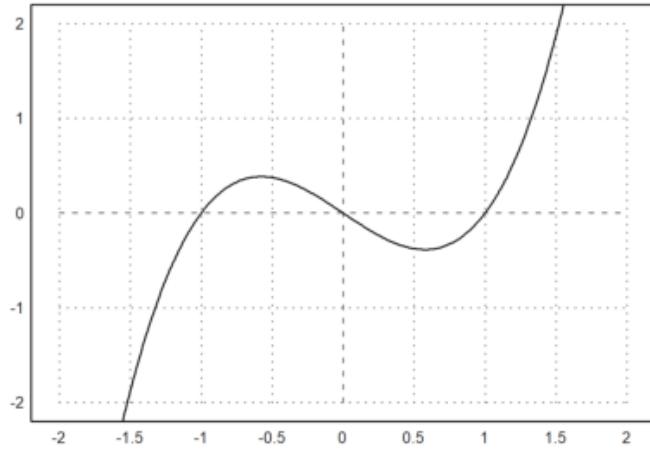
Catatan bahwa  $x^x$  tidak terdefinisi untuk  $x \leq 0$ . Fungsi plot2d mengetahui error ini, dan mulai membuat plot pada fungsi yang terdefinisi. Ini bekerja untuk semua fungsi yang mana mengembalikan NAN dari selang definisi.

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

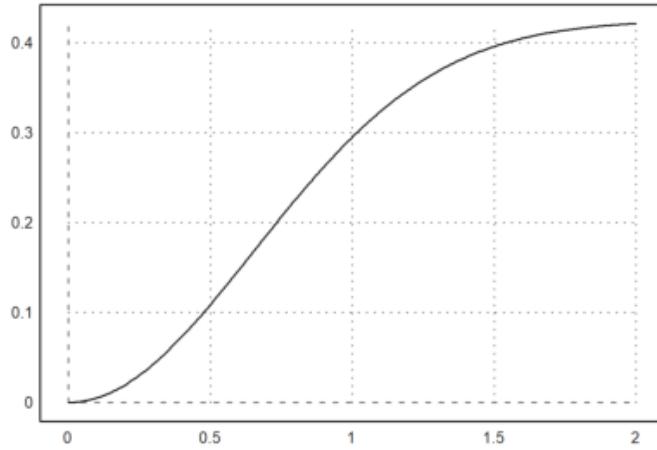


Parameter `square=true` (atau `>square`) memilih selang- $y$  secara otomatis jadi bahwa hasil adalah jendela kotak plot. Catatan bahwa secara bawaan, Euler menggunakan sebuah spasi kotak didalam layar plot.

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

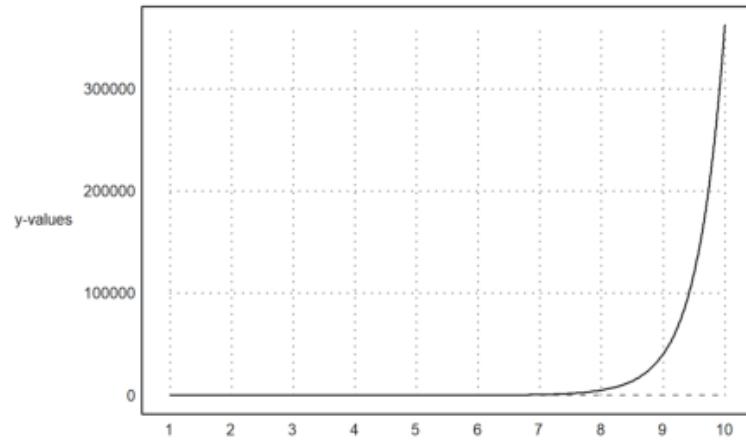


```
>plot2d(''integrate("sin(x) *exp(-x^2)", 0, x)'', 0, 2); // plot integral
```



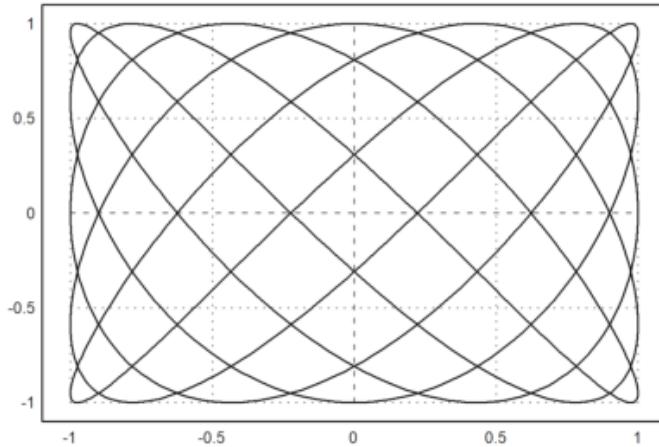
Jika kamu membutuhkan ruang yang lebih untuk label-y, panggil shrinkwindow() dengan parameter yang lebih kecil, atau atur sebuah nilai positif untuk "smaller" dalam plot2d().

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

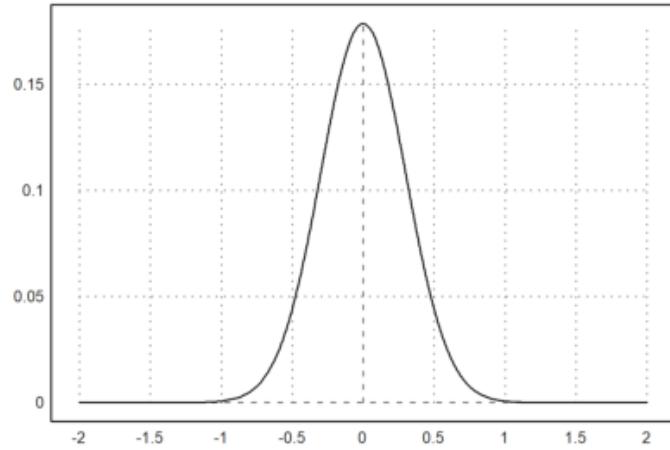


Ekspresi-ekspresi simbolik dapat juga digunakan, karena mereka disimpan sebagai ekspresi-ekspresi string yang sederhana.

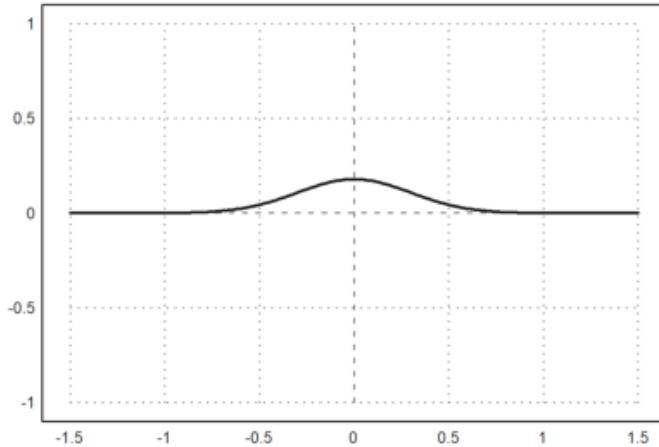
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x));
```



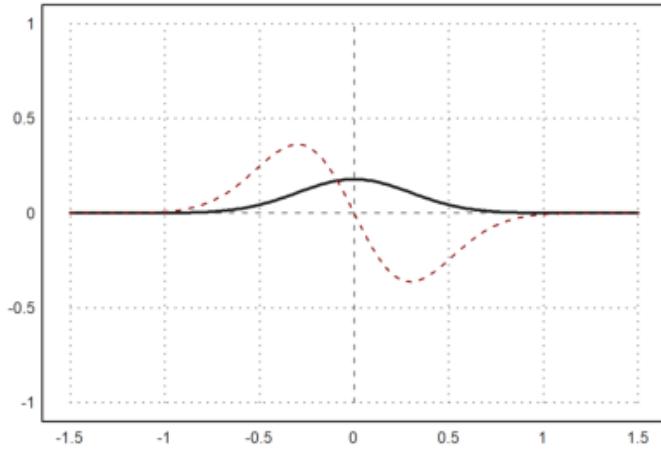
```
>a:=5.6; expr &= exp(-a*x^2)/a; // mendefinisikan ekspresi  
>plot2d(expr,-2,2); // membuat plot dari -2 ke 2
```



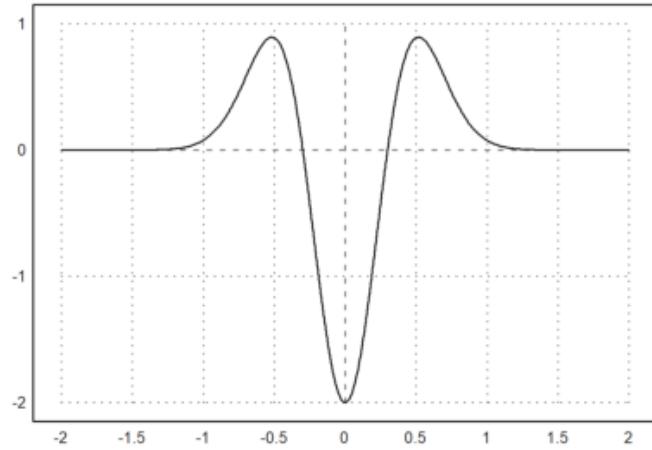
```
>plot2d(expr,r=1,thickness=2); // membuat plot dalam sebuah persegi sekitar (0,0)
```



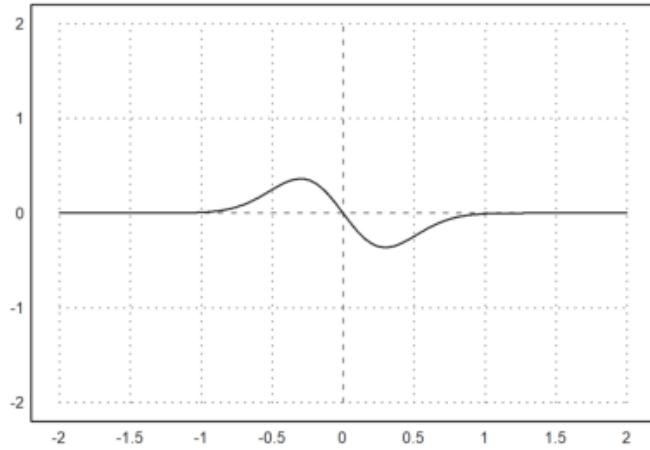
```
>plot2d(&diff(expr,x),>add,style="--",color=red); // menambahkan plot lainnya
```



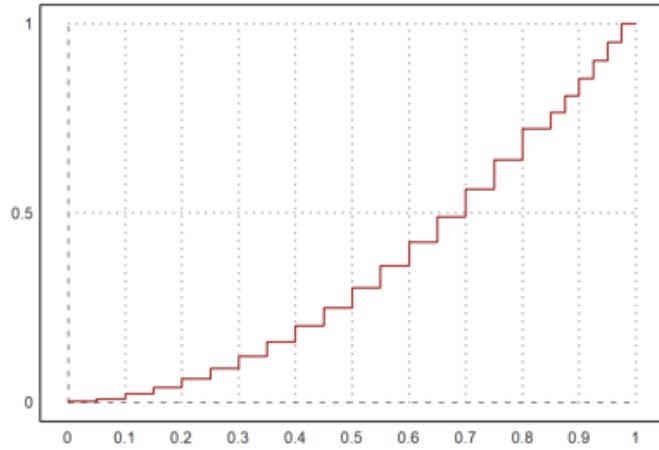
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1); // plot dalam persegi panjang
```



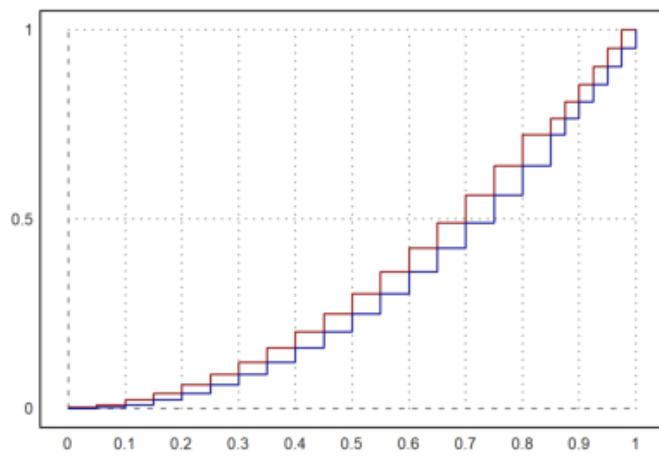
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // tetap membuat plot pada persegi
```



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



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



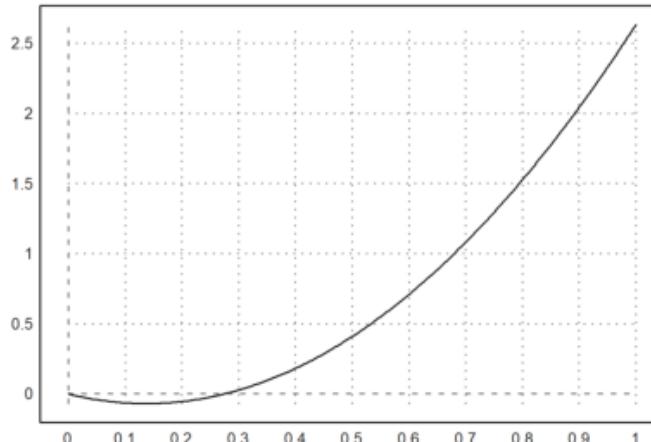
## Fungsi-fungsi dalam satu Parameter

---

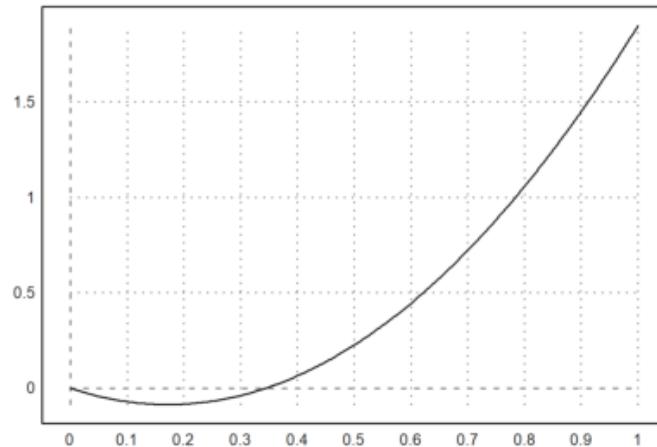
Hal yang paling penting dalam membuat fungsi plot adalah plot-plot planar adalah plot2d. Fungsi adalah pengimplementasian dalam bahawa Euler dalam berkas "plot.e", yang mana dimuat pada saat program dimulai.

Ini adalah beberapa contoh menggunakan sebuah fungsi. Seperti biasa dalam EMT, fungsi-fungsi itu bekerja untuk fungsi fungsi lainnya atau ekspresi-ekspresi, kamu dapat menambahkan parameter tambahan (disamping  $x$ ) yang mana bukan variabel-variabel global untuk fungsi dengan parameter parameter titik dua atau dengan sebuah koleksi pemanggilan.

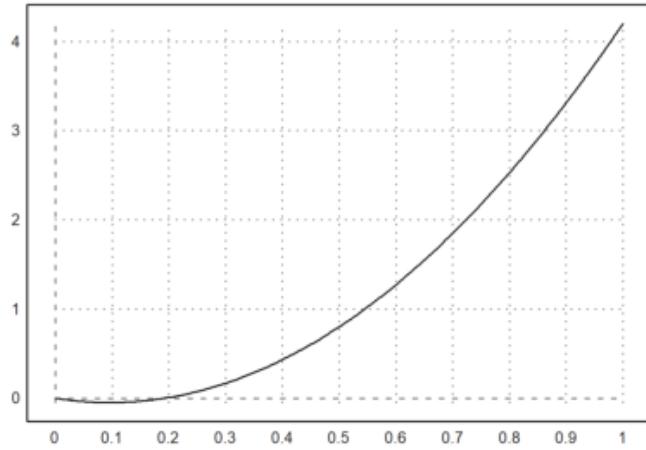
```
>function f(x, a) := x^2/a+a*x^2-x; // mendefinisikan sebuah fungsi  
>a = 0.3; plot2d("f", 0, 1; a); // plot fungsi f dengan a = 0.3
```



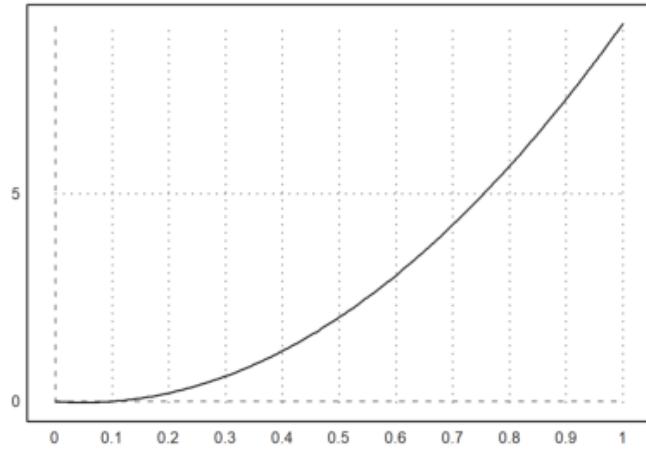
```
>plot2d("f", 0, 1; 0.4); // plot fungsi f dengan a = 0.4
```



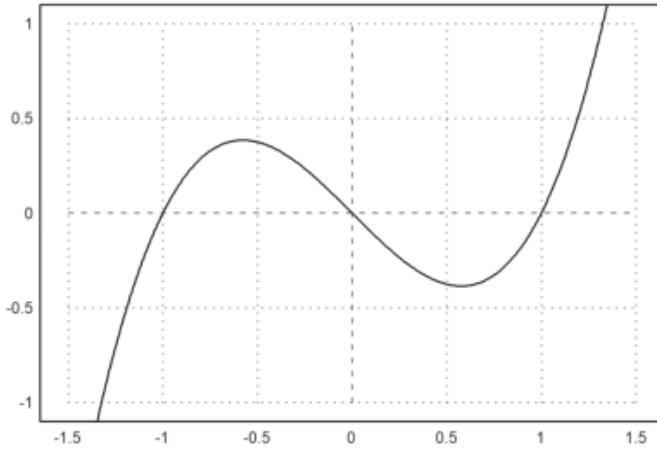
```
>plot2d({{"f",0.2}}, 0, 1); // plot dengan a = 0.2
```



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



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



Ini adalah rangkuman dari fungsi-fungsi yang diterima

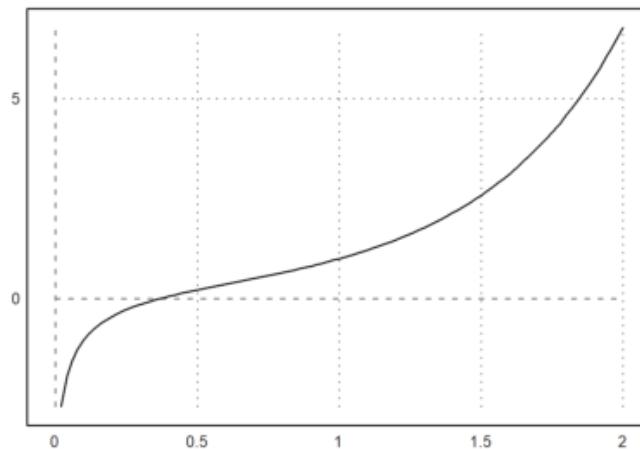
- ekspresi-ekspresi atau ekspresi-ekspresi simbolik dalam x
- fungsi-fungsi atau simbolik fungsi-fungsi dengan nama sebagai "f"
- fungsi-fungsi simbolik hanya dengan nama f

fungsi plot2d() juga menerima fungsi-fungsi simbolik. Untuk simbolik fungsi, nama bekerja secara sendiri.

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

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

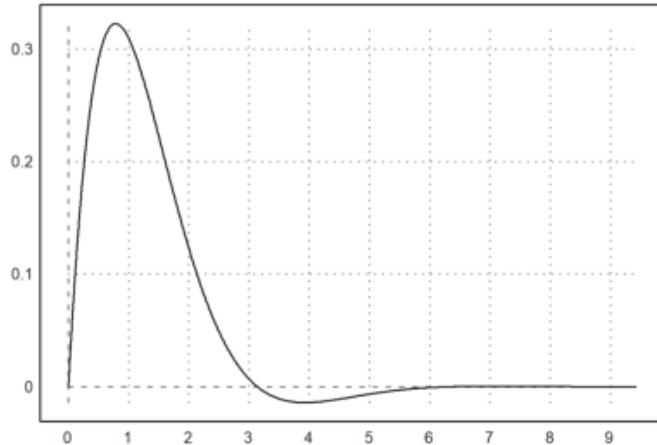
```
>plot2d(f, 0, 2):
```



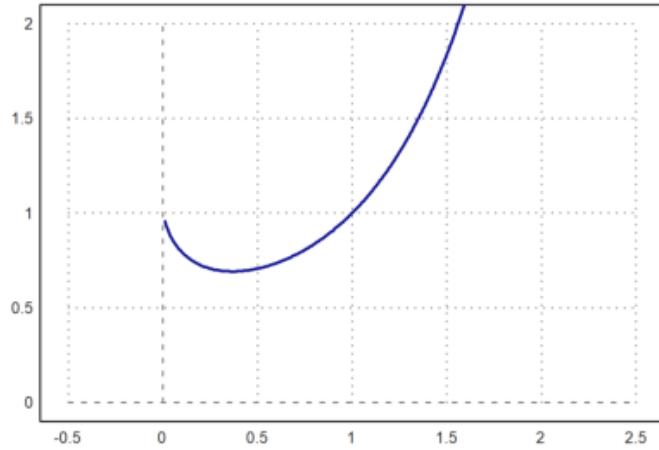
```
>expr &= sin(x) * exp(-x)
```

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

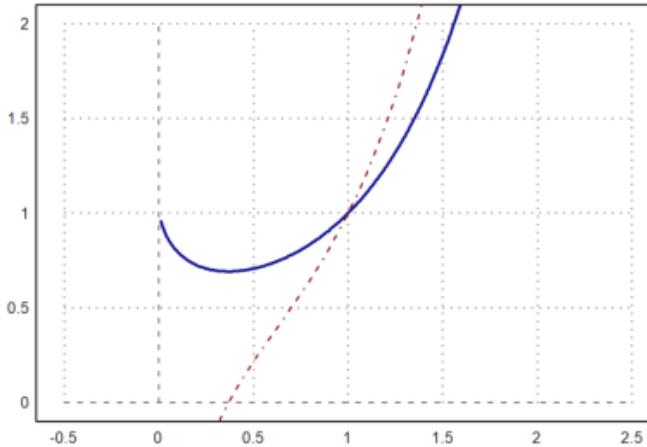
```
>plot2d(expr, 0, 3pi):
```



```
>function f(x) &= x^x;  
>plot2d(f, r = 1, cx = 1, cy = 1, color = blue, thickness = 2):
```



```
>plot2d(&diff(f(x), x), >add, color = red, style="-.-"):
```



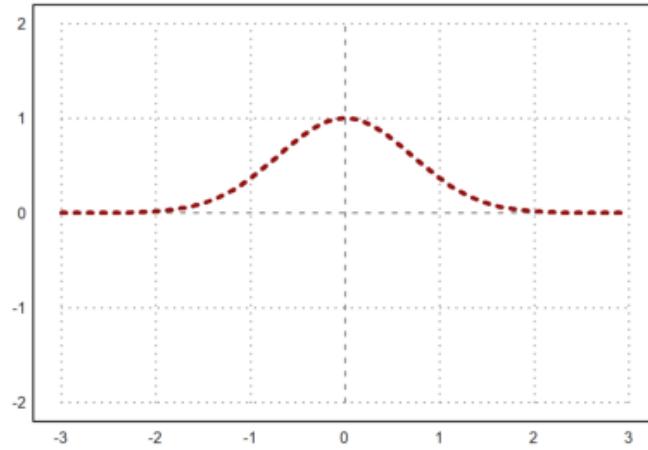
Untuk gaya garis disini memiliki banyak opsi.

- style = "...". Memilih dari "-", "-.", "-.", ".-", "-.-".
- color: Lihat dibawah untuk warna-warna.
- thickness: Bawaan adalah 1.

Warna warna dapat dipilih sebagai satu dari warna bawaan, atau sebagai sebuah warna RGB.

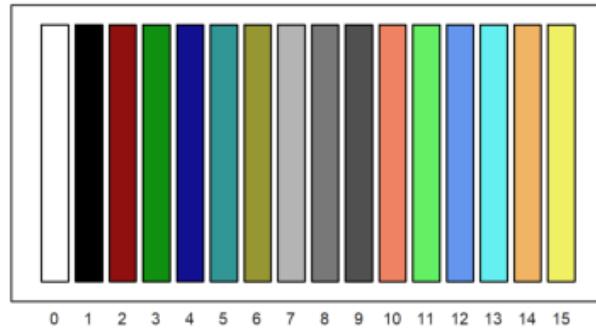
- 0..5: indeks warna bawaan.
- konstanta warna: white, black, red, green, blue, cyan, olive, lightgray, gray, darkgray, orange, lightgreen, turquoise, lightblue, lightorange, yellow.
- rgb(red, green, blue): parameter marameter adalah bilangan rill dalam [0, 1].

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



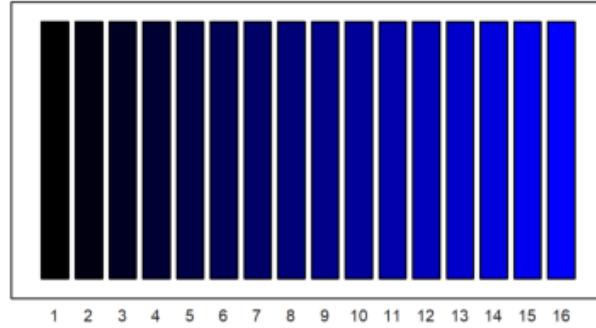
Ini adalah pandangan dari color yang telah terdefinisi dari EMT.

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



Tapi kamu dapat menggunakan warna apapun.

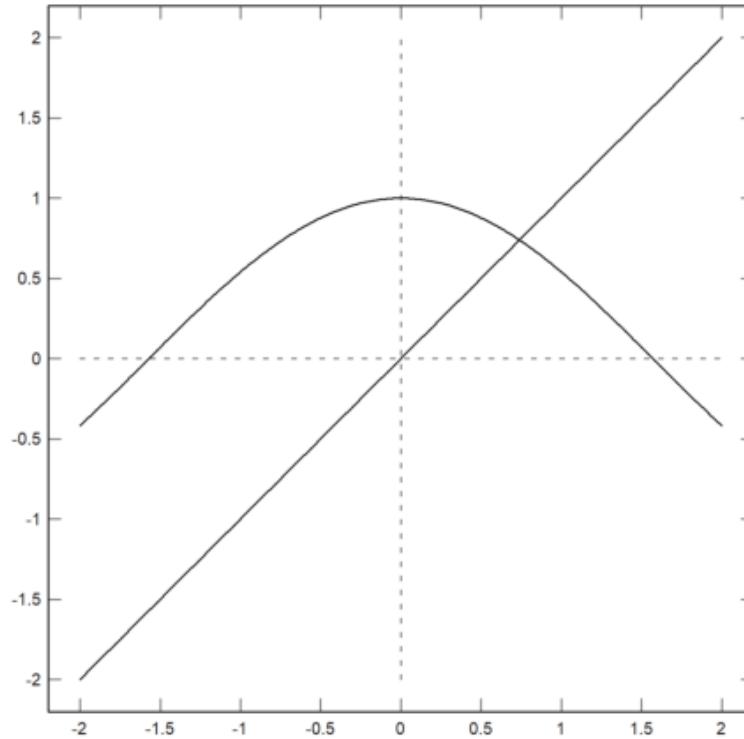
```
>columnsplot(ones(1, 16), grid = 0, color = rgb(0, 0, linspace(0, 1, 15))):
```



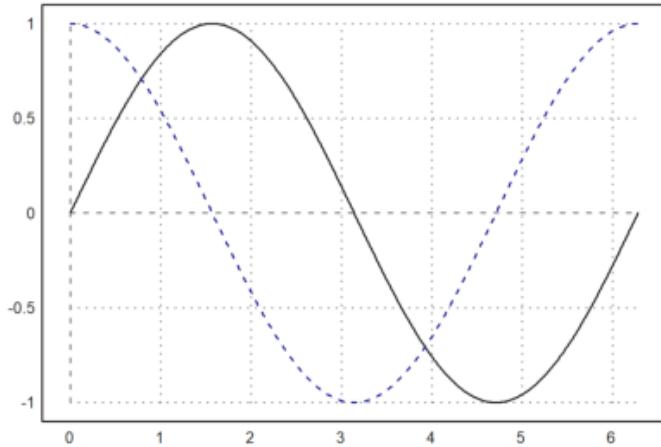
## Menggambar Beberapa Kurva pada bidang koordinat yang sama

Membuat plot lebih dari satu fungsi (fungsi-fungsi yang banyak) kedalam satu jendela dapat dilakukan dengan cara-cara yang berbeda. Salah satu metode yang digunakan >add untuk beberapa pemanggilan ke plot2d dalam semua, tetapi pemanggilan pertama. Kami telah menggunakan fitur ini dalam contoh diatas.

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

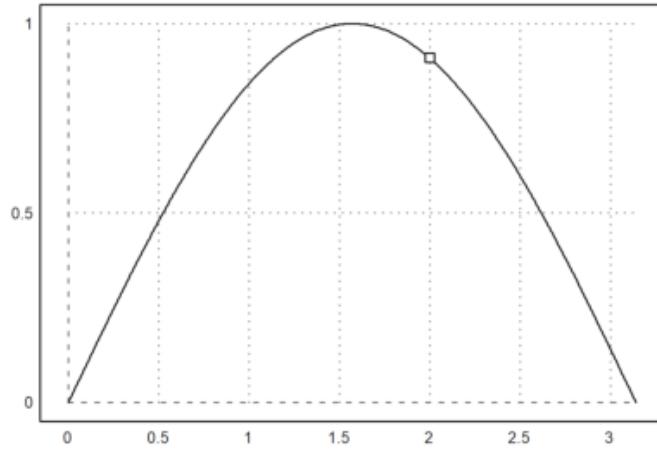


```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



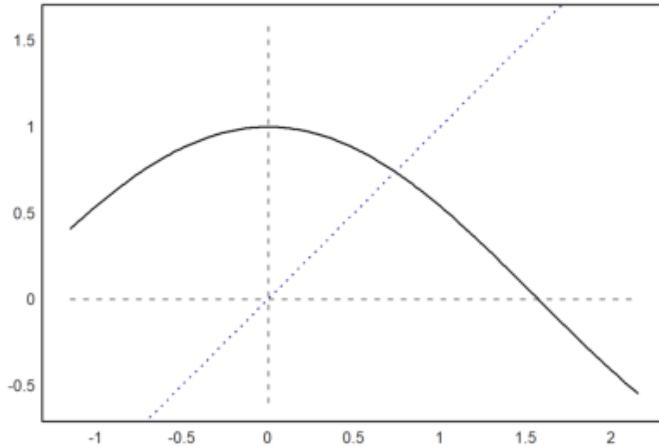
Salah satu kegunaan >add adalah untuk menambahkan titik pada kurva.

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

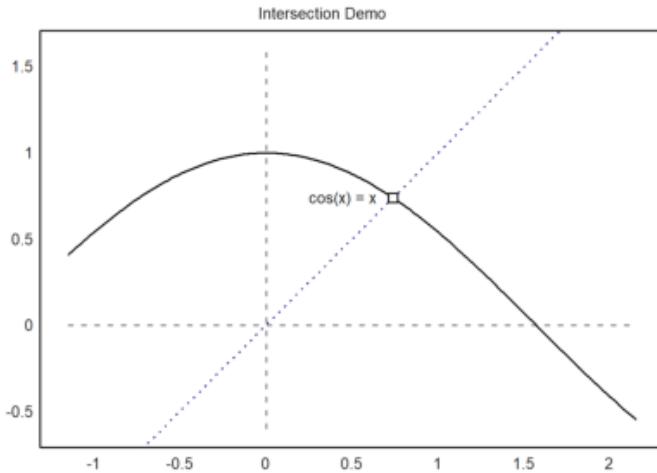


Kami menambahkan titik singgung dengan sebuah label (pada posisi "cl" untuk kiri tengah), dan memasukkan hasil kedalam notebook. Kami juga menambahkan sebuah judul kedalam plot.

```
>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):
```



Mengikuti demo, kami plot fungsi  $\text{sinc}(x) = \sin(x) / x$  dan ke-8 dan ke-16 ekspansi Taylor. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolik.

Plot ini dibuat mengikuti perintah baris-banyak dengan tiga kali pemanggilan untuk `plot2d()`. Pengaturan kedua dan ketiga `>add`, yang mana membuat plot menggunakan rentang sebelumnya.

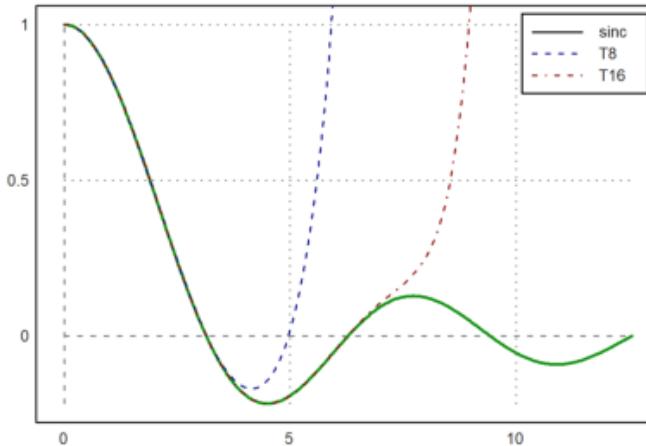
Kita menambahkan sebuah boks label untuk menjelaskan fungsi.

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

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

```

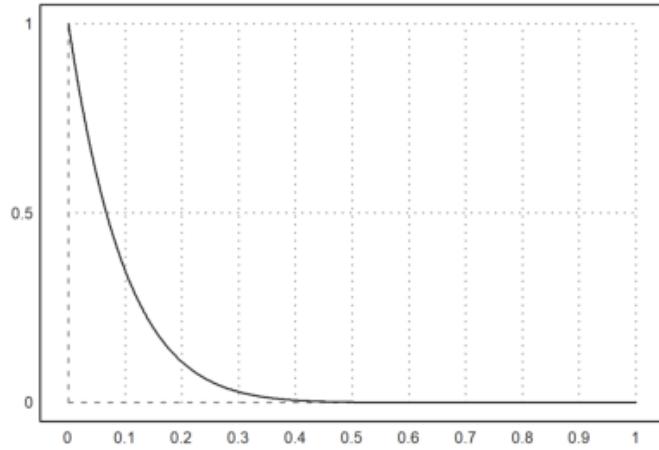
>plot2d("sinc(x)", 0, 4pi, color = green, thickness = 2);...
>plot2d(&taylor(sin(x) / x, x, 0, 8), >add, color = blue, style = "--");...
>plot2d(&taylor(sin(x) / x, x, 0, 16), >add, color = red, style = "-.-");...
>labelbox(["sinc", "T8", "T16"], styles=["--", "--", "-.-"],...
>colors=[black, blue, red]):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



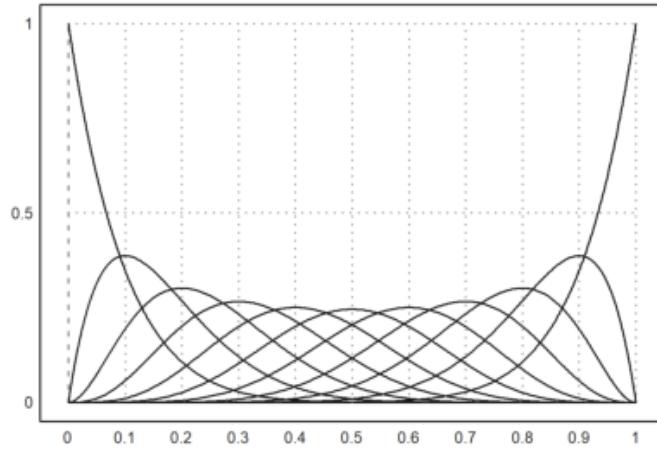
Contoh berikut, kami membuat Bernstein-Polynomials.

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

```
>plot2d("(1-x)^10",0,1): // plot fungsi pertama
```



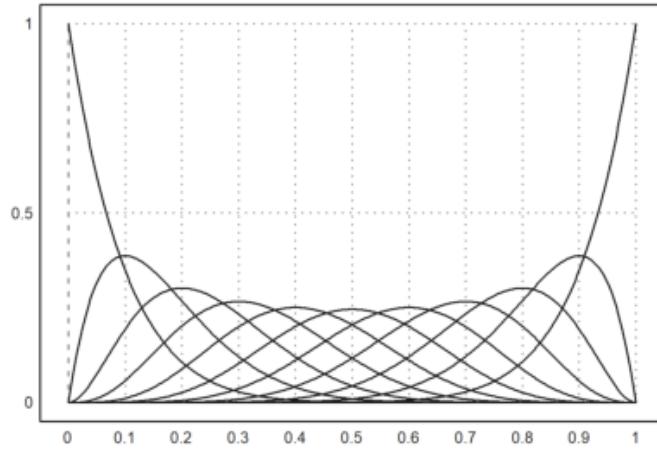
```
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;  
>insimg;
```



Metode kedua menggunakan sepasang dari sebuah matriks dari nilai-x dan sebuah matriks dari nilai-y dari ukuran yang sama.

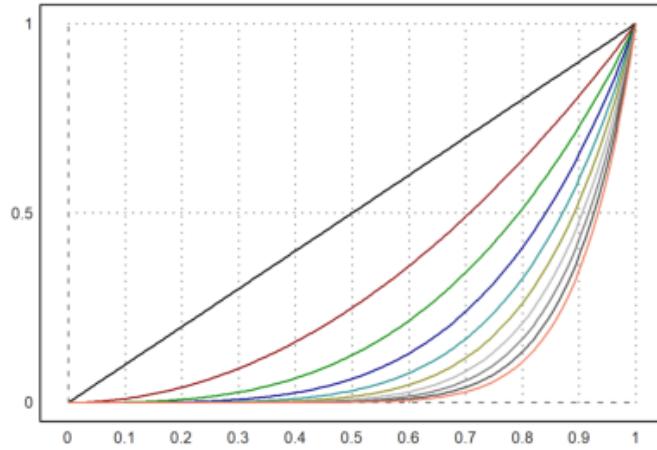
Kami menghasilkan sebuah matriks dari nilai-nilai dengan satu Bernstein-Polynomial dalam setiap baris. Untuk ini, kami mempersingkat menggunakan sebuah vektor kolom dari i. Lihat kembali ke pengantar tentang bahasa matriks untuk mempelajari lebih rinci.

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



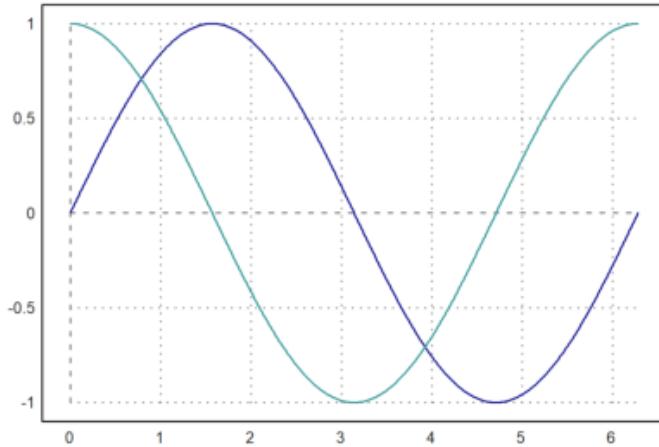
Catatan bahwa parameter warna dapat berupa sebuah vektor. Lalu setiap warna dapat digunakan untuk setiap baris dari matriks.

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

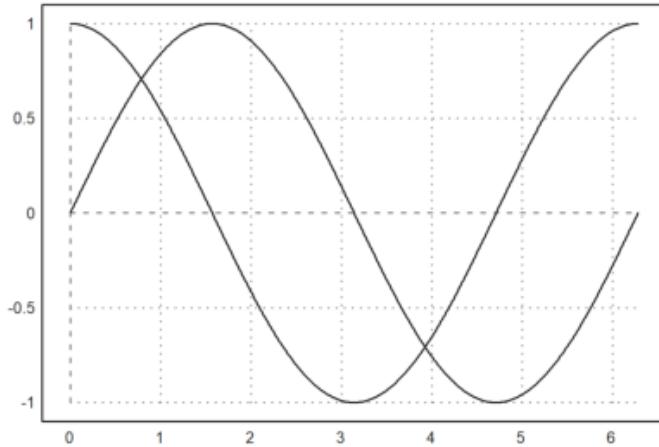


Metode lain adalah menggunakan vektor ekspresi (string). Kamu dapat menggunakan sebuah array warna, sebuah array dari gaya, dan sebuah array dari ketebalan dari panjang yang sama.

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



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



Kami dapat mendapatkan sebuah vektor dari Maxima menggunakan makelist() dan mxm2str().

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

$$[(1-x)^{10}, 10(1-x)^9x, 45(1-x)^8x^2, 120(1-x)^7x^3,$$

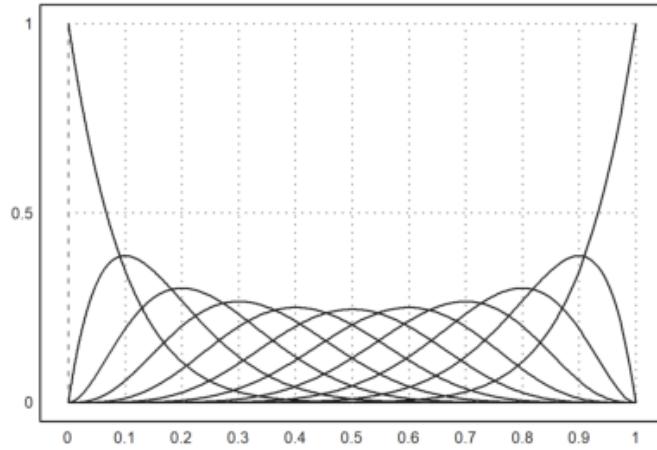
$$210(1-x)^6x^4, 252(1-x)^5x^5, 210(1-x)^4x^6, 120(1-x)^3x^7,$$

$$45(1-x)^2x^8, 10(1-x)x^9, x^{10}]$$

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

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

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

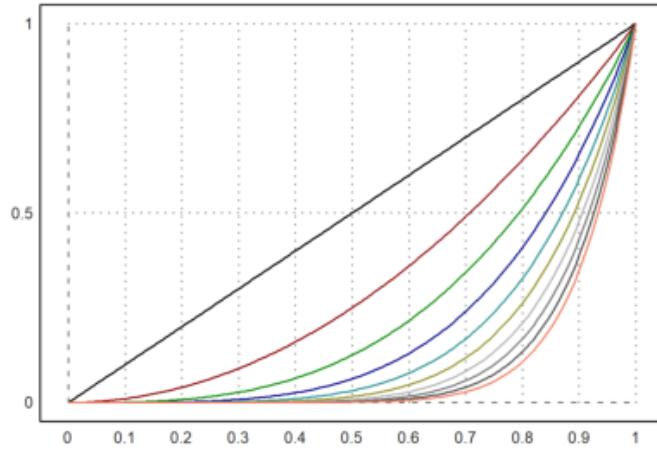


Alternatif lainnya adalah menggunakan bahasa matriks dari Euler.

Jika sebuah ekspresi menghasilkan sebuah matriks dari fungsi, dengan satu fungsi dalam setiap baris, semua fungsi-fungsi ini akan di plotkan menjadi satu plot.

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

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

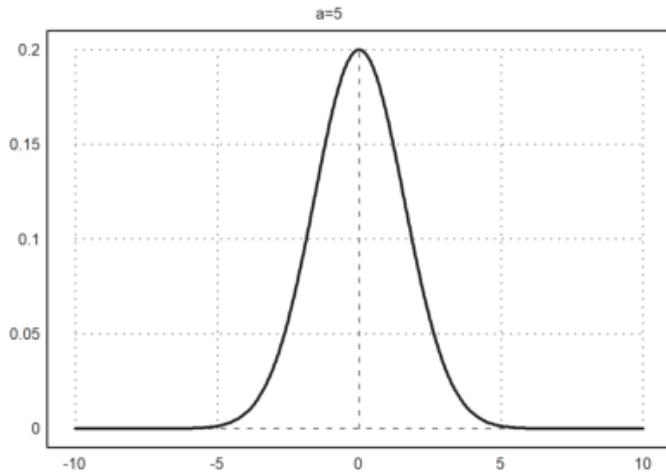


Ekspresi-ekspresi dan fungsi-fungsi satu-baris dapat dilihat variabel global.

Jika kamu tidak menggunakan variabel global, kamu membutuhkan untuk menggunakan sebuah fungsi dengan parameter ekstra, dan memasukkan parameter ini seperti sebuah parameter titik dua.

Berhati-hati, untuk menggambil semua parameter yang terpasang ke akhir dari perintah plot2d. Dalam contoh kami memasukkan a = 5 ke fungsi f, yang mana kami plot dari -10 ke 10.

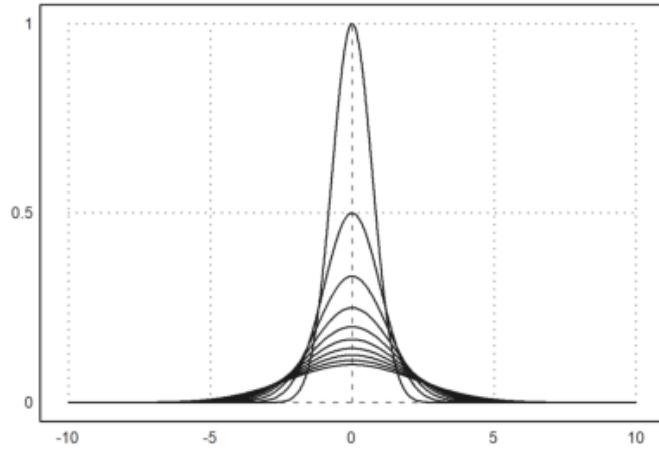
```
>function f(x,a) := 1/a *exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5");
```



Sebagai gantinya, gunakan sebuah collection dengan nama fungsi dan semua parameter-parameter tambahan. Daftar spesial ini disebut collection dan ini adalah cara yang disarankan untuk memasukkan argumen kedalam sebuah fungsi yang mana itu sendiri dimasukkan sebagai argumen kedalam fungsi lainnya.

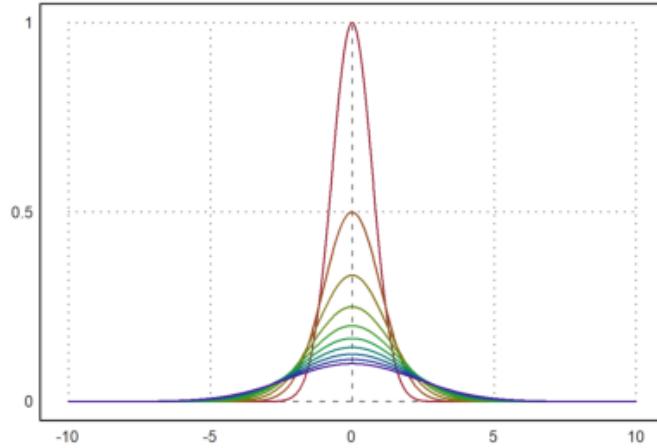
Contoh berikut, kami menggunakan sebuah iterasi untuk membuat plot beberapa fungsi (lihat tutorial tentang pemrograman untuk iterasi).

```
>plot2d({{"f",1}}, -10, 10); ...
>for a = 2:10; plot2d({{"f", a}}, >add); end:
```



Kami dapat menghasilkan hasil yang sama dengan menggunakan cara seperti menggunakan bahasa matriks EMT. Setiap baris dari matriks  $f(x, a)$  adalah satu fungsi. Terlebih, kita dapat mengatur warna untuk setiap baris dari matriks. Untuk penjelasan klik dua kali ke fungsi getspectral().

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



## Label Teks

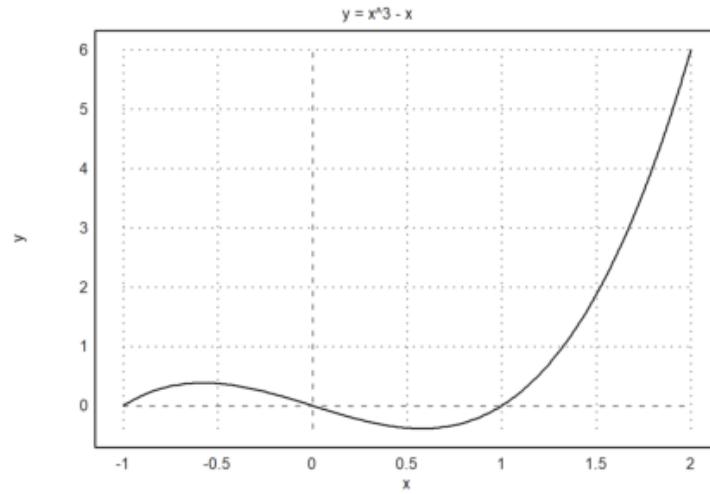
---

Dekorasi sederhana dapat berupa

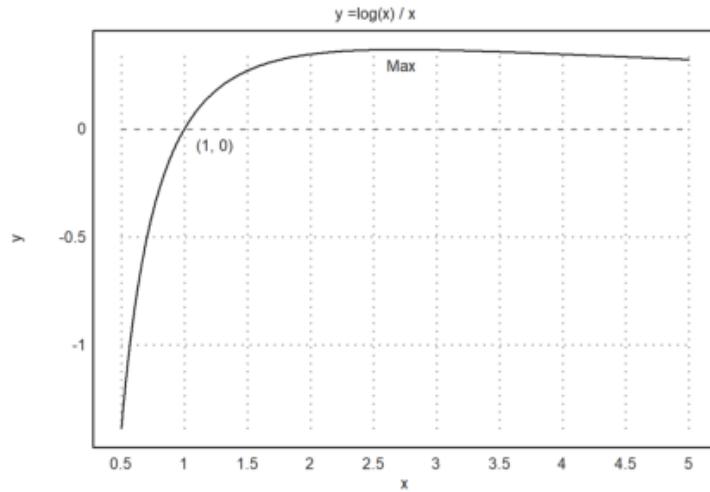
- sebuah judul dengan title="..."
- label-x dan label-y dengan xl="...", yl="..."
- teks label lainnya dengan label("...", x, y)

perintah label akan memplot ke plot sekarang sebagai koordinat plot (x, y). Ini dapat mengambil argumen posisi.

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

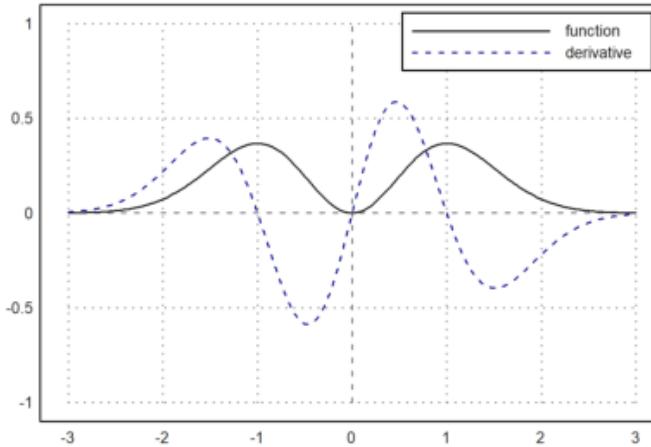


```
>expr := "log(x) / x";...
>plot2d(expr, 0.5, 5, title = "y =" +expr, xl = "x", yl = "y");...
>label("(1, 0)", 1, 0); label("Max", E, expr(E), pos="1c");
```



Ini juga fungsi `labelbox()`, yang mana dapat memperlihatkan fungsi-fungsi dan sebuah teks. Dengan mengambil vektor dari string dan warna, satu dari 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):
```

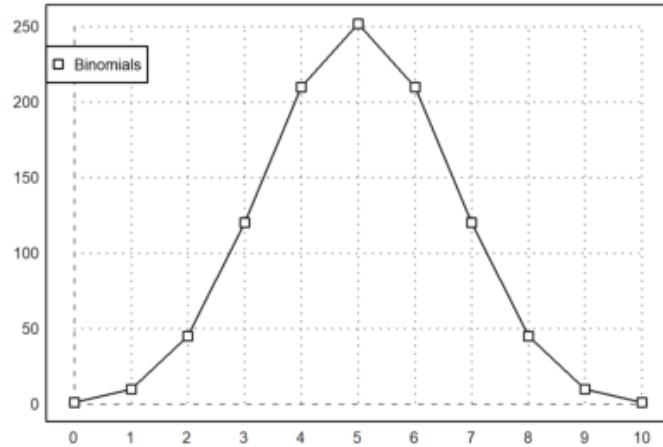


Box tersebut diposisikan pada kanan atas secara bawaan, tetapi posisi `>left` membuatnya pada kiri atas. Kamu dapat mengubahnya sebarang tempat yang kamu suka. Posisi terdapat pada pojok kanan atas dari box, dan angka-angkanya adalah pecahan dari ukuran jendela. Lebar secara otomatis diatur.

Untuk titik-titik plot, label box juga demikian. Tambahkan parameter `>points`, atau sebuah vektor dari pernyataan, satu untuk setiap label.

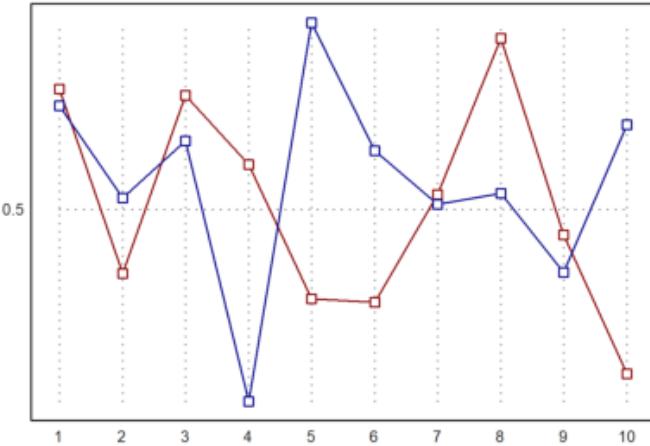
Seperti contoh yang ditunjukkan, ini hanya satu fungsi. Jadi kami dapat menggunakan strings daripada vektor string. Kami mengatur warna teks ke hitam untuk contoh ini.

```
>n = 10; plot2d(0: n, bin(n, 0: n), >addpoints);...
>labelbox("Binomials", styles = "[]", >points, x = 0, y = 0.1, ...
>tcolor = black, >left):
```



Gaya dari plot ini juga tersedia dalam statplot(). Seperti dalam plot2d() warna-warna dapat diatur untuk setiap baris dari plot. Ini adalah plot-plot spesial untuk tujuan statistika (lihat tutorial tentang statistika).

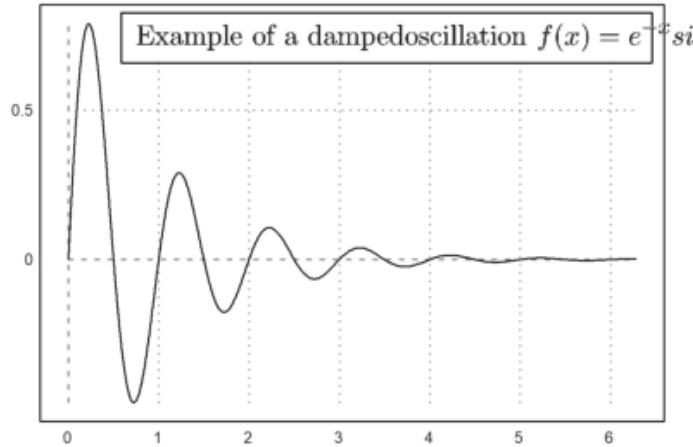
```
>statplot(1: 10, random(2, 10), color = [red, blue]):
```



Fitur yang sama dari fungsi `textbox()`.

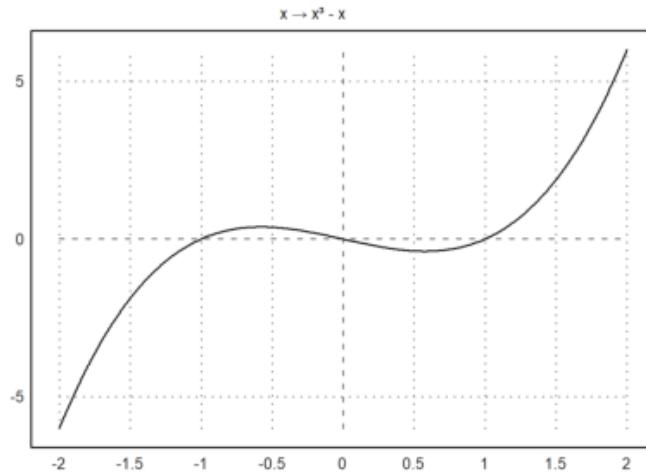
Secara bawaan lebar diatur setara dengan lebar maksimal dari lebar baris teks. Tapi ini dapat diatur oleh pengguna juga.

```
>function f(x) &= exp(-x) * sin(2 * pi * x); ...
>plot2d("f(x)", 0, 2pi); ...
>textbox(latex("\text{Example of a damped oscillation}\\" f(x) = e^{-x}sin(2 \ pi x)'), w = 0.85):
```



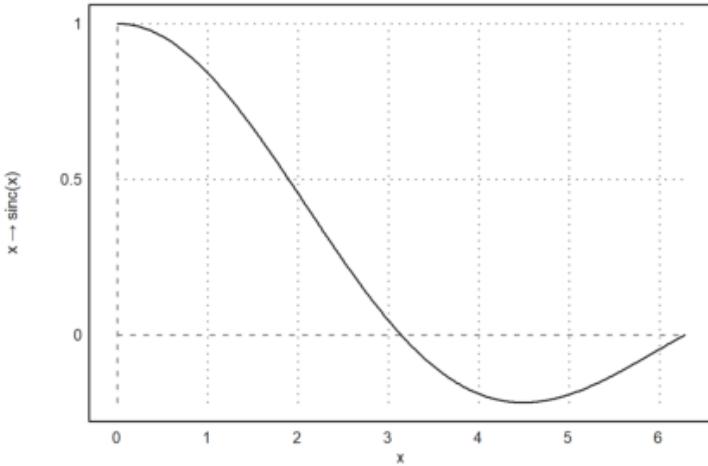
Label-label teks, judul, label box dan teks lainnya dapat berupa string Unicode (lihat sintak dari EMT untuk lebih lanjut tentang string Unicode).

```
>plot2d("x^3 - x", title = u"x ↗ x³ - x"):
```



Label-label dalam sumbu x dan y dapat secara vertikal, mengikuti sumbunya.

```
>plot2d("sinc(x)", 0, 2pi, xl = "x", yl = u"x &rarr; sinc(x)", >vertical):
```



LaTeX

---

Kamu juga dapat mem-plot formula LaTex jika kamu telah menginstall sistem LaTex. Saya merekomendasikan MiKTeX. Jalur ke binari "latex" dan "dvipng" harus ada dalam jalur sistem, ttau kamu telah mengatur LaTeX dalam menu options.

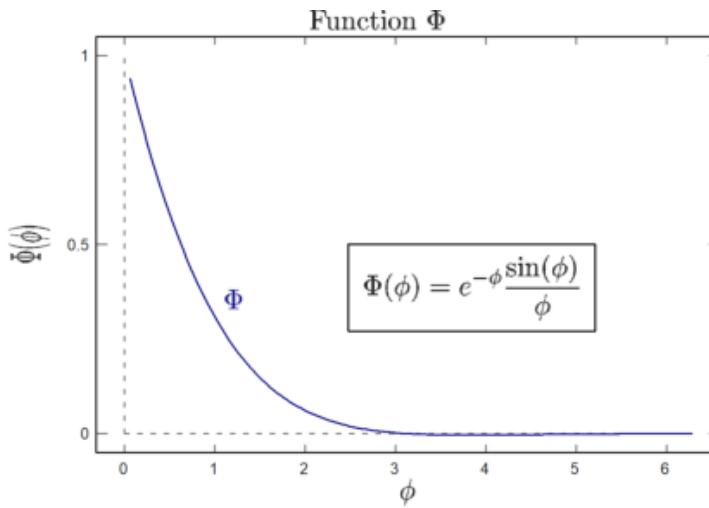
Catatan, bahwa pengubah LaTeX itu lambat. Jika kamu ingin LaTex dalam plot yang memiliki animasi, kamu harus memanggil `latex()` sebelum iterasi pertama dan gunakan hasilnya (sebuah gambar dalam sebuah matriks RGB).

Seperti plot berikut, kita menggunakan LaTeX untuk label x dan label y, sebuah label, sebuah label box 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):

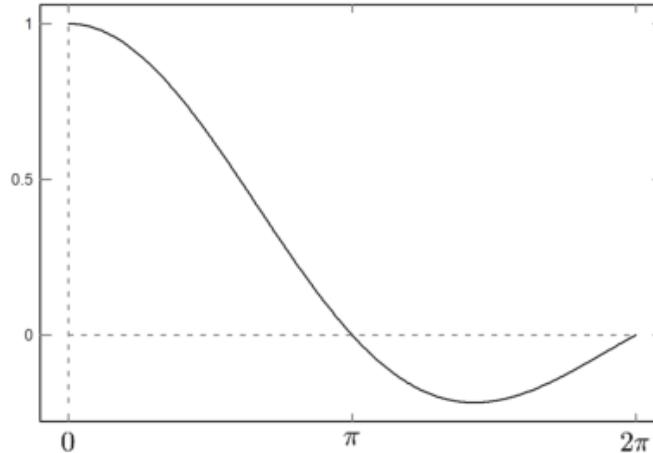
```



Terkadang, kita ingin spasi yang non-conformal dan label teks dalam sumbu-x. Kita dapat menggunakan xaxis() dan yaxis() seperti yang akan ditunjukkan kemudian.

Cara termudah untuk melakuakn sebuah plot yang tidak ada dengan sebuah frame menggunakan grid = 4 dan lalu tambahkan grid dengan ygrid() dan xgrid(). Seperti contoh berikut, kita gunakan tiga string LaTeX untuk label dalam 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); ...
>xlabel([0, pi, 2pi], ["0", "\pi", "2\pi"], >tex):
```

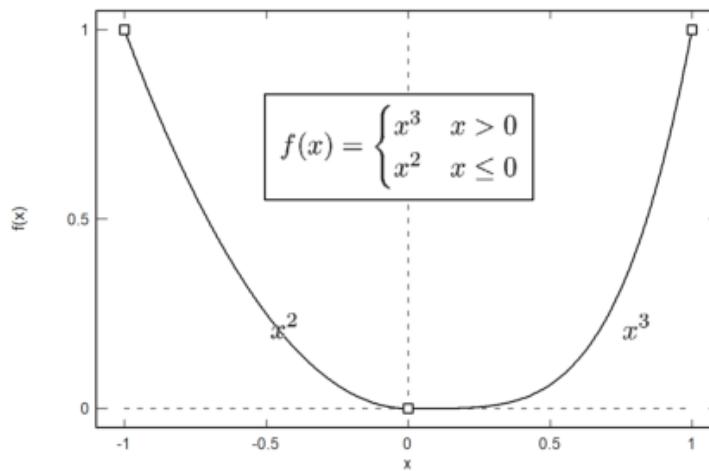


Tentu saja, fungsi-fungsi dapat juga digunakan.

```
>function map f(x) ...
if x > 0 then return x^4
else return x^2
endif
endfunction
```

Parameter "map" membantu untuk menggunakan fungsi dari vektor. Untuk membuat plot, ini tidak akan digunakan. Tetapi untuk demonstrasi bahwa vektorisasi sangat berguna, kita tambahkan beberapa titik kedalam plot pada  $x = -1$ ,  $x = 0$  dan  $x = 1$ . Seperti plot berikutnya kita juga memasukkan beberapa code LaTeX. Kami gunakan ini untuk dua label dan sebuah teks box. Tentu saja, kamu hanya ingin menggunakan LaTeX jika kamu telah menginstal LaTeX secara 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 = "11");...
>textbox(..., ...
>latex("f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0 \end{cases}"), ...
>x = 0.7, y = 0.2):
```



Ketika membuat plot sebuah fungsi atau sebuah ekspresi, parameter `>user` memperbolehkan pengguna untuk memperbesar dan memindahkan plot dengan kursor atau mouse.

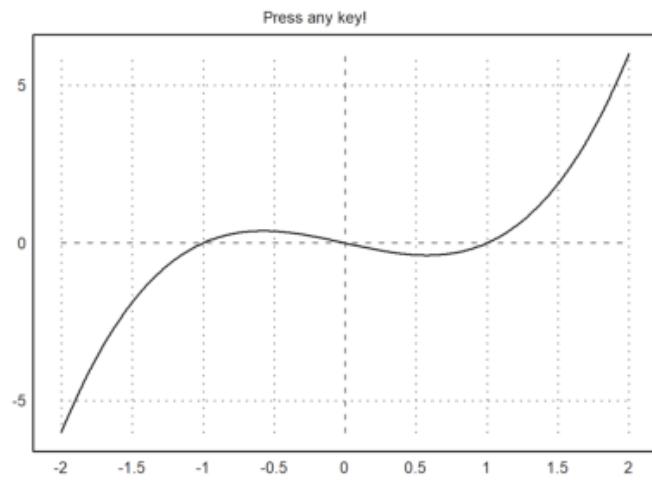
Pengguna dapat

- memperbesar dengan + or -
- memindahkan plot dengan kursor
- memilih jendela sebuah plot dengan mouse
- mengatur ulang pandangan dengan spasi
- keluar dengan return

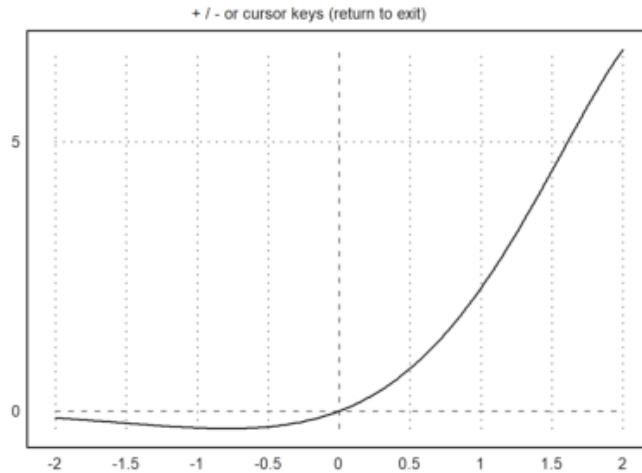
Spasi akan mengatur ulang plot menjadi jendela plot asli.

Ketika membuat plot sebuah data, `>user` parameter akan secara sederhana menunggu tombol diketikkan.

```
>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)":
```



Demonstrasi berikut sebuah cara lanjutan dari interaksi pengguna (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi yang sudah ada mousedrag() menunggu mouse atau pergerakan keyboard. ini akan melaporkan mouse yang bergerak dan keyboard tertekan. fungsi dragpoints() menggunakan ini, dan memungkinkan pengguna menyeret sebarang titik dalam sebuah plot.

Kita butuh sebuah fungsi plot terlebih dahulu. Sebagai contoh, kita menginterpolasi dalam 5 titik oleh sebuah polinomial. Fungsi akan memplot sampai sebuah luasan plot yang tetap.

```
>function plotf(xp, yp, select)...
d=interp(xp, yp);
plot2d("interpval(xp, d, x)"; d, xp, r = 2);
plot2d(xp, yp, >points, >add);
if select > 0 then
```

```

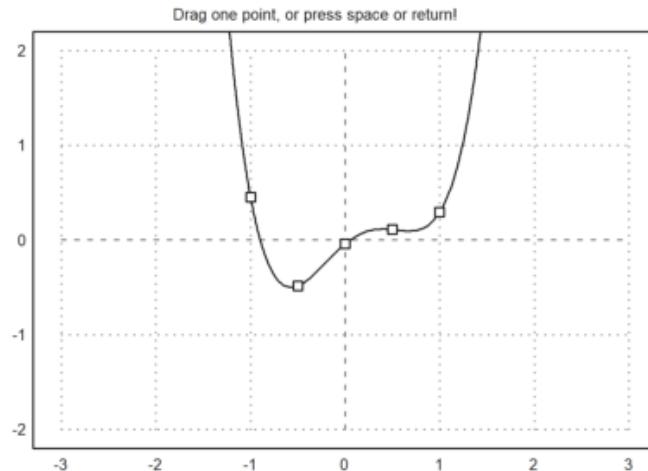
plot2d(xp[select], yp[select], color = red, >points, >add);
endif;
title("Drag one point, or press space or return!");
endfunction

```

Catatan parameter semicolon dalam plot2d(d dan xp), yang mana dimasukkan ke evaluasi dari fungsi interp(). Tanpa ini, kita harus menuliskan sebuah fungsi plotinterp() terlebih dahulu, mengakses nilai secara global.

Sekarang kita menghasilkan beberapa nilai acak dan biarkan pengguna menggeser titik-titiknya.

```
>t = -1:0.5:1; dragpoints("plotf", t, random(size(t)) - 0.5);
```



Ini juga sebuah fungsi, yang mana membuat plot fungsi lainnya tergantung dalam sebuah vektor dari parameter, dan membiarkan pengguna menyesuaikan parameter-parameternya.

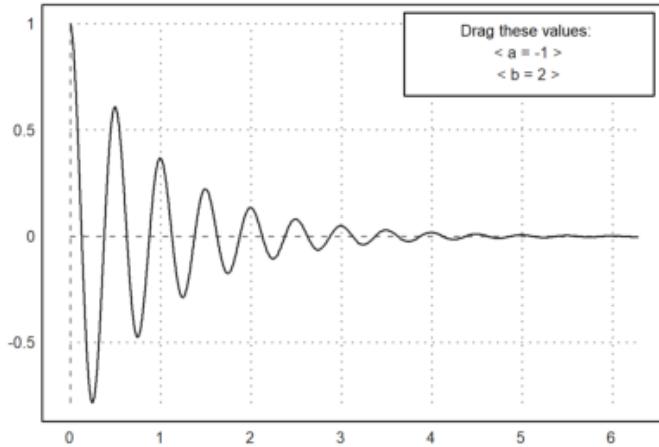
Pertama-tama kita butuh fungsi plot.

```
>function plotf([a, b]) := plot2d("exp(a * x) * cos(2pi * b * x)", 0, 2pi; a, b);
```

Lalu kita butuhkan nama untuk parameter-parameter, nilai awal dn sebuah matriks nx2 dari selang, secara pilihan sebuah baris judul.

Ini adalah slider yang interaktif, dimana dapat merubah nilai oleh pengguna. Fungsi dragvalues() tersedia untuk hal ini.

```
>dragvalues("plotf", ["a", "b"], [-1, 2], [[-2, 2]; [1, 10]], ...
>heading="Drag these values: ", hcolor = black):
```

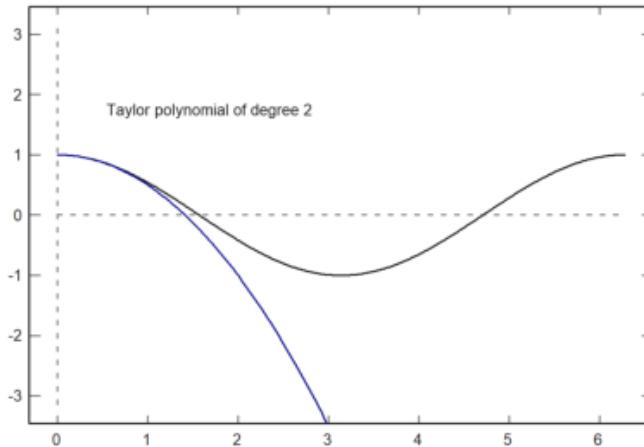


Ini memungkinkan untuk membatasi nilai yang dapat digerakkan hanya bilangan bulat. Sebagai contoh, kita menulis sebuah fungsi plotf, yang mana membuat plot sebuah polinomial Taylor berderajat n ke fungsi cosinus.

```
>function plotf(n):
    plot2d("cos(x)", 0, 2pi, >square, grid=6);
    plot2d(&"taylor(cos(x), x, 0, @n)", color = blue, >add);
    textbox("Taylor polynomial of degree " + n, 0.1, 0.2, style = "t", >left);
endfunction
```

Sekarang kita bisa membuat derajat n menjadi bervariasi dari 0 ke 20 dengan 20 pemberhentian. Hasil dari dragvalues() digunakan untuk membuat plot sketsa dengan n, dan memasukkan plot kedalam notebook.

```
>nd = dragvalues("plotf", "degree", 2, [0, 20], 20, y = 0.8, ...
>heading="Drag the value: "); ...
>plotf(nd):
```



Demonstrasi sederhana ini dari sebuah fungsi. Pengguna dapat menggambar diatas jendela plot, meninggalkan titik-titik lintasan.

```
>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
```

## Gaya-Gaya Plot 2D

---

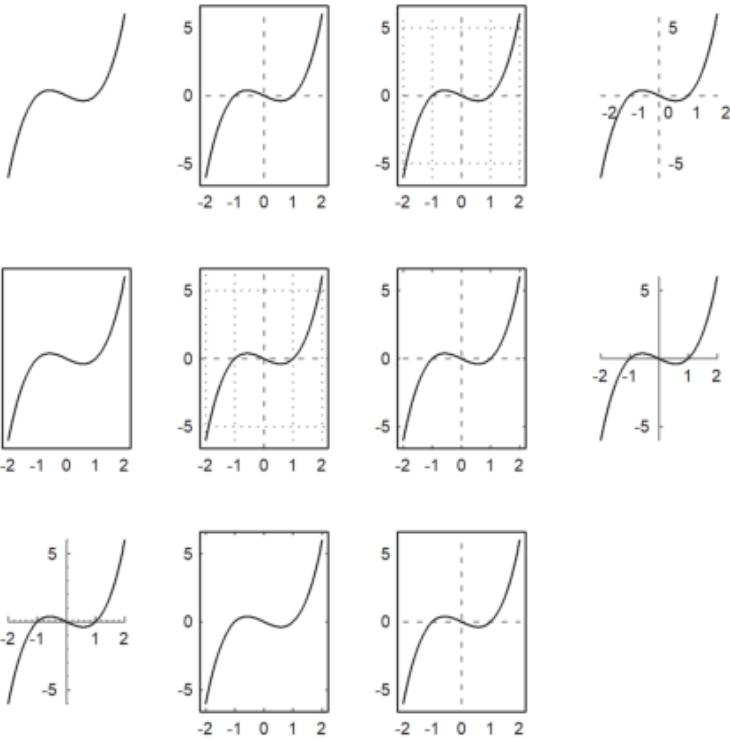
Secara bawaan, EMT secara otomatis menghitung titik sumbu dan menambahkan label ke titiknya. Ini dapat diubah dengan parameter grid. Gaya bawaan dari sumbu dan labelnya dapat diubah. Sebagai tambahan, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang gaya bawaan, gunakan reset().

```

>aspect();
>figure(3, 4);
>figure(1); plot2d("x^3-x", grid=0);... // tidak ada garis bantu, rangka atau sumbu
>figure(2); plot2d("x^3-x", grid=1);... // sumbu-x-y
>figure(3); plot2d("x^3-x", grid=2);... // titik bantu bawaan
>figure(4); plot2d("x^3-x", grid=3);... // sumbu x-y dengan label didalam

```

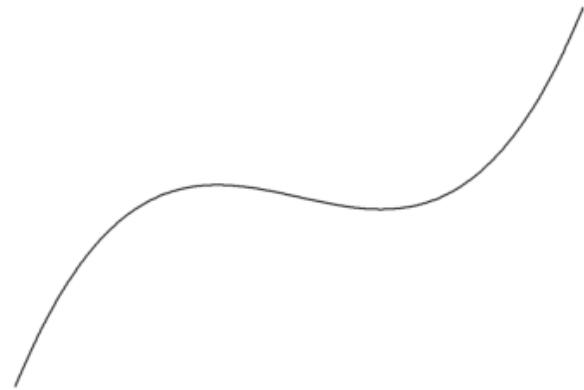
```
>figure(5); plot2d("x^3-x", grid=4);... // tidak ada titik bantu, hanya label  
>figure(6); plot2d("x^3-x", grid=5);... // bawaan, tetapi tidak ada margin  
>figure(7); plot2d("x^3-x", grid=6);... // hanya sumbu  
>figure(8); plot2d("x^3-x", grid=7);... // hanya sumbu, titik bantu pada sumbu  
>figure(9); plot2d("x^3-x", grid=8);... // hanya sumbu, titik bantu lebih mulus pada sumbu  
>figure(10); plot2d("x^3-x", grid=9);... // bawaan, titik bantu kecil didalam  
>figure(11); plot2d("x^3-x", grid=10);... // tidak ada titik bantu, hanya sumbu  
>figure(0):
```



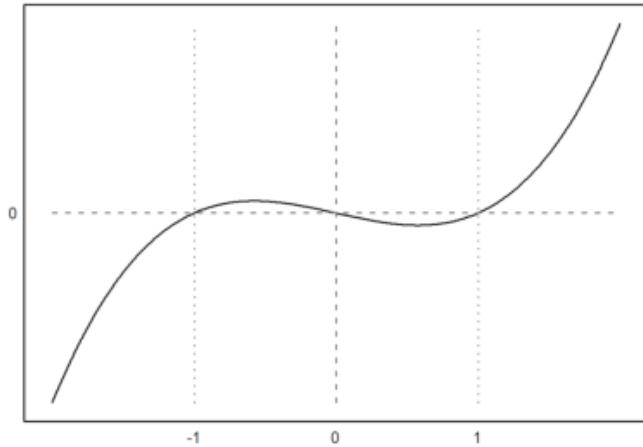
Parameter `<frame` membuat frame tidak ada, dan `famecolor = blue` mengubah frame menjadi berwarna biru.

Jika kamu ingin titik-titik kamu, kamu dapat menggunakan `style = 0`, dan tambahkan kesemuanya nanti.

```
>aspect(1.5);
>plot2d("x^3 - x", grid = 0); // plot
```

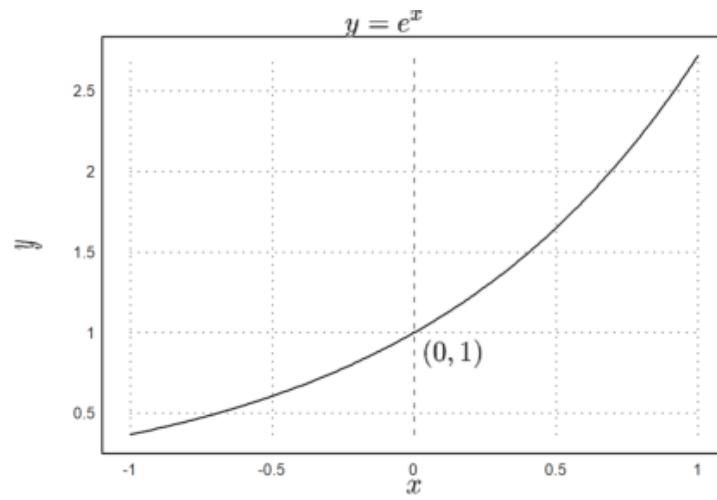


```
>frame; xgrid([-1, 0, 1]); ygrid(0); // menambahkan kerangka dan garis bantu
```



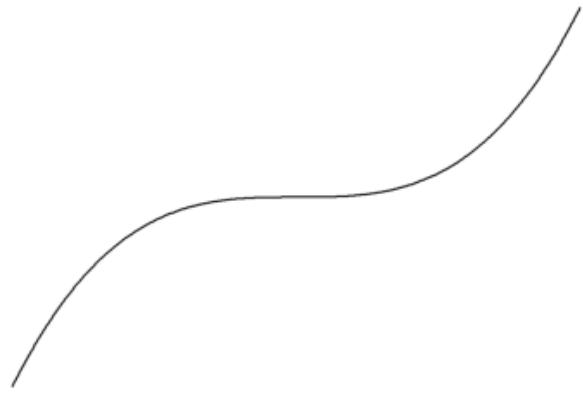
Untuk judul plot dan sumbu label, perhatikan contoh berikut

```
>plot2d("exp(x)", -1, 1);
>textcolor(black); // atur warna teks menjadi hitam
>title(latex("y = e^x")); // judul diatas plot
>xlabel(latex("x")); // "x" untuk sumbu-x
>ylabel(latex("y"), >vertical); // "y" secara vertikal untuk sumbu-y
>label(latex("(0, 1)'), 0, 1, color = blue); // label sebuah titik
```

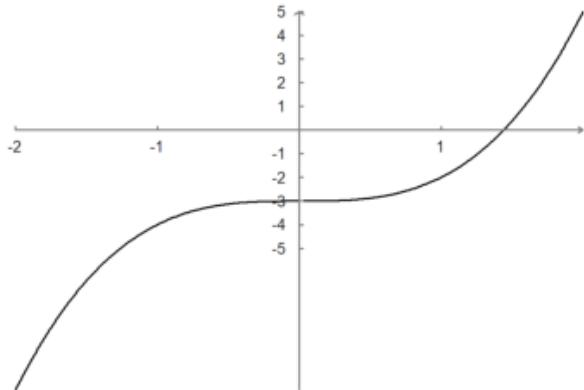


Sumbu-sumbu dapat digambarkan secara terpisah dengan xaxis() dan yaxis().

