Kegiatan Belajar Menggunakan LaTeX
dan Markdown untuk Pengolahan Dokumen

Anisah Daffa Citra Nareswari
22305141044
Matematika E 2022

PRODI MATEMATIKA
DEPARTEMEN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2023

DAFTAR ISI

BAB 1

MENGGUNAKAN EMT UNTUK MENYELESAIKAN MASALAH-MASALAH ALJABAR

[a4paper,10pt]article eumat

Aplikasi Komputer

Nama : Anisah Daffa Citra Nareswari
NIM : 22305141044
Kelas : Matematika E 2022

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))))
```

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

Baris Perintah

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

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

```
16.7551608191
```

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

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

```
50.2654824574
100.530964915
```

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

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

```
0.540302305868
```

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

```
0.857553215846
```

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

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

```
1.41666666667
1.41421568627
1.41421356237
```

Ini juga merupakan cara yang baik untuk menyebarkan perintah panjang pada dua atau lebih baris. Anda dapat menekan Ctrl+Return untuk membagi garis menjadi dua pada posisi kursor saat ini, atau Ctrl+Back untuk menggabungkan garis.
Untuk melipat semua multi-garis tekan Ctrl + L. Kemudian garis-garis berikutnya hanya akan terlihat, jika salah satunya memiliki fokus. Untuk melipat satu multi-baris, mulailah baris pertama dengan "%+".

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

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

```
81
```

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

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

```
1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237
```

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

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

```
1.41421356237
```

Struktur bersyarat juga berfungsi.

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

```
Thought so!
```

Saat Anda menjalankan perintah, kursor dapat berada di posisi mana pun di baris perintah. Anda dapat kembali ke perintah sebelumnya atau melompat ke perintah berikutnya dengan tombol panah. Atau Anda dapat mengklik ke bagian komentar di atas perintah untuk menuju ke perintah.
Saat Anda menggerakkan kursor di sepanjang garis, pasangan tanda kurung atau kurung buka dan tutup akan disorot. Juga, perhatikan baris status. Setelah kurung buka fungsi sqrt(), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol kembali.

```
>sqrt(sin(10°)/cos(20°))
```

```
0.429875017772
```

Untuk melihat bantuan untuk perintah terbaru, buka jendela bantuan dengan F1. Di sana, Anda dapat memasukkan teks untuk dicari. Pada baris kosong, bantuan untuk jendela bantuan akan ditampilkan. Anda dapat menekan escape untuk menghapus garis, atau untuk menutup jendela bantuan.
Anda dapat mengklik dua kali pada perintah apa pun untuk membuka bantuan untuk perintah ini. Coba klik dua kali perintah exp di bawah ini di baris perintah.

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

```
2.5
```

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

Sintaks Dasar

Euler tahu fungsi matematika biasa. Seperti yang Anda lihat di atas, fungsi trigonometri bekerja dalam radian atau derajat. Untuk mengonversi ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilainya, atau gunakan fungsi rad(x). Fungsi akar kuadrat disebut kuadrat dalam Euler. Tentu saja, x^(1/2) juga dimungkinkan.
Untuk menyetel variabel, gunakan "=" atau ":=". Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir. Spasi tidak masalah. Tapi ruang antara perintah diharapkan.
Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma menekan output dari perintah. Di akhir baris perintah "," diasumsikan, jika ";" hilang.

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

```
30.65625
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Memasuki

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

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

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

```
8.77908249441
```

Untuk menghitung ekspresi rumit seperti

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

Anda harus memasukkannya dalam bentuk baris.

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

```
23.2671801626
```

Letakkan tanda kurung dengan hati-hati di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyorot ekspresi bahwa braket penutup selesai. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi.
Hasil dari perhitungan ini adalah bilangan floating point. Secara default dicetak dengan akurasi sekitar 12 digit. Di baris perintah berikut, kita juga belajar bagaimana kita bisa merujuk ke hasil sebelumnya dalam baris yang sama.

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

```
0.47619047619
10/21
```

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

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

```
14.4978445072
```

Operator numerik Euler meliputi

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

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

```
sin,cos,tan,atan,asin,acos,rad,deg
log,exp,log10,sqrt,logbase
bin,logbin,logfac,mod,lantai,ceil,bulat,abs,tanda
conj,re,im,arg,conj,nyata,kompleks
beta,betai,gamma,complexgamma,ellrf,ellf,ellrd,elle
bitand, bitor, bitxor, bitnot
```

Beberapa perintah memiliki alias, mis. Untuk log.

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

```
2
0.5
```

```
>sin(30°)
```

```
0.5
```

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

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

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

Bilangan Asli

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

```
>longest 1/3
```

```
     0.3333333333333333
```

Representasi ganda internal membutuhkan 8 byte.

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

```
1.0101010101010101010101010101010101010101010101010101*2^-2
```

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

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

String

Sebuah string dalam Euler didefinisikan dengan "...".

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

```
A string can contain anything.
```

String dapat digabungkan dengan | atau dengan +. Ini juga berfungsi dengan angka, yang dikonversi menjadi string dalam kasus itu.

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

```
The area of the circle with radius 2 cm is 12.5663706144 cm^2.
```

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

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

```
Golden Ratio : 1.61803
```

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

```
>none
```

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

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

```
1234.5
```

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

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

```
affe
charlie
bravo
```

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

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

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

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

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

```
 = 45°
```

I

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

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

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

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

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

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

```
Ü is a German letter
```

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

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

```
We have =.
```

Dimungkinkan juga untuk menggunakan entitas numerik.

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

```
Ähnliches
```

Nilai Boolean

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

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

```
0
1
```

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

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

```
1
```

Operator Boolean mematuhi aturan bahasa matriks.

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

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

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

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

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

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

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

Format Keluaran

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

```
>defformat; pi
```

```
3.14159265359
```

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

```
>longest pi
```

```
      3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhex(pi)
```

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

Format output dapat diubah secara permanen dengan perintah format.

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

```
      0.33333
      3.14159
      0.84147
```

Standarnya adalah format (12).

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

```
  0.333333333333
```

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

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

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

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

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

```
  3.1416
```

Fungsi "format terpanjang" mengatur format skalar juga.

```
>longestformat; pi
```

```
  3.141592653589793
```

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

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

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

```
>longestformat; pi,
```

```
  3.141592653589793
```

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

```
    3.14159
```

Standarnya adalah defformat().

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

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

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

```
      4.934802200544679
```

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

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

```
25/12
```

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

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

```
 -1.110223024625157e-16
```

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

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

```
0
```

Ekspresi

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

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

```
      12.56637061435917
```

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

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

```
-0.919535764538
```

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

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

```
36
```

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

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

```
45
```

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

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

```
45
```

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

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

```
12
```

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

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

$$x^x\,\left(\log x+1\right)$$

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

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

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

Ini memungkinkan untuk memanipulasi ekspresi dalam variabel lain untuk fungsi EMT yang membutuhkan ekspresi dalam "x".
Cara paling dasar untuk mendefinisikan fungsi sederhana adalah dengan menyimpan rumusnya dalam ekspresi simbolis atau numerik. Jika variabel utama adalah x, ekspresi dapat dievaluasi seperti fungsi.
Seperti yang Anda lihat dalam contoh berikut, variabel global terlihat selama evaluasi.

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

```
  -0.475
```

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

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

```
  -0.425
```

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

Matematika Simbolik

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

```
>$&44!
```

$$2658271574788448768043625811014615890319638528000000000$$

Dengan cara ini, Anda dapat menghitung hasil yang besar dengan tepat. Mari kita hitung
lateks: C(44,10) = \frac{44!}{34! \cdot 10!}

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

$$2481256778$$

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

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

$$2481256778$$

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

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

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

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

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

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

$$120$$

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

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

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

Ekspresi simbolik diuraikan oleh Euler. Jika Anda membutuhkan sintaks yang kompleks dalam satu ekspresi, Anda dapat menyertakan ekspresi dalam "...". Untuk menggunakan lebih dari ekspresi sederhana adalah mungkin, tetapi sangat tidak disarankan.

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

```
                                  25
```

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

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

$$x^4+4\,x^3+6\,x^2+4\,x+1$$

$$4\,\left(x+1\right)^3$$

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))
```

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

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

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

```
                          2432902008176640000
```

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

```
                                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 sintaks asli Maxima. Anda dapat melakukannya dengan ":::".

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

```
                                 2
                                g
```

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

```
                              3  x
                             x  E
```

$$x^3\,e^x$$

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

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

```
                                   5
                             125 E
```

$$125\,e^5$$

```
                        18551.64488782208
```

```
>fx(5)
```

```
  18551.6448878
```

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

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

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

```
                       2.20079141499189e+7
```

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

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

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

```
>tex(fx)
```

```
x^3\,e^{x}
```

Ekspresi simbolik dapat dievaluasi seperti ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

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

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

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

Penugasan juga bisa bersifat simbolis.

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

$$\left(t+1\right)^3\,e^{t+1}$$

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

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

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

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

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

```
1.56155281281
```

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

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

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

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

$$\left[x=-3.23606797749979,x=1.23606797749979\right]$$

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

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

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

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

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

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

$$\left[\left[x=2,y=1\right],\left[x=1,y=2\right]\right]$$

$$2$$

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

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

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

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

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

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

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

Fungsi

Dalam EMT, fungsi adalah program yang didefinisikan dengan perintah "fungsi". Ini bisa berupa fungsi satu baris atau fungsi multibaris.
Fungsi satu baris dapat berupa numerik atau simbolis. Fungsi satu baris numerik didefinisikan oleh ":=".

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

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

```
>f(2)
```

```
4.472135955
```

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

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

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

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

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

```
0.786151377757
```

Secara default, jika Anda perlu menimpa fungsi bawaan, Anda harus menambahkan kata kunci "menimpa". Menimpa fungsi bawaan berbahaya dan dapat menyebabkan masalah untuk fungsi lain tergantung pada fungsi tersebut.
Anda masih dapat memanggil fungsi bawaan sebagai "_...", jika itu adalah fungsi di inti Euler.

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

```
0.707106781187
```

Lebih baik kita menghapus redefinisi dosa ini.

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

```
0.707106781187
```

Parameter Default

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

    16

Menyetelnya akan menimpa nilai default.

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

    80

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

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

    16

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

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

    24

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

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

    20

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

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

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

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

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

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

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

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

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

```
178.635099908
```

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik lainnya.

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

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

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

```
0.703467422498
```

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

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

```
0.703467422498
```

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

$$\left(2\,x-1\right)^{n}$$

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

$$\left(x+2\right)^{n}$$

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

$$\left(2\,x-1\right)^4$$

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

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

```
625
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$x^3-x$$

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

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

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

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

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

yang tidak dapat dinilai secara simbolis.
Jika kita mendefinisikan kembali fungsi dengan kata kunci "peta" dapat digunakan untuk vektor x. Secara internal, fungsi dipanggil untuk semua nilai x satu kali, dan hasilnya disimpan dalam vektor.

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

Sekarang fungsi dapat dipanggil dengan atau tanpa parameter "basis".

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

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

```
2
```

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

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

$$b^2 - a\,b + b + a^2$$

$$y^2 - x\,y + y + x^2$$

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

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

```
17
```

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

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

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

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

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

$$0$$

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

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

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

Untuk meringkas
- &= mendefinisikan fungsi simbolis,
- := mendefinisikan fungsi numerik,
- &&= mendefinisikan fungsi simbolis murni.

Memecahkan Ekspresi

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

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

```
1.41421356237
```

Ini juga berfungsi untuk ekspresi simbolis. Ambil fungsi berikut.

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

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

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

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

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

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

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

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

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

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

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

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

```
0.966715594851
2
```

Memecahkan ekspresi simbolis dalam bentuk simbolis mengembalikan daftar solusi. Kami menggunakan pemecah simbolik solve() yang disediakan oleh Maxima.

```
>sol &= solve(x^2-x-1,x); $&sol
```

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

Cara termudah untuk mendapatkan nilai numerik adalah dengan mengevaluasi solusi secara numerik seperti ekspresi.

```
>longest sol()
```

```
    -0.6180339887498949       1.618033988749895
```

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

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

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

$$0$$

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

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

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

Fungsi f() dapat melihat variabel global. Namun seringkali kita ingin menggunakan parameter lokal.
lateks: a^x-x^a = 0.1
dengan a=3.

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

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

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

```
2.54116291558
```

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

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

```
2.54116291558
```

Menyelesaikan Pertidaksamaan

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

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

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

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

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

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

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

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

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

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

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

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

$$emptyset$$

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

$$universalset$$

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

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

```
>$&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])
```

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

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

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

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

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

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

```
          [6 < x, x < 8, y < - 11] or [8 < x, y < - 11]
   or [x < 8, 13 < y] or [x = y, 13 < y] or [8 < x, x < y, 13 < y]
   or [y < x, 13 < y]
```

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

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

Bahasa Matriks

Dokumentasi inti EMT berisi 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
```

Produk matriks dilambangkan dengan 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 untuk elemen.

```
>A.A
```

```
              7            10
             15            22
```

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

```
              1             4
              9            16
```

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

```
             37            54
             81           118
```

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

```
             37            54
             81           118
```

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

```
              1             1
              1             1
```

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

```
0.333333      0.666667
    0.75             1
```

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

```
      -2
     2.5
```

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

```
      -2
     2.5
```

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

```
       1             0
       0             1
```

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

```
       1             0
       0             1
```

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

```
       1             4
       9            16
```

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

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

```
       9
      16
```

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

```
>2*A
```

```
       2             4
       6             8
```

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

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

```
       1             4
       3             8
```

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

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

```
            2           6
            6          12
```

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

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

```
            1           2
            1           2
```

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

```
            1           4
            3           8
```

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

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

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

Sekali lagi, ingat bahwa ini bukan produk matriks!

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

```
  55
```

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

```
  55
```

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

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

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

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

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

```
5
```

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

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

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

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

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

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

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

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

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

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

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

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

EMT tidak sepenuhnya efektif untuk perhitungan bilangan bulat. Ini menggunakan titik mengambang presisi ganda secara internal. Namun, seringkali sangat berguna.
Kita dapat memeriksa keutamaan. Mari kita cari tahu, berapa banyak kuadrat ditambah 1 adalah bilangan prima.

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

```
112
```

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

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

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

Ini mengembalikan indeks elemen, yang bukan nol.

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

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

Indeks ini dapat digunakan untuk mengatur elemen ke beberapa nilai.

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

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

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

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

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

Dan dimungkinkan untuk mendapatkan elemen dalam vektor.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Fungsi Matriks Lainnya (Membangun Matriks)

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

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

```
              1             2             3
              1             2             3
```

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

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

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

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

```
>A|1
```

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

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

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

```
              1             2             3
              1             2             3
```

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

```
              1             1
              2             2
              3             3
```

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

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

```
              1               1
              2               4
              3               9
```

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut.

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

```
2
4
[2,  4]
4
```

Untuk vektor, ada panjang().

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

```
9
```

Ada banyak fungsi lain, yang menghasilkan matriks.

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

```
              1               1
              1               1
```

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

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

```
[6,  6,  6,  6,  6]
```

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

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

```
        0.66566        0.831835
          0.977        0.544258
```

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

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

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

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

```
>function rep(v,n) := 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 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() mengembalikan urutan baris atau kolom matriks. Yaitu, fungsi flipx() membalik secara horizontal.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
1    0    0    0    0
0    1    0    0    0
0    0    1    0    0
0    0    0    1    0
0    0    0    0    1
```

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

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

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

Perhatikan bahwa kami tidak mengubah matriks A. Kami mendapatkan matriks baru sebagai hasil dari setdiag().
Berikut adalah fungsi, yang mengembalikan matriks tri-diagonal.

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

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

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

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

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

Sekarang kita dapat mengekstrak diagonal.

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

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

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

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

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

Vektorisasi

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

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

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

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

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

Dengan ini dan operator titik dua a:delta:b, vektor nilai fungsi dapat dihasilkan dengan mudah.
Pada contoh berikut, kita membangkitkan vektor nilai t[i] dengan spasi 0,1 dari -1 hingga 1. Kemudian kita membangkitkan vektor nilai fungsi
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 menjadi matriks, jika operator diterapkan. Berikut ini, v' adalah vektor yang ditransposisikan (vektor kolom).

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

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

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

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

```
55
```

Secara default, vektor baris dicetak dalam format yang ringkas.

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

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

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

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

```
5
25
```

Untuk mentranspos matriks kita menggunakan apostrof.

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

```
              1
              2
              3
              4
```

Jadi kita dapat menghitung matriks A kali vektor b.

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

```
             30
             70
```

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

```
>v'.v
```

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

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

```
>v.v'
```

```
30
```

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

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

```
30
```

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

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

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

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

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

```
              1             2             3
              2             4             6
              3             6             9
```

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan *!

```
>v.v'
```

```
14
```

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

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

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

```
[1,  4,  9]
14
[1,  5,  14]
```

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

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

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

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

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

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

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

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

```
6
```

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

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

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

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

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

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

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

```
4
```

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

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

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

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

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

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

Sekarang kita menambahkan kolom ke matriks.

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

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

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

Dalam contoh berikut, kita menghitung t[j]^i untuk i dari 1 hingga n. Kami mendapatkan matriks, di mana setiap baris adalah tabel t^i untuk satu i. Yaitu, matriks memiliki elemen lateks: a_{i,j} = t_j^i, \quad 1 \le j \le 101, \quad 1 \le i \le n

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

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

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

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

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

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

Sub-Matriks dan Matriks-Elemen

Untuk mengakses elemen matriks, gunakan notasi braket.

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

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

Kita dapat mengakses satu baris matriks yang lengkap.

```
>A[2]
```

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

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

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

```
2
```

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

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

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

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

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

```
            1             2             3
            4             5             6
```

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

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

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

Trik indeks bekerja dengan kolom juga.
Contoh ini memilih semua baris A dan kolom kedua dan ketiga.

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

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

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

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

```
               3
               6
               9
```

Atau, biarkan indeks pertama kosong.

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

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

Kita juga bisa mendapatkan baris terakhir dari A.

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

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

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

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

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

Kami juga dapat menetapkan nilai ke baris A.

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

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

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

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

```
            5             6            -1
            7             8             6
            7             8             9
```

Selain itu, beberapa jalan pintas diperbolehkan.

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

```
            0             0            -1
            0             0             6
            7             8             9
```

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

```
>A[4]
```

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

Menyortir dan Mengacak

Fungsi sort() mengurutkan vektor baris.

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

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

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

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

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

Indeks berisi urutan yang tepat dari v.

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

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

Ini bekerja untuk vektor string juga.