```
>plot2d("x^3 - 3", <grid, <frame):
```



```
>xaxis(0, xx = -2:1, style = "->"); yaxis(0, yy = -5:5, style = "->"):
```

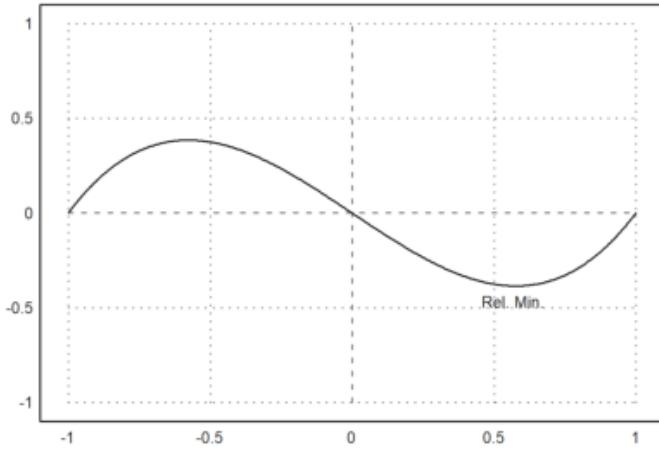


Teks dalam plot dapat diatur dengan label(). Seperti contoh, "lc" berarti tengah bawah. Ini mengatur posisi dari label secara relatif ke dalam koordinat plot.

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

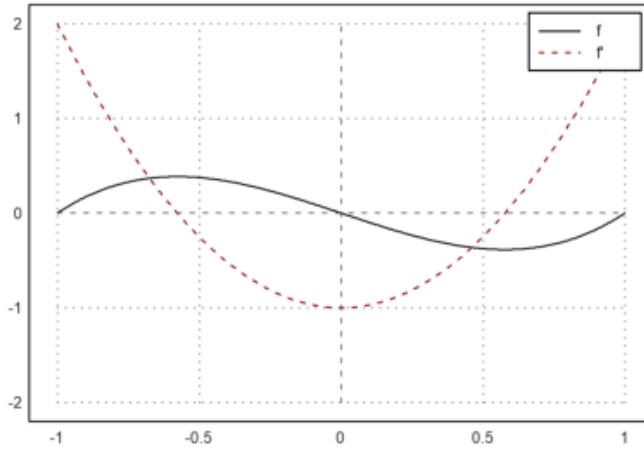
$$x^3 - x$$

```
>plot2d(f, -1, 1, >square);
>x0 = fmin(f, 0, 1); // menghitung titik minimum
>label("Rel. Min. ", x0, f(x0), pos = "lc"); // menambahkan sebuah label disini
```

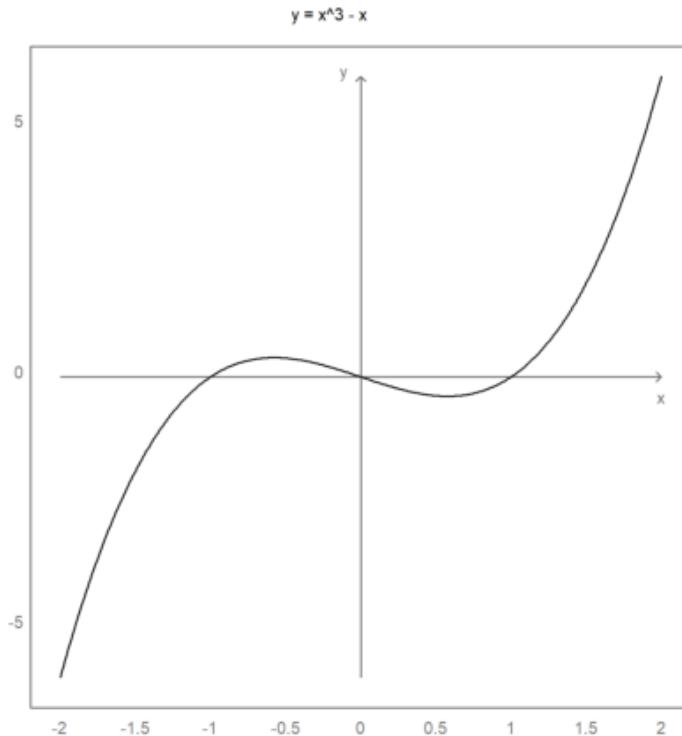


Ini juga teks box.

```
>plot2d(&f(x), -1, 1, -2, 2); // fungsi  
>plot2d(&diff(f(x), x), >add, style = "--", color = red); // turunan  
>labelbox(["f", "f'"], ["-", "--"], [black, red]): // box label
```



```
>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();
```

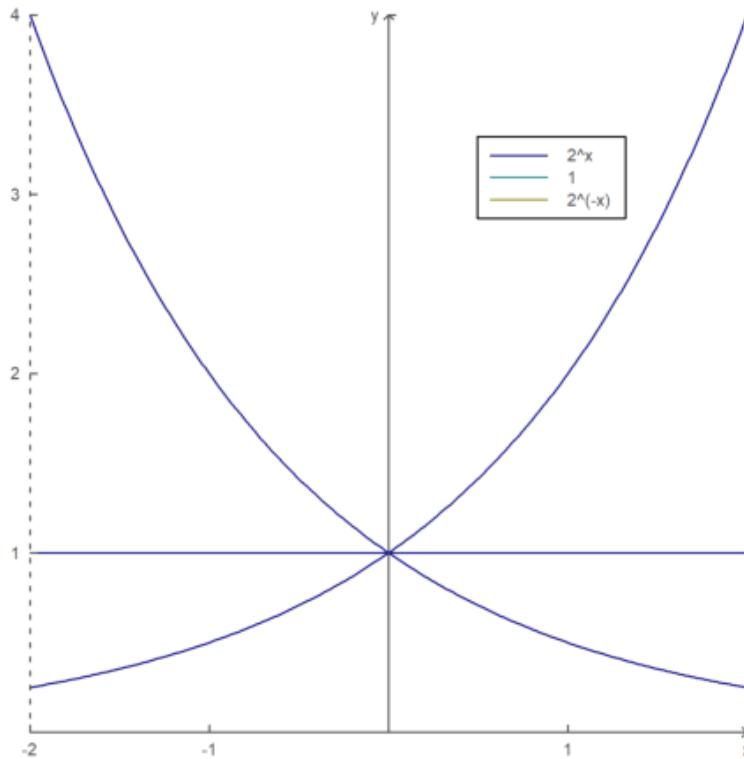


Untuk kontrol lebih, sumbu-x dan sumbu-y dapat dilakukan secara manual. Perintah `fullwindow()` menyebarkan jendela plot karena kita tidak membutuhkan tempat untuk label luar pada judul plot. Gunakan `shrinkwindow()` atau `reset()` untuk mengatur ulang ke pengaturan bawaan.

```

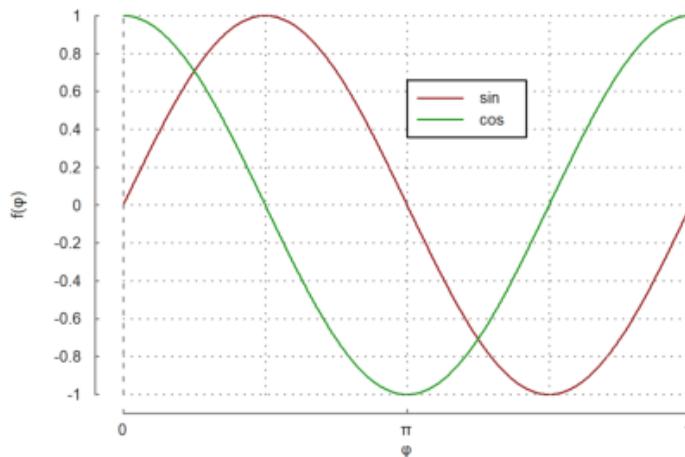
>fullwindow; ...
>gridstyle(color = darkgray, textcolor = darkgray); ...
>plot2d(["2^x", "1", "2^(-x)"], a=-2, b=2, c=0, d=4, <grid, color=4.6, <frame);...
>xaxis(0, -2:1, style="->"); xaxis(0, 2, "x", <axis); ...
>yaxis(0, 4, "y", style="->"); yaxis(-2, 1:4, >left); ...
>labelbox(["2^x", "1", "2^(-x)"], colors=4:6, x=0.8, y=0.2);...
>reset;

```



Ini adalah contoh lainnya, dimana string Unicode digunakan dan sumbu diluar luasan 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"\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"\u03c0"); ylabel(u"f(\u03c0)"):
```



## Memploting data 2D

---

Jika x dan y adalah vektor-vektor data, data ini akan digunakan sebagai koordinat x dan y dari kurva. Dalam kasus ini, a, b, c dan d atau sebuah radius r dapat dispesifikan atau jendela plot akan menyesuaikan data secara otomatis.

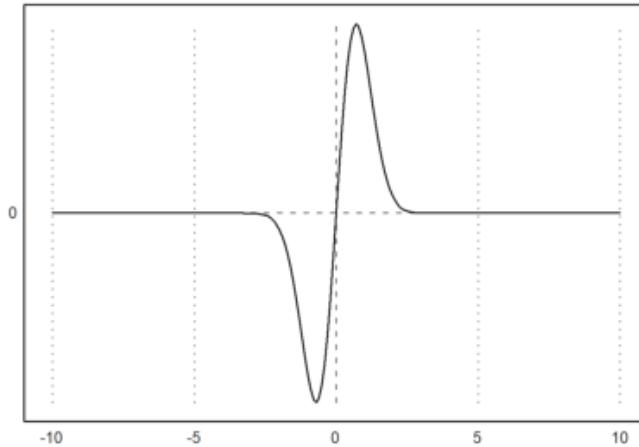
Membuat plot ekspresi hanya sebuah singkatan untuk plot data. Untuk plot data, kamu membutuhkan satu atau lebih barisan dari nilai-x dan satu atau lebih barisan dari nilai-y. Dari rentang dan nilai-x fungsi plot2d akan menghitung data untuk plot, secara bawaan dengan evaluasi adaptasi dari fungsi. Untuk plot-plot titik gunakan ">points", untuk campuran garis dan titik gunakan ">addpoints".

Tetapi kamu dapat memasukkan data secara langsung:

- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks-matriks untuk x dan y diplotkan garis dengan garis.

Ini adalah contoh untuk satu baris untuk x dan y.

```
>x = -10:0.1:10; y = exp(-x^2)*x; plot2d(x, y);
```



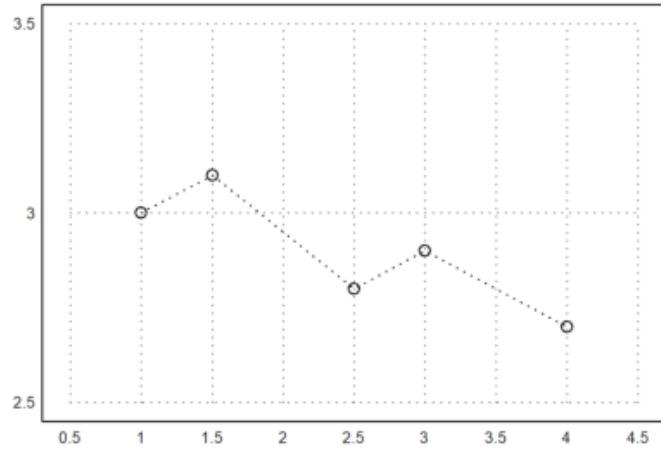
Data juga dapat diplotkan sebagai titik-titik. Gunakan `points=true` untuk melakukan ini. Plot akan bekerja seperti poligon, tetapi menggambar hanya sudut-sudutnya.

- `style = "...":` pilih dari `"[]", "<>", "o", ".," .., "+", "[]", "<>", "o", ".., "", "|".`

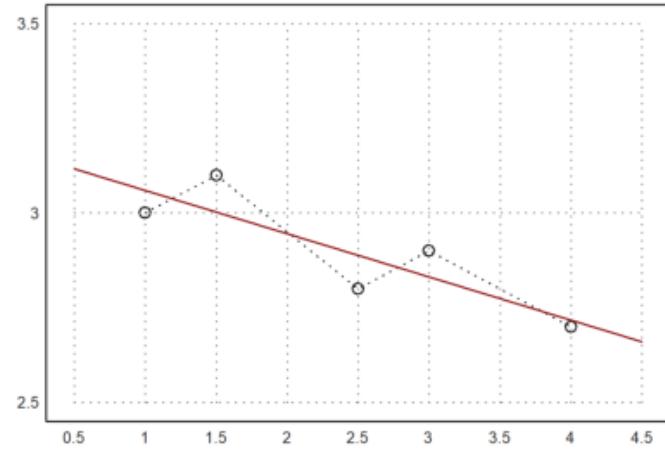
Untuk barisan plot dari titik-titik gunakan `>points`. Jika warna adalah sebuah vektor warna, setiap titik akan memiliki warna yang berbeda. Untuk sebuah matriks koordinat dan vektor kolom, warna diaplikasikan ke barisan dari matriks.

Parameter `>addpoint` menambahkan titik ke segmen bari untuk plot data.

```
>xdata = [1, 1.5, 2.5, 3, 4]; ydata = [3, 3.1, 2.8, 2.9, 2.7]; // data
>plot2d(xdata, ydata, a=0.5, b=4.5, c=2.5, d=3.5, style="."); // garis
>plot2d(xdata, ydata, >points, >add, style="o"); // menambahkan titik
```



```
>p = polyfit(xdata, ydata, 1); // mendapatkan garis regresi  
>plot2d("polyval(p, x)", >add, color=red); // menambah plot garis
```



## Menggambar Daerah yang dibatasi Kurva

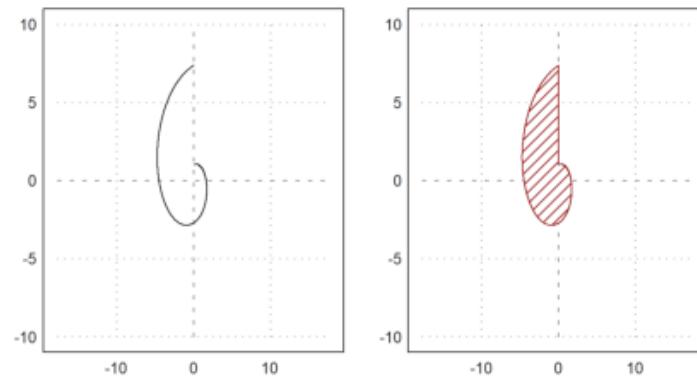
---

Plot-plot data memang benar-benar poligon. Kita dapat juga membuat plot kurva atau kurva yang berisi.

- filled = true, akan mengisi plot.
- style = "...": memilih dari "", "/", "\", "\\".
- fillcolor: Lihat atas untuk warna yang tersedia.

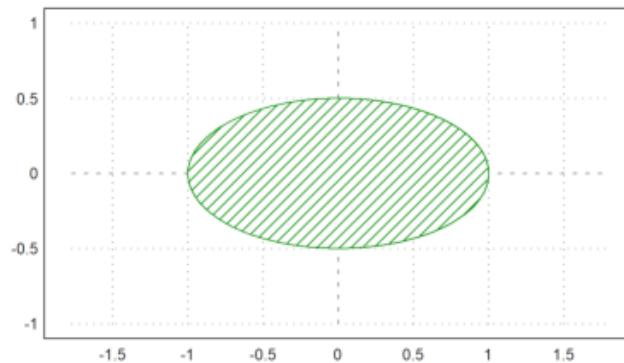
Warna isian dapat ditentukan dengan argumen "fillcolor" dan secara pilihan <outline mencegah membuat batasan dari setiap gaya tetapi salah satunya bawaan.

```
>t = linspace(0, 2pi, 1000); // parameter untuk kurva
>x = sin(t) * exp(t/pi); y = cos(t) * exp(t/pi); // x(t) dan y(t)
>figure(1, 2); aspect(16/9);
>figure(1); plot2d(x, y, r=10); // kurva plot
>figure(2); plot2d(x, y, r=10, >filled, style="/", fillcolor=red); // kurva isian
>figure(0):
```

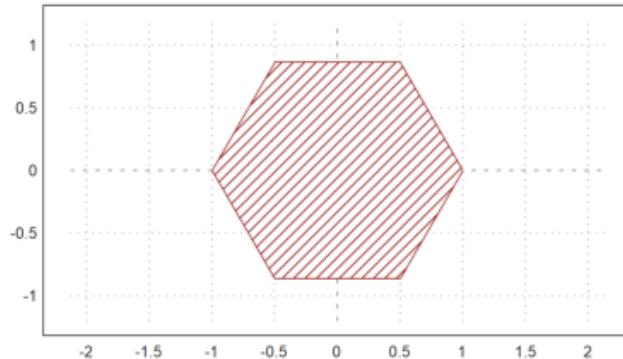


Contoh berikut kami membuat plot sebuah ellips yang terisi dan dua segidelapan yang terisi menggunakan kurva tertutup dengan 6 titik-titik dengan gaya isian yang berbeda.

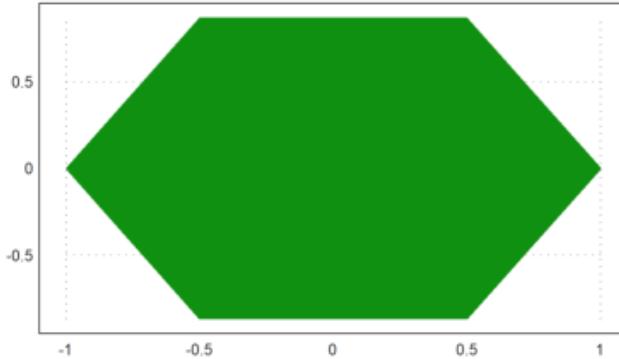
```
>x = linspace(0, 2pi, 1000); plot2d(sin(x), cos(x) * 0.5, r=1, >filled, style="/"):
```



```
>t = linspace(0, 2pi, 6); ...
>plot2d(cos(t), sin(t), >filled, style="/", fillcolor=red, r=1.2):
```

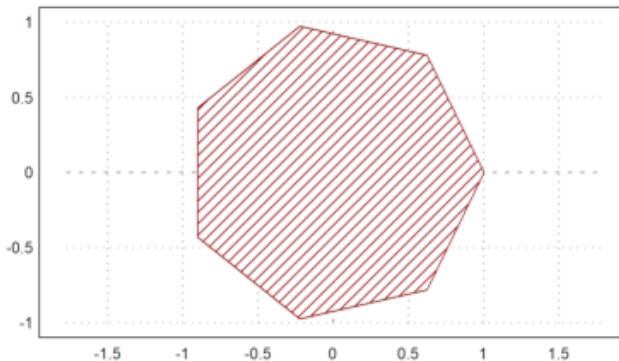


```
>t = linspace(0, 2pi, 6); plot2d(cos(t), sin(t), >filled, style="#"):
```



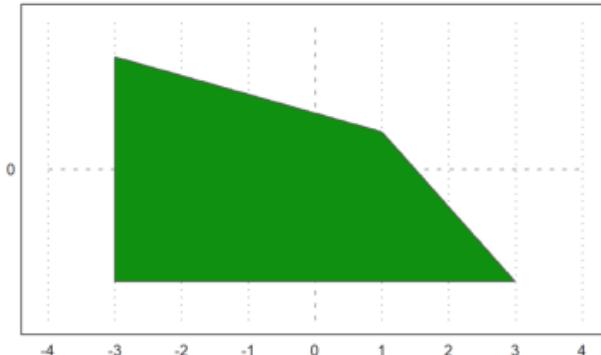
Contoh lainnya adalah sebuah segi-7, yang mana kami membuat 7 titik-titik dalam lingkaran.

```
>t = linspace(0, 2pi, 7); ...
>plot2d(cos(t), sin(t), r=1, >filled, style="/", fillcolor=red):
```



Contoh ini adalah barisan dari nilai maksimal dari empat kondisi linear kurang atau sama dengan 3. Ini adalah  $A[k].v \leq 3$  untuk semua barisan A. Untuk membuat sudut yang bagus, kita menggunakan n dengan relatif yang 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);
```

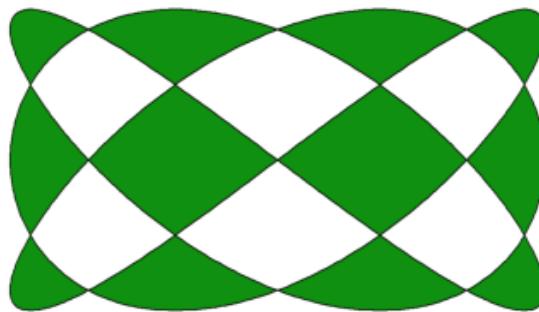


Titik utama dari bahasa matriks adalah membiarkannya untuk membuat tabel dari fungsi secara mudah.

```
>t = linspace(0, 2pi, 1000); x = cos(3*t); y = sin(4*t);
```

Kami sekarang memiliki vektor x dan y dari nilai-nilai. `plot2d()` dapat membuat plot nilai-nilai ini sebagai sebuah kurva yang terkoneksi dengan titik-titik. Plot dapat diisi. Dalam kasus ini menghasilkan hasil yang bagus karena aturan berlaku, yang mana digunakan untuk pengisian.

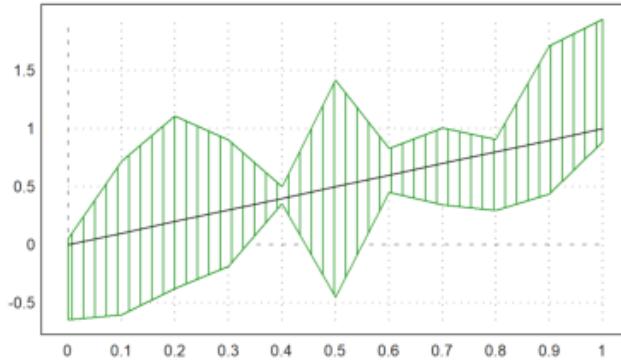
```
>plot2d(x, y, <grid, <frame, >filled):
```



Sebuah vektor dari interval-interval diplotkan dengan nilai-nilai x sebagai wilayah antara batas bawah dan batas atas dari interval.

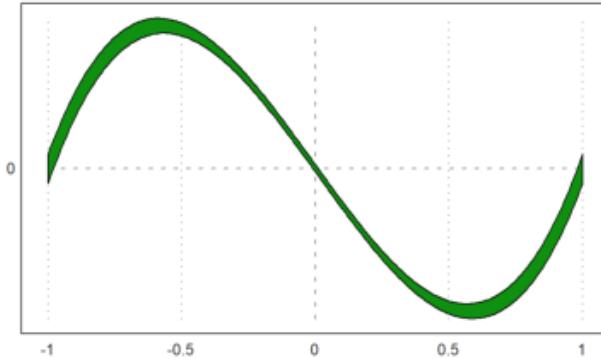
Ini dapat berguna untuk membuat galat plot dari komputasi. Tapi ini dapat juga digunakan untuk galat statistik.

```
>t = 0:0.1:1;...
>plot2d(t, interval(t - random(size(t)), t + random(size(t))), style="|"); ...
>plot2d(t, t, >add):
```



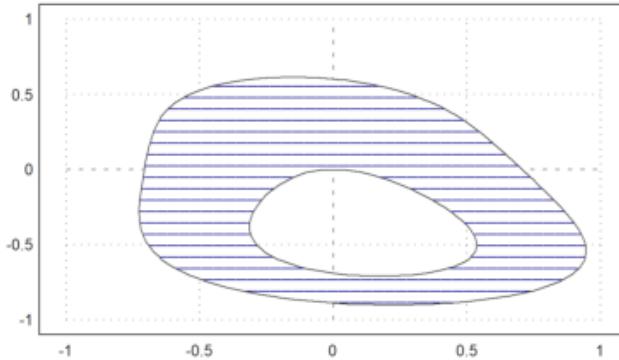
Jika  $x$  adalah vektor yang terurut dan  $y$  adalah interval vektor, lalu `plot2d` akan diplotkan pada rentang yang terisi dari interval dalam bidang. Gaya-gaya pengisian sama seperti gaya-gaya dari poligon.

```
>t = -1:0.01:1; x=t - 0.01, t + 0.01; y = x^3-x;  
>plot2d(t, y):
```



Sangat mungkin untuk mengisi nilai nilai untuk sebuah fungsi speifik. Untuk ini, level harus sebuah matriks 2xn. Baris pertama adalah batas bawah dan baris kedua mencakup batas atas.

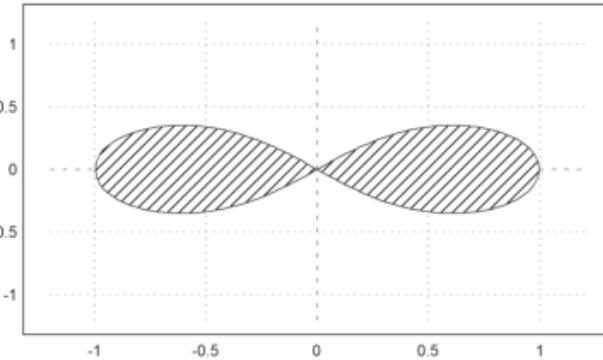
```
>expr := "2*x^2+x*y+3*y^4+y"; // mendefinisikan ekspresi f(x, y)
>plot2d(expr, level=[0; 1], style="-", color=blue); // 0 <= f(x, y) <= 1
```



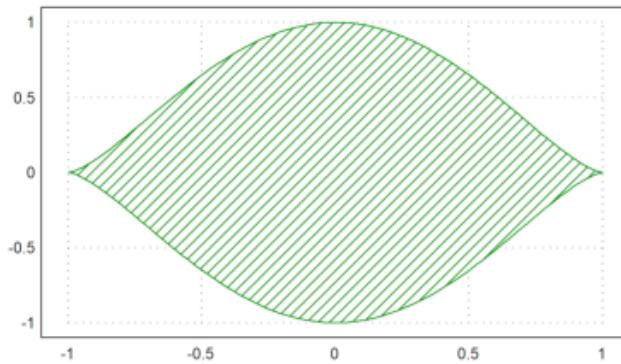
Kami dapat juga mengisi rentang dengan nilai seperti

$$-1 \leq (x^2 + y^2)^2 - x^2 + y^2 \leq 0$$

```
>plot2d("(x^2 + y^2)^2 - x^2 + y^2", r=1.2, level=[-1; 0], style="/"):
```



```
>plot2d("cos(x)", "sin(x)^3", xmin=0, xmax=2pi, >filled, style="/"):
```



## Grafik Fungsi Parametrik

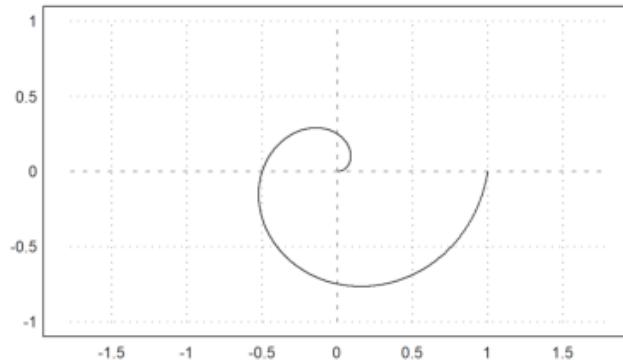
---

Nilai x butuh tidak diurutkan. (x, y) secara sederhana menggambarkan sebuah kurva. Jika x terurut, kurva adalah sebuah grafik dari sebuah fungsi. Contoh ini, kami akan membuat plot spiral

$$\gamma(t) = t \cdot (\cos(2\pi t), \sin(2\pi t))$$

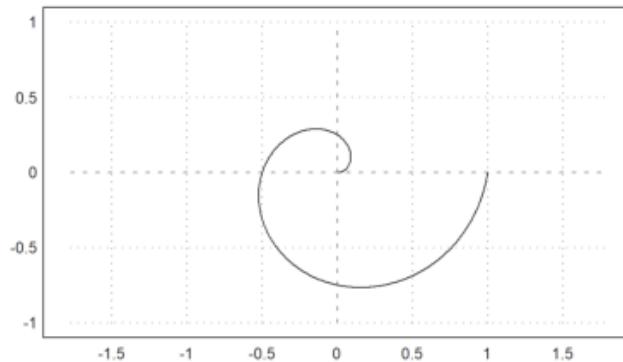
Kami juga membutuhkan banyak titik-titik untuk memperhalus atau fungsi adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk lebih detail).

```
>t = linspace(0, 1, 1000); ...
>plot2d(t * cos(2*pi*t), t*sin(2*pi*t), r=1):
```

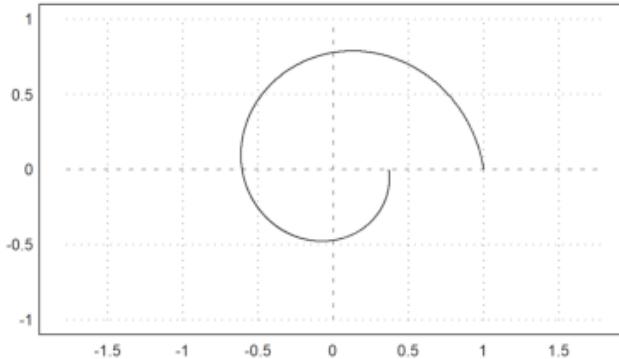


Sebagai alternatif, yang memungkinkan untuk menggunakan 2 ekspresi untuk kurva. Contoh plot ini sama seperti kurva diatas.

```
>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):
```



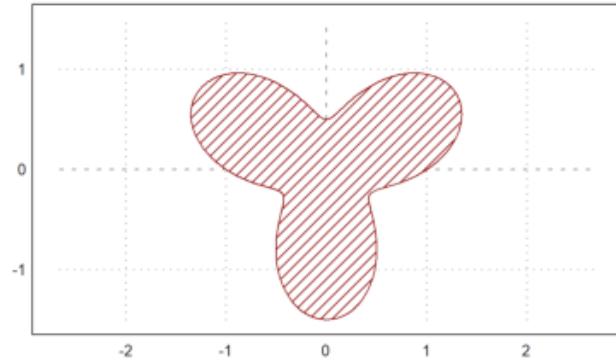
Contoh selanjutnya, kami membuat plot dari kurva

$$\gamma(t) = (r(t)\cos(t), r(t)\sin(t))$$

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}$$

```
>t = linspace(0, 2pi, 1000); r = 1+sin(3*t) / 2; x = r*cos(t); y = r*sin(t); ...
>plot2d(x, y, >filled, fillcolor=red, style="/", r=1.5);
```



## Menggambar Grafik Bilangan Kompleks

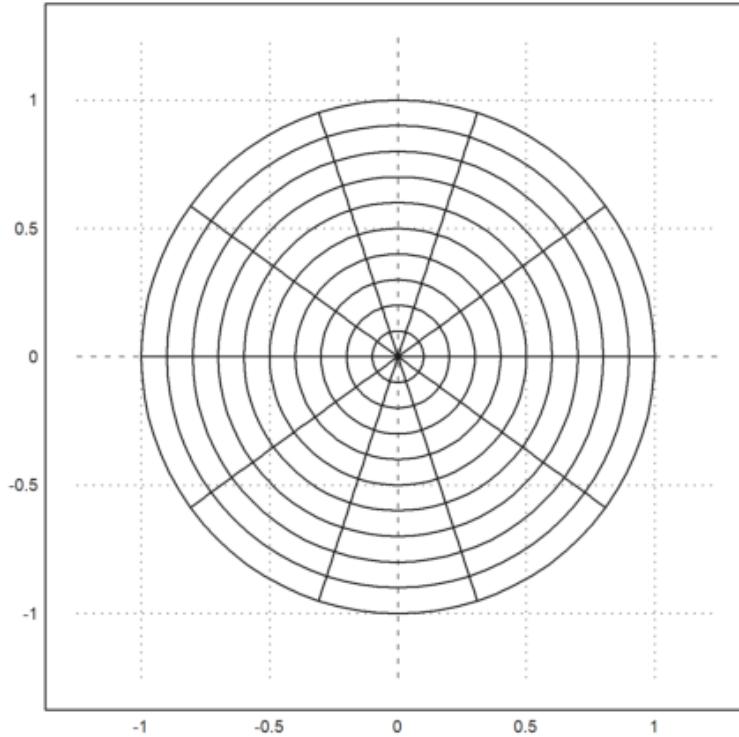
---

Sebuah array dari angka-angka kompleks juga dapat dibuat plot. Lalu titik-titik batas akan terkoneksi. Jika sebuah angkka dari batas garis yang dispesifikasikan (atau sebuah 1x2 vektor dari batas garis) dalam sebuah argumen cgrid hanya terlihat baris batas ini.

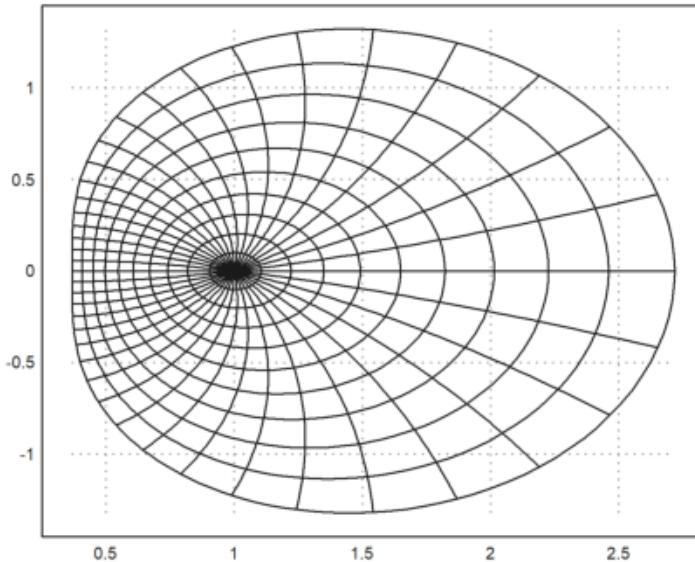
Sebuah matriks dari angka-angka kompleks akan secara otomatis diplotkan sebagai sebuah batas dari bidang kompleks.

Sebagai contoh, kami membuat plot gambar dengan lingkaran unit dari fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva batasnya.

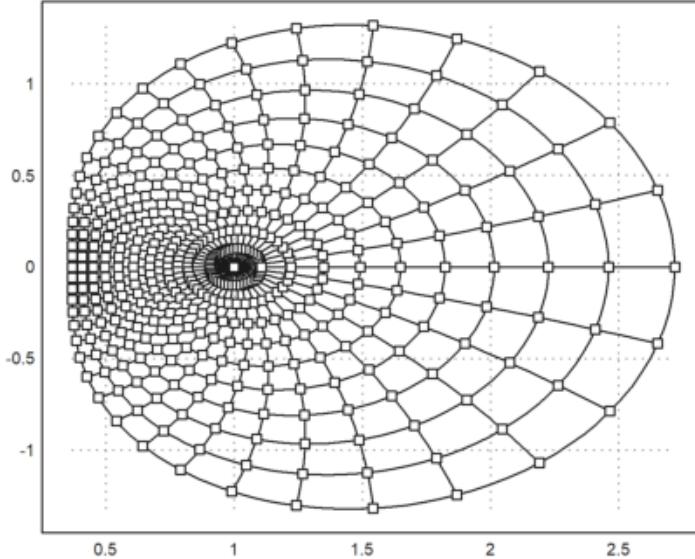
```
>aspect(); r = linspace(0, 1, 50); a = linspace(0, 2pi, 80)'; z = r*exp(I*a); ...
>plot2d(z, a=-1.25, b=1.25, c=-1.25, d=1.25, cgrid=10);
```



```
>aspect(1.25); r = linspace(0, 1, 50); a = linspace(0, 2pi, 200)'; z = r*exp(I*a);
>plot2d(exp(z), cgrid=[40, 10]):
```



```
>r = linspace(0, 1, 10); a = linspace(0, 2pi, 40)'; z = r*exp(I*a);
>plot2d(exp(z), >points, >add):
```

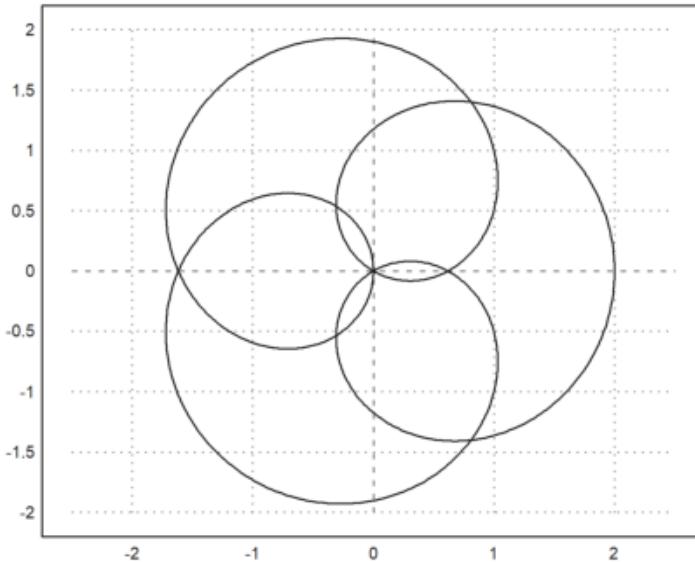


Sebuah vektor dari angka-angka kompleks secara otomatis terplotkan sebagai sebuah kurva dalam bidang kompleks dengan bagian rill dan bagian imajiner.

Contoh berikut, kami membuat plot dengan unit lingkaran dengan

$$\gamma(t) = e^{it}$$

```
>t = linspace(0, 2pi, 1000); ...
>plot2d(exp(I*t) + exp(4*I*t), r=2):
```



## Plot-Plot Statistik

---

Terdapat banyak fungsi yang terspesialisasikan untuk plot-plot statistik. Salah satu yang sering digunakan adalah plot kolom.

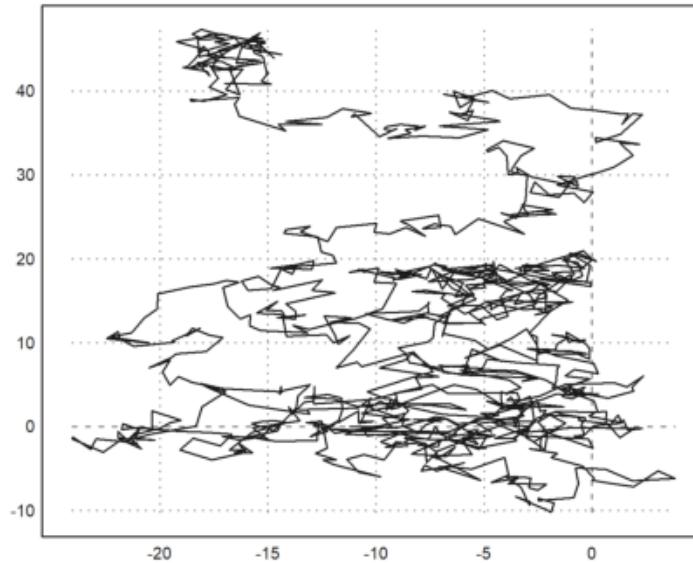
Sebuah jumlahan kumulatif dari sebuah 0-1 distribusi nilai normal menghasilkan pergerakan secara acak.

```
>plot2d(cumsum(randnormal(1, 1000))):
```

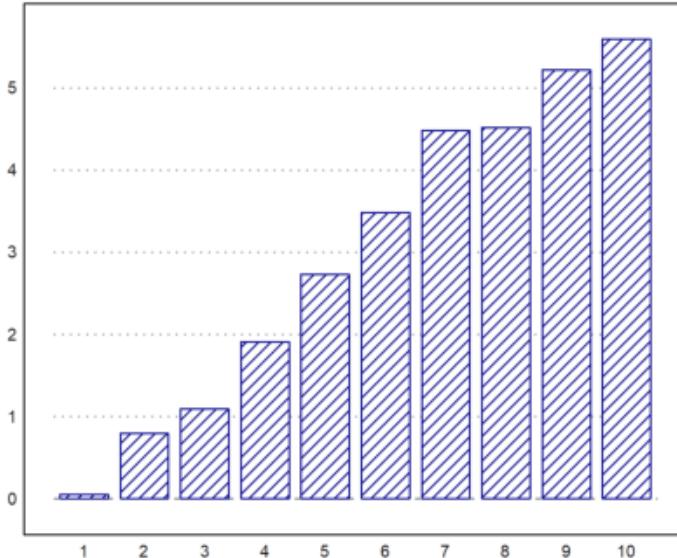


Menggunakan dua baris untuk menunjukkan dua dimensi.

```
>X = cumsum(randnormal(2, 1000)); plot2d(X[1], X[2]):
```

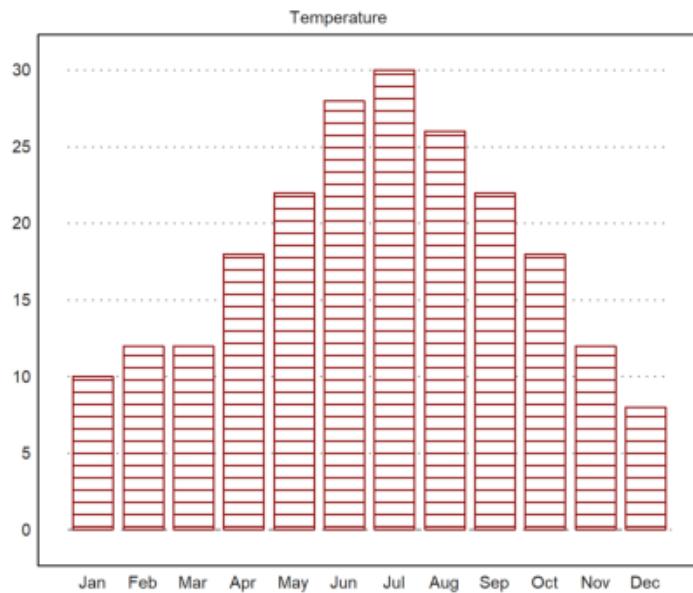


```
>columnsplot(cumsum(random(10)), style="/", color=blue):
```

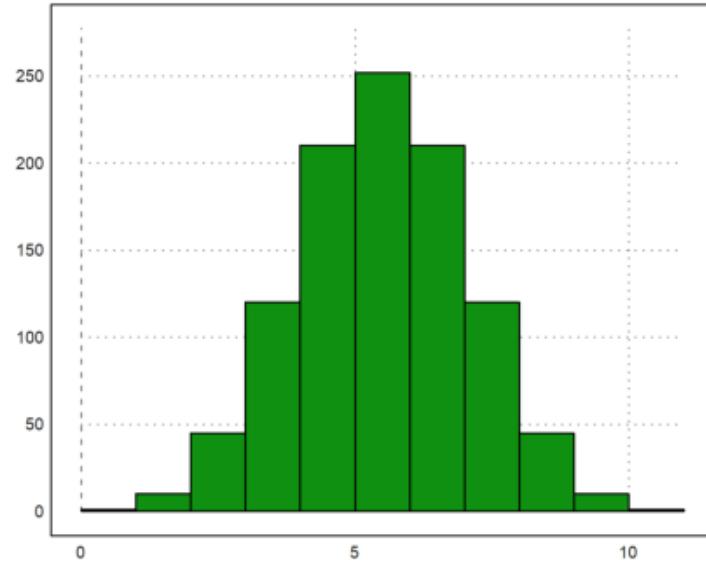


Ini dapat juga menunjukkan string sebagai label.

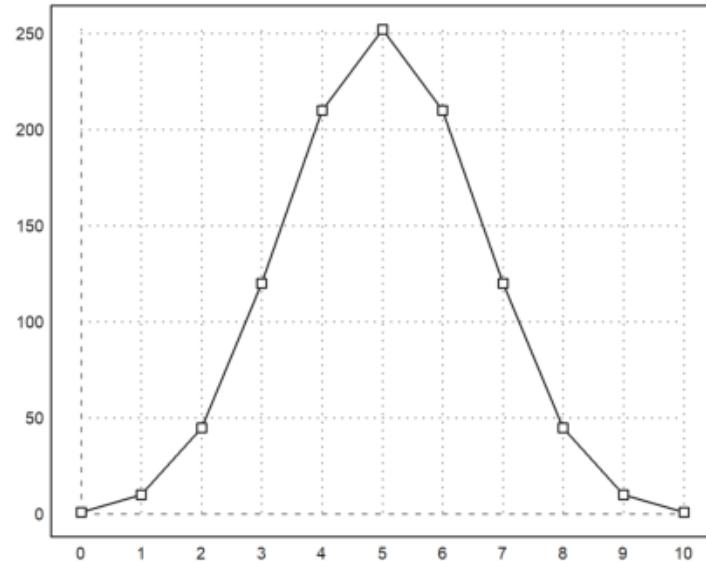
```
>months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", ...
>"Jul", "Aug", "Sep", "Oct", "Nov", "Dec"];
>values = [10, 12, 12, 18, 22, 28, 30, 26, 22, 18, 12, 8];
>columnsplot(values, lab=months, color=red, style="-");
>title("Temperature");
```



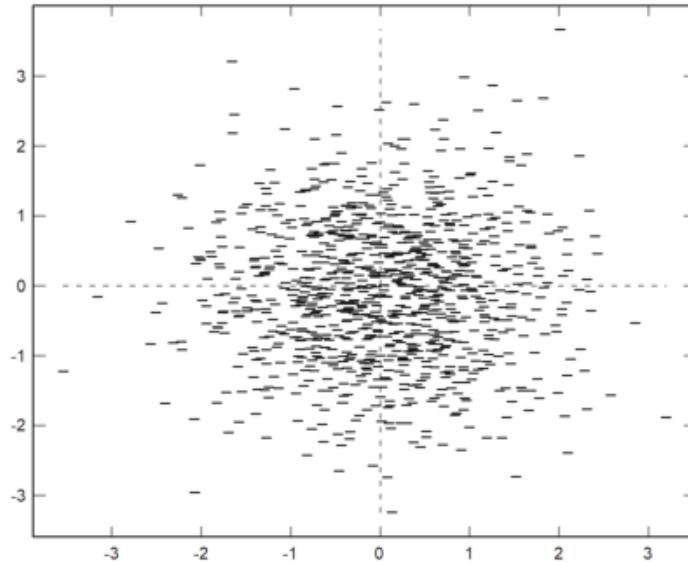
```
>k = 0:10;  
>plot2d(k, bin(10, k), >bar):
```



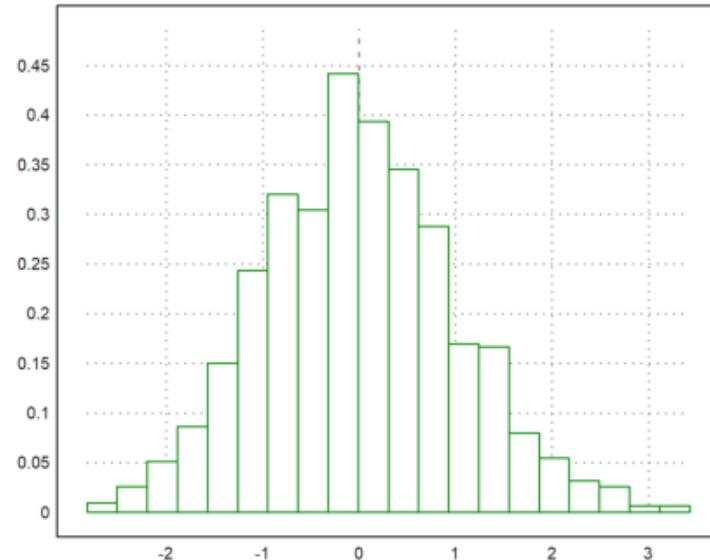
```
>plot2d(k, bin(10, k)); plot2d(k, bin(10, k), >points, >add):
```



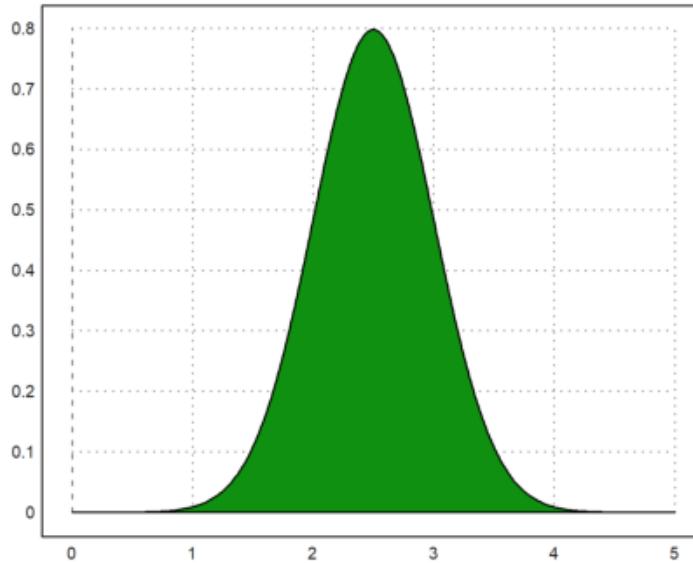
```
>plot2d(normal(1000), normal(1000), >points, grid=6, style="-"):
```



```
>plot2d(normal(1, 1000), >distribution, style="0"):
```

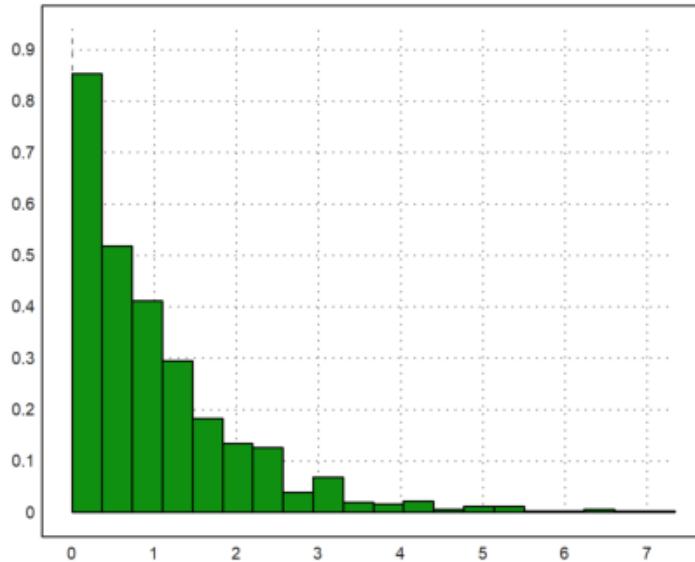


```
>plot2d("qnormal", 0, 5; 2.5, 0.5, >filled):
```



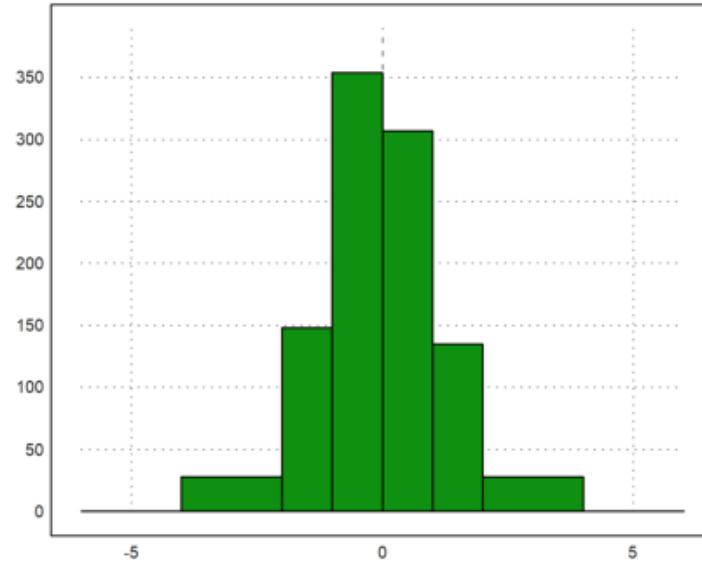
Untuk plot sebuah distribusi eksperimen statistik, kamu juga dapat menggunakan distribution=n dengan plot2d.

```
>w = randexponential(1, 1000); // distribusi eksponensial  
>plot2d(w, >distribution); // atau distribution=n dengan interval n
```



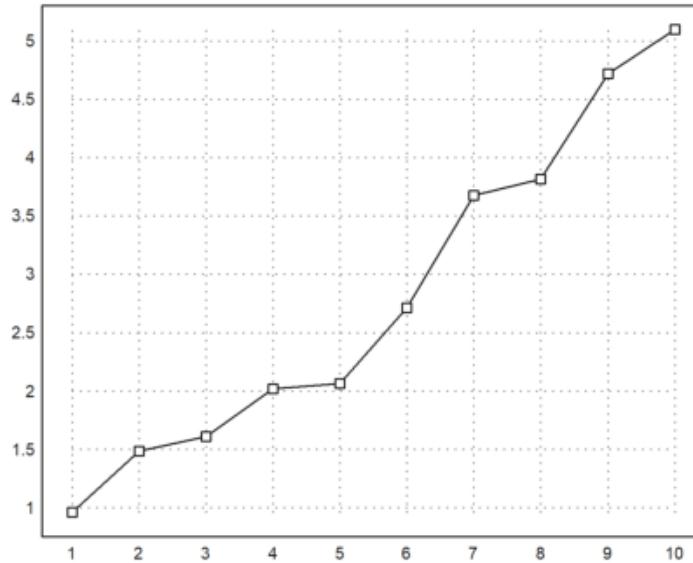
Atau kamu juga menghitung bentuk distribusi data dan plot hasil dengan >bar dalam plot3d, atau dengan sebuah plot kolom.

```
>w = normal(1000); // 0-1-distribusi normal
>{x, y} = histo(w, 10, v=[-6, -4, -2, -1, 0, 1, 2, 4, 6]); // batas interval v
>plot2d(x, y, >bar):
```

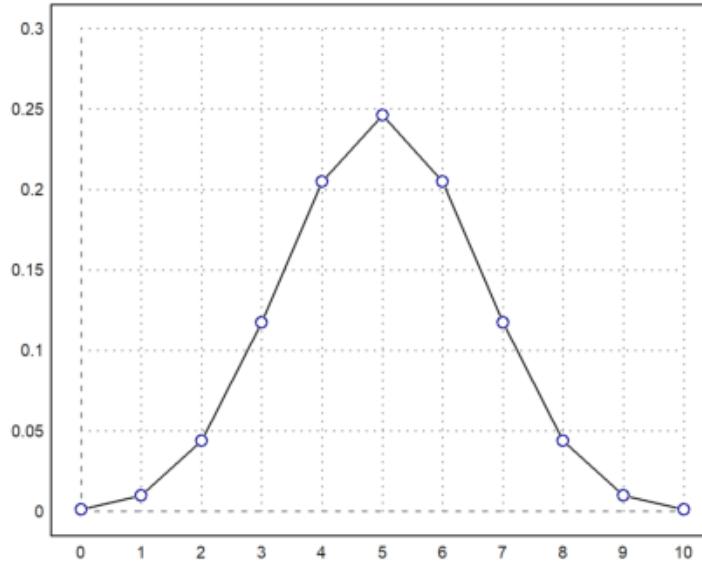


fungsi statplot() mengatur gaya dengan string yang sederhana.

```
>statplot(1:10, cumsum(random(10)), "b"):
```



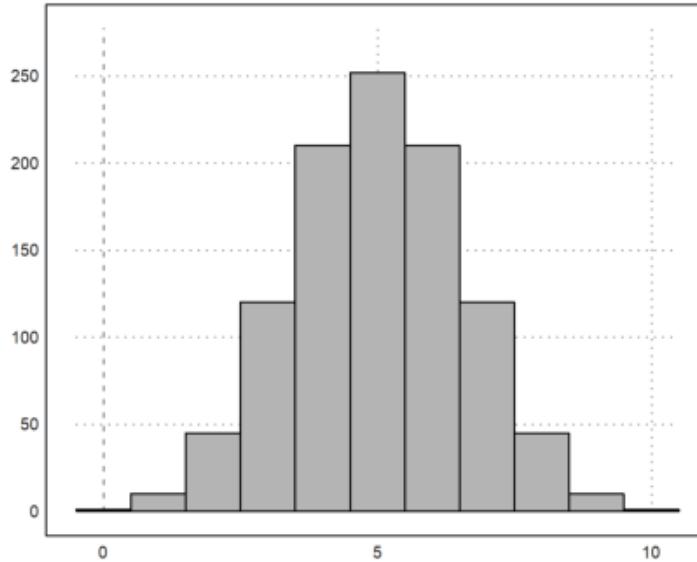
```
>n = 10; i = 0:n; ...
>plot2d(i, bin(n, i) / 2^n, a=0, b=10, c=0, d=0.3);...
>plot2d(i, bin(n, i) / 2^n, >points, >add, style="ow", color=blue):
```



Selebihnya, data dapat diplotkan sebagai baris. Dalam kasus ini, x akan diurutkan dan satu elemen lebih panjang daripada y. Barang akan memanjang dari  $x[i]$  ke  $x[i+1]$  dengan nilai  $y[i]$ . Jika x memiliki ukuran yang sama seperti y, akan memanjang dengan satu elemen dengan spasi dibagian akhir.

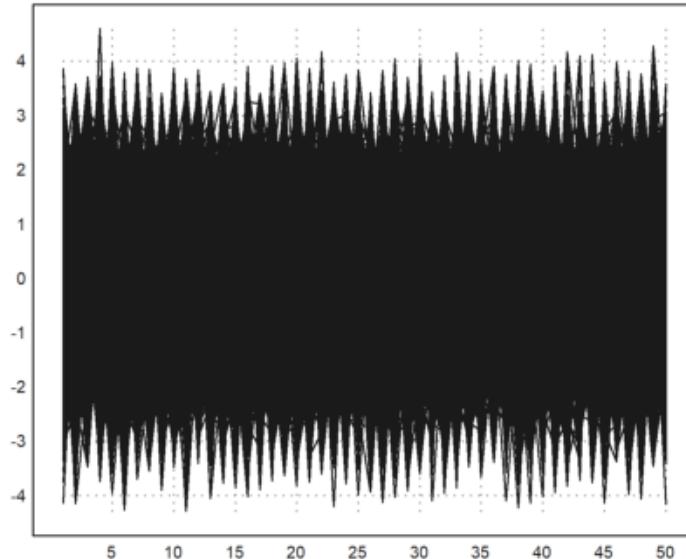
Gaya isian akan digunakan seperti diatas.

```
>n = 10; k = bin(n, 0:n); ...
>plot2d(-0.5:n + 0.5, k, >bar, fillcolor=lightgray):
```

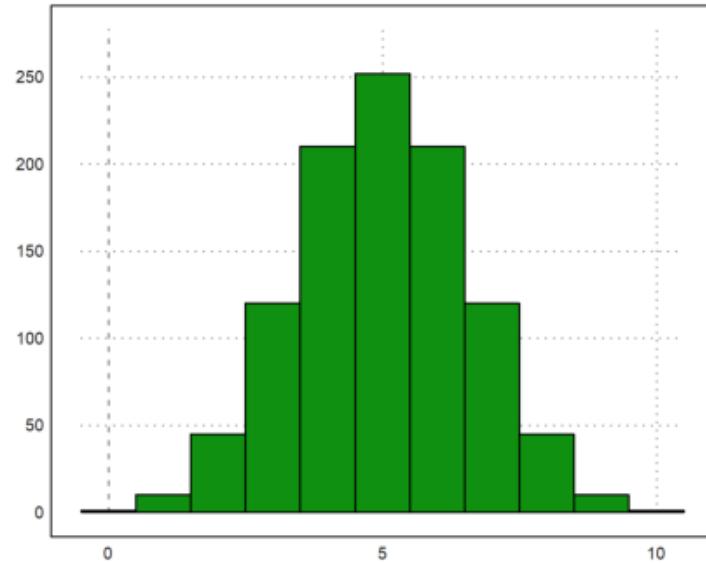


Data untuk plot batang (bar = 1) dan histogram (histogram = 1) dapat secara eksplisit diberikan dalam xv dan yv atau dapat dihitung dari sebuah distribusi empiris dalam xv dengan >distribution (or distribution = n). Histogram dari nilai xv dapat dihitung secara otomatis dengan >histogram. Jika >even secara spesifik, nilai xv akan dihitung interval bilangan bulat.

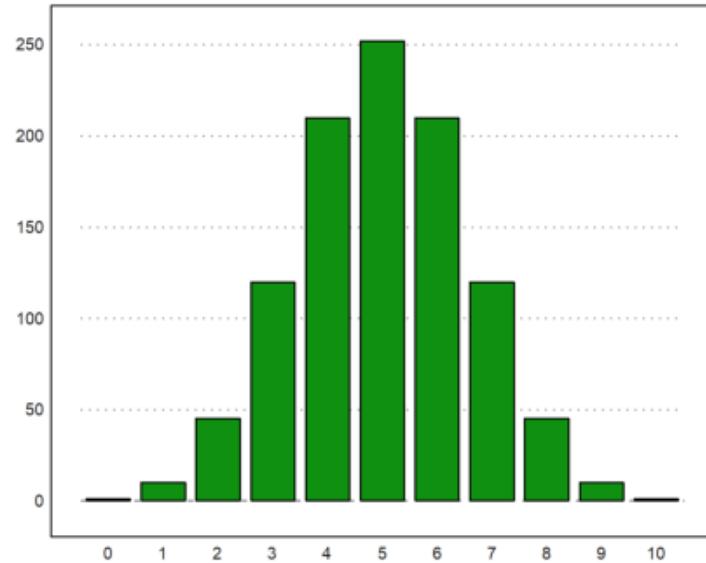
```
>plot2d(normal(10000, distribution=50)):
```



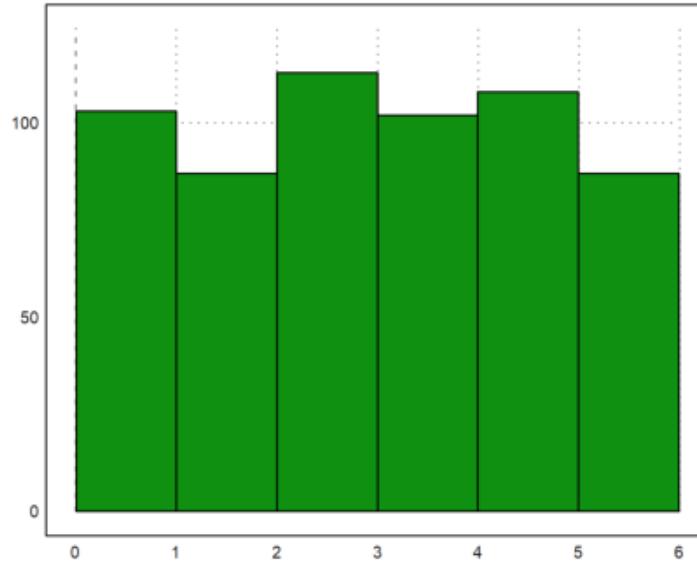
```
>k = 0:10; m = bin(10, k); x = (0:11)-0.5; plot2d(x, m, >bar):
```



```
>columnsplot(m, k):
```

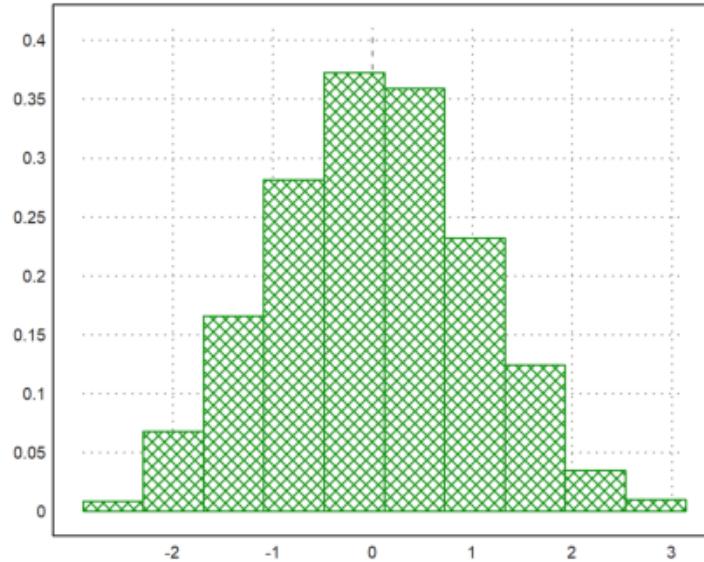


```
>plot2d(random(600) * 6, histogram=6):
```



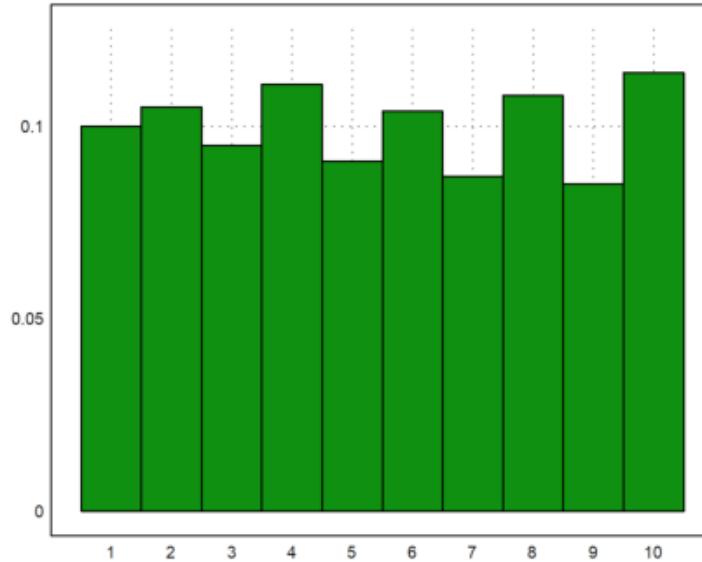
Untuk distribusi, parameter distribusi = n ini, yang mana menghitung nilai-nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

```
>plot2d(normal(1,1000),distribution=10,style="\/"):
```



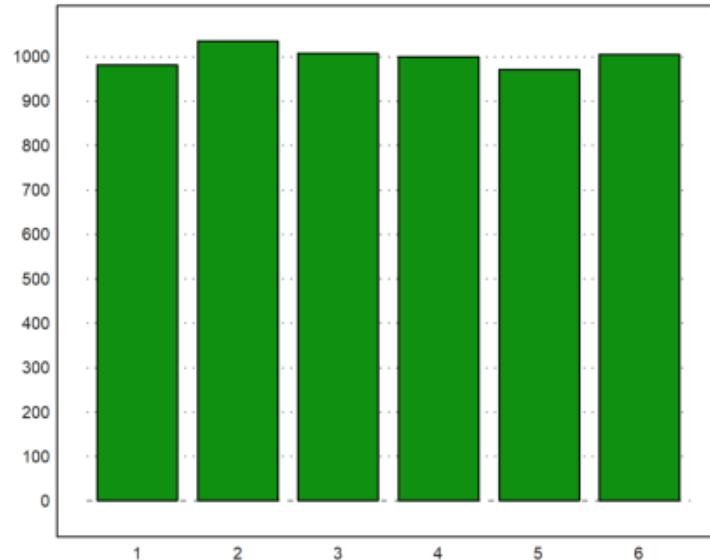
Dengan parameter even=true, ini akan menggunakan interval bilangan bulat.

```
>plot2d(intrandom(1, 1000, 10), distribution=10, >even):
```

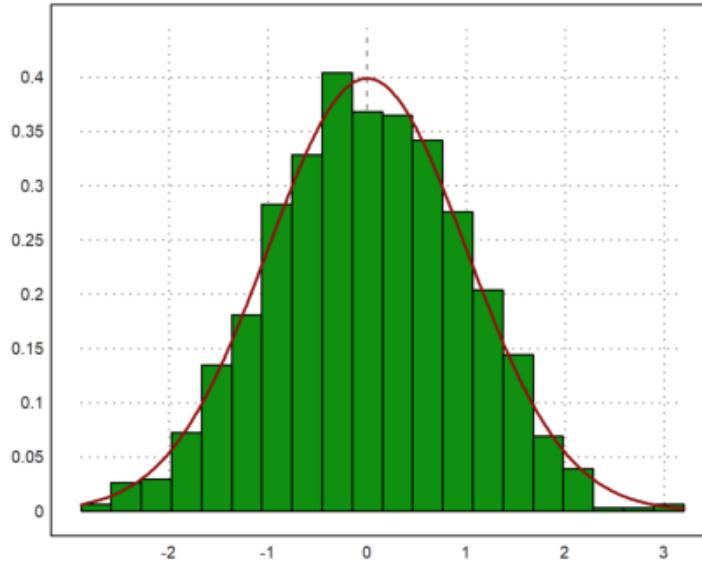


Catat bahwa terdapat banyak plot statistik, yang mana mungkin akan berguna. Lihatlah tutorial tentang statistik.

```
>columnsplot(getmultiplicities(1:6, intrandom(1, 6000, 6))):
```

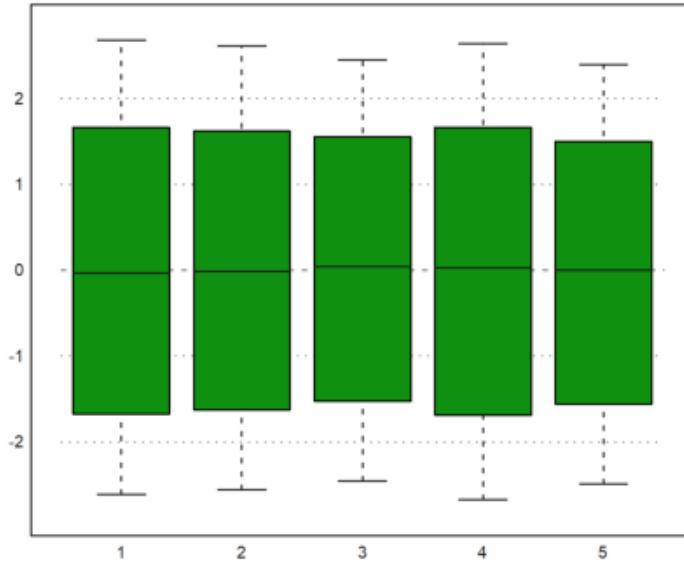


```
>plot2d(normal(1, 1000), >distribution); ...
>plot2d("qnormal(x)", color=red, thickness=2, >add):
```



Ini juga terdapat banyak plot spesial untuk statistik. Sebuah boxplot memperlihatkan kuartil dari distribusi ini dan banyak luaran. Dengan definisi, luaran dalam sebuah boxplot adalah data yang melebihi 1.5 kali dari tengahan 50% rentang dari plot.

```
>M = normal(5, 1000); boxplot(quartiles(M)):
```



## Fungsi-Fungsi Implisit

---

Plot implisit menunjukkan barisan level dengan menyelesaikan  $f(x, y) = \text{level}$ , dimana "level" dapat berupa nilai tunggal atau sebuah vektor dari nilai. Jika  $\text{level}=\text{"auto"}$ , ini akan menjadi nc baris level, yang mana akan menyebar diantara nilai minimum dan maksimum dari fungsi. Gelap atau terangnya warna dapat ditambahkan dengan `>huw` untuk menunjukkan nilai dari fungsi. Untuk fungsi implisit, xv harus menjadi sebuah fungsi dari parameter x dan y, atau, secara alternatif, xv dapat menjadi sebuah matriks dari nilai.

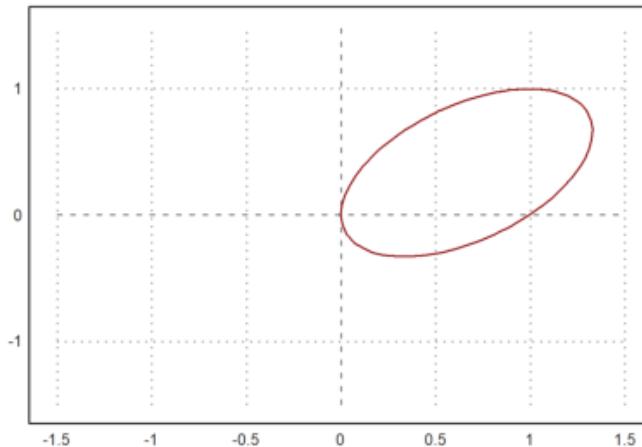
Euler dapat menandai garis level

$$f(x, y) = c$$

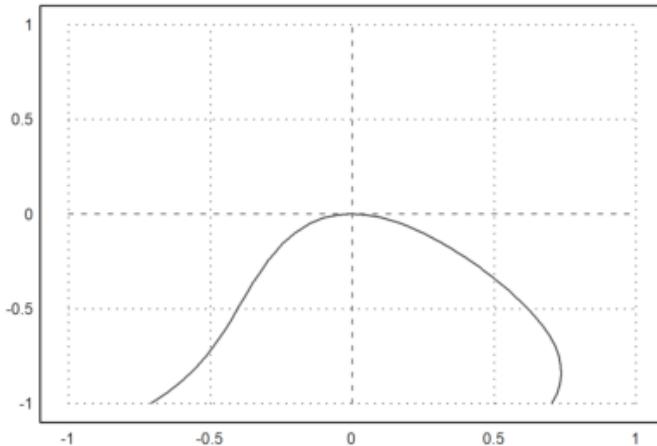
dari sebarang fungsi

Untuk menggambar himpunan  $f(x, y) = c$  untuk satu atau lebih konstan  $c$  kamu dapat menggunakan `plot2d()` dengan ini plot implisit dalam bidang. Parameter untuk  $c$  adalah `level = c`, dimana  $c$  dapat berupa vektor dari baris level. Dengan tambahan, sebuah skema warna dapat digambarkan dalam latar belakang untuk mengindikasi nilai dari fungsi untuk setiap titik dalam plot. Parameter "n" menentukan kemulusan dari plot.

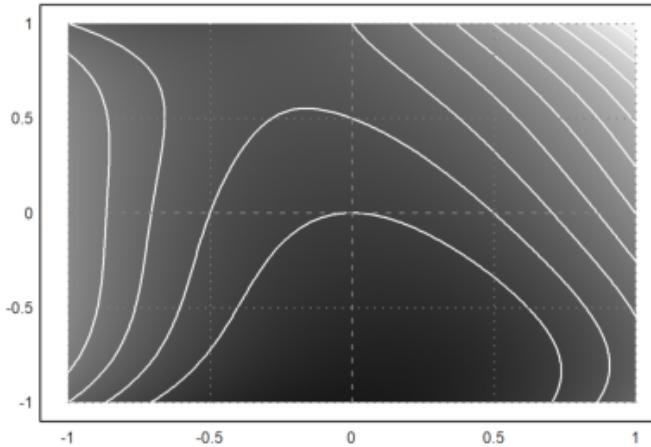
```
>aspect(1.5);
>plot2d("x^2 + y^2 - x*y - x", r=1.5, level=0, contourcolor=red);
```



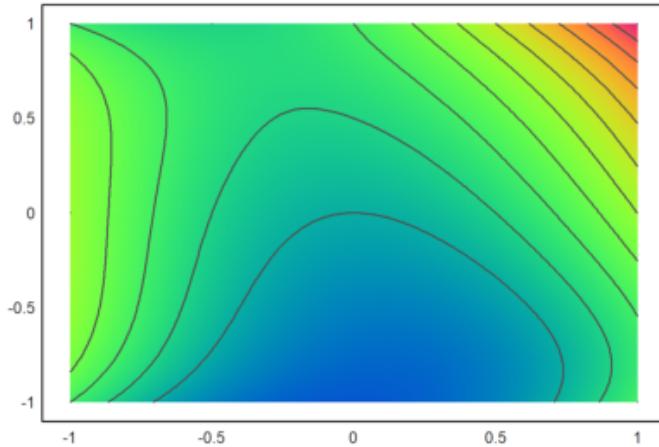
```
>expr := "2*x^2 + x*y + x*y^4 + y"; // mendefinisikan sebuah ekspresi f(x, y)
>plot2d(expr, level=0); // solusi dari f(x, y) = 0
```



```
>plot2d(expr, level=0:0.5:20, >hue, contourcolor=white, n=200); // bagus
```

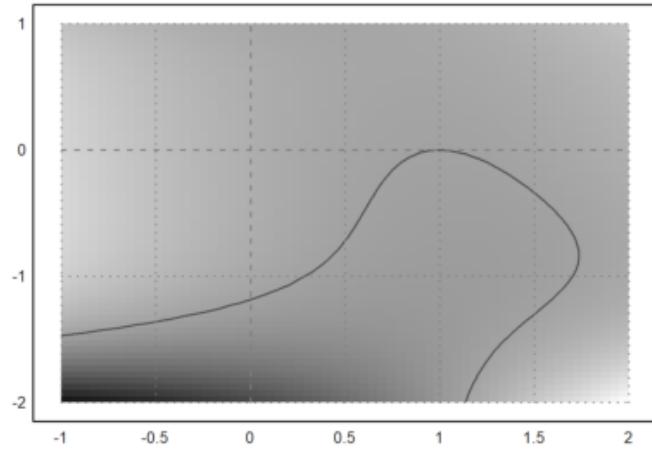


```
>plot2d(expr, level=0:0.5:20, >hue, >spectral, n=200, grid=4): // lebih bagus
```

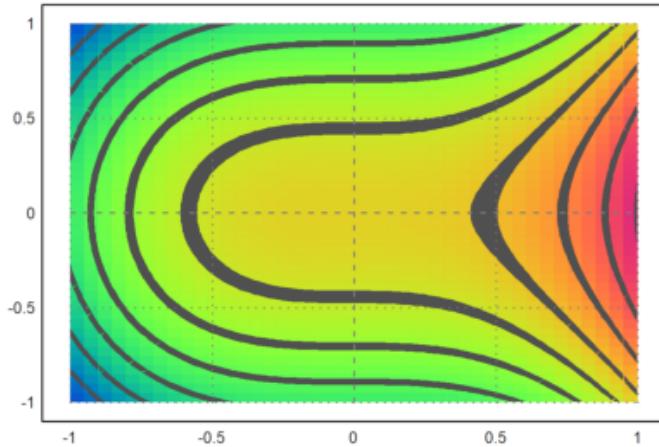


Ini bekerja untuk data plot juga. Tetapi kamu harus menspesifikasiikan rentang untuk sumbu-sumbu label.

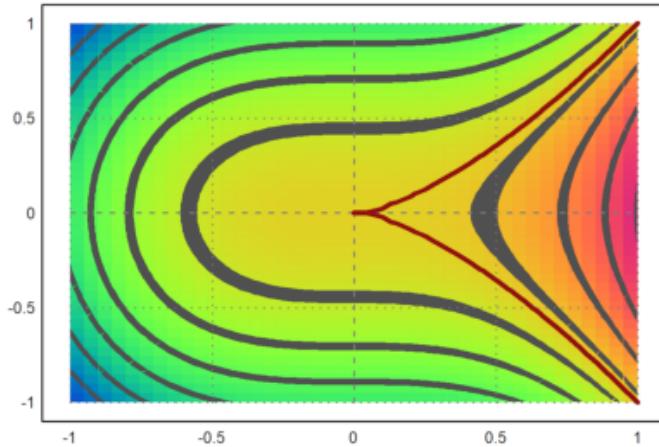
```
>x = -2:0.05:1; y = x'; z = expr(x, y);
>plot2d(z, level=0, a=-1, b=2, c=-2, d=1, >hue):
```



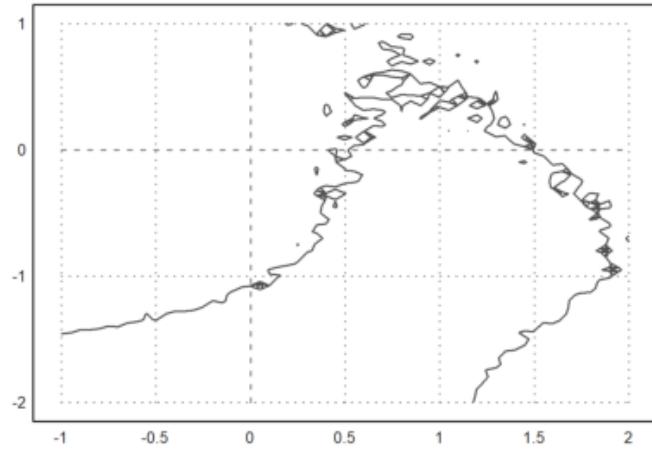
```
>plot2d("x^3-y^2", >contour, >hue, >spectral):
```



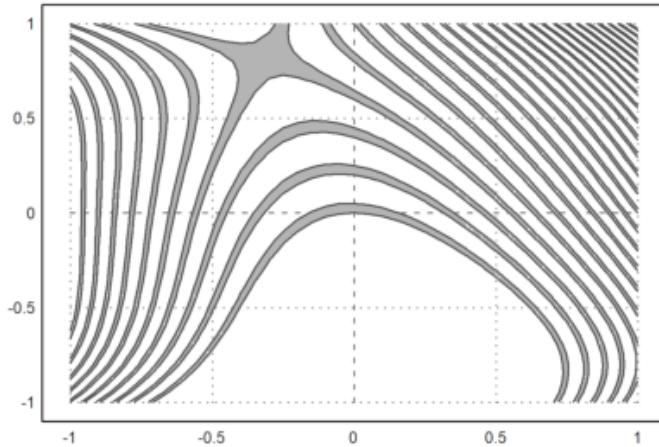
```
>plot2d("x^3-y^2", level=0, contourwidth=3, >add, contourcolor=red):
```



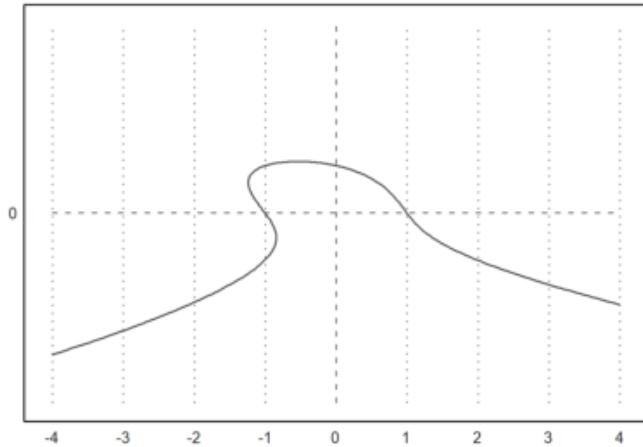
```
>z = z + normal(size(z)) * 0.2;  
>plot2d(z, level=0.5, a=-1, b=2, c=-2, d=1):
```



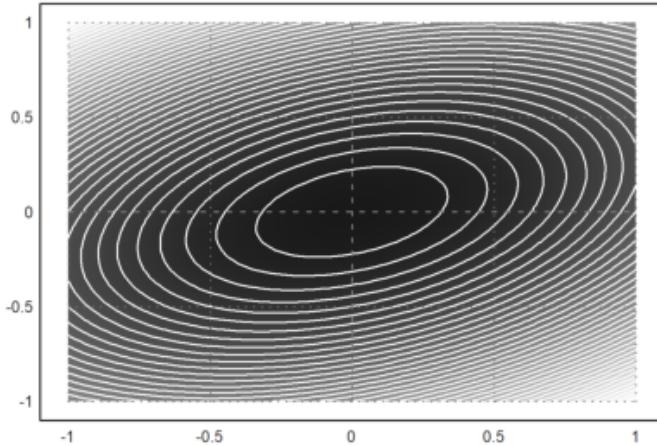
```
>plot2d(expr, level=[0:0.2:5; 0.05:0.2:5.05], color=lightgray):
```



```
>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):
```



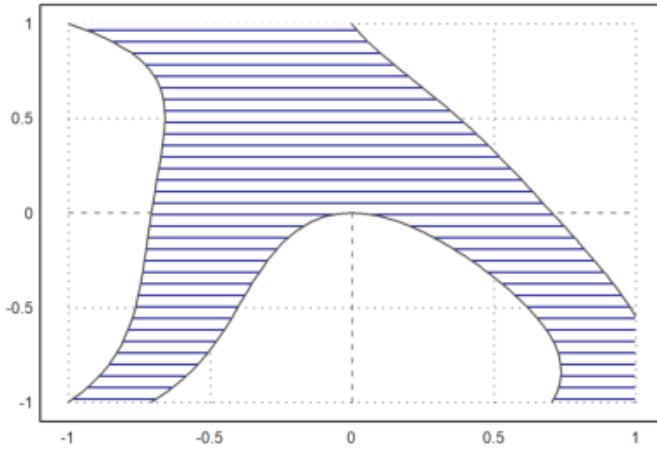
Ini juga memungkinkan untuk mengisi himpunan

$$a \leq f(x, y) \leq b$$

dengan sebuah rentang level.

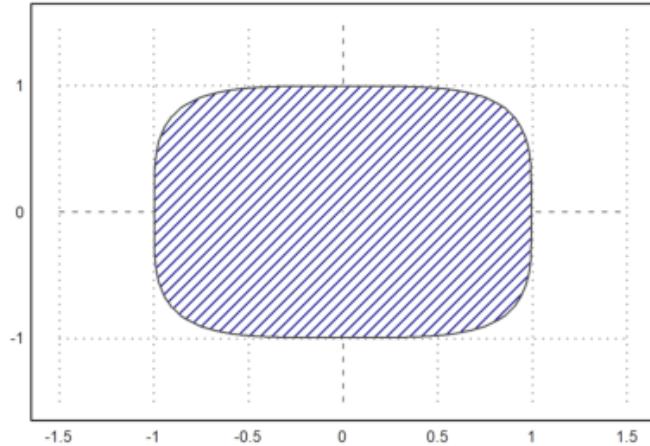
Ini memungkinkan untuk mengisi wilayah dari nilai-nilai untuk fungsi tertentu. Untuk ini, level haruslah sebuah matriks 2xn. Baris pertama adalah batasan bawah dan baris kedua mengandung batasan atas.