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

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

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

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

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

Fungsi unik mengembalikan daftar elemen unik vektor yang diurutkan.

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

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

Ini bekerja untuk vektor string juga.

```
>unique(s)
```

```
a
aa
d
e
```

Aljabar linier

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

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

```
             -4
            4.5
```

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

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

```
    8.790745908981989e-13
```

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

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

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

Determinan matriks ini adalah 0.

```
>det(A)
```

```
  0
```

Matriks Simbolik

Maxima memiliki matriks simbolis. Tentu saja, Maxima dapat digunakan untuk masalah aljabar linier sederhana seperti itu. Kita dapat mendefinisikan matriks untuk Euler dan Maxima dengan &:=, dan kemudian menggunakannya dalam ekspresi simbolis. Bentuk [...] biasa untuk mendefinisikan matriks dapat digunakan di Euler untuk mendefinisikan matriks simbolik.

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

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

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

$$a\,\left(a^2-1\right)-2\,a+2$$

$$\left(a-1\right)^2\,\left(a+2\right)$$

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

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

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

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

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

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

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

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

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

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

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

Untuk mengekstrak vektor eigen tertentu perlu pengindeksan yang cermat.

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

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

```
[1, - 1]
```

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

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

```
              1              4
              5              2
```

Dalam ekspresi simbolik, gunakan dengan.

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

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

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

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

$$[1,a]$$

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

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

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

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

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

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

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

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

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

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

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

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

Euler juga memiliki fungsi xinv() yang kuat, yang membuat upaya lebih besar dan mendapatkan hasil yang lebih tepat.
Perhatikan, bahwa dengan &:= matriks B telah didefinisikan sebagai simbolik dalam ekspresi simbolik dan sebagai numerik dalam ekspresi numerik. Jadi kita bisa menggunakannya di sini.

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

```
                          1                        0
                          0                        1
```

Misalnya. nilai eigen dari A dapat dihitung secara numerik.

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

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

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

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

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

Nilai Numerik dalam Ekspresi simbolis

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

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

```
            1         3.14159
            4               5
```

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

```
>$&A
```

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

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

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

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

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

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

$$1.4142135623730950488016887242097_B \times 10^{0}$$

$$1.414213562373095$$

Ketepatan angka floating point besar dapat diubah.

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

```
        3.14159265358979323846264338327950288419716939937510582097494\
  4592307816406286208998628034825342117068b0
```

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

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

$$-5.424777960769379$$

Demo - Suku Bunga

Di bawah ini, kami menggunakan Euler Math Toolbox (EMT) untuk perhitungan suku bunga. Kami melakukannya secara numerik dan simbolis untuk menunjukkan kepada Anda bagaimana Euler dapat digunakan untuk memecahkan masalah kehidupan nyata.
Asumsikan Anda memiliki modal awal 5000 (katakanlah dalam dolar).

```
>K=5000
```

```
5000
```

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

```
>K*1.03
```

```
5150
```

Euler akan memahami sintaks berikut juga.

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

```
5150
```

Tetapi lebih mudah menggunakan faktornya

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

```
1.03
5150
```

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

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

```
6719.58189672
```

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

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

```
6719.58
```

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

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

```
 Starting from 5000$ you get 6719.58$.
```

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

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

```
 Real 1 x 11 matrix

     5000.00      5150.00      5304.50      5463.64      ...
```

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

```
>short 0:10
```

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

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

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

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

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

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

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

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

Mari kita bandingkan dua hasil, dengan dan tanpa pembulatan.

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

```
                  1271.61
                 1271.6071
```

Sekarang tidak ada rumus sederhana untuk tahun ke-n, dan kita harus mengulang selama bertahun-tahun. Euler memberikan banyak solusi untuk ini.
Cara termudah adalah iterasi fungsi, yang mengulangi fungsi tertentu beberapa kali.

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

```
Real 1 x 11 matrix

     5000.00      5150.00      5304.50      5463.64      ...
```

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

```
>VKr'
```

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

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

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

```
     5150.00
     5000.00      5150.00      5304.50
```

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

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

```
     6719.60
     6719.58
```

Perbedaannya sangat kecil.

Memecahkan Persamaan

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

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

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

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

```
  Real 1 x 11 matrix

      5000.00      5350.00      5710.50      6081.82      ...
```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

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

```
  Real 1 x 11 matrix

      5000.00      4950.00      4898.50      4845.45      ...
```

Kami melihat bahwa uang berkurang. Jelas, jika kita hanya mendapatkan 150 bunga di tahun pertama, tetapi menghapus 200, kita kehilangan uang setiap tahun.
Bagaimana kita bisa menentukan berapa tahun uang itu akan bertahan? Kita harus menulis loop untuk ini. Cara termudah adalah dengan iterasi cukup lama.

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

```
  Real 1 x 51 matrix

      5000.00      4950.00      4898.50      4845.45      ...
```

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

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

```
         48.00
```

Alasan untuk ini adalah bahwa bukan nol(VKR<0) mengembalikan vektor indeks i, di mana VKR[i]<0, dan min menghitung indeks minimal.
Karena vektor selalu dimulai dengan indeks 1, jawabannya adalah 47 tahun.
Fungsi iterate() memiliki satu trik lagi. Itu bisa mengambil kondisi akhir sebagai argumen. Kemudian akan mengembalikan nilai dan jumlah iterasi.

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

```
        -19.83
         47.00
```

Mari kita coba menjawab pertanyaan yang lebih ambigu. Asumsikan kita tahu bahwa nilainya adalah 0 setelah 50 tahun. Apa yang akan menjadi tingkat bunga?
Ini adalah pertanyaan yang hanya bisa dijawab dengan angka. Di bawah ini, kita akan mendapatkan formula yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada formula yang mudah untuk tingkat bunga. Tapi untuk saat ini, kami bertujuan untuk solusi numerik.
Langkah pertama adalah mendefinisikan fungsi yang melakukan iterasi sebanyak n kali. Kami menambahkan semua parameter ke fungsi ini.

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

Iterasinya sama seperti di atas

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

Tapi kami tidak lagi menggunakan nilai global R dalam ekspresi kami. Fungsi seperti iterate() memiliki trik khusus di Euler. Anda dapat meneruskan nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R.
Selain itu, kami hanya tertarik pada nilai terakhir. Jadi kita ambil indeks [-1].
Mari kita coba tes.

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

```
-19.83
```

Sekarang kita bisa menyelesaikan masalah kita.

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

```
3.15
```

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

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

```
-336.08
```

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

Solusi Simbolik untuk Masalah Suku Bunga

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

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

$$R + q\,K$$

Kita sekarang dapat mengulangi ini.

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

$$q\,\left(q\,\left(q\,\left(R+q\,K\right)+R\right)+R\right)+R$$

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

Kami melihat sebuah pola. Setelah n periode yang kita miliki
lateks: K_n = q^n K + R (1+q+\ldots+q^{n-1}) = q^n K + \frac{q^n-1}{q-1} R
Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

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

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

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

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

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

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

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

```
      -19.82504734650985
      -19.82504734652684
```

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

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

```
       20.51
```

Jawaban ini mengatakan bahwa itu akan menjadi negatif setelah 21 tahun.
Kita juga dapat menggunakan sisi simbolis Euler untuk menghitung formula pembayaran.
Asumsikan kita mendapatkan pinjaman sebesar K, dan membayar n pembayaran sebesar R (dimulai setelah tahun pertama) meninggalkan sisa hutang sebesar Kn (pada saat pembayaran terakhir). Rumus untuk ini jelas

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

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

Biasanya rumus ini diberikan dalam bentuk

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

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

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

Kita dapat memecahkan tingkat R secara simbolis.

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

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

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

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

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

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

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

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

Penyelesaian Soal Aljabar

1. Sederhanakan bentuk eksponen berikut

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

Penyelesaian:

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

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

2. Sederhanakan bentuk eksponen berikut

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

Penyelesaian :

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

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

3. Hitunglah

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

Penyelesaian :

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

```
5
```

4. Hitunglah

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

Penyelesaian :

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

```
- 5
```

5. Hitunglah

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

Penyelesaian :

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

$$-2\,x^2+6\,x-2$$

6. Menyelesaikan pemfaktoran

$$t^2+8t+15$$

Penyelesaian :

```
>$&factor(t^2+8*t+15)
```

$$(t+3)\ (t+5)$$

7. Menyelesaikan operasi yang di tunjukkan

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

Penyelesaian :

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

$$-2\,z+2\,y+3\,x-3$$

8. Hitunglah :

$$x^3-4x^2+5x-20$$

Penyelesaian :

```
>$&factor(x^3-4*x^2+5*x-20)
```

$$(x-4)\ \left(x^2+5\right)$$

9. Faktorkan

$$250z^4-2z$$

Penyelesaian :

```
>$&factor(250*z^4-2*z)
```

$$2\,z\ (5\,z-1)\ \left(25\,z^2+5\,z+1\right)$$

10. Faktorkan :

$$18a^2b-15ab^2$$

Penyelesaian :

```
>$&factor(18*a^2*b-15*a*b^2)
```

$$-3\,a\,b\ (5\,b-6\,a)$$

11. Faktorkan

$$25ab^4 - 25az^4$$

Penyelesaian :

```
>$&factor(25*a*b^4-25*a*z^4)
```

$$-25\,a\,\left(z-b\right)\,\left(z+b\right)\,\left(z^2+b^2\right)$$

12. Faktorkan :

$$16a^7b + 54ab^7$$

Penyelesaian :

```
>$&factor(16*a^7*b+54*a*b^7)
```

$$2\,a\,b\,\left(3\,b^2+2\,a^2\right)\,\left(9\,b^4-6\,a^2\,b^2+4\,a^4\right)$$

13. Hitunglah :

$$m^6 + 8m^3 - 20$$

Penyelesaian :

```
>$&factor(m^6+8*m^3-20)
```

$$\left(m^3-2\right)\,\left(m^3+10\right)$$

14. Asumsikan bahwa semua eksponen adalah bilangan asli. Kalikan

$$(a^n + b^n)(a^n - b^n)$$

Penyelesaian :

```
>$&expand ((a^n+b^n)*(a^n-b^n))
```

$$a^{2\,n}-b^{2\,n}$$

15. Asumsikan bahwa semua eksponen adalah bilangan asli. Kalikan

$$(x^{3m} - t^{5n})^2$$

Penyelesaian :

```
>$&expand((x^(3*m)-t^(5*n))^2)
```

$$x^{6\,m}-2\,t^{5\,n}\,x^{3\,m}+t^{10\,n}$$

16. Asumsikan bahwa semua eksponen adalah bilangan asli. Kalikan

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

Penyelesaian :

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

$$c^2+2\,b\,c+2\,a\,c+b^2+2\,a\,b+a^2$$

17. Menyelesaikan operasi yang ditunjukkan

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

Penyelesaian :

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

$$12\,x\,y-y+7\,x^2-2\,x-9$$

18. Asumsikan bahwa semua variabel dalam eksponen merepresentasikan bilangan asli. Faktorkan

$$25y^{2m}-(x^{2n}-2x^n+1)$$

Penyelesaiann :

```
>$&factor(25*y^(2*m)-(x^(2*n)-2*x^n+1))
```

$$\left(5\,y^{m}-x^{n}+1\right)\,\left(5\,y^{m}+x^{n}-1\right)$$

19. Asumsikan bahwa semua variabel dalam eksponen merepresentasikan bilangan asli. Faktorkan

$$bdy^2+ady+bcy+ac$$

Penyelesaian :

```
>$&factor(b*d*y^2+a*d*y+b*c*y+a*c)
```

$$\left(b\,y+a\right)\,\left(d\,y+c\right)$$

20. Asumsikan bahwa semua variabel dalam eksponen merepresentasikan bilangan asli. Faktorkan

$$4x^{2n} - 4x^{n} - 3$$

Penyelesaian :

```
>$&factor(b*d*y^2+a*d*y+b*c*y+a*c)
```

$$(b\,y+a)\,(d\,y+c)$$

BAB 2

MENGGUNAKAN EMT UNTUK MENGGAMBAR GRAFIK DIMENSI (2D)

[a4paper,10pt]article eumat

Identitas

Nama: Anisah Daffa Citra Nareswari
NIM: 22305141044
Kelas: Matematika E 2022

Menggambar Grafik 2D dengan EMT

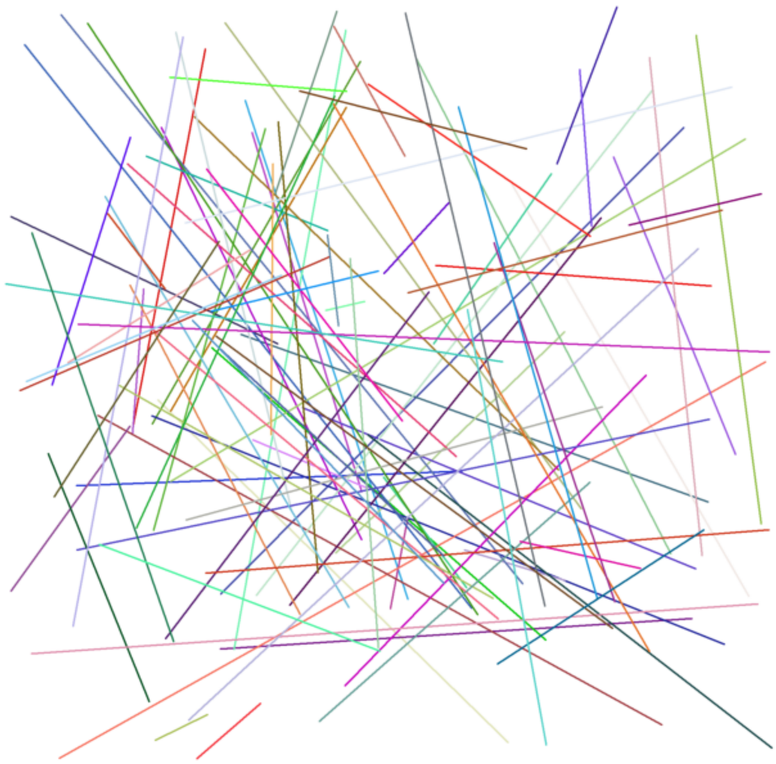
Notebook ini menjelaskan tentang cara menggambar berbagaikurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi plot2d() untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

Basic Plots

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

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

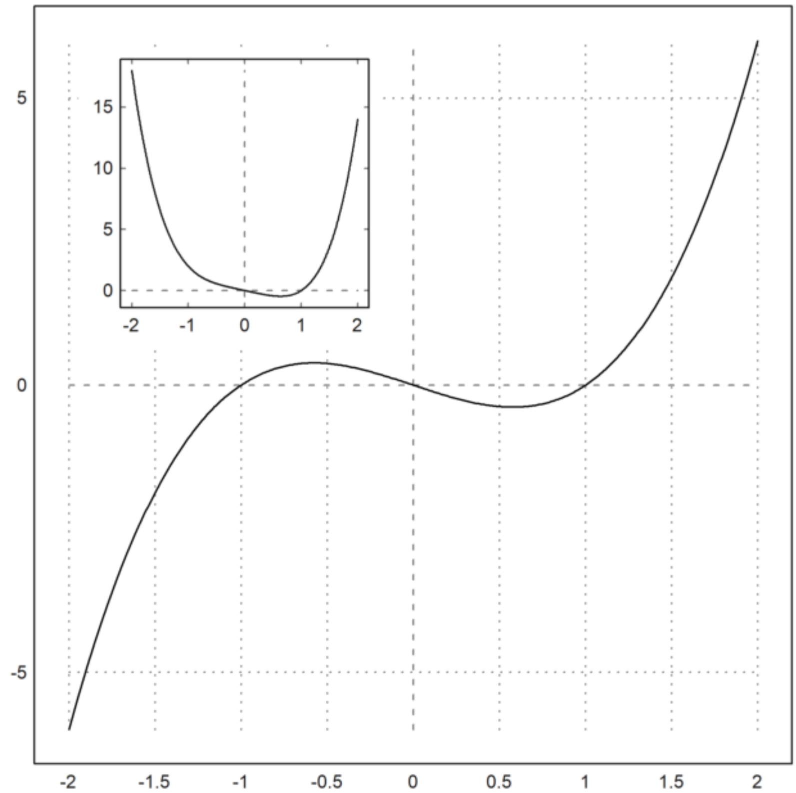
```
>clg; // clear screen
>window(0,0,1024,1024); // use all of the window
>setplot(0,1,0,1); // set plot coordinates
>hold on; // start overwrite mode
>n=100; X=random(n,2); Y=random(n,2);  // get random points
>colors=rgb(random(n),random(n),random(n)); // get random colors
>loop 1 to n; color(colors[#]);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



```
>reset;
```

Grafik perlu ditahan, karena perintah plot() akan menghapus plot
jendela.
Untuk membersihkan semuanya, we use 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 lain, kami menggambar plot sebagai sisipan di plot lain. Ini dilakukan untuk mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak menyediakan ruang untuk sumbu label di luar jendela plot. Kita harus menambahkan beberapa margin untuk ini sesuai kebutuhan. Perhatikan bahwa kami menyimpan dan memulihkan jendela penuh, dan menahan plot saat ini saat kami memplot inset.

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



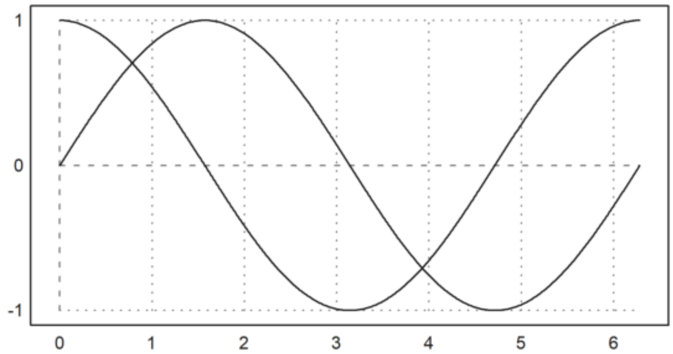