```
>plot2d(expr, level=[0; 1], style="-", color=blue); // 0 <= f(x, y) <= 1
```

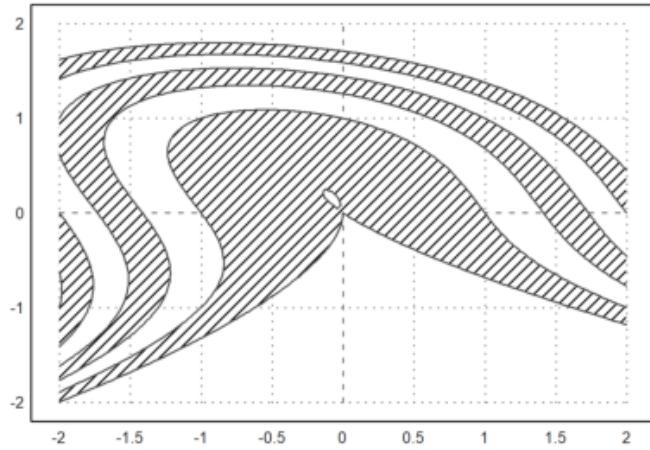


Plot implisit juga dapat memperlihatkan rentang dari level. Lalu level haruslah sebuah matriks 2xn dari interval level, dimana baris pertama mengandung awal dan baris kedua adalah akhir dari setiap interval. Sebagai gantinya, sebuah baris vektor dapat digunakan untuk level, dan sebuah dl parameter meluaskan nilai level untuk interval.

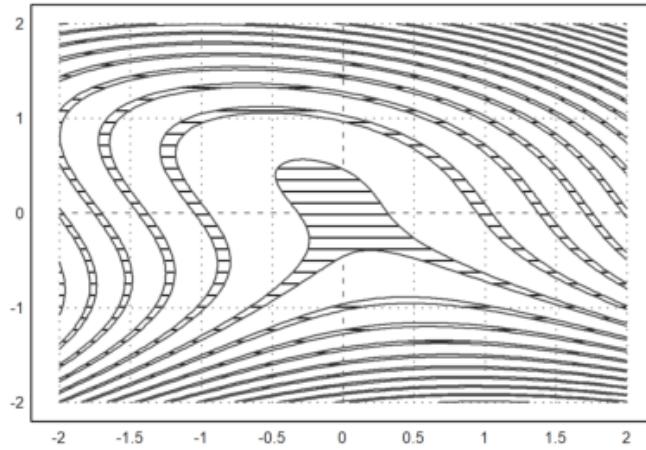
```
>plot2d("x^4 + y^4", r=1.5, level=[0;1], color=blue, style="/"):
```



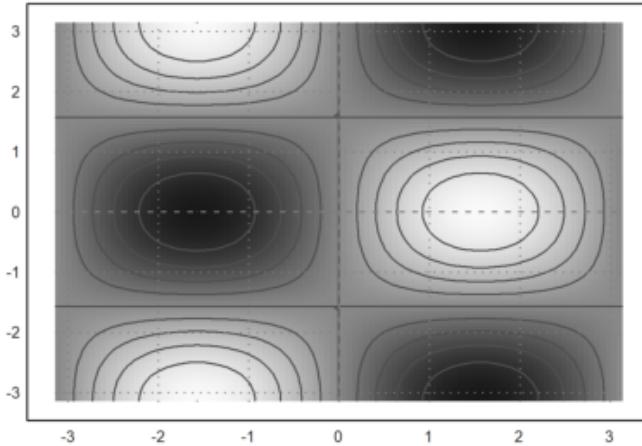
```
>plot2d("x^2 + y^3 + x*y", level=[0, 2, 4; 1, 3, 5], style="/", r=2, n=100):
```



```
>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):
```

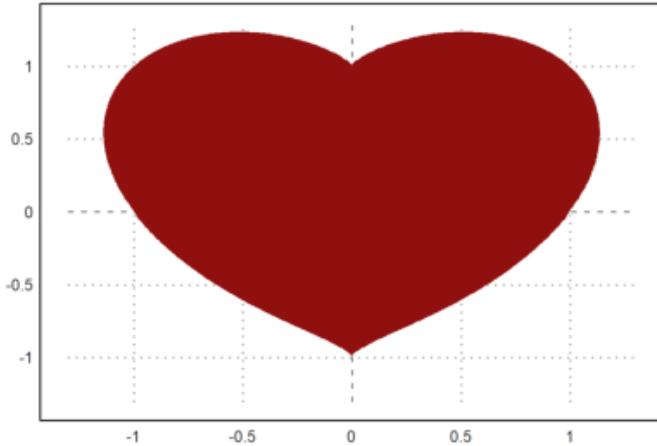


Ini juga memungkinkan untuk menandai sebuah wilayah

$$a \leq f(x, y) \leq b$$

ini dapat dilakukan dengan menambahkan sebuah level dengan 2 baris.

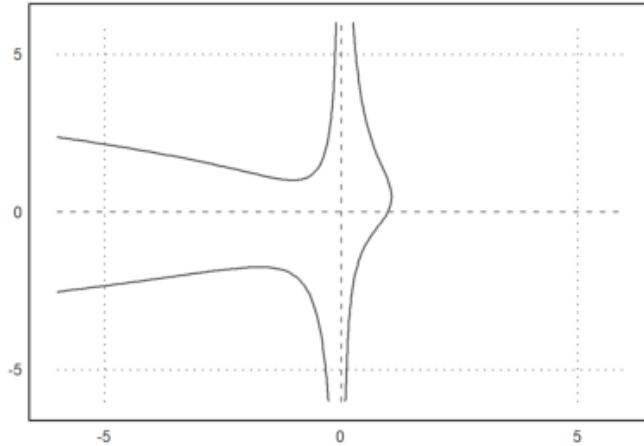
```
>plot2d("(x^2 + y^2 - 1)^3 - x^2*y^3", r=1.3, ...
>style="#", color=red, <outline, ...
>level=[-2; 0], n=100):
```



Ini juga memungkinkan untuk secara level spesifik. Sebagai contoh, kami akan membuat plot solusi dari sebuah persamaan seperti

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

```
>plot2d("x^3 - x*y + x^2 * y^2", r=6, level=1, n=100):
```



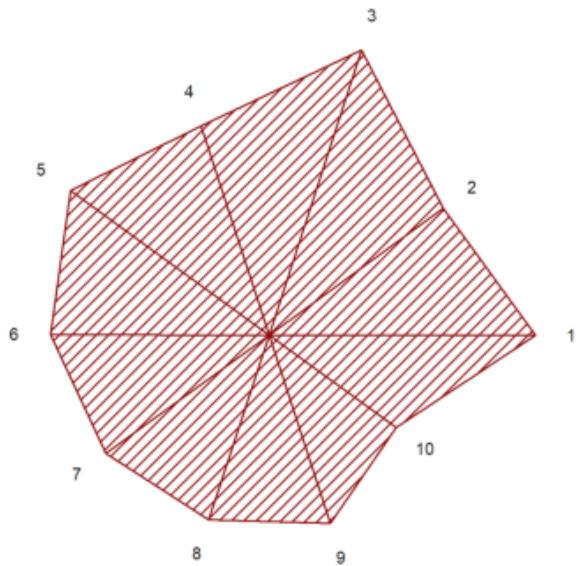
```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

```
    if !holding() then clg; endif;
    w=window(); window(0,0,1024,1024);
    h=holding(1);
    r=max(abs(v))*1.2;
    setplot(-r,r,-r,r);
    n=cols(v); t=linspace(0,2pi,n);
    v=v|v[1]; c=v*cos(t); s=v*sin(t);
    cl=barcolor(color); st=barstyle(style);
    loop 1 to n
        polygon([0,c[#,c[#+1]], [0,s[#,s[#+1]],1];
        if lab!=none then
            rlab=v[#]+r*0.1;
            {col,row}=toscreen(cos(t[#])*rlab,sin(t[#])*rlab);
            ctext(""+lab#[#],col,row-textheight()/2);
```

```
    endif;
end;
barcolor(cl); barstyle(st);
holding(h);
window(w);
endfunction
```

Tidak ada titik-titik atau batasan sumbu disini. Selebihnya, kami gunakan dengan jendela penuh untuk plot. Kami memanggil reset sebelumnya, kami test plot ini untuk mengatur ulang grafik menjadi seperti bawaan. Ini tidak dibutuhkan, jika kamu memastikan bahwa plot kamu berkerja.

```
>reset; starplot1(normal(1, 10) + 5, color=red, lab=1:10):
```



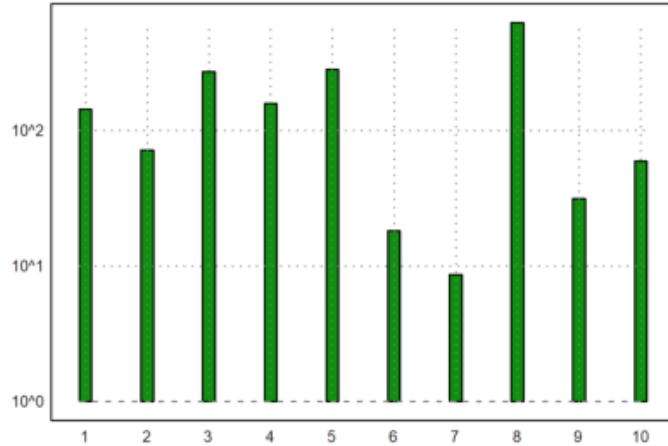
Terkadang, kamu ingin untuk membuat plot sesuatu dengan plot2d tidak dapat melakukannya, tetapi hampir. Contoh fungsi berikut, kami melakukan plot impuls logaritmik. plot2d juga dapat mem plot logaritmik, tetapi tidak untuk batang impuls.

```
>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 tes ini dengan nilai yang terdistribusi secara eksponensial.

```
>aspect(1.5); x = 1:10; y = -log(random(size(x))) * 200;...
>logimpulseplot(x, y);
```



Mari kita animasikan sebuah kurva 2D menggunakan plot langsung. Perintah `plot(x, y)` secara sederhana membuat sebuah plot menjadi jendela plot. `setplot(a, b, c, d)` mengatur jendela ini.

Fungsi `wait(0)` memaksa plot untuk muncul dalam gambar windows. Sebaliknya, menggambar ulang mengambil tempat interval nilai untuk membuatnya.

```
>function animliss(n, m) ...
```

```
t=linspace(0,2pi,500);
f=0;
c=framecolor(0);
l=linewidth(2);
setplot(-1,1,-1,1);
repeat
    clg;
```

```
plot(sin(n*t),cos(m*t+f));
wait(0);
if testkey() then break; endif;
f=f+0.02;
end;
framecolor(c);
linewidth(l);
endfunction
```

Tekan tombol apapun untuk menghentikan animasi ini.

```
>animliss(2, 3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

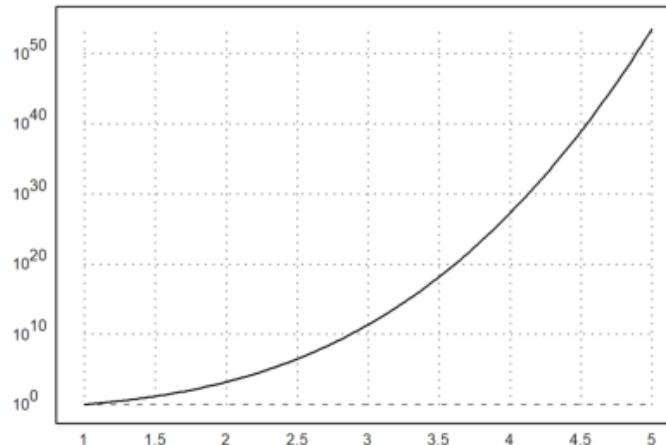
## Plot-Plot Logaritmik

EMT menggunakan parameter "logplot" untuk skala logaritmik.

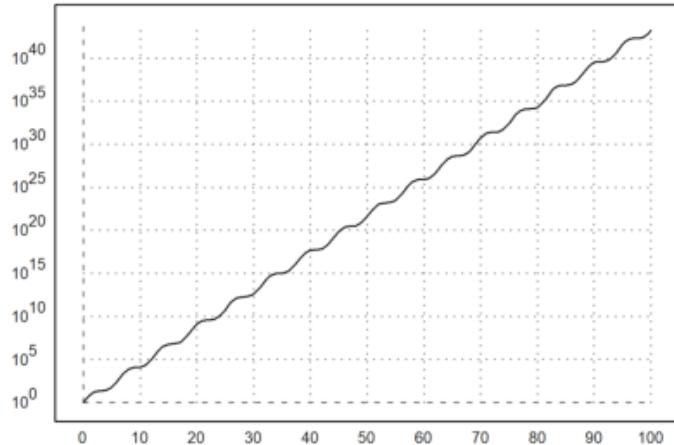
Plot logaritmik dapat diplotkan dengan menggunakan skala logaritmik dalam y dengan logplot=1 atau menggunakan skala logaritmik dalam x dan y dengan logplot=2 atau dalam x dengan logplot=3.

- logplot=1: logaritmik-y
- logplot=2: logaritmik-x-y
- logplot=3: logaritmik-y

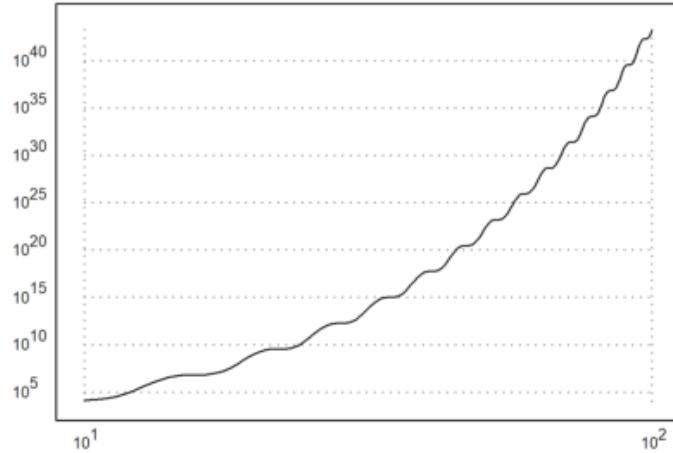
```
>plot2d("exp(x^3-x)*x^2", 1, 5, logplot=1):
```



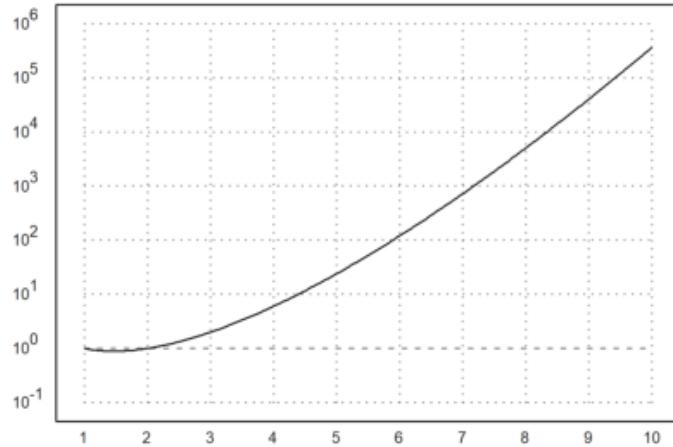
```
>plot2d("exp(x + sin(x))", 0, 100, logplot=1):
```



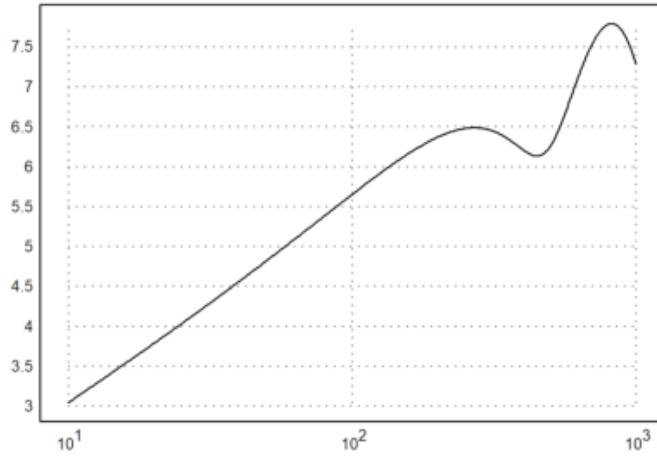
```
>plot2d("exp(x + sin(x))", 10, 100, logplot=2):
```



```
>plot2d("gamma(x)", 1, 10, logplot=1):
```

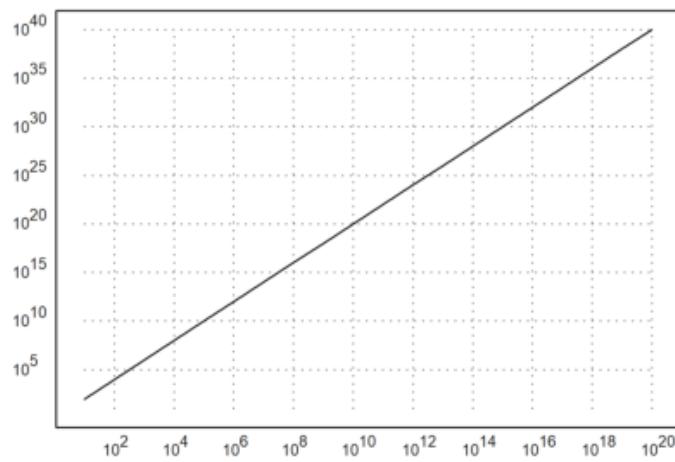


```
>plot2d("log(x * (2 + sin(x / 100)))", 10, 1000, logplot=3):
```



Ini juga dapat bekerja dengan plot-plot data.

```
>x = 10^(1:20); y = x^2 - x;  
>plot2d(x, y, logplot=2):
```



## Rujukan Lengkap Fungsi plot2d()

---

```
function plot2d (xv, yv, btest, a, b, c, d, xmin, xmax, r, n,..  
logplot, grid, frame, framecolor, square, color, thickness, style,..  
auto, add, user, delta, points, addpoints, pointstyle, bar, histogram,..  
distribution, even, steps, own, adaptive, hue, level, contour, ..  
nc, filled, fillcolor, outline, title, xl, yl, maps, contourcolor, ..  
contourwidth, ticks, margin, clipping, cx, cy, insimg, spectral, ..  
cgrid, vertical, smaller, dl, niveau, levels)
```

fungsi plot serbaguna untuk membuat plot dalam bidang (plot 2D). Fungsi ini dapat melakukan plotting fungsi satu variabel, plot data, kurva bidang, plot garis, batas-batas angka kompleks, dan plot implisit dari fungsi dua variabel.

Parameter

```
x, y      : persamaan, fungsi atau vektor data  
a, b, c, d : luasan plot (secara bawaan a = -2, b = 2)  
r          : jika r diatur, maka a=cx-r, b=cx+r, c=cy-r, d=cy+r  
            r dapat berupa sebuah vektor [rx, ry] atau sebuah vektor  
            [rx1, rx2, ry1, ry2]  
xmin,xmax : selang dari parameter untuk kurva  
auto       : Menentukan selang-y secara otomatis (bawaan)  
square     : jika true, akan mempertahankan kotak selang-x-y  
n          : jumlah rentang interval (adaptif secara default)  
grid       : 0 = tidak mempunyai batas dan label,  
            1 = hanya sumbu,  
            2 = batas normal(lihat bawah untuk jumlah baris batasan)  
            3 = didalam sumbu  
            4 = tidak ada batas  
            5 = batas penuh termasuk margin  
            6 = titik bantu pada frame  
            7 = hanya sumbu
```

8 = hanya sumbu, sub-titik bantu  
frame : 0 = tidak ada frame  
framecolor : warna dari frame dan batas  
margin : angka antara 0 dan 0.4 untuk margin sekeliling plot  
color : Warna kurva. Jika ini adalah sebuah vektor warna, ini akan digunakan pada setiap baris dari sebuah plot matriks.  
Dalam kasus ini titik-titik plot, seharusnya sebuah vektor kolom. Jika sebuah baris vektor atau sebuah matriks warna lengkap yang digunakan untuk titik plot, ini akan digunakan untuk setiap titik data.  
thickness : ketebalan garis dari kurva  
Nilai ini dapat lebih kecil dari 1 untuk garis yang sangat tipis.  
style : Gaya plot untuk garis, tanda dan isian.  
Untuk titik gunakan  
"[]", "<>", ".", "..", "...",  
"\*", "+", "|", "-", "o"  
"[]#", "<>#", "o#" (bentuk isian)  
"[]w", "<>w", "ow" (tidak-transparan)  
Untuk penggunaan garis  
"--", "--", "-.", ".", "-.-", "-.-", "->"  
Untuk penggunaan isian poligon atau plot garis  
"#", "#0", "0", "/", "\", "\\",  
"+", "|", "-", "t"  
points : membuat plot titik tunggal daripada baris segmen  
addpoints : Jika true, segmen garis plot dan titik  
add : menambahkan plot kedalam plot yang telah ada  
user : membuat pengguna untuk berinteraksi pada fungsi  
delta : ukuran langkah untuk interaksi pengguna  
bar : plot batang (x adalah batas interval, y nilai interval)  
histogram : plot frekuensi dari x dalam subinterval n  
distribution=n : plot distribusi dari x dalam subinterval n  
even : gunakan dalam nilai untuk histogram otomatis  
steps : plot fungsi sebagai sebuah fungsi langkah (steps=1,2)

```
adaptive : penggunaan plot adaptif (n adalah jumlah minimal langkah)
level     : plot garis level dari sebuah fungsi implisit dua variabel
outline   : menggambar batasan dari selang level
```

Jika nilai level adalah sebuah matriks  $2 \times n$ , selang dari level akan

digambarkan dalam warna menggunakan gaya isian. Jika luaran bernilai true, ini akan digambarkan sebagai warna kontur. Gunakan fitur ini, wilayah dari  $f(x, y)$  diantara limit dapat dicatat.

```
hue      : tambahkan warna hue untuk plot level untuk
            mengindikasikan nilai fungsi
contour  : Gunakan plot level dengan level otomatis
nc       : Banyaknya baris level otomatis
title    : Judul plot (bawaan "")
xl, yl   : label untuk sumbu-x dan sumbu-y
smaller  : jika  $> 0$ , akan memberi ruang lebih ke kiri untuk label
vertical :
            Mengubah arah label secara vertikal. Ini mengubah variabel global
            verticallabel secara local pada satu plot. Nilai 1 mengatur hanya
            teks vertikal, nilai 2 digunakan untuk label numerik vertical pada
            sumbu y.
filled   : mengisi plot dari sebuah kurva
fillcolor: mengisi warna untuk batang dan kurva yang terisi
outline  : batasan untuk poligon yang terisi
logplot  : mengatur plot logaritmik
            1 = plot logartimik dalam y,
            2 = plot logaritmik dalam xy,
            3 = plot logaritmik dalam x
own      :
            Sebuah string, yang mana titik-titik ke sebuah rutin plot sendiri.
            Dengan >user, kamu akan mendapatkan interaksi pengguna seperti
```

```
dalam plot2d. Rentang akan diatur sebelum tiap-tiap panggilan pada fungsi.  
maps      : ekspresi peta (0 tercepat), fungsi selalu terpetakan  
contourcolor : warna dari garis kontur  
contourwidth : tebal dari garis kontur  
clipping   : menggaliikan cetakan (bawaan adalah true)  
title      :  
            Ini dapat digunakan untuk mendeskripsikan plot. Judul akan muncul diatas plot. Terlebih, sebuah label untuk sumbu x dan y akan ditambahkan dengan xl="string" atau yl="string". Label lainnya akan ditambahkan dengan fungsi label() atau labelbox(). Judul dapat berupa sebuah string unicode atau sebuah gambar dari sebuah formula Latex.  
cgrid      :  
            Menentukan banyaknya baris batasan untuk plot dari batasan kompleks. Seharusnya sebuah pembagi dari ukuran matriks dikurangi 1 (banyak nya sub-interval). cgrid dapat berupa sebuah vektor [cx, cy].
```

## Pratinjau

Fungsi ini dapat membuat plot

- Ekspresi, pemanggilan collection atau fungsi dari satu variabel,
- Kurva parametrik,
- data x beserta data y,
- fungsi implisit,
- plot batang,
- batasan kompleks,
- poligon.

Jika sebuah fungsi atau ekspresi untuk xv diberikan, plot2d akan menghitung nilai yang rentang berikan menggunakan fungsi atau ekspresi. Ekspresi haruslah sebuah ekspresi dalam variabel x. Selang haruslah didefinisikan dalam parameter a dan b kecuali memiliki rentang bawaan [-2, 2] akan digunakan. Selang-y akan secara otomatis dihitung, kecuali c dan d dispesifikasikan, atau sebuah jari-jari r, yang mana menghasilkan selang [-r, r] untuk x dan y. Untuk plot dari fungsi, plot2d akan menggunakan evaluasi adaptif dari fungsi secara bawaan. Untuk mempercepat membuat plot pada fungsi yang rumit, matikan <adaptive ini, dan secara pilihan mengurangi banyaknya interval n. Sebaliknya, plot2d() akan secara bawaan menggunakan pemetaan. Dengan kata lain, ini akan menghitung elemen plot untuk elemen. Jika ekspresi kamu atau fungsi kamu dapat dipegang oleh vektor x, kamu dapat mengubah <maps agar tidak aktif untuk evaluasi yang lebih cepat.

Catatan bahwa plot adaptif selalu menghitung elemen untuk elemen. Jika fungsi atau ekspresi untuk keduanya xv dan untuk yv dispesifikasi, plot2d() akan dihitung sebuah kurva dengan nilai xv sebagai koordinat-x dan nilai yv sebagai koordinat-y. Dalam kasus ini, selang harusnya didefinisikan untuk parameter menggunakan xmin, xmax. Ekspresi yang dimuat dalam string harus selalu berupa ekspresi dengan parameter variabel x.

## Beberapa Materi untuk Fungsi dan Plot

---

Dalam matematika plot tidak dapat dipisahkan dengan sebuah fungsi. Karena plot itu sendiri merupakan gambaran/sketsa yang dihasilkan oleh fungsi tersebut. Dalam kasus ini, EMT menyediakan fungsi pemrograman yaitu plot2d yang mana dapat membuat plot untuk beberapa kondisi dalam suatu bidang dua dimensi. Ini berarti fungsi matematika yang akan kita pakai adalah fungsi untuk satu variabel, sederhananya  $f(x) = y$ .

Dalam bahasan selanjutnya, saya akan membahas beberapa fungsi yang ada di matematika, seperti:

- fungsi linear
- fungsi kuadrat
- fungsi polinomial
- fungsi eksponensial
- fungsi logaritmik
- fungsi trigonometri

Kemudian akan membahas fungsi khusus seperti fungsi tangga, fungsi atap, dan fungsi absolut.

Selanjutnya, akan dibahas beberapa plot dalam statistika, seperti plot batang, plot pencar, dan plot distribusi normal

Terakhir disuguhkan beberapa hal yang bisa dilakukan menggunakan EMT ini dengan sedikit bermain.

---

### Fungsi Linear

Dalam matematika, sebuah fungsi dapat dikatakan linear apabila membentuk sebuah garis lurus. Dimana fungsi tersebut merupakan fungsi polinomial yang variabel-variabelnya memiliki pangkat tertinggi adalah 1.

Bentuk umum fungsi linear adalah

$$f(x) = y = mx + c$$

dengan:

- y adalah variabel tak bebas
- m adalah gradien
- x adalah variabel bebas
- c sebarang bilangan

Contoh:

1. Buatlah plot dengan garis lurus yang melalui titik (4, 2) dan (2, 6)

Penyelesaian:

Dapat menggunakan persamaan umum dari pencarian gradien, yaitu

$$\frac{(y - y_1)}{(y_2 - y_1)} = \frac{(x - x_1)}{(x_2 - x_1)}$$

dengan mengubah menjadi persamaan  $y = \dots$

```
>persamaanY &= solve((y - y1) / (y2 - y1) = (x - x1) / (x2 - x1), y)
```

$$[y = \frac{-(x_1 - x)y_2 - (x - x_2)y_1}{x_2 - x_1}]$$

Kemudian masukkan titik titiknya, misalkan

$$y_1 = 2, y_2 = 6, x_1 = 4, x_2 = 2$$

```
>&persamaanY with [y1 = 2, y2 = 6, x1 = 4, x2 = 2]
```

$$[y = \frac{2(x - 2) + 6(4 - x)}{2}]$$

Kemudian ubah menjadi ekspresi program

```
>ekspresiPersamaanY = "(2 * (x - 2) + 6 * (4 - x)) / 2"
```

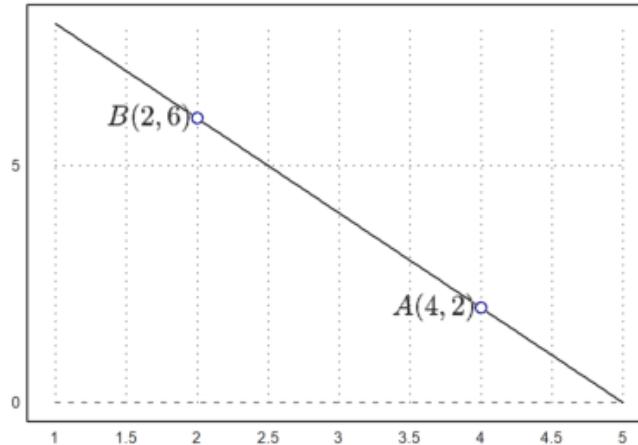
$$(2 * (x - 2) + 6 * (4 - x)) / 2$$

Terakhir adalah buat plotnya dengan memastikan titik-titik yang dilaluinya.

```

>y1 = 2;...
>y2 = 6;
>x1 = solve(ekspresiPersamaanY, 0, 1, y1); ...
>x2 = solve(ekspresiPersamaanY, 0, 1, y2); ...
>plot2d(ekspresiPersamaanY, 1, 5); ...
>plot2d(x1, y1, >points, >add, color=blue, style="ow");...
>label(latex("A(4, 2)'), x1, y1, pos = "cl", offset=10); ...
>plot2d(x2, y2, >points, >add, color=blue, style="ow");...
>label(latex("B(2, 6)'), x2, y2, pos = "cl", offset=10):

```



Jadi, persamaan garis lurus yang melalui titik  $(4, 2)$  dan  $(2, 6)$  adalah

$$y = \frac{2(x - 2) + 6(4 - x)}{2}$$

dengan plot seperti diatas

2. Buatlah plot persamaan grafik fungsi linear melalui titik (2, 4) dengan gradien 2.

Pembahasan:

Menggunakan rumus

$$y - y_1 = m(x - x_1)$$

dengan:

$$y_1 = 4, m = 2, x_1 = 2$$

```
>persamaan &= "y - y1 = m*(x - x1)"
```

$$y - y_1 = m(x - x_1)$$

```
>persamaanY &= solve(persamaan, y)
```

$$[y = y_1 - m x_1 + m x]$$

```
>&persamaanY with [m = 2, y1 = 4, x1 = 2]
```

[y = 2 x]

```
>ekspresiPersamaanY = "y = 2*x";
>y1 = 4;
>x1 = solve(ekspresiPersamaanY, 0, 1, y1);...
```

```
Error in Evaluate, superfluous characters found.
Error in expression: y = 2*x
secant:
    y0=f$(x0,args())-y; y1=f$(x1,args())-y;
Try "trace errors" to inspect local variables after errors.
solve:
    if eps then return secant(f$,a,b,y;args(),eps=eps);
```

```
>plot2d(ekspresiPersamaanY, 0, 6); ...
>plot2d(x1, y1, >points, >add, color=blue, style="ow");...
>label(latex("A(2, 4)'), x1, y1, pos="cl", offset=10):
```

Jadi persamaan garis lurus bergradien 2 yang melalui (2, 4) adalah

$$y = 2x$$

dengan plot seperti diatas.

## Fungsi Kuadrat

---

Biasanya fungsi kuadrat ini berupa parabola apabila di gambarkan. Definisi umum dari fungsi ini merupakan fungsi polinomial dengan pangkat tertinggi variabelnya adalah 2. Bentuk umumnya dinotasikan sebagai berikut

$$f(x) = ax^2 + bx + c, a \neq 0$$

Contoh soal

Buatlah grafik fungsi

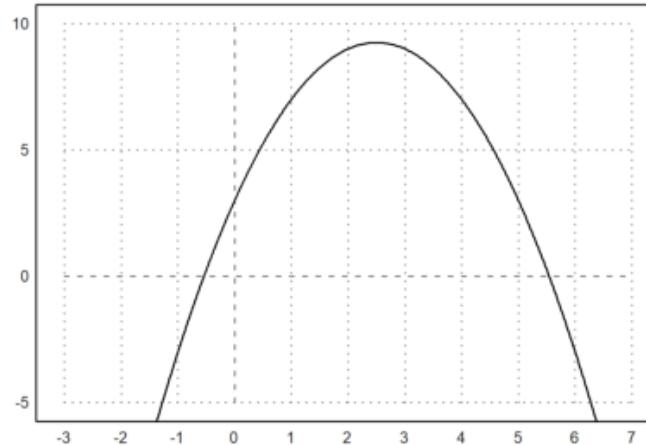
$$f(x) = -x^2 + 5x + 3$$

dalam sistem koordinat. Temukan titik puncak, titik potong terhadap sumbu-x dan sumbu-y.

Penyelesaian:

- Grafik fungsi

```
>function f(x) &= -(x)^2 + 5*x + 3;  
>plot2d(f, -3, 7, -5, 10):
```



- Titik puncak

Dapat dihitung dengan

$$\left( \frac{-b}{2a}, f\left(\frac{-b}{2a}\right) \right)$$

dengan:

$$a = -1, b = 5, c = 3$$

```
>titikPuncak &= "-b/2*a"
```

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

```
>titikPuncakX := &titikPuncak with [b = 5, a = -1]
```

$$\begin{array}{r} 5 \\ - \\ 2 \end{array}$$

```
>titikPuncakY = f(5/2)
```

Jadi titik puncak dari fungsi kuadrat tersebut adalah

$$(2.5, 9.25)$$

- Titik Potong sumbu-x

```
>ekspresiPersamaan &= "y = -x^2 + 5*x + 3";
>titikPotongY := &solve(ekspresiPersamaan with y = 0, x)
```

$$[x = \frac{5 - \sqrt{37}}{2}, x = \frac{\sqrt{37} + 5}{2}]$$

Jadi diperoleh titik potong persamaan kuadrat tersebut terhadap sumbu-x adalah

$$\left(\frac{5 - \sqrt{37}}{2}, 0\right), \left(\frac{\sqrt{37} + 5}{2}, 0\right)$$

- Titik potong sumbu-y

```
> $&solve(ekspresiPersamaan with x = 0, y)
```

$$[y = 3]$$

Jadi diperoleh titik potong persamaan kuadrat tersebut terhadap sumbu-y adalah

$$(0, 3)$$

## Fungsi Polinomial

---

fungsi polinomial adalah sebuah fungsi yang dapat ditulis dengan bentuk umum:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

untuk n adalah bilangan bulat tidak negatif, dengan koefisinya

$$a_0, a_1, \dots$$

adalah bilangan rill, dan

$$a_n \neq 0$$

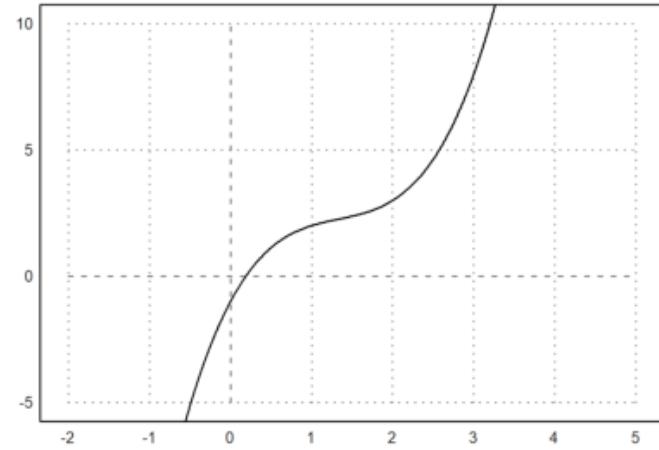
Soal:

Buatlah plot dari fungsi

$$f(x) = x^3 - 4x^2 + 6x - 1$$

penyelesaian:

```
>function f(x) := x^3 - 4*x^2 + 6*x - 1;...
>plot2d(&f, -2, 5, -5, 10):
```



## Chapter 3

### Plot 3D dengan EMT

## Menggambar Plot 3D dengan EMT

---

Ini adalah pengenalan untuk plot 3D pada Euler. Kita membutuhkan plot 3D untuk memvisualisasikan sebuah fungsi dari dua variabel.

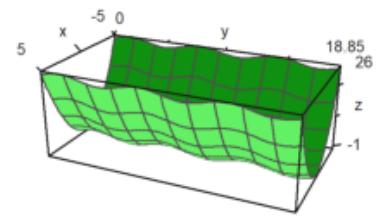
Euler menggunakan fungsi ini untuk algoritma pengurutan untuk menyembunyikan bagian dalam latar belakang. Umumnya, Euler menggunakan proyeksi sentral. Secara bawaan dari kuadran x-y positif melewati titik asal  $x=y=z=0$ , tetapi sudut= $0^\circ$  terlihat arah dari sumbu-y. Pandangan sudut dan tinggi dapat berubah.

Euler dapat memplotkan

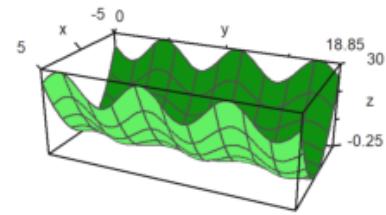
- permukaan dengan bayangan dan garis level atau rentang level
- titik titik yang tersebar
- kurva parametrik
- permukaan implisit

Sebuah plot 3D dari sebuah fungsi menggunakan plot3d. Cara paling mudah untuk membuatnya adalah dengan ekspresi dalam x dan y. Parameter r dapat diatur dalam rentang dari sekeliling plot (0, 0).

```
>aspect(1.5); plot3d("x^2+sin(y)",-5,5,0,6*pi):
```



```
>plot3d("x^2+x*sin(y)", -5,5,0,6*pi):
```



Silakan lakukan modifikasi agar gambar "talang bergelombang" tersebut tidak lurus melainkan lengkung/melingkar, baik melingkar secara mendatar maupun melingkar turun/naik (seperti papan peluncur pada kolam renang. Temukan rumusnya.

### Fungsi fungsi dari Dua Variabel

---

Untuk membuat grafik dari sebuah fungsi, gunakan

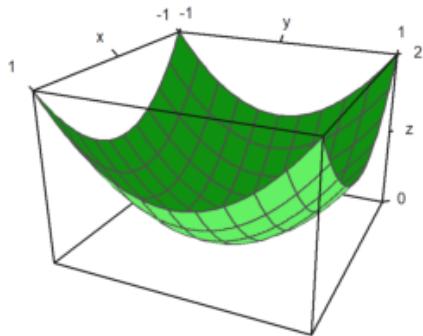
- sebuah ekspresi sederhana dalam x dan y,
- nama fungsi dari dua variabel
- atau matriks data

Secara bawaan adalah kabel yang terisi dengan warna-warna yang berbeda dalam kedua sisi. Catatan bahwa nomor bawaan dari interval grid adalah 10, tetapi plot menggunakan angka bawaan dari persegi 40x40 untuk membuat permukaan. Ini dapat diubah.

- n=40, n=[40,40]: banyaknya garis grid dalam setiap arah
- grid=10, grid=[10,10]: banyaknya garis grid dalam setiap arah

Kami menggunakan bawaan n = 40 dan grid = 40.

```
>plot3d("x^2+y^2"):
```

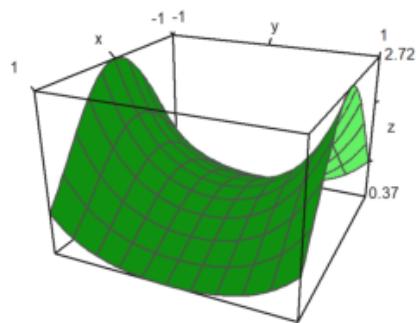


Interaksi pengguna dapat memungkinkan dengan mengatur parameter >user. Pengguna dapat menekan tombol berikut.

- left,right,up,down: mengubah pandangan sudut
- +,-: memperbesar atau mengecilkan
- a: mengeluarkan sebuah anaglyph (lihat bawah)
- l: mengalihkan asal cahaya (lihat bawah)
- space: atur ulang ke bawaan
- return: selesaikan interaksi

```
>plot3d("exp(-x^2+y^2)",>user, ...
> title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



Plot selang untuk fungsi dapat dispesifikan dengan

- a,b: selang-x
- c,d: selang-y
- r: sebuah persegi sekitar yang simetri (0,0)
- n: banyaknya subinterval untuk plot

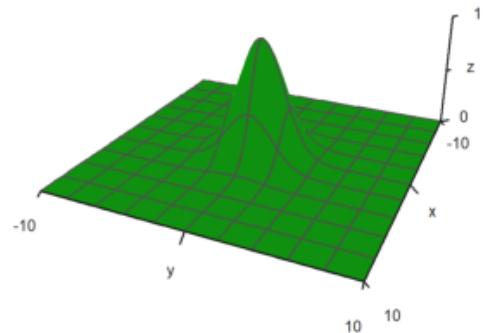
Ini beberapa parameter untuk menskalakan fungsi atau mengubah tampak dari grafik.

fscale: menskalakan fungsi nilai (bawaan <fscale>)

scale: angka atau vektor 1x2 untuk menskalakan ke arah x atau y

frame: tipe frame (bawaan 1)

```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):
```



Pandangan dapat diubah dengan banyak cara.

- distance: arah pandangan ke plot.
- zoom: nilai pembesaran.
- angle: arah ke sumbu-y negatif dalam radian.
- height: the height of the view in radians.

Nilai bawaan dapat diinspeksi atau diubah dengan fungsi `view()`. Ini akan mengembalikan parameter dalam urutan diatas.

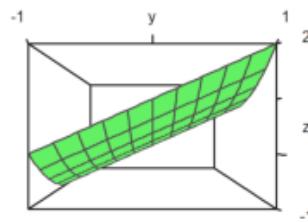
```
>view
```

```
[5, 2.6, 2, 0.4]
```

Jarak terdekat membutuhkan pembesaran yang lebih sedikit. Efeknya kira kira seperti sebuah lensa sudut yang lebar.

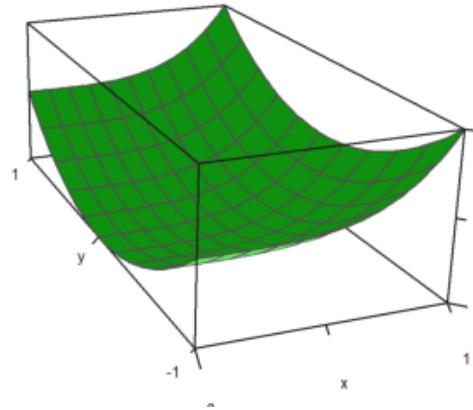
Seperti contoh berikut, angle=0 dan height=0 terlihat dari sumbu-y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```



Plot selalu terlihat ke titik tengah dari kubus plot. Kamu dapat menggerakkan tengah dengan parameter center.

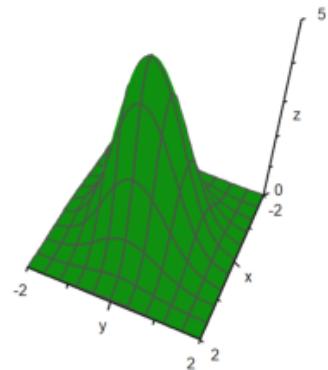
```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
> center=[0.4,0,0],zoom=5):
```



Plot dapat diskalakan untuk menyesuaikan ke bagian kubus untuk dilihat. Jadi ini tidak membutuhkan perubahan jarak atau memperbesar tergantung ukuran dari plot. Label ini merujuk pada ukuran sebenarnya.

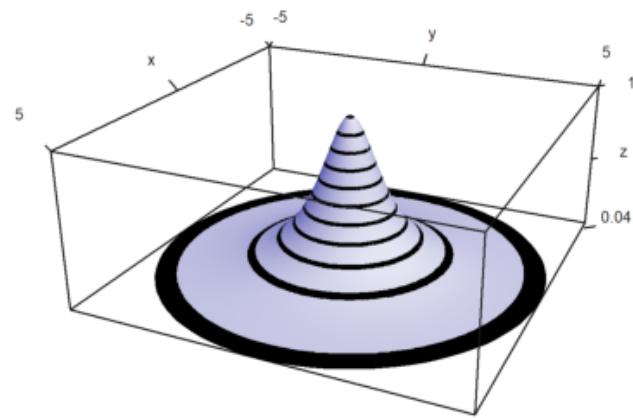
Jika kamu mematikan dengan `scale=false`, kamu harus berhati-hati, yang mana plot masih menyesuaikan ke jendela plot, dengan mengubah pandangan arah atau perbesaran dan menggerakkan titik tengah.

```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
> center=[0,0,-2],frame=3):
```

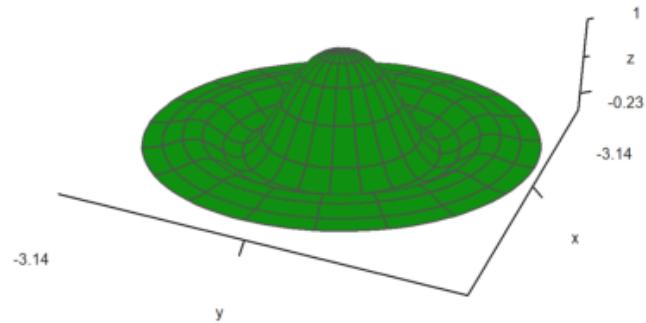


Plot kutub juga tersedia. Parameter `polar=true` menggambar sebuah plot polar. Fungsi harus tetap sebagai fungsi dari  $x$  dan  $y$ . Parameter `"fscale"` menskalakan fungsi dengan skala tersendiri. Sebaliknya fungsi diskalakan mengikuti sebuah kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=blue):
```



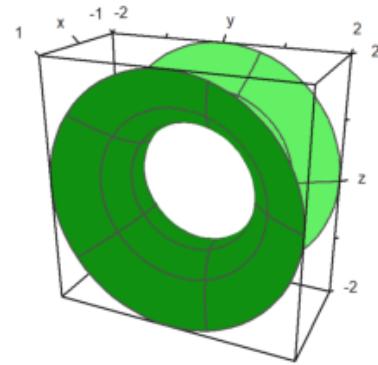
```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```



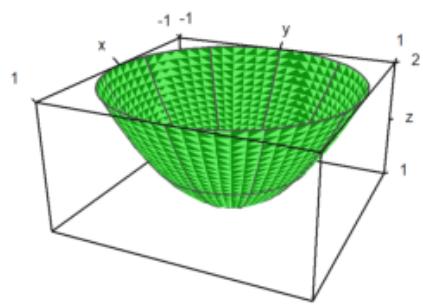
Parameter rotate memutar sebuah fungsi dalam x mengelilingi sumbu-x.

- rotate=1: Menggunakan sumbu-x
- rotate=2: Menggunakan sumbu-z

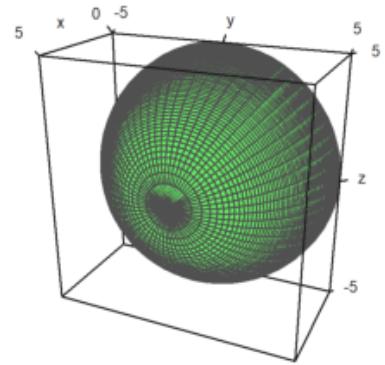
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



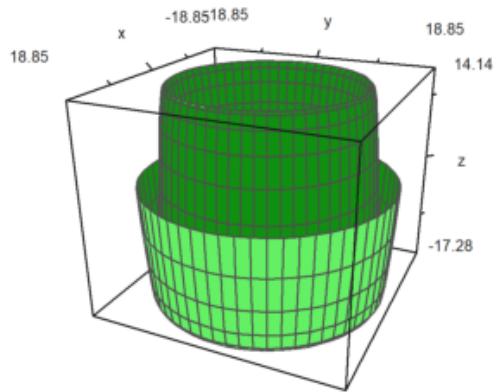
```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```



```
>plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1):
```

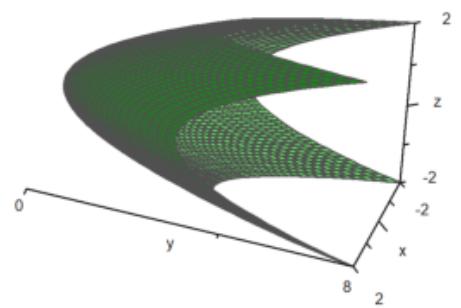


```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```



Ini adalah plot dengan tiga fungsi.

```
>plot3d("x", "x^2+y^2", "y", r=2, zoom=3.5, frame=3):
```



## Plot Kontur

---

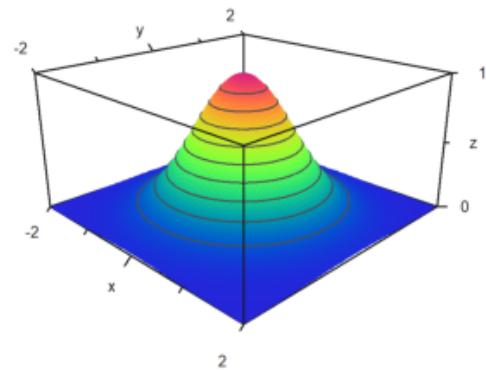
Untuk plot, Euler menambahkan garis bata. Ini memungkinkan untuk menggunakan level garis dan sebuah-warna hue atau sebuah spektral berwarna hue. Euler dapat menggambar tinggi dari fungsi dalam sebuah plot dengan bayangan. Dalam semua plot 3D Euler dapat menghasilkan anaglyph merah/cyan.

- >hue: Menggubah bayangan cahaya daripada garis.
- >contour: Membuat plot garis kontur otomatis dalam sebuah plot.
- level=... (or levels): Sebuah vektor dari nilai untuk garis kontur.

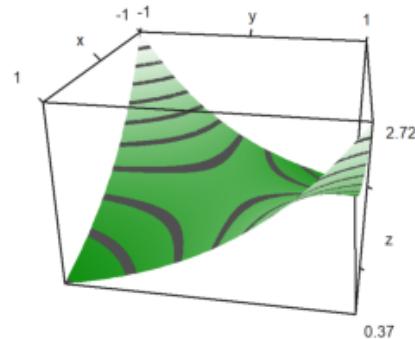
Bawaannya level="auto", yang mana menghitung beberapa garis level secara otomatis. Seperti yang kamu lihat dalam plot, level dalam rentang level.

Gaya bawan dapat diubah. Untuk plot kontur, kami menggunakan sebuah batasan yang halus untuk titik titik 100x100, menskalakan fungsi dan plot, dan menggunakan sudut yang berbeda dari pandangan.

```
>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):
```

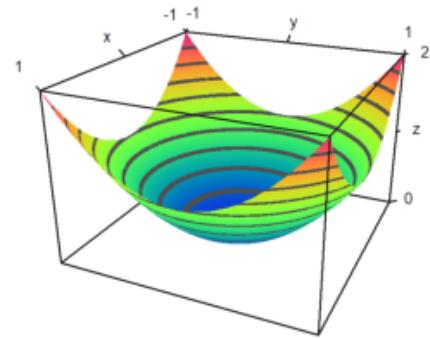


Bayangan bawaan menggunakan warna abu-abu. Tetapi selang spektral dari warna juga tersedia.

- >spectral: Digunakan skema spektral bawaan
- color=...: Menggunakan warna spesial atau skema spektral

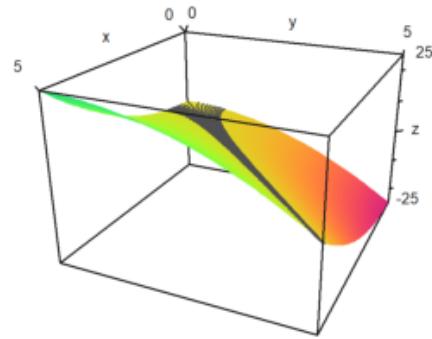
Untuk plot berikut, kami menggunakan skema spektral bawaan dan meningkatkan angka dari titik untuk mendapatkan tampilan yang halus.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```



Selain garis level otomatis, kami dapat juga mengatur nilai dari garis level. Ini akan menghasilkan garis level yang tipis daripada rentang level.

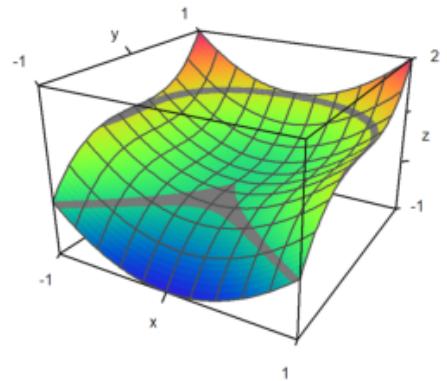
```
>plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=redgreen):
```



Plot berikut, kami menggunakan dua level yang sangat luas dari -0.1 ke 1 dan dari 0.9 ke 1. Ini dimasukkan sebagai sebuah matrik dengan level batas seperti kolom.

Selebihnya, kami melapisi sebuah batasan dengan 10 interval pada setiap arah.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...
> >spectral,angle=30°,grid=10,contourcolor=gray):
```

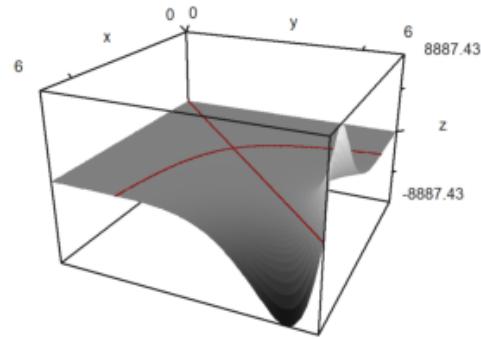


Contoh berikut, kami mengatur plot, dimana

$$f(x, y) = x^y - y^x = 0$$

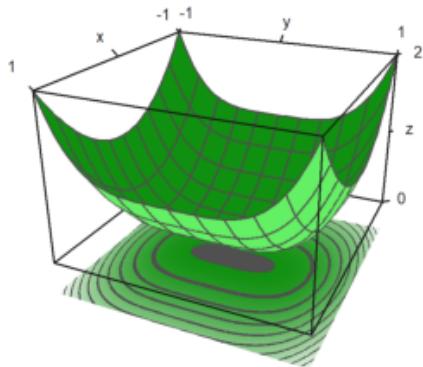
Kami menggunakan garis tipis tunggal untuk garis level.

```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```



Ini memungkinkan untuk menampilkan sebuah bidang kontur dibawah plot. Warna dan jarak ke plot dapat di spesifikan.

```
>plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2):
```



Ini beberapa gaya yang lainnya. Kami selalu mematikan frame dan menggunakan skema warna yang bervariasi untuk plot dan garis batas.

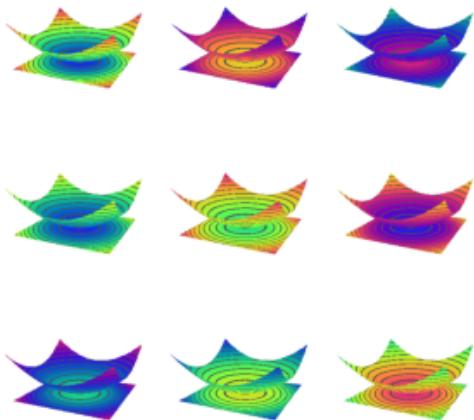
```
>figure(2,2); ...
>expr="y^3-x^2"; ...
>figure(1); ...
> plot3d(expr,<frame,>cp,cpcolor=spectral); ...
>figure(2); ...
> plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
>figure(3); ...
> plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
>figure(4); ...
> plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...
>figure(0):
```



Ini beberapa skema spektral yang lainnya, dihitung dari 1 ke 9. Tetapi kamu juga menggunakan color=value, dimana value

- spectral: untuk sebuah selang dari biru ke merah
- white: untuk selang pewarna
- yellowblue,purplegreen,blueyellow,greenred
- blueyellow, greenpurple,yellowblue,redgreen

```
>figure(3,3); ...
>for i=1:9; ...
> figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4); ...
>end; ...
>figure(0):
```



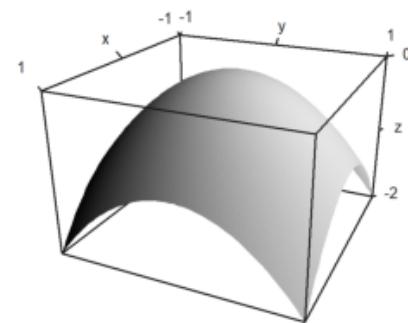
Sumber cahaya dapat diubah dengan 1 dan kunci kursor selama interaksi pengguna. Ini juga dapat diubah dengan parameter.

- light: arah cahaya
- amb: cahaya sekitar antara 0 dan 1

Catatan bahwa program tidak membuat perbedaan antara sisi dari plot. Tidak ada bayangan. Untuk hal ini akan membutuhkan Povray.

```
>plot3d("-x^2-y^2", ...
>  hue=true,light=[0,1,1],amb=0,user=true, ...
>  title="Press 1 and cursor keys (return to exit)":
```

Press I and cursor keys (return to exit)



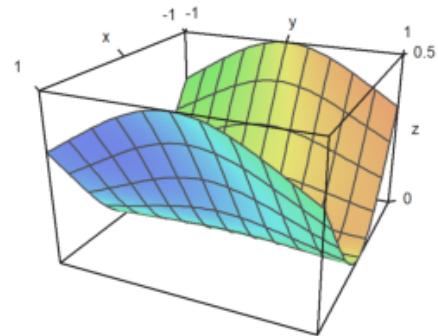
Parameter color mengubah warba dari permukaan. Warna dari garis level juga dapat diubah.

```
>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):
```



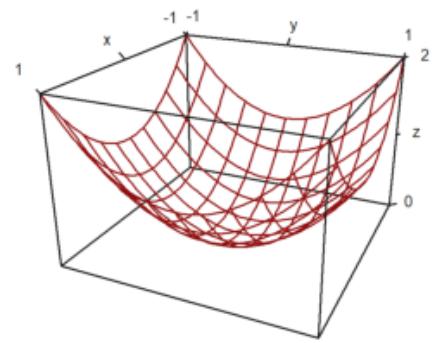
Color 0 memberikan spesial pelangi efek.

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



Permukaan juga dapat transparan.

```
>plot3d("x^2+y^2",>transparent,grid=10,wirecolor=red):
```



## Plot Implisit

---

Ini juga plot implisit dalam dimensi tiga. Euler menghasilkan potongan melalui objek. Fitur dari plot3d mencakup plot implisit. Plot ini menunjukkan pengaturan 0 dari fungsi dalam variabel 3.

Solusi dari

$$f(x, y, z) = 0$$

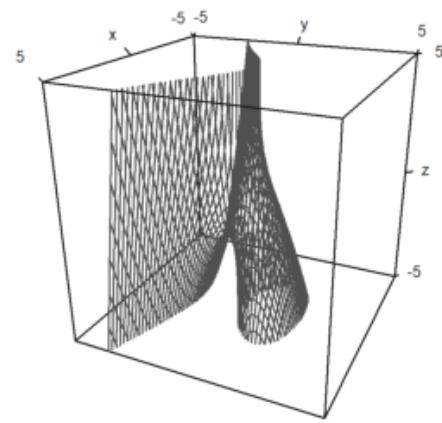
dapat divisualisasikan dalam potongan paralel ke bidang x-y-, x-z- dan y-z.

- implicit=1: memotong secara paralel ke bidang y-z
- implicit=2: memotong secara paralel ke bidang x-z
- implicit=4: memotong secara paralel ke bidang x-y

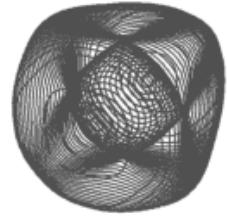
Menambahkan nilai-nilai ini, jika kamu suka. Dalam contoh ini kami membuat plot

$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

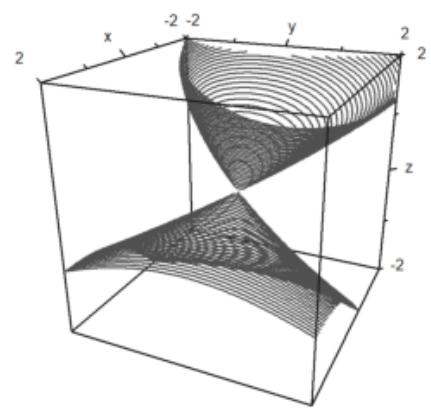
```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```



```
>c=1; d=1;  
>plot3d("((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)-d",r=2,
```



```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):
```



## Membuat plot data 3D

---

Seperti plot2d, plot3d menerima data. Untuk objek 3D, kamu harus menyediakan sebuah matriks dari nilai x, y dan z, atau fungsi tiga atau ekspresi  $fx(x, y)$ ,  $fy(x, y)$ ,  $fz(x, y)$ .

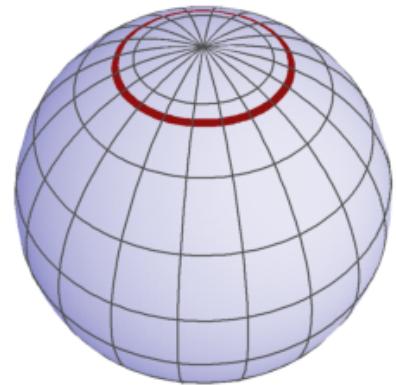
$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

Karena x, y, z merupakan matriks, kami mengasumsikan bahwa (t, s) melewati sebuah batasan persegi. Sebagai hasilnya, kamu dapat membuat gambar persegi dalam ruang.

Kamu dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

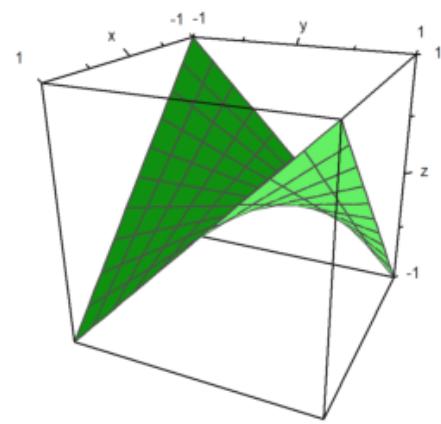
Seperti contoh berikut, kami menggunakan vektor dari nilai t dan sebuah kolom vektor dari nilai s untuk memparameterisasi wilayah dari bola. dalam penggambaran kami menandai wilayah-wilayah, dalam kasus kami wilayah polar.

```
>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°):
```



Ini sebuah contoh, yang mana grafik dari sebuah fungsi.

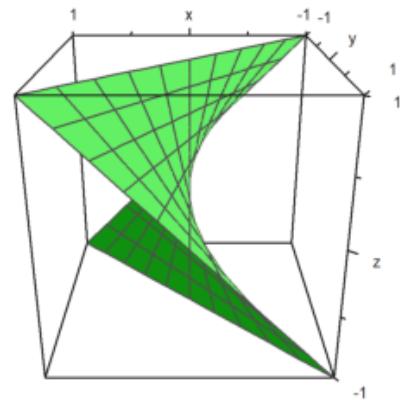
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



Namun, kamu juga membuat semua pengurutan dari wilayah. Ini wilayah yang sama seperti fungsi

$$x = y z$$

```
>plot3d(t*s,t,s,angle=180°,grid=10):
```



Dengan usaha yang lebih, kami menghasilkan lebih banyak permukaan.

Seperti contoh berikut kami membuat pendangan bayangan untuk bola terdistorsi. Koordinat biasa untuk bola adalah

$$\gamma(t, s) = (\cos(t) \cos(s), \sin(t) \sin(s), \cos(s))$$

dengan

$$0 \leq t \leq 2\pi, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2}.$$

Kami mendistorsi ini dengan sebuah faktor

$$d(t, s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

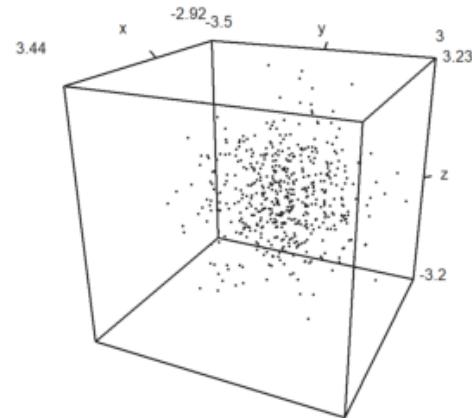
```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
> light=[1,0,1],frame=0,zoom=5):
```



Tentu saja, titik awan juga mungkin. Untuk membuat data titik dalam ruang, kami membutuhkan tiga vektor untuk koordinat dari titik-titik.

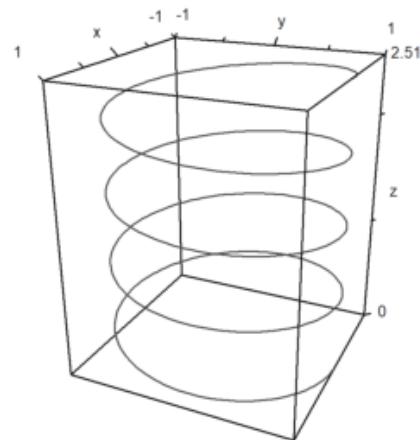
Gaya seperti dalam plot3d dengan points=true;

```
>n=500; ...
> plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

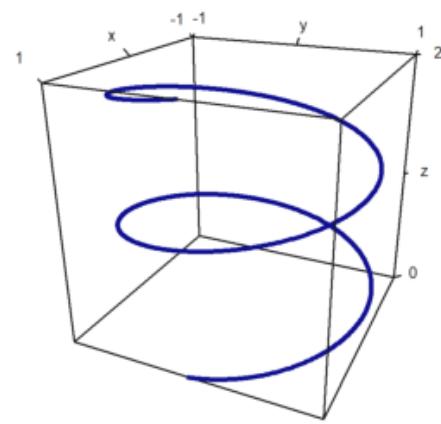


Ini juga mungkin untuk membuat plot sebuah kurva 3D. Dalam kasus ini, mudah untuk menghitung titik-titik dari kurva sebelumnya. Untuk kurva dalam bidang kamu menggunakan urutan dari koordinat dan parameter wire=true.

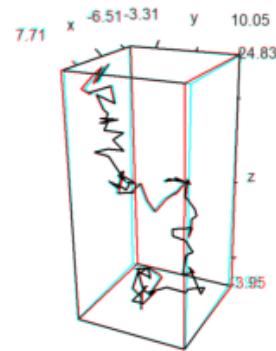
```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
>lineWidth=3,wirecolor=blue):
```

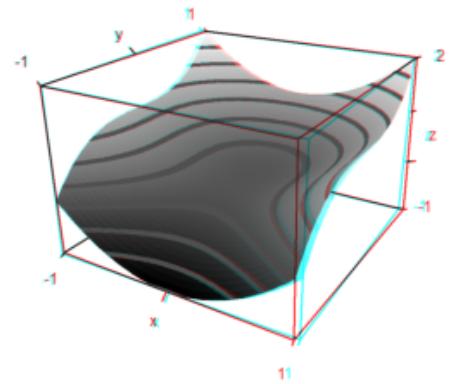


```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



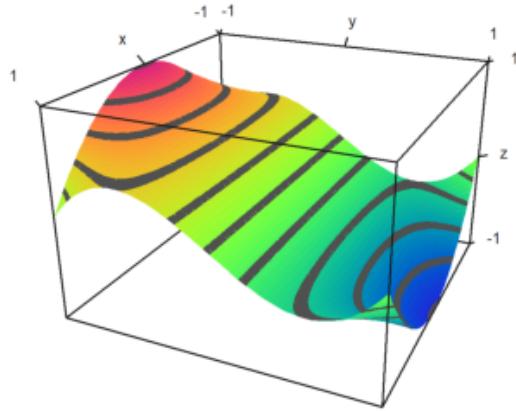
EMT dapat juga membuat plot dalam mode anaglyph. Untuk memandang seperti sebuah plot, kamu membutuhkan kacamata merah/cyan.

```
>plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```



Sering kali, sebuah skema spektral warna menggunakan plot. Menegaskan tinggi dari fungsi.

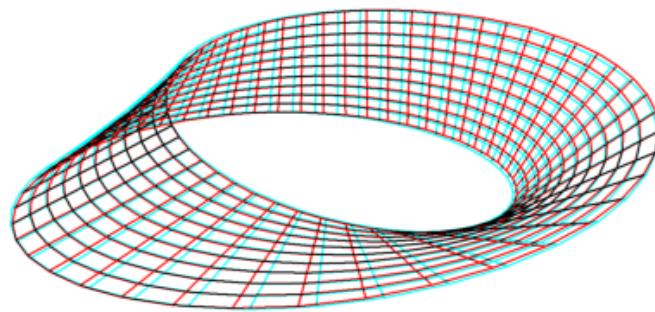
```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```



Euler dapat membuat permukaan yang terparameter juga, ketika parameter nilai x, y, dan z dari sebuah gambar dari sebuah batasan persegi dalam ruang.

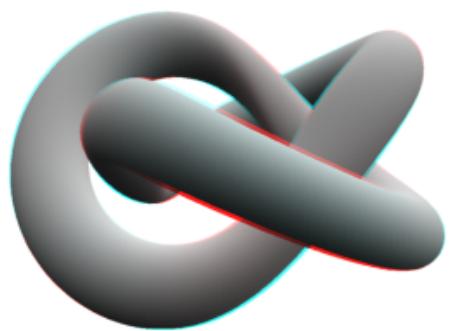
Seperti demo berikut, kami mengatur parameter u dan v, dan membuat koordinat ruang untuk ini.

```
>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):
```



Ini merupakan contoh yang lebih kompleks, yang mana terlihat megah dengan kacamata merah/cyan.

```
>u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)'; ...
>x:=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...
>y:=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...
>z=sin(u)+2*cos(3*v); ...
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```



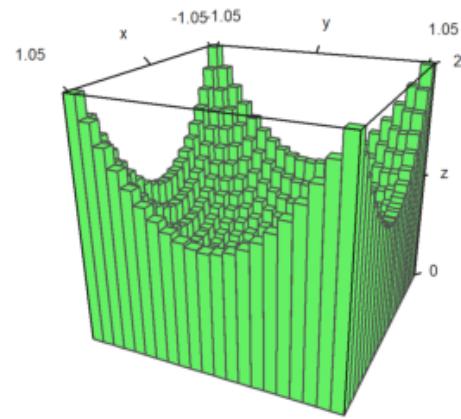
Plot batang juga mungkin untuk dibuat. Seperti ini, kami menyediakan

- x: vektor baris dengan elemen n+1
- y: vektor kolom dengan elemen n+1
- z: matriks nxn nilai.

z dapat diperbesar, tetapi hanya nilai nxn yang akan digunakan.

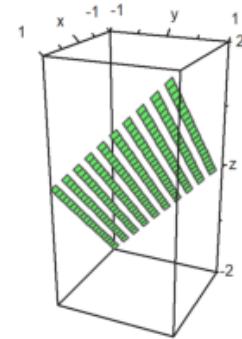
Dalam contoh, kami pertama-tama menghitung nilai. Kemudian kita menyesuaikan x dan y, jadi tengah vektor pada nilai yang digunakan.

```
>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);
```



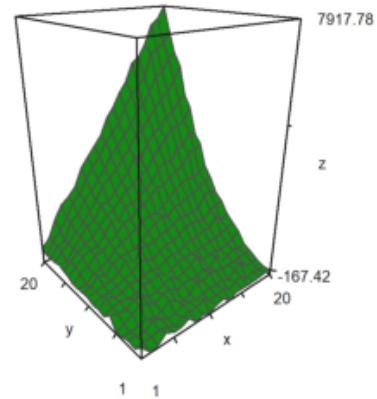
Ini memungkinkan untuk memisah plot dari permukaan dalam dua bagian atau lebih

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20);
```

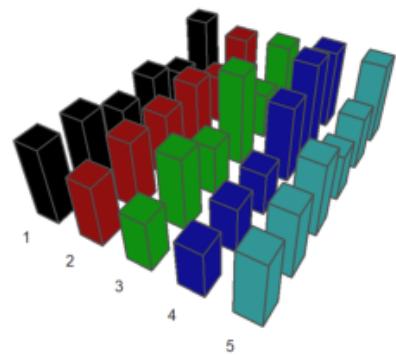


Jika memuat atau menghasilkan sebuah matriks data M dari sebuah file dan membutuhkan untuk membuat plot itu dalam 3D kamu dapat menskalakan matriks ke [-1, 1] dengan skala (M), atau skala matriks dengan >zscale. Ini dapat juga dikombinasikan dengan pemfaktoran skala tersendiri yang aman dapat diaplikasikan sebagai tambahan.

```
>i=1:20; j=i'; ...
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8):
```

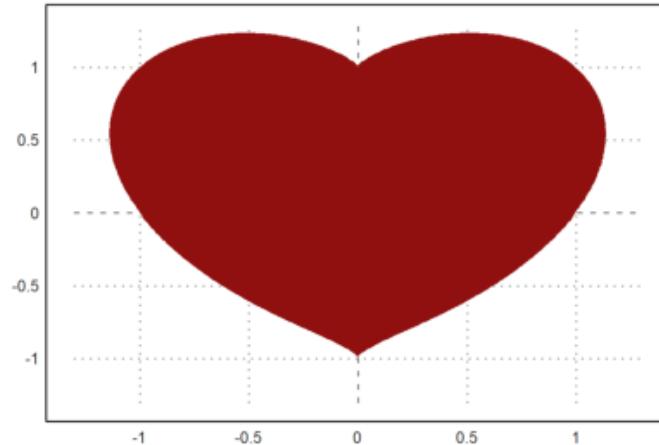


```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v',scols=1:5,ccols=[1:5]):
```



## Permukaan Benda Putar

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>style="#",color=red,<outline, ...
>level=[-2;0],n=100):
```



```
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

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

Kami mengharapkan untuk merubah kurva hati secara mengelilingi sumbu-y. Ini adalah ekspresi, yang mana mendefinisikan hati:

$$f(x, y) = (x^2 + y^2 - 1)^3 - x^2 \cdot y^3.$$

Selanjutnya kami mengatur

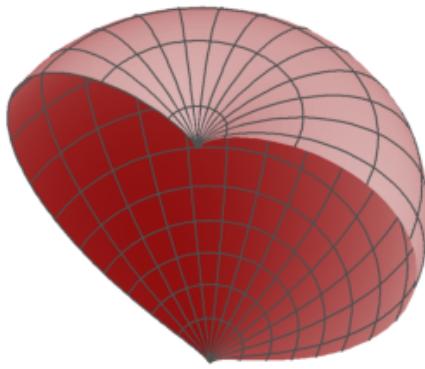
$$x = r \cos(a), \quad y = r \sin(a).$$

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2 \sin a) r^5}{16}$$

Ini memperbolehkan untuk mendefinisikan fungsi numerik, yang mana menyelesaikan untuk r, jika diberikan a. Dengan fungsi itu kami dapat membuat plot dengan mengubah hati 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°):
```

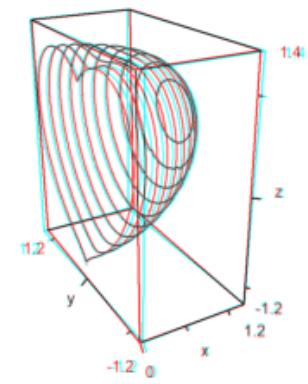


Berikut merupakan plot 3D dari penggambaran diatas diputar mengelilingi sumbu-y. Kami mendefinisikan fungsi yang mana mendeskripsikan objek

```
>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=[10,100,60],>anaglyph):
```



## Plot 3D Spesial

---

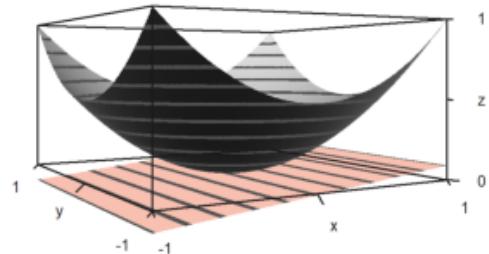
Fungsi plot3d sangat bagus untuk dimiliki, tetapi tidak memenuhi semua yang kita butuhkan. Disamping rutinitas umum yang lainnya, ini mungkin untuk mendapatkan sebuah plot berframe dari sebarang objek yang kamu suka.

Meskipun Euler bukanlah program 3D, ini dapat mengkombinasikan objek objek sederhana. Kami mencoba untuk memvisualisasikan sebuah paraboloida dan garis singgung.

```
>function myplot ...
y=-1:0.01:1; x=(-1:0.01:1)';
plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ...
    hues=0.5,>contour,color=orange);
h=holding(1);
plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
holding(h);
endfunction
```

Sekarang framedplot() menyediakan frame dan mengatur pandangan.

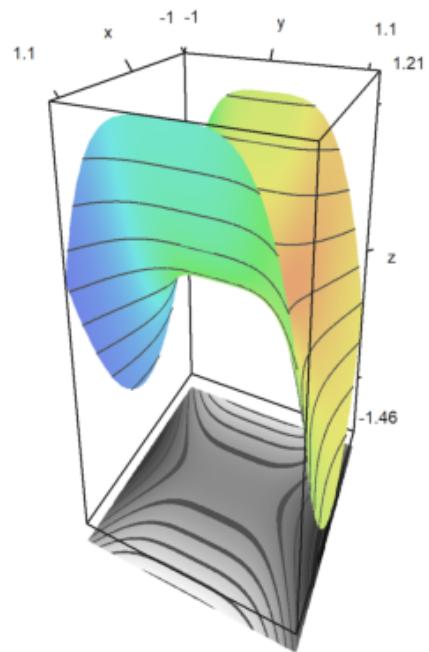
```
>framedplot("myplot",[-1,1,-1,1,0,1],height=0,angle=-30°, ...
> center=[0,0,-0.7],zoom=3):
```



Dengan jalan yang sama, kamu dapat mem-plotkan bidang kontur secara manual. Catatan bahwa plot3d() secara bawaaan mengatur jendela ke fullwindow(), tetapi asumsikan demikian untuk plotcontourplane().

```
>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):
```

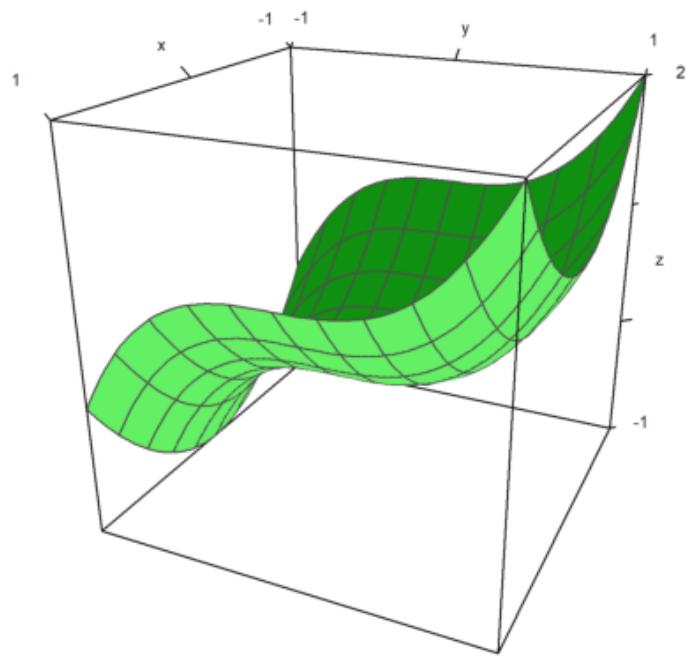


Euler dapat menggunakan frame untuk menghitung animasi dalam langkah sebelumnya

Satu fungsi, yang mana membuat teknik ini digunakan adalah rotate. Ini dapat merubah sudut dari pandang dan menggambar ulang sebuah plot 3D. Pemanggilan fungsi addpage() untuk setiap plot baru. Akhirnya plot ini dianimasikan.

Tolong pelajari sumber dari rotasi untuk lihat lebih detail.

```
>function testplot () := plot3d("x^2+y^3"); ...
>rotate("testplot"); testplot():
```



## Menggambar Povray

---

Dengan bantuan dari file Euler povray.e, Euler dapat membuat file Povray. Hasilnya sangat bagus untuk dilihat.

Kamu butuh menginstall Povray (32bit atau 64bit) dari <http://www.povray.org/>, dan ambillah sub-direktori "bin" dari Povray ke environment path atau atur ke variabel "defaultpovray" dengan path yang lengkap untuk menunjukkan ke "pvengine.exe".

Tampilan Povray dari Euler membuat file Povray dalam direktori home dari pengguna, dan memanggil Povray untuk menguraikan file ini. Nama file bawaan adalah current.pov dan direktori bawaan adalah eulerhome(), biasanya c:\Users\Username\Euler. Povray membuat sebuah file PNG, yang mana dimuat oleh Euler kedalam sebuah notebook. Untuk membersihkan file ini, gunakan povclear().

Fungsi pov3d sama seperti plot3d. Ini akan menghasilkan grafik dari fungsi  $f(x, y)$  atau sebuah permukaan dengan koordinat X, Y, Z dalam matriks, termasuk secara opsional garis level. Fungsi ini dimulai secara otomatis raytracer dan memuat pemandangan kedalam Euler notebook.

Disamping pov3d(), terdapat banyak sekali fungsi, yang mana menghasilkan objek Povray. Fungsi ini mengembalikan strings, mencakup kode Povray untuk objek. Untuk menggunakan fungsi-fungsi ini, mulai file Povray dengan povstart(). Lalu gunakan writeln(...) untuk menulis objek ke dalam file pemandangan. Terakhir, akhiri file dengan povend(). Secara bawaan, raytracer akan mulai dan file PNG akan dimasukkan kedalam notebook Euler.

Fungsi objek memiliki sebuah parameter yang bernama "look", yang mana membutuhkan string dengan kode Povray untuk tekstur dan akhiri objek. Fungsi povlook() dapat digunakan untuk menghasilkan string ini. Ini memiliki parameter untuk warna, transparansi, bayangan Phong, dan lainnya

Catatan bahwa semesta Povray memiliki sistem koordinat lain. Antar muka ini menerjemahkan semua koordinat kedalam sistem Povray. Jadi kamu dapat tetap berpikir dalam koordinat sistem Euler dengan menunjukkan z secara vertikal keatas, dan sumbu x, y, z dalam penalaran sistem tangan kanan.

Kamu butuh untuk memuat file povray.

```
>load povray;
```

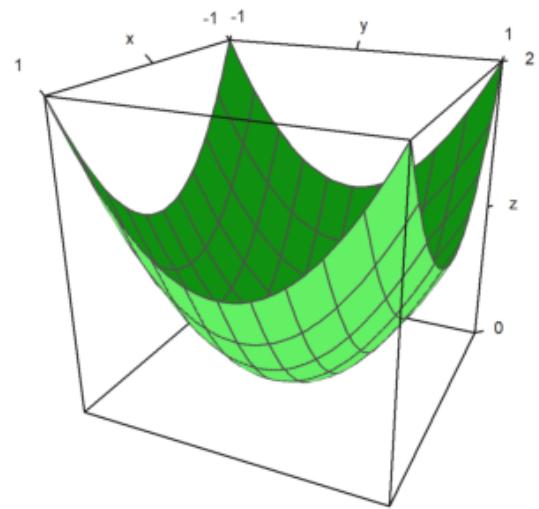
Pastikan, direktori bin Povray dalam jalur. Jika tidak atur mengikuti variabel, jadi ini mengandung jalur ke executable povray.

```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Untuk pertama kali pemakaian, kami mem-plotkan fungsi sederhana. Perintah berikut membuat sebuah file povray dalam direktori pengguna kamu, dan menjalankan Povray untuk penelusuran sinar dalam file. Jika kamu memulai perintah berikut, GUI Povray seharusnya terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, kamu akan diminta, untuk memberikan izin untuk menjalankan file exe. Kamu dapat mengklik cancel untuk memberhentikan pertanyaan selanjutnya. Kamu mungkin akan mengklik OK dalam jendela Povray untuk menyetujui percakapan awal dari Povray.

```
>plot3d("x^2+y^2",zoom=2):
```

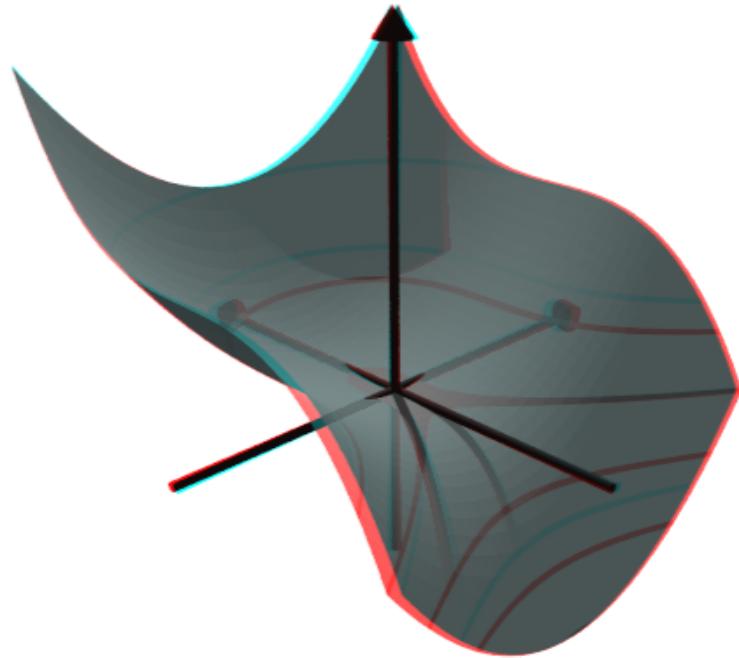


```
>pov3d("x^2+y^2",zoom=3);
```



Kami dapat membuat fungsi transparan dan menambahkan pada hasil akhir lainnya. Kami juga dapat menambahkan garis level ke fungsi plot.

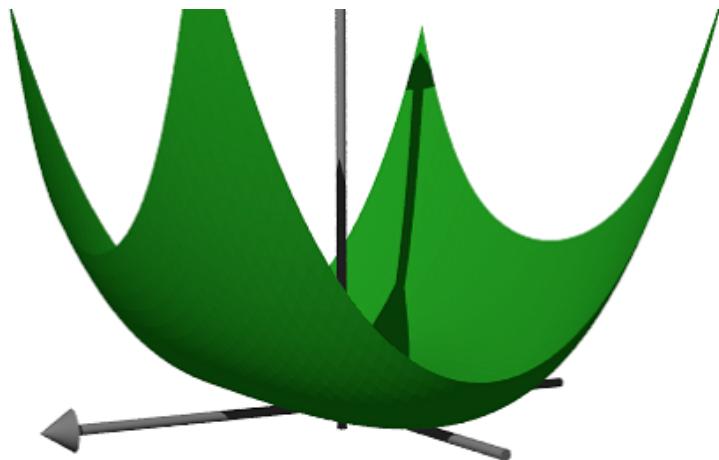
```
>pov3d("x^2+y^3",axiscolor=red,angle=-45°,>anaglyph, ...
> look=povlook(cyan,0.2),level=-1:0.5:1,zoom=3.8);
```



Terkadang ini dibutuhkan untuk mencegah penskalaan fungsi dan skala fungsi dengan tangan.

Kami memplot himpunan titik-titik dalam bidang kompleks, dimana hasil dari jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=2, ...
> angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=10°,n=50, ...
> <fscale,zoom=3.8);
```



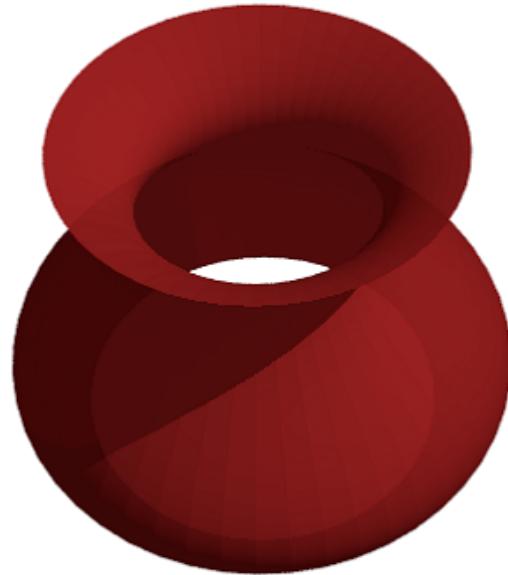
## Membuat Plot dengan Koordinat

---

Selain fungsi, kami dapat mem-plot dengan koordinat. Seperti dalam plot3d, kami membutuhkan tiga matiks untuk mendefinisikan objek.

Dalam contoh kami memutar mengelilingi sumbu-z.

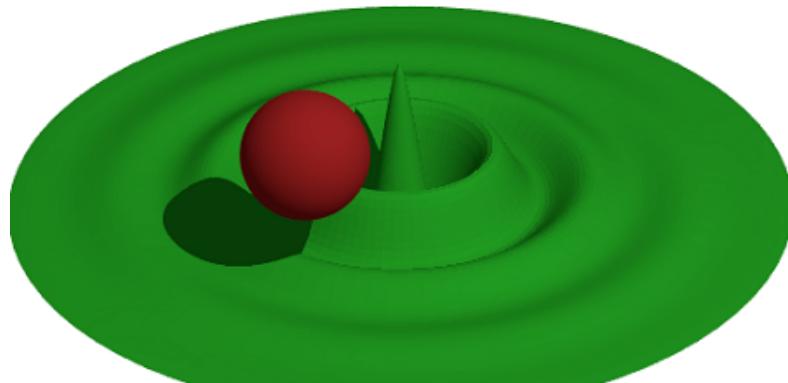
```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,50)';
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light=[10,5,15]);
```



Dalam contoh berikut, kami membuat plot glombang yang mengecil. Kami membuat gelombang dengan bahasa matriks Euler.

Kami juga menunjukkan tambahan objek yang dapat ditambahkan ke layar pov3d. Untuk pembuatan objek, lihat contoh berikut. Catatan bahwa plot3d menskalakan plot, jadi ini akan menyesuaikan ke unit kubus.

```
>r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ...
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3; ...
>pov3d(x,y,z,zoom=6,axis=0,height=30°,add=povsphere([0.5,0,0.25],0.15,povlook(red)), ...
> w=500,h=300);
```



Dengan metode bayangan lebih lanjut dari Povray, sangat sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya pada saat batasan dan dalam bayangan trik ini mungkin akan terlihat jelas.

Untuk ini, kami membutuhkan vektor normal dalam setiap titik matriks.

```
>Z &= x^2*y^3
```

$$\begin{matrix} 2 & 3 \\ x & y \end{matrix}$$

Persamaan dari permukaan adalah  $[x, y, Z]$ . Kami menghitung dua turunan ke  $x$  dan  $y$  dari ini dan mengambil hasil kali silang sebagai normal.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

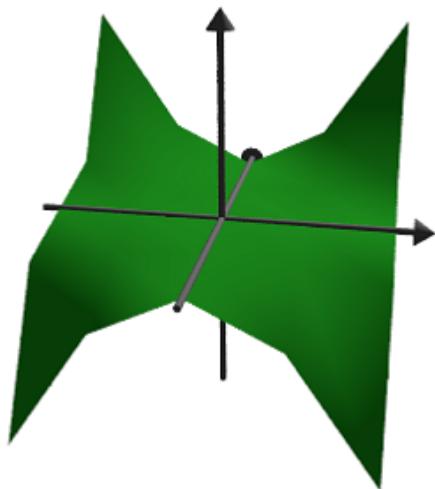
Kami mendefinisikan normal sebagai perkalian produk dari turunan ini dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

$$\begin{bmatrix} 3 & 2 & 2 \\ -2x^3y & -3x^2y & 1 \end{bmatrix}$$

Kami hanya menggunakan 25 titik.

```
>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 merupakan knot Trefoil dilakukan dengan A. Busser dalam Povray. Ini merupakan versi yang telah diperbarui dalam contoh ini.

Contoh: Examples\Trefoil Knot | Trefoil Knot

Untuk pemandangan yang bags dengan tidak beberapa titik, kami menambahkan vektor normal disini. Kami menggunakan Maxima untuk menghitung normal untuk kami. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolik.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
>Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
>Z &= sin(x)+2*cos(3*y);
```

Lalu dua vektor turunan ke x dan y.

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang normal, yang mana adalah hasil kali dari dua turunan.

```
>dn &= crossproduct(dx,dy);
```

Kami sekarang menilai semua ini secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

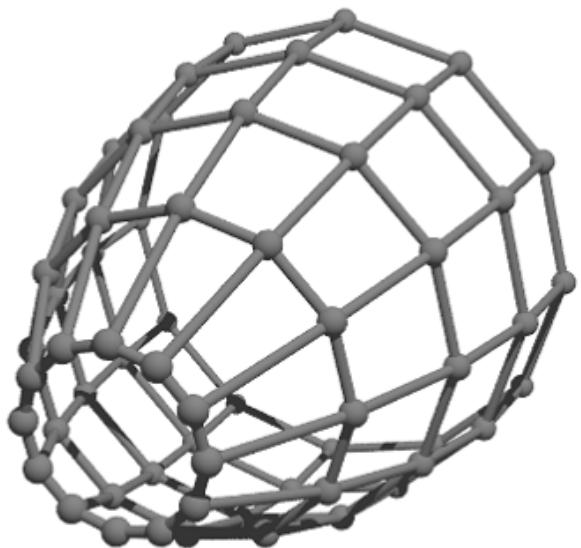
Vektor-vektor normal merupakan penilaian dari ekspresi simbolik  $dn[i]$  untuk  $i=1,2,3$ . Sintaks untuk ini adalah &”ekspresi”(parameter). Ini merupakan sebuah alternatif dari metode dalam contoh sebelumnya, kami dapat mendefinisikan ekspresi simbolik  $NX$ ,  $NY$ ,  $NZ$  terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),>anaglyph,axis=0,zoom=5,w=450,h=350, ...
> <shadow,look=povlook(blue), ...
> xv=&"dn[1] "(x,y), yv=&"dn[2] "(x,y), zv=&"dn[3] "(x,y));
```



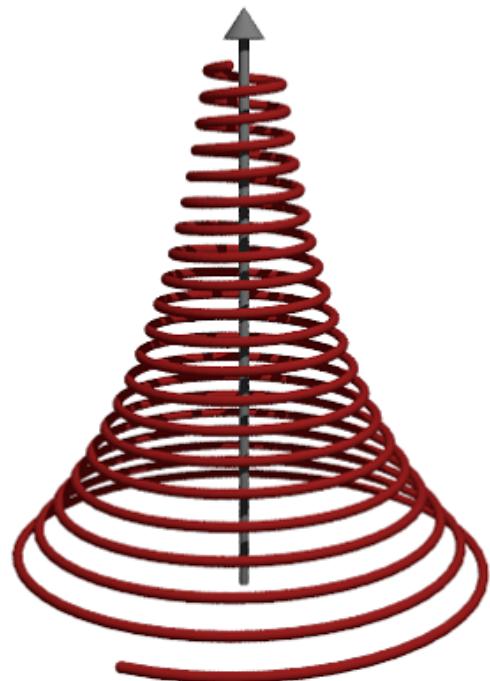
Kami juga dapat menghasilkan jaring-jaring dalam 3D.

```
>povstart(zoom=4); ...
>x=-1:0.5:1; r=1-(x+1)^2/6; ...
>t=(0°:30°:360°)'; y=r*cos(t); z=r*sin(t); ...
>writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...
>povend();
```



Dengan povgrid(), kurva-kurva jadi memungkinkan.

```
>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

---

Diatas, kami menggunakan pov3d untuk mem-plot permukaan. Antarmka povray dalam Euler dapat membuat objek Povray. Objek-objek ini disimpan sebagai string dalam Euler, dan perlu untuk menulis ke file Povray.

Kami mulai mengeluarkan dengan povstart().

```
>povstart(zoom=4);
```

Pertama kami definisikan tiga silinder dan menyimpannya ke string dalam Euler.

Fungsi-fungsi povx() dan lainnya secara sederhana mengembalikan vektor [1, 0, 0], yang mana dapat digunakan sebagai gantinya.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...
>c2=povcylinder(-povy,povy,1,povlook(yellow)); ...
>c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

The strings contain Povray code, which we need not understand at that point.

```
>c2
```

```
cylinder { <0,0,-1>, <0,0,1>, 1
    texture { pigment { color rgb <0.941176,0.941176,0.392157> }  }
    finish { ambient 0.2 }
}
```

Seperti yang kamu lihat, kami menambahkan tekstur ke objek dalam tiga warna yang berbeda.

Ini dapat dilakukan dengan `povlook()`, yang mana mengembalikan sebuah string dengan code Povray yang relevan. Kami dapat menggunakan warna bawaan Euler, atau mendefinisikan sendiri warna kami. Kami juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

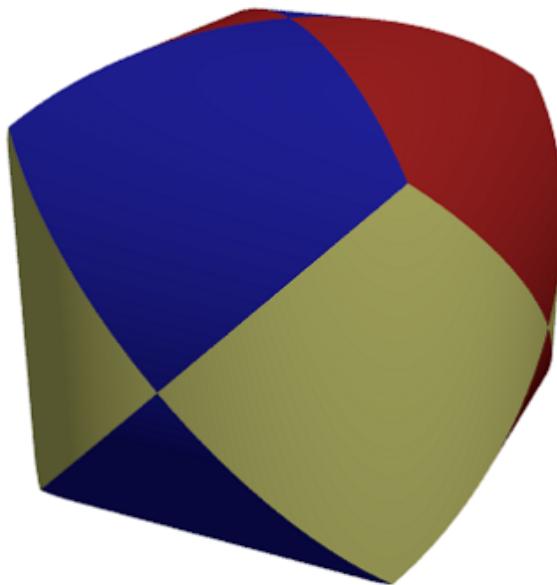
```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> }  }
finish { ambient 0.5 }
```

Sekarang kami mendefinisikan objek yang saling memotong dan menulis hasilnya kedalam file.

```
>writeln(povintersection([c1,c2,c3]));
```

Potongan dari silinder tiga sangat sulit untuk digambarkan, jika kamu belum pernah melihatnya.

```
>povend;
```



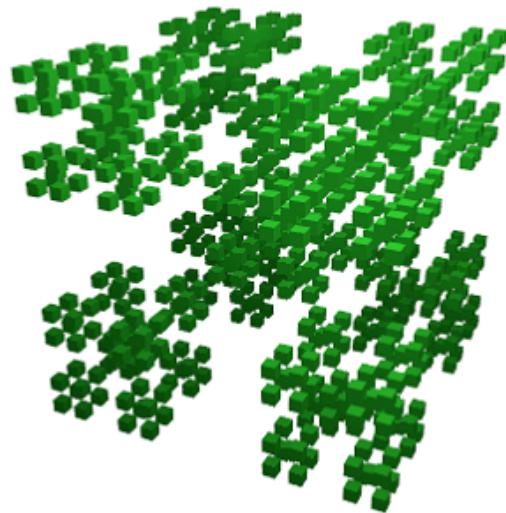
Fungsi-fungsi berikut membuat sebuah fraktal secara rekursif.

Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana. povbox() fungsi mengembalikan sebuah string, yang mana memuat koordinat-koordinat boks, tekstur dan selesai.

```
>function onebox(x,y,z,d) := povbox([x,y,z],[x+d,y+d,z+d],povlook());  
>function fractal (x,y,z,h,n) ...
```

```
if n==1 then writeln(onebox(x,y,z,h));  
else  
    h=h/3;  
    fractal(x,y,z,h,n-1);  
    fractal(x+2*h,y,z,h,n-1);  
    fractal(x,y+2*h,z,h,n-1);  
    fractal(x,y,z+2*h,h,n-1);  
    fractal(x+2*h,y+2*h,z,h,n-1);  
    fractal(x+2*h,y,z+2*h,h,n-1);  
    fractal(x,y+2*h,z+2*h,h,n-1);  
    fractal(x+2*h,y+2*h,z+2*h,h,n-1);  
    fractal(x+h,y+h,z+h,h,n-1);  
endif;  
endfunction
```

```
>povstart(fade=10,<shadow);  
>fractal(-1,-1,-1,2,4);  
>povend();
```



Perbedaan memungkinkan pemotongan satu objek dari objek lainnya. Seperti perpotongan, ini merupakan bagian dari objek CSG dari Povray.

```
>povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kami mendefinisikan sebuah objek dalam Povray, daripada menggunakan sebuah string dalam Euler. Definisi ditulis ke file secara langsung.

Sebuah koordinat dimensi tiga dari -1 berarti [-1, -1, -1].

```
>povdefine("mycube",povbox(-1,1));
```

Kami menggunakan objek ini dengan povobject(), yang mana mengembalikan sebuah string seperti biasanya.

```
>c1=povobject("mycube",povlook(red));
```

Kami membuat sebuah kubus kedua dan merotasi dan men-skalakan sedikit.

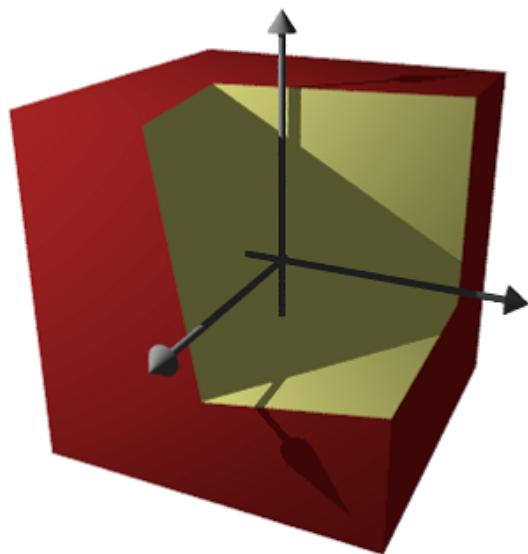
```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...
>   rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Lalu kami ambil selisih 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-Fungsi Implisit

---

Povray dapat mem-plot himpunan yang mana  $f(x, y, z) = 0$ , seperti parameter implisit dalam plot3d. Namun, hasilnya terlihat lebih baik.

Sintaks untuk fungsi-fungsi sedikit berbeda. Kamu tidak dapat menggunakan keluaran dari Maxima atau ekspresi Euler.

$$((x^2 + y^2 - c^2)^2 + (z^2 - 1)^2) * ((y^2 + z^2 - c^2)^2 + (x^2 - 1)^2) * ((z^2 + x^2 - c^2)^2 + (y^2 - 1)^2) = d$$

```
>povstart(angle=70°,height=50°,zoom=4);
>c=0.1; d=0.1; ...
>writeln(povsurface("(pow(pow(x,2)+pow(y,2)-pow(c,2),2)+pow(pow(z,2)-1,2))*(pow(pow(y,2)+pow(z,2)-po
>povend();
```

Error : Povray error!

Error generated by error() command

```
povray:
    error("Povray error!");
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

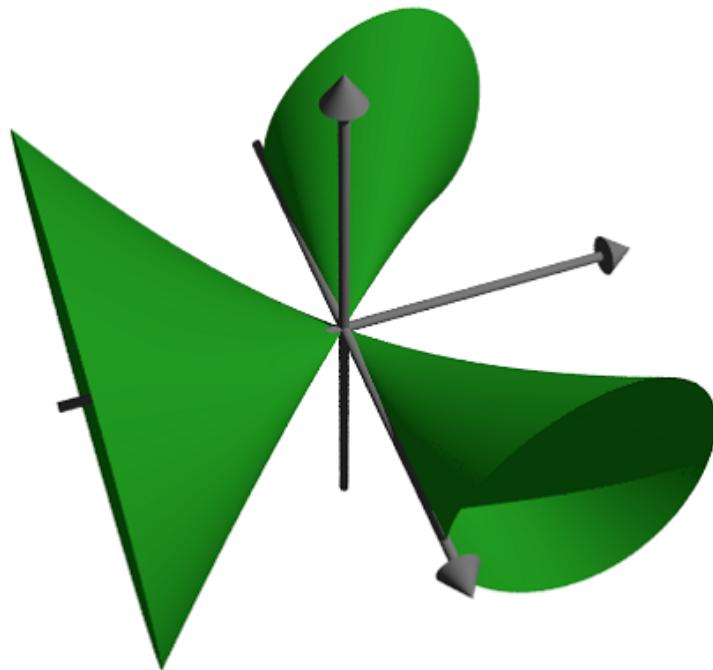
```
>povstart(angle=25°,height=10°);
>writeln(povsurface("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,2,"")));
>povend();
```



```
>povstart(angle=70°,height=50°,zoom=4);
```

Buat permukaan implisit. Catat sintaks yang berbeda dari ekspresi.

```
>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...
>writeAxes(); ...
>povend();
```



## Objek Mesh

---

Contoh ini, kami menunjukkan bagaimana untuk membuat objek mesh, dan menggambarnya dengan informasi tambahan.

Kami ingin memaksimalkan  $xt$  dibawah kondisi  $x + y = 1$  dan mendemonstrasikan sentuan tangential 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. Itu dapat menerima vektor normal seperti pov3d().

Dibawah ini mendefinisikan objek mesh dan menulisnya langsung kedalam 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 cakram, yang mana akan dipotong dengan permukaan.

```
>cl=povdisc([0.5,0.5,0],[1,1,0],2); ...
>ll=povdisc([0,0,1/4],[0,0,1],2);
```

Tulis pemukaan minus dua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tulis potongan keduanya.

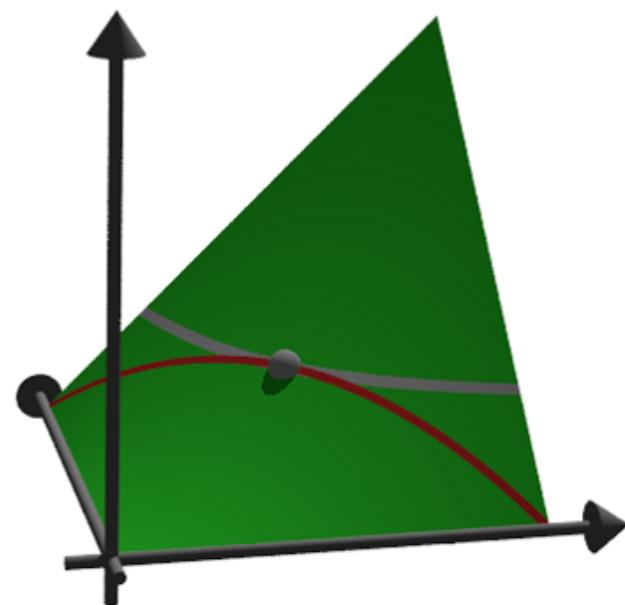
```
>writeln(povintersection([mesh,cl],povlook(red))); ...
>writeln(povintersection([mesh,ll],povlook(gray)));
```

Tulis maksimal titik.

```
>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();
```



## Anaglyphs dalam Povray

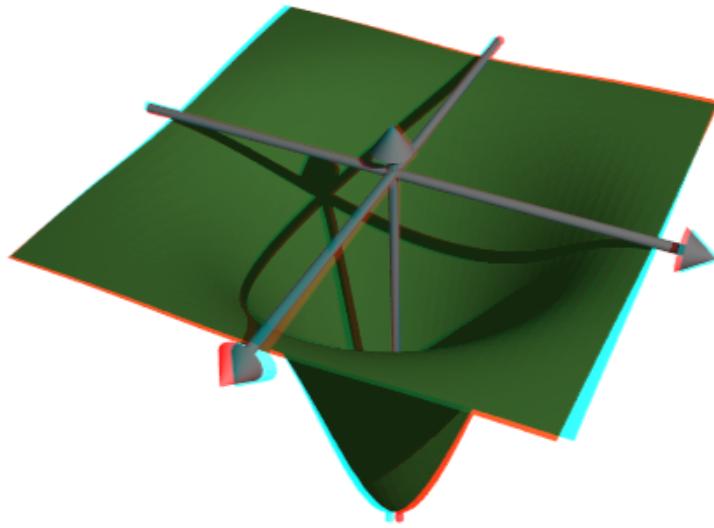
---

Untuk membuat sebuah anaglyph untuk kacamata merah/cyan, Povray harus menjalankan dua kamera dengan posisi yang berbeda. Ini akan menghasilkan dua Povray file dan dua file PNG, yang mana dimuat dalam fungsi loadanaglyph().

Tentu saja, kamu membutuhkan kacamata merah/cyan untuk melihat contoh berikut secara benar.

Fungsi pov3d() memiliki penggantian sederhana untuk membuat anaglyphs.

```
>pov3d("-exp(-x^2-y^2)/2",r=2,height=45°,>anaglyph, ...
> center=[0,0,0.5],zoom=3.5);
```



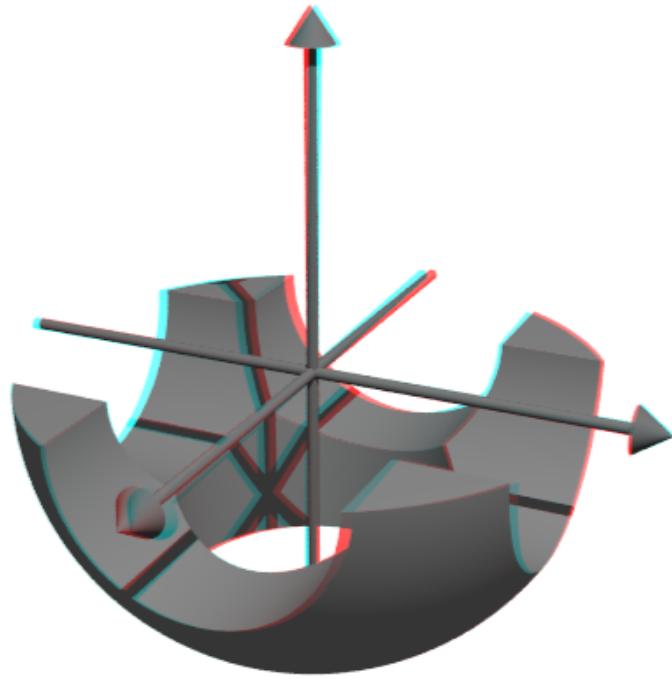
Jika kamu membuat sebuah pemandangan dengan objek, kamu perlu untuk menempatkan pembuatan pemandangan ke dalam suatu fungsi dan jalankan keduannya dengan nilai berbeda untuk parameter anaglyph.

```
>function myscene ...
```

```
s=povsphere(povc,1);
cl=povcylinder(-povz,povz,0.5);
clk=povobject(cl,rotate=xrotate(90°));
cly=povobject(cl,rotate=yrotate(90°));
c=povbox([-1,-1,0],1);
un=povunion([cl,clk,cly,c]);
obj=povdifference(s,un,povlook(red));
writeln(obj);
writeAxes();
endfunction
```

Fungsi povanglyph() akan menjalankan ini semua. Parameter seperti povstart() dan povend() dikombinasikan.

```
>povanaglyph("myscene",zoom=4.5);
```



## Mendefinisikan Objek milik sendiri

---

Tampilan povray dari Euler memiliki banyak objek. Tetapi kamu tidak dibatasi olehnya. Kamu dapat membuat objekmu tersendiri, yang mana mengombinasikan objek yang lainnya, atau secara penuh objek baru.

Kami mendemonstrasikan sebuah torys. Perintah Povray untuk ini adalah "torus". Jadi kami mengembangkan sebuah string dengan perintah ini dan parameternya. Catatan bahwa torus selalu berada ditengah tepat pada titik asal.

```
>function povdonat (r1,r2,look "") ...  
  
    return "torus {" + r1 + "," + r2 + look + "}";  
endfunction
```

Ini merupakan torus pertama kami.

```
>t1=povdonat(0.8,0.2)  
  
torus {0.8,0.2}
```

Andaikan kami menggunakan objek ini untuk membuat sebuah torus kedua, mentranslatikan dan merotasinya.

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
object { torus {0.8,0.2}
  rotate 90 *x
  translate <0.8,0,0>
}
```

Sekarang kita menempatkan objek kedalam sebuah tempat. Untuk tampilannya, kami menggunakan Phong Shading.

```
>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...
>writeln(povobject(t1,povlook(green,phong=1))); ...
>writeln(povobject(t2,povlook(green,phong=1))); ...
```

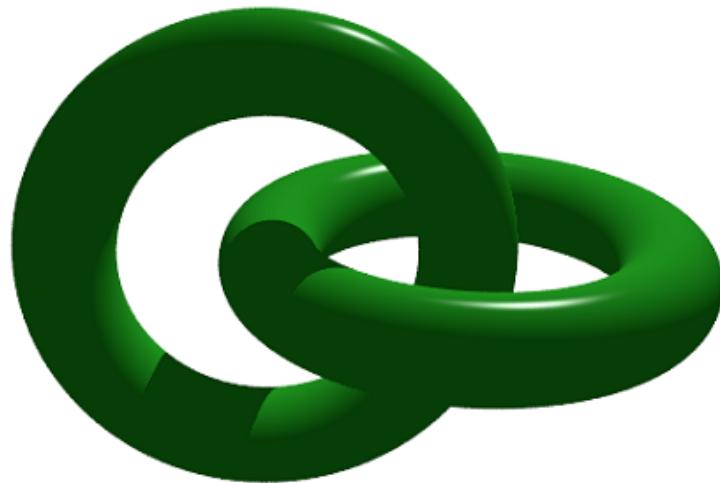
```
>povend();
```

memanggil program Povray. Namun, dalam kasus error, ini tidak akan menampilkan error. Kamu harus melakukan

```
>povend(<exit>);
```

jika apapun tidak bekerja. Ini akan meninggalkan jendela Povray terbuka.

```
>povend(h=320,w=480);
```



Ini merupakan contoh yang lebih elaborasi. Kami menyelesaikan

$$Ax \leq b, \quad x \geq 0, \quad c.x \rightarrow \text{Max.}$$

dan melihatkan titik-titik yang mungkin 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 cek, jika contoh ini memiliki sebuah solusi pada semuanya.

```
>x=simplex(A,b,c,>max,>check)'
```

[0, 1, 0.5]

Ya, itu mempunyainya.

Selanjutnya kami mendefinisikan dua objek. Pertama merupakan sebuah bidang

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look=""') ...
```

```
    return povplane(a,b,look)
endfunction
```

Lalu kami mendefinisikan perpotongan dari pertengahan bidang semua dan sebuah kubus.

```
>function adm (A, b, r, look="") ...
ol=[];
loop 1 to rows(A); ol=ol|oneplane(A[#,b[#]]; end;
ol=ol|povbox([0,0,0],[r,r,r]);
return povintersection(ol,look);
endfunction
```

Sekarang kami dapat membuat plot hasilnya.

```
>povstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); ...
>writeln(adm(A,b,2,povlook(green,0.4))); ...
>writeAxes(0,1.3,0,1.6,0,1.5); ...
```

Berikut merupakan sebuah lingkaran di sekitar optimum.

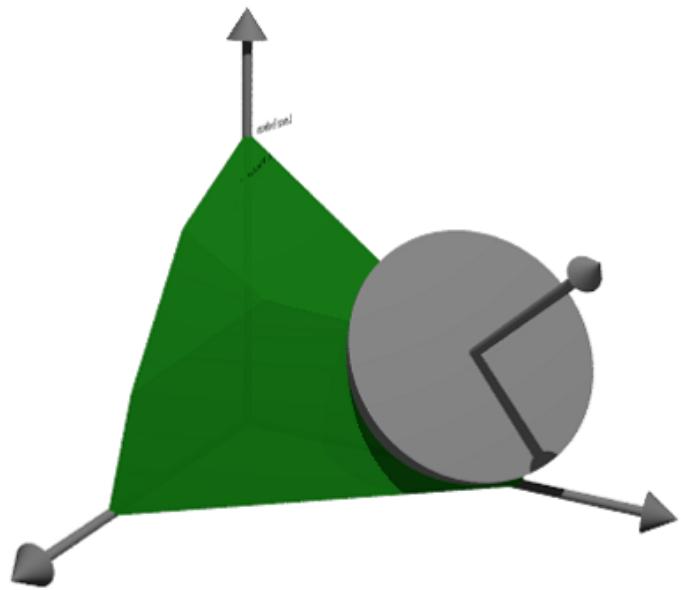
```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
> povlook(red,0.9)));
```

Dan sebuah error dalam arah dari optimum.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami tambahkan teks ke layar. Teks hanya sebuah objek 3D. Kami butuh tempat untuk membalikkannya mengikuti pandangan kita.

```
>writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=5°)); ...
>povend();
```



## **Contoh Lainnya**

---

Kamu dapat menemukan contoh lainnya untuk Povray dalam Euler dengan file berikut.

See: Examples/Dandelin Spheres

See: Examples/Donut Math

See: Examples/Trefoil Knot

See: Examples/Optimization by Affine Scaling

## Chapter 4

# Menggunakan EMT untuk Kalkulus

# Kalkulus dengan EMT

---

Materi Kalkulus mencakup di antaranya:

- Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi)
- Limit Fungsi,
- Turunan Fungsi,
- Integral Tak Tentu,
- Integral Tentu dan Aplikasinya,
- Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

## Mendefinisikan Fungsi

---

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:

- Menggunakan format `nama_fungsi := rumus fungsi` (untuk fungsi numerik),
- Menggunakan format `nama_fungsi &= rumus fungsi` (untuk fungsi simbolik, namun dapat dihitung secara numerik),
- Menggunakan format `nama_fungsi &&= rumus fungsi` (untuk fungsi simbolik murni, tidak dapat dihitung langsung),
- Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah `function` (bukan sebagai ekspresi).

Berikut adalah beberapa contoh cara mendefinisikan fungsi:

$$f(x) = 2x^2 + e^{\sin(x)}.$$

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik  
>f(0), f(1), f(pi)
```

```
1  
4.31977682472  
20.7392088022
```

```
>f(a) // tidak dapat dihitung nilainya
```

```
Variable or function a not found.  
Error in:  
f(a) // tidak dapat dihitung nilainya ...  
^
```

Silakan Anda plot kurva fungsi di atas!

Berikutnya kita definisikan fungsi:

$$g(x) = \frac{\sqrt{x^2 - 3x}}{x + 1}.$$

```
>function g(x) := sqrt(x^2-3*x)/(x+1)  
>g(3)
```

0

```
>g(0)
```

0

```
>g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik
```

```
Floating point error!
Error in sqrt
Try "trace errors" to inspect local variables after errors.
g:
    useglobal; return sqrt(x^2-3*x)/(x+1)
Error in:
g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik ...  
^
```

Silakan Anda plot kurva fungsi di atas!

```
>f(g(5)) // komposisi fungsi
```

2.20920171961

```
>g(f(5))
```

0.950898070639

```
>function h(x) := f(g(x)) // definisi komposisi fungsi  
>h(5) // sama dengan f(g(5))
```

2.20920171961

Silakan Anda plot kurva fungsi komposisi fungsi f dan g:

$$h(x) = f(g(x))$$

dan

$$u(x) = g(f(x))$$

bersama-sama kurva fungsi f dan g dalam satu bidang koordinat.

```
>f(0:10) // nilai-nilai f(0), f(1), f(2), ..., f(10)
```

```
[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,  
99.929, 130.69, 163.51, 200.58]
```

```
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
```

```
[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,  
99.929, 130.69, 163.51, 200.58]
```

```
>gmap(200:210)
```

```
[0.987534, 0.987596, 0.987657, 0.987718, 0.987778, 0.987837,  
0.987896, 0.987954, 0.988012, 0.988069, 0.988126]
```

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0. \end{cases}$$

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format `:=`, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.

```
>function map f(x) ...
if x>0 then return x^3
else return x^2
endif;
endfunction
```

```
>f(1)
```

1

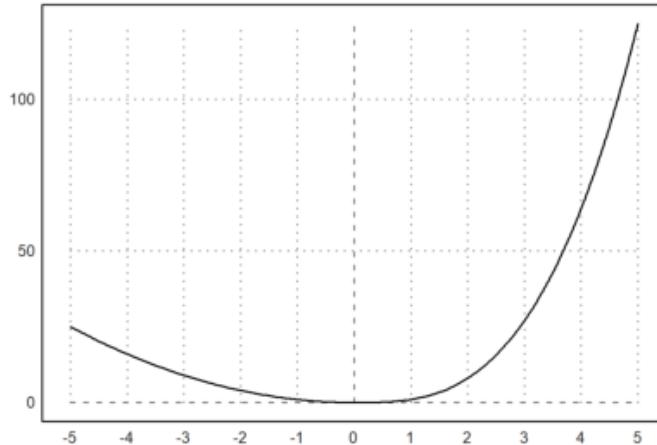
```
>f(-2)
```

4

```
>f(-5:5)
```

[25, 16, 9, 4, 1, 0, 1, 8, 27, 64, 125]

```
>aspect(1.5); plot2d("f(x)",-5,5):
```



```
>function f(x) &= 2*E^x // fungsi simbolik
```

$$2 e^x$$

```
>$f(a) // nilai fungsi secara simbolik
```

$$2 e^a$$

```
>f(E) // nilai fungsi berupa bilangan desimal
```

30.308524483

```
>$f(E), $float(%)
```

30.30852448295852

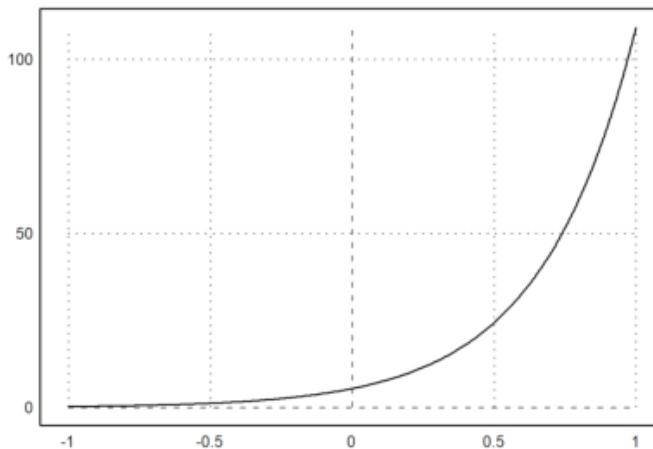
```
>function g(x) &= 3*x+1
```

3 x + 1

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

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

```
>plot2d("h(x)", -1, 1):
```



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.

1. fungsi linear

```
>function map linearFunctionF(x) &= 2*x + 1
```

$$2x + 1$$

```
>linearFunctionF(2)
```

Nilai fungsi apabila  $x = a$ , dengan  $a$  adalah sebuah konstanta

```
>$linearFunctionF(a)
```

$$2a + 1$$

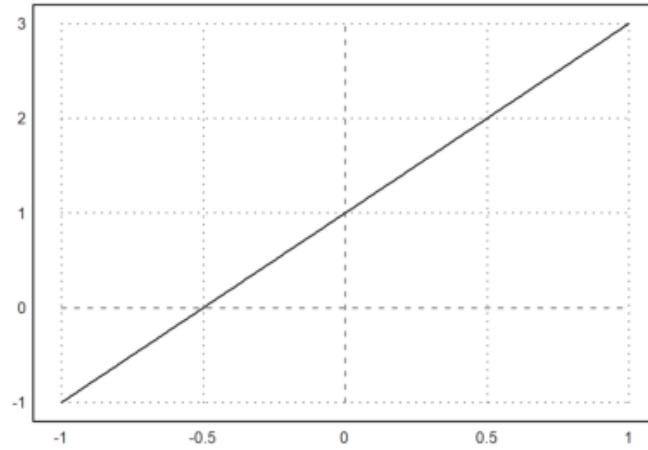
Nilai fungsi apabila diberi nilai selang (vektor)

```
>linearFunctionF(-1:1)
```

$[-1, 1, 3]$

Membuat plotnya

```
>plot2d("linearFunctionF", -1, 1):
```



## 2. fungsi polinomial

```
>function map polynomialFunctionF(x) &= x^2 - x + 4
```

$$x^2 - x + 4$$

```
>polynomialFunctionF(3)
```

Nilai dari fungsi polinomial, jika diberikan  $x = b$ , dengan  $b$  adalah sebuah konstanta

```
>$polynomialFunctionF(b)
```

$$b^2 - b + 4$$

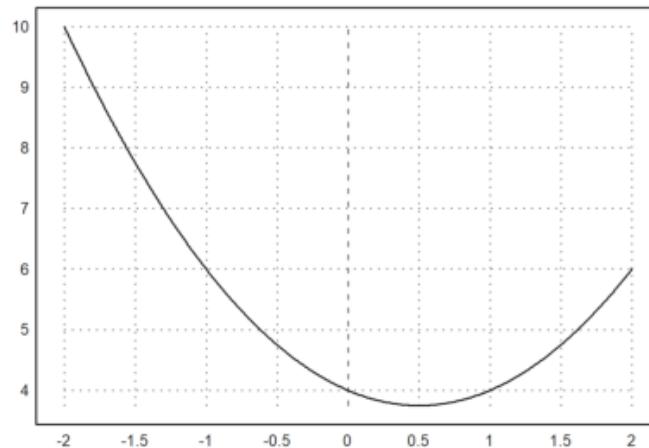
Jika diberikan selang nilai dalam bentuk vektor

```
>polynomialFunctionF(-2:2)
```

$$[10, \quad 6, \quad 4, \quad 4, \quad 6]$$

Akan digambar plotnya

```
>plot2d("polynomialFunctionF", -2, 2):
```



Dibuat komposisi fungsi dengan komposisi sebagai berikut  
fungsi\_linear(fungsi\_polinomial(x))

```
>function map compositionLinAndPolyFuncF(x) &= linearFunctionF(polynomialFunctionF(x))
```

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

```
>compositionLinAndPolyFuncF(10)
```

189

Jika diberikan konstanta  $x = c$

```
>$compositionLinAndPolyFuncF(c)
```

$$2(c^2 - c + 4) + 1$$

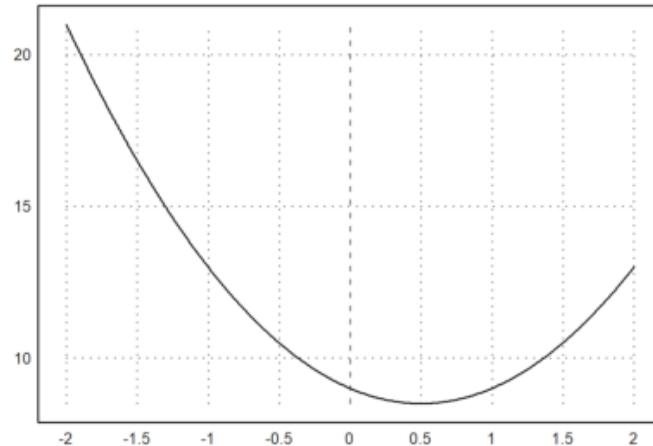
Diberikan selang  $[-2, 2]$  sebagai vektor

```
>compositionLinAndPolyFuncF(-2:2)
```

[21, 13, 9, 9, 13]

Gambar plot dari komposisi fungsinya

```
>plot2d("compositionLinAndPolyFuncF", -2, 2):
```



Dibuat komposisi fungsi dengan komposisi sebagai berikut  
fungsi\_polinomial(fungsi\_linear(x))

```
>function map compositionLinAndPolyFuncG(x) &= polynomialFunctionF(linearFunctionF(x))
```

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

```
>compositionLinAndPolyFuncG(10)
```

424

Jika diberikan sebarang konstanta  $x = d$

```
>$compositionLinAndPolyFuncG(d)
```

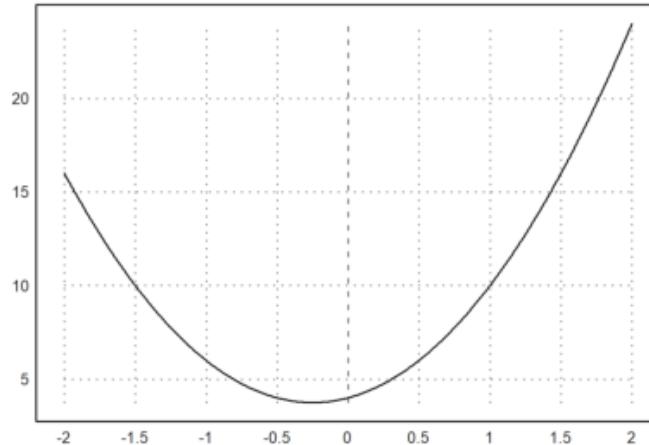
$$(2d + 1)^2 - 2d + 3$$

Apabila diberikan selang  $[-2, 2]$  berupa vektor

```
>compositionLinAndPolyFuncG(-2:2)
```

[16, 6, 4, 10, 24]

```
>plot2d("compositionLinAndPolyFuncG", -2, 2):
```



### 3. fungsi trigonometri

```
>function map trigonometryFunctionF(x) &= 2*sin(x) + 3*cos(x)
```

$$2 \sin(x) + 3 \cos(x)$$

```
>trigonometryFunctionF(2*pi)
```

3

```
>$trigonometryFunctionF(e)
```

$$2 \sin e + 3 \cos e$$

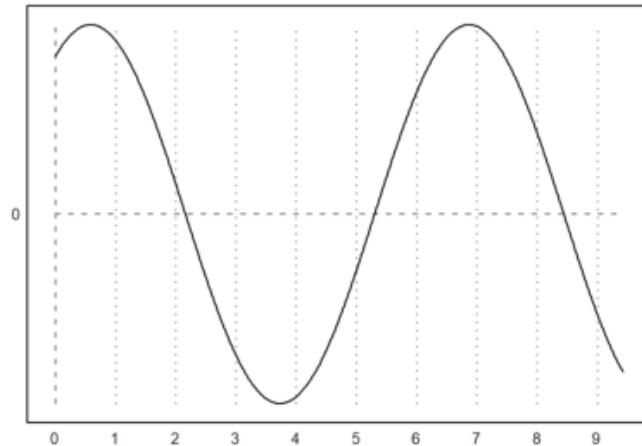
Jika diberikan selang  
maxima: [0, \pi/2, \pi, 3\*\pi/2, 2\*pi]

```
>trigonometryFunctionF(0:(1/2)*Pi:2*Pi)
```

[3, 2, -3, -2, 3]

Membuat plotnya

```
>plot2d("trigonometryFunctionF", 0, 3*Pi):
```



Komposisi fungsi trigonometri dengan fungsi linear

```
>function map compositionLinAndTriFuncF(x) &= trigonometryFunctionF(linearFunctionF(x))
```

$$2 \sin(2x + 1) + 3 \cos(2x + 1)$$

```
>compositionLinAndTriFuncF(Pi)
```

3.30384888722

Diberikan konstanta  $x = f$

```
>$compositionLinAndTriFuncF(f)
```

$$2 \sin(2f + 1) + 3 \cos(2f + 1)$$

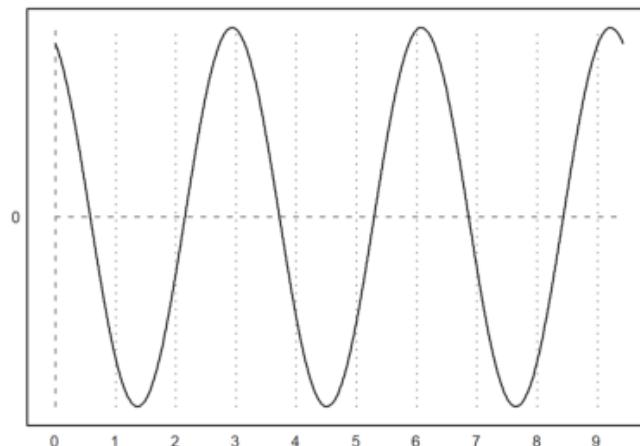
Apabila diberikan selang  
maxima: [0,  $\pi/2$ ,  $\pi$ ,  $3\pi/2$ ,  $2\pi$ ]

```
>compositionLinAndTriFuncF(0:(1/2)*Pi:2*Pi)
```

[3.30385, -3.30385, 3.30385, -3.30385, 3.30385]

Buat plotnya untuk fungsi komposisi

```
>plot2d("compositionLinAndTriFuncF", 0, 3*Pi):
```



4. fungsi eksponensial

```
>function map exponentialFunctionF(x) &= 100*3^(x/2)
```

$$100 \cdot 3^{x/2}$$

```
>exponentialFunction(4)
```

900

Jika diberikan sebarang konstanta g

```
>$exponentialFunction(g)
```

$100 \cdot 3^{\frac{g}{2}}$

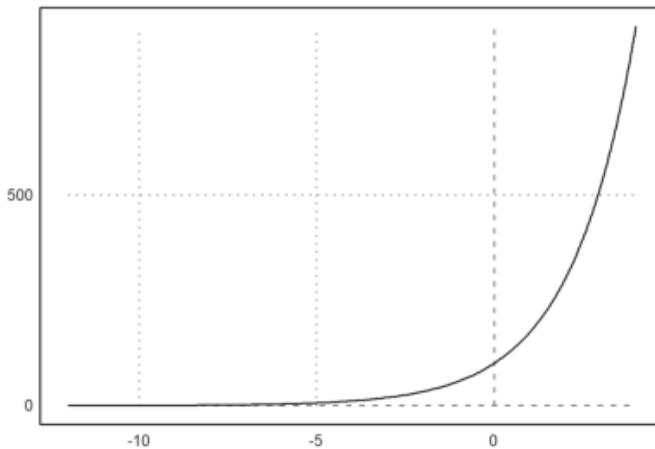
Apabila diberikan vektor rentang [-12, -8]

```
>exponentialFunction(-12:-8)
```

[0.137174, 0.237593, 0.411523, 0.712778, 1.23457]

Membuat plot

```
>plot2d("exponentialFunctionF", -12, 4):
```



Komposisi eksponensial fungsi dengan polinomial fungsi

```
>function map compositionExpAndPolyFuncF(x) &= exponentialFunctionF(polynomialFunctionF(x))
```

$$\begin{array}{r}
 \phantom{0}2 \\
 x^2 - x + 4 \\
 \hline
 \phantom{0}2 \\
 100 \ 3
 \end{array}$$

```
>compositionExpAndPolyFuncF(0)
```

900

Apabila diberikan sebarang konstanta h

```
>$compositionExpAndPolyFuncF(h)
```

$$100 \ 3^{\frac{h^2-h+4}{2}}$$

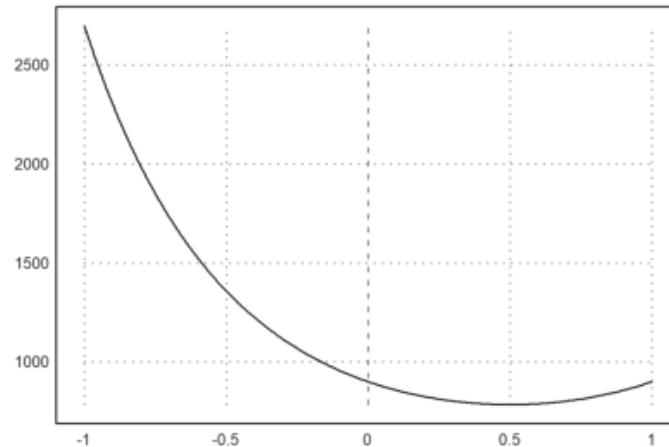
Diberikan rentang nilai [-1,1] berupa vektor

```
>compositionExpAndPolyFuncF(-1:1)
```

[2700, 900, 900]

Membuat plot dari komposisi fungsi tersebut

```
>plot2d("compositionExpAndPolyFuncF", -1, 1):
```



## 5. fungsi logaritmik

```
>function map logaritmicFunctionF(x) &= 2*log(x)
```

$$2 \log(x)$$

```
>logaritmicFunctionF(E^2)
```

4

Diberikan sebarang konstanta i

```
>$logaritmicFunctionF(i)
```

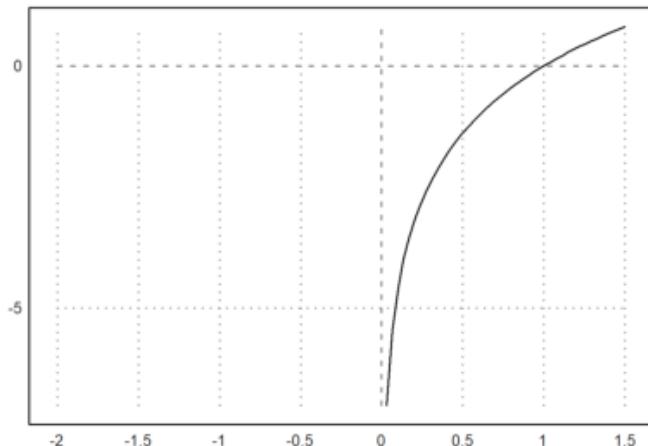
$2 \log i$

```
>logaritmicFunctionF(E^(1/2):E:E^2)
```

[1, 2.94815, 3.91604]

Apabila diberikan vektor  
maxima:  $[E^{(1/2)}, E, E^2]$

```
>plot2d("logaritmicFunctionF", -2, 1.5):
```



```
>function map compositionLinAndLogFuncF(x) &= logaritmicFunctionF(linearFunctionF(x))
```

$$2 \log(2x + 1)$$

```
>compositionLinAndLogFuncF(E)
```

$$3.72398960812$$

Dimisalkan untuk konstanta  $j$

```
>$compositionLinAndLogFuncF(j)
```

$$2 \log(2j + 1)$$

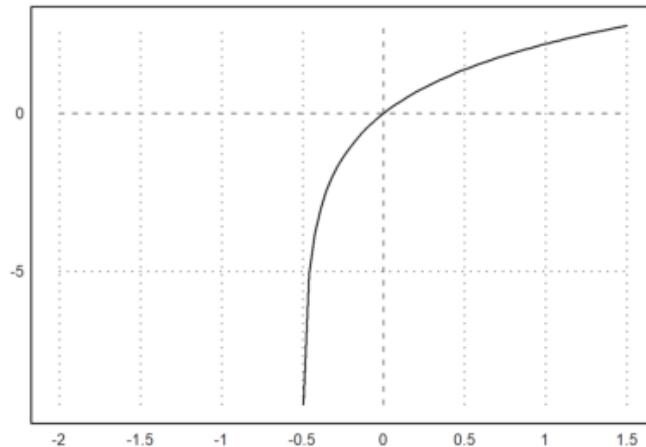
Apabila diberikan vektor  
maxima:  $[E^{(1/2)}, E, E^2]$

```
>compositionLinAndLogFuncF(E^(1/2):E:E^2)
```

$$[2.91604, 4.55125, 5.43871]$$

Membuat plot untuk fungsi komposisinya

```
>plot2d("compositionLinAndLogFuncF", -2, 1.5):
```



## 6. fungsi multivariabel

```
>function map multivariableFunctionF(x, y) &= y - sin(x)
```

$$y - \sin(x)$$

```
>multivariableFunction(Pi, 3)
```

3

Misalkan untuk  $y = 1$  dan  $x = k$

```
>$multivariableFunction(k, 1)
```

$$l - \sin k$$

Diberikan rentang dalam bentuk vektor untuk x  
maxima:  $[(1/2)*\pi, \pi, 2*\pi]$

```
>multivariableFunction((1/2)*Pi:Pi:2*Pi, 2)
```

[1, 3]

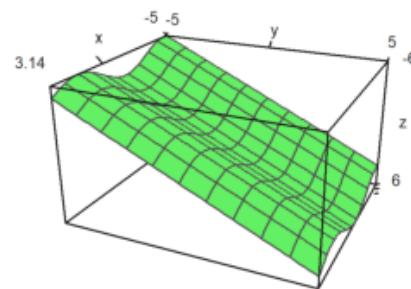
Apablia untuk rentang y diberikan vektor  
maxima: [2, 4]

```
>multivariableFunctionF(Pi, 2:4)
```

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

Dibuat plot untuk fungsinya

```
>plot3d("multivariableFunctionF", Pi, -5, -5, 5):
```



Dibuat fungsi komposisi dari fungsi multivariabel dengan fungsi linear

```
>function map compositionLinAndMultiFuncF(x, y) &= multivariableFunctionF(linearFunctionF(x), y)
```

$$y = \sin(2x + 1)$$

```
>compositionLinAndMultiFuncF(2*Pi - 1/2, 3)
```

3

Diberikan sebarang

$$x = m$$

$$y = n$$

```
>$compositionLinAndMultiFuncF(m, n)
```

$$n - \sin(2m + 1)$$

```
>compositionLinAndMultiFuncF((1/2)*Pi:Pi:2*Pi, 3)
```

[3.84147, 3.84147]

Misalkan diberikan  
maxima:  $x = \pi$

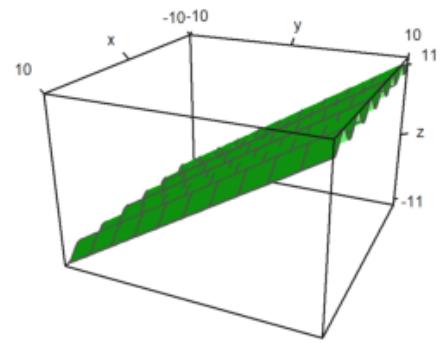
$$y \in \{-1, 0, 1\}$$

```
>compositionLinAndMultiFuncF(Pi, -1:1)
```

[-1.84147, -0.841471, 0.158529]

Membuat plot dari fungsi komposisi

```
>plot3d("compositionLinAndMultiFuncF", -10:10, 3, -10, 10, -10, 10):
```



## Menghitung Limit

---

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi "plus" atau "minus". Hasil limit dapat berupa nilai, "und" (tak definisi), "ind" (tak tentu namun terbatas), "infinity" (kompleks tak hingga).

Perhatikan beberapa contoh berikut. Perhatikan cara menampilkan perhitungan secara lengkap, tidak hanya menampilkan hasilnya saja.

```
>$showev('limit(sqrt(x^2-3*x)/(x+1),x,inf))
```

$$\lim_{x \rightarrow \infty} \frac{\sqrt{x^2 - 3x}}{x + 1} = 1$$

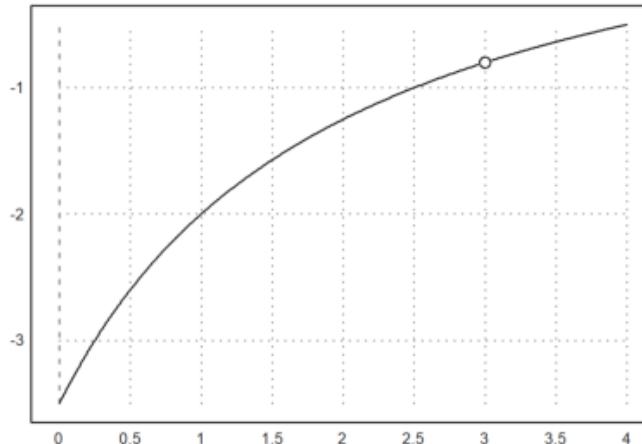
```
>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

$$-\frac{4}{5}$$

```
maxima: '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")
```

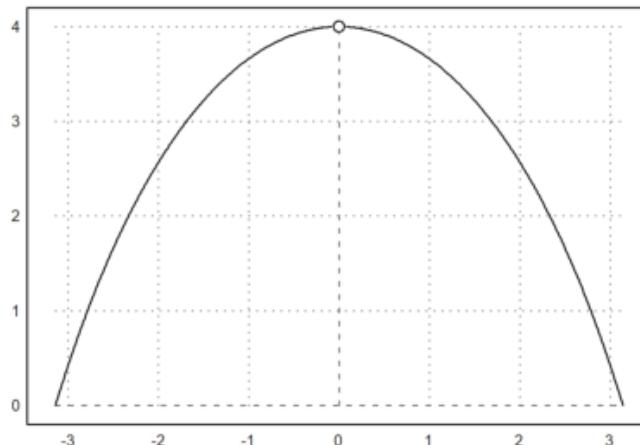


```
>$limit(2*x*sin(x)/(1-cos(x)),x,0)
```

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):
```



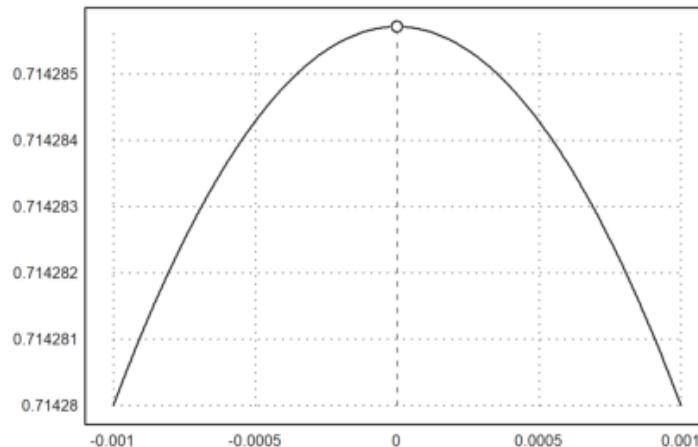
```
>$limit(cot(7*h)/cot(5*h),h,0)
```

$$\frac{5}{7}$$

```
maxima: showev('limit(cot(7*h)/cot(5*h),h,0))
```

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di  $x=0$ . Berikut adalah grafiknya.

```
>plot2d("cot(7*x)/cot(5*x)",-0.001,0.001); plot2d(0,5/7,>points,style="ow",>add):
```



```
>$showev('limit(((x/8)^(1/3)-1)/(x-8),x,8))
```

$$\lim_{x \rightarrow 8} \frac{\frac{x^{\frac{1}{3}}}{2} - 1}{x - 8} = \frac{1}{24}$$

Akan ditunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>function f(x) &= ((x/8)^(1/3)-1)/(x-8)
```

$$\frac{x^{1/3} - 1}{x - 8}$$

```
>plot2d("f", 6, 10); plot2d(8, f(8), >points, style="ow", >add):
```

```
Floating point error!
Try "trace errors" to inspect local variables after errors.
f:
useglobal; return (x^(1/3)/2-1)/(x-8)
Error in:
plot2d("f", 6, 10); plot2d(8, f(8), >points, style="ow", >add) ...
```

```
>$showev('limit(1/(2*x-1),x,0))
```

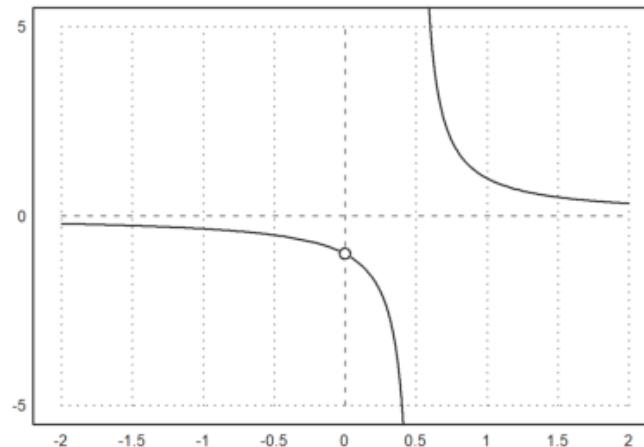
$$\lim_{x \rightarrow 0} \frac{1}{2x-1} = -1$$

Akan ditunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>function f(x) &= (1/(2*x-1))
```

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

```
>plot2d("f", -2, 2, -5, 5); plot2d(0, f(0), >points, style="ow", >add):
```



```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

$$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$$

Akan ditunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>function f(x) &= ((x^2-3*x-10)/(x-5))
```

$$\begin{array}{r} 2 \\ x - 3x - 10 \\ \hline x - 5 \end{array}$$

```
>plot2d("f", 3, 7); plot2d(5, f(5), >points, style="ow", >add):
```

```
Floating point error!
Try "trace errors" to inspect local variables after errors.
f:
useglobal; return (x^2-3*x-10)/(x-5)
Error in:
plot2d("f", 3, 7); plot2d(5, f(5), >points, style="ow", >add): ...
```

```
>$showev('limit(sqrt(x^2+x)-x,x,inf))
```

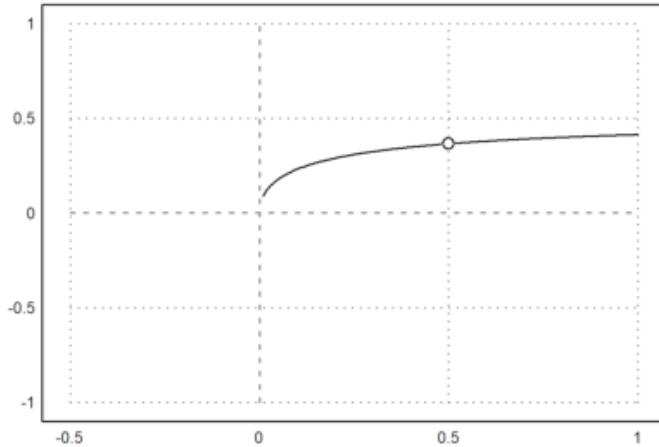
$$\lim_{x \rightarrow \infty} \sqrt{x^2 + x} - x = \frac{1}{2}$$

Akan ditunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>function f(x) &= sqrt(x^2+x)-x
```

$$\sqrt{x^2 + x} - x$$

```
>plot2d("f", -0.5, 1, -1, 1); plot2d(0.5, 0.366025, >points, style="ow", >add):
```



```
>$showev('limit(abs(x-1)/(x-1),x,1,minus))
```

$$\lim_{x \uparrow 1} \frac{|x-1|}{x-1} = -1$$

Akan dihitung limit di atas untuk  $x$  menuju 1 dari kanan.