```
>hold off;
>window(ow);
```

Sebuah plot dengan beberapa angka dicapai dengan cara yang sama. Ada angka utilitas() fungsi untuk ini.

Plot Aspect

Plot default menggunakan jendela plot persegi. Anda dapat mengubah ini dengan fungsi aspek(). Jangan lupa untuk mengatur ulang aspek nanti. Anda juga dapat mengubah default ini di menu dengan "Set Aspect" ke rasio aspek tertentu atau ke ukuran jendela grafis saat ini.
Tetapi Anda juga dapat mengubahnya untuk satu plot. Untuk ini, ukuran area plot saat ini berubah, dan jendela diatur sehingga label memiliki cukup ruang.

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




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

Fungsi reset() mengembalikan default plot termasuk rasio aspek.

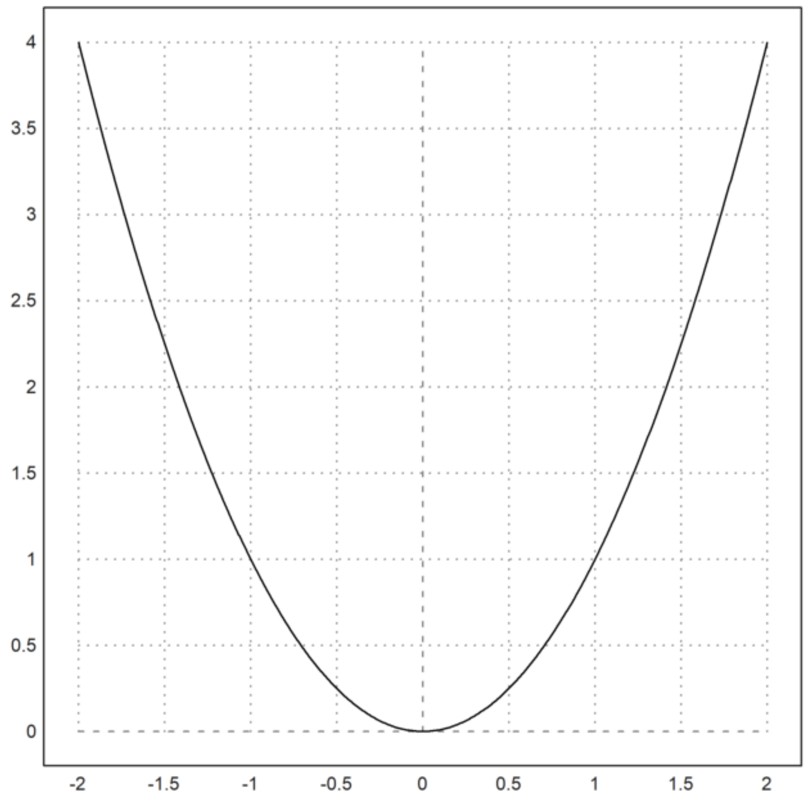
Plot 2D di Euler

EMT Math Toolbox memiliki plot dalam 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d. Fungsi ini dapat memplot fungsi dan data.
Dimungkinkan untuk membuat plot di Maxima menggunakan Gnuplot atau dengan Python menggunakan Math Plot Lib.
Euler dapat memplot plot 2D dari
- ekspresi
- fungsi, variabel, atau kurva parameter,
- vektor nilai x-y,
- awan titik di pesawat,
- kurva implisit dengan level atau wilayah level.
- Fungsi kompleks
Gaya plot mencakup berbagai gaya untuk garis dan titik, plot batang dan plot berbayang.

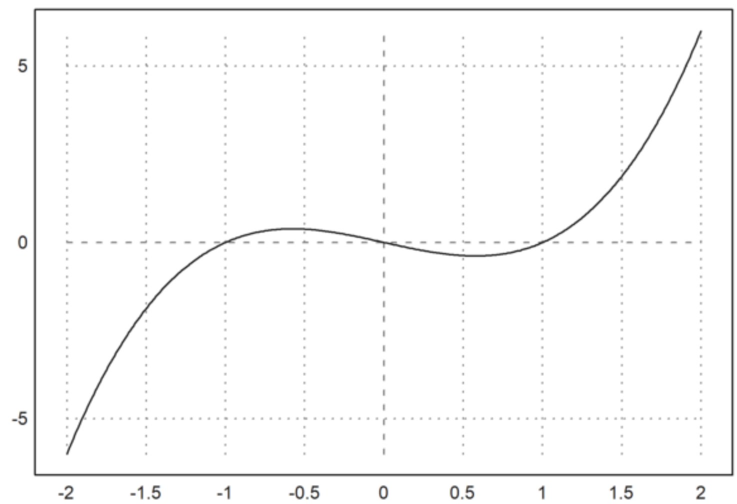
Plot Ekspresi atau Variabel

Ekspresi tunggal dalam "x" (mis. "4*x^2") atau nama fungsi (mis. "f") menghasilkan grafik fungsi.
Berikut adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsi.
Catatan: Jika Anda mengakhiri baris perintah dengan titik dua ":", plot akan dimasukkan ke dalam jendela teks. Jika tidak, tekan TAB untuk melihat plot jika jendela plot tertutup.

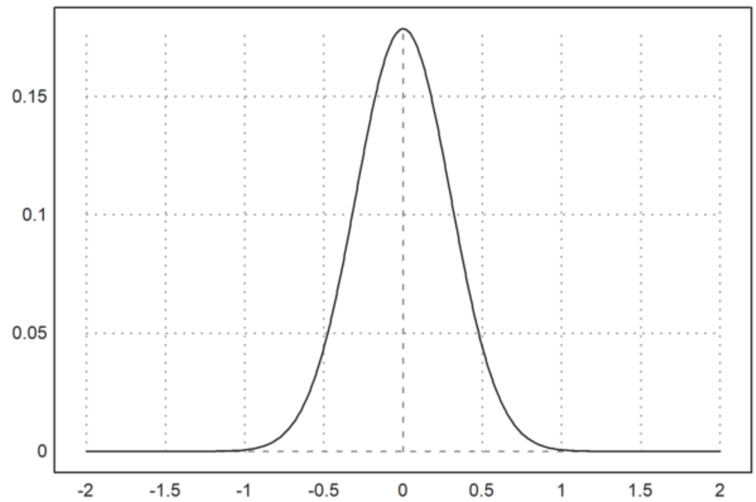
```
>plot2d("x^2"):
```




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

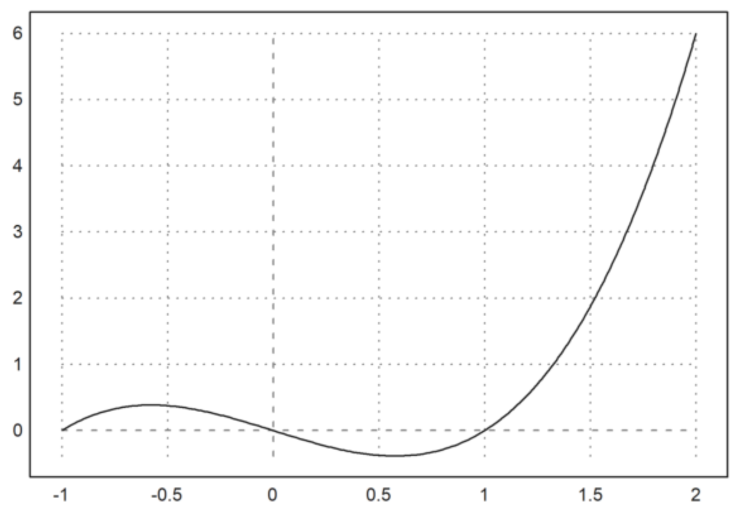



```
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25
```

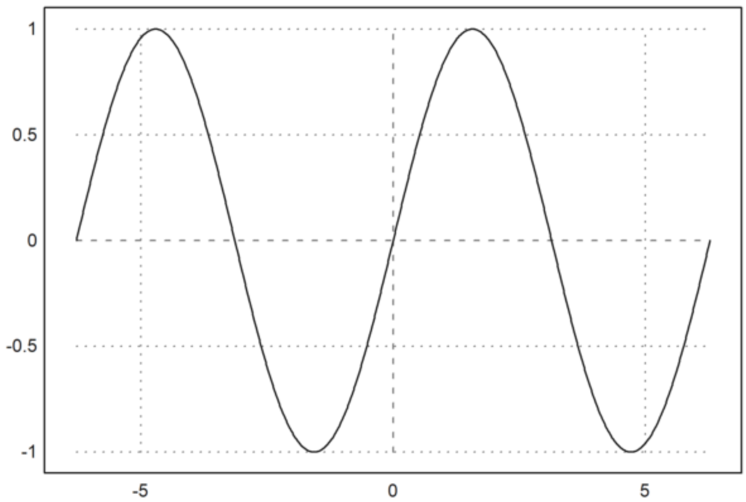


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.
Rentang plot diatur dengan persamaan parameter berikut:
- a,b: rentang x (default -2,2)
- c,d: rentang y (default: skala dengan nilai)
- r: alternatif radius disekitar pusat plot
- cx,xy: koordinat pusat plot (default 0,0)

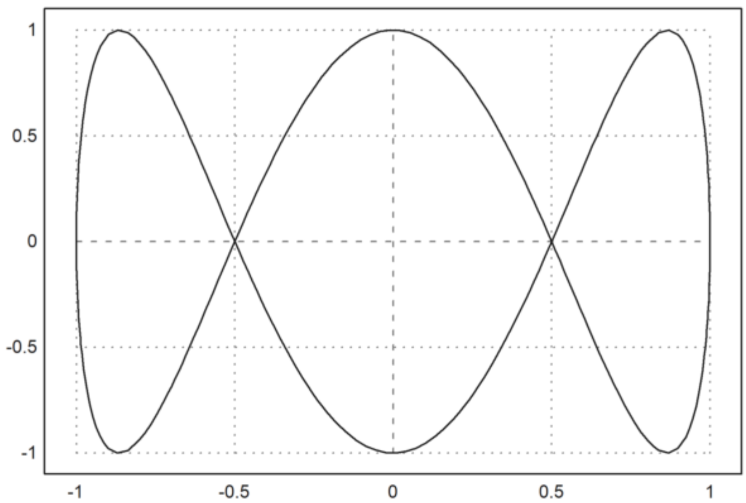
```
>plot2d("x^3-x",-1,2):
```




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

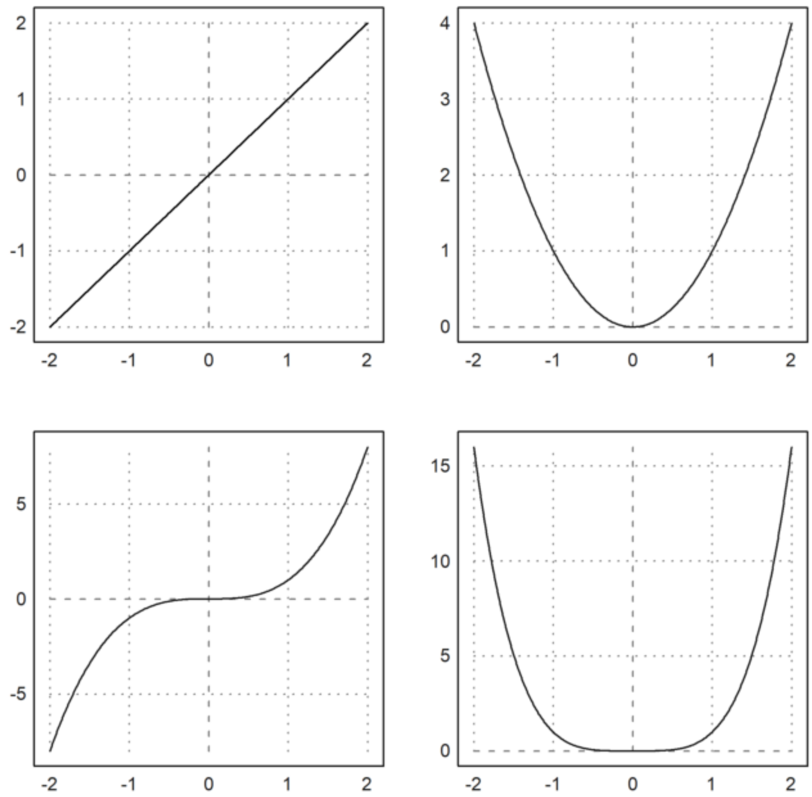



```
>plot2d("cos(x)","sin(3*x)",xmin=0,xmax=2pi):
```



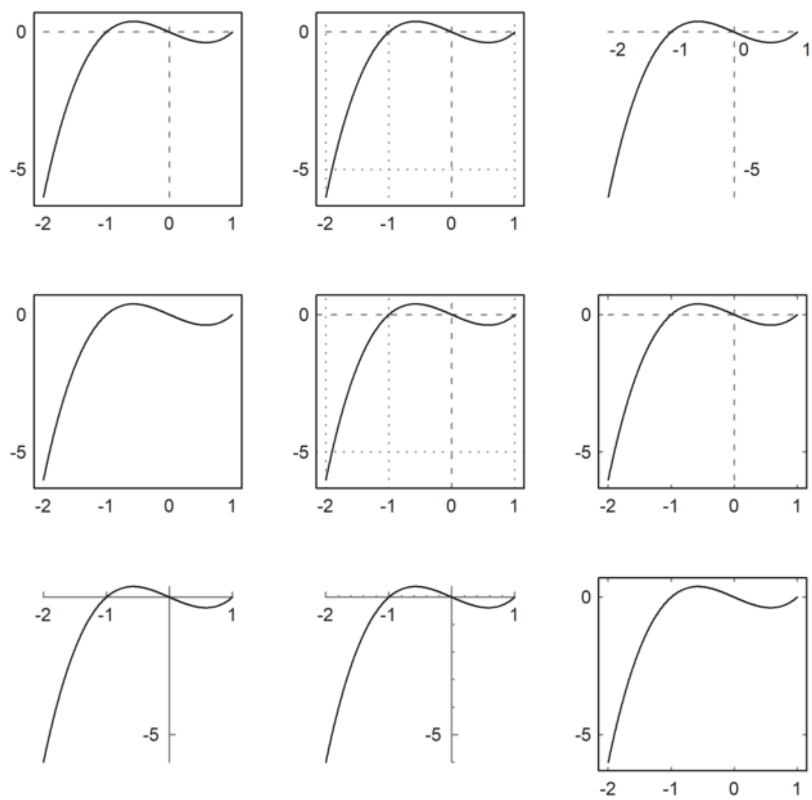
Alternatif untuk titik dua adalah perintah insimg(baris), yang menyisipkan plot yang menempati sejumlah baris teks tertentu.
Dalam opsi, plot dapat diatur untuk muncul
- di jendela terpisah yang dapat diubah ukurannya,
- di jendela buku catatan.
Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.
Bagaimanapun, tekan tombol tabulator untuk melihat plot, jika disembunyikan.
Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Dalam contoh, kami memplot x^1 hingga x^4 menjadi 4 bagian jendela. figure(0) mengatur ulang jendela default.

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



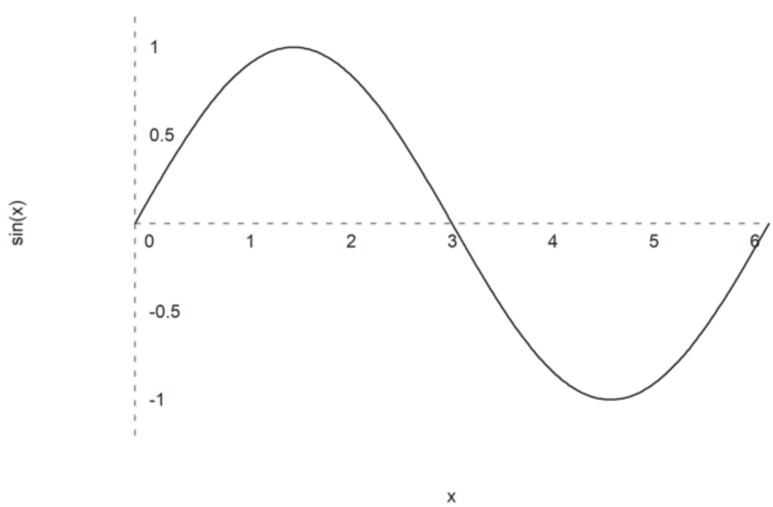
Di plot2d(), ada gaya alternatif yang tersedia dengan grid=x. Untuk gambaran umum, kami menunjukkan berbagai gaya kisi dalam satu gambar (lihat di bawah untuk perintah figure()). Gaya kisi=0 tidak disertakan. Ini menunjukkan tidak ada grid dan tidak ada bingkai.

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

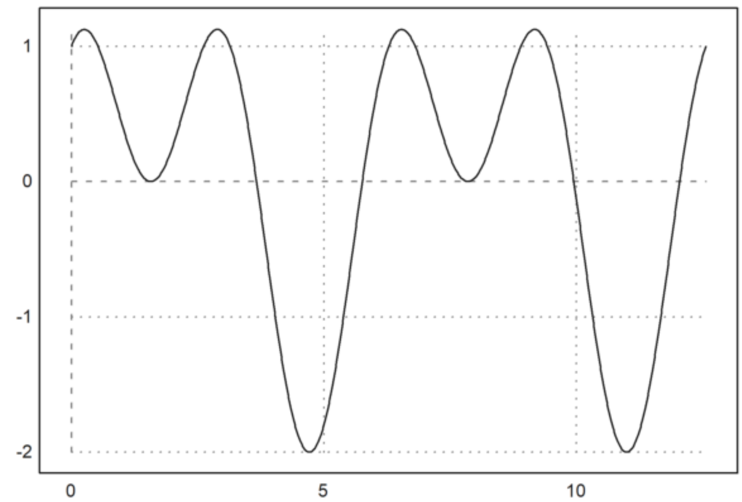


Jika argumen ke plot2d() adalah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot.
Atau, a, b, c, d dapat ditentukan sebagai parameter yang ditetapkan sebagai a=... dll.
Dalam contoh berikut, kita mengubah gaya kisi, menambahkan label, dan menggunakan label vertikal untuk sumbu y.

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



```
>plot2d("sin(x)+cos(2*x)",0,4pi):
```

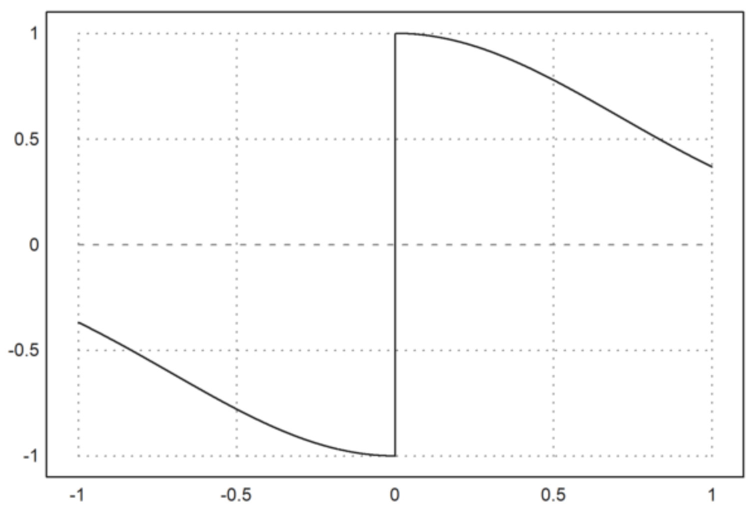


Gambar yang dihasilkan dengan memasukkan plot ke dalam jendela teks disimpan di direktori yang sama dengan buku catatan, secara default di subdirektori bernama "gambar". Mereka juga digunakan oleh ekspor HTML.

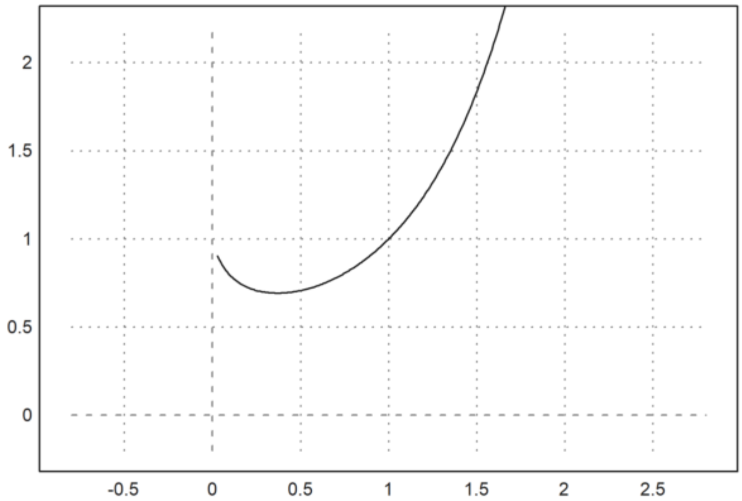
Anda cukup menandai gambar apa saja dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengekspor grafik saat ini dengan fungsi di menu File.

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

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

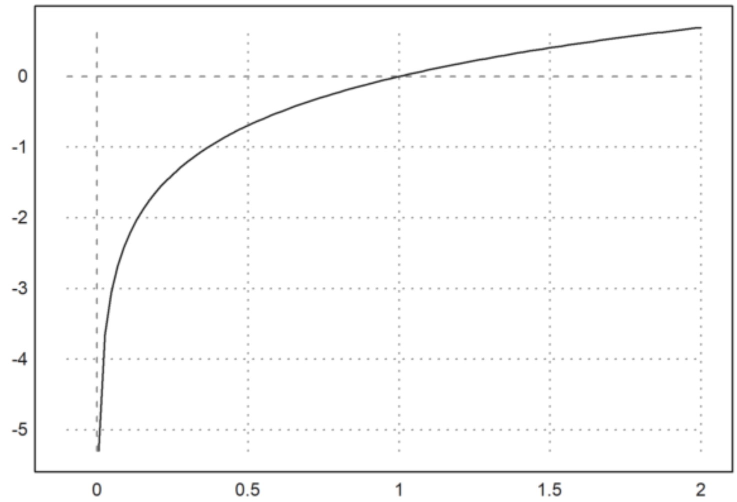



```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



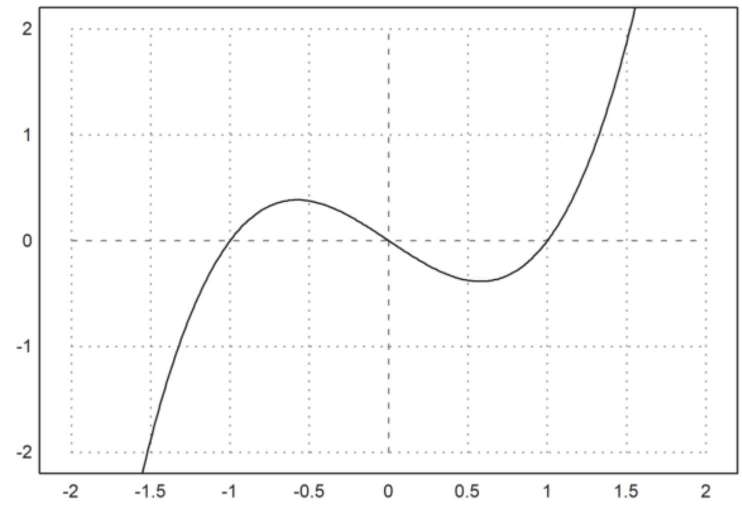
Perhatikan bahwa x^x tidak didefinisikan untuk x<=0. Fungsi plot2d menangkap kesalahan ini, dan mulai merencanakan segera setelah fungsi didefinisikan. Ini berfungsi untuk semua fungsi yang mengembalikan NAN keluar dari jangkauan definisinya.

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

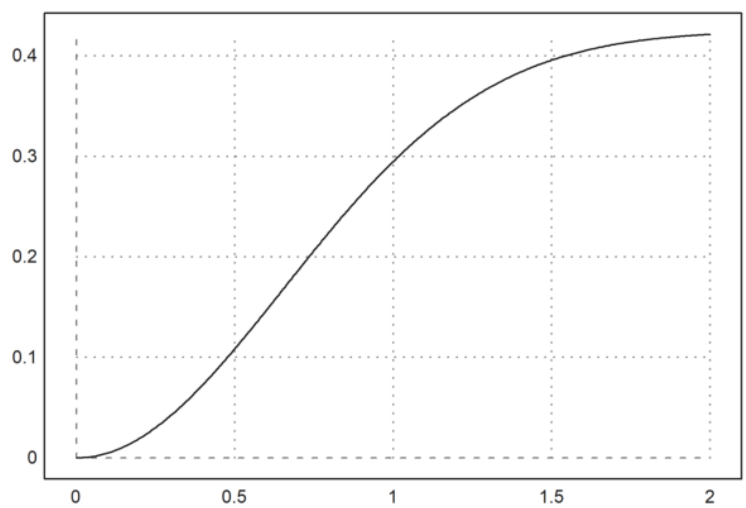


Parameter square=true (atau >square) memilih y-range secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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

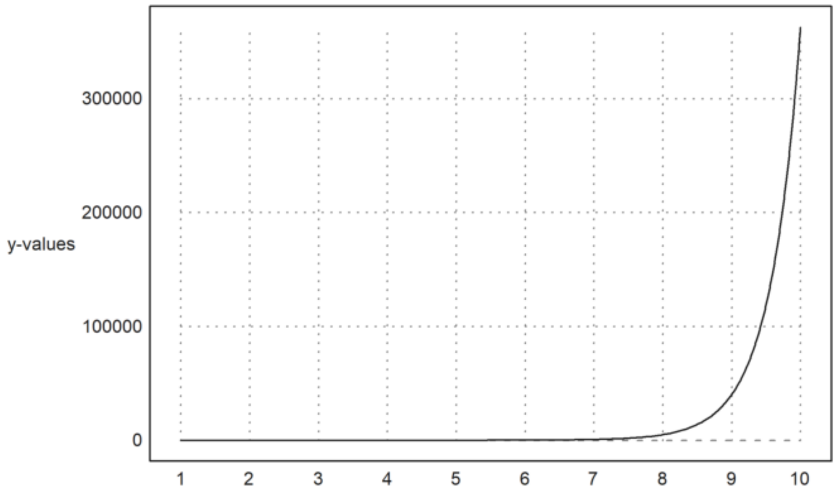



```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```



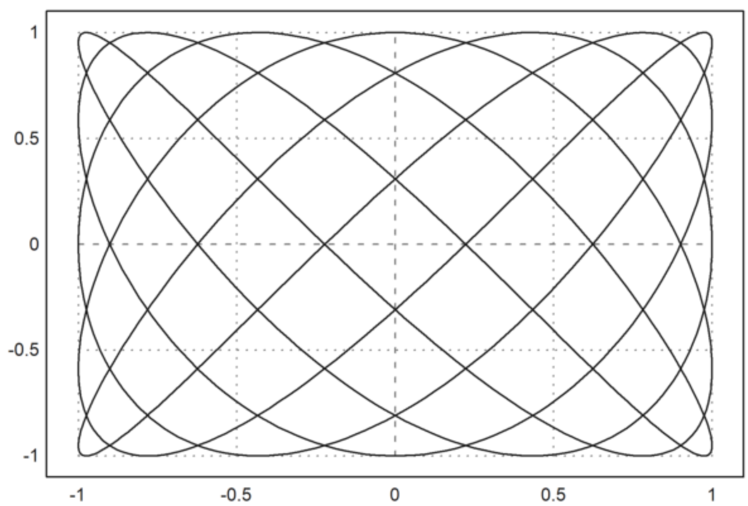
Jika Anda membutuhkan lebih banyak ruang untuk label-y, panggil shrinkwindow() dengan parameter yang lebih kecil, atau tetapkan nilai positif untuk "lebih kecil" di plot2d().

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

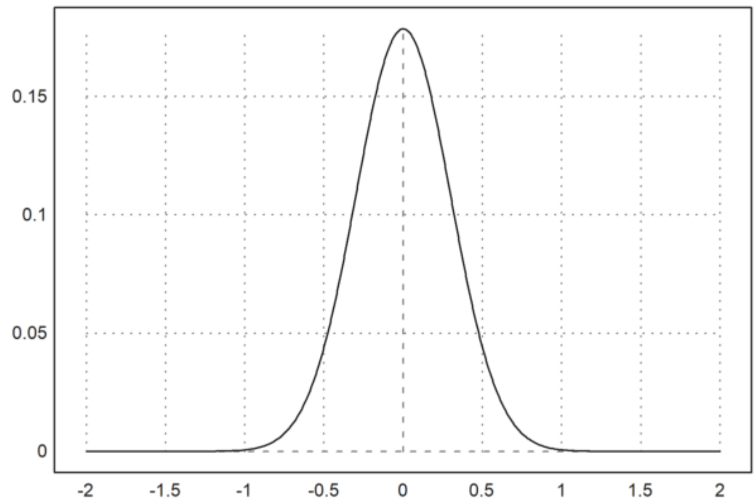


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

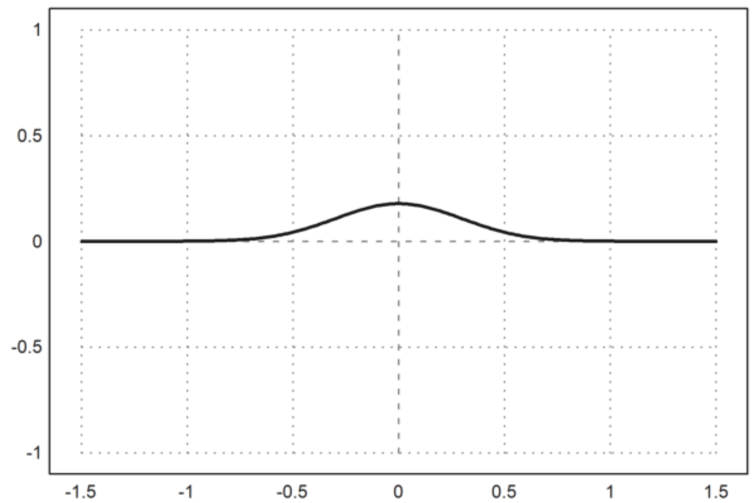
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



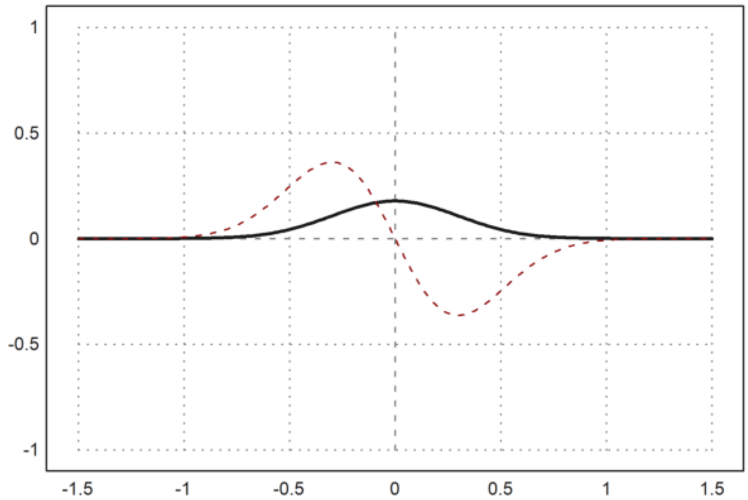

```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```



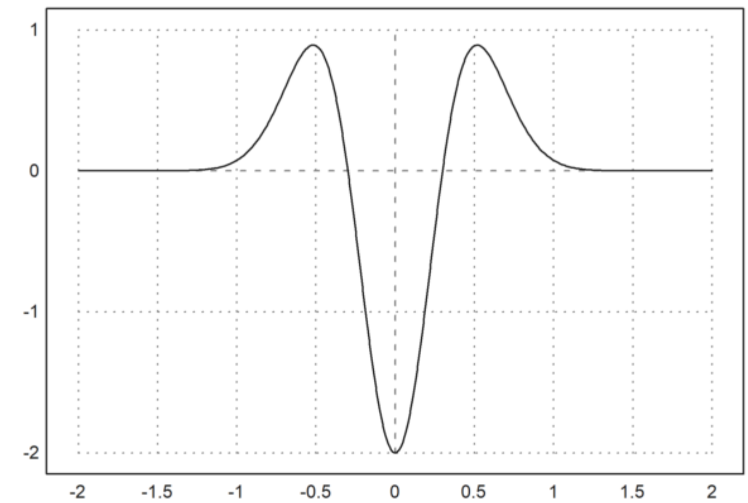

```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



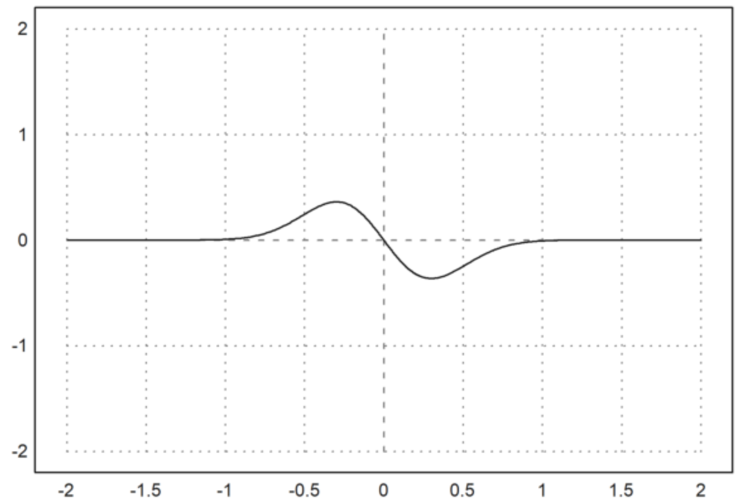

```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



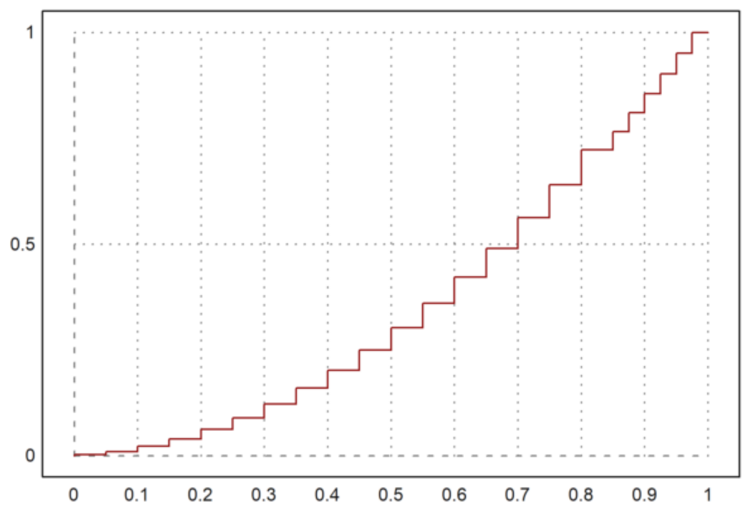

```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```



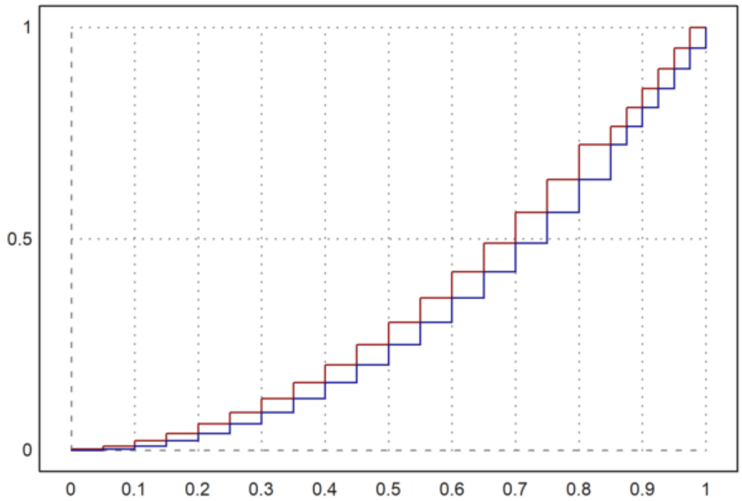