Akan ditunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

Nilai limit diatas untuk  $x$  menuju 1 dari kanan

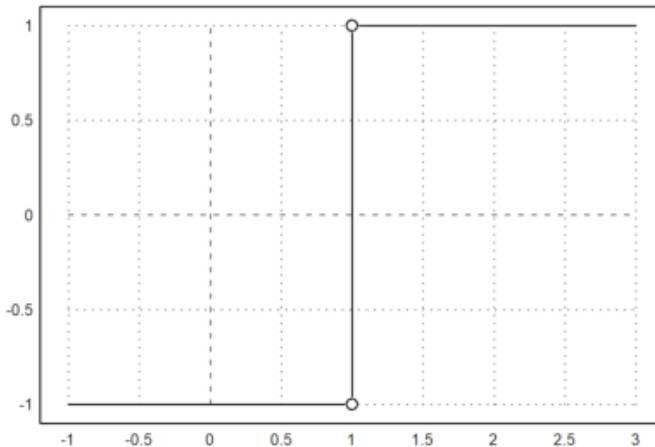
```
>$showev('limit(abs(x-1)/(x-1),x,1,plus))
```

$$\lim_{x \downarrow 1} \frac{|x - 1|}{x - 1} = 1$$

```
>function f(x) &= abs(x-1)/(x-1)
```

$$\frac{mabs(x - 1)}{x - 1}$$

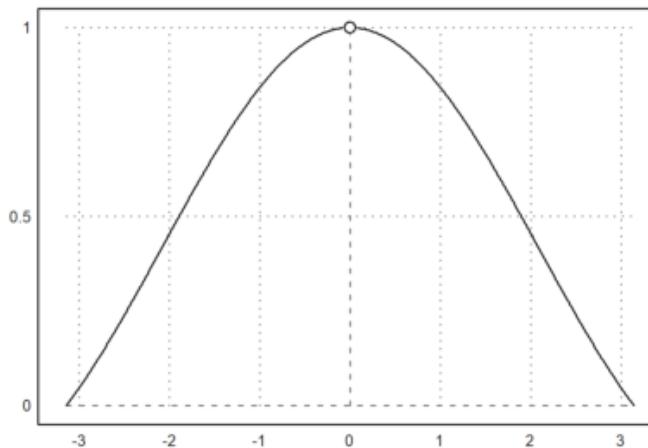
```
>plot2d("f", -1, 3); plot2d(1, -1, >points, style="ow", >add); plot2d(1, 1, >points, style="ow", >ad
```



```
>$showev('limit(sin(x)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

```
>plot2d("sin(x)/x",-pi,pi); plot2d(0,1,>points,style="ow",>add):
```



```
>$showev('limit(sin(x^3)/x,x,0))
```

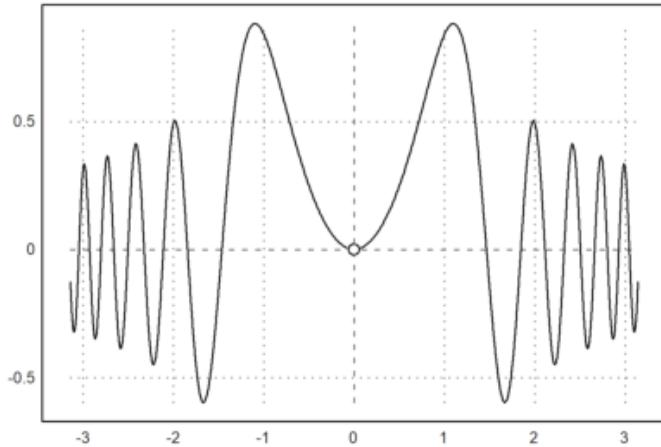
$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>function f(x) &= sin(x^3)/x
```

$$\frac{\sin(x^3)}{x}$$

```
>plot2d("f", -Pi, Pi); plot2d(0, 0, >points, >add, style="ow"):
```



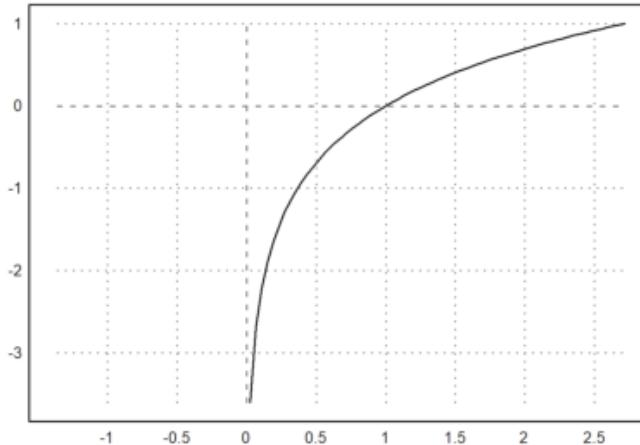
```
>$showev('limit(log(x), x, minf))
```

$$\lim_{x \rightarrow -\infty} \log x = \text{infinity}$$

```
>function f(x) &= log(x)
```

$\log(x)$

```
>plot2d("f", -(1/2)*E, E):
```



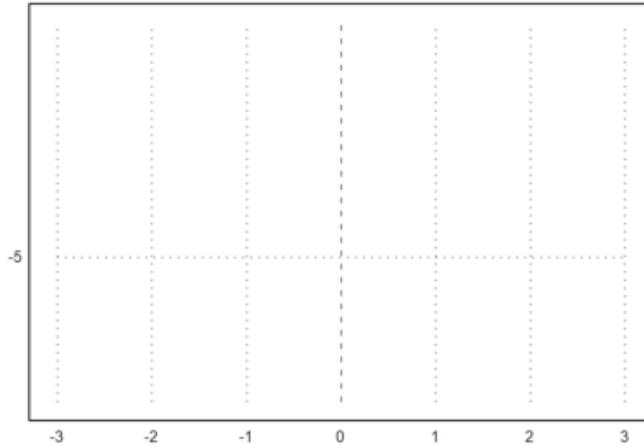
```
>$showev('limit((-2)^x,x, inf))
```

$$\lim_{x \rightarrow \infty} (-2)^x = \text{infinity}$$

```
>function f(x) &= (-2)^x
```

$$\begin{matrix} x \\ (-2) \end{matrix}$$

```
>plot2d("f", -3, 3):
```



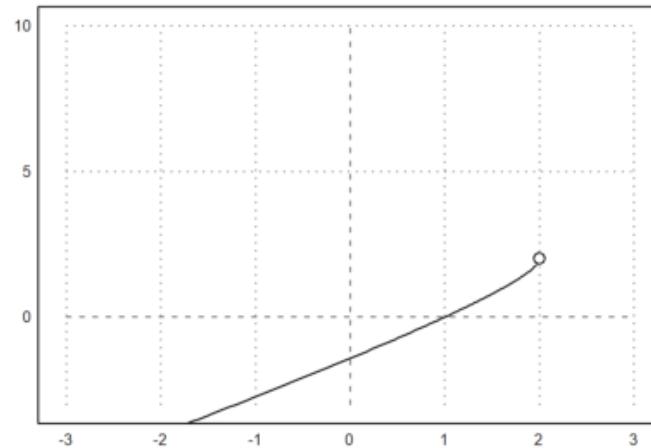
```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t \uparrow 2} t - \sqrt{2-t} = 2$$

```
>function f(t) &= t-sqrt(2-t)
```

$$t - \sqrt{2 - t}$$

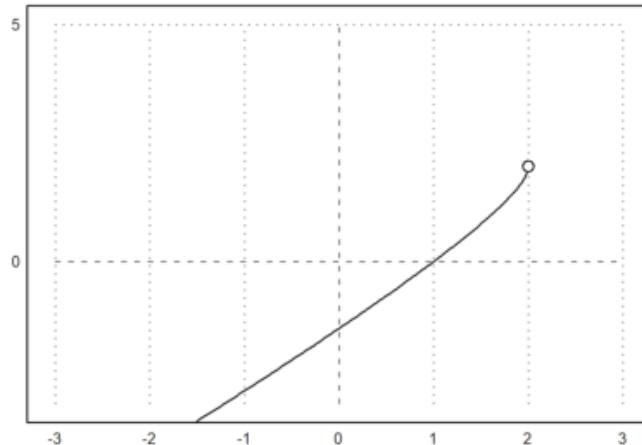
```
>plot2d("f", -3, 3, -3, 10); plot2d(2, f(2), >points, >add, style="ow"):
```



```
>$showev('limit(t-sqrt(2-t),t,2,plus))
```

$$\lim_{t \downarrow 2} t - \sqrt{2-t} = 2$$

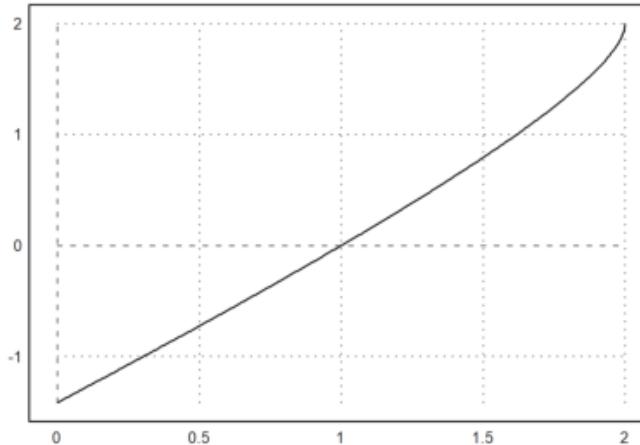
```
>plot2d("f", -3, 3, -3, 5); plot2d(2, f(2), >points, >add, style="ow"):
```



```
>$showev('limit(t-sqrt(2-t), t, 5, plus))
```

$$\lim_{t \downarrow 5} t - \sqrt{2-t} = 5 - \sqrt{3} i$$

```
>plot2d("x-sqrt(2-x)",0,2):
```



```
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
```

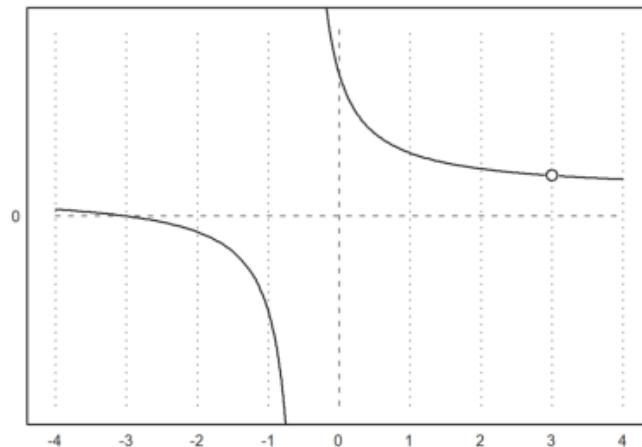
$$\lim_{x \rightarrow 3} \frac{x^2 - 9}{2x^2 - 5x - 3} = \frac{6}{7}$$

Akan ditunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>function f(x) &= (x^2-9)/(2*x^2-5*x-3)
```

$$\begin{array}{c} 2 \\ x - 9 \\ \hline 2 \\ 2 x - 5 x - 3 \end{array}$$

```
>plot2d("f", -4, 4, -4, 4); plot2d(3, 6/7, >points, >add, style="ow"):
```



```
>$showev('limit((1-cos(x))/x,x,0))
```

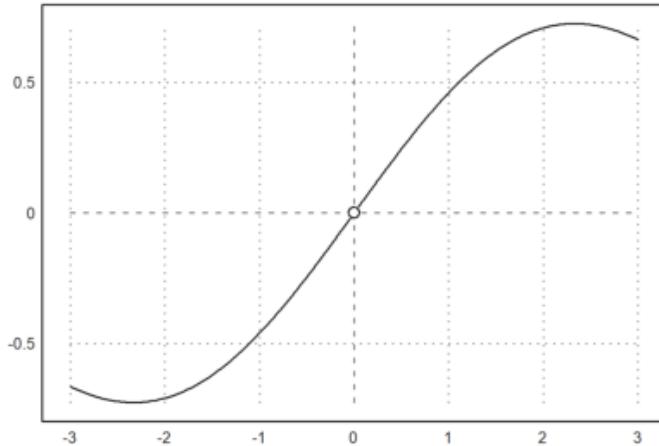
$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} = 0$$

Akan ditunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>function f(x) &= (1-cos(x))/x
```

$$\frac{1 - \cos(x)}{x}$$

```
>plot2d("f", -3, 3); plot2d(0, 0, >points, >add, style="ow"):
```



```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

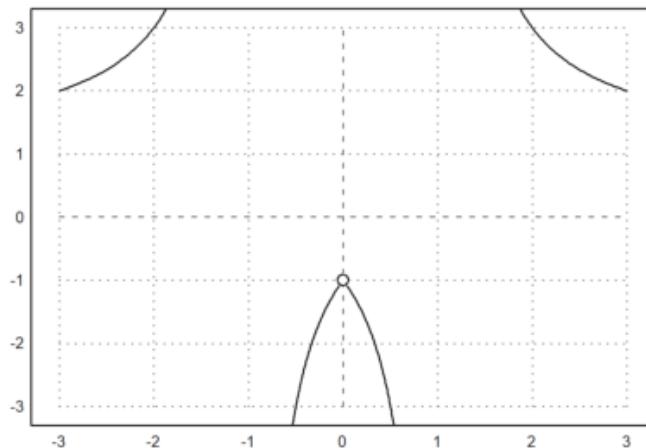
$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

Akan ditunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>function f(x) &= (x^2+abs(x))/(x^2-abs(x))
```

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

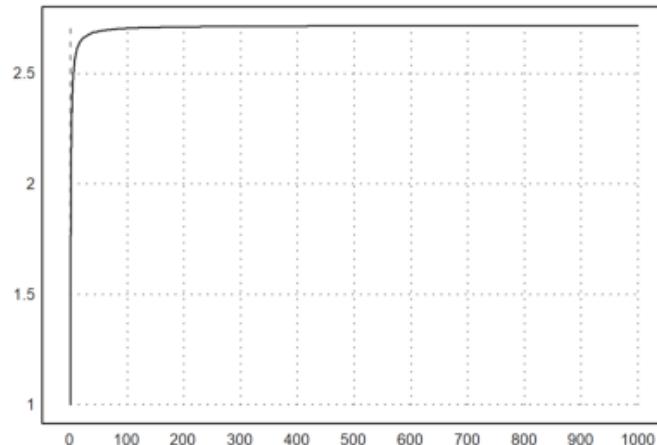
```
>plot2d("f", -3, 3, -3, 3); plot2d(0, -1, >points, >add, style="ow"):
```



```
>$showev('limit((1+1/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{1}{x} + 1 \right)^x = e$$

```
>plot2d("(1+1/x)^x",0,1000):
```



```
>$showev('limit((1+k/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{k}{x} + 1 \right)^x = e^k$$

```
>$showev('limit((1+x)^(1/x),x,0))
```

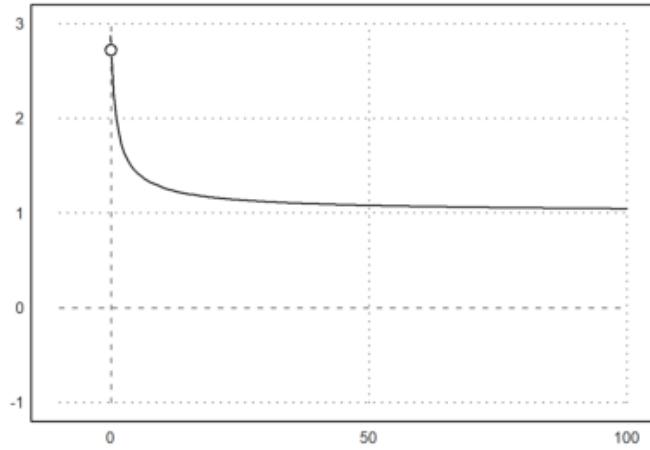
$$\lim_{x \rightarrow 0} (x + 1)^{\frac{1}{x}} = e$$

Akan ditunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

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

$$(x + 1)^{1/x}$$

```
>plot2d("f", -10, 100, -1, 3); plot2d(0, E, >points, >add, style="ow"):
```



```
>$showev('limit((x/(x+k))^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{x}{x+k} \right)^x = e^{-k}$$

```
>$showev('limit((E^x-E^2)/(x-2),x,2))
```

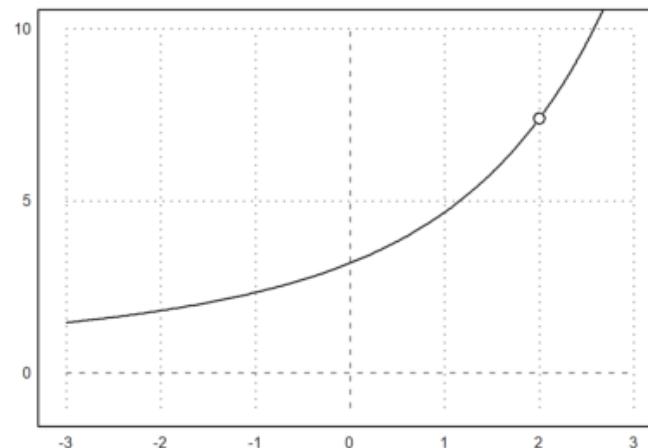
$$\lim_{x \rightarrow 2} \frac{e^x - e^2}{x - 2} = e^2$$

Akan ditunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>function f(x) &= (E^x-E^2)/(x-2)
```

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

```
>plot2d("f", -3, 3, -1, 10); plot2d(2, E^2, >points, >add, style="ow"):
```



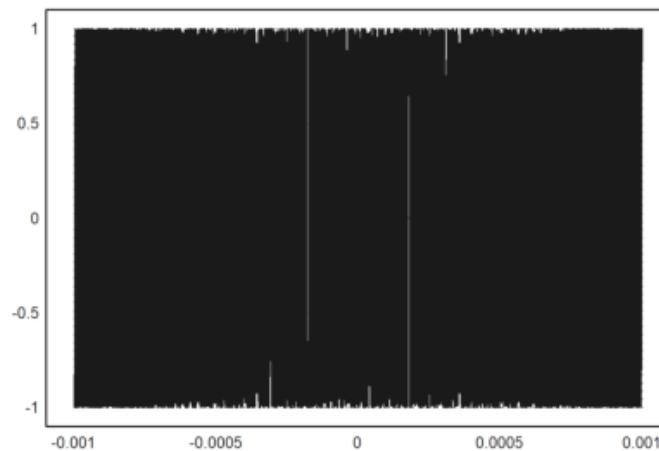
```
>$showev('limit(sin(1/x),x,0))
```

$$\lim_{x \rightarrow 0} \sin\left(\frac{1}{x}\right) = \text{ind}$$

```
>$showev('limit(sin(1/x),x,inf))
```

$$\lim_{x \rightarrow \infty} \sin\left(\frac{1}{x}\right) = 0$$

```
>plot2d("sin(1/x)",-0.001,0.001):
```



Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

### 1. Limit

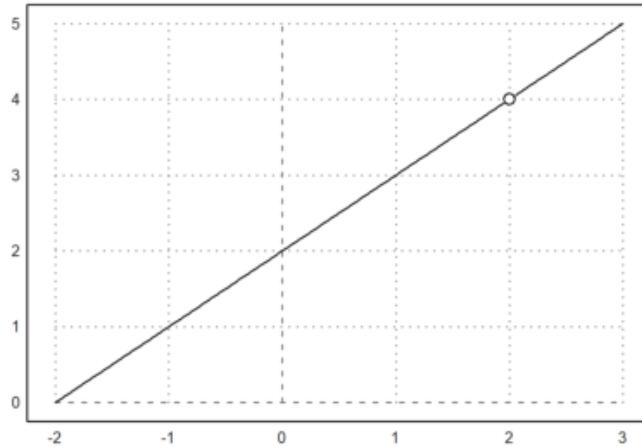
```
>$showev('limit((x^2-4)/(x-2),x,2))
```

$$\lim_{x \rightarrow 2} \frac{x^2 - 4}{x - 2} = 4$$

```
>function f(x) &= (x^2-4)/(x-2)
```

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

```
>plot2d("f", -2, 3); plot2d(2, 4, >points, >add, style="ow"):
```



## 2. Limit

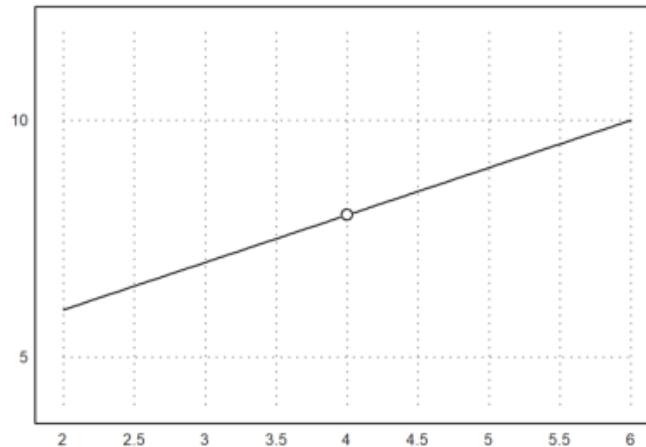
```
>$showev('limit((x^2-16)/(x-4),x,4))
```

$$\lim_{x \rightarrow 4} \frac{x^2 - 16}{x - 4} = 8$$

```
>function f(x) &= (x^2-16)/(x-4)
```

$$\begin{aligned} & 2 \\ & x - 16 \\ & \hline \\ & x - 4 \end{aligned}$$

```
>plot2d("f", 2, 6, 4, 12); plot2d(4, 8, >points, >add, style="ow"):
```



### 3. Limit

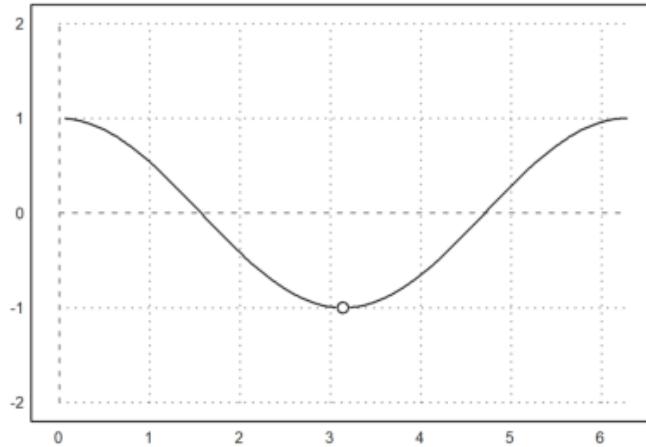
```
>$showev('limit((sin(x))/(tan(x)),x,pi))
```

$$\lim_{x \rightarrow \pi} \frac{\sin x}{\tan x} = -1$$

```
>function f(x) &= (sin(x))/(tan(x))
```

$$\frac{\sin(x)}{\tan(x)}$$

```
>plot2d("f", 0, 2*pi, -2, 2); plot2d(pi, f(pi), >points, >add, style="ow"):
```



#### 4. Limit

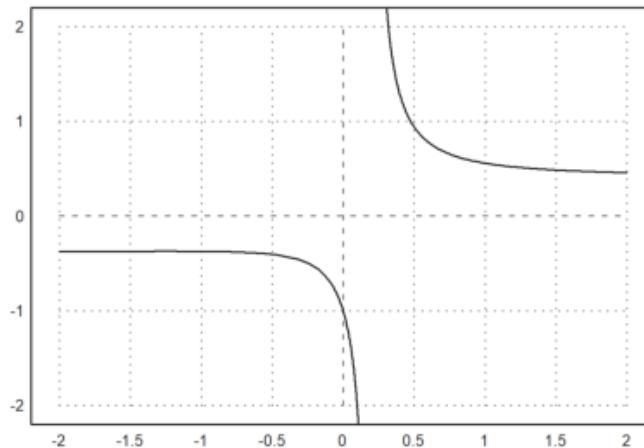
```
>$showev('limit((sqrt(4*x^2+1))/(5*x-1),x,minf))
```

$$\lim_{x \rightarrow -\infty} \frac{\sqrt{4x^2 + 1}}{5x - 1} = -\frac{2}{5}$$

```
>function f(x) &= (sqrt(4*x^2+1))/(5*x-1)
```

$$\frac{\sqrt{4x^2 + 1}}{5x - 1}$$

```
>plot2d("f", -2, 2, -2, 2):
```



## 5. Limit

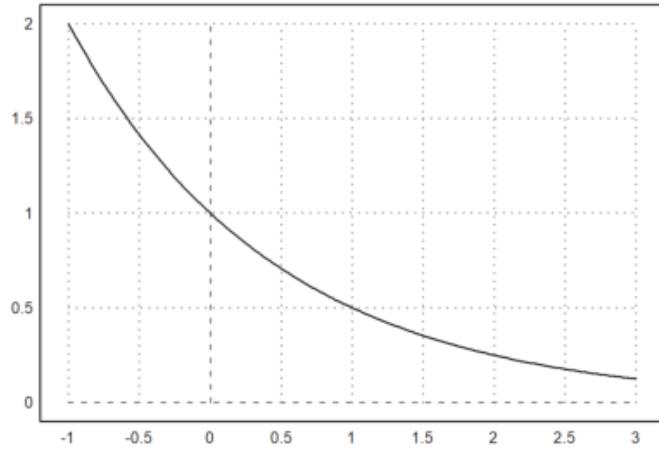
```
>$showev('limit(2^x,x,inf))
```

$$\lim_{x \rightarrow \infty} 2^x = \infty$$

```
>function f(x) &= 2^(-x)
```

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

```
>plot2d("f", -1, 3, 0, 2):
```



## Turunan Fungsi

---

Definisi turunan:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>$showev('limit(((x+h)^2-x^2)/h,h,0)) // turunan x^2
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2}{h} = 2x$$

```
>p &= expand((x+h)^2-x^2)|simplify; $p //pembilang dijabarkan dan disederhanakan
```

$$2hx + h^2$$

```
>q &=ratsimp(p/h); $q // ekspresi yang akan dihitung limitnya disederhanakan
```

$2x + h$

```
>$limit(q,h,0) // nilai limit sebagai turunan
```

$2x$

```
>$showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} = n x^{n-1}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunannya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan  $(x+h)^n$  dengan menggunakan teorema binomial.

```
>sym('x');
```

Syntax error in expression, or unfinished expression!

Error in:

```
sym('x'); ...  
^
```

```
>p &= expand(((x+h)^n-x^n)/h); $p
```

$$\frac{(x+h)^n}{h} - \frac{x^n}{h}$$

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

$$\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h} = \cos x$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini. Sebagai petunjuk, ekspansikan  $\sin(x+h)$  dengan menggunakan rumus jumlah dua sudut.

```
>&expand(sin(x+h))
```

$$\sin(x + h)$$

```
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
```

$$\lim_{h \rightarrow 0} \frac{\log(x+h) - \log x}{h} = \frac{1}{x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, gunakan sifat-sifat logaritma dan hasil limit pada bagian sebelumnya di atas.

```
>$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x
```

$$\lim_{h \rightarrow 0} \frac{\frac{1}{x+h} - \frac{1}{x}}{h} = -\frac{1}{x^2}$$

```
>$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x
```

```
Answering "Is x an integer?" with "integer"
Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is x an integer?
```

Use assume!

Error in:

```
$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x ...  
^
```

Maxima bermasalah dengan limit:

$$\lim_{h \rightarrow 0} \frac{e^{x+h} - e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{e^h - 1}{h} = 1$$

```
>$showev('factor(E^(x+h)-E^x))
```

$$factor(e^{x+h} - e^x) = (e^h - 1) e^x$$

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

$$\left( \lim_{h \rightarrow 0} \frac{e^h - 1}{h} \right) e^x = e^x$$

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

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

```
>$showev('limit(f(x),x,0))
```

$$\lim_{x \rightarrow 0} x^x = 1$$

Silakan Anda gambar kurva

$$y = x^x.$$

```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = \text{infinity}$$

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h}.$$

Dalam hal ini diperlukan asumsi nilai x.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x + h)^{x+h} - x^x}{h} = x^x (\log x + 1)$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

[x > 0]

```
>&forget(x<0)
```

[x < 0]

```
>&facts()
```

[]

```
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
```

$$\lim_{h \rightarrow 0} \frac{\arcsin(x+h) - \arcsin x}{h} = \frac{1}{\sqrt{1-x^2}}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

$$\lim_{h \rightarrow 0} \frac{\tan(x+h) - \tan x}{h} = \frac{1}{\cos^2 x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

sinh(x)

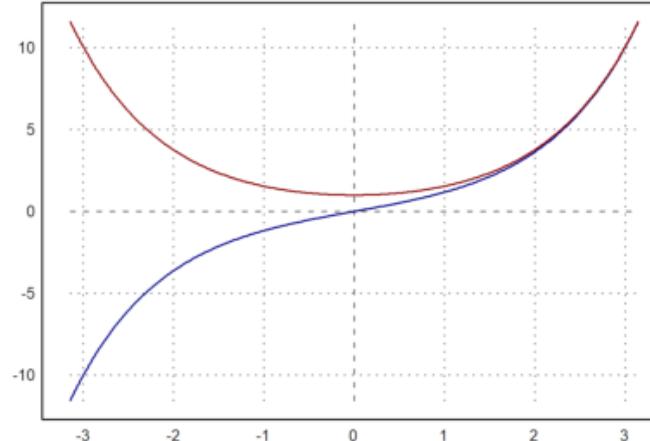
```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

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

Hasilnya adalah  $\cosh(x)$ , karena

$$\frac{e^x + e^{-x}}{2} = \cosh(x).$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```



```
>function f(x) &= sin(3*x^5+7)^2
```

$$\sin^2(3x^5 + 7)$$

```
>diff(f,3), diffc(f,3)
```

```
1198.32948904  
1198.72863721
```

Apakah perbedaan diff dan diffc?

```
>$showev('diff(f(x),x))
```

$$\frac{d}{dx} \sin^2(3x^5 + 7) = 30x^4 \cos(3x^5 + 7) \sin(3x^5 + 7)$$

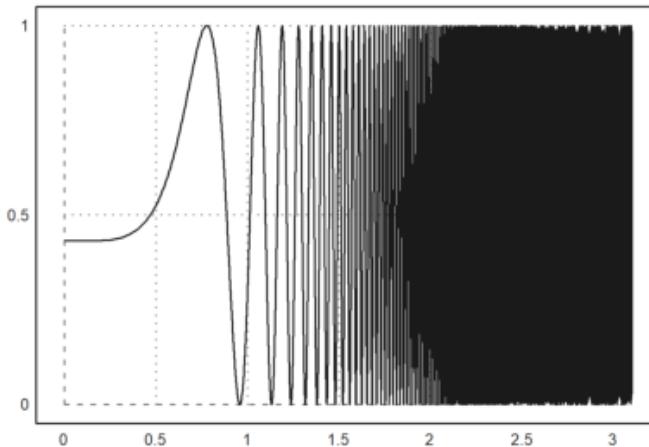
```
>$% with x=3
```

$$\%at\left(\frac{d}{dx} \sin^2(3x^5 + 7), x = 3\right) = 2430 \cos 736 \sin 736$$

```
>$float(%)
```

$$\%at \left( \frac{d^{1.0}}{dx^{1.0}} \sin^2 (3.0x^5 + 7.0), x = 3.0 \right) = 1198.728637211748$$

```
>plot2d(f,0,3.1):
```



```
>function f(x) &=5*cos(2*x)-2*x*sin(2*x) // mendefinisikan fungsi f
```

$$5 \cos(2x) - 2x \sin(2x)$$

```
>function df(x) &=diff(f(x),x) // fd(x) = f'(x)
```

$$- 12 \sin(2x) - 4x \cos(2x)$$

```
>$'f(1)=f(1), $float(f(1)), '$f(2)=f(2), $float(f(2)) // nilai f(1) dan f(2)
```

$$-0.2410081230863468$$

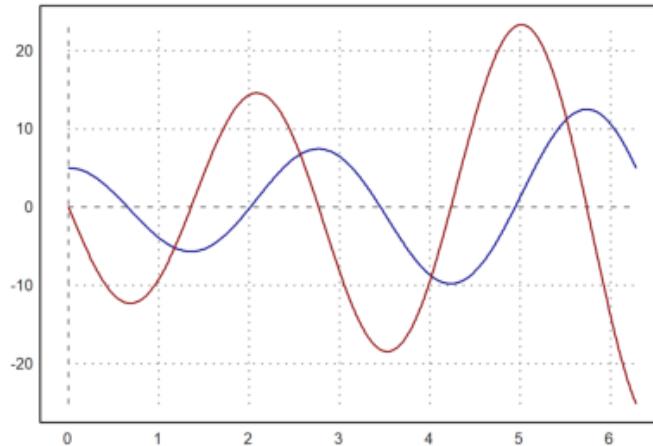
```
>xp=solve("df(x)",1,2,0) // solusi f'(x)=0 pada interval [1, 2]
```

$$1.35822987384$$

```
>df(xp), f(xp) // cek bahwa f'(xp)=0 dan nilai ekstrim di titik tersebut
```

```
0  
-5.67530133759
```

```
>plot2d(["f(x)","df(x)"],0,2*pi,color=[blue,red]): //grafik fungsi dan turunannya
```



Perhatikan titik-titik "puncak" grafik  $y=f(x)$  dan nilai turunan pada saat grafik fungsinya mencapai titik "puncak" tersebut.

**Latihan**

---

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), menggunakan perintah diff, dan secara manual (langkah demi langkah yang dihitung dengan Maxima) seperti contoh-contoh di atas. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

## Integral

---

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).

Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

$$\int_a^b f(x) \, dx = F(b) - F(a), \quad \text{dengan } F'(x) = f(x).$$

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak dapat di integralkan, kita tidak akan menggunakan metode ini).

```
>$showev('integrate(x^n,x))
```

Answering "Is n equal to -1?" with "no"

$$\int x^n \, dx = \frac{x^{n+1}}{n+1}$$

```
>$showev('integrate(1/(1+x),x))
```

$$\int \frac{1}{x+1} dx = \log(x+1)$$

```
>$showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2 + 1} dx = \arctan x$$

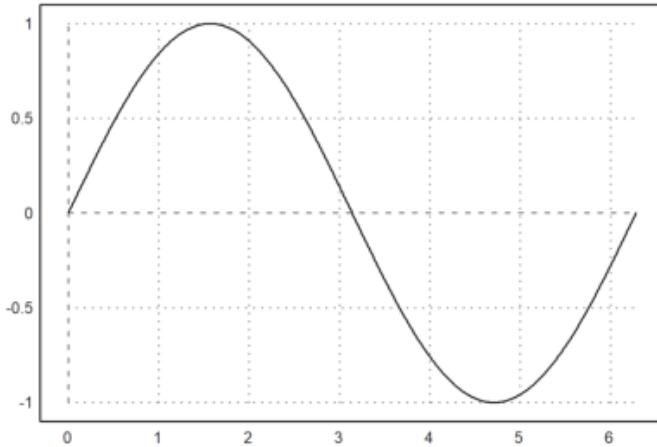
```
>$showev('integrate(1/sqrt(1-x^2),x))
```

$$\int \frac{1}{\sqrt{1-x^2}} dx = \arcsin x$$

```
>$showev('integrate(sin(x),x,0,pi))
```

$$\int_0^\pi \sin x dx = 2$$

```
>plot2d("sin(x)",0,2*pi):
```



```
>$showev('integrate(sin(x),x,a,b))
```

$$\int_a^b \sin x \, dx = \cos a - \cos b$$

```
>$showev('integrate(x^n,x,a,b))
```

Answering "Is n positive, negative or zero?" with "positive"

$$\int_a^b x^n \, dx = \frac{b^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
```

$$\int x^2 \sqrt{2x+1} dx = \frac{(2x+1)^{\frac{7}{2}}}{28} - \frac{(2x+1)^{\frac{5}{2}}}{10} + \frac{(2x+1)^{\frac{3}{2}}}{12}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
```

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{25^{\frac{5}{2}}}{21} - \frac{2}{105}$$

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

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{25^{\frac{7}{2}} - 2}{105}$$

```
>$showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^\pi - 1) \sin a + (e^\pi + 1) \cos a$$

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

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^\pi - 1) (\sin a - \cos a)$$

```
>function map f(x) &= E^(-x^2)
```

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

```
>$showev('integrate(f(x),x))
```

$$\int e^{-x^2} dx = \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

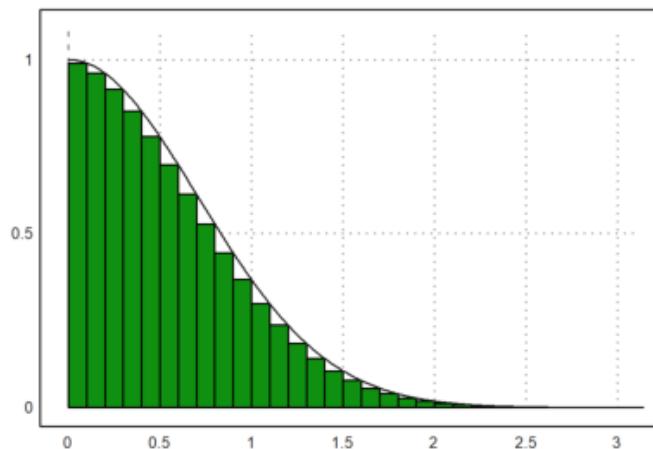
$$\operatorname{erf}(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}} dx.$$

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

maxima: 'integrate(f(x),x,0,pi)

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```



Integral tentu

maxima: 'integrate(f(x),x,0,pi)

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva  $y=f(x)$  tersebut. Langkah-langkahnya adalah sebagai berikut.

```
>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-nilai x  
>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)  
>/ jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!
```

Hasilnya adalah:

maxima: 'integrate(f(x),x,0,pi) = 0.1\*sum(fx[i],i,1,length(fx))

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai  $f(x)$  untuk  $x = 0.1, 0.2, 0.3, \dots, 3.2$ .

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

0.836219610253

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

Tunjukkan kebenaran hasil di atas!

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

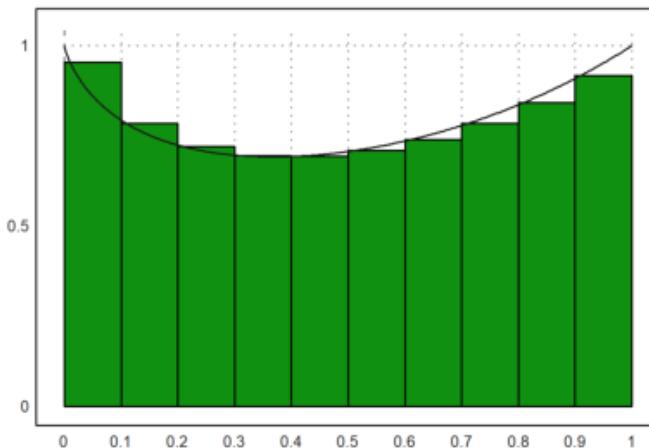
```
>function f(x) &= x^x
```

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

```
>$showev('integrate(f(x),x,0,1))
```

$$\int_0^1 x^x \, dx = \int_0^1 x^x \, dx$$

```
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```



Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
```

```
maxima: 'integrate(f(x),x,0,1) = 0.01*sum(fx[i],i,1,length(fx))
```

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
>function f(x) &= sin(3*x^5+7)^2
```

$$\sin^2(3x^5 + 7)$$

```
>integrate(f,0,1)
```

0.542581176074

```
>&showev('integrate(f(x),x,0,1))
```

$$\begin{aligned} & \int_0^{4/5} \sin^2(3x^5 + 7) dx = \frac{\gamma(-\frac{1}{5}) \sin(\frac{14}{5}\pi) \sin(\frac{6}{10}\pi)}{10^{1/5}} \\ & \quad \text{where } \gamma(z) = \frac{1}{2\pi i} \int_{-\infty}^{\infty} \frac{e^{zt}}{t^z} dt \end{aligned}$$

$$\begin{aligned}
& - \left( \left( 6 \operatorname{gamma\_incomplete}(-, 6) + 6 \operatorname{gamma\_incomplete}(-, -6) \right) \right. \\
& \quad \left. \frac{4}{5} \sin(14) + \left( 6 \operatorname{I gamma\_incomplete}(-, 6) \right) \right. \\
& \quad \left. \frac{4}{5} \operatorname{I gamma\_incomplete}(-, -6) \right) \cos(14) \sin(-60) / 120 \\
& - 6 \operatorname{I gamma\_incomplete}(-, -6) \cos(14) \sin(-60) / 120
\end{aligned}$$

```
>&float(%)
```

$$\begin{aligned}
& \frac{1.0}{\int [ \sin^2(x^5 + 7.0) dx ]} = \\
& \frac{0.0}{0.09820784258795788 - 0.008333333333333333} \\
& (0.3090169943749474 (0.1367372182078336 \\
& (4.192962712629476 \operatorname{I gamma\_incomplete}(0.2, 6.0) \\
& - 4.192962712629476 \operatorname{I gamma\_incomplete}(0.2, -6.0)) \\
& + 0.9906073556948704 (4.192962712629476 \operatorname{gamma\_incomplete}(0.2, 6.0) \\
& + 4.192962712629476 \operatorname{gamma\_incomplete}(0.2, -6.0))) - 60.0
\end{aligned}$$

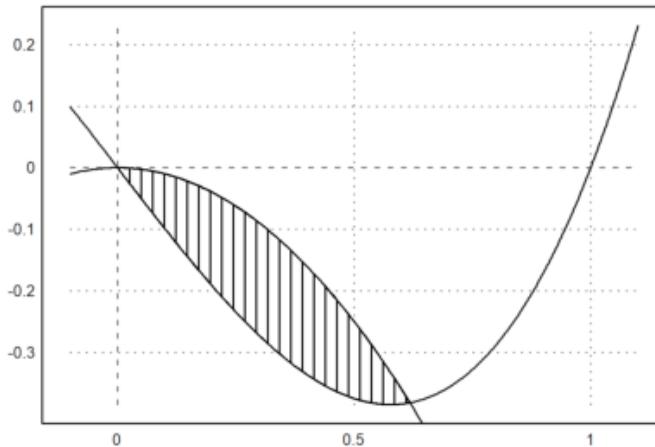
```
>$showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)
```

$$\int_0^1 x e^{-x} dx = 1 - 2 e^{-1}$$

## Aplikasi Integral Tentu

---

```
>plot2d("x^3-x",-0.1,1.1); plot2d("-x^2",>add); ...
>b=solve("x^3-x+x^2",0.5); x=linspace(0,b,200); xi=flipx(x); ...
>plot2d(x|xi,x^3-x|-xi^2,>filled,style="|",fillcolor=1,>add); // Plot daerah antara 2 kurva
```



```
>a=solve("x^3-x+x^2",0), b=solve("x^3-x+x^2",1) // absis titik-titik potong kedua kurva
```

0  
0.61803398875

```
>integrate("(-x^2)-(x^3-x)",a,b) // luas daerah yang diarsir
```

0.0758191713542

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```
>a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik potong kedua kurva secara eksak
```

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

```
>$showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara eksak
```

$$\int_0^{\frac{\sqrt{5}-1}{2}} -x^3 - x^2 + x \, dx = \frac{13 - 5^{\frac{3}{2}}}{24}$$

```
>$float(%)
```

$$\int_{0.0}^{0.6180339887498949} -1.0 x^3 - 1.0 x^2 + x \, dx = 0.07581917135421037$$

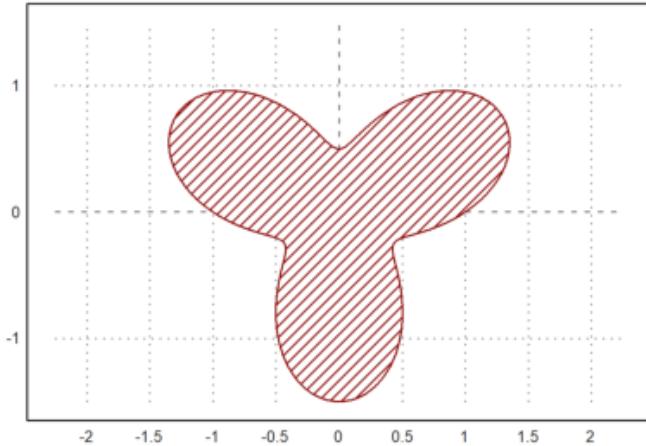
Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

$$\gamma(t) = (r(t) \cos(t), r(t) \sin(t))$$

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}, \quad 0 \leq t \leq 2\pi.$$

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
>plot2d(x,y,>filled,fillcolor=red,style="/" ,r=1.5); // Kita gambar kurvanya terlebih dahulu
```



```
>remvalue t;
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
```

$$r(t) = \frac{\sin(3t)}{2} + 1$$

```
>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
```

$$fx(t) = \cos t \left( \frac{\sin(3t)}{2} + 1 \right)$$

```
>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
```

$$fy(t) = \sin t \left( \frac{\sin(3t)}{2} + 1 \right)$$

```
>function ds(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t)
```

$$ds(t) = \frac{\sqrt{4 \cos(6t) + 4 \sin(3t) + 9}}{2}$$

```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
```

$$\frac{\int_0^{2\pi} \sqrt{4 \cos(6x) + 4 \sin(3x) + 9} dx}{2}$$

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara umerik dengan perintah EMT.

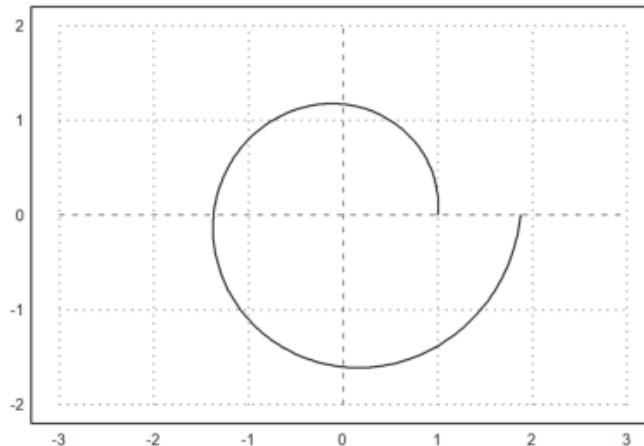
```
>integrate("ds(x)",0,2*pi)
```

9.0749467823

## Spiral Logaritmik

$$x = e^{ax} \cos x, \quad y = e^{ax} \sin x.$$

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi):
```



```
>&kill(a) // hapus expresi a
```

done

```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
```

$$fx(t) = e^{at} \cos t$$

```
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
```

$$fy(t) = e^{at} \sin t$$

```
>function df(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t)
```

$$df(t) = \sqrt{a^2 + 1} e^{at}$$

```
>S &=integrate(df(t),t,0,2*pi); $S // panjang kurva (spiral)
```

$$\sqrt{a^2 + 1} \left( \frac{e^{2\pi a}}{a} - \frac{1}{a} \right)$$

```
>S(a=0.1) // Panjang kurva untuk a=0.1
```

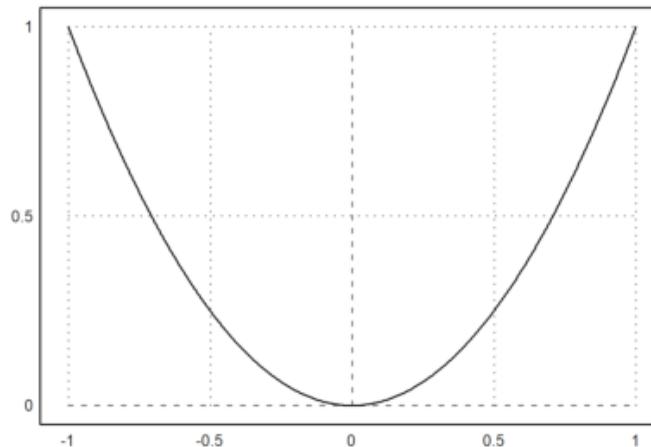
8.78817491636

Soal:

Tunjukkan bahwa keliling lingkaran dengan jari-jari r adalah  $K=2\pi r$ .

Berikut adalah contoh menghitung panjang parabola.

```
>plot2d("x^2",xmin=-1,xmax=1):
```



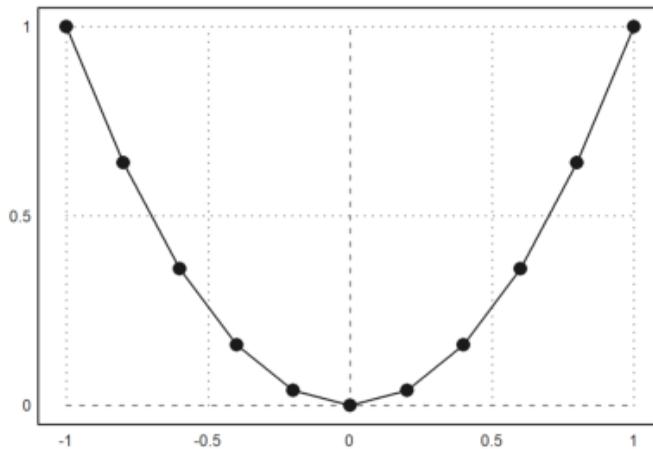
```
>remvalue x  
>$showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))
```

$$\int_{-1}^1 \sqrt{4x^2 + 1} dx = \frac{\operatorname{asinh} 2 + 2\sqrt{5}}{2}$$

```
>${float(%)}
```

$$\int_{-1.0}^{1.0} \sqrt{4.0 x^2 + 1.0} \, dx = 2.957885715089195$$

```
>x=-1:0.2:1; y=x^2; plot2d(x,y); ...
> plot2d(x,y,points=1,style="o#",add=1):
```



Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```
>i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])^2+(y[i+1]-y[i])^2))
```

2.95191957027

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

## Koordinat Kartesius

---

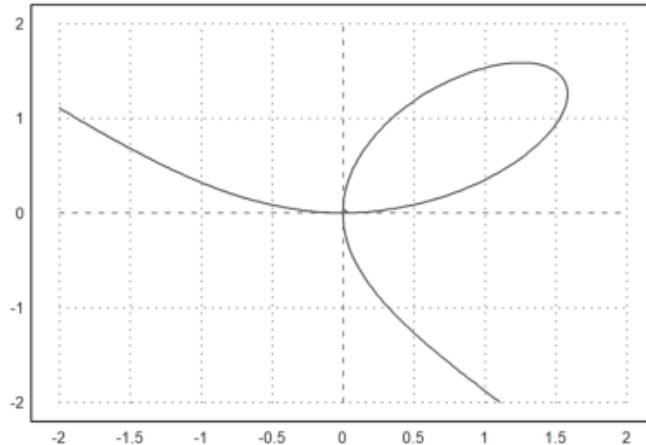
Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

$$x^3 + y^3 - 3xy = 0.$$

```
>remvalue z  
>z &= x^3+y^3-3*x*y; $z
```

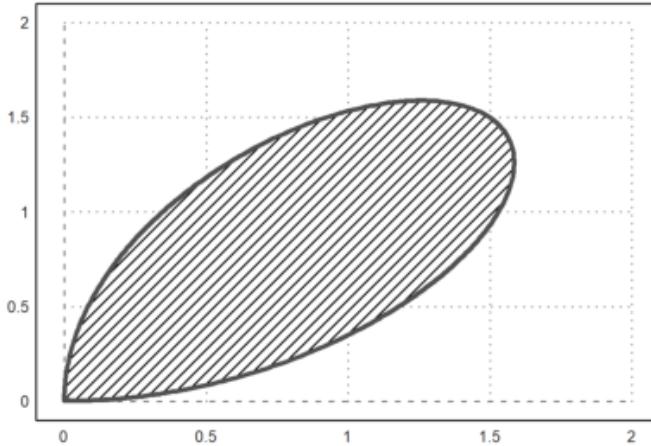
$$y^3 - 3xy + x^3$$

```
>plot2d(z,r=2,level=0,n=100):
```



Kita tertarik pada kurva di kuadran pertama.

```
>plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="/"):
```



Kita selesaikan persamaannya untuk x.

```
>$z with y=l*x, sol &= solve(%,x); $sol
```

$$\left[ x = \frac{3l}{l^3 + 1}, x = 0 \right]$$

$$\left[ x = \frac{3l}{l^3 + 1}, x = 0 \right]$$

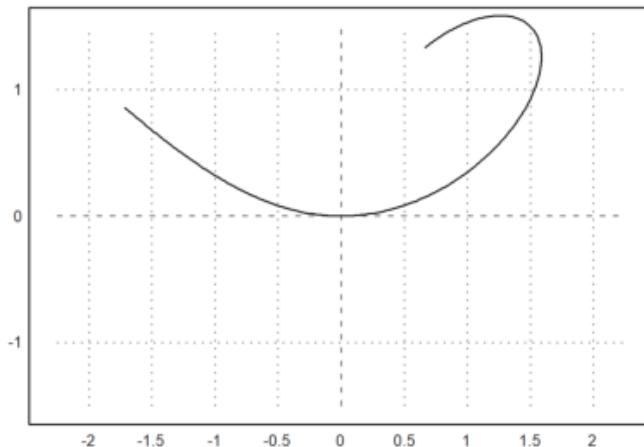
Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); \$'f(l)=f(l)
```

$$f(l) = \frac{3l}{l^3 + 1}$$

Fungsi tersebut juga dapat digunakan untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai y=l\*x, yakni x=f(l) dan y=l\*f(l).

```
>plot2d(&f(x),&x*f(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):
```



Elemen panjang kurva adalah:

$$ds = \sqrt{f'(l)^2 + (lf'(l) + f(l))^2}.$$

```
>function ds(l) &= ratsimp(sqrt(diff(f(l),l)^2+diff(l*f(l),l)^2)); $'ds(l)=ds(l)
```

$$ds(l) = \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}}$$

```
>$integrate(ds(l),l,0,1)
```

$$\int_0^1 \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}} dl$$

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral tersebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
>2*integrate("ds(x)",0,1)
```

4.91748872168

```
>2*romberg(&ds(x),0,1)// perintah Euler lain untuk menghitung nilai hampiran integral yang sama
```

4.91748872168

Perhitungan di datas dapat dilakukan untuk sebarang fungsi x dan y dengan mendefinisikan fungsi EMT, misalnya kita beri nama panjangkurva. Fungsi ini selalu memanggil Maxima untuk menurunkan fungsi yang diberikan.

```
>function panjangkurva(fx,fy,a,b) ...  
  
ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");  
return romberg(ds,a,b);  
endfunction
```

```
>panjangkurva("x","x^2",-1,1) // cek untuk menghitung panjang kurva parabola sebelumnya
```

2.95788571509

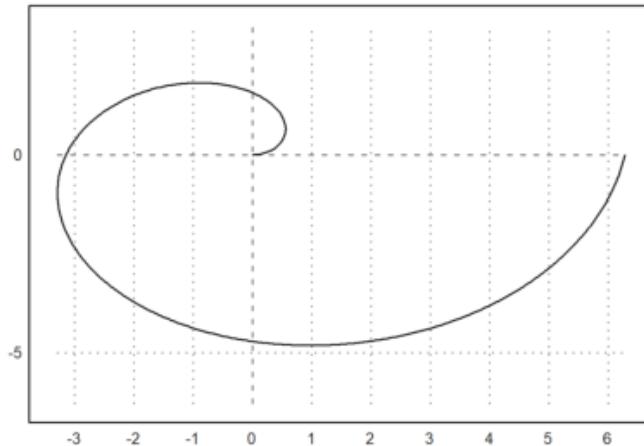
Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)"),mxm("x*f(x)"),0,1) // cek contoh terakhir, bandingkan hasilnya!
```

4.91748872168

Kita hitung panjang spiral Archimedes berikut ini dengan fungsi tersebut.

```
>plot2d("x*cos(x)","x*sin(x)",xmin=0,xmax=2*pi,square=1):
```



```
>panjangkurva("x*cos(x)","x*sin(x)",0,2*pi)
```

21.2562941482

Berikut kita definisikan fungsi yang sama namun dengan Maxima, untuk perhitungan eksak.

```
>&kill(ds,x,fx,fy)
```

done

```
>function ds(fx,fy) &=& sqrt(diff(fx,x)^2+diff(fy,x)^2)
```

$$\sqrt{\left(\frac{d}{dx}f_x(x)\right)^2 + \left(\frac{d}{dx}f_y(x)\right)^2}$$

```
>sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk menghitung panjang kurva terakhir di atas
```

$$\sqrt{(\cos x - x \sin x)^2 + (\sin x + x \cos x)^2}$$

```
>$sol | trigreduce | expand, $integrate(% ,x,0,2*pi), %()
```

$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$

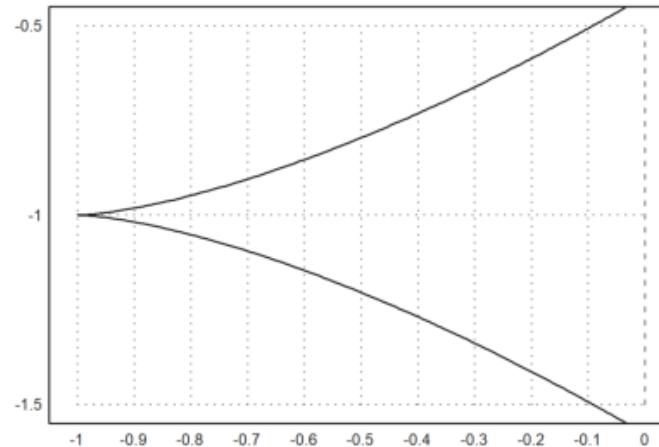
$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$

21.2562941482

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.

Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
>plot2d("3*x^2-1","3*x^3-1",xmin=-1/sqrt(3),xmax=1/sqrt(3),square=1):
```



```
>sol &= radcan(ds(3*x^2-1,3*x^3-1)); $sol
```

$$3x\sqrt{9x^2 + 4}$$

```
>$showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%); // panjang kurva di atas
```

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0x^2 + 4.0} dx = 2.337835372767141$$

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0x^2 + 4.0} dx = 2.337835372767141$$

## Sikloid

---

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar ke kanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah  $r$ . Posisi titik pusat lingkaran pada saat  $t$  adalah:

$$(rt, r).$$

Misalkan posisi titik pada lingkaran tersebut mula-mula  $(0,0)$  dan posisinya pada saat  $t$  adalah:

$$(r(t - \sin(t)), r(1 - \cos(t))).$$

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika  $t=0$ ,  $t=\pi/2$ ,  $t=r\pi$ .

```
>x &= r*(t-sin(t))
```

$$r (t - \sin(t))$$

```
>y &= r*(1-cos(t))
```

$$r (1 - \cos(t))$$

```
>remvalue t
```

Berikut kita gambar sikloid untuk  $r=1$ .

```
>ex &= x-sin(x); ey &= 1-cos(x); aspect(1);
>plot2d(ex,ey,xmin=0,xmax=4*pi,square=1); ...
> plot2d("2+cos(x)","1+sin(x)",xmin=0,xmax=2*pi,>add,color=blue); ...
> plot2d([2,ex(2)], [1,ey(2)],color=red,>add); ...
> plot2d(ex(2),ey(2),>points,>add,color=red); ...
```

```
> plot2d("2*pi+cos(x)","1+sin(x)",xmin=0,xmax=2*pi,>add,color=blue); ...
> plot2d([2*pi,ex(2*pi)],[1,ey(2*pi)],color=red,>add); ...
> plot2d(ex(2*pi),ey(2*pi),>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); $ds
>$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:

$2\pi$ .

---

## 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:

$$\gamma(s) = (x(s), y(s)),$$

dengan x dan y adalah fungsi riil yang diferensiabel, yang memenuhi:

$$\|\gamma'(s)\| = \sqrt{x'(s)^2 + y'(s)^2} = 1.$$

Ini berarti bahwa vektor singgung

$$\mathbf{T}(s) = (x'(s), y'(s))$$

memiliki norm 1 dan merupakan vektor singgung satuan.

Apabila kurvanya memiliki turunan kedua, artinya turunan kedua x dan y ada, maka  $T'(s)$  ada. Vektor ini merupakan normal kurva yang arahnya menuju pusat kurvatur, norm-nya merupakan nilai kurvatur (kelengkungan):

$$\begin{aligned}\mathbf{T}(s) &= \gamma'(s), \\ \mathbf{T}^2(s) &= 1 \text{ (konstanta)} \Rightarrow \mathbf{T}'(s) \cdot \mathbf{T}(s) = 0 \\ \kappa(s) &= \|\mathbf{T}'(s)\| = \|\gamma''(s)\| = \sqrt{x''(s)^2 + y''(s)^2}.\end{aligned}$$

Nilai

$$R(s) = \frac{1}{\kappa(s)}$$

disebut jari-jari (radius) kelengkungan kurva.

Bilangan riil

$$k(s) = \pm \kappa(s)$$

disebut nilai kelengkungan bertanda.

Contoh:

Akan ditentukan kurvatur lingkaran

$$x = r \cos t, \quad y = r \sin t.$$

```
>fx &= r*cos(t); fy &=r*sin(t);
>&assume(t>0,r>0); s &=integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s // elemen panjang kurva,
```

r t

```
>&kill(s); fx &= r*cos(s/r); fy &=r*sin(s/r); // definisi ulang persamaan parametrik terhadap s dengan r
>k &= trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur lingkaran dengan menggunakan r
```

$$\frac{1}{r}$$

Untuk representasi parametrik umum, misalkan

$$x = x(t), \quad y = y(t)$$

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

$$\begin{aligned}\kappa &= \frac{d\phi}{ds} = \frac{\frac{d\phi}{dt}}{\frac{ds}{dt}} \quad (\phi \text{ adalah sudut kemiringan garis singgung dan } s \text{ adalah panjang kurva}) \\ &= \frac{\frac{d\phi}{dt}}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}} = \frac{\frac{d\phi}{dt}}{\sqrt{x'(t)^2 + y'(t)^2}}.\end{aligned}$$

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

$$\sec^2 \phi \frac{d\phi}{dt} = \frac{d}{dt} (\tan \phi) = \frac{d}{dt} \left( \frac{dy}{dx} \right) = \frac{d}{dt} \left( \frac{dy/dt}{dx/dt} \right) = \frac{d}{dt} \left( \frac{y'(t)}{x'(t)} \right) = \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2}.$$

$$\begin{aligned}\frac{d\phi}{dt} &= \frac{1}{\sec^2 \phi} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1 + \tan^2 \phi} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1 + \left(\frac{y'(t)}{x'(t)}\right)^2} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2 + y'(t)^2}.\end{aligned}$$

Jadi, rumus kurvatur untuk kurva parametrik

$$x = x(t), \quad y = y(t)$$

adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}.$$

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kutub

$$x = r(\theta) \cos \theta, \quad y = r(\theta) \sin \theta,$$

maka rumus kurvaturnya adalah

$$\kappa(\theta) = \frac{r(\theta)^2 + 2r'(\theta)^2 - r(\theta)r''(\theta)}{(r'(\theta)^2 + r''(\theta)^2)^{3/2}}.$$

(Silakan Anda turunkan rumus tersebut!)

Contoh:

Lingkaran dengan pusat (0,0) dan jari-jari r dapat dinyatakan dengan persamaan parametrik

$$x = r \cos t, \quad y = r \sin t.$$

Nilai kelengkungan lingkaran tersebut adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}} = \frac{r^2}{r^3} = \frac{1}{r}.$$

Hasil cocok dengan definisi kurvatur suatu kelengkungan.

## Kurva

$$y = f(x)$$

dapat dinyatakan ke dalam persamaan parametrik

$$x = t, \quad y = f(t), \quad \text{dengan } x'(t) = 1, \quad x''(t) = 0,$$

sehingga kurvaturnya adalah

$$\kappa(t) = \frac{y''(t)}{(1 + y'(t)^2)^{3/2}}.$$

Contoh:

Akan ditentukan kurvatur parabola

$$y = ax^2 + bx + c.$$

```
>function f(x) &= a*x^2+b*x+c; $y=f(x)
```

$$r (1 - \cos t) = b r (t - \sin t) + a r^2 (t - \sin t)^2 + c$$

```
>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
```

$$r(1 - \cos t) = r(t - \sin t) + r^2(t - \sin t)^2 + 1$$

```
>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

$$(-1/2, 3/4), \quad 1/k(2) = 1/2,$$

sehingga pusat lingkaran oskulasi adalah  $(-1/2, 5/4)$ .

```
>plot2d(["f(x)", "k(x)"], -2, 1, color=[blue, red]); plot2d("1/k(x)", -1.5, 1, color=green, >add); ...
>plot2d("-1/2+1/k(-1/2)*cos(x)", "5/4+1/k(-1/2)*sin(x)", xmin=0, xmax=2pi, >add, color=blue):
```

```
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

$$F_x = \frac{\partial F}{\partial x}, \quad F_y = \frac{\partial F}{\partial y}, \quad F_{xy} = \frac{\partial}{\partial y} \left( \frac{\partial F}{\partial x} \right), \quad F_{xx} = \frac{\partial}{\partial x} \left( \frac{\partial F}{\partial x} \right), \quad F_{yy} = \frac{\partial}{\partial y} \left( \frac{\partial F}{\partial y} \right),$$

berlaku

$$F_x dx + F_y dy = 0 \text{ atau } \frac{dy}{dx} = -\frac{F_x}{F_y},$$

sehingga kurvaturnya adalah

$$\kappa = \frac{F_y^2 F_{xx} - 2F_x F_y F_{xy} + F_x^2 F_{yy}}{(F_x^2 + F_y^2)^{3/2}}.$$

(Silakan Anda turunkan sendiri!)

Contoh 1:

Parabola

$$y = ax^2 + bx + c$$

dapat dinyatakan ke dalam persamaan implisit

$$ax^2 + bx + c - y = 0.$$

```
>function F(x,y) &=a*x^2+b*x+c-y; $F(x,y)
```

$$b r (t - \sin t) + a r^2 (t - \sin t)^2 - r (1 - \cos t) + c$$

```
>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), Fy
```

```

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 parabol
```

$$k(r(t - \sin t)) = \frac{Fx^2 Fyy + Fxx Fy^2 - 2 Fx Fxy Fy}{(Fy^2 + Fx^2)^{\frac{3}{2}}}$$

Hasilnya sama dengan sebelumnya yang menggunakan persamaan parabola biasa.

## Latihan

- 
- Bukalah buku Kalkulus.
  - Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi).
  - Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah integral tentu dengan batas-batas yang menarik (Anda tentukan sendiri), seperti contoh-contoh tersebut.
  - Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat diintegralkan (cari sedikitnya 3 fungsi).
  - Gambar grafik fungsi dan daerah integrasinya pada sumbu koordinat yang sama.
  - Gunakan integral tentu untuk mencari luas daerah yang dibatasi oleh dua kurva yang berpotongan di dua titik. (Cari dan gambar kedua kurva dan arsir (warnai) daerah yang dibatasi oleh keduanya.)
  - Gunakan integral tentu untuk menghitung volume benda putar kurva  $y=f(x)$  yang diputar mengelilingi sumbu x dari  $x=a$  sampai  $x=b$ , yakni

$$V = \int_a^b \pi(f(x)^2) dx.$$

(Pilih fungsinya dan gambar kurva dan benda putar yang dihasilkan. Anda dapat mencari contoh-contoh bagaimana cara menggambar benda hasil perputaran suatu kurva.)

- Gunakan integral tentu untuk menghitung panjang kurva  $y=f(x)$  dari  $x=a$  sampai  $x=b$  dengan menggunakan rumus:

$$S = \int_a^b \sqrt{1 + (f'(x))^2} dx.$$

(Pilih fungsi dan gambar kurvanya.)

- Apabila fungsi dinyatakan dalam koordinat kutub  $x=f(r,t)$ ,  $y=g(r,t)$ ,  $r=h(t)$ ,  $x=a$  bersesuaian dengan  $t=t_0$  dan  $x=b$  bersesuaian dengan  $t=t_1$ , maka rumus di atas akan menjadi:

$$S = \int_{t_0}^{t_1} \sqrt{x'(t)^2 + y'(t)^2} dt.$$

- Pilih beberapa kurva menarik (selain lingkaran dan parabola) dari buku kalkulus. Nyatakan setiap kurva tersebut dalam bentuk:
  - a. koordinat Kartesius (persamaan  $y=f(x)$ )
  - b. koordinat kutub ( $r=r(\theta)$ )
  - c. persamaan parametrik  $x=x(t)$ ,  $y=y(t)$
  - d. persamaan implisit  $F(x,y)=0$
- Tentukan kurvatur masing-masing kurva dengan menggunakan keempat representasi tersebut (hasilnya harus sama).
- Gambarlah kurva asli, kurva kurvatur, kurva jari-jari lingkaran oskulasi, dan salah satu lingkaran oskulasinya.

**Barisan dan Deret**

---

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan (menggunakan titik dua ":");
- menggunakan perintah "sequence" dan rumus barisan (suku ke -n);
- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create\_list" atau "makelist" untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana
```

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

>1:2:30

```
[1,  3,  5,  7,  9,  11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
```

## Iterasi dan Barisan

---

EMT menyediakan fungsi iterate("g(x)", x0, n) untuk melakukan iterasi

$$x_{k+1} = g(x_k), \quad x_0 = x_0, \quad k = 1, 2, 3, \dots, n.$$

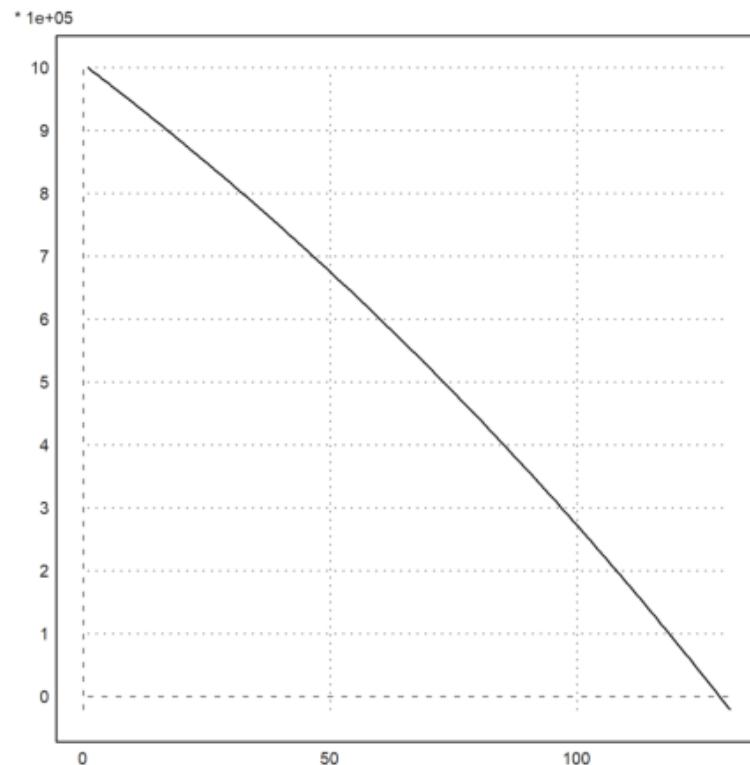
Berikut ini disajikan contoh-contoh penggunaan iterasi dan rekursi dengan EMT. Contoh pertama menunjukkan pertumbuhan dari nilai awal 1000 dengan laju pertambahan 5%, selama 10 periode.

```
>q=1.05; iterate("x*q",1000,n=10)'
```

```
1000  
1050  
1102.5  
1157.63  
1215.51  
1276.28  
1340.1  
1407.1  
1477.46  
1551.33  
1628.89
```

Contoh berikutnya memperlihatkan bahaya menabung di bank pada masa sekarang! Dengan bunga tabungan sebesar 6% per tahun atau 0.5% per bulan dipotong pajak 20%, dan biaya administrasi 10000 per bulan, tabungan sebesar 1 juta tanpa diambil selama sekitar 10 tahunan akan habis diambil oleh bank!

```
>r=0.005; plot2d(iterate("(1+0.8*r)*x-10000",1000000,n=130)):
```



Silakan Anda coba-coba, dengan tabungan minimal berapa agar tidak akan habis diambil oleh bank dengan ketentuan bunga dan biaya administrasi seperti di atas.

Berikut adalah perhitungan minimal tabungan agar aman di bank dengan bunga sebesar  $r$  dan biaya administrasi  $a$ , pajak bunga 20%.

```
>$solve(0.8*r*A-a,A), $% with [r=0.005, a=10]
```

$$[A = 2500.0]$$

Berikut didefinisikan fungsi untuk menghitung saldo tabungan, kemudian dilakukan iterasi.

```
>function saldo(x,r,a) := round((1+0.8*r)*x-a,2);
>iterate({{"saldo",0.005,10}},1000,n=6)
```

```
[1000, 994, 987.98, 981.93, 975.86, 969.76, 963.64]
```

```
>iterate({{"saldo",0.005,10}},2000,n=6)
```

```
[2000, 1998, 1995.99, 1993.97, 1991.95, 1989.92, 1987.88]
```

```
>iterate({{"saldo",0.005,10}},2500,n=6)
```

```
[2500, 2500, 2500, 2500, 2500, 2500, 2500]
```

Tabungan senilai 2,5 juta akan aman dan tidak akan berubah nilai (jika tidak ada penarikan), sedangkan jika tabungan awal kurang dari 2,5 juta, lama kelamaan akan berkurang meskipun tidak pernah dilakukan penarikan uang tabungan.

```
>iterate({{"saldo",0.005,10}},3000,n=6)
```

```
[3000, 3002, 3004.01, 3006.03, 3008.05, 3010.08, 3012.12]
```

Tabungan yang lebih dari 2,5 juta baru akan bertambah jika tidak ada penarikan.

Untuk barisan yang lebih kompleks dapat digunakan fungsi "sequence()". Fungsi ini menghitung nilai-nilai  $x[n]$  dari semua nilai sebelumnya,  $x[1], \dots, x[n-1]$  yang diketahui.

Berikut adalah contoh barisan Fibonacci.

$$x_n = x_{n-1} + x_{n-2}, \quad x_1 = 1, \quad x_2 = 1$$

```
>sequence("x[n-1]+x[n-2]",[1,1],15)
```

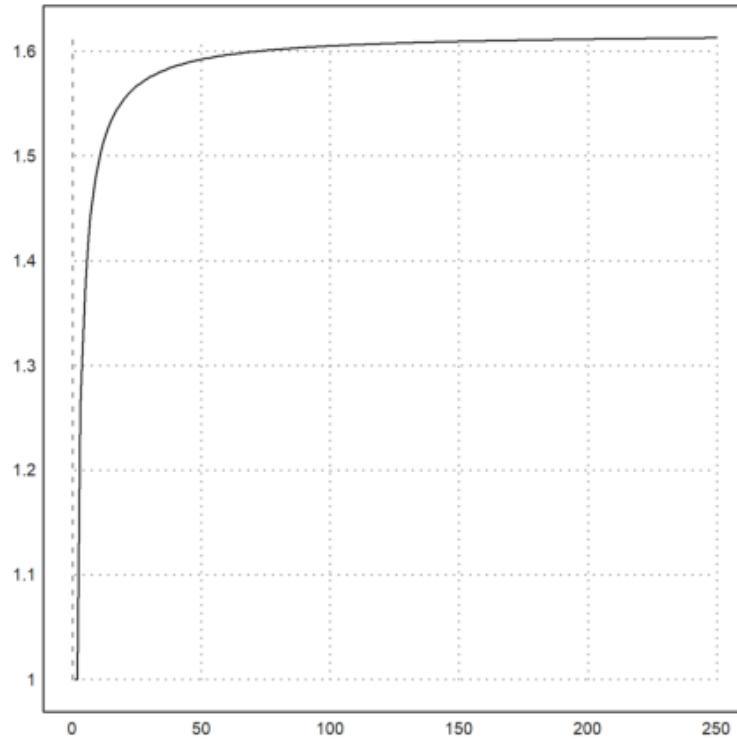
```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]
```

Barisan Fibonacci memiliki banyak sifat menarik, salah satunya adalah akar pangkat ke-n suku ke-n akan konvergen ke pecahan emas:

```
>$(1+sqrt(5))/2=float((1+sqrt(5))/2)
```

$$\frac{\sqrt{5} + 1}{2} = 1.618033988749895$$

```
>plot2d(sequence("x[n-1]+x[n-2]", [1,1], 250)^(1/(1:250))):
```



Barisan yang sama juga dapat dihasilkan dengan menggunakan loop.

```
>x=ones(500); for k=3 to 500; x[k]=x[k-1]+x[k-2]; end;
```

Rekursi dapat dilakukan dengan menggunakan rumus yang tergantung pada semua elemen sebelumnya. Pada contoh berikut, elemen ke-n merupakan jumlah (n-1) elemen sebelumnya, dimulai dengan 1 (elemen ke-1). Jelas, nilai elemen ke-n adalah  $2^n$ , untuk n=2, 4, 5, ....

```
>sequence("sum(x)",1,10)
```

```
[1, 1, 2, 4, 8, 16, 32, 64, 128, 256]
```

Selain menggunakan ekspresi dalam x dan n, kita juga dapat menggunakan fungsi.

Pada contoh berikut, digunakan iterasi

$$x_n = A \cdot x_{n-1},$$

dengan A suatu matriks 2x2, dan setiap  $x[n]$  merupakan matriks/vektor 2x1.

```
>A=[1,1;1,2]; function suku(x,n) := A.x[,n-1]
>sequence("suku",[1;1],6)
```

Real 2 x 6 matrix

1	2	5	13	...
1	3	8	21	...

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "matrix-power()". Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak  $\log_2(n)$ .

$$x_n = A \cdot x_{n-1} = A^2 \cdot x_{n-2} = A^3 \cdot x_{n-3} = \dots = A^{n-1} \cdot x_1.$$

```
>sequence("matrixpower(A,n).[1;1]",1,6)
```

Real 2 x 6 matrix

1	5	13	34	...
1	8	21	55	...

## Spiral Theodorus

---

image: Spiral\_of\_Theodorus.png

Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

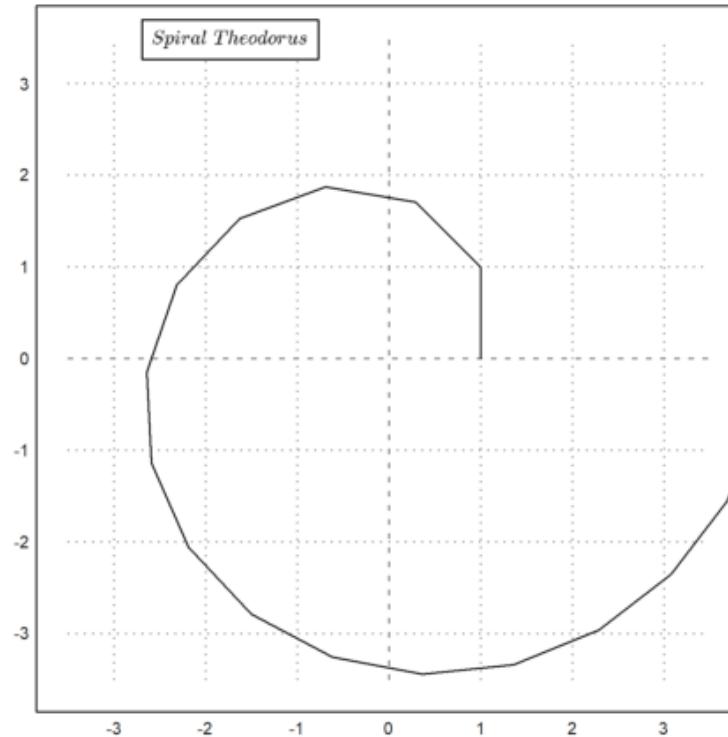
$$x_n = \left(1 + \frac{i}{\sqrt{n-1}}\right) x_{n-1}, \quad x_1 = 1,$$

yang menghasilkan barisan bilangan kompleks.

```
>function g(n) := 1+I/sqrt(n)
```

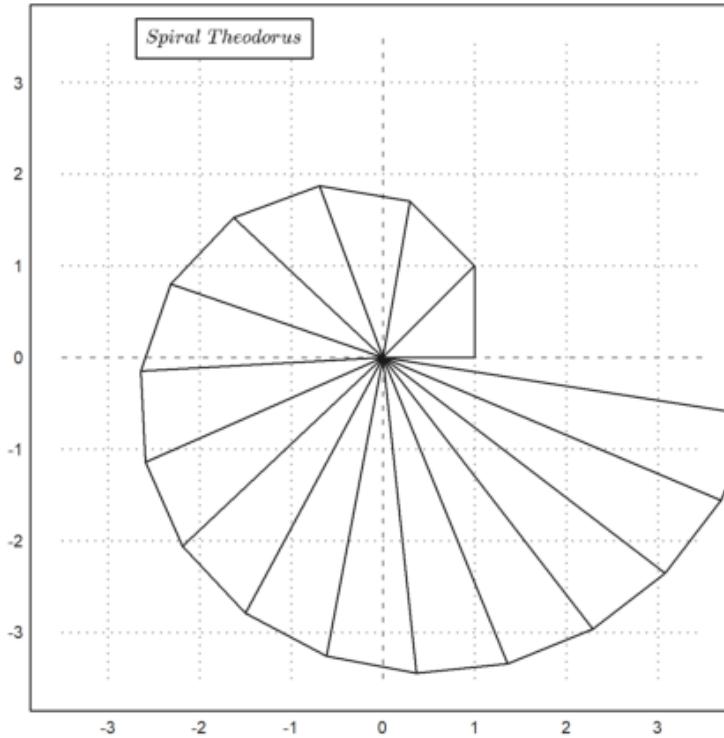
Rekursinya dapat dijalankan sebanyak 17 untuk menghasilkan barisan 17 bilangan kompleks, kemudian digambar bilangan-bilangan kompleksnya.

```
>x=sequence("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox(latex("Spiral\\ Theodorus"),0.4):
```



Selanjutnya dihubungkan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end:
```



>

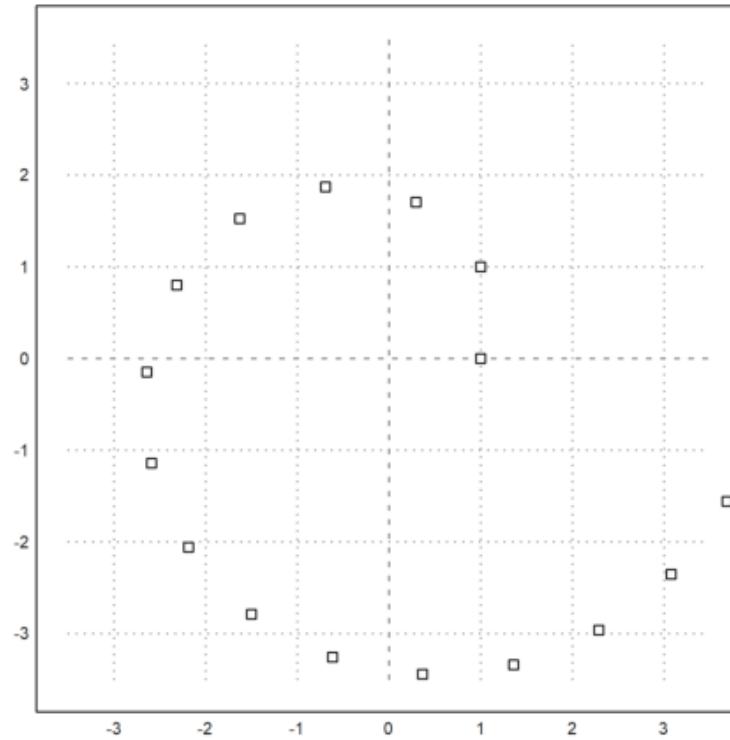
Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memerlukan indeks dan bilangan kompleks. Dalam hal ini digunakan vektor kolom pada bidang.

```
>function gstep (v) ...
```

```
w=[-v[2];v[1]];
return v+w/norm(w);
endfunction
```

Jika dilakukan iterasi 16 kali dimulai dari [1;0] akan didapatkan matriks yang memuat vektor-vektor dari setiap iterasi.

```
>x=iterate("gstep",[1;0],16); plot2d(x[1],x[2],r=3.5,>points):
```



## Kekonvergenan

---

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```
>iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1.
```

0.739085133216

Iterasi tersebut konvergen ke penyelesaian persamaan

$$x = \cos(x).$$

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.

```
>hasil := iterate("cos(x)",~1,2~) //iterasi x(n+1)=cos(x(n)), dengan interval awal (1, 2)
```

~0.739085133211,0.7390851332133~

Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan  $x=\cos(x)$ .

```
>h=expand(hasil,100), cos(h) << h
```

```
~0.73908513309,0.73908513333~  
1
```

Iterasi juga dapat digunakan pada fungsi yang didefinisikan.

```
>function f(x) := (x+2/x)/2
```

Iterasi  $x(n+1)=f(x(n))$  akan konvergen ke akar kuadrat 2.

```
>iterate("f",2), sqrt(2)
```

```
1.41421356237  
1.41421356237
```

Jika pada perintah iterate diberikan tambahan parameter n, maka hasil iterasinya akan ditampilkan mulai dari iterasi pertama sampai ke-n.

```
>iterate("f",2,5)
```

```
[2, 1.5, 1.41667, 1.41422, 1.41421, 1.41421]
```

Untuk iterasi ini tidak dapat dilakukan terhadap interval.

```
>niterate("f",~1,2~,5)
```

```
[ ~1,2~, ~1,2~, ~1,2~, ~1,2~, ~1,2~, ~1,2~ ]
```

Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya adalah perhitungan dengan interval bersifat terlalu longgar. Untuk meingkatkan perhitungan pada ekspresi dapat digunakan pembagian intervalnya, menggunakan fungsi ieval().

```
>function s(x) := ieval("(x+2/x)/2",x,10)
```

Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal, dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang memuat akar persamaan:

$$x = \frac{1}{2} \left( x + \frac{2}{x} \right).$$

Satu-satunya solusi adalah

$$x = \sqrt{2}.$$

```
>iterate("s",~1,2~)
```

```
~1.41421356236,1.41421356239~
```

Fungsi "iterate()" juga dapat bekerja pada vektor. Berikut adalah contoh fungsi vektor, yang menghasilkan rata-rata aritmetika dan rata-rata geometri.

$$(a_{n+1}, b_{n+1}) = \left( \frac{a_n + b_n}{2}, \sqrt{a_n b_n} \right)$$

Iterasi ke-n disimpan pada vektor kolom x[n].