```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```




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



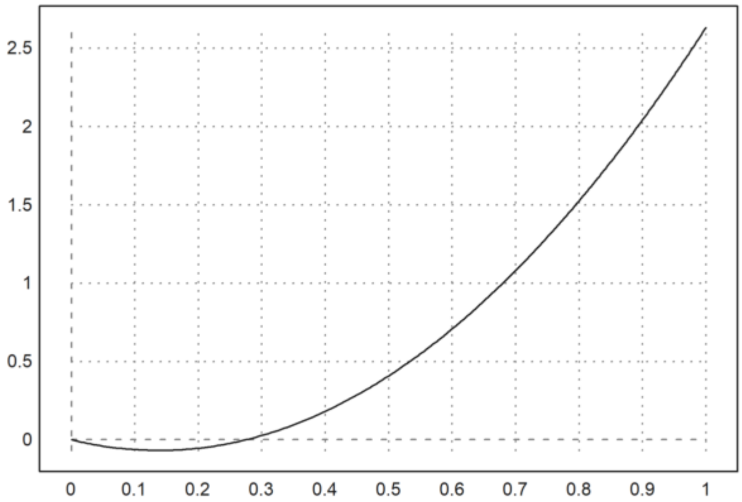

```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```



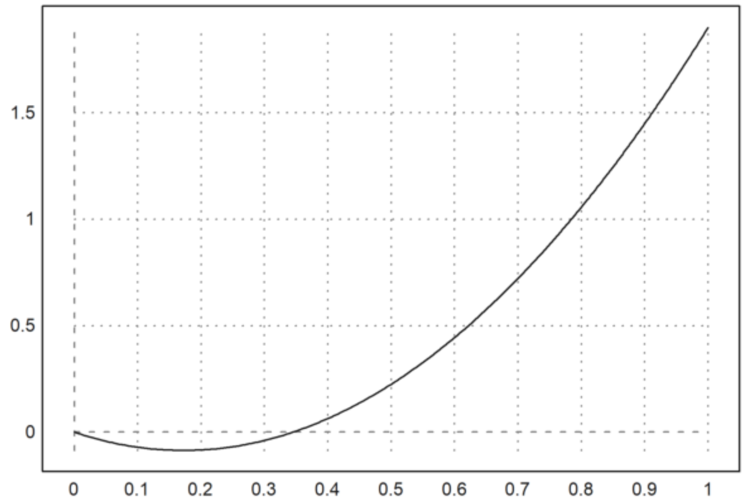
Fungsi dalam satu Parameter

Fungsi plot yang paling penting untuk plot planar adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat di awal program.
Berikut adalah beberapa contoh menggunakan fungsi. Seperti biasa di EMT, fungsi yang berfungsi untuk fungsi atau ekspresi lain, Anda dapat meneruskan parameter tambahan (selain x) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan koleksi panggilan.

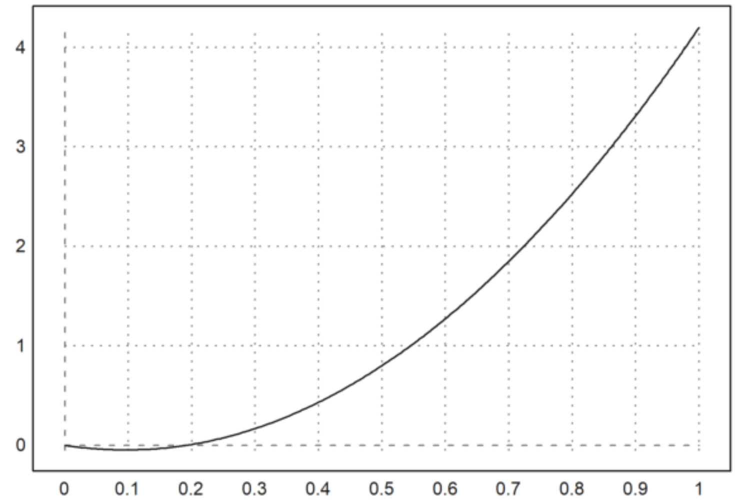
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```



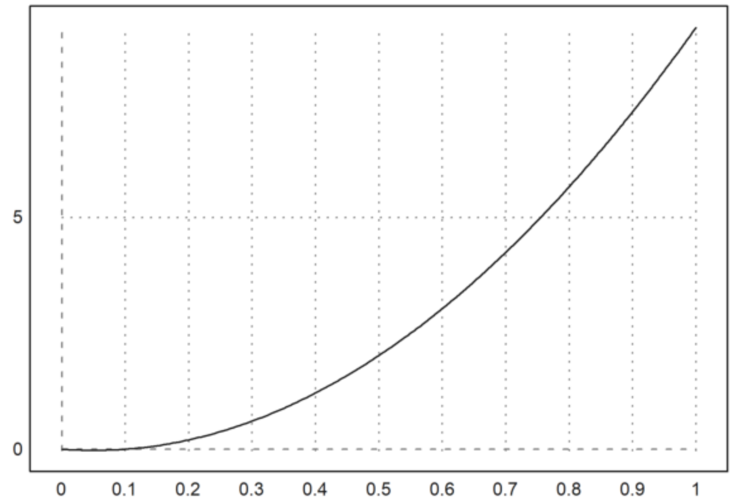

```
>plot2d("f",0,1;0.4): // plot with a=0.4
```



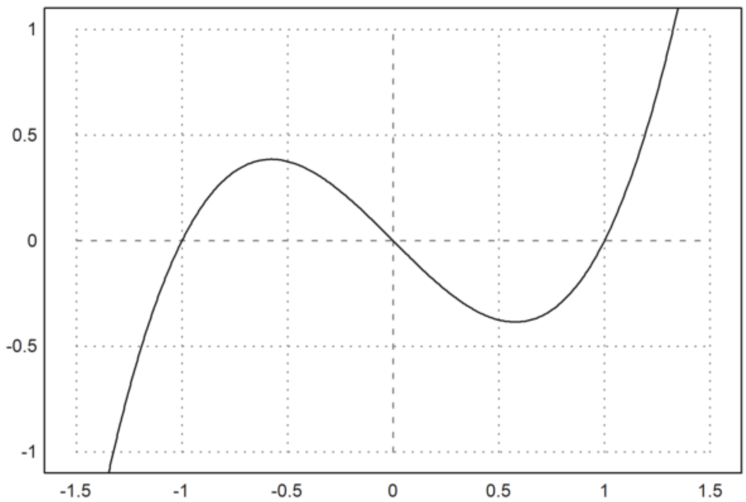

```
>plot2d({{"f",0.2}},0,1): // plot with a=0.2
```




```
>plot2d({{"f(x,b)",b=0.1}},0,1): // plot with 0.1
```




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



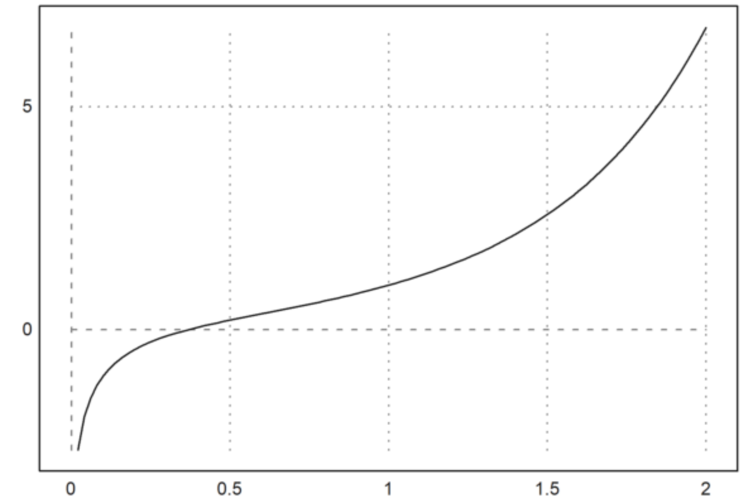
Berikut adalah ringkasan dari fungsi yang diterima
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolis dengan nama sebagai "f"
- fungsi simbolis hanya dengan nama f

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

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

```
                               x
                              x  (log(x) + 1)
```

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

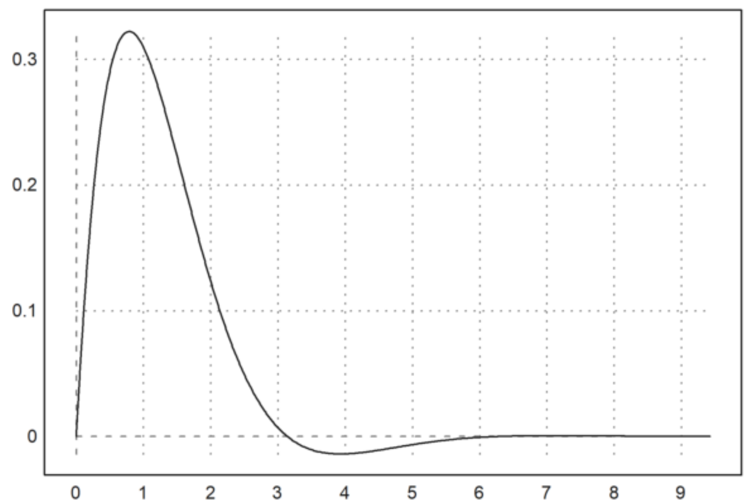


Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel sudah cukup untuk memplotnya.

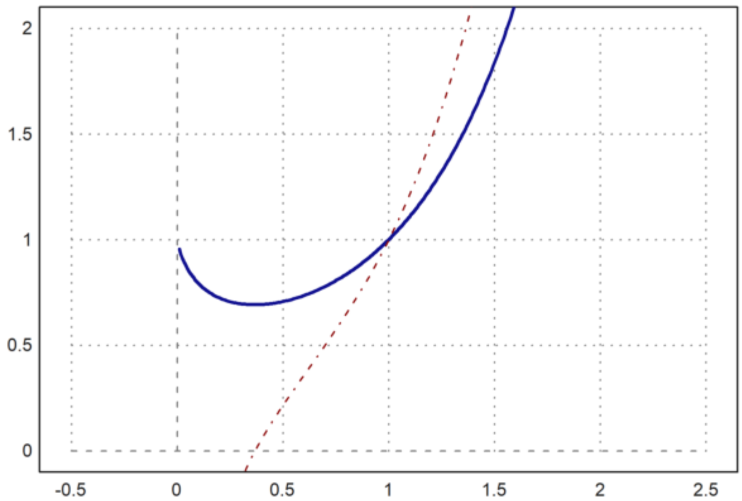
```
>expr &= sin(x)*exp(-x)
```

```
                                 - x
                                E     sin(x)
```

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



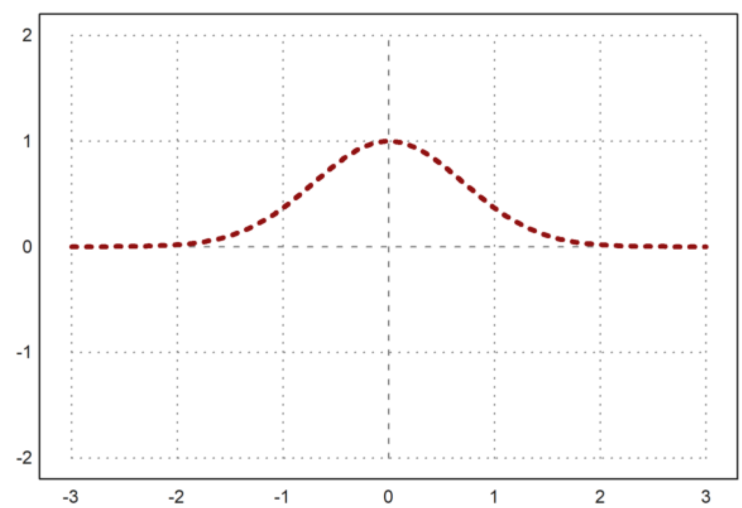
```
>function f(x) &= x^x;
>plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
>plot2d(&diff(f(x),x),>add,color=red,style="-.-"):
```



Untuk gaya garis ada berbagai pilihan.
- gaya="...". Pilih dari "-", "–", "-.", ".", ".-.", "-.-".
- warna: Lihat di bawah untuk warna.
- ketebalan: Default adalah 1.

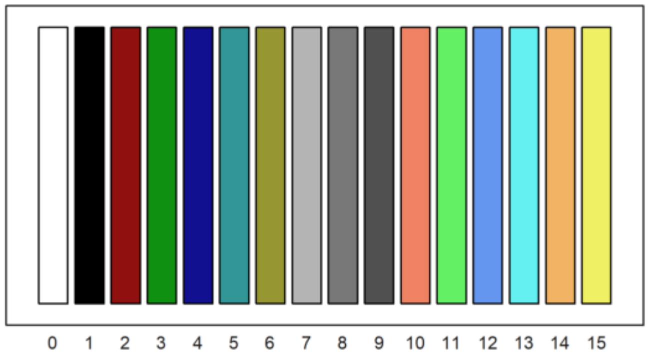
Warna dapat dipilih sebagai salah satu warna default, atau sebagai warna RGB.
- 0.15: indeks warna default.
- konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu muda, abu-abu, abu-abu tua, oranye, hijau muda, pirus, biru muda, oranye terang, kuning
- rgb(merah, hijau, biru): parameter adalah real dalam [0,1].

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



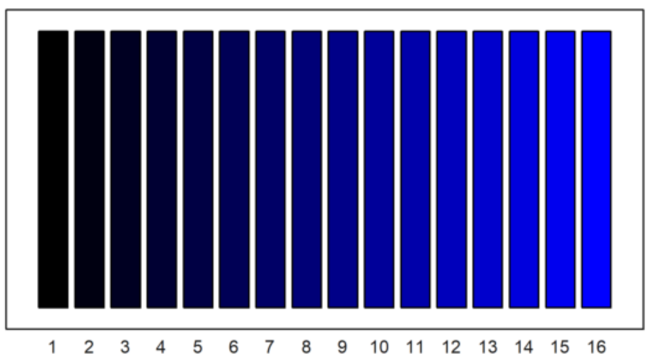
Berikut adalah tampilan warna EMT yang telah ditentukan sebelumnya.

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



Tapi Anda bisa menggunakan warna apa saja.

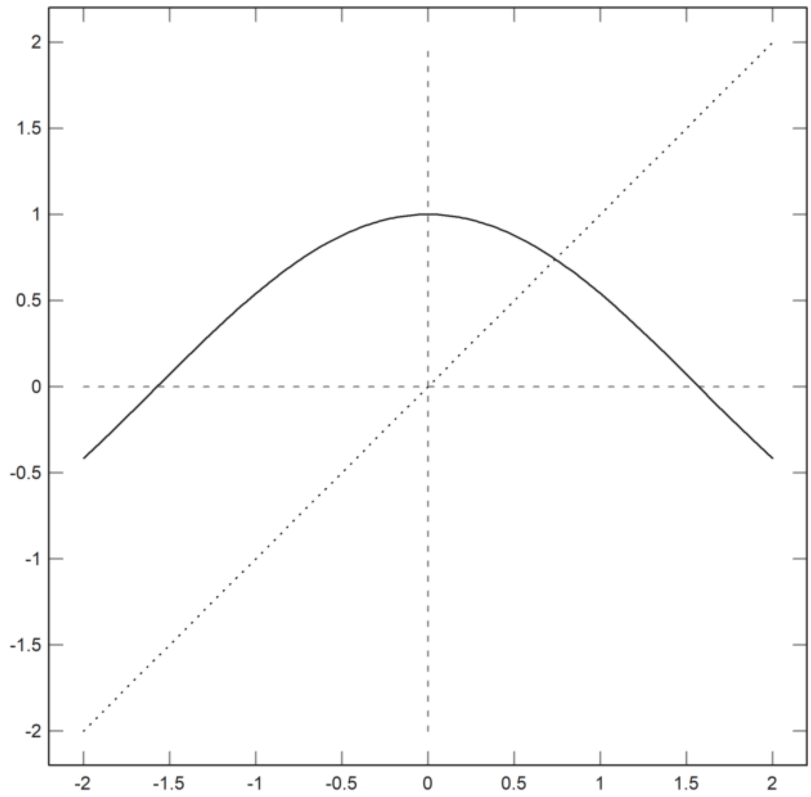
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



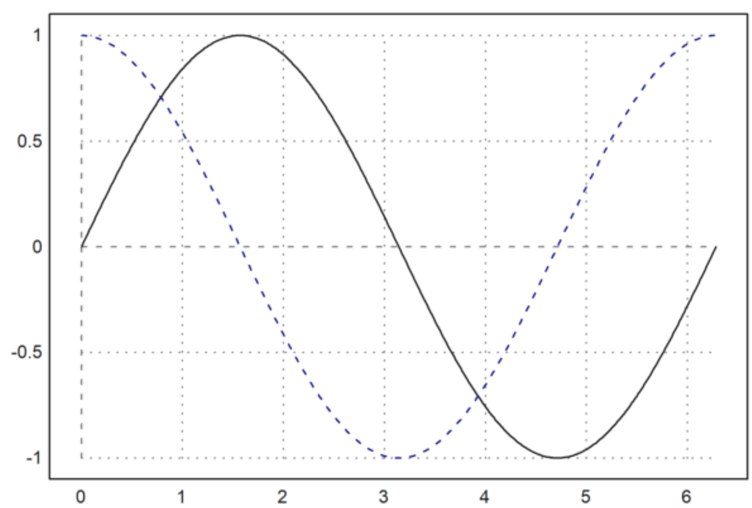
Menggambar Beberapa Kurva pada bidang koordinat yang sama

Plot lebih dari satu fungsi (multiple function) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu metode menggunakan >add untuk beberapa panggilan ke plot2d secara keseluruhan, tetapi panggilan pertama. Kami telah menggunakan fitur ini dalam contoh di atas.

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

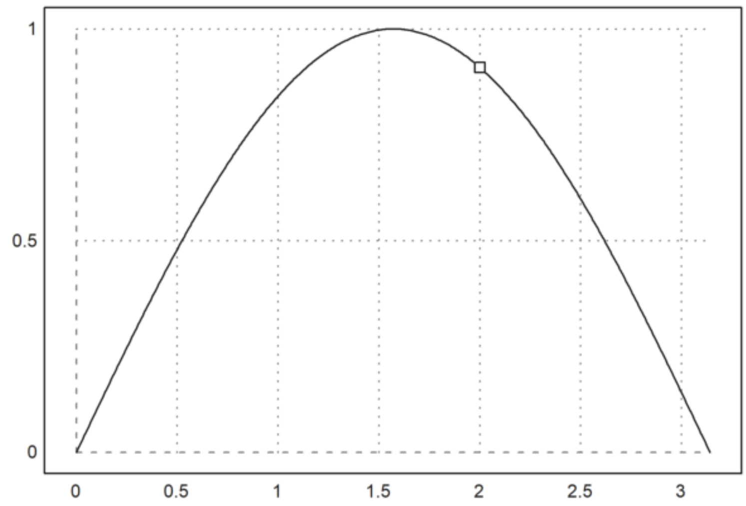



```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



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

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

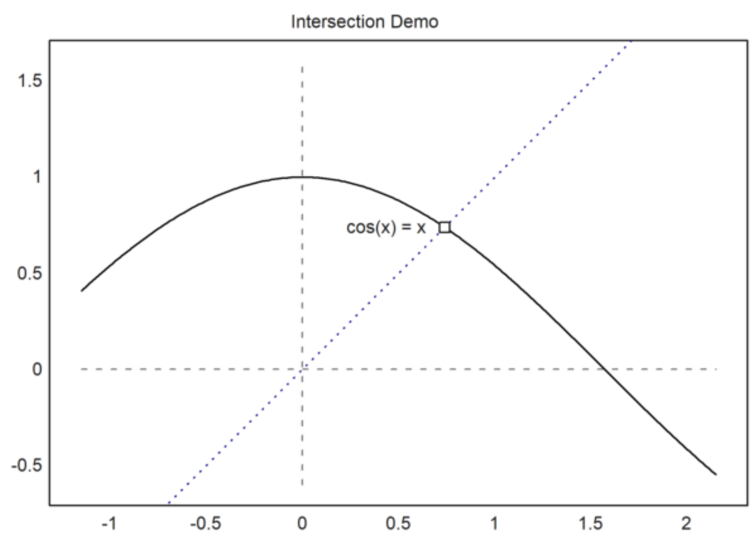


Kami menambahkan titik persimpangan dengan label (pada posisi "cl" untuk kiri tengah), dan memasukkan hasilnya ke dalam notebook. Kami juga menambahkan judul ke plot.

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

```
Illegal parameter after named parameter!
Error in:
... ,   color=[black,blue],style=["-","."],   grid=1); ...
                                                     ^
```

```
>x0=solve("cos(x)-x",1);  ...
>  plot2d(x0,x0,>points,>add,title="Intersection Demo");  ...
>  label("cos(x) = x",x0,x0,pos="cl",offset=20):
```

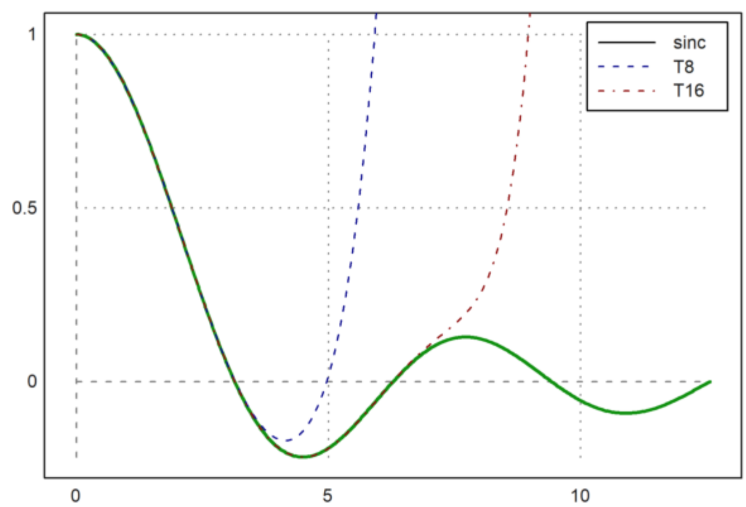


Dalam demo berikut, kami memplot fungsi sinc(x)=sin(x)/x dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolis.
Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke plot2d(). Yang kedua dan yang ketiga memiliki set flag >add, yang membuat plot menggunakan rentang sebelumnya.
Kami menambahkan kotak label yang menjelaskan fungsi.

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

$$\frac{x^4}{120} - \frac{x^2}{6} + 1$$

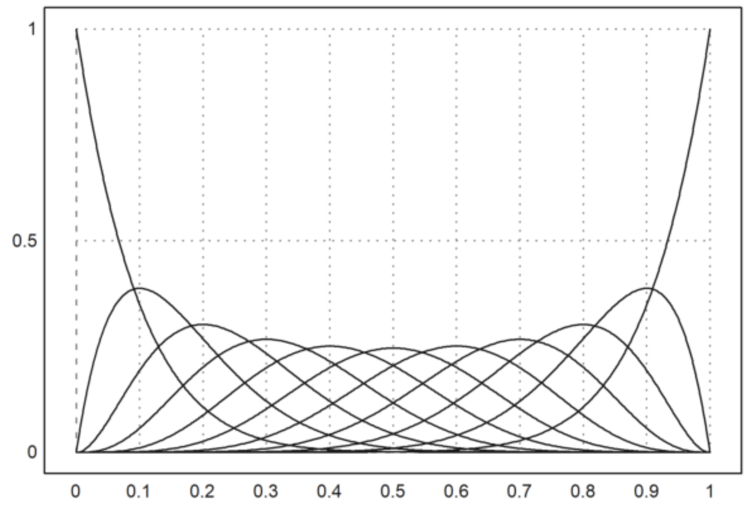
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
>  plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
>  plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
>  labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
>    colors=[black,blue,red]):
```



Dalam contoh berikut, kami menghasilkan Bernstein-Polinomial.

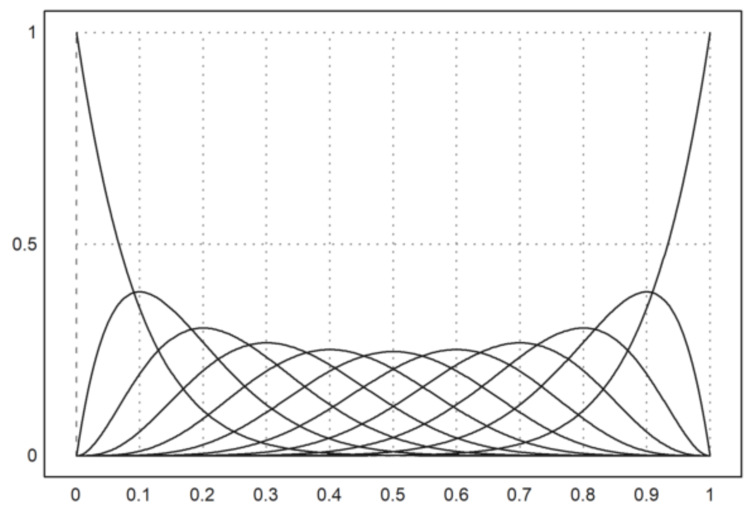
```
B_i(x) = \binom{n}{i} x^i (1-x)^{n-i}
```

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



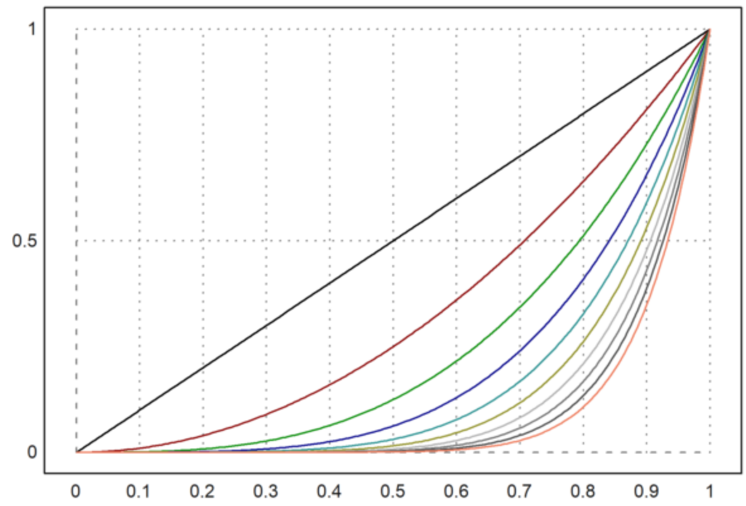
Metode kedua menggunakan pasangan matriks nilai-x dan matriks nilai-y yang berukuran sama.
Kami menghasilkan matriks nilai dengan satu Polinomial Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihat pengantar tentang bahasa matriks untuk mempelajari lebih detail.

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



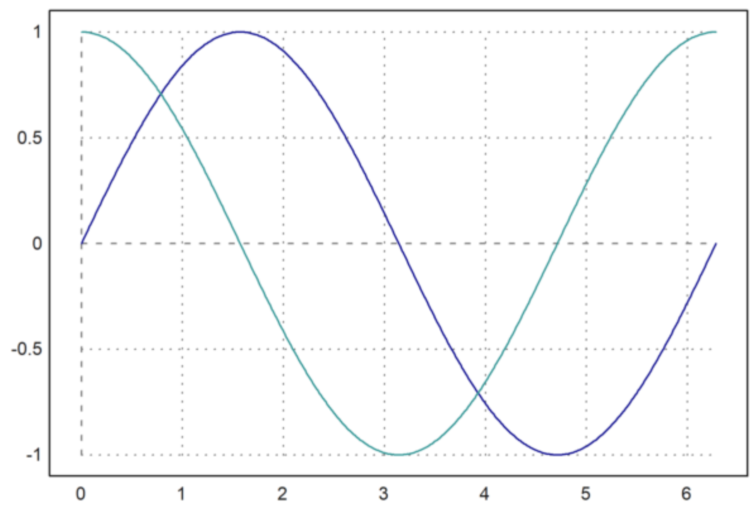
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

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

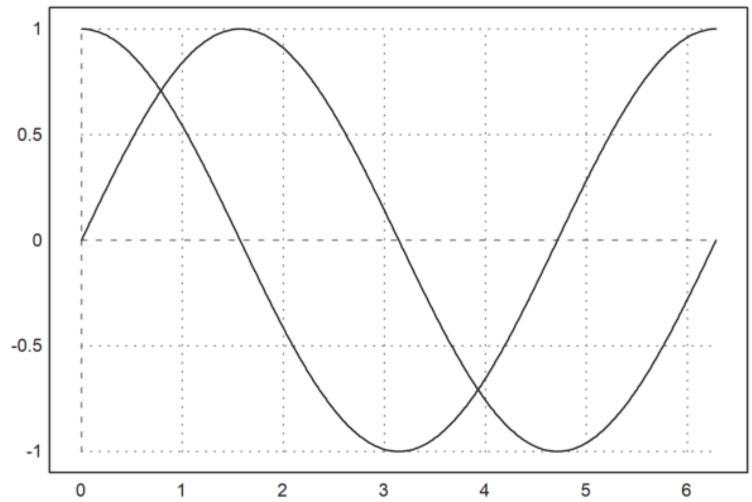


Metode lain adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan larik warna, larik gaya, dan larik ketebalan dengan panjang yang sama.

```
>plot2d(["sin(x)","cos(x)"],0,2pi,color=4:5):
```




```
>plot2d(["sin(x)","cos(x)"],0,2pi): // plot vector of expressions
```



Kita bisa mendapatkan vektor seperti itu dari Maxima menggunakan makelist() dan mxm2str().

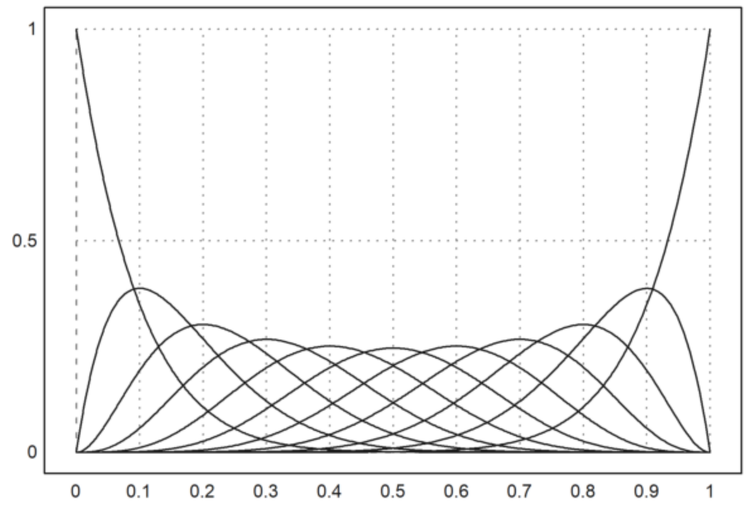
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

```
                10             9                8  2               7  3
       [(1 - x)  , 10 (1 - x)  x, 45 (1 - x)  x , 120 (1 - x)  x ,
           6  4              5  5              4  6              3  7
 210 (1 - x)  x , 252 (1 - x)  x , 210 (1 - x)  x , 120 (1 - x)  x ,
          2  8               9   10
 45 (1 - x)  x , 10 (1 - x) x , x  ]
```

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

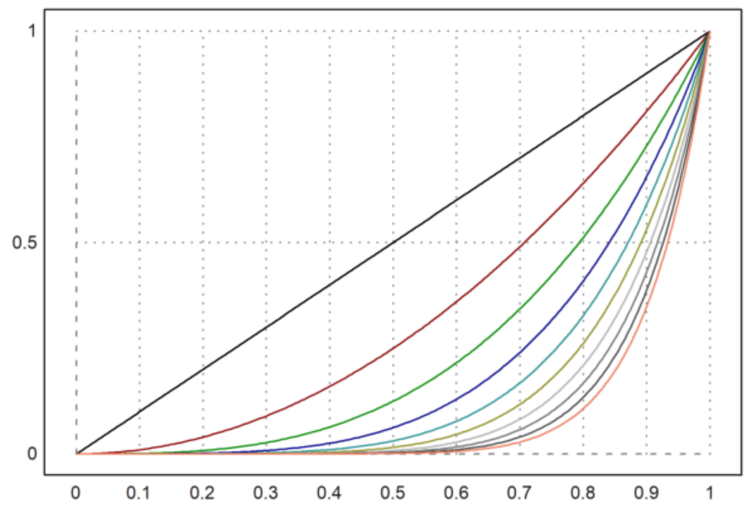
```
 (1-x)^10
 10*(1-x)^9*x
 45*(1-x)^8*x^2
 120*(1-x)^7*x^3
 210*(1-x)^6*x^4
 252*(1-x)^5*x^5
 210*(1-x)^4*x^6
 120*(1-x)^3*x^7
 45*(1-x)^2*x^8
 10*(1-x)*x^9
 x^10
```

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



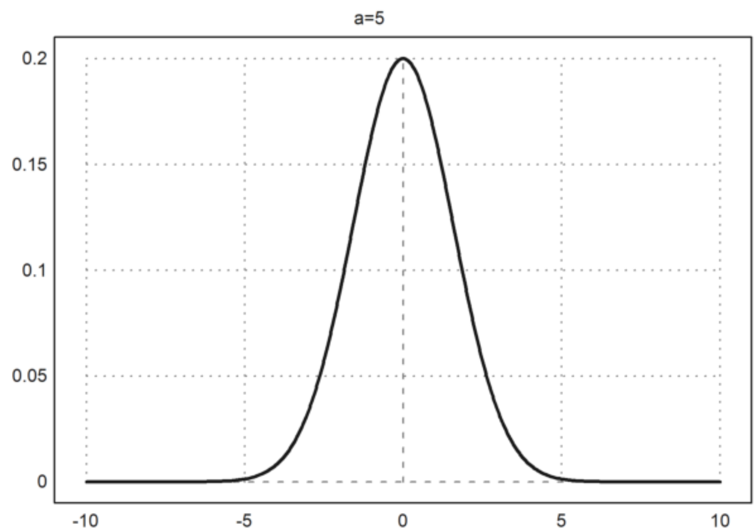
Alternatif lain adalah dengan menggunakan bahasa matriks Euler.
Jika ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, semua fungsi ini akan diplot ke dalam satu plot.
Untuk ini, gunakan vektor parameter dalam bentuk vektor kolom. Jika array warna ditambahkan, itu akan digunakan untuk setiap baris plot.

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



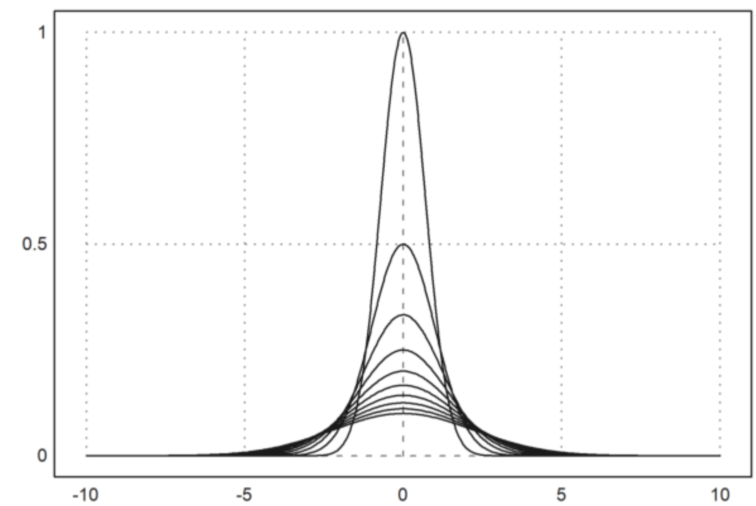
Ekspresi dan fungsi satu baris dapat melihat variabel global.
Jika Anda tidak dapat menggunakan variabel global, Anda perlu menggunakan fungsi dengan parameter tambahan, dan meneruskan parameter ini sebagai parameter titik koma.
Berhati-hatilah, untuk meletakkan semua parameter yang ditetapkan di akhir perintah plot2d. Dalam contoh kita meneruskan a=5 ke fungsi f, yang kita plot dari -10 hingga 10.

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



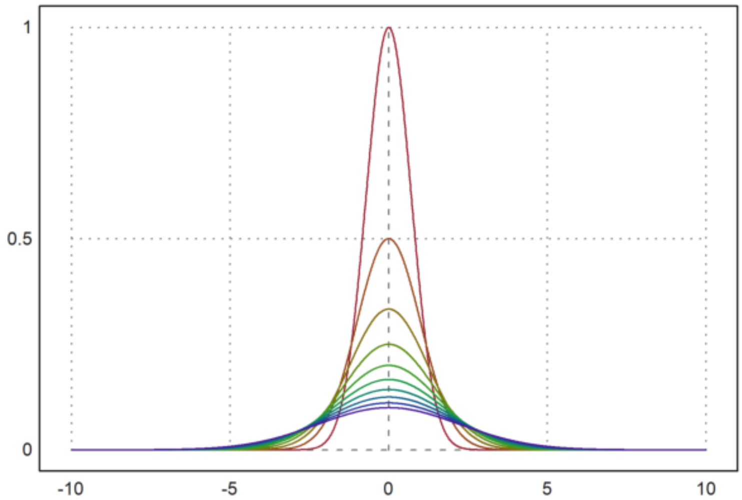
Atau, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut koleksi panggilan, dan itu adalah cara yang lebih disukai untuk meneruskan argumen ke fungsi yang dengan sendirinya diteruskan sebagai argumen ke fungsi lain.
Dalam contoh berikut, kami menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loop).