```
>function g(x) := [(x[1]+x[2])/2,sqrt(x[1]*x[2])]
```

Iterasi dengan menggunakan fungsi tersebut akan konvergen ke rata-rata aritmetika dan geometri dari nilai-nilai awal.

```
>iterate("g", [1;5])
```

```
2.60401  
2.60401
```

Hasil tersebut konvergen agak cepat, seperti kita cek sebagai berikut.

```
>iterate("g", [1;5], 4)
```

1	3	2.61803	2.60403	2.60401
5	2.23607	2.59002	2.60399	2.60401

Iterasi pada interval dapat dilakukan dan stabil, namun tidak menunjukkan bahwa limitnya pada batas-batas yang dihitung.

```
>iterate("g", [~1~;~5~], 4)
```

Interval 2 x 5 matrix

```
~0.99999999999999778, 1.00000000000000022~ ...  
~4.999999999999911, 5.0000000000000089~ ...
```

Iterasi berikut konvergen sangat lambat.

$$x_{n+1} = \sqrt{x_n}.$$

```
>iterate("sqrt(x)",2,10)
```

```
[2, 1.41421, 1.18921, 1.09051, 1.04427, 1.0219, 1.01089,
1.00543, 1.00271, 1.00135, 1.00068]
```

Kekonvergenan iterasi tersebut dapat dipercepat dengan percepatan Steffenson:

```
>steffenson("sqrt(x)",2,10)
```

```
[1.04888, 1.00028, 1, 1]
```

## Iterasi menggunakan Loop yang ditulis Langsung

Berikut adalah beberapa contoh penggunaan loop untuk melakukan iterasi yang ditulis langsung pada baris perintah.

```
>x=2; repeat x=(x+2/x)/2; until x^2~=2; end; x,
```

1.41421356237

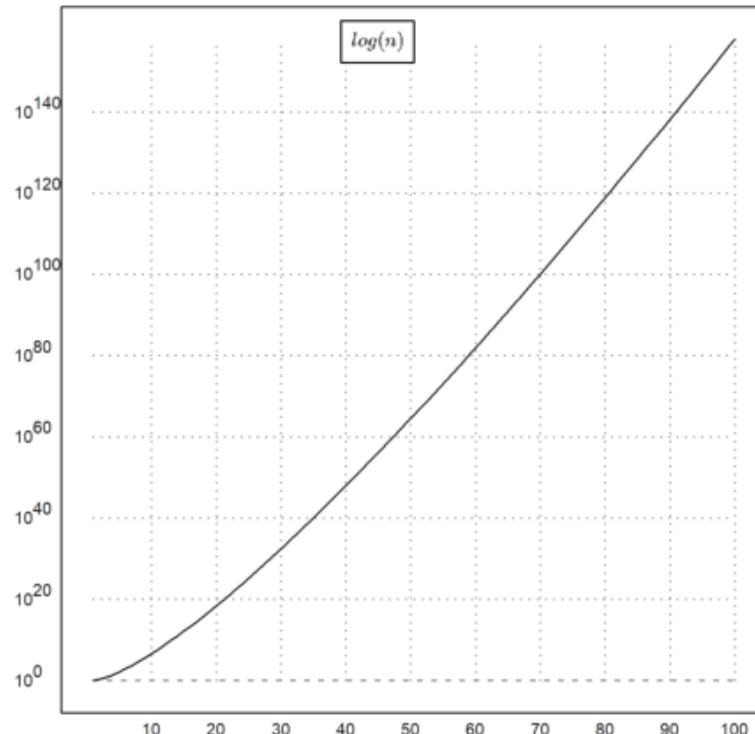
Penggabungan matriks menggunakan tanda "|" dapat digunakan untuk menyimpan semua hasil iterasi.

```
>v=[1]; for i=2 to 8; v=v|(v[i-1]*i); end; v,
```

[1, 2, 6, 24, 120, 720, 5040, 40320]

hasil iterasi juga dapat disimpan pada vektor yang sudah ada.

```
>v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]*i; end; ...
>plot2d(v,logplot=1); textbox(latex(&log(n)),x=0.5):
```



```
>A =[0.5,0.2;0.7,0.1]; b=[2;2]; ...
>x=[1;1]; repeat xnew=A.x-b; until all(xnew~=x); x=xnew; end; ...
>x,
```

-7.09677  
-7.74194

## Iterasi di dalam Fungsi

---

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.

Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

```
>function map hiter(f$,x0) ...
```

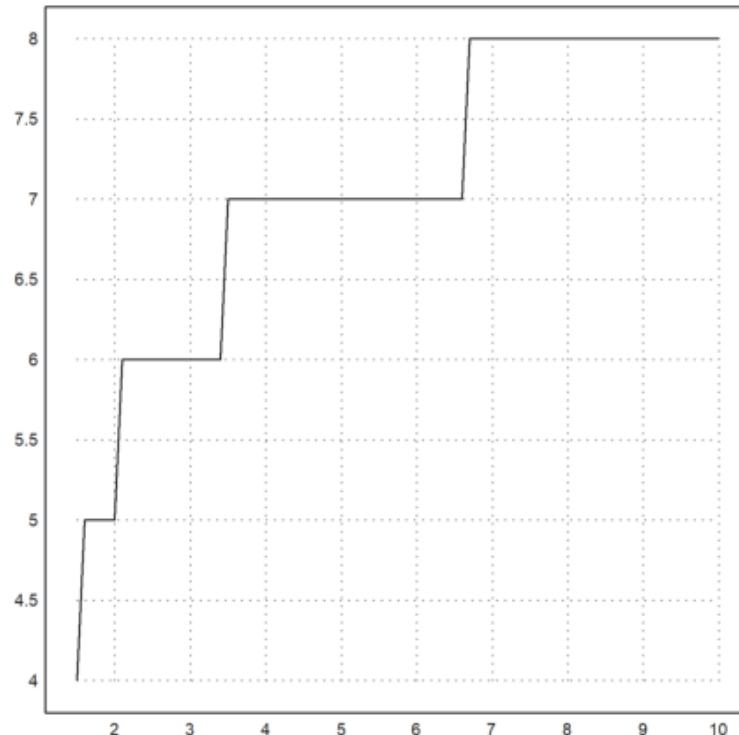
```
x=x0;
maxiter=0;
repeat
    xnew=f$(x);
    maxiter=maxiter+1;
    until xnew~=x;
    x=xnew;
end;
return maxiter;
endfunction
```

Misalnya, berikut adalah iterasi untuk mendapatkan hampiran akar kuadrat 2, cukup cepat, konvergen pada iterasi ke-5, jika dimulai dari hampiran awal 2.

```
>hiter("(x+2/x)/2",2)
```

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):
```

```
Variable W not found!  
Error in:  
C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1) ...  
^
```

## Iterasi Simbolik

---

Seperti sudah dibahas sebelumnya, untuk menghasilkan barisan ekspresi simbolik dengan Maxima dapat digunakan fungsi makelist().

```
>&powerdisp:true // untuk menampilkan deret pangkat mulai dari suku berpangkat terkecil
```

```
true
```

```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Taylor untuk e^x
```

```
Maxima said:  
taylor: 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]
```

*deret*<sub>3</sub>

```
>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)
```

$$\sin(r(t - \sin t)) + \frac{\sin(2r(t - \sin t))}{2} + \frac{\sin(3r(t - \sin t))}{3} + \frac{\sin(4r(t - \sin t))}{4} + \frac{\sin(5r(t - \sin t))}{5}$$

Berikut adalah cara menggambar kurva

$$y = \sin(x) + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \dots$$

```
>plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi):
```

```

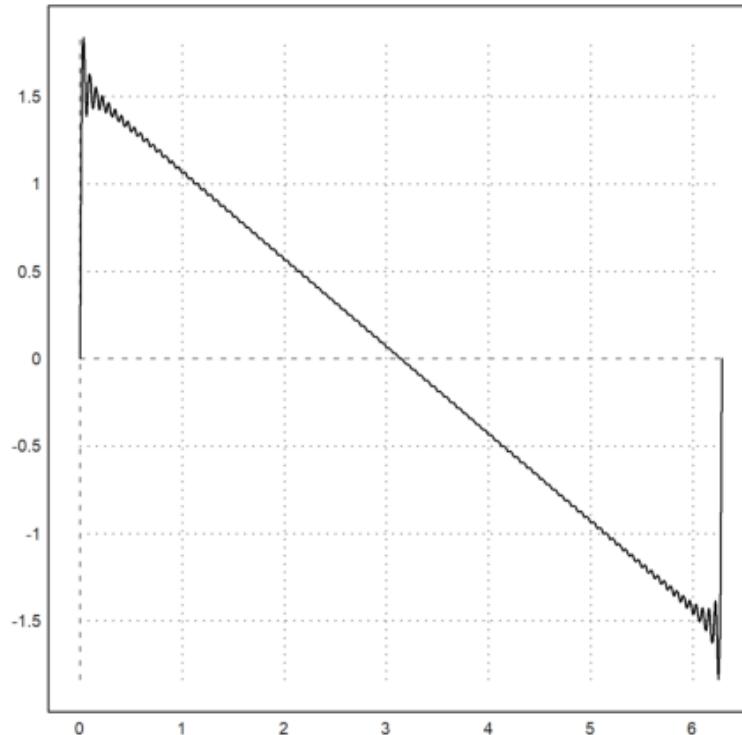
Variable or function t not found.
Error in expression: sin(r*(t-sin(t)))+sin(3*r*(t-sin(t)))/3+sin(5*r*(t-sin(t)))/5+sin(7*r*(t-sin(t)))/7
%ploteval:
y0=f$(x[1],args());
adaptiveevalone:
s=%ploteval(g$,t,args());
Try "trace errors" to inspect local variables after errors.
plot2d:
dw/n,dw/n^2,dw/n,auto,args());

```

Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan kita akan menggambar kurva

$$y = \sum_{k=1}^{100} \frac{\sin(kx)}{k}, \quad 0 \leq x \leq 2\pi.$$

```
>x=linspace(0,2pi,1000); k=1:100; y=sum(sin(k*x')/k)'; plot2d(x,y):
```



## Tabel Fungsi

---

Terdapat cara menarik untuk menghasilkan barisan dengan ekspresi Maxima. Perintah mxmtable() berguna untuk menampilkan dan menggambar barisan dan menghasilkan barisan sebagai vektor kolom.

Sebagai contoh berikut adalah barisan turunan ke-n  $x^n$  di  $x=1$ .

```
>mxmtable("diffat(x^n,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);  
  
%mxmegttable:  
    return mxm("@expr,@var=@value")();  
Try "trace errors" to inspect local variables after errors.  
mxmtable:  
y[#,1]=%mxmegttable(expr,var,x[#]);
```

```
>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:menghitung deret secara sim
```

$$\sum_{k=1}^n k = \frac{n(1+n)}{2}$$

```
> $'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))
```

$$\sum_{k=0}^{\infty} \frac{1}{k+3^k} = \sum_{k=0}^{\infty} \frac{1}{k+3^k}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
> $'sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai ekspresi
```

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

```
> $'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^{-1+k}}{k} = - \sum_{x=1}^{\infty} \frac{(-1)^x}{x}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
> $'sum((-1)^k/(2*k-1), k, 1, inf) = factor(ev(sum((-1)^k/(2*k-1), k, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{-1 + 2k} = \sum_{k=1}^{\infty} \frac{(-1)^k}{-1 + 2k}$$

```
> $'ev(sum(1/n!, n, 0, inf),simpsum=true)
```

$$\sum_{n=0}^{\infty} \frac{1}{n!}$$

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
> &assume(abs(x)<1); $'sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forget(abs(x)<1)
```

Answering "Is  $-1+abs(-r*t+r*sin(t))$  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 ...  
^

Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1.

```
>$'sum(x^k/k! ,k,0,inf)=ev(sum(x^k/k! ,k,0,inf),simpsum=true)
```

$$\sum_{k=0}^{\infty} \frac{r^k (t - \sin t)^k}{k!} = \sum_{k=0}^{\infty} \frac{r^k (t - \sin t)^k}{k!}$$

```
>$limit(sum(x^k/k! ,k,0,n) ,n,inf)
```

$$\lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{r^k (t - \sin t)^k}{k!}$$

```
>function d(n) &= sum(1/(k^2-k) ,k,2,n); '$'d(n)=d(n)
```

$$d(n) = \sum_{k=2}^n \frac{1}{-k + k^2}$$

```
>$d(10)=ev(d(10),simpsum=true)
```

$$\sum_{k=2}^{10} \frac{1}{-k + k^2} = \frac{9}{10}$$

```
>$d(100)=ev(d(100),simpsum=true)
```

$$\sum_{k=2}^{100} \frac{1}{-k + k^2} = \frac{99}{100}$$

**Deret Taylor**

---

Deret Taylor suatu fungsi  $f$  yang diferensiabel sampai tak hingga di sekitar  $x=a$  adalah:

$$f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^k f^{(k)}(a)}{k!}.$$

```
> $'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai suku ke-11
```

```
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 ...  
^
```

## Chapter 5

# Penggunaan EMT untuk Geometri

## 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 lingkran c
circleCircleIntersections (c1, c2): titik potong lingkaran c1 dan c2
planeThrough(A, B, C): bidang melalui titik A, B, C
```

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

```
getLineEquation (g,x,y): persamaan garis g dinyatakan dalam x dan y  
getHesseForm (g,x,y,A): bentuk Hesse garis g dinyatakan dalam x dan y dengan titik A pada  
sisi positif (kanan/atas) garis  
quad(A,B): kuadrat jarak AB  
spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni  $\sin(\alpha)^2$  dengan  
alpha sudut yang menghadap sisi a.  
crosslaw(a,b,c,sa): persamaan 3 quads dan 1 spread pada segitiga dengan panjang sisi a, b, c.  
triplespread(sa,sb,sc): persamaan 3 spread sa,sb,sc yang 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  
>function f(x)
```

Sekarang atur ketiga titik dan plotkan

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik  
>B=[0,1]; plotPoint(B,"B");  
>C=[2,2]; plotPoint(C,"C");
```

Lalu tiga segmen

```
>plotSegment(A,B,"c"); // c=AB  
>plotSegment(B,C,"a"); // a=BC  
>plotSegment(A,C,"b"); // b=AC
```

Fungsi geometri termasuk kedalam fungsi untuk membuat garis dan lingkaran. Format untuk garis adalah [a, b, c] yang mana merepresentasikan garis dengan persamaan  $ax+by = c$ .

```
>lineBC = lineThrough(B,C);  
>lineAB = lineThrough(A,B);  
>lineAC = lineThrough(A,C);  
>lineBC // garis yang melalui B dan C
```

[-1, 2, 2]

Menghitung garis singgung melalui A dalam BC.

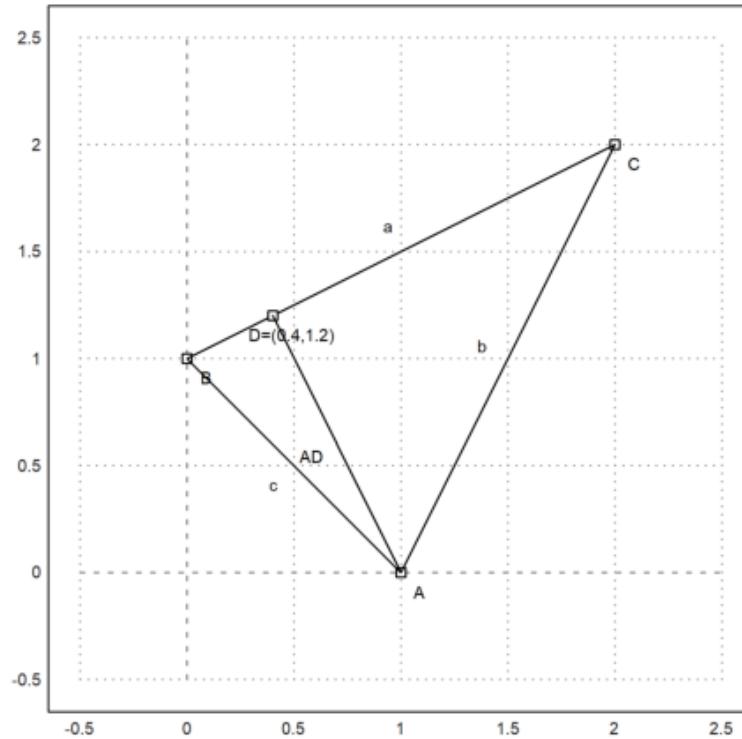
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

dan titik potong dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Buat plotnya.

```
>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 formula determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langsung dengan fungsi
```

1.5

Cara lain menghitung luas segitiga ABC:

```
>distance(A,D)*distance(B,C)/2
```

1.5

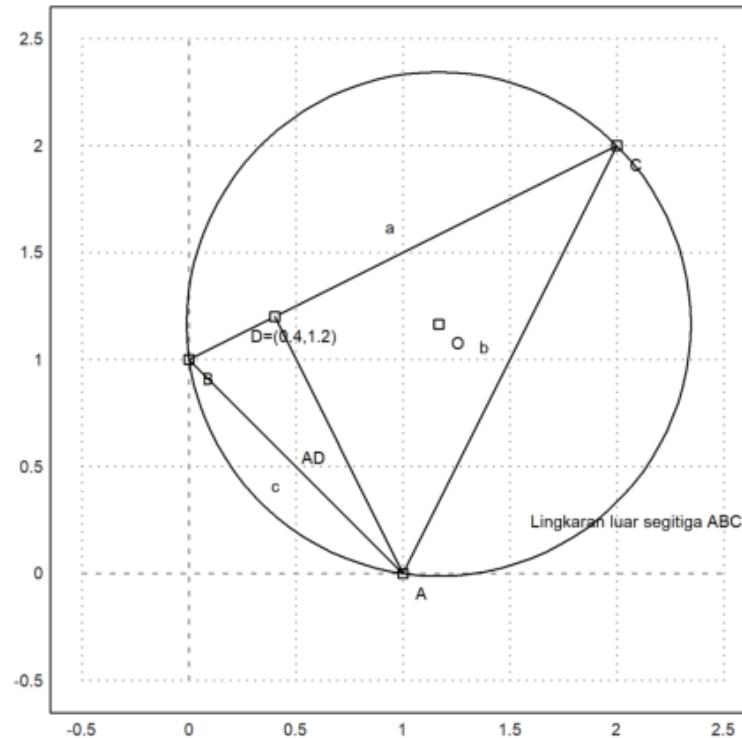
Sudut pada C.

```
>degprint(computeAngle(B,C,A))
```

$36^\circ 52' 11.63''$

Sekarang keliling lingkaran dari segitiga

```
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC  
>R=getCircleRadius(c); // jari2 lingkaran luar  
>O=getCircleCenter(c); // titik pusat lingkaran c  
>plotPoint(O,"O"); // gambar titik "O"  
>plotCircle(c,"Lingkaran luar segitiga ABC");
```



Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>0, 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
```

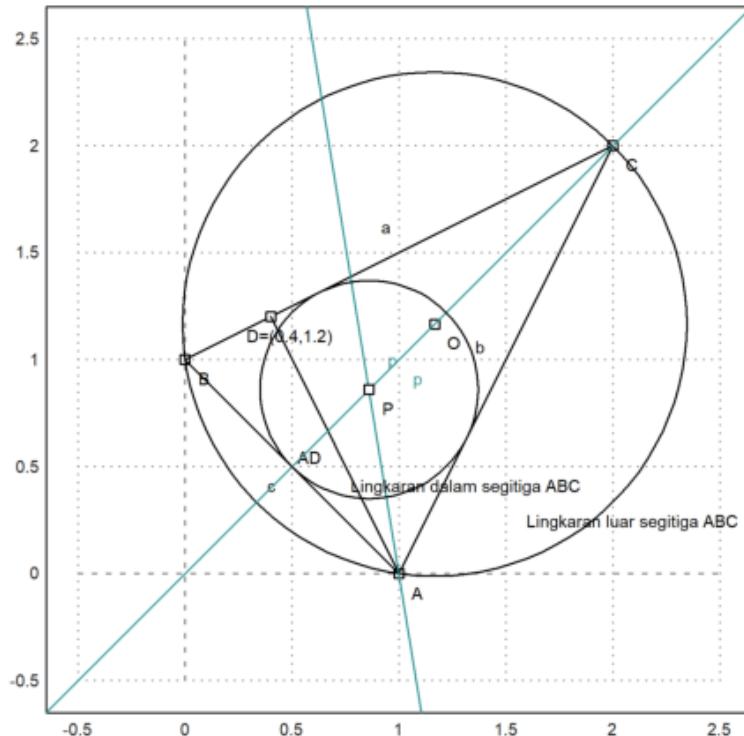
```
[0.86038, 0.86038]
```

dan plotkan semuanya.

```
>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
```

```
>circleInside=circleWithCenter(P,r);
>plotCircle(circleInside, "Lingkaran dalam segitiga ABC"); // gambar lingkaran dalam
```



1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.
  2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. Merupakan segitiga apakah itu?
  3. Hitung luas segitiga tersebut.
  4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.
  5. Gambar jari-jari lingkaran dalam.
  6. Hitung luas lingkaran luar dan luas lingkaran dalam segitiga ABC. Adakah hubungan antara luas kedua lingkaran tersebut dengan luas segitiga ABC?
- 
1. Akan ditentukan Ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC

```
>E = lineCircleIntersections(lineAB, circleInside) // intersection line with line AB
```

[0.5, 0.5]

```
>F = lineCircleIntersections(lineAC, circleInside) // intersection line with line AC
```

[1.31623, 0.632456]

```
>G = lineCircleIntersections(lineBC, circleInside) // intersection line with line BC
```

[0.632456, 1.31623]

Jadi titik singgung lingkaran dalam dengan sisi-sisi segitiga AB adalah  
Untuk titik singgung lingkaran dalam dengan ruas garis AB adalah

(0.5, 0.5)

Untuk titik singgung lingkaran dalam dengan ruas garis AC adalah

(1.31, 0.632)

Untuk titik singgung lingkaran dalam dengan ruas garis BC adalah

(0.632, 1.316)

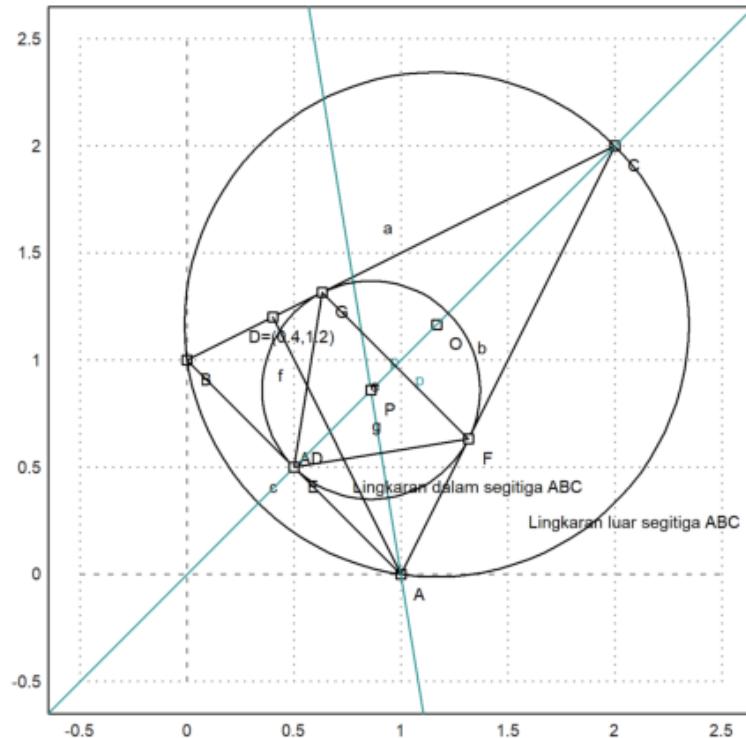
2. Akan digambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. dan akan ditentukan jenis segitiagnya.

Pertama-tama buat titik

```
>plotPoint(E, "E");
>plotPoint(F, "F");
>plotPoint(G, "G");
```

Kemudian buat segmennya

```
>plotSegment(E, F, "g"); // g = EF  
>plotSegment(E, G, "f"); // f = EG  
>plotSegment(F, G, "e"); // e = FG
```



```
>distanceEF = distance(E, F);
>distanceEG = distance(E, G);
>distanceFG = distance(F, G);
>distanceEF
```

0.826905214631

```
>distanceEG
```

0.826905214631

```
>distanceFG
```

0.966999966873

Karena ruas garis EF dan EG sama, akan tetapi ruas garis FG berbeda dengan ketiganya, maka segitiga ini termasuk kedalam segitiga sama kaki.

3. Akan dihitung luas segitiga tersebut  
dapat dengan mudah dengan fungsi areaTriangle

```
>areaTriangle(E, F, G)
```

0.324341649025

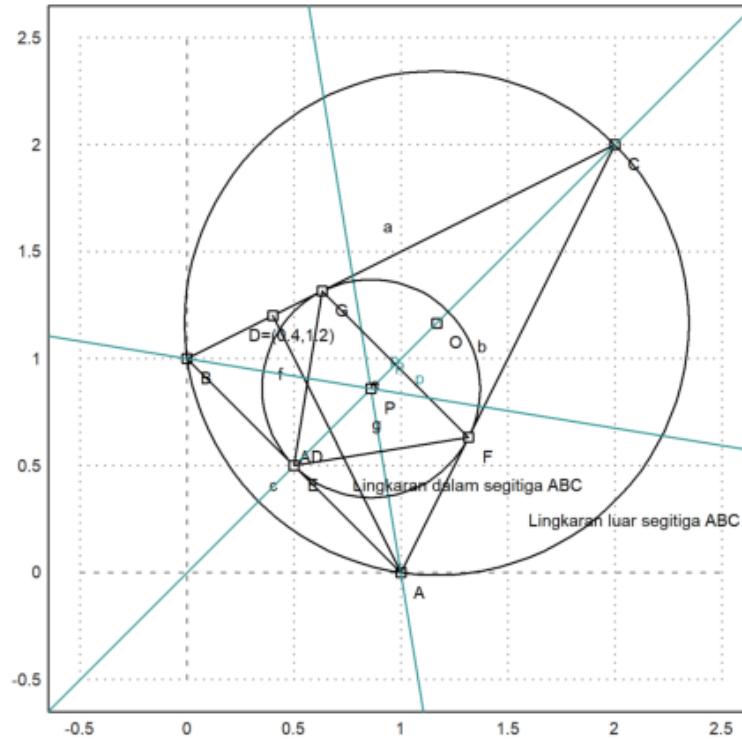
Diperoleh luas segitiga sekitar

0.3243

dengan satuan luas

4. Akan ditunjukkan garis bagi sudut yang ketiga melalui titik pusat lingkaran dalam

```
>h = angleBisector(C, B, A); // garis bagi <CBA  
>color(5); plotLine(h):
```



Terlihat bahwa garis bagi untuk CBA akan melalui titik pusat lingkaran dalam

```
>S = lineIntersection(l, h);
>T = lineIntersection(g, h);
>S
```

[0.86038, 0.86038]

>T

[0.86038, 0.86038]

>P

[0.86038, 0.86038]

Lebih lanjut dibuktikan bahwa titik potong untuk garis dengan titik sudut CBA akan melalui titik pusat lingkaran dalam dimana titik potongnya

untuk titik potong garis bagi CBA dengan garis bagi ACB pada titik

(0.86, 0.86)

untuk titik potong garis bagi CBA dengan garis bagi CAB pada titik

(0.86, 0.86)

dengan titik pusat lingkaran dalamnya adalah

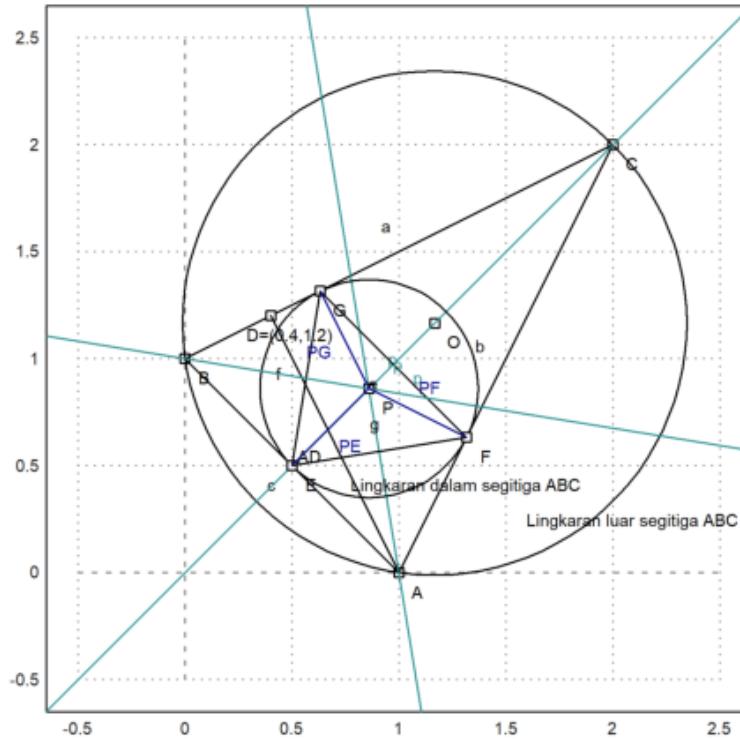
(0.86, 0.86)

Sehingga, garis bagi untuk CBA akan melalui titik pusat lingkaran dalam

5. Akan digambar jari-jari lingkaran dalam

Dengan mudah dapat digambarkan antara titik PG, PF, atau PE

```
>color(4); plotSegment(P, G, "PG"); plotSegment(P, F, "PF"); plotSegment(P, E, "PE"); color(1):
```



6. Akan dihitung luas lingkaran luar dan luas lingkaran dalam segitiga ABC. Akan ditentukan hubungan-gannya.

```
>areaCircleOutside = pi * R^2;
>areaCircleInside = pi * r^2;
>areaCircleOutside
```

4.36332312999

```
>areaCircleInside
```

0.81601903655

Hubungan yang didapatkan adalah

- Jari-jari lingkaran luar lebih panjang daripada jari-jari lingkaran dalam segitiga, sehingga luasnya juga demikian

## Contoh 2: Geometri Smbolik

---

Kita dapat menghitung dengan tepat dan geometri simbolik menggunakan Maxima.

File geometry.e menyediakan fungsi-fungsi yang sama (dan lebih) dalam Maxima. Namun. kita sekarang dapat menggunakan perhitungan simbolik.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Fungsi-fungsi untuk garis dan lingkaran bekerja seperti fungsi-fungsi Euler, tetapi menyediakan komputasi simbolik.

```
>c &= lineThrough(B,C) // c=BC
```

[- 1, 2, 2]

Kita dapat mendapatkan persamaan untuk sebuah garis dengan mudah.

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$\left[ y = \frac{x}{2} + 1 \right]$$

$$\left[ y = \frac{x}{2} + 1 \right]$$

```
>$getLineEquation(lineThrough([x1,y1],[x2,y2]),x,y), $solve(%,y) // persamaan garis melalui(x1, y1)
```

$$\left[ y = \frac{-(x_1 - x) y_2 - (x - x_2) y_1}{x_2 - x_1} \right]$$

$$\left[ y = \frac{-(x_1 - x) y_2 - (x - x_2) y_1}{x_2 - x_1} \right]$$

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
```

$$(x_1 - 1) y - x y_1 = -y_1$$

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

[2, 1, 2]

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

$$\begin{bmatrix} 2 & 6 \\ [-, -] \\ 5 & 5 \end{bmatrix}$$

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\left[ \frac{2}{5}, \frac{6}{5} \right]$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[ \frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}} \right]$$

```
>r=&getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

1.178511301977579

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 garis bagi s
```

$$\left[ \frac{\sqrt{2}\sqrt{5}+2}{6}, \frac{\sqrt{2}\sqrt{5}+2}{6} \right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

```
[0.86038, 0.86038]
```

## Perpotongan Garis dan Lingkaran

---

Tentu saja, kita juga dapat membuat perpotongan garis dengan lingkaran, dan lingkaran dengan lingkaran.

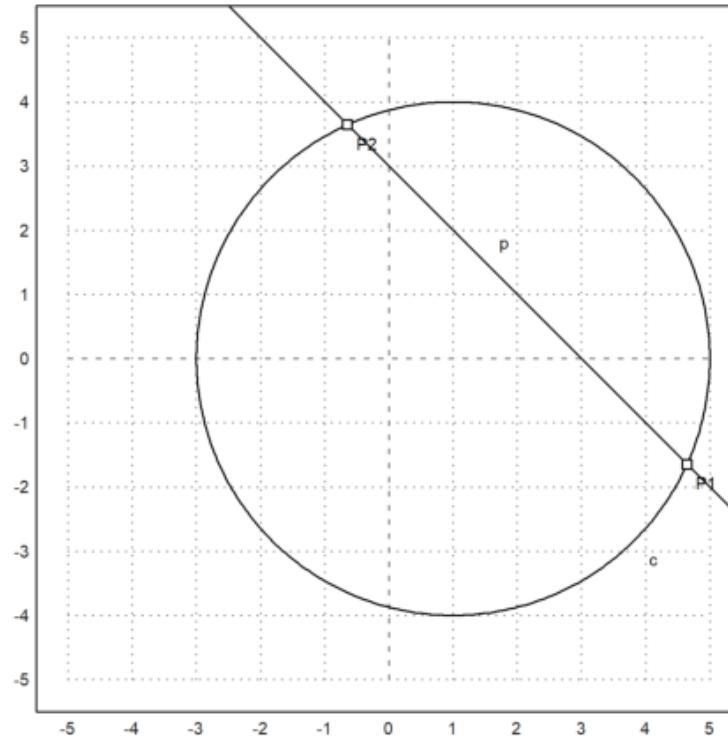
```
>A &:= [1,0]; c=circleWithCenter(A,4);
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l);
```

Perpotongan garis dengan lingkaran mengembalikan dua titik dan jumlah titik yang berpotongan

```
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2, f
```

```
[4.64575, -1.64575]
[-0.645751, 3.64575]
2
```

```
>plotPoint(P1); plotPoint(P2);
```



Sama dalam Maxima.

```
>c &= circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

[1, 0, 4]

```
>l &= lineThrough(B,C) // garis l melalui B dan C
```

[1, 1, 3]

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan garis l
```

$$\left[ \left[ \sqrt{7} + 2, 1 - \sqrt{7} \right], \left[ 2 - \sqrt{7}, \sqrt{7} + 1 \right] \right]$$

Akan ditunjukkan bahwa sudut-sudut yang menghadap busur yang sama adalah sama besar.

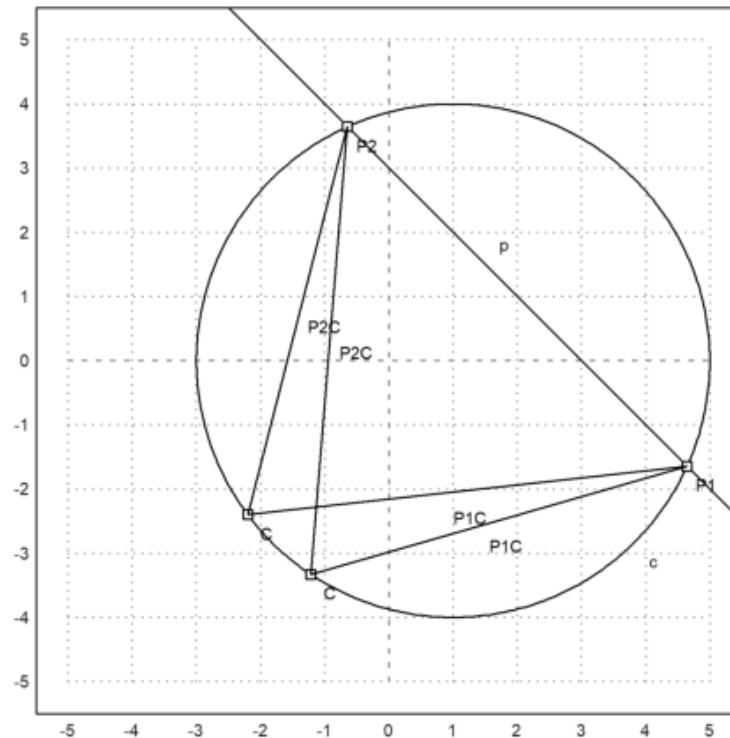
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
```

69°17'42.68''

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
```

$69^{\circ}17'42.68''$

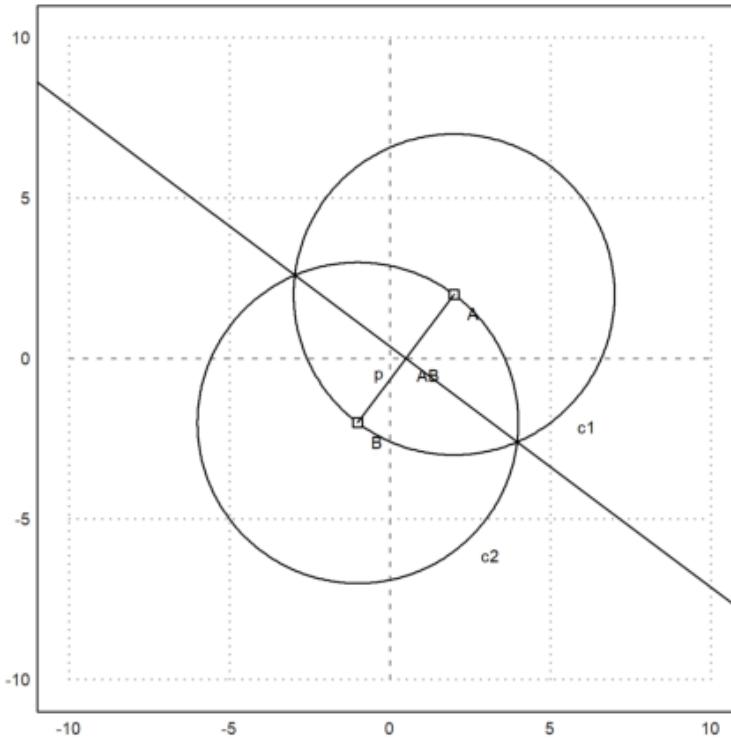
```
>insimg;
```



Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B.
2. Gambar lingkaran dengan pusat B melalui A.
3. Tarik garis melalui kedua titik potong kedua lingkaran tersebut. Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

```
>A=[2,2]; B=[-1,-2];
>c1=circleWithCenter(A,distance(A,B));
>c2=circleWithCenter(B,distance(A,B));
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(-10, 10, -10, 10); plotCircle(c1); plotCircle(c2);
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l);
```



Selanjutnya, kami lakukan yang sama pada 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-persamaan untuk perpotongan sedikit rumit. Tetapi kita dapat menyederhanakan, jika kita menyelesaikan untuk y.

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);  
>$solve(g,y)
```

$$\left[ y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

Ini sebenarnya sama seperti perpotongan tengah, yang mana dihitung dengan cara yang secara keseluruhan berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[ y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

```
>h &= getLineEquation(lineThrough(A,B),x,y);  
>$solve(h,y)
```

$$\left[ y = \frac{(b_2 - a_2)x - a_1 b_2 + a_2 b_1}{b_1 - a_1} \right]$$

Perhatikan hasil kali gradien garis g dan h adalah:

$$\frac{-(b_1 - a_1)}{(b_2 - a_2)} \times \frac{(b_2 - a_2)}{(b_1 - a_1)} = -1.$$

Artinya kedua garis tegak lurus.

### Contoh 3: Rumus Heron

---

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a, b dan c adalah:

$$L = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{dengan } s = (a+b+c)/2,$$

atau bisa ditulis dalam bentuk lain:

$$L = \frac{1}{4} \sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}$$

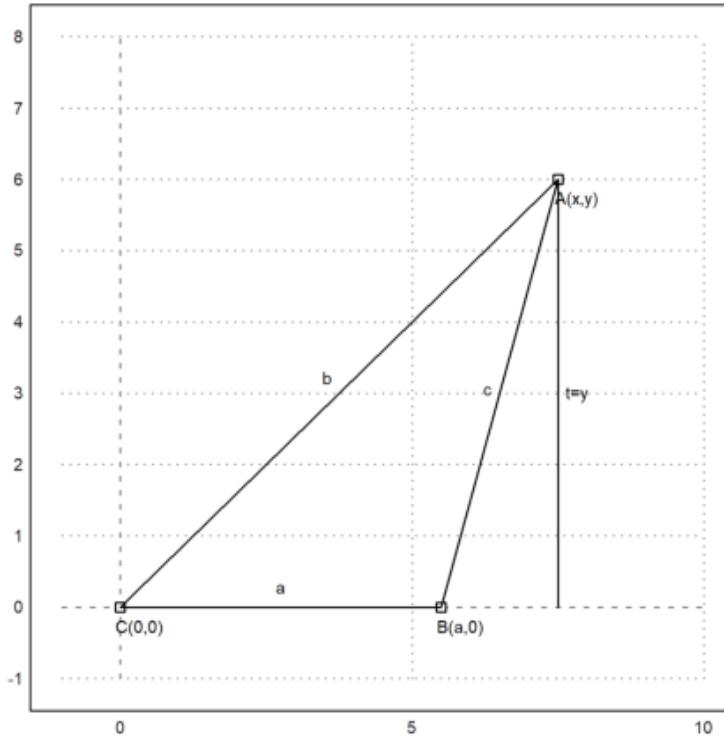
Untuk membuktikan hal ini kita misalkan C(0,0), B(a,0) dan A(x,y), b=AC, c=AB. Luas segitiga ABC adalah

$$L_{\triangle ABC} = \frac{1}{2} a \times y.$$

Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2 + y^2 = b^2, \quad (x - a)^2 + y^2 = c^2.$$

```
>setPlotRange(-1,10,-1,8); plotPoint([0,0], "C(0,0)"); plotPoint([5.5,0], "B(a,0)"); ...
> plotPoint([7.5,6], "A(x,y)");
>plotSegment([0,0],[5.5,0], "a",25); plotSegment([5.5,0],[7.5,6],"c",15); ...
>plotSegment([0,0],[7.5,6],"b",25);
>plotSegment([7.5,6],[7.5,0],"t=y",25):
```



```
>&assume(a>0); sol &= solve([x^2+y^2=b^2, (x-a)^2+y^2=c^2], [x, y])
```

[]

Ekstrak solusi y.

```
>ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2))
```

```
Maxima said:  
part: invalid index of list or matrix.  
-- an error. To debug this try: debugmode(true);
```

```
Error in:  
ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2)) ...  
^
```

Kami dapatkan formula Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H(a, b, [1, 0, 4]) = \frac{a |ysol|}{2}$$

```
>$'Luas=H(2,5,6) // luas segitiga dengan panjang sisi-sisi 2, 5, 6
```

$$Luas = |ysol|$$

Tentu saja, setiap bidang segitiga adalah kasus yang diketahui.

```
>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 a*abs(ysol)/2  
Error in:  
H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...  
^
```

dan ini tentu saja jelas, bahwa ini adalah segitiga dengan luas maxima dan dua sisi 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7); // Kurva luas segitiga sengan panjang sisi 3, 4, x (1<= x <=7)
```

```
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 dapat bekerja juga.

```
>$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 andaikan kita menemukan himpunan dari semua titik-titik dimana  $b+c=d$  untuk sebarang konstan d. Seperti yang diketahui bahwa ini adalah sebuah 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 ...
```

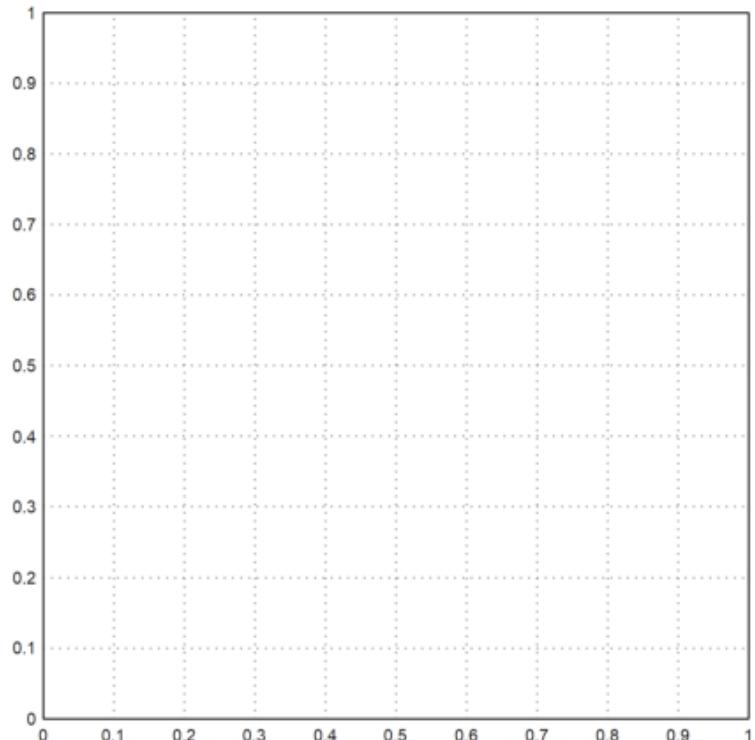
dan buat fungsi seperti berikut.

```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); $fy(a,c,d)
```

0

Sekarang kami dapat menggambar himpunan. Sisi b bervariasi dari 1 ke 4. Yang mana kami dapatkan sebuah elips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```



Kami dapat menguji persamaan umum untuk elips ini, dengan kata lain.

$$\frac{(x - x_m)^2}{u^2} + \frac{(y - y_m)^2}{v^2} = 1,$$

dimana  $(xm, ym)$  adalah titik tengah, dan  $u$  dan  $v$  merupakan pertengahan sumbu-sumbu.

```
>$ratsimp((fx(a,c,d)-a/2)^2/u^2+fy(a,c,d)^2/v^2 with [u=d/2,v=sqrt(d^2-a^2)/2])
```

$$\frac{a^2}{d^2}$$

Kami lihat bahwa tinggi dan luas dari segitiga adalah nilai maksimal untuk  $x=0$ . Lantas luas dari segitiga dengan  $a+b+c=d$  adalah maksimal, jika ini adalah ekuilateral. Kami berharap menurunkan ini secara analitik.

```
>eqns &= [diff(H(a,b,d-(a+b))^2,a)=0,diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

$$\left[ \frac{a ysol^2}{2} = 0, 0 = 0 \right]$$

Kami dapatkan beberapa minimum, yang mana bagian dari segitiga satu sisi 0, dan solusi  $a=b=c=d/3$ .

```
>$solve(eqns,[a,b])
```

$$[[a = 0, b = \%r_1]]$$

Ini juga merupakan metode Langrange, memaksimalkan  $H(a, b, c)^2$  dengan kaitan ke  $a+b+c=d$ .

```
>&solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la, ...
>      diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la])
```

```
Maxima said:  
diff: second argument must be a variable; found [1,0,4]  
-- an error. To debug this try: debugmode(true);  
  
Error in:  
... la,      diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la]) ...  
^
```

Kami dapat membuat sebuah plot dari situasi.

Pertama atur titik-titik dalam 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
```

[ $a, 0]$

Kemudian atur plot range, dan titik-titik dari plot.

```
>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"): ...
```

Plotkan segmen-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 perpotongan tengah-tengah dalam Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

dan tengah dari keliling.

```
>U &= lineIntersection(h,g);
```

Kami dapatkan formula untuk jari-jari keliling.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

$$\frac{\sqrt{a_2^2 + a_1^2} \sqrt{a_2^2 + a_1^2 - 2 a a_1 + a^2}}{2 |a_2|}$$

Biarkan kami menambahkan 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 turunkan formula sederhana.

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk jari-jari. Dapat kita uji, jika ini benar dengan Maxima. Maxima akan membuat faktor ini hanya jika kita meng-kuadratkannya.

```
>$c^2/sin(computeAngle(A,B,C))^2 | factor
```

$$\left[ \frac{a_2^2 + a_1^2}{a_2^2}, 0, \frac{16 (a_2^2 + a_1^2)}{a_2^2} \right]$$

#### Contoh 4: Garis Euler dan Parabola

---

Garis Euler adalah sebuah garis yang ditentukan dari sebarang segitiga yang mana tidak ekuilateral. Ini merupakan garis pusat dari segitiga dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk orthocenter, titik pusat dan sentroid, titik Exeter dan pusat adalah sembilan-titik lingkaran dari segitiga.

Untuk demonstrasi, kami menghitung dan mem-plot garis Euler dalam sebuah segitiga.

Pertama, kita definisikan sudut-sudut dari segitiga dalam Euler. Kami gunakan sebuah definisi, yang mana bisa dilihat dalam ekspresi simbolik.

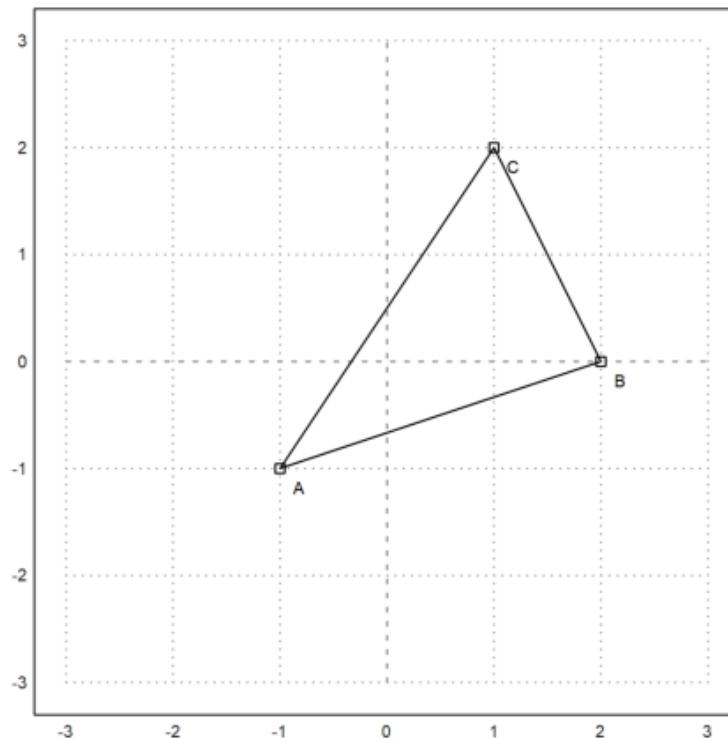
```
>A:=[-1,-1]; B:=[2,0]; C:=[1,2];
```

Untuk membuat plot objek-objek geometrik, kami mengatur sebuah luasan plot dan menambahkan titik-titik terhadap itu. Semua plot dari objek-objek geometrik ditambahkan ke plot sekarang.

```
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```

Kami juga menambahkan sisi-sisi dari segitiga

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,"");
```



Ini adalah luasan segitiga, menggunakan formula determinan. Tentusaja, kita dapat mengambil nilai absolut dari hasil ini.

```
>$areaTriangle(A,B,C)
```

$$-\frac{7}{2}$$

Kami dapat menghitung koefisien-koefisien dari sisi c.

```
>c &= lineThrough(A,B)
```

$$[-1, 3, -2]$$

Dan juga mendapatkan sebuah formula untuk ini.

```
>$getLineEquation(c,x,y)
```

$$3y - x = -2$$

Untuk bentuk Hesse, kita butuh untuk menentukan sebuah titik, yang mana titik dalam sisi positif dari bentuk Hesse. Memasukkan titik menghasilkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$$\frac{7}{\sqrt{10}}$$

$$\frac{7}{\sqrt{10}}$$

Sekarang kita menghitung keliling dari ABC

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

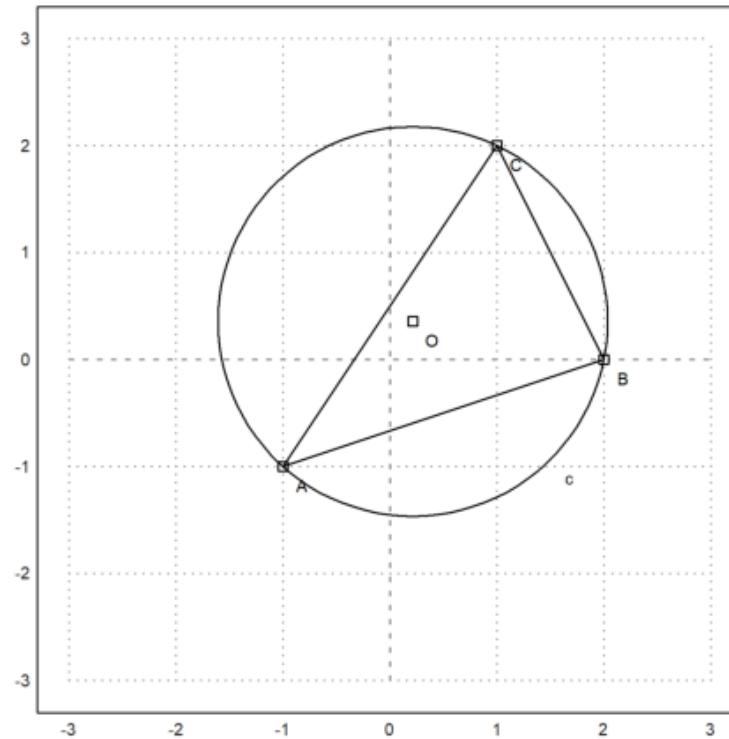
$$\left(y - \frac{5}{14}\right)^2 + \left(x - \frac{3}{14}\right)^2 = \frac{325}{98}$$

```
>O &= getCircleCenter(LL); $O
```

$$\left[\frac{3}{14}, \frac{5}{14}\right]$$

Menggambarkan lingkaran dan tengahnya. Cu dan U secara simbolik. Kami mengevaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(0(),"O");
```



Kita dapat menghitung potongan dari tinggi dalam ABC (orthocenter) secara numerik dengan mengikuti perintah berikut.

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...  
>  perpendicular(B,lineThrough(A,C))); $H
```

$$\left[ \frac{11}{7}, \frac{2}{7} \right]$$

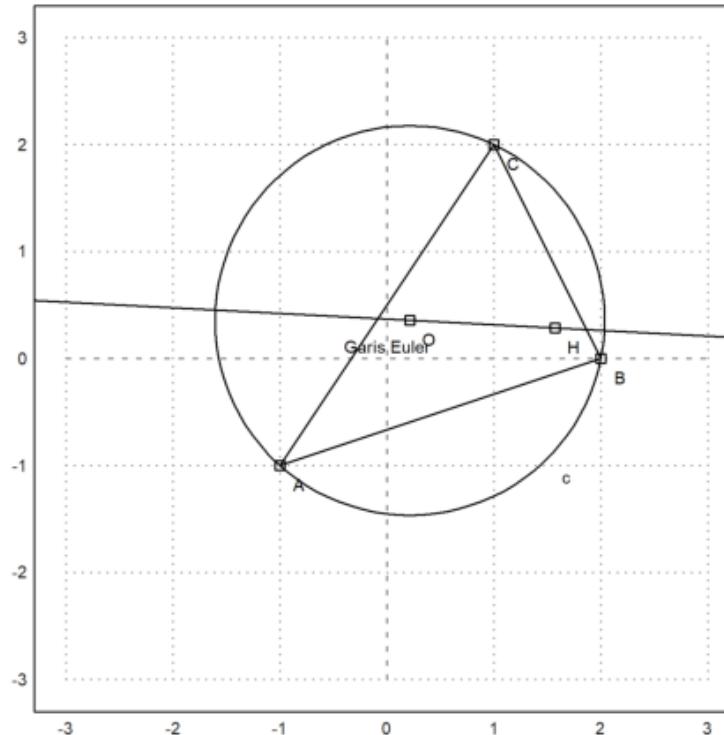
Sekarang kita dapat menghitung garis Euler dari segitiga.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$-\frac{19y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan itu ke gambar kami.

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler");
```

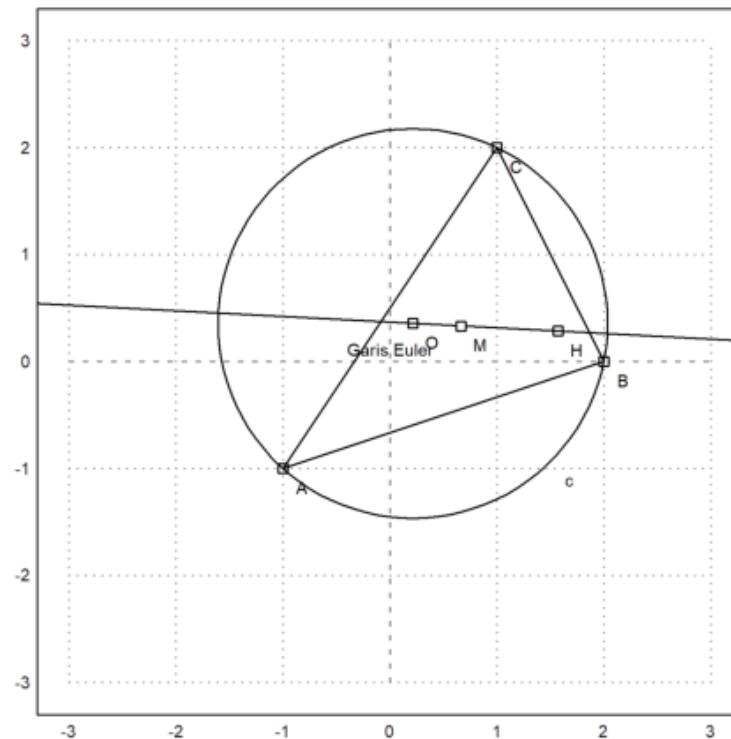


Pusat gravitasi seharusnya ada pada garis ini.

```
>M &= (A+B+C)/3; $getLineEquation(el,x,y) with [x=M[1],y=M[2]]
```

$$-\frac{1}{2} = -\frac{1}{2}$$

```
>plotPoint(M(),"M") // titik berat
```



Teori memberitahu kita  $MH=2^*MO$ . Kita butuh untuk menyederhanakan dengan radcan untuk memperoleh ini.

```
>$distance(M,H)/distance(M,O)|radcan
```

2

Fungsi-fungsi termasuk fungsi untuk sudut juga.

```
>$computeAngle(A,C,B), degprint(%())
```

$$\arccos\left(\frac{4}{\sqrt{5}\sqrt{13}}\right)$$

$60^\circ 15' 18.43''$

Persamaan untuk pusat dari segitiga dalam lingkaran tidak begitu baik.

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $Q
```

$$\left[ \frac{\left(2^{\frac{3}{2}} + 1\right) \sqrt{5} \sqrt{13} - 15 \sqrt{2} + 3}{14}, \frac{(\sqrt{2} - 3) \sqrt{5} \sqrt{13} + 5 2^{\frac{3}{2}} + 5}{14} \right]$$

Lalu kita hitung juga ekspresi untuk jari-jari linkaran yang dalam.

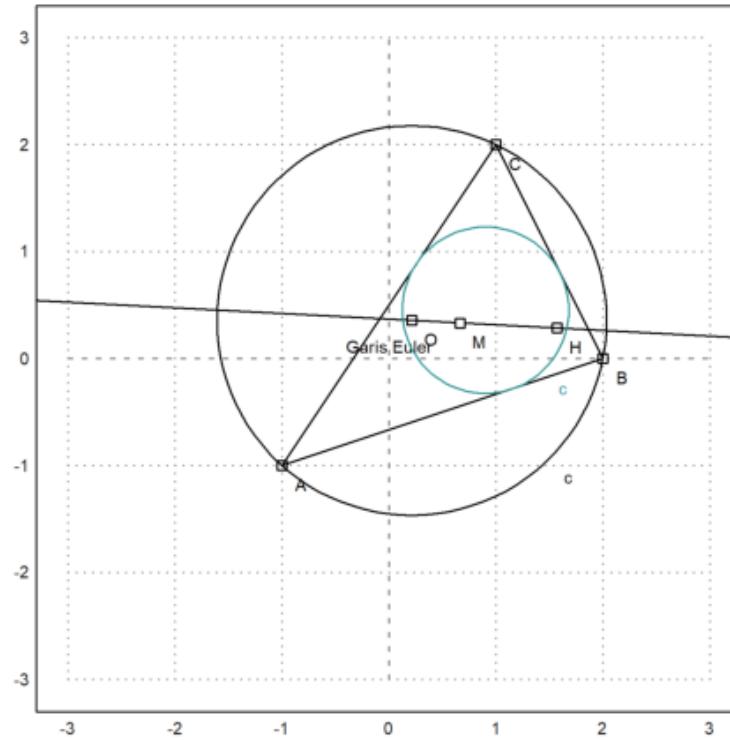
```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{(-41\sqrt{2} - 31)\sqrt{5}\sqrt{13} + 115\sqrt{2} + 614}}{7\sqrt{2}}$$

```
>LD &= circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini kedalam plot.

```
>color(5); plotCircle(LD()):
```



Parabola

---

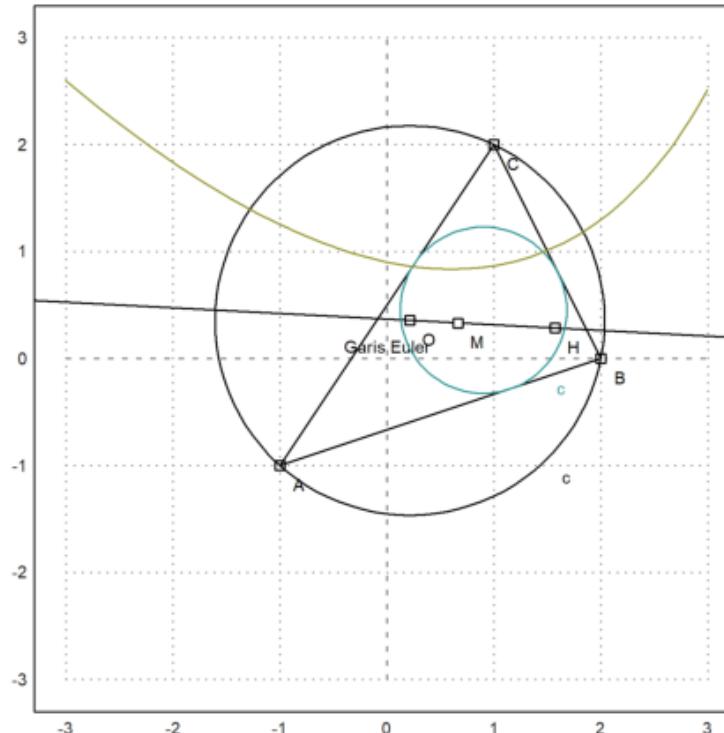
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3y - x + 2}{\sqrt{10}} - \sqrt{(2-y)^2 + (1-x)^2} = 0$$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```



Ini seharusnya merupakan suatu fungsi, tetapi penyelesaian default Maxima hanya dapat menemukan solusinya, jika kita mengkuadratkan persamaannya. Akibatnya, kita mendapatkan solusi palsu.

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

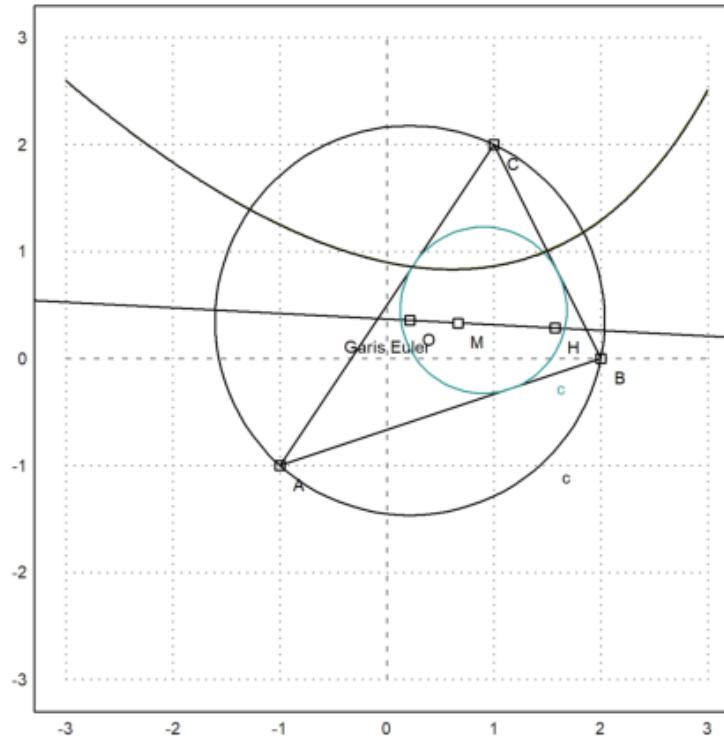
```
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26,
y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 26]
```

Solusi pertama adalah

maxima: akar[1]

Dengan menambahkan solusi pertama ke dalam plot, terlihat bahwa itu memang jalur yang kita cari. Teori tersebut memberi tahu kita bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1):
```



```
>function g(x) &= rhs(akar[1]); '$g(x)= g(x)// fungsi yang mendefinisikan kurva di atas
```

$$g(x) = -3x - \sqrt{70} \sqrt{9 - 2x} + 26$$

```
>T &= [-1, g(-1)]; // ambil sebarang titik pada kurva tersebut  
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
```

2.135605779339061

```
>U &= projectToLine(T,lineThrough(A,B)); $U // proyeksi T pada garis AB
```

$$\left[ \frac{80 - 3\sqrt{11}\sqrt{70}}{10}, \frac{20 - \sqrt{11}\sqrt{70}}{10} \right]$$

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jarak T ke AB
```

2.135605779339061

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

## Contoh 5: Trigonometri Rasional

---

Hal ini terinspirasi dari ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti konsep klasik jarak dan sudut dengan kuadran dan sebaran.

Dengan menggunakan konsep ini, memang memungkinkan untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

Berikut ini, saya memperkenalkan konsep-konsep tersebut, dan memecahkan beberapa masalah. Saya menggunakan perhitungan simbolik

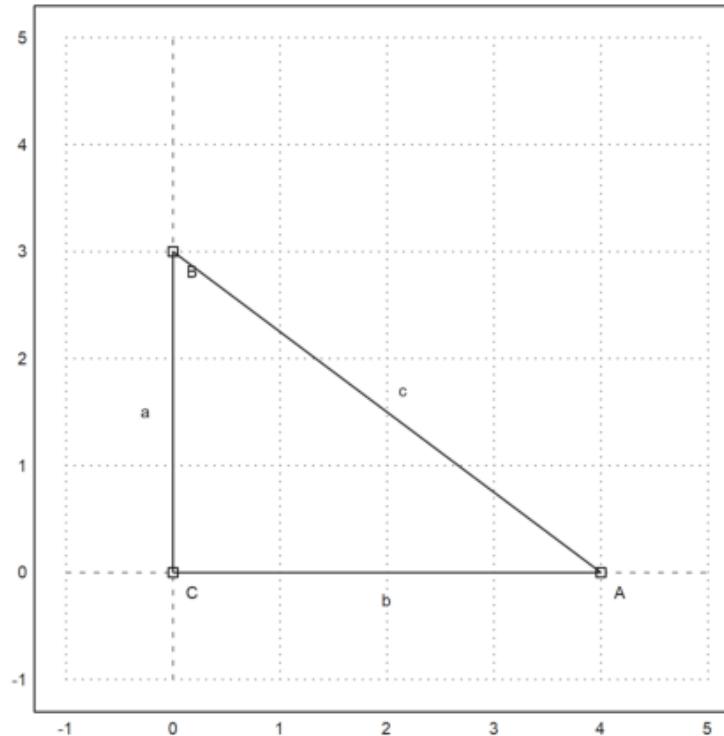
Maxima di sini, yang menyembunyikan keuntungan utama trigonometri rasional bahwa perhitungan dapat dilakukan hanya dengan kertas dan pensil. Anda diundang untuk memeriksa hasilnya tanpa komputer.

Intinya adalah bahwa perhitungan rasional simbolik 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 gunakan segitiga persegi dengan proporsi Mesir yang terkenal 3, 4, dan 5. Mengikuti perintah adalah perintah Euler untuk menggambarkan bidang geometri yang terbungkus 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,

$$\sin(w_a) = \frac{a}{c},$$

dimana wa adalah sudut pada A. cara umum untuk menghitung sudut ini, adalah mengambil balikan dari fungsi sine. Menghasilkan sudut yang sulit dipahami, yang mana hanya dapat secara perkiraan dicetak.

```
>wa := arcsin(3/5); degprint(wa)
```

36°52'11.63'',

Trigonometri rasional mencoba untuk menghindari ini.

Notasi pertama dari trigonometri rasional adalah sebuah kuadran, yang mana menggantikan jarak. Faktanya, ini hanya jarak kuadrat. Ini mengikuti a, b, dan c yang dinotasikan dari sisi-sisi kuadran.

Teorema Pythagoras kemudian menyederhanakan nya menjadi  $a+b=c$

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

25 = 25

Notasi kedua dari trigonometri rasional adalah penyebaran. Persebaran menghitung bukkan antara garis-garis. Ini adalah 0, jika garis adalah paralel, dan 1 jika garis adalah persegi. Itu merupakan persegi dari sin dari sudut antara dua garis.

Persebaran dari garis AB dan AC dalam gambar diatas didefinisikan sebagai

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

dimana a dan c merupakan kuadran-kuadran dari sebarang segitiga siku-siku dengan sudut dalam A.

```
>sa &= a/c; $sa
```

$$\frac{9}{25}$$

Ini merupakan cara termudah untuk menghitung daripada sudut, tentusaja. Tapi kamu kehilangan sifat sudut-sudut dapat ditambahkan dengan mudah.

Tentusaja, kita dapat mengubah nilai perkiraan untuk sudut wa untuk sebuah sebaran dan mencetak itu sebagai sebuah perbandingan.

```
>fracprint(sin(wa)^2)
```

9/25

Aturan kosinus dari trigonometri klasik diterjemahkan ke dalam "aturan perkalian".

$$(c + b - a)^2 = 4bc(1 - s_a)$$

Dimana a, b, dan c merupakan kuadran-kuadran dari sisi-sisi segitiga, dan sa merupakan sebaran pada sudut A. Sisi a adalah, seperti biasa, sebaliknya untuk ke sudut A.

Aturan ini diimplementasikan kedalam file geometry.e yang kami muat ke Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

$$\left[ \left( bb - aa + \frac{7}{6} \right)^2, \left( bb - aa + \frac{7}{6} \right)^2, \left( bb - aa + \frac{5}{3\sqrt{2}} \right)^2 \right] = \left[ \frac{14bb(1-saa)}{3}, \frac{14bb(1-saa)}{3}, \frac{52^{\frac{3}{2}}bb(1-saa)}{3} \right]$$

Dalam kasus kami, kami dapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024 = 1024$$

Mari kita gunakan aturan perkalian untuk menemukan sebaran pada A. Untuk melakukan ini, kita hasilkan aturan perkalian untuk kuadran-kuadran a,b,dan c, dan selesaikannya untuk sebaran tak diketahui sa.

Kamu dapat melakukan ini dengan tangan secara mudah, tetapi Saya menggunakan Maxima. Tentusaja, kami dapatkan hasil, seperti berikut.

```
>$crosslaw(a,b,c,x), $solve(% ,x)
```

$$\left[ x = \frac{9}{25} \right]$$

$$\left[ x = \frac{9}{25} \right]$$

Kami telah tahu ini. Definisi dari sebaran adalah sebuah kasus khusus dari aturan perkalian.

Kami dapat juga menyelesaikan ini secara umum untuk a,b,c. Hasil nya merupakan formula yang mana menghitung sebaran dari sebuah sudut dari sebuah segitiga diberikan kuadran-kuadran dari ketiga sisi-sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[ \left[ \frac{168 bb x + 36 bb^2 + (-72 aa - 84) bb + 36 aa^2 - 84 aa + 49}{36}, \frac{168 bb x + 36 bb^2 + (-72 aa - 84) bb + 36 aa^2 - 84 aa + 49}{36} \right] \right]$$

Kami dapat menggunakan sebuah fungsi dari hasil. Seperti sebuah fungsi yang telah didefinisikan dalam file geometry.e dari Euler.

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Seperti sebuah contoh, kami dapat menggunakannya untuk menghitung sudut dari sebuah segitiga dengan sisi-sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya adalah rasional, yang mana sulit untuk mendapatkannya jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini merupakan sudut dalam derajat.

```
>degprint(arcsin(sqrt(6/7)))
```

$$67^\circ 47' 32.44''$$

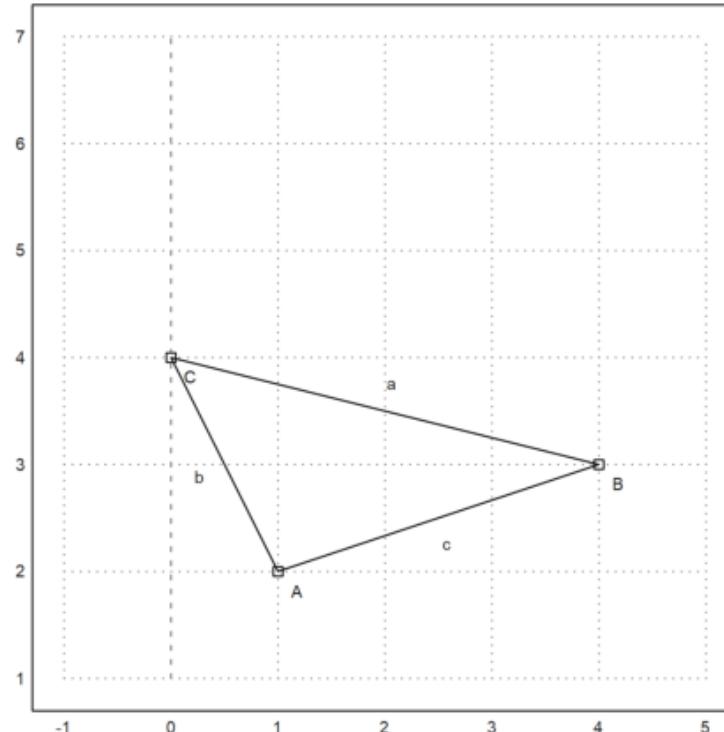
[Contoh Lainnya](#)

---

Sekarang, mari kita coba sebuah contoh yang lebih jauh.

Kami atur tiga titik-titik dari sebuah segitiga mengikuti.

```
>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, sangat mudah untuk menghitung jarak antara dua titik-titik. Pertama, saya gunakan fungsi jarak dari file Euler untuk geometri. Fungsi jarak menggunakan geometri klasik.

```
>$distance(A,B)
```

$$\sqrt{10}$$

Euler juga memuat fungsi-fungsi untuk kuadran-kuadran diantara dua titik-titik.

Dalam contoh berikut, karena  $c+b$  bukan  $a$ , segitiga bukan siku-siku.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

17

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode umum berdasarkan hasil-dot dari dua vektor-vektor. Hasilnya beberapa titik aproksimasi.

$$A = \langle 1, 2 \rangle \quad B = \langle 4, 3 \rangle, \quad C = \langle 0, 4 \rangle$$

$$\mathbf{a} = C - B = \langle -4, 1 \rangle, \quad \mathbf{c} = A - B = \langle -3, -1 \rangle, \quad \beta = \angle ABC$$

$$\mathbf{a} \cdot \mathbf{c} = |\mathbf{a}| \cdot |\mathbf{c}| \cos \beta$$

$$\cos \angle ABC = \cos \beta = \frac{\mathbf{a} \cdot \mathbf{c}}{|\mathbf{a}| \cdot |\mathbf{c}|} = \frac{12 - 1}{\sqrt{17} \sqrt{10}} = \frac{11}{\sqrt{17} \sqrt{10}}$$

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

$$\arccos\left(\frac{11}{\sqrt{10}\sqrt{17}}\right)$$

32.4711922908

Menggunakan pensil dan kertas, kami dapat melakukan hal yang sama dengan aturan cross. Kami memasukkan kuadran-kuadran a,b,dan c kedalam aturan crosss dan menyelesaikan x.

```
>$crosslaw(a,b,c,x), $solve(% ,x), // (b+c-a)^=4b.c(1-x)
```

$$\left[ x = \frac{49}{50} \right]$$

$$\left[ x = \frac{49}{50} \right]$$

Bahwa, apa fungsi sebar didefinisikan dalam "geometry.e" lakukan.

```
>sb &= spread(b,a,c); $sb
```

$$\frac{49}{170}$$

Maxima mendapatkan hasil yang sama menggunakan trigonometri umum, jika kita paksakannya. Ini dapat menyelesaikan  $\sin(\arccos(...))$  untuk hasil pecahan. Banyak siswa tidak dapat melakukan ini.

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{49}{170}$$

Selanjutnya, kita miliki sebaran pada B, kami dapat mengitung tinggi ha pada sisi a. Ingat bahwa

$$s_b = \frac{h_a}{c}$$

secara definisi.

```
>ha &= c*sb; $ha
```

$$\frac{49}{17}$$

Gambar berikut telah dihasilkan dengan program geometri C.a.R., yang mana menggambarkan kuadran-kuadran dan sebaran-sebaran.

image: (20) Rational\_Geometry\_CaR.png

Secara definisi panjang dari ha adalah akar kuadrat dari kuadran.

```
>$sqrt(ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang kami dapat menghitung luas dari segitiga. Jangan lupa, bahwa kami berurusan dengan kuadran-kuadran.

```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{7}{2}$$

Formula determinan biasa menghasilkan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

$$\frac{7}{2}$$

Sekarang, mari kita selesaikan masalah ini dalam umum!

```
>&remvalue(a,b,c,sb,ha);
```

Pertama kami hitung sebaran pada B untuk sebuah segitiga dengan sisi-sisi a,b dan c. Lantas kami hitung luasan persegi ("kuadrea"?), faktorkan itu dengan Maxima, dan kami dapatkan formula terkenal dari Heron.

Memang, ini susah untuk dilakukan dengan pensil dan kertas.

```
>$spread(b^2,c^2,a^2), $factor(%*c^2*a^2/4)
```

$$\frac{(-c + b + a) (c - b + a) (c + b - a) (c + b + a)}{16}$$

$$\frac{(-c + b + a) (c - b + a) (c + b - a) (c + b + a)}{16}$$

---

## Aturan Sebaran Tiga Kali Lipat

Kerugian dari sebaran adalah mereka tidak lagi secara sederhana menambahkan sudut-sudut. Namun, sebaran-sebaran tiga dari segitiga memenuhi aturan "triple spread" berikut.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

$$(sc + sb + sa)^2 = 2 (sc^2 + sb^2 + sa^2) + 4 sa sb sc$$

Aturan ini valid untuk setiap tiga sudut bahwa menambahkan ke  $180^\circ$ .

$$\alpha + \beta + \gamma = \pi$$

Karena sebaran dari

$$\alpha, \pi - \alpha$$

adalah sama, aturan triple spread juga benar, jika

$$\alpha + \beta = \gamma$$

Karena sebaran adalah sudut negatif adalah sama, aturan triple spread juga terpenuhi, jika

$$\alpha + \beta + \gamma = 0$$

Untuk contoh, kami dapat mengitung sebaran dari sudut  $60^\circ$ . Ini adalah  $3/4$ . Persamaan-persamaan memiliki solusi kedua, bagaimanapun, jika semua menyebar adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

$$\left[ x = \frac{3}{4}, x = 0 \right]$$

Sebaran dari  $90^\circ$  secara jelas adalah 1. Jika dua sudut ditambahkan  $90^\circ$ , mereka menyebar menyelesaikan persamaan triple spread dengan a,b,l. Dengan mengikuti perhitungan kami dapatkan  $a+b=l$ .

```
>$triplespread(x,y,1), $solve(% ,x)
```

$$[x = 1 - y]$$

Karena sebaran dari  $180^\circ - t$  adalah sama seperti sebaran dari  $t$ , formula triple spread juga demikian, jika satu sudut adalah jumlahan atau beda dari dua sudut lainnya.

Jadi kami temukan sebaran dari sudut berlipat ganda. Catatan bahwa ini terdapat dua solusi kembali. Kita buat ini adalah sebuah fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))
```

$$[x = 4a - 4a^2, x = 0]$$

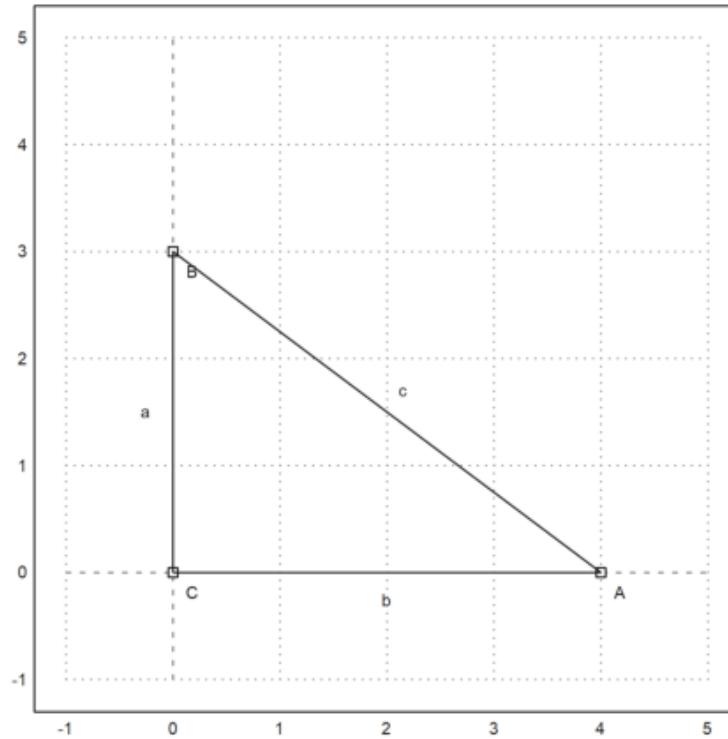
$$- 4(a - 1)a$$

## Sudut Bisektor

---

Ini merupakan situasi, kita telah ketahui,

```
>C:=[0,0]; A:=[4,0]; B:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```



Mari kita hitung panjang dari sudut bisektor pada A. Tapi kami ingin menyelesaikan bahwa untuk umum a,b,c.

```
>&remvalue(a,b,c);
```

Jadi pertama kami hitung sebaran dari sudut bisected pada A, menggunakan formula triple spread.

Masalah dengan formula ini terlihat kembali. ini memiliki dua solusi. Kami ambil salah satunya. Solusi lainnya merujuk pada sudut bisected  $180^\circ$ -wa.

```
>$triplespread(x,x,a/(a+b)), $solve(%,x), sa2 &= rhs(%[1]); $sa2
```

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

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

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

Mari kita periksa untuk persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

Kami dapat mencetak sudut dalam Euler, setelah mentransfer sebaran ke radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

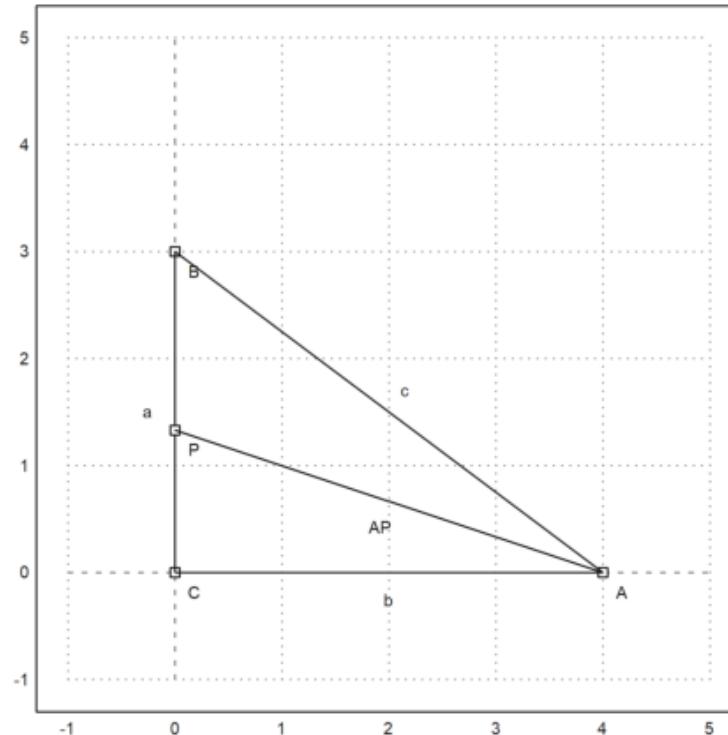
$18^\circ 26' 5.82''$

Titik P adalah irisan dari bisector sudut dengan sumbu-y

```
>P := [0,tan(wa2)*4]
```

[0, 1.33333]

```
>plotPoint(P,"P"); plotSegment(A,P):
```



Mari kita periksa dalam contoh kita spesifik

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

0.321750554397  
0.321750554397

Sekarang kita hitung panjang dari Bisector AP

Kami gunakan teorema sine dalam segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

untuk sebarang segitiga. Kuadratkan, ini mengubahnya ke "spread law"

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

dimana a,b,c dinotasikan kuadran

Karena sebaran CPA adalah  $1-sa^2$  kami dapatkan itu dari  $bisa/1=b/(1-sa^2)$  dan dapat mengitung bisa (kuadran dari sudut bisektor)

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$\frac{2 b (b + a)}{\sqrt{b} \sqrt{b + a} + b + a}$$

Mari kita periksa formula ini untuk nilai-nilai Mesir.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])")), distance(A,P)
```

4.21637021356

4.21637021356

Kami dapat juga menghitung P menggunakan formula sebaran.

```
>py&=factor(ratsimp(sa2*bisa)); $py
```

$$-\frac{b \left(\sqrt{b} \sqrt{b+a}-b-a\right)}{\sqrt{b} \sqrt{b+a}+b+a}$$

Nilai adalah sama yang kita dapatkan dengan formula trigonometrik

```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

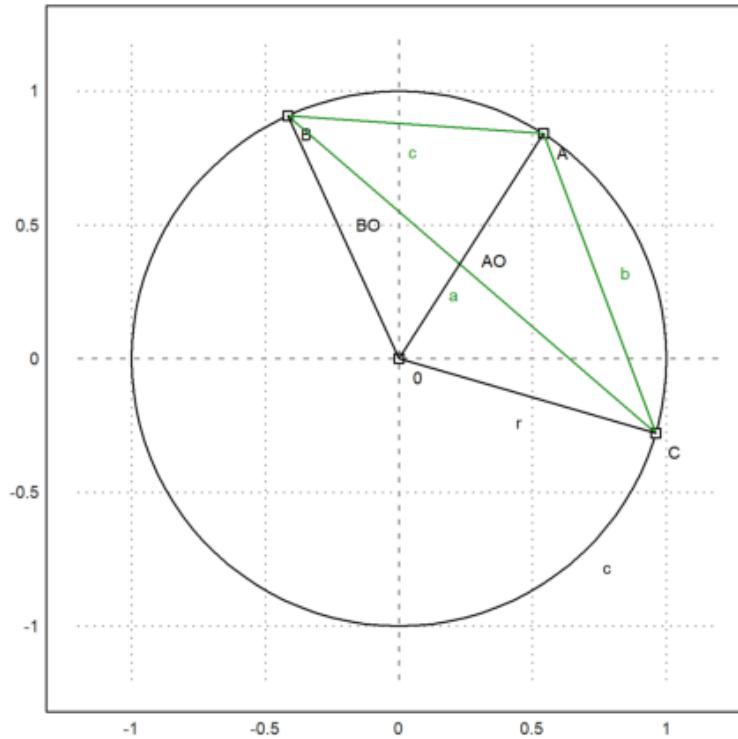
1.33333333333

**Sudut Tali Busur**

---

Lihat situasi berikut.

```
>setPlotRange(1.2); ...
>color(1); plotCircle(circleWithCenter([0,0],1)); ...
>A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>color(1); O:=[0,0]; plotPoint(O,"O"); ...
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
>insimg;
```



Kami dapat menggunakan Maxima untuk menyelesaikan formula spread triple untuk sudut-sudut pada tengah 0 untuk  $r$ . Lalu, kita dapatkan sebuah formula untuk jari-jari kuadratik dari lingkaran luar dalam istilah dari kuadran dari sisi-sisi.

Waktu ini, Maxima menghasilkan beberapa nol-nol kompleks, yang mana dapat kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru  
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); $rabc
```

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

Kami dapat membuat sebuah fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasil untuk titik-titik kami A,B,C

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Jari-jari adalah 1

```
>periradius(a,b,c)
```

Faktanya adalah, bahwa sebaran CBA tergantung hanya pada b dan c. Ini adalah teorema tali busur.

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Fakta bahwa sebaran  $b/(4r)$ , dan kami lihat bahwa tali busur dari busur b adalah setengahnya sudut pusat.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

$$0$$

## Contoh 6: Jarak Minimal pada Bidang

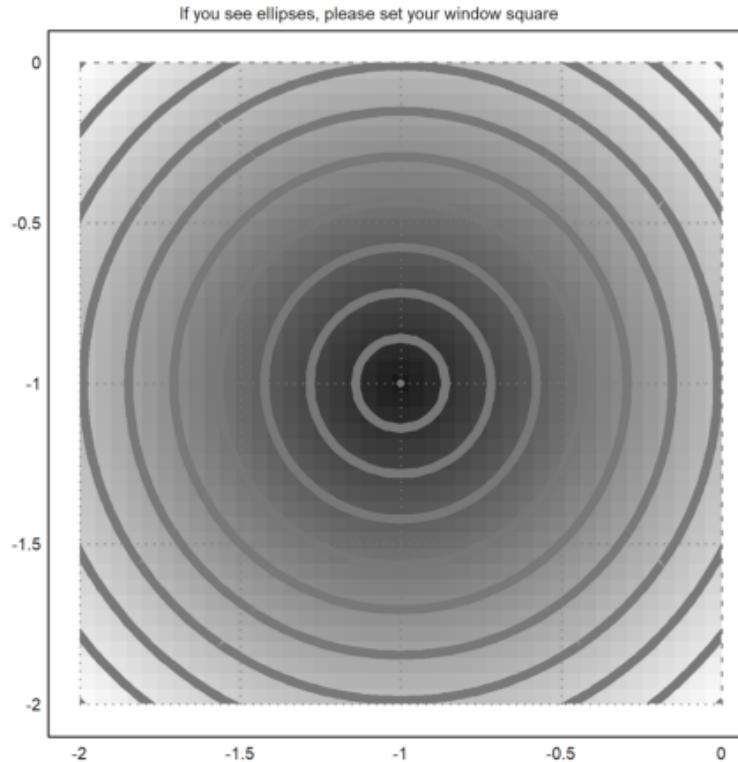
---

### Preliminary remark

---

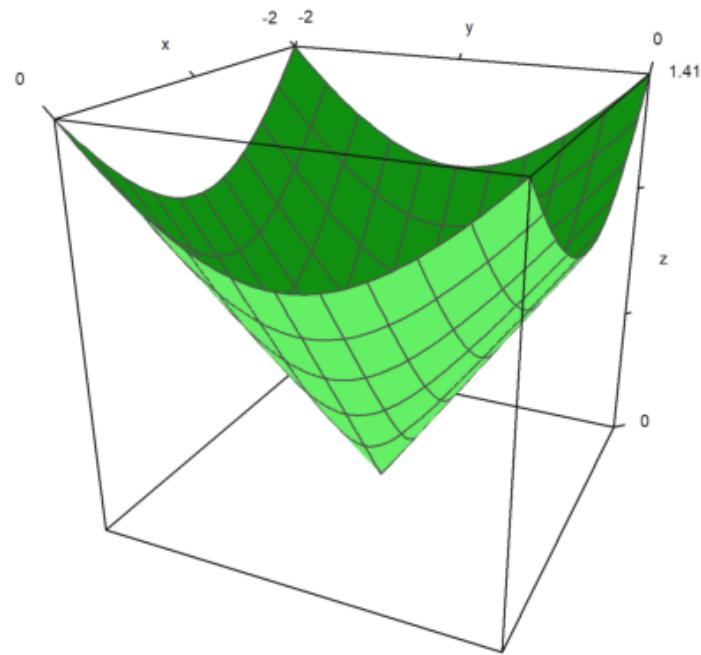
The function which, to a point M in the plane, assigns the distance AM between a fixed point A and M, has rather simple level lines: circles centered in A.

```
>&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");
```



and the graph is rather simple too: the upper part of a cone:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```

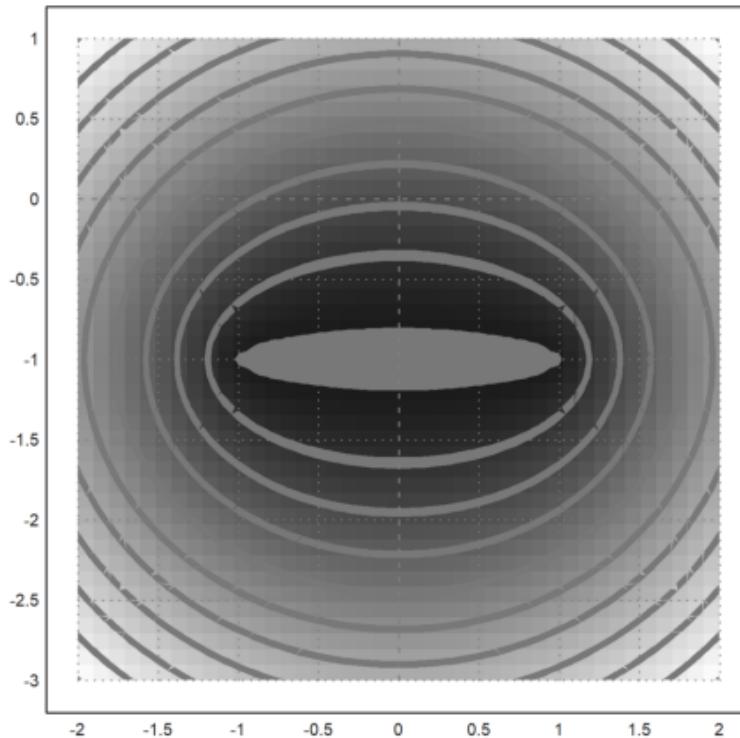


Of course the minimum 0 is attained in A.

Two points

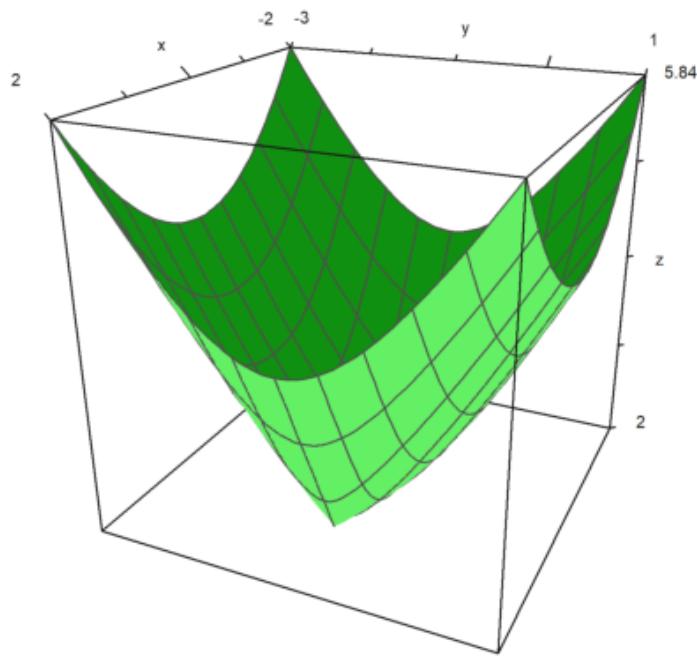
Now we look at the function  $MA+MB$  where A and B are two points (fixed). It is a "well-known fact" that the level curves are ellipses, the focal points being A and B; except for the minimum  $AB$  which is constant on the segment [AB]:

```
>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):
```



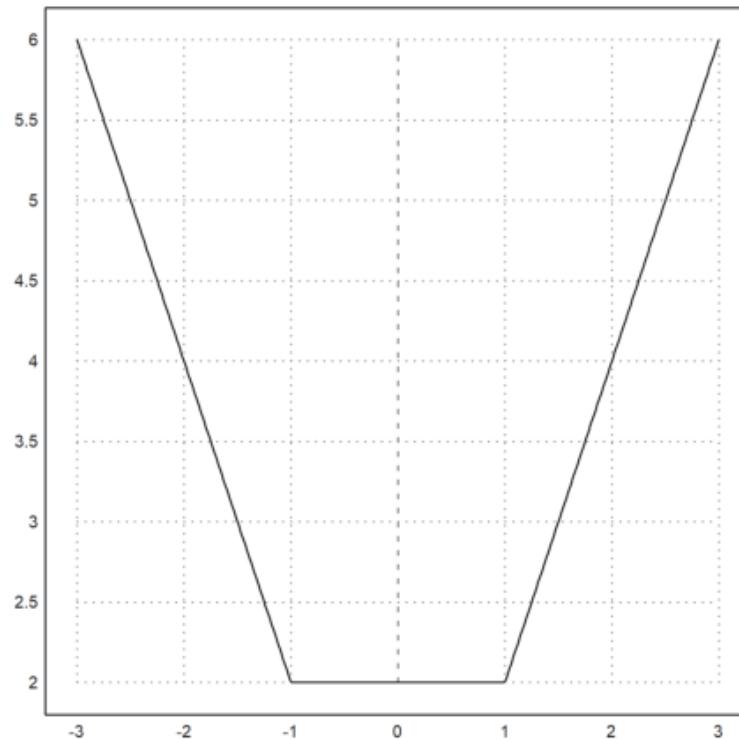
The graph is more interesting:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



The restriction to line (AB) is more famous:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

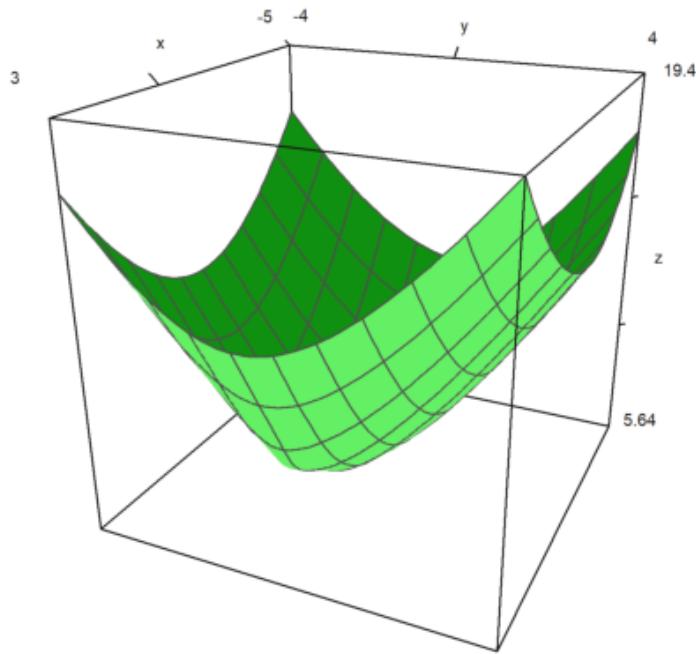


Now things are less simple: It is a little less well-known that  $MA+MB+MC$  attains its minimum at one point of the plane but to determine it is less simple:

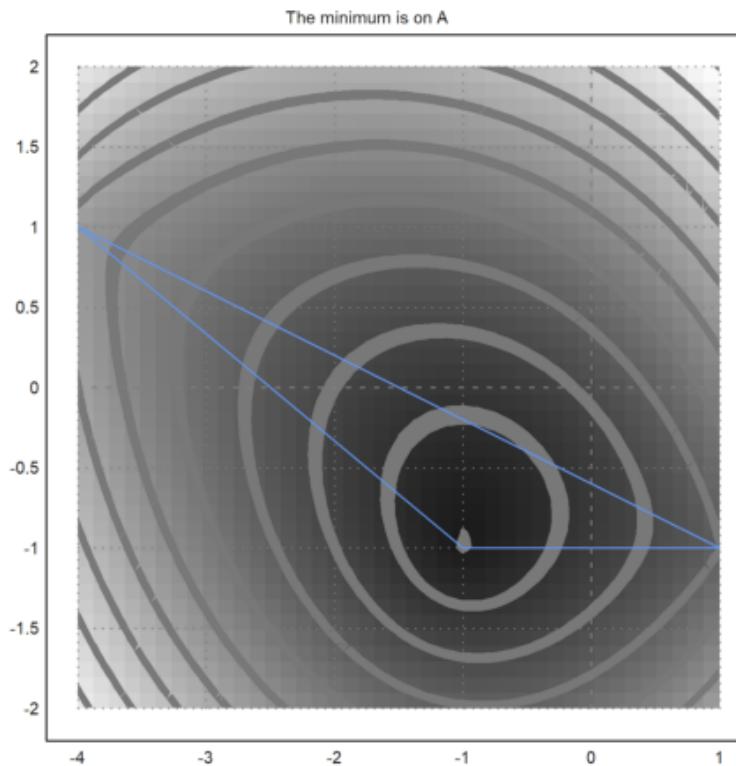
- 1) If one of the angles of the triangle ABC is more than  $120^\circ$  (say in A), then the minimum is attained at this very point (say  $AB+AC$ ).

Example:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
>insimg;
```

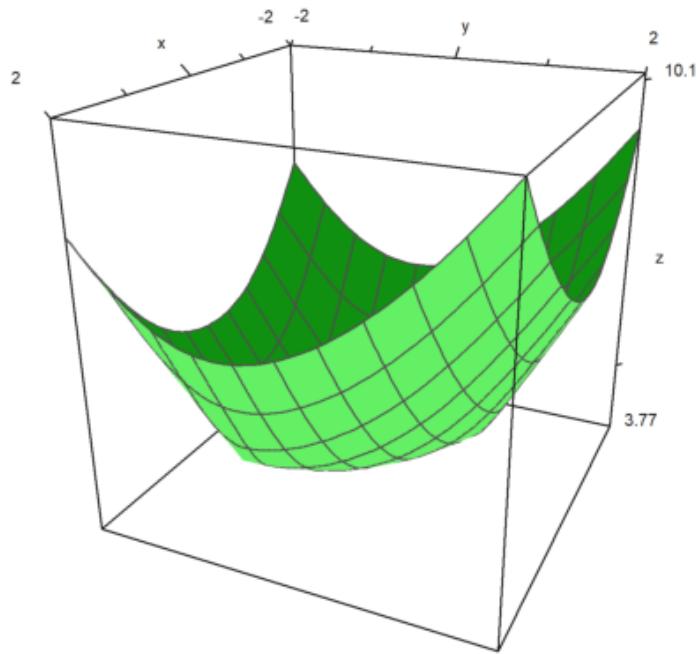


```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

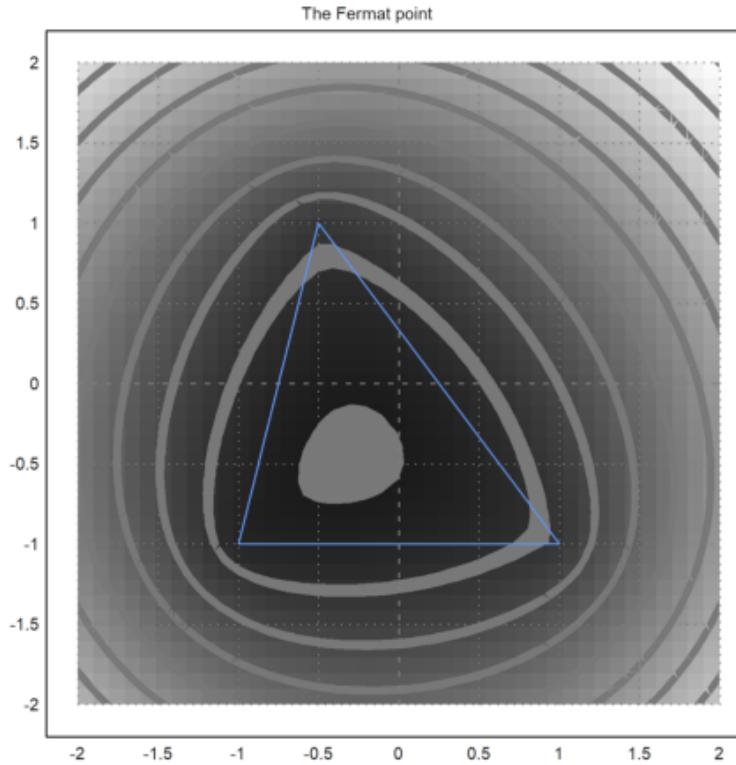


- 2) But if all the angles of triangle ABC are less than  $120^\circ$ , the minimum is on a point F in the interior of the triangle, which is the only point which sees the sides of ABC with the same angles (then  $120^\circ$  each):

```
>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;
```

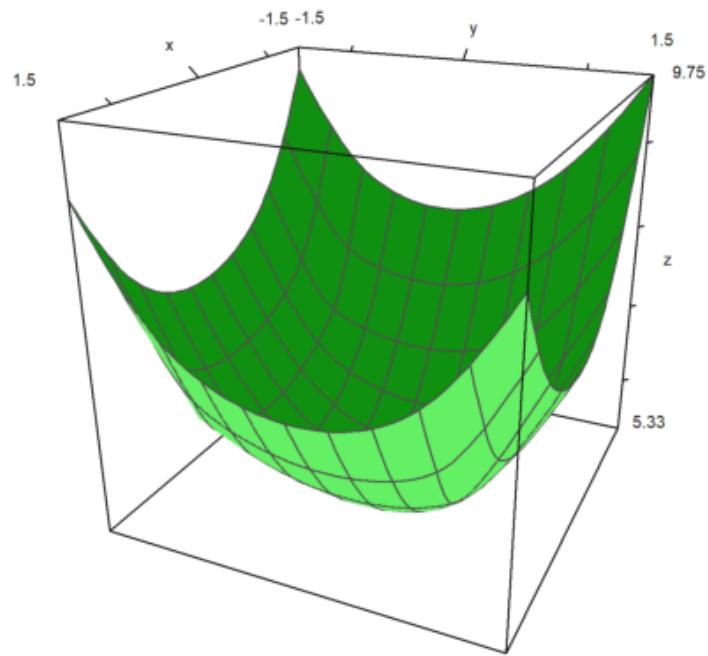


It is an interesting activity to realize the above figure with a geometry software; for example, I know a soft written in Java which has a "contour lines" instruction...

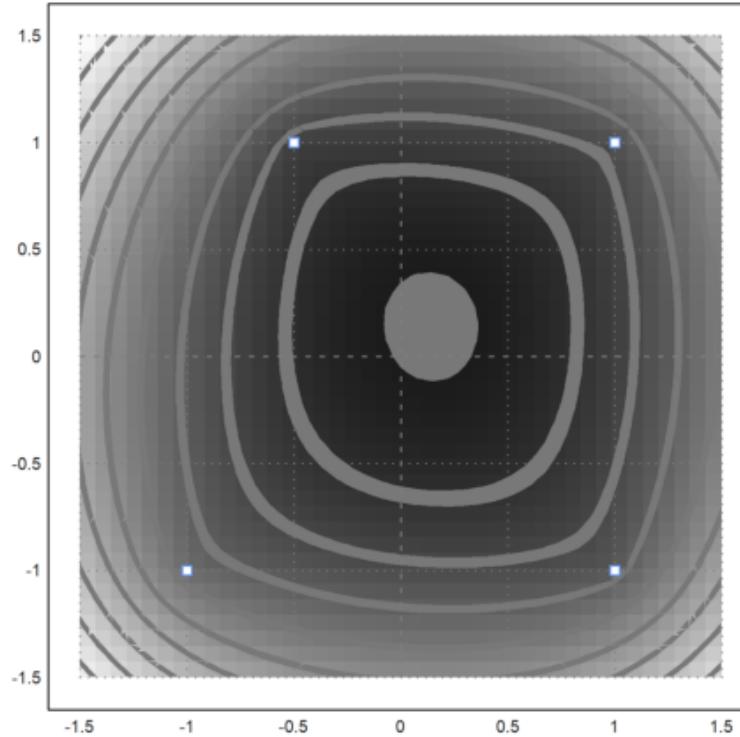
All of this above have been discovered by a french judge called Pierre de Fermat; he wrote letters to other dilettants like the priest Marin Mersenne and Blaise Pascal who worked at the income taxes. So the unique point  $F$  such that  $FA+FB+FC$  is minimal, is called the Fermat point of the triangle. But it seems that a few years before, the italian Torricelli had found this point before Fermat did! Anyway the tradition is to note this point  $F$ ...

The next step is to add a 4th point D and try and minimize MA+MB+MC+MD; say that you are a cable TV operator and want to find in which field you must put your antenna so that you can feed four villages and use as little cable length as possible!

```
>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;
```



There is still a minimum and it is attained at none of the vertices A, B, C nor D:

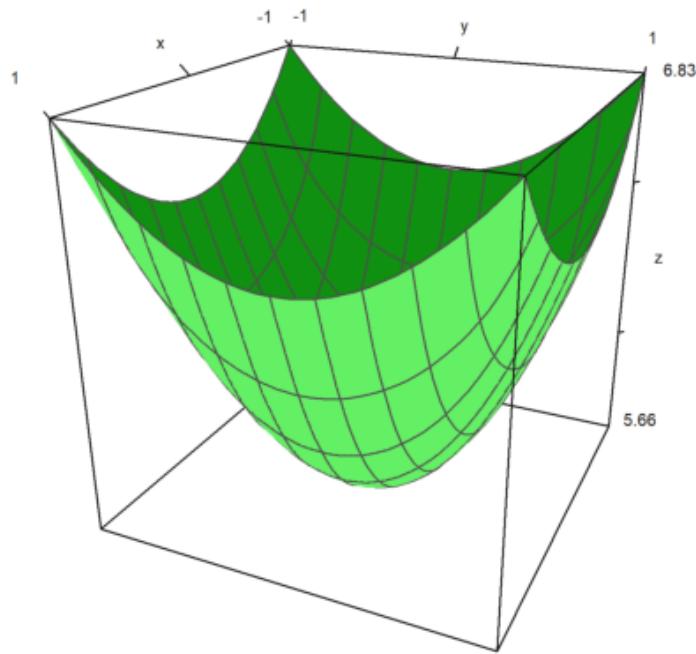
```
>function f(x):=d4(x[1],x[2])
>neldermin("f",[0.2,0.2])
```

```
[0.142858, 0.142857]
```

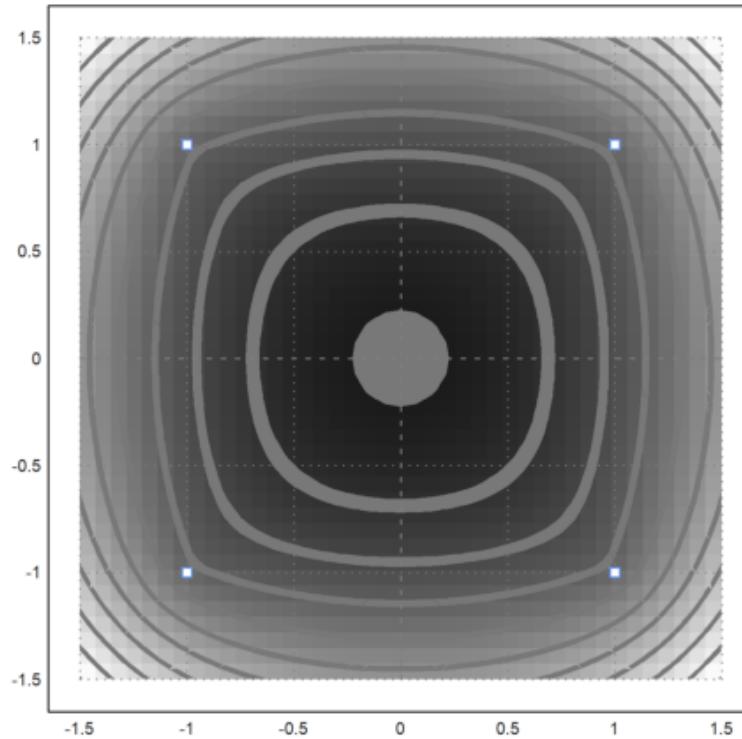
It seems that in this case, the coordinates of the optimal point are rational or near-rational...

Now ABCD is a square we expect that the optimal point will be the center of 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

---

You can run this demonstration, if you have Povray installed, and pvengine.exe in the program path.

First we compute the radii of the spheres.

If you look at the figure below, you see that we need two circles touching the two lines which form the cone, and one line which forms the plane cutting the cone.

We use the geometry.e file of Euler for this.

```
>load geometry;
```

First the two lines forming the cone.

```
>g1 &= lineThrough([0,0],[1,a])
```

```
[- a, 1, 0]
```

```
>g2 &= lineThrough([0,0],[-1,a])
```

```
[ - a, - 1, 0]
```

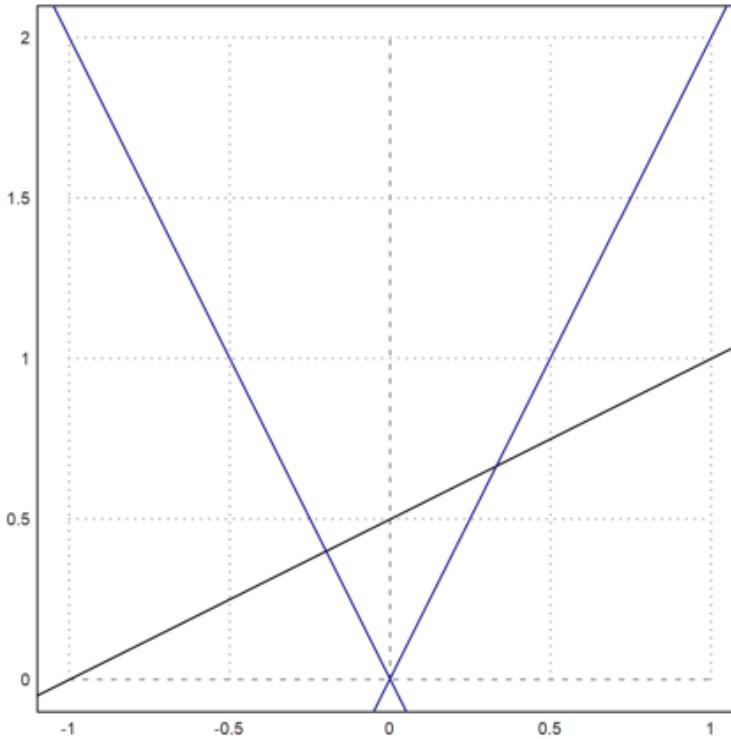
Then m a third line.

```
>g &= lineThrough([-1,0],[1,1])
```

```
[ - 1, 2, 1]
```

We plot everything so far.

```
>setPlotRange(-1,1,0,2);
>color(black); plotLine(g(), "")
>a:=2; color(blue); plotLine(g1(), ""), plotLine(g2(), ""):
```



Now we take a general point on the y-axis.

```
>P &= [0,u]
```

```
[0, u]
```

Compute the distance to g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

$$\sqrt{\left(\frac{a^2 u}{a^2 + 1} - u\right)^2 + \frac{a^2 u^2}{(a^2 + 1)^2}}$$

Compute the distance to g.

```
>d &= distance(P,projectToLine(P,g)); $d
```

$$\sqrt{\left(\frac{u + 2}{5} - u\right)^2 + \frac{(2 u - 1)^2}{25}}$$

And find the centers of the two circles, where the distances are equal.

```
>sol &= solve(d1^2=d^2,u); $sol
```

$$\left[ u = \frac{-\sqrt{5} \sqrt{a^2 + 1} + 2 a^2 + 2}{4 a^2 - 1}, u = \frac{\sqrt{5} \sqrt{a^2 + 1} + 2 a^2 + 2}{4 a^2 - 1} \right]$$

There are two solutions.

We evaluate the symbolic solutions, and find both centers, and both distances.

```
>u := sol()
```

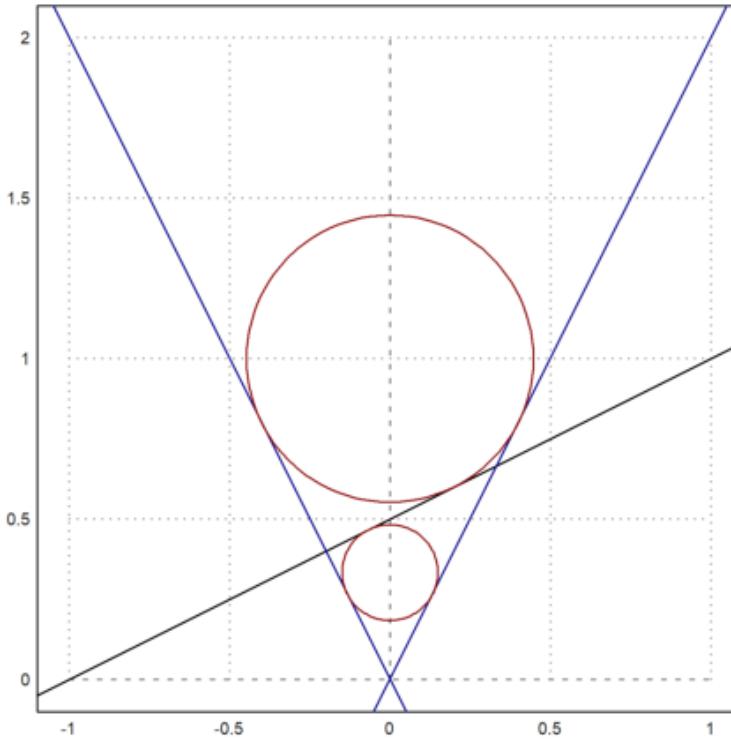
```
[0.333333, 1]
```

```
>dd := d()
```

```
[0.149071, 0.447214]
```

Plot the circles into the figure.

```
>color(red);
>plotCircle(circleWithCenter([0,u[1]],dd[1]), "");
>plotCircle(circleWithCenter([0,u[2]],dd[2]), "");
>insimg;
```



Plot with Povray

---

Next we plot everything with Povray. Note that you change any command in the following sequence of Povray commands, and rerun all commands with Shift-Return.

First we load the povray functions.

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe

We setup the scene appropriately.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Next we write the two spheres to the Povray file.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

And the cone, transparent.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

We generate a plane restricted to the cone.

```
>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));
```

Now we generate two points on the circles, where the spheres touch the cone.

```
>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)));
```

Then we generate the two points where the spheres touch the plane. These are the foci of the ellipse.

```
>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))));
```

Next we compute the intersection of P1P2 with the plane.

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
>writeln(povpoint(P5,povlook(yellow))));
```

We connect the points with line segments.

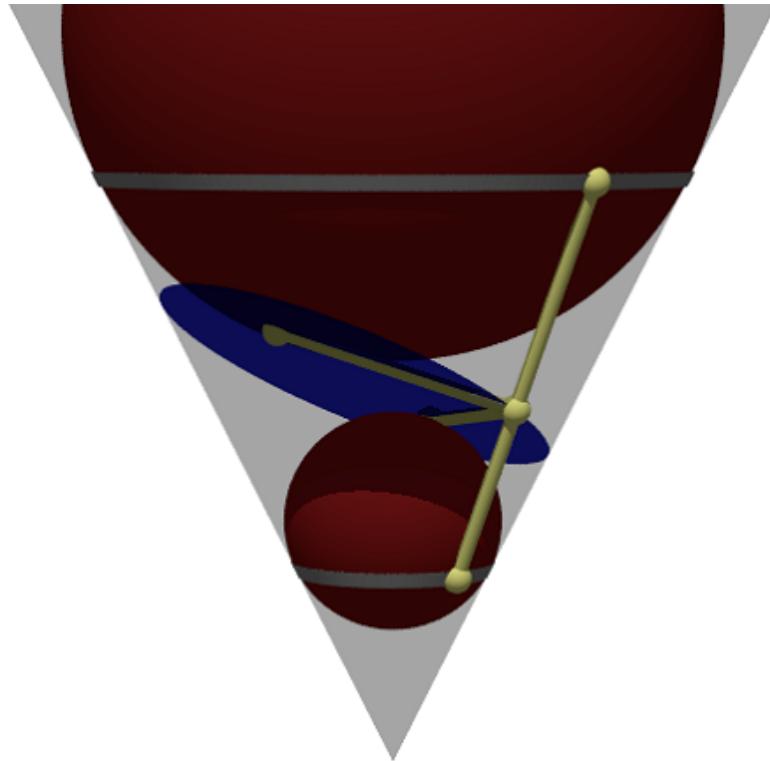
```
>writeln(povsegment(P1,P2,povlook(yellow)));
>writeln(povsegment(P5,P3,povlook(yellow)));
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Now we generate a gray band, where the spheres touch the cone.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);
>pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
>writeln(povintersection([pcw,pc1],povlook(gray)));
>pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
>writeln(povintersection([pcw,pc2],povlook(gray))));
```

Start the Povray program.

```
>povend();
```



To get an Anaglyph of this we need to put everything into a scene function. This function will be used twice later.

```
>function scene () ...
```

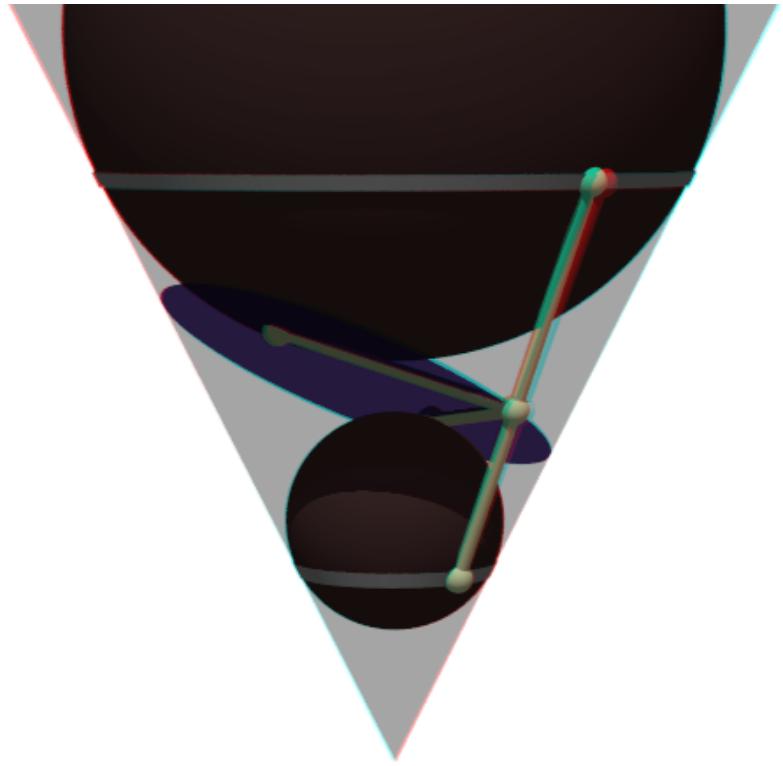
```

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=turnnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow)));
writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow)));
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc1],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction

```

You need red/cyan glasses to appreciate the following effect.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```



## Contoh 8: Geometri Bumi

---

In this notebook, we want to do some spherical computations. The functions are contained in the file "spherical.e" in the examples folder. We need to load that file first.

```
>load "spherical.e";
```

To enter a geographical position, we use a vector with two coordinates in radians (north and east, negative values for south and west). The following are the coordinates for the Campus of the FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569, 1.92657]
```

You can print this position with sposprint (spherical position print).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
S 7°46.467' E 110°23.050'
```

Let us add two more towns, Solo and 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'
```

First we compute the vector from one to the other on an ideal ball. This vector is [heading,distance] in radians. To compute the distance on the earth, we multiply with the earth radius at a latitude of  $7^\circ$ .

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan jarak FMIPA-Solo
```

```
65°20'26.60''  
53.8945384608
```

This is a good approximation. The following routines use even better approximations. On such a short distance the result is almost the same.

```
>esdist(FMIPA,Semarang)->" km" // perkiraan jarak FMIPA-Semarang
```

```
Commands must be separated by semicolon or comma!
Found: // perkiraan jarak FMIPA-Semarang (character 32)
You can disable this in the Options menu.
Error in:
esdist(FMIPA,Semarang)->" km" // perkiraan jarak FMIPA-Semaran ...
```

There is a function for the heading, taking the elliptical shape of the earth into account. Again, we print in an advanced way.

```
>sdegprint(esdir(FMIPA,Solo))
```

65.34°

The angle of a triangle exceeds 180° on the sphere.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,Semarang,Solo); degprint(
```

180°0'10.77''

This can be used to compute the area of the triangle. Note: For small triangles, this is not accurate due to the subtraction error in `asum-pi`.

```
>(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FMIPA-Solo-Semarang
```

```
Commands must be separated by semicolon or comma!
Found: // perkiraan luas segitiga FMIPA-Solo-Semarang (character 32)
You can disable this in the Options menu.
Error in:
(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FM ...
```

There is a function for this, which uses the mean latitude of the triangle to compute the earth radius, and takes care of rounding errors for very small triangles.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi esarea()
```

2123.64310526 km<sup>2</sup>

We can also add vectors to positions. A vector contains the heading and the distance, both in radians. To get a vector, we use svector. To add a vector to a position, we use saddvector.

```
>v=svector(FMIPA,Solo); sposprint(saddvector(FMIPA,v)), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'  
S 7°34.333' E 110°49.683'
```

These functions assume an ideal ball. The same on the earth.

```
>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'
```

Let us turn to a larger example, 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'
```

According to Google Earth, the distance is 429.66km. We get a good approximation.

```
>esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Monas Jakarta
```

```
Commands must be separated by semicolon or comma!
Found: // perkiraan jarak Tugu Jogja - Monas Jakarta (character 32)
You can disable this in the Options menu.
Error in:
esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Mona ...
```

The heading is the same as the one computed in Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

294°17'2.85''

However, we do no longer get the exact target position, if we add the heading and distance to the orginal position. This is so, since we do not compute the inverse function exactly, but take an approximation of the earth radius along the path.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

S 6°10.500' E 106°48.717'

The error is not large, however.

```
>sposprint(Monas),
```

S 6°10.500' E 106°48.717'

Of course, we cannot sail with the same heading from one destination to another, if we want to take the shortest path. Imagine, you fly NE starting at any point on the earth. Then you will spiral to the north pole. Great circles do not follow a constant heading!

The following computation shows that we are way off the correct destination, if we use the same heading during our travel.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Now we add 10 times one-tenth of the distance, using the heading to Monas, we got in Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

The result is far off.

```
>sposprint(p), skmpprint(esdist(p,Monas))
```

S 6°11.250' E 106°48.372'  
1.529km

As another example, let us take two points on the earth at the same lattitude.

```
>P1=[30°,10°]; P2=[30°,50°];
```

The shortest path from P1 to P2 is not the circle of lattitude 30°, but a shorter path starting 10° further north at P1.

```
>sdegprint(esdir(P1,P2))
```

79.69°

But, if we follow this compass reading, we will spiral to the north pole! So we must adjust our heading along the way. For rough purposes, we adjust it at 1/10 of the total distance.

```
>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°

The distances are not right, since we will add a bit off error, if we follow the same heading for too long.

```
>skmprint(esdist(p,P2))
```

0.203km

We get a good approximation, if we adjust our heading after each 1/100 of the total distance from Tugu to 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

For navigational purposes, we can get a sequence of GPS position along the great circle to Monas with the function `navigate`.

```
>load spherical; v=navigate(Tugu,Monas,10); ...
> loop 1 to rows(v); sposprint(v[#]), end;
```

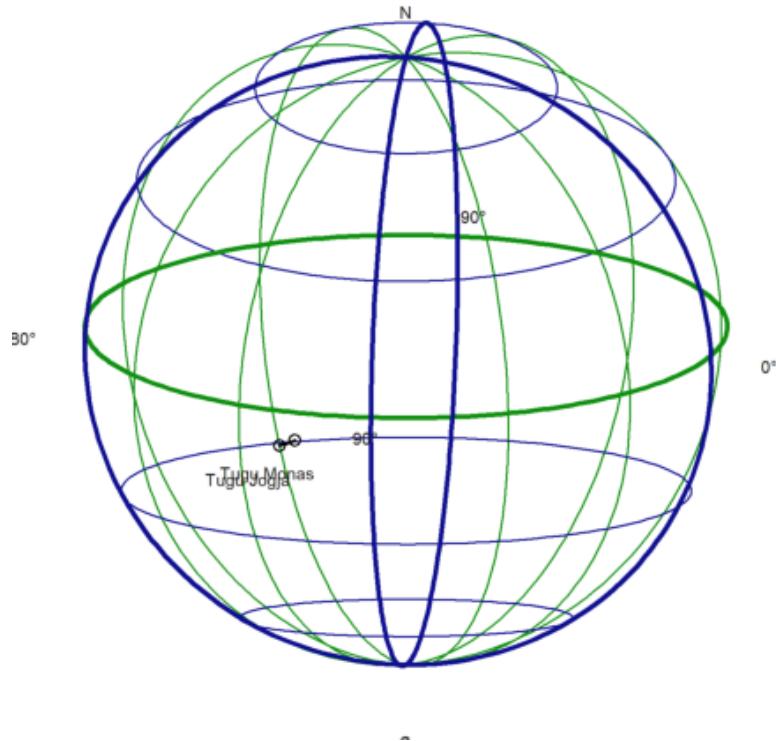
S 7°46.998' E 110°21.966'  
S 7°37.422' E 110°0.573'  
S 7°27.829' E 109°39.196'  
S 7°18.219' E 109°17.834'  
S 7°8.592' E 108°56.488'  
S 6°58.948' E 108°35.157'  
S 6°49.289' E 108°13.841'  
S 6°39.614' E 107°52.539'  
S 6°29.924' E 107°31.251'  
S 6°20.219' E 107°9.977'  
S 6°10.500' E 106°48.717'

We write a function, which plots the earth, the two positions, and the positions in between.

```
>function testplot ...  
  
useglobal;  
plotearth;  
plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");  
plotposline(v);  
endfunction
```

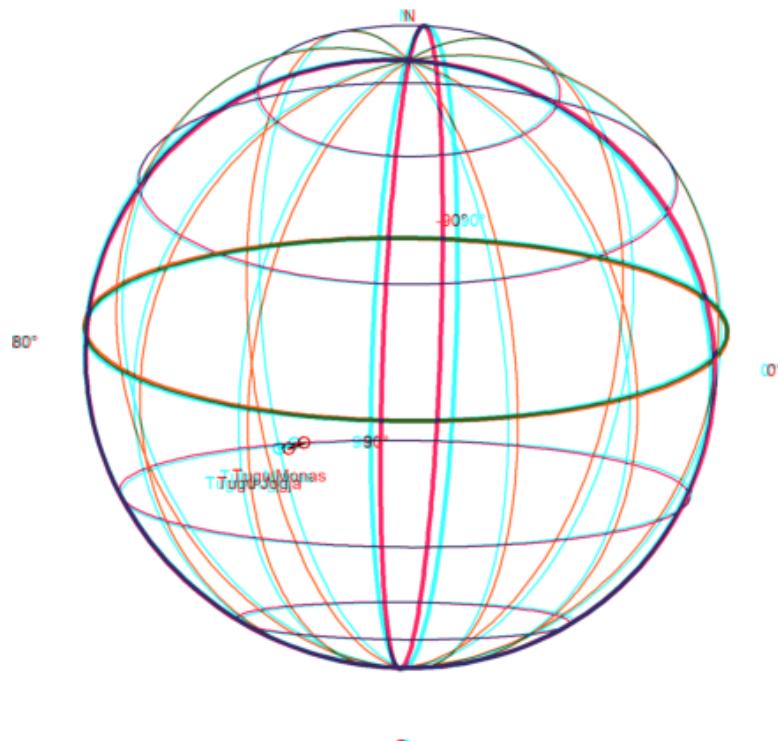
Now plot everything.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```



Or use `plot3d` to get an anaglyph view of it. This looks really great with red/cyan glasses.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```



## Latihan

---

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah  $(360/n)$ .
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan  $(360/n)$ .
- Untuk n ganjil, pilih salah satu titik sudut adalah di atas.
- Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat.
- Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:

- Misalkan persamaan parabolanya  $y = ax^2 + bx + c$ .
- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

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-sisintya

merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).

- Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya

bertemu di satu titik.

- Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya.
- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali

panjang sisi-sisi yang berhadapan sama.

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).
5. Gambarlah suatu 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).

## Chapter 6

# Statistika pada EMT

23030630053  
Hamemayu Hayuningrat

**EMT untuk Statistika**

---

Dalam notebook ini, kami akan mendemonstrasikan plot utama statistika, uji dan distribusi dalam Euler. Mari kita mulai dengan beberapa deskriptif statistik. Ini bukanlah pengantar untuk statistik. Jadi kamu mungkin membutuhkan beberapa latar belakang untuk memahami secara detail.

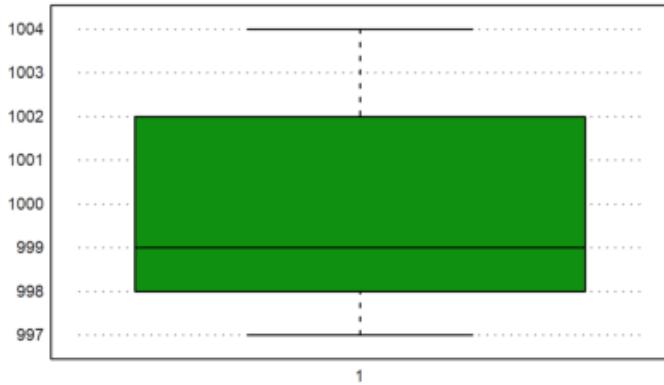
Asumsikan ukuran mengikuti berikut. Kita berharap untuk menghitung nilai rata-rata dan menghitung standar deviasi.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...
>median(M), mean(M), dev(M),
```

```
999
999.9
2.72641400622
```

Kami dapat membuat plot box-and-whiskers untuk data. Dalam kasus kami dimana tidak ada data yang keluar.

```
>aspect(1.75); boxplot(M):
```



Kami menghitung kemungkinan bahwa sebuah nilai lebih dari 1005, asumsikan hitungan nilai dari sebuah distribusi normal.

Semua fungsi untuk distibusi-distribusi dalam Euler dan dengan ...dis dan menghitung cumulative probability distribution (CPF)

$$\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-m}{d}\right)^2} dt.$$

Kami mencetak hasil dalam % dengan akurasi 2 digit menggunakan fungsi print.

```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
```

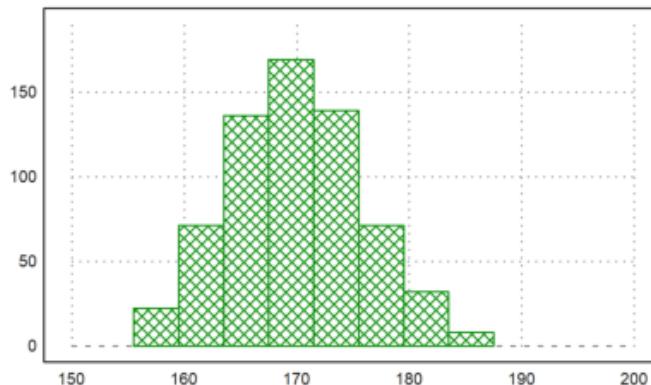
3.07 %

Untuk contoh selanjutnya, kami asumsi angka-angka berikut dari pria dengan diberikan sebuah rentang ukuran.

```
>r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Ini merupakan sebuah plot distribusi.

```
>plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="\\"":
```



Kami dapat mengambil sebarang data mentah ke sebuah tabel.

Tabels merupakan sebuah method untuk menyimpan data statistik. Tabel kami seharusnya mengandung tiga kolom: rentang mulai, rentang akhir, banyaknya pria dalam rentang.

Tables dapat juga dicetak dengan headers. Kami menggunakan vektor string untuk mengatur headers.

```
>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 kami menggunakan nilai rata-rata dan statistik lain dari ukuran, kami membutuhkan untuk menghitung titik tengah dari rentang. Kami dapat menggunakan kolom kedua pertama dari tabel kami untuk ini.

Simbol "|" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan pilihan "labc" adalah secara spesifik kolom headers.

```
>(T[,1]+T[,2])/2 // the midpoint of each interval
```

```
157.5  
161.5  
165.5  
169.5  
173.5  
177.5  
181.5  
185.5
```

Tetapi ini lebih mudah, untuk melipat rentang dengan vektor  $[1/2, 1/2]$ .

```
>M=fold(r,[0.5,0.5])
```

```
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
```

Sekarang kita menghitung rata-rata dan deviasi dari contoh dengan frekuensi yang diberikan.

```
>{m,d}=meandev(M,v); m, d,
```

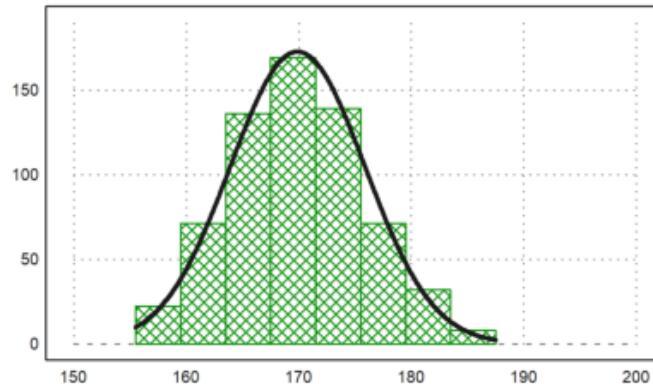
```
169.901234568  
5.98912964449
```

Mari kita tambahkan distribusi normal dari nilai nilai menjadi plot batang diatas. Formula untuk distribusi normal dengan rata-rata m dan standar deviasi d adalah:

$$y = \frac{1}{d\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2d^2}}.$$

Karena ini nilai-nilai diantara 0 dan 1, untuk membuat plot itu dalam plot batang kita harus mengalikan dengan 4 kali jumlahan angka dari data.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...
> xmin=min(r),xmax=max(r),thickness=3,add=1):
```



## Tables

---

Dalam direktori dari notebook ini kamu menemukan sebuah file dengan sebuah tabel. Data tersebut menunjukkan hasil dari sebuah survei. Disini merupakan empat baris pertama dari file. Data berasal dari sebuah buku online Jerman "Einführung in die Statistik mit R" oleh A. Handl.

```
>printfile("table.dat",4);
```

```
Person Sex Age Titanic Evaluation Tip Problem
1 m 30 n . 1.80 n
2 f 23 y g 1.80 n
3 f 26 y g 1.80 y
```

Tabel berisi 7 kolom dari angka-angka atau token (string). Kami ingin membaca tabel dari file. Pertama, kami menggunakan translasi sendiri untuk tokens.

Untuk ini, kami mendefinisikan himpunan dari token. Fungsi `strtokens()` mengambil sebuah vektor string dari token dari sebuah string yang diberikan.

```
>mf := ["m", "f"]; yn := ["y", "n"]; ev := strtokens("g vg m b vb");
```

Sekarang kita membaca tabel dengan translasi berikut.

Argumen-argumen tok2, tok4, dll merupakan translasi dari kolom tabel. Argumen-argumen ini tidak dalam daftar parameter dari readtable(), jadi kamu butuh untuk menyediakan mereka dengan ":=".

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);  
>load over statistics;
```

Untuk mencetak, kami membutuhkan secara spesifik himpunan token yang sama. Kami mencetak empat baris pertama saja.

```
>writetable(MT[1:10],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n

Titik ”.” mewakili nilai, yang mana tidak tersedia.

Jika kita tidak ingin untuk secara spesifik token untuk translasi lebih lanjut, kami hanya butuh untuk secara spesifik, yang mana kolom berisi token dan bukan angka-angka.

```
>ctok=[2,4,5,7]; {MT,hd,tok}=readtable("table.dat",ctok=ctok);
```

Sekarang fungsi readtable() mengembalikan sebuah himpunan dari token.

```
>tok
```

```
m  
n  
f  
y  
g  
vg
```

Tabel berisi isian dari file dengan tokens yang diubah ke angka.

String spesial NA=”..” diinterpretasikan sebagai ”Not Available” dan menghasilkan NAN (Not A Number) dalam tabel. Translasi ini dapat diubah dengan parameter-parameter NA dan NAvl.

```
>MT[1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Disini merupakan isi tabel dengan angka-angka yang tidak diubah.

```
>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 lebih jelas, kamu dapat mengambil keluaran dari readtable() ke sebuah list.

```
>Table={{readtable("table.dat",ctok=ctok)}};


```

Menggunakan kolom-kolom token yang sama dan token token membaca dari file, kami dapat mencetak tabel. Kami dapat juga secara spesifik ctok, tok, dll. atau menggunakan list Table.

```
>writetable(Table,ctok=ctok,wc=5);


```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n
11	f	23	y	vg	1.8	y
12	m	32	y	g	1.8	n
13	m	29	y	vg	1.8	y
14	f	25	y	g	1.8	y
15	f	31	y	g	0.8	n
16	m	26	y	g	2.8	n
17	m	37	n	.	3.8	n
18	m	38	y	g	.	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-nilai dari kolom tabel. Melewati sebarang baris dengan nilai-nilai(“.”) dalam file) NAN, dan indeks-indeks dari kolom yang mana berisi nilai-nilai ini.

```
>{c,i}=tablecol(MT,[5,6]);
```

Kami dapat menggunakan ini untuk mengekstrasi kolom-kolom dari tabel untuk sebuah 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, kami butuh untuk mengekstrasi tabel itu sendiri dari Table list dalam kasus ini.

```
>MT=Table[1];
```

Of course, we can also use it to determine the mean value of a column or any other statistical value.

```
>mean(tablecol(MT,6))
```

2.175

Fungsi getstatistics() mengembalikan elemen-elemen dalam sebuah vektor dan hitungannya. Kami men-erapkannya ke nilai "m" dan "f" dalam kolom kedua dari tabel kami

```
>{xu,count}=getstatistics(tablecol(MT,2)); xu, count,
```

```
[1, 3]
[12, 13]
```

Kami dapat mencetak hasil dalam sebuah tabel baru.

```
>writetable(count',labr=tok[xu])
```

```
m      12
f      13
```

Fungsi selecttable() mengembalikan sebuah tabel baru dengan nilai-nilai dalam satu kolom yang dipilih dari indeks vektor. Pertama kami lihat indeks-indeks dari dua dari nilai kami dalam tabel token.

```
>v:=indexof(tok,["g","vg"])
```

```
[5, 6]
```

Sekarang kami dapat memilih baris-baris dari tabel, yang mana memiliki sebarang nilai dari v dalam baris ke-5 mereka.

```
>MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5);
```

Sekarang kami dapat mencetak tabel, dengan nilai-nilai yang diekstrasi dan diurutkan dalam 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 utnuk menghubungkan dua kolom dari tabel. Jadi kami mengekstrak kolom 2 dan 4 dan mengurutkan tabel.

```
>i=sortedrows(MT,[2,4]); ...  
> writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)
```

Dengan getstatistics(), kami dapat juga menghubungkan hitungan dalam dua kolom dari tabel pada tiap-tiap yang 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 juga ditulis ke sebuah file.

```
>filename="test.dat"; ...
>writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
```

Lalu kami dapat membaca tabel dari file.

```
>{MT2,hd,tok2,hdr}=readtable(filename,>clabs,>rlabs); ...
>writetable(MT2,labr=hdr,labc=hd)
```

	n	y
m	7	5
f	1	12

Dan menghapus file.

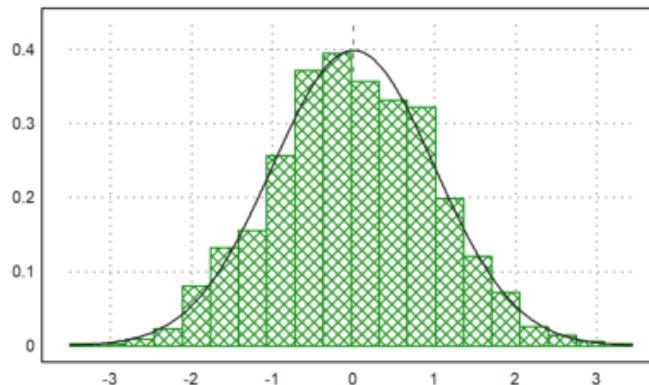
```
>fileremove(filename);
```

## Distribusi-Distribusi

---

Dengan plot2d, ini merupakan method paling mudah untuk membuat plot sebuah distribusi dari data eksperimental.

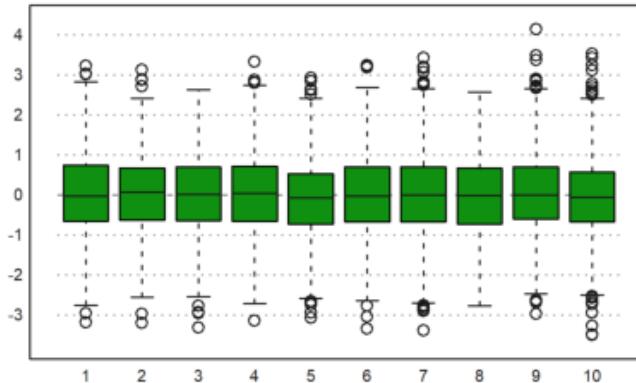
```
>p=normal(1,1000); //1000 random normal-distributed sample p  
>plot2d(p,distribution=20,style="\\\"/\\\""); // plot the random sample p  
>plot2d("qnormal(x,0,1)",add=1); // add the standard normal distribution plot
```



Tolong catat perbedaan antara plot batang (contoh) dan kurva normal (distribusi real). Enter kembali tiga perintah untuk melihat hasil lainnya.

Ini merupakan sebuah perbandingan dari 10 simulasi dari 1000 nilai berdistribusi normal menggunakan sebuah yang dipanggil box plot. Plot ini menunjukkan median, quartil 25% dan 75%, nilai minimal dan maksimal dan percikan.

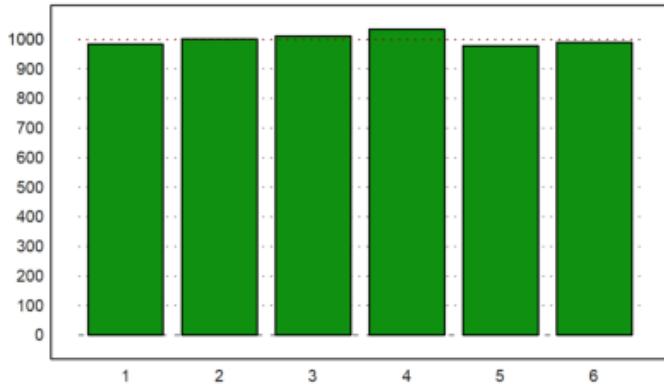
```
>p=normal(10,1000); boxplot(p):
```



Untuk menghasilkan bilangan acak, Euler memiliki intrandom. Mari kita simulasikan lemparan dadu dan plot distribusinya.

Kami menggunakan fungsi getmultiplicities(v,x), yang mana menghitung seberapa sering elemen dari v muncul dalam x. Lantas kami membuat plot dari hasilnya menggunakan columnsplot().

```
>k=intrandom(1,6000,6); ...
>columnsplot(getmultiplicities(1:6,k)); ...
>ygrid(1000,color=red):
```



Ketika `inrandom(n,m,k)` mengembalikan bilangan bulat secara seragam berdistribusi dari 1 ke k, ini memungkinkan untuk menggunakan sebarang lainnya diberikan distribusi dari bilangan bulat dengan `randpint()`.

Contoh berikut, peluang untuk 1,2,3 adalah 0.4,0.1,0.5 secara berurutan.

```
>randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,%)
```

[378, 102, 520]

Euler dapat menghasilkan nilai-nilai acak dari distibusi-distribusi lainnya. Lihatlah ke referensi berikut.

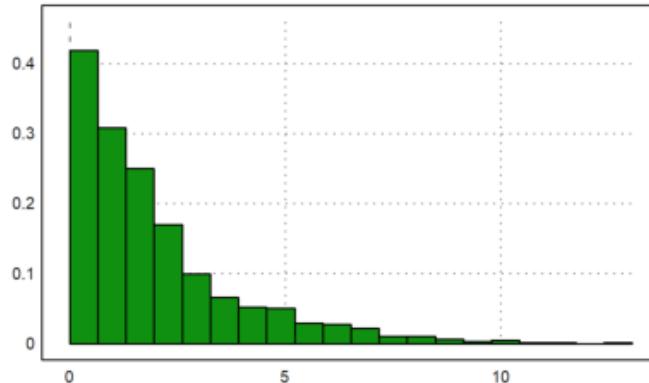
Contoh, kami mencoba distribusi eksponensial. Sebuah variabel kontinu acak X dikatakan memiliki distribusi eksponensial, jika PDF diberikan sebagai

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

dengan parameter

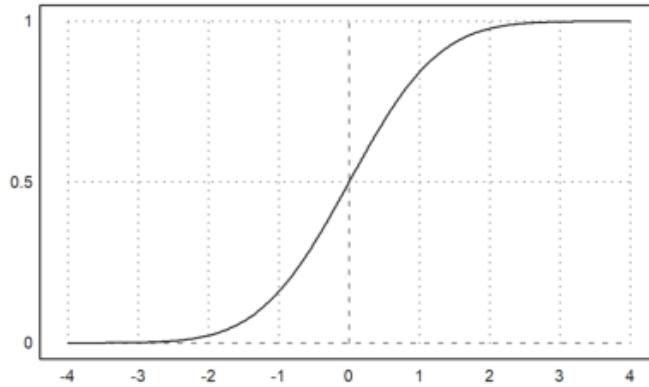
$$\lambda = \frac{1}{\mu}, \quad \mu \text{ is the mean, and denoted by } X \sim \text{Exponential}(\lambda).$$

```
>plot2d(randexponential(1,1000,2),>distribution):
```



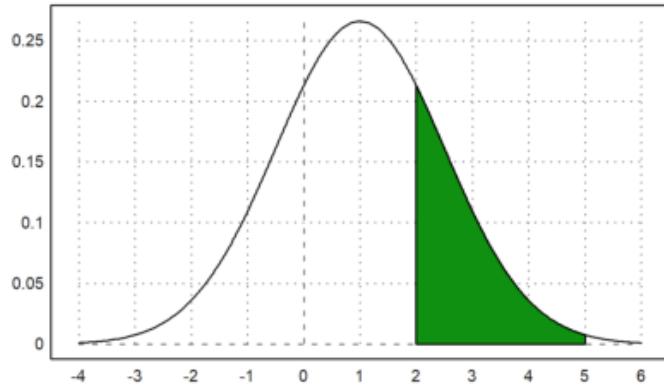
Untuk distribusi lainnya, Euler dapat menghitung fungsi distribusi dan sebaliknya.

```
>plot2d("normaldis",-4,4):
```



Berikut merupakan sebuah cara untuk membuat plot sebuah quantile.

```
>plot2d("qnormal(x,1,1.5)",-4,6); ...  
>plot2d("qnormal(x,1,1.5)",a=2,b=5,>add,>filled):
```



$$\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} dt.$$

Peluang dalam luasan hijau seperti berikut.

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

0.248662156979

Ini dapat dihitung secara numerik dengan integral berikut.

$$\int_2^5 \frac{1}{1.5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-1}{1.5})^2} dx.$$

```
>gauss("qnormal(x,1,1.5)",2,5)
```

0.248662156979

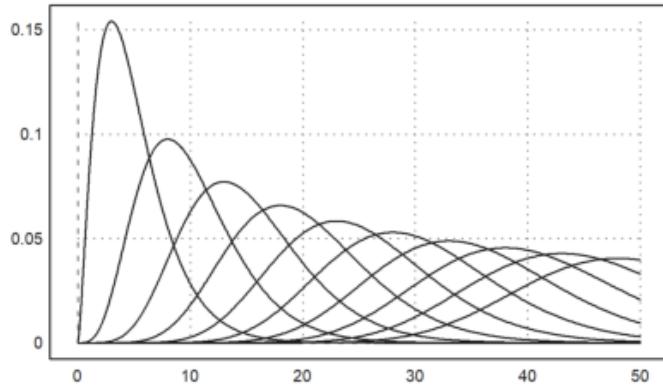
Mari kita bandingkan distribusi binomial dengan distribusi normal dari rata-rata yang sama dan deviasi. Fungsi invbindis() menyelesaikan sebuah interpolasi linear antara nilai-nilai bilangan bulat.

```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

525.516721219  
526.007419394

Fungsi qdis() adalah sebuah kepadatan dari distribusi chi-square. Seperti biasa, Euler memetakan vektor-vektor ke fungsi ini. Lantas kami mendapatkan sebuah plot dari semua distribusi chi-square dengan derajat 5 ke 30 secara mudah mengikuti langkah berikut.

```
>plot2d("qchidis(x,(5:5:50)')",0,50):
```



Euler memiliki fungsi akurasi untuk mengevaluasi distribusi. Mari kita periksa chidis() dengan sebuah integral.

Penamaan mencoba untuk lebih konsisten. Contoh,

- distribusi chi-square adalah chidis(),
- fungsi invers adalah invchidis(),
- kepadatan adalah qchidis().

Komplemen dari distribusi (ekor atas) merupakan chicdis().

```
>chidis(1.5,2), integrate("qchidis(x,2)",0,1.5)
```

```
0.527633447259  
0.527633447259
```

## Distribusi-Distribusi Diskret

---

Untuk mendefinisikan distribusi diskret milik anda, kamu dapat menggunakan method berikut.  
Pertama kami mengatur fungsi distribusi.

```
>wd = 0|((1:6)+[-0.01,0.01,0,0,0,0])/6
```

```
[0, 0.165, 0.335, 0.5, 0.666667, 0.833333, 1]
```

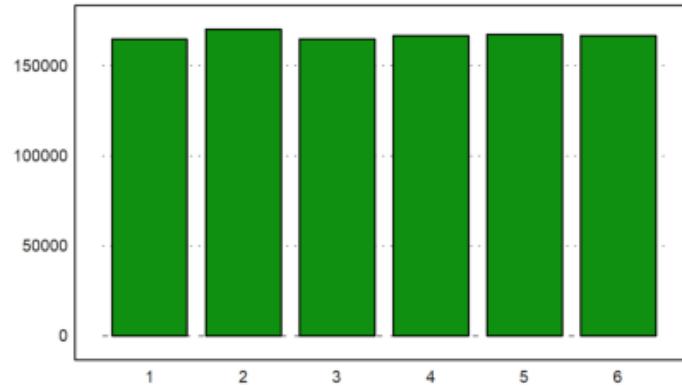
Arti ini bahwa dengan peluang  $wd[i+1]-wd[i]$  kami menghasilkan nilai i acak.

Ini hampir sebuah distribusi seragam. Mari kita definisikan sebuah pencetak angka acak untuk ini. Fungsi `find(v,x)` menemukan nilai x dalam vektor v. Ini bekerja untuk vektor x juga.

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

Error merupakan jelas bahwa kami melihatnya hanya dengan iterasi banyak sekali.

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```



Ini merupakan fungsi sederhana untuk memeriksa untuk distribusi seragam dari nilai-nilai 1...K dalam v. Kami menerima hasil ini, jika untuk semua frekuensi.

$$\left| f_i - \frac{1}{K} \right| < \frac{\delta}{\sqrt{n}}.$$

```
>function checkrandom (v, delta=1) ...
```

```
K=max(v); n=cols(v);
fr=getfrequencies(v,1:K);
return max(fr/n-1/K)<delta/sqrt(n);
endfunction
```

Tentusaja fungsi akan menolak distribusi seragam.

```
>checkrandom(wrongdice(1,1000000))
```

0

Dan ini menerima pencetak acak built-in.

```
>checkrandom(intrandom(1,1000000,6))
```

1

Kami dapat menghitung distribusi binomial. Pertama ini adalah binomials() , yang mana mengembalikan peluang dari i kurang dari n percobaan.

```
>bindis(410,1000,0.4)
```

0.751401349654

Fungsi invers Beta digunakan untuk menghitung konfidens interval Clopper-Pearson untuk parameter p. Level bawaan adalah alpha.

Arti dari interval ini adalah bahwa jika p berada diluar interval, hasil yang diamati dari 410 dalam 1000 adalah jarang.

```
>clopperpearson(410,1000)
```

```
[0.37932, 0.441212]
```

Perintah-perintah berikut merupakan cara langsung untuk mendapatkan hasil diatas. Tetapi untuk n besar, penjumlahan langsung tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

```
0.751401349655
```

Selain itu, invbinsum() menghitung invers dari binomialsu().

```
>invbindis(0.75,1000,0.4)
```

```
409.932733047
```

Dalam Bridge, kami asumsikan 5 kartu yang keluar (dari 52) dalam dua tangan (26 kartu), kami mengasumsikan peluang dari sebuah distribusi yang buruk dari 3:2 (contohnya 0:5, 1:4, 4:1 atau 5:0).

```
>2*hypergeomsum(1,5,13,26)
```

0.321739130435

Ini juga sebuah simulasi dari distribusi multinomial.

```
>randmultinomial(10,1000,[0.4,0.1,0.5])
```

381	100	519
376	91	533
417	80	503
440	94	466
406	112	482
408	94	498
395	107	498
399	96	505
428	87	485
400	99	501

## Membuat Plot Data

---

Untuk memplot data, kami mencoba hasil dari pemilihan Jerman sejak 1990, dihitung dalam tempat duduk.

```
>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 partisi-partisi, kami gunakan sebuah string dari nama-nama.

```
>P := ["CDU/CSU", "SPD", "FDP", "Gr", "Li"];
```

Mari kita cetak presentase secara bagus.

Pertama kami ekstrak kolom-kolom yang dibutuhkan. Kolom 3 ke 7 merupakan tempat duduk dari setiap partai, dan kolom 2 merupakan banyaknya tempat duduk. Kolom merupakan tahun pemilihan.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Lalu kami mencetak statistik dalam bentuk tabel. Kami menggunakan nama-nama seperti kolom headers dan tahun sebagai headers untuk baris-baris. Lebar bawaan untuk kolom adalah `wc=10`, tapi kami menyarankan output yang lebih padat. Kolom-kolom akan di ekspansi untuk label-label dari kolom-kolom, jika dibutuhkan.