```
>plot2d({{"f",1}},-10,10); ...
>for a=2:10; plot2d({{"f",a}},>add); end:
```



Kami dapat mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris matriks f(x,a) adalah satu fungsi. Selain itu, kita dapat mengatur warna untuk setiap baris matriks. Klik dua kali pada fungsi getspectral() untuk penjelasannya.

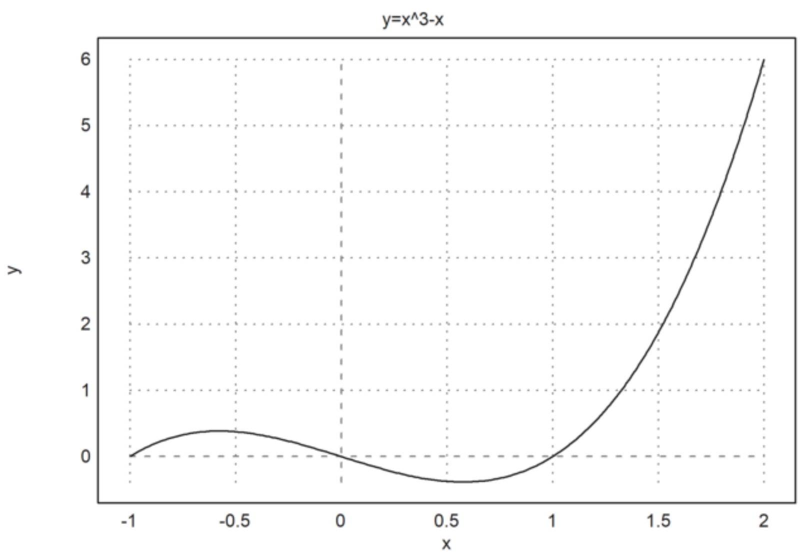
```
>x=-10:0.01:10; a=(1:10)'; plot2d(x,f(x,a),color=getspectral(a/10)):
```



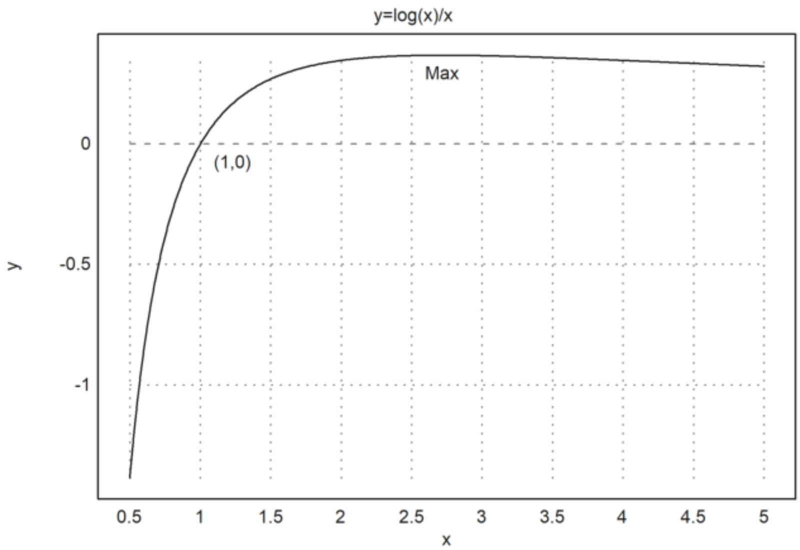
Label Teks

Dekorasi sederhana bisa
- judul dengan judul="..."
- x- dan y-label dengan xl="...", yl="..."
- label teks lain dengan label("...",x,y)
Perintah label akan memplot ke dalam plot saat ini pada koordinat plot (x,y). Itu bisa mengambil argumen posisi.

```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x"):
```

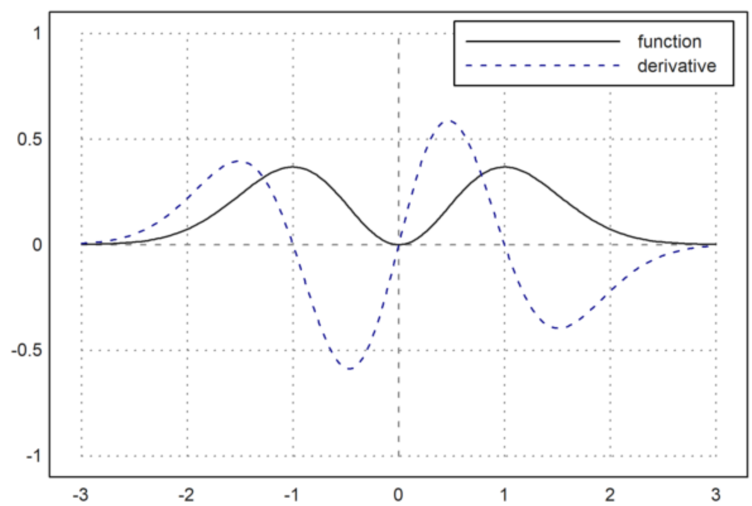



```
>expr := "log(x)/x"; ...
>  plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
>  label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```



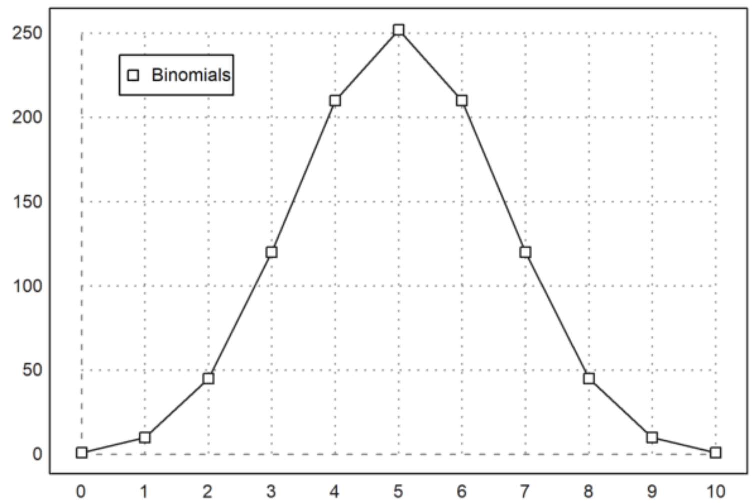
Ada juga fungsi labelbox(), yang dapat menampilkan fungsi dan teks. Dibutuhkan vektor string dan warna, satu item untuk setiap fungsi.

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



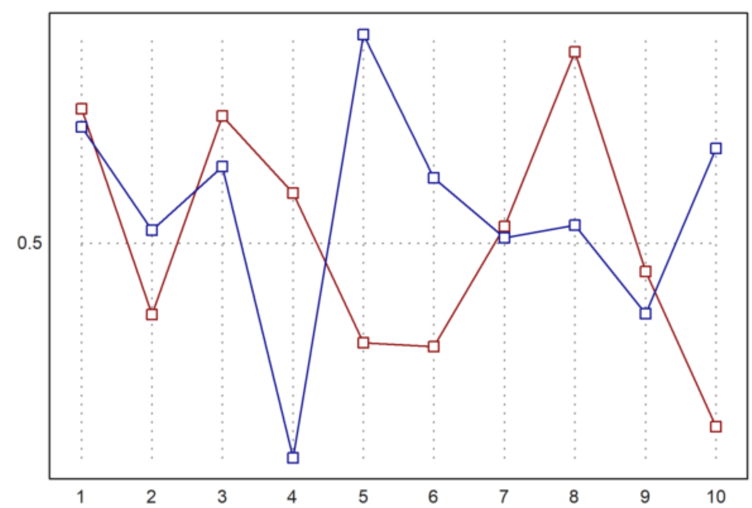
Kotak ditambatkan di kanan atas secara default, tetapi > kiri menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafik. Lebarnya otomatis.
Untuk plot titik, kotak label juga berfungsi. Tambahkan parameter >points, atau vektor flag, satu untuk setiap label.
Dalam contoh berikut, hanya ada satu fungsi. Jadi kita bisa menggunakan string sebagai pengganti vektor string. Kami mengatur warna teks menjadi hitam untuk contoh ini.

```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
>labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
>tcolor=black,>left):
```



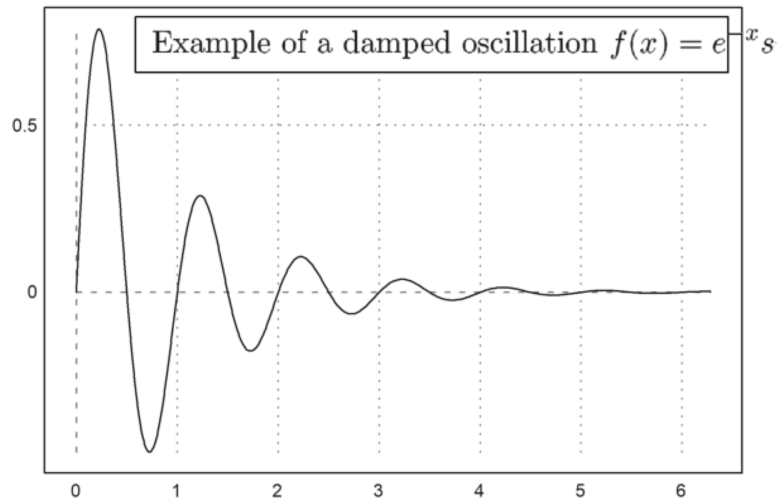
Gaya plot ini juga tersedia di statplot(). Seperti di plot2d() warna dapat diatur untuk setiap baris plot. Ada lebih banyak plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

```
>statplot(1:10,random(2,10),color=[red,blue]):
```



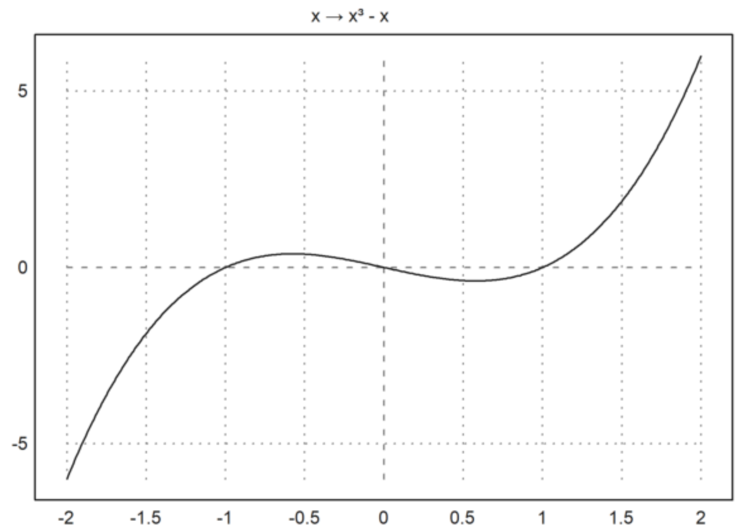
Fitur serupa adalah fungsi textbox().
Lebar secara default adalah lebar maksimal dari baris teks. Tapi itu bisa diatur oleh pengguna juga.

```
>function f(x) &= exp(-x)*sin(2*pi*x); ...
>plot2d("f(x)",0,2pi); ...
>textbox(latex("\text{Example of a damped oscillation}\ f(x)=e^{-x}sin(2\pi x)"),w=0.85):
```



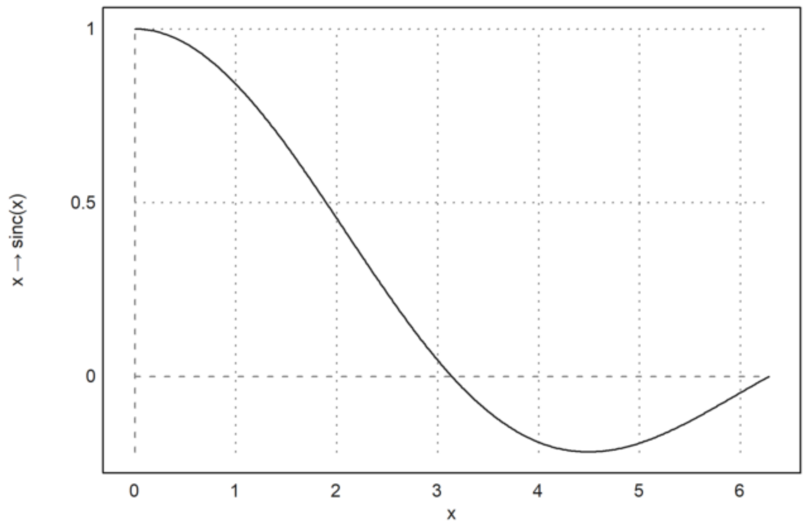
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

```
>plot2d("x^3-x",title=u"x &rarr; x&sup3; - x"):
```



Label pada sumbu x dan y bisa vertikal, begitu juga sumbunya.

```
>plot2d("sinc(x)",0,2pi,xl="x",yl=u"x &rarr; sinc(x)",>vertical):
```

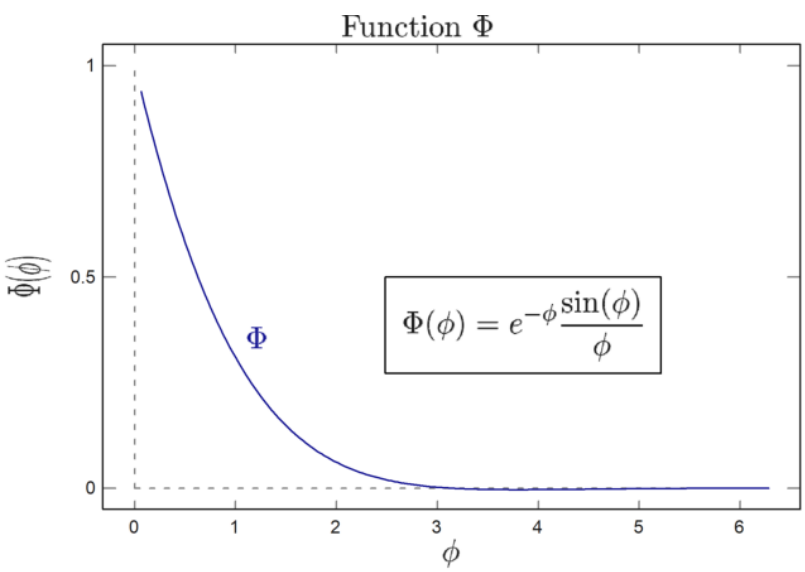


Anda juga dapat memplot rumus LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke biner "lateks" dan "dvipng" harus berada di jalur sistem, atau Anda harus mengatur LaTeX di menu opsi.

Perhatikan, bahwa penguraian LaTeX lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil latex() sebelum loop sekali dan menggunakan hasilnya (gambar dalam matriks RGB).

Dalam plot berikut, kami menggunakan LaTeX untuk label x dan y, label, kotak label, dan judul plot.

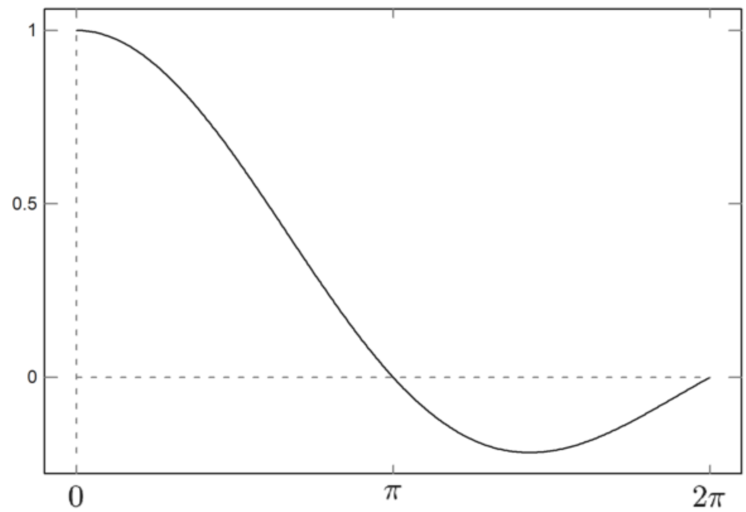
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
>  title=latex("\text{Function $\Phi$}"), ...
>  xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
>textbox( ...
>  latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
>label(latex("\Phi",color=blue),1,0.4):
```



Seringkali, kami menginginkan spasi dan label teks non-konformal pada sumbu x. Kita dapat menggunakan xaxis() dan yaxis() seperti yang akan kita tunjukkan nanti.

Cara termudah adalah dengan membuat plot kosong dengan bingkai menggunakan grid=4, lalu menambahkan grid dengan ygrid() dan xgrid(). Dalam contoh berikut, kami menggunakan tiga string LaTeX untuk label pada sumbu x dengan xtick().

```
>plot2d("sinc(x)",0,2pi,grid=4,<ticks); ...
>ygrid(-2:0.5:2,grid=6); ...
>xgrid([0:2]*pi,<ticks,grid=6);  ...
>xtick([0,pi,2pi],["0","\pi","2\pi"],>latex):
```



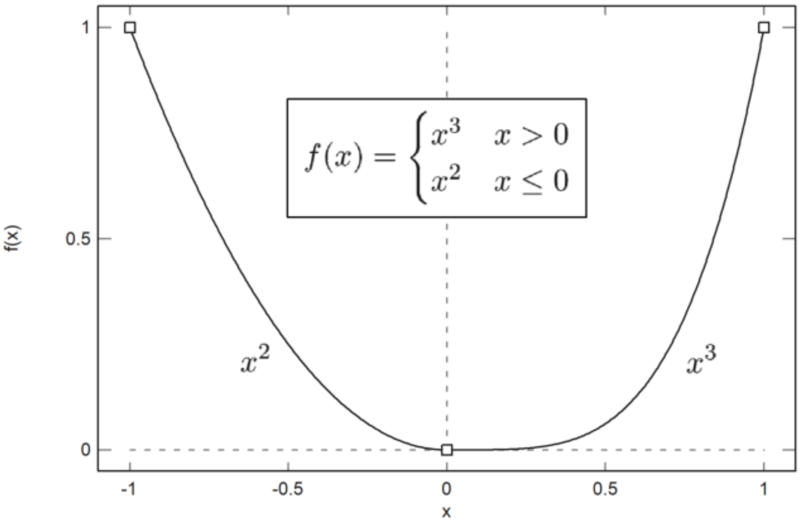
Tentu saja, fungsi juga dapat digunakan.