```
>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

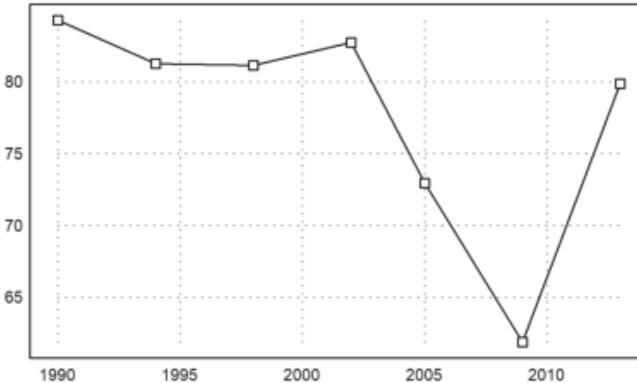
Perkalian matriks berikut ekstraks penjumlahan dari persentase dari dua partai besar menunjukkan partai kecil mendapatkan sisa dari parlemen hingga 2009.

```
>BT1:=(BT.[1;1;0;0;0])*100
```

```
[84.29, 81.25, 81.1659, 82.7529, 72.9642, 61.8971, 79.8732]
```

Ini juga merupakan sebuah plot statistikal sederhana. Kami menggunakan ini untuk menampilkan garis dan titik secara simultan. Cara lainnya adalah memanggil plot2d dua kali dengan >add.

```
>statplot(YT,BT1,"b"):
```

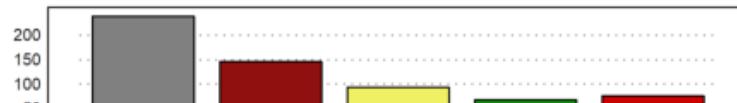


Definisi beberapa warna untuk setiap partai.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

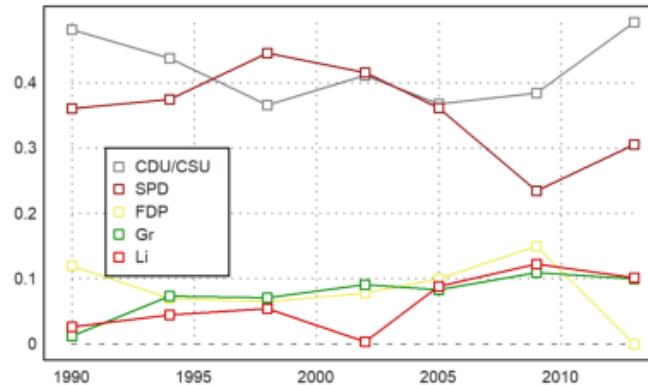
Sekarang kami dapat membuat plot hasil dari pemilihan 2009 dan mengubah menjadi satu plot menggunakan gambar. Kami dapat menambahkan sebuah vektor dari kolumn ke setiap plot.

```
>figure(2,1); ...
>figure(1); columnsplot(BW[6,3:7],P,color=CP); ...
>figure(2); columnsplot(BW[6,3:7]-BW[5,3:7],P,color=CP); ...
>figure(0);
```



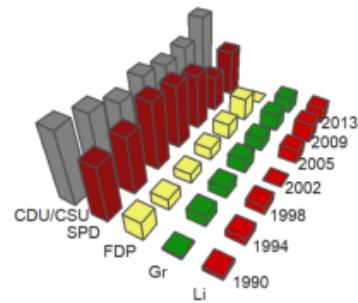
Plot data menggabungkan baris-baris dari data statistik dalam satu plot.

```
>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):
```



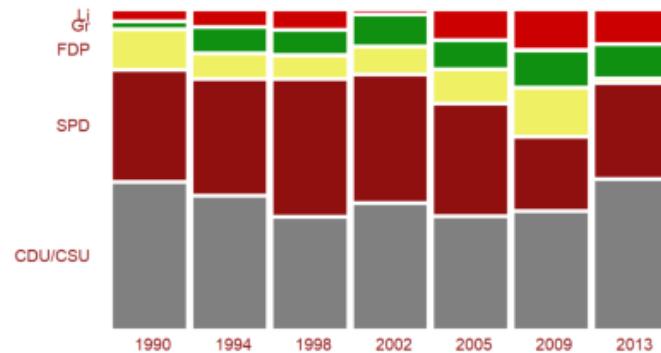
Sebuah plot 3D kolom menunjukkan baris baris dari data statistik dalam bentuk kolom. Kami menyediakan label-label untuk baris-baris dan kolom-kolom. Sudut adalah sudut pandangan.

```
>columnspplot3d(BT,scols=P,srows=YT, ...
> angle=30°,ccols=CP):
```



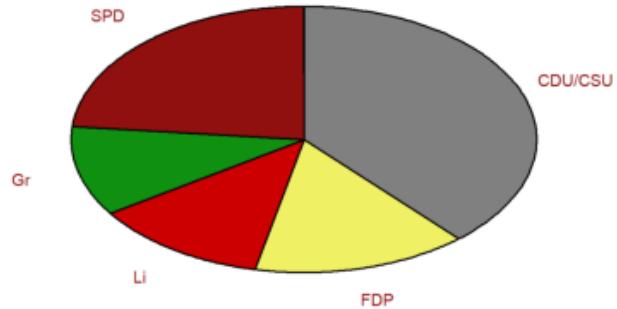
Representasi lainnya adalah plot mosaik. Catatan bahwa kolom-kolom dari plot mewakili kolom-kolom dari matrik ini. Karena panjang dari label CDU/CSU, kami menggambil jendela terkecil daripada biasanya.

```
>shrinkwindow(>smaller); ...
>mosaicplot(BT', srows=YT, scols=P, color=CP, style="#");
>shrinkwindow():
```



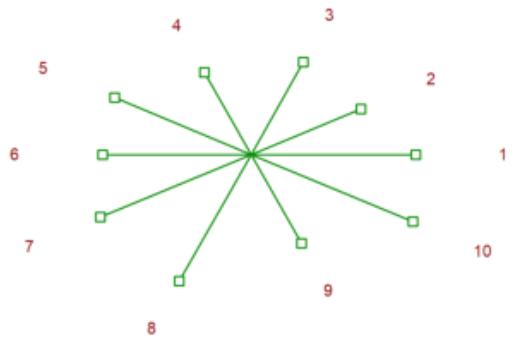
Kami dapat juga melakukan sebuah chat pie. Karena hitam dan kuning membentuk sebuah koalisi, kami mengurutkan kembali elemen-elemen.

```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```



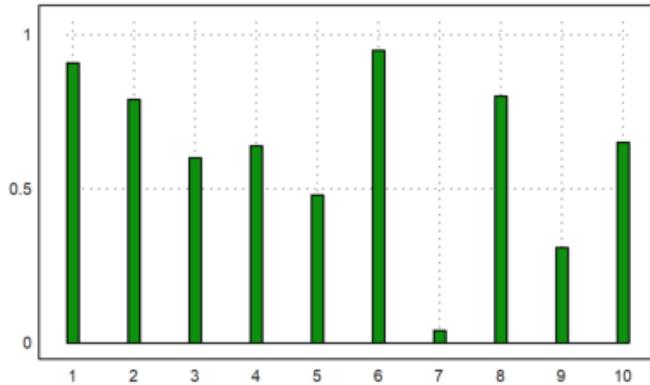
Ini merupakan jenis plot lainnya.

```
>starplot(normal(1,10)+4,lab=1:10,>rays):
```



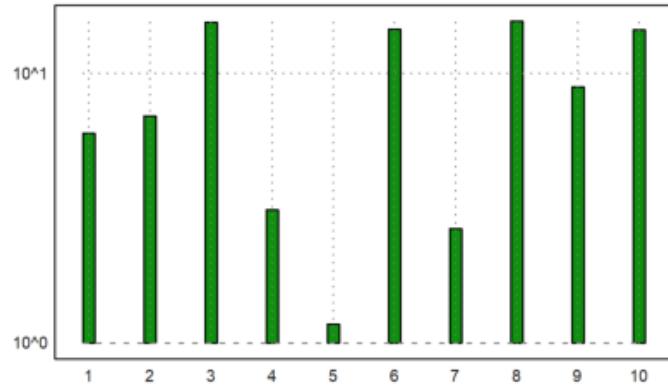
Beberapa plot dalam plot2d bagus dalam statistik. Ini merupakan plot implus dari data acak secara seragam berdistribusi dalam [0,1].

```
>plot2d(makeimpulse(1:10,random(1,10)),>bar):
```



Tetapi secara eksponensial data berdistribusi, kami membutuhkan sebuah plot logaritmik

```
>logimpulseplot(1:10,-log(random(1,10))*10):
```



Fungsi `columnsplot()` mudah untuk digunakan, karena ini hanya membutuhkan sebuah vektor dari nilai-nilai. Selain itu, ini dapat diatur label-labelnya untuk sebarang yang kami inginkan, kami mendemonstrasikan ini setelah tutorial ini.

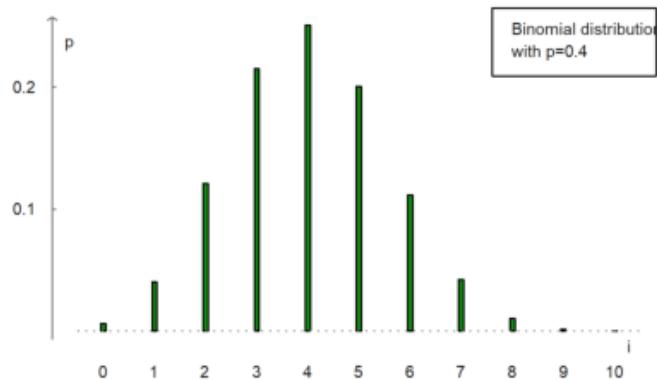
Ini merupakan aplikasi lainnya, dimana kami menghitung karakter dalam sebuah kalimat dan plot sebuah statistik.

```
>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):
```



Ini juga memungkinkan secara manual mengatur sumbu-sumbu.

```
>n=10; p=0.4; i=0:n; x=bin(n,i)*p^i*(1-p)^(n-i); ...
>columnsplot(x,lab=i,width=0.05,<frame,<grid); ...
>yaxis(0,0:0.1:1,style="->",>left); xaxis(0,style="."); ...
>label("p",0,0.25), label("i",11,0); ...
>textbox(["Binomial distribution","with p=0.4"]):
```



Berikut merupakan sebuah cara untuk membuat plot frekuensi dari angka-angka dalam sebuah vektor. Kami membuat sebuah vektor dari bilangan bulat acak 1 ke 6.

```
>v:=intrandom(1,10,10)
```

```
[8, 5, 8, 8, 6, 8, 8, 3, 5, 5]
```

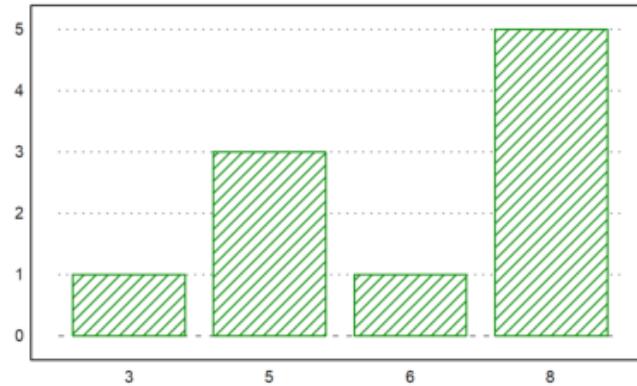
Lalu mengekstrasi angka-angka unique dalam v.

```
>vu:=unique(v)
```

```
[3, 5, 6, 8]
```

Dan plot frekuensi-frekuensi dalam sebuah plot kolom.

```
>columnsplot(getmultiplicities(vu,v),lab=vu,style="/"):
```



Kami ingin mendemonstrasikan fungsi untuk distribusi empiris dari nilai-nilai.

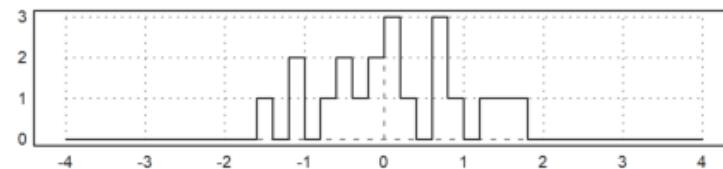
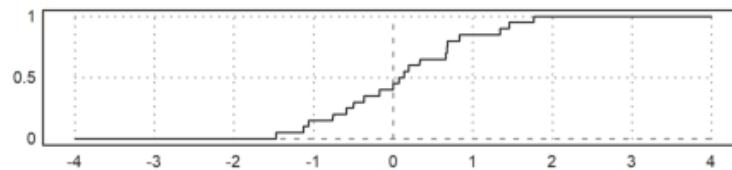
```
>x=normal(1,20);
```

Fungsi empdist(x,vs) membutuhkan sebuah nilai yang terurut dari nilai-nilai. Jadi kami harus mengurutkan x sebelum kami digunakan.

```
>xs=sort(x);
```

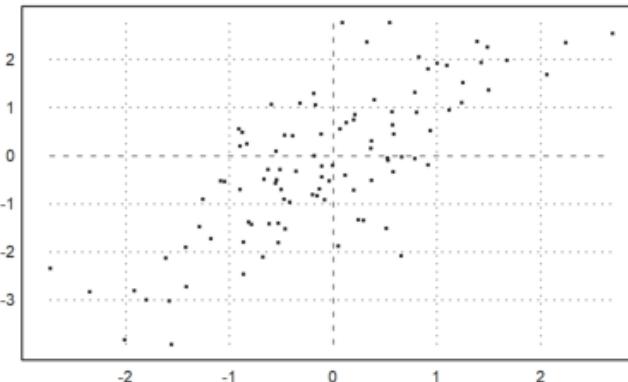
Lalu kami plot distribusi empirik dan beberapa kotak kepadatan kedalam satu plot. Daripada membuat plot batang untuk distribusi kami dengan sebuah sawtooth plot pada waktu ini.

```
>figure(2,1); ...
>figure(1); plot2d("empdist",-4,4;xs); ...
>figure(2); plot2d(histo(x,v=-4:0.2:4,<bar)); ...
>figure(0):
```



Sebuah plot pencar mudah untuk dilakukan pada Euler dengan titik plot umum. Graph berikut menunjukkan bahwa X dan X+Y terlihat memiliki korelasi positif.

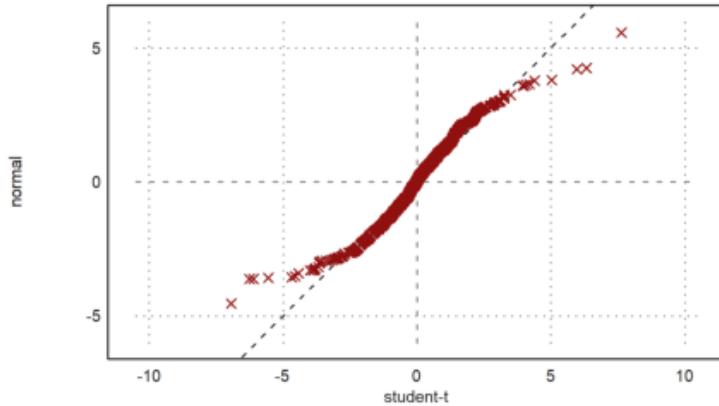
```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style=".."):
```



Terkadang, kami berharap untuk membandingkan dua contoh dari distribusi yang berbeda. Ini dapat dilakukan dengan sebuah quantile-quantile-plot.

Untuk menguji, 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):
```



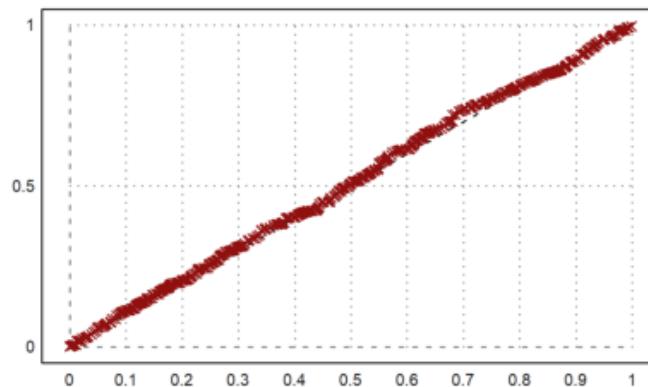
Dengan jelas plot menunjukkan bahwa nilai nilai normal terdistribusi cenderung lebih kecil pada ekstrim terakhirnya.

Jika kita memiliki dua distribusi dari ukuran yang berbeda, kami dapat memperluas yang satu lebih kecil atau menyusut yang terbesar. Fungsi berikut bagus dikeduanya. Ini mengambil nilai median dengan persentase diantara 0 dan 1.

```
>function medianexpand (x,n) := median(x,p=linspace(0,1,n-1));
```

Mari kita membandingkan dua distribusi yang seimbang.

```
>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 linear dapat dilakukan dengan fungsi polyfit() atau fungsi fit lainnya.

Untuk memulainya kami 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 tidak memiliki beban dan memiliki beban yang cocok. Pertama koefisien dari kecocokan linear.

```
>p=polyfit(x,y,1)
```

[0.733333, 0.812121]

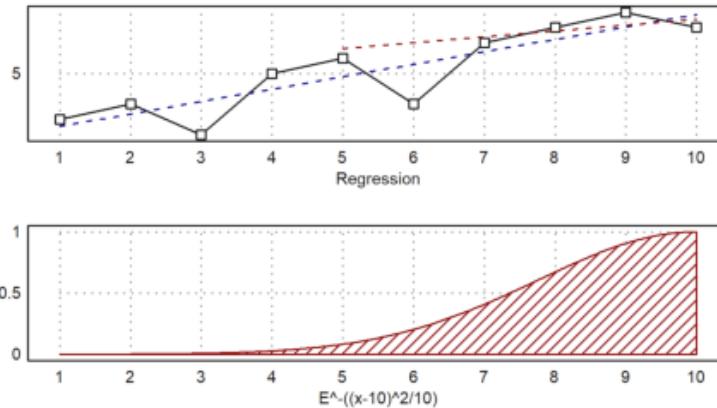
Sekarang koefisien dengan bobot yang menekankan nilai terakhir.

```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

```
[4.71566,  0.38319]
```

Kita ambil semuanya ke dalam sebuah plot untuk titik titik dan garis regresi, dan bobot yang digunakan.

```
>figure(2,1); ...
>figure(1); statplot(x,y,"b",xl="Regression"); ...
> plot2d("evalpoly(x,p)",>add,color=blue,style="--"); ...
> plot2d("evalpoly(x,pw)",5,10,>add,color=red,style="--"); ...
>figure(2); plot2d(w,1,10,>filled,style="/",fillcolor=red,xl=w); ...
>figure(0):
```



Untuk contoh lainnya kami membaca sebuah survey dari murid-murid, umur mereka dan umur orang tua mereka dan banyaknya saudara dari sebuah file.

Tabel ini berisi "m" dan "f" di kolom kedua. Kami gunakan variabel tok2 untuk mengatur translasi yang pas daripada membiarkan readtable() mengumpulkan translasi.

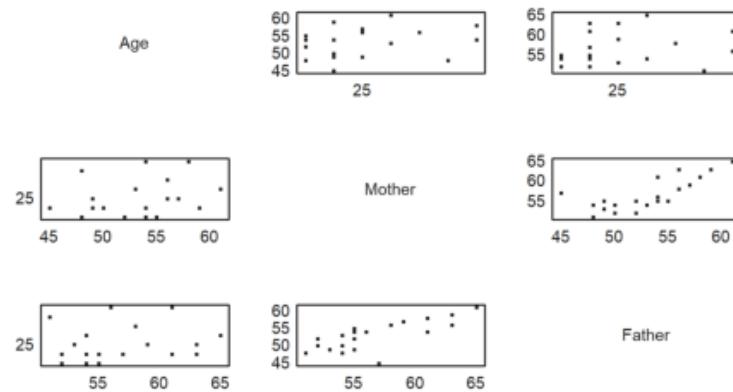
```
>{MS,hd}:=readtable("table1.dat",tok2:=[ "m", "f"]);  ...
>writetable(MS,labc=hd,tok2:=[ "m", "f"]);
```

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 umur bergantung dari yang lainnya? Kesan pertama datang dari plot pencar berpasangan.

```
>scatterplots(tablecol(MS,3:5),hd[3:5]):
```



Jelas bahwa umur dari ayah dan ibu tergantung pada setiapnya. Mari kita tentukan dan plot garis regresi.

```
>cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1)
```

```
[17.3789, 0.740964]
```

Ini secara jelas merupakan model yang salah. Garis regresi seharusnya = $17+0.74t$ , dimana t adalah umur dari ibu dan s adalah umur dari ayah. Perbedaan umur mungkin tergantung sedikit dari umur, tetapi tidak banyak.

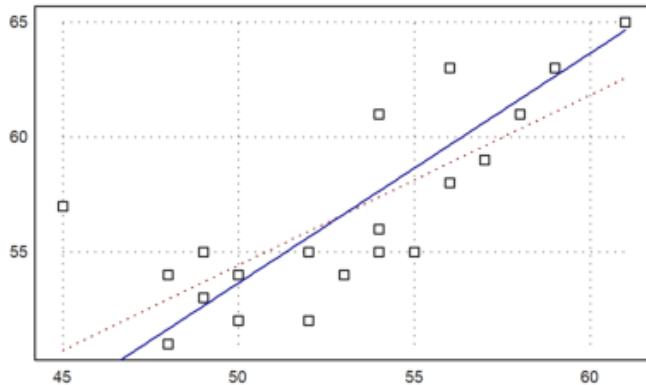
Selain itu, kami mengira sebuah fungsi seperti  $s=a+t$ . Lalu a adalah rata rata dari s-t. Ini merupakan usia rata-rata perbedaan antara ayah dan ibu.

```
>da:=mean(cs[2]-cs[1])
```

```
3.65
```

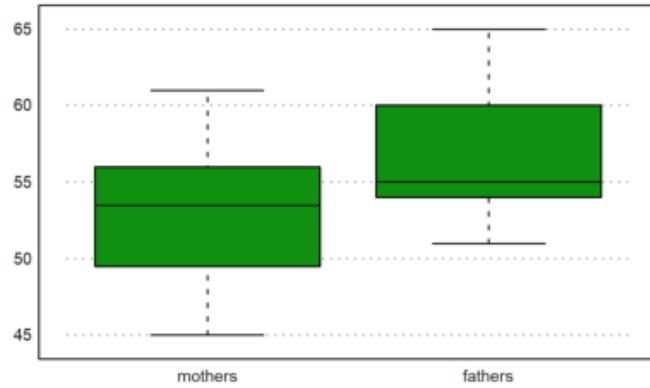
Mari kita plot ini ke satu plot pencar.

```
>plot2d(cs[1],cs[2],>points);  ...
>plot2d("evalpoly(x,ps)",color=red,style=".",>add);  ...
>plot2d("x+da",color=blue,>add):
```



Ini merupakan plot kotak dari dua umur. Ini hanya menunjukkan, bahwa umur berbeda.

```
>boxplot(cs,["mothers","fathers"]):
```



Ini menarik bahawa perbezaan dalam nilai tengah tidak sebesar seperti perbezaan dalam nilai rata-rata.

```
>median(cs[2])-median(cs[1])
```

1.5

Koefisien korelasi menyarankan korelasi positif.

```
>correl(cs[1],cs[2])
```

0.7588307236

Korelasi dari peringkat merupakan sebuah ukuran untuk urutan yang sama dalam vektor keduannya. Ini juga sedikit positif.

```
>rankcorrel(cs[1],cs[2])
```

0.758925292358

## Membuat fungsi-fungsi baru

---

Tentu saja, bahasa EMT dapat digunakan untuk membuat program fungsi baru. Sebagai contoh kami mendefinisikan fungsi skewness.

$$\text{sk}(x) = \frac{\sqrt{n} \sum_i (x_i - m)^3}{\left(\sum_i (x_i - m)^2\right)^{3/2}}$$

dimana 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 kamu lihat, kami dengan mudah menggunakan bahasa matriks untuk mendapatkan sebuah implementasi yang sangat singkat dan efisien. Mari kita coba fungsi ini.

```
>data=normal(20); skew(normal(10))
```

-0.198710316203

Ini merupakan fungsi lainnya, sebut saja koefisien skewness Pearson.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)
>skew1(data)
```

-0.0801873249135

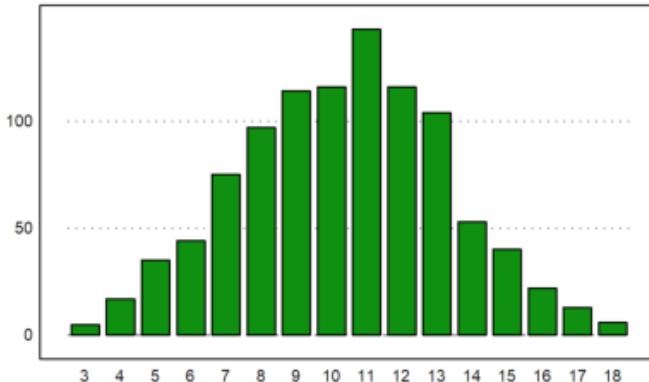
Euler dapat digunakan untuk mensimulasikan kejadian acak. Kami telah melihatnya contoh sederhana di atas. Ini merupakan yang laninya, yang mana mensimulasikan 1000 kali 3 dadu dilempar, dan menanyakan distribusi dari jumlahannya.

```
>ds:=sum(intrandom(1000,3,6)); fs=getmultiplicities(3:18,ds)
```

```
[5, 17, 35, 44, 75, 97, 114, 116, 143, 116, 104, 53, 40,  
22, 13, 6]
```

Kami dapat plot ini sekarang.

```
>columnsplot(fs,lab=3:18):
```



Untuk menentukan distribusi yang diinginkan tidak mudah. Kami menggunakan sebuah rekursi lanjut untuk ini.

Berdasarkan perhitungan fungsi jumlah cara angka k dapat merepresentasikan sebagai jumlahan dari angka n dalam rentang 1 ke m. ini bekerja secara rekursif dalam cara yang jelas.

```
>function map countways (k; n, m) ...
if n==1 then return k>=1 && k<=m
else
  sum=0;
  loop 1 to m; sum=sum+countways(k-#,n-1,m); end;
  return sum;
end;
endfunction
```

Ini merupakan hasil dari lemparan tiga dadu.

```
>countways(5:25,5,5)
```

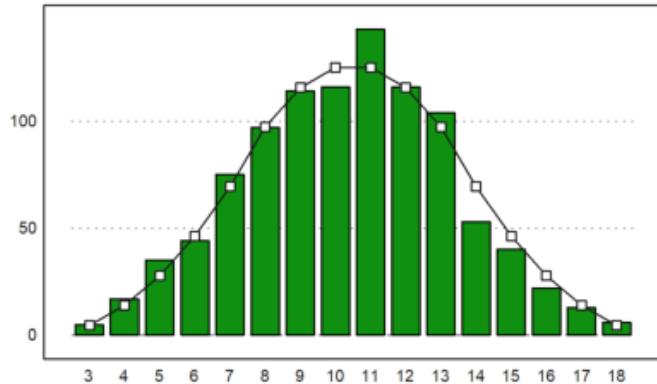
```
[1,  5,  15,  35,  70,  121,  185,  255,  320,  365,  381,  365,  320,
255,  185,  121,  70,  35,  15,  5,  1]
```

```
>cw=countways(3:18,3,6)
```

```
[1,  3,  6,  10,  15,  21,  25,  27,  27,  25,  21,  15,  10,  6,  3,
1]
```

Kami tambahkan nilai harapan ke plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```



Untuk simulasi lainnya, deviasi dari nilai rata-rata n 0-1-normal variabel acak distribusi adalah  $1/\sqrt{n}$ .

```
>longformat; 1/sqrt(10)
```

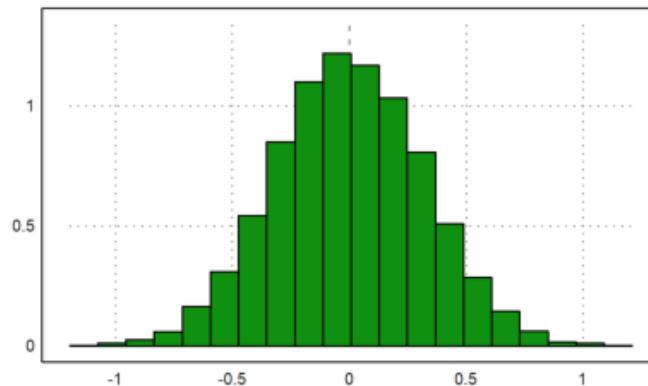
0.316227766017

Mari kita periksa ini dengan sebuah simulasi. Kami menghasilkan 10000 kali 10 vektor acak.

```
>M=normal(10000,10); dev(mean(M))
```

0.319493614817

```
>plot2d(mean(M)',>distribution):
```



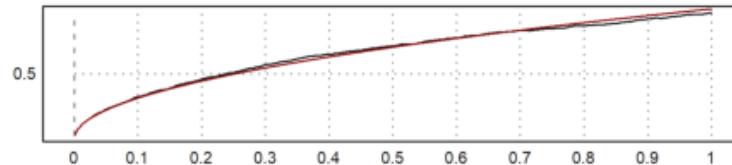
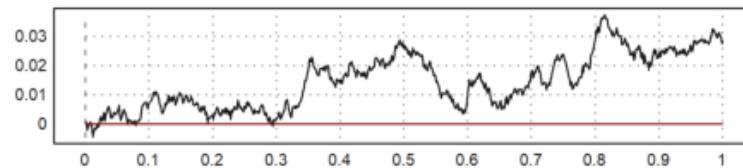
Nilai median dari 10 0-1-normal distribusi angka acak memiliki deviasi yang lebih besar.

```
>dev(median(M'))
```

0.374460271535

Karena kami dapat dengan mudah membuat langkah acak, kami dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami lalu plot standar deviasi dan rata-rata dari langkah ke-n dari proses bersama ini dengan nilai harapan dalam 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):
```



## Pengujian

---

Pengujian merupakan alat yang penting dalam statistika. Dalam Euler, banyak uji yang diimplementasikan. Semua uji ini mengembalikan error yang dapat kita terima apabila menolak hipotesis nol.

Sebagai contoh, kami menguji pelemparan dadu untuk distribusi seragam. Pada lemparan 600, kami mendapatkan nilai berikut, yang mana dapat kami masukkan kedalam uji chi-square.

```
>chitest([90,103,114,101,103,89],dup(100,6)')
```

0.498830517952

Uji chi-square juga memiliki sebuah mode, yang mana menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya seharusnya hampir sama. Parameter >p menunjukkan vektor-y sebagai vektor peluang.

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

0.526

Error ini terlalu besar. Jadi kami dapat menolak distribusi seragam. Ini tidak menunjukkan bahwa dadu kami tidak adil. Tapi kami tidak dapat menolak hipotesis kami.

Selanjutnya kami membuat 1000 dadu dilemparkan menggunakan pembuat angka acak dan menggunakan uji yang sama.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

0.528028118442

Mari kita uji untuk nilai rata-rata 100 dengan uji-t.

```
>s=200+normal([1,100])*10; ...
>ttest(mean(s),dev(s),100,200)
```

0.0218365848476

ttest() fungsi membutuhkan nilai rata-rata, deviasi, banyaknya data, dan nilai rata-rata untuk mengujinya.

Sekarang mari kita periksa perhitungan untuk nilai rata-rata yang sama. Kami menolak hipotesis bahwa mereka memiliki rata-rata yang sama, jika hasilnya adalah  $< 0.05$ .

```
>tcomparedata(normal(1,10),normal(1,10))
```

0.38722000942

Jika kita menambahkan sebuah bias ke distribusi satu, kami mendapatkan lebih banyak penolakan. Mengulang simulasi ini beberapa kali untuk melihat pengaruhnya.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

5.60009101758e-07

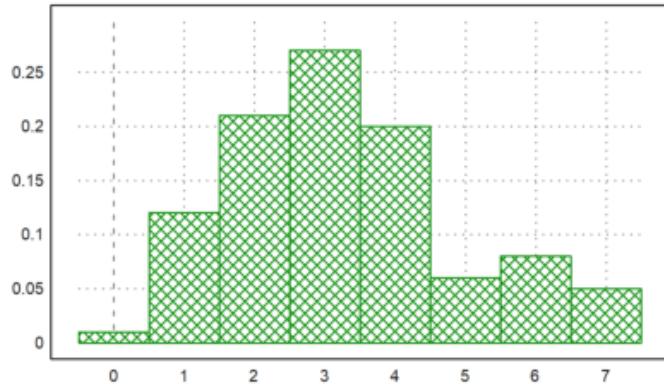
Contoh selanjutnya, kami membuat 20 dadu acak dilempar 100 kali dan menghitung satunya. Terdapat satu  $20/6=3.3$  dalam rata-rata.

```
>R=random(100,20); R=sum(R*6<=1); mean(R)
```

3.28

Kami sekarang membandingkan angka satu dengan distribusi binomial. Pertama kami plot distribusi angka satu.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\/"):
```



```
>t=count(R,21);
```

Lalu kami hitung nilai harapan.

```
>n=0:20; b=bin(20,n)*(1/6)^n*(5/6)^(20-n)*100;
```

Kami mengumpulkan beberapa angka untuk mendapatkan kategori, yang mana 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 kami merupakan sebuah distribusi binomial, jika hasilnya adalah  $< 0.05$ .

```
>chitest(t1,b1)
```

0.53921579764

Contoh berikut memiliki hasil dari dua grup dari orang-orang (laki-laki dan perempuan, katakanlah) 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

Kami berharap menguji secara bebas dari suara untuk jenis kelamin. Tabel uji chi^2 melakukan ini. Hasilnya terlalu besar untuk menolak saling bebas. Jadi kita tidak dapat mengatakan, jika suara tergantung pada jenis kelamin untuk data ini.

```
>tabletest(A)
```

0.990701632326

Tabel nilai harapan berikut, jika kita berasumsi 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

Kami dapat menghitung koefisien kontigensi yang dikoreksi. Karena ini sangat dekat dengan 0, kami menyimpulkan bahwa pemungutan suara tidak tergantung pada jenis kelamin.

```
>contingency(A)
```

0.0427225484717

## Beberapa Pengujian Lainnya

---

Selanjutnya kami menggunakan analisis variansi (uji-F) untuk menguji tiga sample dari data distribusi secara normal untuk nilai rata-rata yang sama. Metode ini dinamakan ANOVA (variansi analisis). Dalam Euluer, fungsi varanalysis() digunakan.

```
>x1=[109,111,98,119,91,118,109,99,115,109,94]; mean(x1),
```

106.545454545

```
>x2=[120,124,115,139,114,110,113,120,117]; mean(x2),
```

119.111111111

```
>x3=[120,112,115,110,105,134,105,130,121,111]; mean(x3)
```

116.3

```
>varanalysis(x1,x2,x3)
```

0.0138048221371

Ini berarti, kami menolak hipotesis dari nilai rata-rata yang sama. Kami melakukan ini dengan peluang error dari 1.3%.

Ini juga pengujian median, yang mana menolak data sampel dengan perbedaan rata-rata distribusi pengujian median dari sampel gabungan.

```
>a=[56,66,68,49,61,53,45,58,54];  
>b=[72,81,51,73,69,78,59,67,65,71,68,71];  
>mediantest(a,b)
```

0.0241724220052

Uji lainnya dalam kesamaan adalah uji rank. Ini lebih tajam daripada uji median.

```
>ranktest(a,b)
```

0.00199969612469

Contoh berikut, distribusi keduanya memiliki nilai rata-rata yang sama.

```
>ranktest(random(1,100),random(1,50)*3-1)
```

0.129608141484

Mari kita coba untuk mensimulasikan perlakuan a dan b diaplikasikan ke orang-orang yang berbeda.

```
>a=[8.0,7.4,5.9,9.4,8.6,8.2,7.6,8.1,6.2,8.9];  
>b=[6.8,7.1,6.8,8.3,7.9,7.2,7.4,6.8,6.8,8.1];
```

Uji signum memutuskan, jika a lebih baik daripada b.

```
>signtest(a,b)
```

0.0546875

Ini memiliki banyak error. Kami tidak bisa menolak bahwa a sebaik b.

Uji Wilcoxon memberikan hasil lebih tajam dari uji ini, tetapi bergantung pada nilai kuantitatif dari perbedaan.

```
>>wilcoxon(a,b)
```

0.0296680599405

Mari kita coba uji dua tes lagi menggunakan deret yang dibuat.

```
>wilcoxon(normal(1,20),normal(1,20)-1)
```

0.0068706451766

```
>wilcoxon(normal(1,20),normal(1,20))
```

0.275145971064

## Angka-Angka Acak

---

Uji berikut untuk membuat angka acak. Euler menggunakan sebuah pengacak angka yang bagus, jadi kita tidak mengharapkan masalah lainnya.

Pertama kita buat sepuluh juta angka acak dalam  $[0, 1]$ .

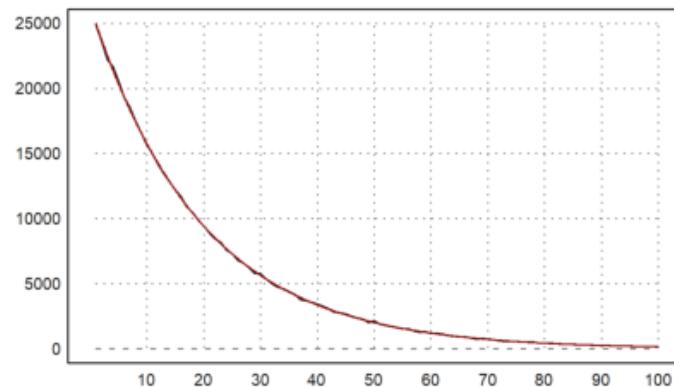
```
>n:=10000000; r:=random(1,n);
```

Sekarang kita hitung jarak antara dua angka kurang dari 0.05.

```
>a:=0.05; d:=differences(nonzeros(r<a));
```

Terakhir, kami plot angka beberapa kali, setiap jarak yang terjadi dan bandingkan dengan nilai harapan.

```
>m=getmultiplicities(1:100,d); plot2d(m); ...
> plot2d("n*(1-a)^(x-1)*a^2",color=red,>add):
```



Hapus data.

```
>remvalue n;
```

## **Pengantar untuk pengguna dalam Projek R**

---

Secara jelas, EMT tidak bersaing dengan R sebagai package statistik. Namun, ini banyak prosedur statistik dan fungsi yang tersedia dalam EMT juga. Jadi EMT mungkin memenuhi kebutuhan dasar. Kemudian, EMT datang dengan numerical package dan sistem komputer aljabar.

Notebook ini untuk kamu jika umum dengan R, tetapi ingin tahu perbedaan sintaks dari EMT dan R. Kami mencoba untuk memberikan gambaran jelas dan kurang jelas yang kamu ketahui.

Selebihnya, kami lihat pada cara untuk bertukar data pada kedua sistem.

Catatan ini dalam proses pengerjaan.

**Sintaks Dasar**

---

Hal pertama yang dipelajari dalam R adalah membuat sebuah vektor. Dalam EMT, perbedaan utamanya pada operator : yang dapat mengambil jarak langkah. Lebih dari itu memiliki daya ikat rendah.

```
>n=10; 0:n/20:n-1
```

```
[0,  0.5,  1,  1.5,  2,  2.5,  3,  3.5,  4,  4.5,  5,  5.5,  6,  6.5,
 7,  7.5,  8,  8.5,  9]
```

Fungsi c() tidak ada. Ini memungkinkan untuk membuat vektor menjadi benda-benda yang bergabung. Contoh berikut, seperti lainnya, dari "Introduction to R" datang dari projek R. Jika kamu membaca PDF ini, kamu akan menemukan bahwa I mengikuti path dalam tutorial ini.

```
>x=[10.4, 5.6, 3.1, 6.4, 21.7]; [x,0,x]
```

```
[10.4,  5.6,  3.1,  6.4,  21.7,  0,  10.4,  5.6,  3.1,  6.4,  21.7]
```

Operator colon dengan ukuran langkah dari EMT menggantikan fungsi seq() dalam R. Kami dapat menulis fungsi ini dalam EMT.

```
>function seq(a,b,c) := a:b:c; ...
>seq(0,-0.1,-1)
```

```
[0, -0.1, -0.2, -0.3, -0.4, -0.5, -0.6, -0.7, -0.8, -0.9, -1]
```

Fungsi rep() dari R tidak menunjukkan dalam EMT. Untuk masukan vektor, ini dapat ditulis seperti berikut.

```
>function rep(x:vector,n:index) := flatten(dup(x,n)); ...
>rep(x,2)
```

```
[10.4, 5.6, 3.1, 6.4, 21.7, 10.4, 5.6, 3.1, 6.4, 21.7]
```

Catatan bahwa "=" atau ":=" digunakan untuk penugasan. Operator "->" digunakan untuk ukuran dalam EMT.

```
>125km -> " miles"
```

```
77.6713990297 miles
```

Operator "`<-`" untuk penugasan adalah cara yang menyesatkan, dan ide buruk dari R. Berikut akan dibandingkan a dan -4 dalam EMT.

```
>a=2; a<-4
```

0

Dalam R, "`a < -4 < 3`" bekerja, tetapi "`a < -4 < -3`" tidak. Saya memiliki ambiguitas yang sama dalam EMT juga, tetapi mencoba untuk mengapuskan mereka sedikit demi sedikit.

EMT dan R memiliki vektor dari tipe boolean. Tetapi dalam EMT, angka 0 dan 1 digunakan untuk mewakili salah dan benar. Dalam R, nilai true dan false dapat demikian digunakan dalam aritmetik biasa seperti dalam EMT.

```
>x<5, %*x
```

```
[0, 0, 1, 0, 0]
[0, 0, 3.1, 0, 0]
```

EMT mengeluarkan error atau menghasilkan NAN tergantung pada "error".

```
>errors off; 0/0, isNaN(sqrt(-1)), errors on;
```

NAN

1

String merupakan hal yang sama dalam R dan EMT. Keduanya memiliki locale sekarang, bukan dalam Unicode.

Dalam R terdapat packages dalam Unicode. Dalam EMT, sebuah string dapat berupa string Unicode. Sebuah string unicode dapat diubah kedalam encode local dan sebaliknya. Selanjutnya, u"..." dapat berisi entitas HTML.

```
>u"\u00e9; Ren\u00e9 Grothmann"
```

© René Grothmann

Berikut mungkin atau mungkin tidak menampilkan secara benar dalam sitem kamu sebagai A dengan dot dan dash diatas. Tergantung pada font yang digunakan.

```
>chartoutf([480])
```

Gabungan string dapat dilakukan dengan "+" atau "|". Ini berisi angka, yang mana akan mencetak dalam format sekarang.

```
>"pi = "+pi
```

```
pi = 3.14159265359
```

Hampir setiap kali, ini bekerja pada R.

Tetapi EMT mengintrepetasikan indeks negatis dari belakang vektor sementara R mengintrepetasikan  $x[n]$  sebagai  $x$  tanpa elemen ke-n.

```
>x, x[1:3], x[-2]
```

```
[10.4,  5.6,  3.1,  6.4, 21.7]
[10.4,  5.6,  3.1]
6.4
```

Perilaku dari R dapat di dapatkan dalam EMT dengan drop().

```
>drop(x,2)
```

```
[10.4,  3.1,  6.4, 21.7]
```

Vektor logika tidak jauh berbeda seperti indeks dalam EMT, dalam beda ke R. Kamu butuh untuk mengekstrak elemen bukan nol pertama dalam 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]
```

Seperti dalam R, indeks vektor dapat berisi repetisi.

```
>x[[1,2,2,1]]
```

```
[10.4, 5.6, 5.6, 10.4]
```

Tetapi nama untuk indeks tidak mungkin dalam EMT. Untuk sebuah paket statistika, ini mungkin dibutuhkan untuk memudahkan akses elemen dari vektor.

Untuk meniru perilaku ini, kami dapat mendefinisikan sebuah fungsi seperti berikut.

```
>function sel (v,i,s) := v[indexof(s,i)]; ...
>s=["first","second","third","fourth"]; sel(x,[ "first","third"],s)
```

```
Trying to overwrite protected function sel!
Error in:
function sel (v,i,s) := v[indexof(s,i)]; ... ...
^
```

```
Trying to overwrite protected function sel!
Error in:
function sel (v,i,s) := v[indexof(s,i)]; ... ...
^
```

```
Trying to overwrite protected function sel!
Error in:
function sel (v,i,s) := v[indexof(s,i)]; ... ...
^
```

```
Trying to overwrite protected function sel!
Error in:
function sel (v,i,s) := v[indexof(s,i)]; ... ...
^
```

[10.4, 3.1]

## Tipe Data

---

EMT memiliki tipe data tetap dari pada R. Secara jelas, dalam R memiliki vektor bertumbuh. KAMu dapat mengatur sebuah vektor numerik kosong v dan menugaskan sebuah nilai ke elemen v[17]. Ini tidak mungkin dalam EMT.

Berikut sedikit tidak efisien.

```
>v=[]; for i=1 to 10000; v=v|i; end;
```

EMT sekarang akan membuat sebuah vektor dengan v dan i diatambahkan pada stck dan menyalin vektor itu kembali ke variabel global v.

Lebih efisien pra-definisi vektor.

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Untuk mengubah tipe data dalam EMT, kamu dapat menggunakan fungsi seperti complex().

```
>complex(1:4)
```

```
[ 1+0i ,  2+0i ,  3+0i ,  4+0i ]
```

Pengubahan ke string memungkinkan untuk tipe data dasar saja. Format sekarang digunakan untuk penggabungan string sederhana. Tetapi terdapat fungsi seperti print() atau frac().

Untuk vektor, kamu dapat dengan mudah menulis fungsi kamu sendiri.

```
>function tostr (v) ...  
  
s="[";  
loop 1 to length(v);  
    s=s+print(v[#],2,0);  
    if #<length(v) then s=s+","; endif;  
end;  
return s+"]";  
endfunction
```

```
>tostr(linspace(0,1,10))
```

```
[0.00,0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1.00]
```

Untuk komunikasi dengan Maxima, terdapat sebuah fungsi convertm xm(), yang mana juga dapat digunakan mengubah sebuah vektor untuk keluaran.

```
>convertm xm(1:10)
```

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

Untuk Latex perintah tex dapat digunakan untuk mendapatkan perintah Latex.

```
>tex(&[1,2,3])
```

```
\left[ 1 , 2 , 3 \right]
```

## Faktor Faktor dan Tabel Tabel

---

Dalam pengantar R terdapat sebuah contoh bernama faktor-faktor.

Berikut merupakan sebuah daftar dari wilayah 30 negara bagian.

```
>austates = ["tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa", ...
>"qld", "vic", "nsw", "vic", "qld", "qld", "sa", "tas", ...
>"sa", "nt", "wa", "vic", "qld", "nsw", "nsw", "wa", ...
>"sa", "act", "nsw", "vic", "vic", "act"];
```

Asumsikan, kita memiliki pendapatan yang berkorespondensi dalam setiap negara bagian.

```
>incomes = [60, 49, 40, 61, 64, 60, 59, 54, 62, 69, 70, 42, 56, ...
>61, 61, 61, 58, 51, 48, 65, 49, 49, 41, 48, 52, 46, ...
>59, 46, 58, 43];
```

Sekarang, kami ingin menghitung rata-rata pendapatan dalam wilayah. Menjadi sebuah program statistik, R memiliki factor dan tapply() untuk ini.

EMT dapat membuat ini dengan menemukan indeks dari wilayah dalam daftar unik dari wilayah.

```
>auterr=sort(unique(austates)); f=indexofsorted(auterr,austates)
```

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

Pada titik ini, kami dapat menulis fungsi putaran untuk melakukan hal ini pada hanya faktor.

Atau kami dapat menyerupai fungsi tapply() dalam cara yang sama.

```
>function map_tappl (i; f$:call, cat, x) ...
```

```
u=sort(unique(cat));  
f=indexof(u,cat);  
return f$(x[nonzeros(f==indexof(u,i))]);  
endfunction
```

Ini sedikit tidak efisien, karena ini menghitung wilayah unik untuk setiap i, tetapi ini bekerja.

```
>tappl(auterr,"mean",austates,incomes)
```

```
[44.5, 57.3333333333, 55.5, 53.6, 55, 60.5, 56, 52.25]
```

Catatan bahwa ini bekerja untuk setiap vektor dalam wilayah.

```
>tapply(["act","nsw"],"mean",austates,incomes)
```

```
[44.5, 57.3333333333]
```

Sekarang, package statistikal dari EMT mendefinisikan tabel seperti dalam R. Fungsi `readtable()` dan `writetable()` dapat digunakan untuk input dan output.

Jadi kita dapat mencetak rata-rata pendapatan negara bagian dalam wilayah sebuah cara yang mudah.

```
>writetable(tapply(auterr,"mean",austates,incomes),labc=auterr,wc=7)
```

act	nsw	nt	qld	sa	tas	vic	wa
44.5	57.33	55.5	53.6	55	60.5	56	52.25

Sekarang dapat juga mencoba untuk meniru cara kerja dari R secara utuh.

Faktor seharusnya dengan jelas menyimpan dalam sebuah collection dengan tipe dan kategori (negara bagian dan wilayah dalam contoh kita). Untuk EMT, kami menambahkan indeks pra-perhitungan.

```
>function makef (t) ...
```

```
## Factor data
## Returns a collection with data t, unique data, indices.
## See: tapply
u=sort(unique(t));
return {{t,u,indexofsorted(u,t)}};
endfunction
```

```
>statef=makef(austates);
```

Sekarang elemen ketiga dari collection akan berisi indeks.

```
>statef[3]
```

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

Sekarang kami dapat meniru tapply() dalam cara berikut. Ini akan mengembalikan sebuah tabel sebagai sebuah collection dari data tabel dan headings kolom.

```
>function tapply (t:vector,tf,f$:call) ...
```

```

## Makes a table of data and factors
## tf : output of makef()
## See: makef
uf=tf[2]; f=tf[3]; x=zeros(length(uf));
for i=1 to length(uf);
    ind=nonzeros(f==i);
    if length(ind)==0 then x[i]=NAN;
    else x[i]=f$(t[ind]);
    endif;
end;
return {{x,uf}};
endfunction

```

Kita tidak menambahkan banyak pemeriksaan tipe disini. Hanya pendekatan pencegahan kategori (faktor-faktor) tanpa data. Tetapi yang seharusnya diperiksa untuk panjang yang benar dari t dan untuk kebenaran dari tf collection.

Tabel ini dapat dicetak sebagai sebuah tabel dengan writetable().

```
>writetable(tapply(incomes,statef,"mean"),wc=7)
```

act	nsw	nt	qld	sa	tas	vic	wa
44.5	57.33	55.5	53.6	55	60.5	56	52.25

EMT hanya memiliki dua dimensi untuk array. Tipe data ini disebut matriks. Ini akan lebih mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau sebuah library C untuk ini, namun

R memiliki lebih dari dua dimensi. Dalam R array adalah sebuah vektor dengan bidang dimensi.

Dalam EMT, sebuah vektor adalah sebuah matriks dengan satu baris. Ini dapat dibuat sebuah matriks dengan `redim()`.

```
>shortformat; X=redim(1:20,4,5)
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Ekstrasi baris dan kolom, atau sub-matriks, seperti dalam R.

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

2	3
7	8
12	13
17	18

Namun, dalam R ini memungkinkan untuk menngatur sebuah baris dari indeks spesifik dari vektor ke sebuah nilai. Kemungkinan sama dalam EMT hanya dengan sebuah putaran.

```
>function setmatrixvalue (M, i, j, v) ...  
  
loop 1 to max(length(i),length(j),length(v))  
    M[i{#},j{#}] = v{#};  
end;  
endfunction
```

Kami mendemonstrasikan ini untuk menunjukkan bahwa matriks dilalui dalam EMT. Jika kamu tidak ingin mengubah matriks asli M, kamu harus menyalininya dalam fungsi.

```
>setmatrixvalue(X,1:3,3:-1:1,0); X,
```

1	2	0	4	5
6	0	8	9	10
0	12	13	14	15
16	17	18	19	20

Perkalian luar dalam EMT hanya dapat dilakukan antara vektor-vektor. Ini secara otomatis karena bahasa matriks. Satu vektor butuh menjadi sebuah kolom vektor dan yang lainnya sebuah vektor baris.

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

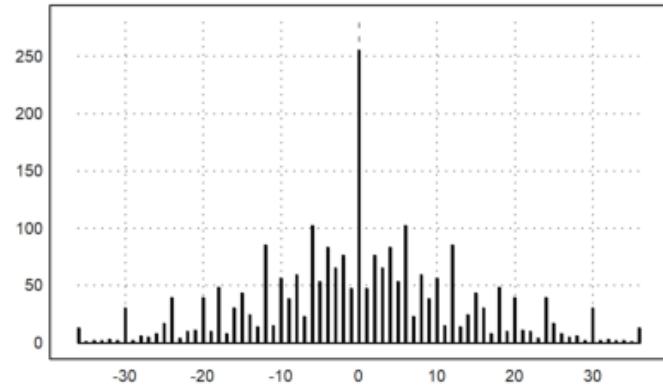
1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Dalam PDF pengantar untuk R disini terdapat sebuah contoh, yang mana menghitung distribusi dari ab-cd untuk a,b,c,d dipilih dari 0 ke n secara acak. Solusi dalam R adalah bentuk sebuah matriks 4-dimensi dan menjalankan table() setelah itu.

Tentu saja, ini dapat tercapai dengan sebuah putaran. Tetapi loop tidak efektif dalam EMT atau R. Dalam EMT, kami dapat menuliskan loop dalam C dan akan menjadi solusi tercepat.

Tetapi kami ingin menyerupai perilaku dari R. Untuk ini, kami butuh untuk meratakan perkalian ab dan membuat sebuah matriks dari 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"):
```



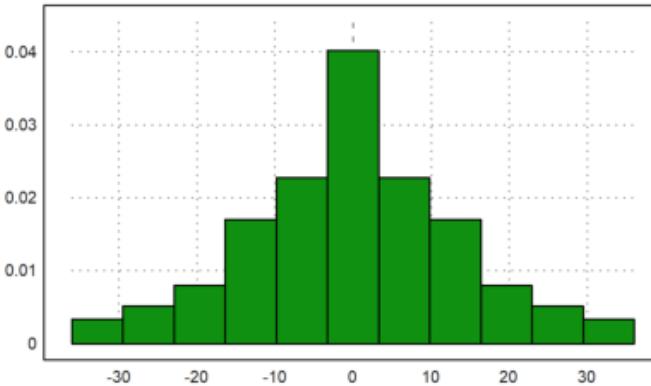
Disamping perkalian eksak, EMT dapat menghitung frekuensi dalam vektor-vektor.

```
>getfrequencies(q,-50:10:50)
```

```
[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]
```

Cara termudah untuk plot ini seperti sebuah distribusi berikut.

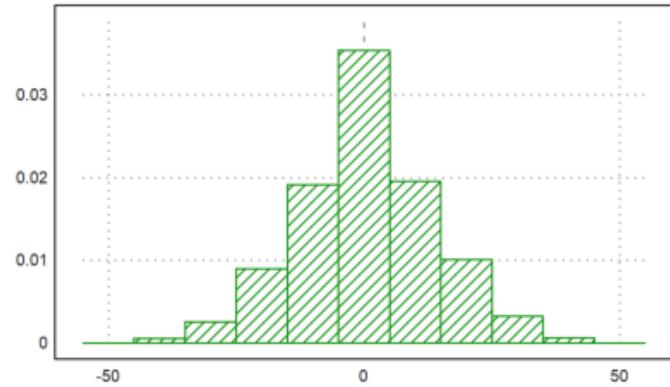
```
>plot2d(q,distribution=11):
```



Tetapi ini juga dapat untuk pra-komputasi menghitung dalam interval yang dipilih sebelumnya. Tentu saja, berikut menggunakan getfrequencies() secara internal.

Karena fungsi histo() mengembalikan frekuensi, kami butuh untuk men-skakukan ini jadi bahwa integral dibawah 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 macam list. Satu dari list global yang mana mutable dan lainnya adalah sebuah tipe list yang mana immutable. Kami tidak peduli tentang list global disini.

Tipe list immutable dinamakan sebuah collection dalam EMT. Ini berperilaku seperti struktur dalam C, tetapi elemen-elemenya hanya diberi angka dan tidak dinamai.

```
>L={"Fred","Flintstone",40,[1990,1992]}
```

```
Fred  
Flintstone  
40  
[1990, 1992]
```

Sekarang elemen-elemen tidak memiliki nama-nama, melalui nama dapat diatur untuk tujuan khusus. Mereka dapat diakses dengan angka.

```
>(L[4])[2]
```

```
1992
```

## Input dan Output File (Membaca dan Menulis Data)

---

Kamu akan sering ingin untuk import sebuah matriks data dari sumber lainnya ke EMT. Tutorial ini memberitahu kamu tentang banyak cara untuk meraihnya. Fungsi sederhana adalah writematrix() dan readmatrix().

Mari kita demonstrasikan bagaimana membaca dan menulis sebuah vektor ke file yang sebenarnya.

```
>a=random(1,100); mean(a), dev(a),
```

```
0.49815  
0.28037
```

Untuk menulis data ke sebuah file, kami menggunakan fungsi writematrix().

Karena pengantar ini banyak seperti dalam direktori, dimana pengguna tidak memiliki akses, kami menulis data ke direktori pengguna home. Untuk notebook sendiri, ini tidak dibutuhkan, karena file data akan ditulis kedalam direktori yang sama.

```
>filename="test.dat";
```

Sekarang kami menulis vektor kolom a' kedalam file. Ini menghasilkan angka satu dalam setiap baris dari file.

```
>writematrix(a',filename);
```

Untuk membaca data, kami gunakan readmatrix().

```
>a=readmatrix(filename)';
```

Dan menghapus file.

```
>fileremove(filename);
>mean(a), dev(a),
```

```
0.49815
0.28037
```

Fungsi writematrix() atau writetable() dapat diatur untuk bahasa lain.

Sebagai contoh, jika kamu menggunakan sistem Indonesia (titik desimal dengan koma), Excel kamu harusnya memiliki nilai dengan desimal koba dipisahkan dengan titik koma dalam sebuah csv (se secara bawaan koma memisahkan nilai) file. File berikut "test.csv" harusnya tampak dalam folder kamu sekarang.

```
>filename="test.csv"; ...
>writematrix(random(5,3),file=filename,separator=",");
```

Kamu sekarang dapat membuka file ini dengan Excel Indonesia secara langsung.

```
>fileremove(filename);
```

Terkadang kami memiliki strings dengan token seperti berikut.

```
>s1:="f m m f m m m f f f m m f"; ...
>s2:="f f f m m f f";
```

Untuk membuat token ini, kami mendefinisikan sebuah vektor dari token-token.

```
>tok:=[ "f" , "m" ]
```

f

m

Lalu kami dapat menghitung angka dari banyaknya token yang muncul dalam string, dan mengambil hasilnya kedalam sebuah tabel.

```
>M:=getmultiplicities(tok,strtokens(s1))_ ...
>  getmultiplicities(tok,strtokens(s2));
```

Tulis tabel dengan header token.

```
>writetable(M,labc=tok,labr=1:2,wc=8)
```

	f	m
1	6	7
2	5	2

Untuk statistik, EMT dapat membaca dan menulis tabel.

```
>file="test.dat"; open(file,"w"); ...
>writeln("A,B,C"); writematrix(random(3,3)); ...
>close();
```

File terlihat seperti ini.

```
>printfile(file)
```

```
A,B,C
0.7003664386138074,0.1875530821001213,0.3262339279660414
0.5926249243193858,0.1522927283984059,0.368140583062521
0.8065535209872989,0.7265910840408142,0.7332619844597152
```

fungsi `readtable()` dalam bentuk yang sederhana dapat membaca ini dan mengembalikan sebuah collection dari nilai-nilai dan garis heading.

```
>L=readtable(file,>list);
```

Collection ini dapat mencetak dengan `writetable()` untuk notebook atau ke dalam sebuah file.

```
>writetable(L,wc=10,dc=5)
```

A	B	C
0.70037	0.18755	0.32623
0.59262	0.15229	0.36814
0.80655	0.72659	0.73326

Nilai-nilai matriks adalah elemen pertama dari L. Catatan bahwa `mean()` dalam EMT menghitung nilai mean dari baris-baris dari sebuah matriks.

```
>mean(L[1])
```

```
0.40472  
0.37102  
0.75547
```

## File-File CSV

---

Pertama, mari kita tulis sebuah matriks kedalam sebuah file. Untuk keluaran, kami menghasilkan sebuah file dalam direktori sekarang yang dikerjakan.

```
>file="test.csv";  ...
>M=random(3,3); writematrix(M,file);
```

Here is the content of this file.

```
>printfile(file)
```

```
0.8221197733097619,0.821531098722547,0.7771240608094004
0.8482947121863489,0.3237767724883862,0.6501422353377985
0.1482301827518109,0.3297459716109594,0.6261901074210923
```

CSV ini dapat dibuka dalam sistem English kedalam Excel dengan sebuah klik dobel. Jika kamu mendapatkan sebuah file dalam sistem German, perlu untuk mengimpor data ke Excel untuk berhati hati dalam titik desimal.

Tetapi titik desimal adalah format bawaan untuk EMT juga. Kamu dapat membaca matriks dari sebuah file dengan `readmatrix()`.

```
>readmatrix(file)
```

```
0.82212  0.82153  0.77712  
0.84829  0.32378  0.65014  
0.14823  0.32975  0.62619
```

Ini memungkinkan untuk menulis beberapa matriks kedalam satu file. Perintah `open()` dapat membuka file untuk menulisnya dengan parameter "w". Secara bawaan adalah "r" untuk membaca.

```
>open(file,"w"); writematrix(M); writematrix(M'); close();
```

Matriks-matriks dipisahkan dengan baris kosong. Untuk membaca matriks-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
```

Dalam Excel atau spreadsheet yang sama, kamu dapat mengexport sebuah matriks sebagai CSV(comma separated values). Dalam Excel 2007, gunakan "save as" dan "other formats", lalu pilih "CSV". Pastikan, tabel sekarang hanya memiliki daya yang ingin kamu ekspor.

Ini sebagai contohnya.

```
>printfile("excel-data.csv")
```

```
0;1000;1000
1;1051,271096;1072,508181
2;1105,170918;1150,273799
3;1161,834243;1233,67806
4;1221,402758;1323,129812
5;1284,025417;1419,067549
6;1349,858808;1521,961556
7;1419,067549;1632,31622
8;1491,824698;1750,6725
9;1568,312185;1877,610579
10;1648,721271;2013,752707
```

Seperti yang dapat kamu lihat, sister German saya menggunakan titik koma sebagai pemisah dan sebuah koma desimal. Kamu dapat merubah ini dalam peraturan sistem atau dalam Excel, tetapi ini tidak dibutuhkan untuk membaca matriks ke dalam EMT.

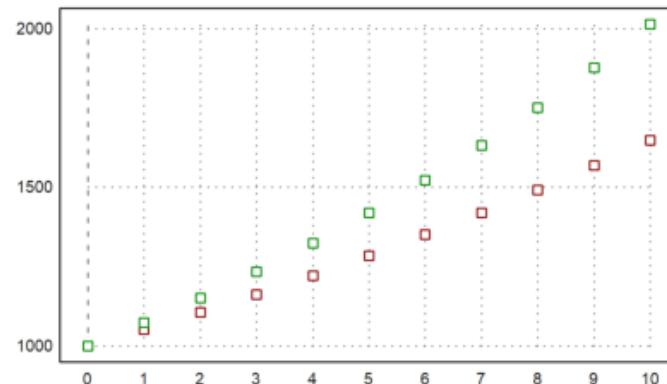
Cara termudah untuk membaca ini kedalam Euler adalah readmatrix(). Semua komma digantikan dengan titik dengan parameter >comma. Untuk CSV English, secara sederhana buang parameter ini.

```
>M=readmatrix("excel-data.csv",>comma)
```

0	1000	1000
1	1051.3	1072.5
2	1105.2	1150.3
3	1161.8	1233.7
4	1221.4	1323.1
5	1284	1419.1
6	1349.9	1522
7	1419.1	1632.3
8	1491.8	1750.7
9	1568.3	1877.6
10	1648.7	2013.8

Mari kita buat plot ini.

```
>plot2d(M'[1],M'[2:3],>points,color=[red,green]):
```



Banyak cara yang lebih mendasar untuk membaca data dari sebuah file. Kamu dapat membuka file dan membaca angka per baris. Fungsi getvectorline() akan membaca angka-angka dari sebuah baris dari data. Secara bawaan, ini mempunyai harapan sebuah titik desimal. Tetapi ini dapat juga menggunakan sebuah koma desimal, jika kamu memanggil setdecimaldot(”,”) sebelum kamu menggunakan fungsi ini.

Fungsi berikut merupakan sebuah contoh untuk ini. Ini akan berhenti pada akhir file atau baris kosong.

```
>function myload (file) ...  
  
    open(file);  
    M=[];  
    repeat  
        until eof();  
        v=getvectorline(3);  
        if length(v)>0 then M=M_v; else break; endif;  
    end;  
    return M;  
    close(file);  
endfunction
```

```
>myload(file)
```

0.82212	0.82153	0.77712
0.84829	0.32378	0.65014
0.14823	0.32975	0.62619

Ini juga akan memungkinkan membaca semua angka dalam file tersebut dengan getvector().

```
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
```

```
0.82212  0.82153  0.77712  
0.84829  0.32378  0.65014  
0.14823  0.32975  0.62619
```

Lalu ini sangat mudah untuk menyimpan sebuah vektor dari nilai-nilai, satu nilai setiap baris dan membaca kembali vektor ini.

```
>v=random(1000); mean(v)
```

```
0.50303
```

```
>writematrix(v',file); mean(readmatrix(file)')
```

```
0.50303
```

## Menggunakan Tabel Tabel

---

Tabel dapat digunakan untuk membaca atau menulis data numerik. Sebagai contoh, kami menulis sebuah tabel dengan baris dan headers 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
0.09,	0.39,	0.86
0.39,	0.86,	0.71
0.2,	0.02,	0.83

Ini dapat diimpor ke Excel.

Untuk membaca file dalam EMT, kami menggunakan readtable().

```
>{M,headings}=readtable(file,>clabs); ...
>writetable(M,labc=headings)
```

one	two	three
0.09	0.39	0.86
0.39	0.86	0.71
0.2	0.02	0.83

## Menganalisis Sebuah Garis

---

Kamu bahkan dapat mengevaluasi setiap baris dengan tangan. Andaikan, kami memiliki baris dengan format berikut.

```
>line="2020-11-03,Tue,1'114.05"
```

2020-11-03, Tue, 1'114.05

Pertama kami dapat membuat token untuk baris.

```
>vt=strtokens(line)
```

2020-11-03  
Tue  
1'114.05

Lalu kami dapat mengevaluasi setiap elemen dari baris menggunakan evaluasi yang dibutuhkan.

```
>day(vt[1]), ...  
>indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])), ...  
>strrepl(vt[3],'"','"')()
```

```
7.3816e+05  
2  
1114
```

Menggunakan ekspresi reguler, ini memungkinkan untuk mengekstrak hampir sebarang informasi dari sebuah baris data.

Asumsikan kita memiliki baris berikut dalam dokumen HTML.

```
>line=<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>"
```

```
<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>
```

Untuk mengekstrak ini, kami menggunakan regular expression, yang mana mencari

- sebuah braket tertutup >,
- sebarang string yang tidak memiliki bracket dengan sebuah sub-cocok

”(...)”,

- sebuah pembuka dan penutup bracket menggunakan solusi terpendek,
- kembali sebarang string uang tidak memiliki bracket,
- dan sebuah bracket terbuka <.

Regular expression merupakan hal yang sulit untuk dipelajari tetapi sangat kuat.

```
>{pos,s,vt}=strxfind(line,>([<>]+)<.+?>([<>]+)<" );
```

Hasil merupakan posisi yang cocok, string yang cocok dan sebuah vektor string untuk sub-cocok.

```
>for k=1:length(vt); vt[k](), end;
```

```
1145.5  
5.6
```

Ini merupakan sebuah fungsi, yang mana membaca bagian bagian numerik diantara <td> dan </td>.

```
>function readtd (line) ...
v=[]; cp=0;
repeat
    {pos,s,vt}=strxfind(line,"<td.*?>(.+?)</td>",cp);
    until pos==0;
    if length(vt)>0 then v=v|vt[1]; endif;
    cp=pos+strlen(s);
end;
return v;
endfunction
```

```
>readtd(line+"<td>non-numerical</td>")
```

```
1145.45
5.6
-4.5
non-numerical
```

Sebuah web site atau sebuah file dengan sebuah URL dapat dibuka dalam EMT dan dapat dibaca baris per baris.

Dalam contoh, kami membaca versi sekarang dari situs EMT. Kami menggunakan regular expression untuk mengamati "Version ..." dalam sebuah heading.

```
>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

## Variabel Input dan Output

---

Kamu dapat menulis sebuah variabel dalam bentuk definisi Euler kedalam file atau ke baris perintah.

```
>writevar(pi,"mypi");
```

```
mypi = 3.141592653589793;
```

Untuk sebuah pengujian, kami membuat sebuah file Euler dalam direktori bekerja dari EMT.

```
>file="test.e"; ...
>writevar(random(2,2),"M",file); ...
>printfile(file,3)
```

```
M = [ ..
0.5991820585590205, 0.7960280262224293;
0.5167243983231363, 0.2996684599070898];
```

Kami sekarang dapat memuat file. Ini dapat didefinisikan matriks M.

```
>load(file); show M,
```

```
M =  
 0.59918  0.79603  
 0.51672  0.29967
```

Omong-omong, jika writevar() digunakan dalam sebuah variabel, ini akan mencetak definisi variabel dengan nama dari variabel ini.

```
>writevar(M); writevar(inch$)
```

```
M = [ ..  
 0.5991820585590205, 0.7960280262224293;  
 0.5167243983231363, 0.2996684599070898];  
inch$ = 0.0254;
```

Kami dapat juga membuka file baru atau menambahkan file yang telah ada. Dalam contoh kami menambahkan file yang dibuat sebelumnya.

```
>open(file,"a"); ...
>writevar(random(2,2),"M1"); ...
>writevar(random(3,1),"M2"); ...
>close();
>load(file); show M1; show M2;
```

```
M1 =
 0.30287  0.15372
 0.7504   0.75401
M2 =
 0.27213
 0.053211
 0.70249
```

Untuk menghapus sebarang file gunakan fileremove().

```
>fileremove(file);
```

vektor baris dalam sebuah file tidak butuh koma, jika setiap angka dalam baris baru. Mari kita buat sedemikian hingga sebuah file, menulis setiap baris satu per satu dengan writeln().

```
>open(file,"w"); writeln("M = ["); ...
>for i=1 to 5; writeln(""+random()); end; ...
>writeln("]"); close(); ...
>printfile(file)
```

```
M = [
0.344851384551
0.0807510017715
0.876519562911
0.754157709472
0.688392638934
];
```

```
>load(file); M
```

```
[0.34485, 0.080751, 0.87652, 0.75416, 0.68839]
```

## Latihan

---

### Latihan 1

---

diberikan lemparan dadu acak sebanyak 1200 kali, tentukan nilai chi-kuadratnya apabila nilai harapan dadunya memiliki distribusi seragam

```
>throws=1200; sides=6;
>randomDice = intrandom(throws, sides);
>freqDice = multofsoted(sort(randomDice), 1:sides);
>expectDice = dup(throws/sides, sides)';
>chitest(freqDice, expectDice)
```

0.91075

Karena error memiliki nilai yang besar (lebih dari 0.05) maka dapat disimpulkan bahwa kita tidak bisa menolak distribusi seragam. Ini belum membuktikan bahwa dadu kita adil tetapi kita tidak bisa menolak hipotesis kita

### Latihan 2:

---

Diberikan hasil dari lemparan koin sebagai berikut, dengan G adalah gambar dan A adalah angka  
G, A, A, G, A, A, A, G, A, A  
Apakah koin tersebut adil?

```
>coinRes = "G, A, A, G, A, A, A, G, A, A";
```

buat fungsi untuk memisahkan setiap token

```
>function replaceTokenToArr(strToken, separateToken, arrToken, arrSubsToken)...
```

```
tokens = strtokens(strToken, separateToken);
transformedArr = [];
for i = 1 to length(tokens) step i
    token = tokens[i];
    indexToken = indexof(arrToken, token);
    subs = arrSubsToken[indexToken];
    transformedArr = [transformedArr, subs];
end;
return transformedArr;
endfunction
```

```
>coinDis = replaceTokenToArr(coinRes, " ", ["G", "A"], [1, 2])
```

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

Tambahkan fungsi untuk menghitung setiap frekuensi

```
>function getFrequencies(arr)...  
  
    uniqueArr = unique(arr);  
    sortArr = sort(arr);  
    firstIndexes = [];  
    freqs = [];  
    for i = 1 to length(uniqueArr) step i  
        currUnique = uniqueArr[i];  
        currFirstIndex = indexof(sortArr, currUnique) - 1;  
        firstIndexes = [firstIndexes, currFirstIndex];  
    end;  
    firstIndexes = [firstIndexes, length(arr)];  
    for i = 1 to (length(firstIndexes) - 1) step i  
        currIndex = firstIndexes[i];  
        nextIndex = firstIndexes[i + 1];  
        currFreq = nextIndex - currIndex;  
        freqs = [freqs, currFreq];  
    end;  
    return freqs;  
endfunction
```

```
>coinFreq = getFrequencies(coinDis)
```

[3, 7]

Didapatkan frekuensi untuk A sebanyak 3 dan G sebanyak 7

```
>coinExpected = dup(sum(coinFreq)/length(coinFreq), length(coinFreq))'
```

```
[5, 5]
```

```
>chitest(coinFreq, coinExpected)
```

```
0.2059
```

Karena nilai error dari chi-kuadratnya besar (lebih dari 0.05), maka kita dapat menolak hipotesis bahwa koin tersebut adil (setimbang).

---

**Latihan 3**

Diberikan tabel yang menyatakan hubungan kondisi keluarga dengan perilaku delinquensi anak pada sebuah SD

Perilaku Delinquensi | Nakal | Nurut

Harmonis | 25 | 85

Broken | 75 | 15

Apakah terdapat hubungan antara kondisi keluarga dengan perilaku delinquensi anak?

```
>delinquencyChilds = [25, 85; 75, 15];
>writetable(delinquencyChilds, wc=10, labr=["Harmonis", "Broken"], labc=["Nakal", "Nurut"])
```

	Nakal	Nurut
Harmonis	25	85
Broken	75	15

Periksa terlebih dahulu dengan uji tabel chi-kuadrat

```
>tabletest(delinquencyChilds)
```

0

Dimana hasilnya kecil, kita tidak bisa menentukan apakah menolak saling asing antara data yang diberikan.

Lalu, dapat dibuat tabel harapan dari matriks tersebut, untuk melihat harapan yang ada dalam setiap barisnya.

```
>writetable(expectedtable(deliquencyChilds), wc=10, dc=1, labr=["Harmonis", "Broken"], labc=["Nakal",
```

	Nakal	Nurut
Harmonis	55	55
Broken	45	45

Selanjutnya, menghitung nilai koefisien kontingensi dari tabel tersebut

```
>contingency(deliquencyChilds)
```

0.7303

## Latihan 4

---

Penelitian dilakukan untuk mengetahui partisipasi warga dalam suatu kepala desa yang dilihat dari jenis kelamin. Hasil penelitian tersebut diperoleh data sebagai berikut

Jenis Kelamin | Ikut | Tidak

Pria | 10 | 5

Wanita | 9 | 6

Apakah ada pengaruh jenis kelamin terhadap keikutseraan dalam pemilihan kepala desa?

```
>absenceGenders = [10,5;9,6]
```

10	5
9	6

Buat tabel agar mudah melihatnya

```
>writetable(absenceGenders, wc = 10, labr=["Pria", "Wanita"], labc = ["Ikut", "Tidak"])
```

	Ikut	Tidak
Pria	10	5
Wanita	9	6

Memeriksa tabel untuk uji chi-kuadratnya

```
>tabletest(absenceGenders)
```

0.70479

Karena nilai terlalu besar, maka belum bisa menyimpulkan apakah terdapat hubungan antara keikutsertaan dengan gender

Lalu, dibuat tabel nilai harapan untuk melihat nilai harapan pada tabel tersebut

```
>writetable(expectedtable(absenceGenders), dc=1, wc=10, labr=["Pria", "Wanita"], labc=["Ikut", "Tidak"])
```

	Ikut	Tidak
Pria	9.5	5.5
Wanita	9.5	5.5

Terakhir, adalah menghitung nilai koefisien kontingensi

```
>contingency(absenceGenders)
```

0.09759

Karena nilai koefisien kontingensi mendekati 0, maka dapat disimpulkan bahwa hubungan antara keikutsertaan dengan jenis kelamin tidak ada (sangat lemah). Terlihat bahwa koefisien korelasi kontingensinya mendekati 1, artinya bahwa variabel kategorik ini saling bergantung dimana hubungan antara kondisi keluarga dengan perilaku delinquensi anak memiliki hubungan yang kuat.

Diperkirakan rata-rata berat badan siswa kelas 2 SMA adalah kurang dari 55 kg. Akan tetapi untuk melakukan pembuktian perkiraan tersebut diambil data dan didapat sebagai berikut

berat badan:

45, 60, 65, 55, 65, 60, 50, 70, 60, 60

Apakah perkiraan ini benar?

Penyelesaian

Rumusan Hipotesis

Parameter populasi yang akan diuji disini adalah berat badan dari populasi siswa kelas 2 SMA

$$\begin{array}{ll} \text{Hipotesis nol} & H_0 : \mu \leq 55. \\ \text{Hipotesis alternatif} & H_1 : \mu > 55. \end{array}$$

```
>weights = [45, 60, 65, 55, 65, 60, 50, 70, 60, 60];
>meanEst = 55;
>dev(weights)
```

7.3786

```
>ttest(mean(weights), dev(weights), length(weights), meanEst)
```

0.06031

$$t_{\text{hitung}} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} = \frac{59 - 55}{7,37/\sqrt{9}} \approx 1,63.$$

Daerah kritis:

Uji yang dilakukan diatas merupakan uji satu arah (kanan). Dengan menggunakan tabel-t, nilai-t, dengan taraf signifikansi 0.05 dan derajat kebebasan  $n - 1 = 10 - 1 = 9$ , maka

$$t_{\text{tabel}} = t_{0,05; 9} \approx 1,833.$$

Dengan demikian, diperoleh bahwa daerah kritis terletak di

$$t > 1,83.$$

Keputusan:

Karena

$$t_{\text{hitung}} = 1,63 < 1,833 = t_{\text{tabel}},$$

disimpulkan bahwa statistik uji akan terdapat diluar daerah kritis. Dengan demikian hipotesis nol tidak ditolak

Kesimpulan:

Dugaan bahwa perkiraan rata-rata berat badan siswa kelas 2 SMA kurang dari 55 kg akan di terima

---

## Latihan 6

Berdasarkan 100 laporan kematian di AS yang diambil secara acak, diperoleh bahwa rata-rata usia hidup orang AS adalah 71.8 tahun dengan simpangan baku 8.9 tahun. Hal ini memberikan dugaan bahwa rata-rata usia hidup orang AS lebih dari 70 tahun. Dengan taraf signifikansi 5% ujilah kebenaran dugaan tersebut.

Penyelesaian

Rumusan Hipotesis

Parameter populasi yang akan diuji disini adalah rata-rata hidup dari populasi orang AS

$$\begin{array}{ll} \text{Hipotesis nol} & H_0 : \mu \leq 70. \\ \text{Hipotesis alternatif} & H_1 : \mu > 70. \end{array}$$

```
>nReport = 100; meanLifeExpect = 71.8; devLife = 8.9; meanLifeEstimate = 70;  
>ttest(meanLifeEstimate, devLife, nReport, meanLifeExpect)
```

0.022912

$$t_{\text{hitung}} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} = \frac{71,8 - 70}{8,9/\sqrt{100}} \approx 2,02.$$

Daerah kritis:

Uji yang dilakukan diatas merupakan uji satu arah (kanan). Dengan menggunakan tabel-t, nilai-t, dengan taraf signifikansi 0.05 dan derajat kebebasan  $n - 1 = 100 - 1 = 99$ , maka

$$t_{\text{tabel}} = t_{0,05; 99} \approx 1,66.$$

Dengan demikian, diperoleh bahwa daerah kritis terletak di

$$t > 1,66.$$

Keputusan:  
Karena

$$t_{\text{hitung}} = 2,02 > 1,66 = t_{\text{tabel}},$$

disimpulkan bahwa statistik uji akan terdapat pada daerah kritis. Dengan demikian hipotesis nol akan ditolak

Kesimpulan:

Dugaan bahwa rata-rata usia hidup orang AS lebih dari 70 tahun diterima.

Kesimpulan dari fungsi:

Karena nilai  $t_{\text{test}} < 0.05$ , maka hipotesis nol akan ditolak

---

## Latihan 7

Seorang perawat ingin mengetahui apakah terdapat perbedaan antara lama hari dirawat dengan kelas tempat dirawat, yang mana terdapat kelas I, II, dan kelas III, sebagai berikut

Pasien | Kelas I | Kelas II | Kelas III

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

Penyelesaian

Rumusan Hipotesis

Parameter populasi yang akan diuji disini adalah rata-rata lama perawatan dari masing-masing kelas

$$\begin{array}{ll} \text{Hipotesis nol} & H_0 : k_i = 0, i = k_1, k_2, k_3 \\ \text{Hipotesis alternatif} & H_1 : \text{setidaknya ada } k_i \neq 0 \end{array}$$

```
>classOne = [5,3,2,6,3];
>classTwo = [2,6,4,2];
>classThree = [3,6,7,3,8];
>varanalysis(classOne, classTwo, classThree)
```

0.32137

Karena nilai dari fungsi  $> 0.05$ , maka kita terima hipotesis nol, yang artinya tidak ada perbedaan antara lama hari dirawat dengan kelas tempat pasien dirawat

## Latihan 8

---

Terdapat tiga kelompok subjek penelitian untuk menguji metode pengajaran mana yang paling baik dalam meningkatkan pembelajaran. Metode pertama adalah ceramah, kedua adalah diskusi dan ketiga adalah praktek

Penelitian tersebut menghasilkan data sebagai berikut.

Ceramah | Diskusi | Praktek

25 | 17 | 26

11 | 16 | 20

16 | 18 | 17

26 | 20 | 26

32 | 10 | 43

25 | 14 | 46

30 | 19 | 35

17 | 34

Penyelesaian

Rumusan Hipotesis

Parameter populasi yang akan diuji disini adalah hasil peningkatan pembelajaran untuk metode-metode diatas

$$\begin{array}{ll} \text{Hipotesis nol} & H_0 : m_i = 0, i = m_1, m_2, m_3 \\ \text{Hipotesis alternatif} & H_1 : \text{setidaknya ada } m_i \neq 0 \end{array}$$

```
>lecture = [25,11,16,26,32,25,30,17];
>discuss = [17,16,18,20,10,14,19];
>practice = [26,20,17,26,43,46,35,34];
>varanalysis(lecture, discuss, practice)
```

0.0060525

karena hasil fungsi  $< 0.05$ , maka kita tolak hipotesis nol, yang artinya bahwa hasil peningkatan pembelajaran setelah melalui metode-metode tersebut memiliki hasil yang berbeda.