```
>function map f(x) ...
```

```
  if x>0 then return x^4
  else return x^2
  endif
  endfunction
```

Parameter "peta" membantu menggunakan fungsi untuk vektor. Untuk
plot, itu tidak perlu. Tetapi untuk mendemonstrasikan vektorisasi itu
berguna, kami menambahkan beberapa poin kunci ke plot di x=-1, x=0 dan x=1.
Pada plot berikut, kami juga memasukkan beberapa kode LaTeX. Kami menggunakannya untuk
dua label dan kotak teks. Tentu saja, Anda hanya akan dapat menggunakan
LaTeX jika Anda telah menginstal LaTeX dengan benar.

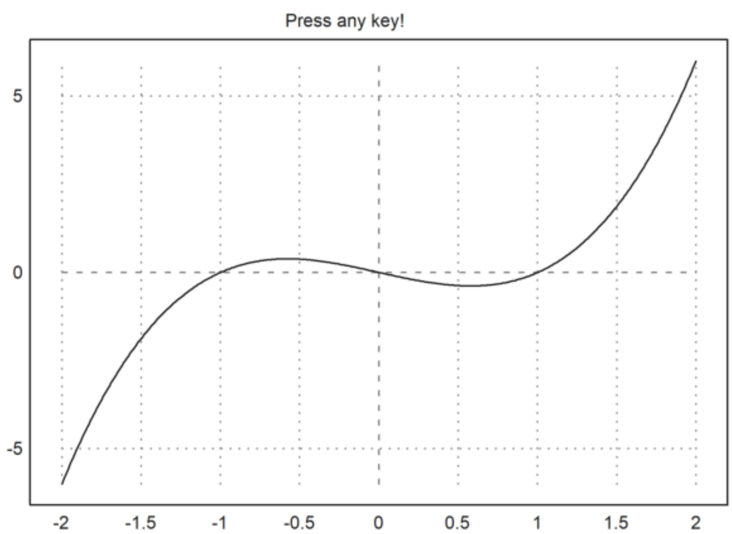
```
>plot2d("f",-1,1,xl="x",yl="f(x)",grid=6);  ...
>plot2d([-1,0,1],f([-1,0,1]),>points,>add); ...
>label(latex("x^3"),0.72,f(0.72)); ...
>label(latex("x^2"),-0.52,f(-0.52),pos="ll"); ...
>textbox( ...
>  latex("f(x)=\begin{cases} x^3 & x>0 \\ x^2 & x \le 0\end{cases}"), ...
>  x=0.7,y=0.2):
```



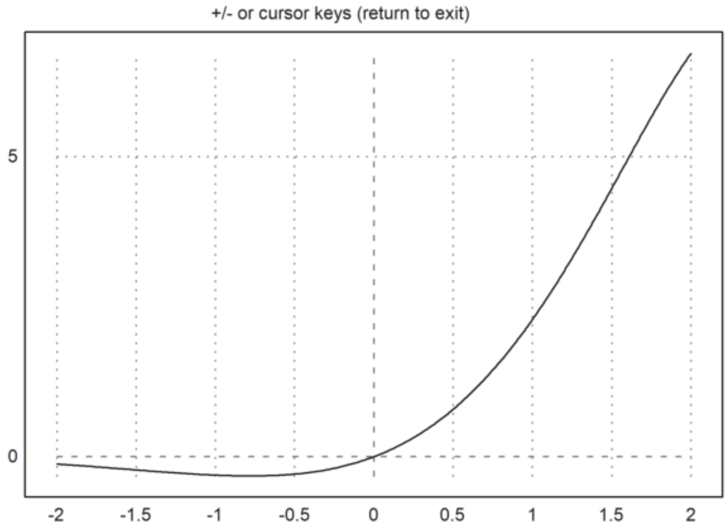
Interaksi pengguna

Saat memplot fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat
- perbesar dengan + atau -
- pindahkan plot dengan tombol kursor
- pilih jendela plot dengan mouse
- atur ulang tampilan dengan spasi
- keluar dengan kembali
Tombol spasi akan mengatur ulang plot ke jendela plot asli.
Saat memplot data, flag >user hanya akan menunggu penekanan tombol.

```
>plot2d({{"x^3-a*x",a=1}},>user,title="Press any key!"):
```




```
>plot2d("exp(x)*sin(x)",user=true, ...
>  title="+/- or cursor keys (return to exit)"):
```



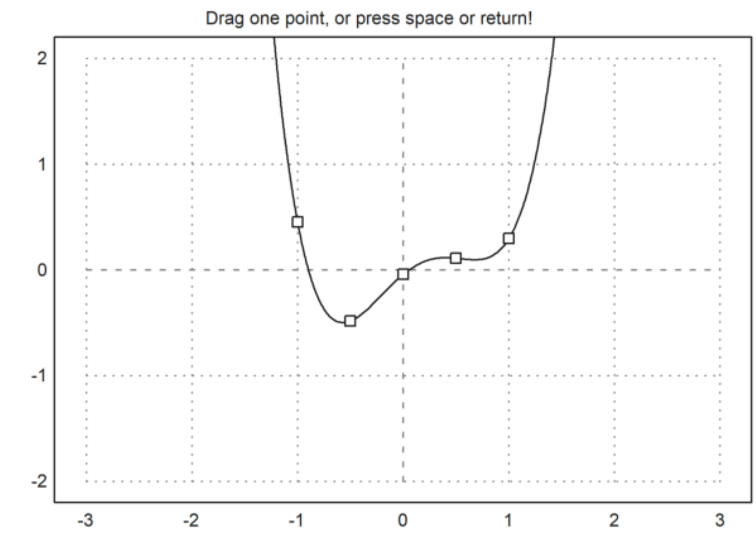
Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut (lihat tutorial tentang pemrograman untuk detailnya).
Fungsi bawaan mousedrag() menunggu event mouse atau keyboard. Ini melaporkan mouse ke bawah, mouse dipindahkan atau mouse ke atas, dan penekanan tombol. Fungsi dragpoints() memanfaatkan ini, dan memungkinkan pengguna menyeret titik mana pun dalam plot.
Kita membutuhkan fungsi plot terlebih dahulu. Sebagai contoh, kita interpolasi dalam 5 titik dengan polinomial. Fungsi harus diplot ke area plot tetap.

```
>function plotf(xp,yp,select) ...
```

```
    d=interp(xp,yp);
    plot2d("interpval(xp,d,x)";d,xp,r=2);
    plot2d(xp,yp,>points,>add);
    if select>0 then
      plot2d(xp[select],yp[select],color=red,>points,>add);
    endif;
    title("Drag one point, or press space or return!");
  endfunction
```

Perhatikan parameter titik koma di plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, mengakses nilai secara global.
Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret poin.

```
>t=-1:0.5:1; dragpoints("plotf",t,random(size(t))-0.5):
```

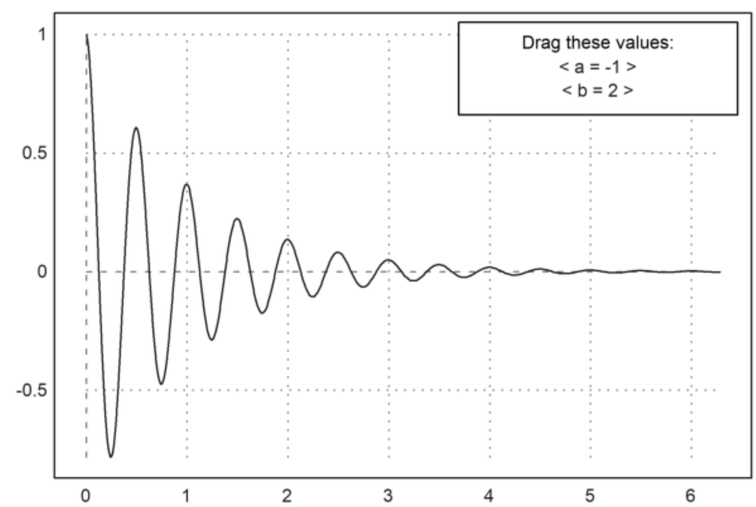


Ada juga fungsi, yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.
Pertama kita membutuhkan fungsi plot.

```
>function plotf([a,b]) := plot2d("exp(a*x)*cos(2pi*b*x)",0,2pi;a,b);
```

Kemudian kita membutuhkan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional baris judul.
Ada slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

```
>dragvalues("plotf",["a","b"],[-1,2],[[-2,2];[1,10]], ...
>  heading="Drag these values:",hcolor=black):
```



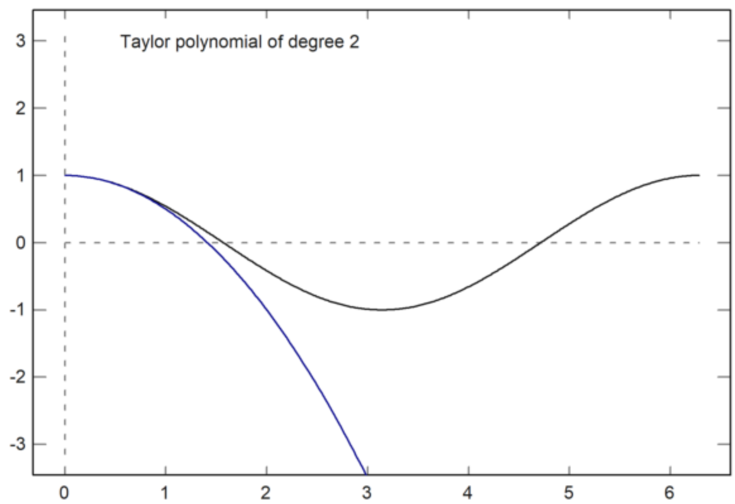
Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor derajat n ke fungsi kosinus.

```
>function plotf(n) ...
```

```
  plot2d("cos(x)",0,2pi,>square,grid=6);
  plot2d(&"taylor(cos(x),x,0,@n)",color=blue,>add);
  textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",>left);
  endfunction
```

Sekarang kami mengizinkan derajat n bervariasi dari 0 hingga 20 dalam 20 pemberhentian. Hasil dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk memasukkan plot ke dalam buku catatan.

```
>nd=dragvalues("plotf","degree",2,[0,20],20,y=0.8, ...
>   heading="Drag the value:"); ...
>plotf(nd):
```



Berikut ini adalah demonstrasi sederhana dari fungsi tersebut. Pengguna dapat menggambar di atas jendela plot, meninggalkan jejak poin.

```
>function dragtest ...
```

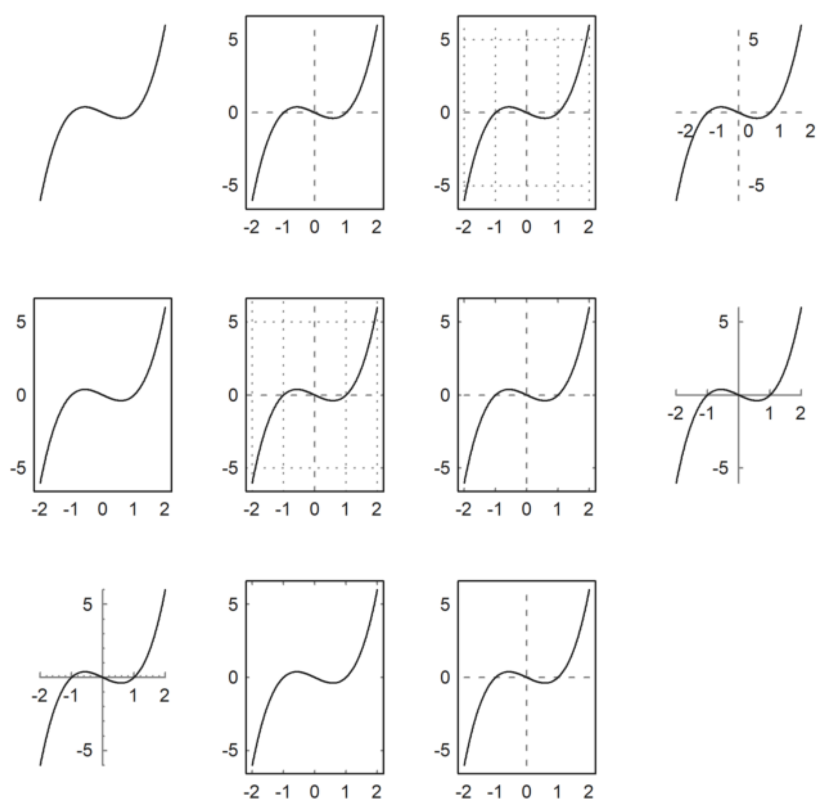
```
    plot2d(none,r=1,title="Drag with the mouse, or press any key!");
    start=0;
    repeat
      {flag,m,time}=mousedrag();
      if flag==0 then return; endif;
      if flag==2 then
        hold on; mark(m[1],m[2]); hold off;
      endif;
    end
  endfunction
```

```
>dragtest // lihat hasilnya dan cobalah lakukan!
```

Gaya Plot 2D

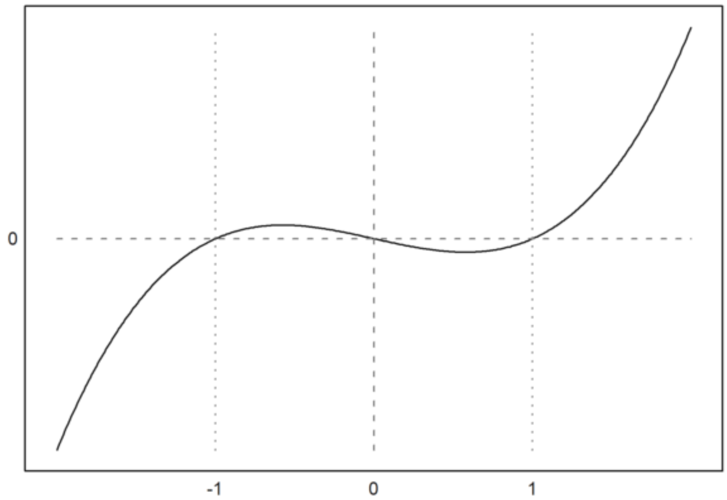
Secara default, EMT menghitung tick sumbu otomatis dan menambahkan label ke setiap tick. Ini dapat diubah dengan parameter grid. Gaya default sumbu dan label dapat dimodifikasi. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang ke gaya default, gunakan reset().

```
>aspect();
>figure(3,4); ...
> figure(1); plot2d("x^3-x",grid=0); ... // no grid, frame or axis
> figure(2); plot2d("x^3-x",grid=1); ... // x-y-axis
> figure(3); plot2d("x^3-x",grid=2); ... // default ticks
> figure(4); plot2d("x^3-x",grid=3); ... // x-y- axis with labels inside
> figure(5); plot2d("x^3-x",grid=4); ... // no ticks, only labels
> figure(6); plot2d("x^3-x",grid=5); ... // default, but no margin
> figure(7); plot2d("x^3-x",grid=6); ... // axes only
> figure(8); plot2d("x^3-x",grid=7); ... // axes only, ticks at axis
> figure(9); plot2d("x^3-x",grid=8); ... // axes only, finer ticks at axis
> figure(10); plot2d("x^3-x",grid=9); ... // default, small ticks inside
> figure(11); plot2d("x^3-x",grid=10); ...// no ticks, axes only
> figure(0):
```



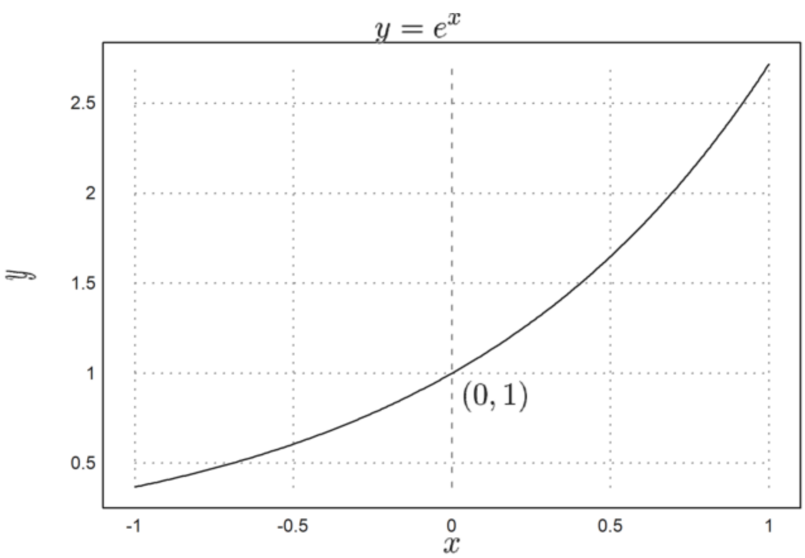
Parameter <frame mematikan frame, dan framecolor=blue mengatur frame ke warna biru.
Jika Anda ingin centang sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

```
>aspect(1.5);
>plot2d("x^3-x",grid=0); // plot
>frame; xgrid([-1,0,1]); ygrid(0): // add frame and grid
```



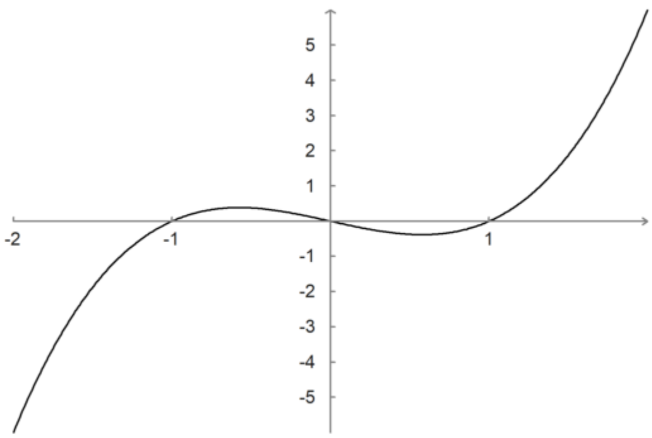
Untuk judul plot dan label sumbu, lihat contoh berikut.

```
>plot2d("exp(x)",-1,1);
>textcolor(black); // set the text color to black
>title(latex("y=e^x")); // title above the plot
>xlabel(latex("x")); // "x" for x-axis
>ylabel(latex("y"),>vertical); // vertical "y" for y-axis
>label(latex("(0,1)"),0,1,color=blue): // label a point
```



Sumbu dapat digambar secara terpisah dengan xaxis() dan yaxis().

```
>plot2d("x^3-x",<grid,<frame);
>xaxis(0,xx=-2:1,style="->"); yaxis(0,yy=-5:5,style="->"):
```

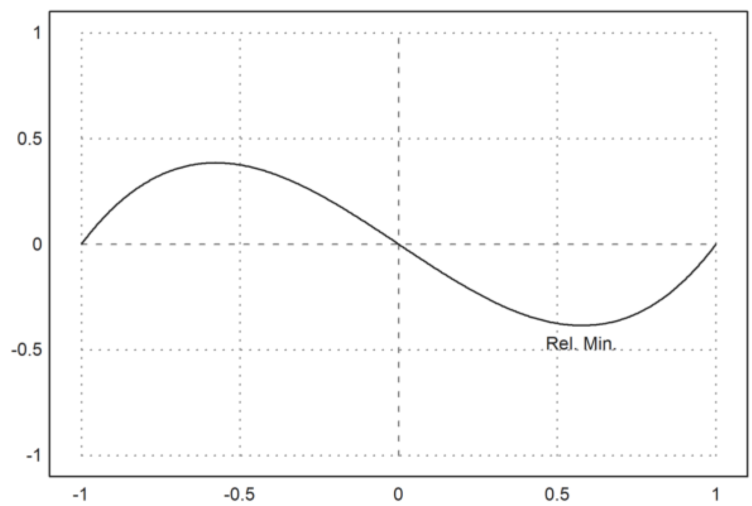


Teks pada plot dapat diatur dengan label(). Dalam contoh berikut, "lc" berarti tengah bawah. Ini mengatur posisi label relatif terhadap koordinat plot.

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

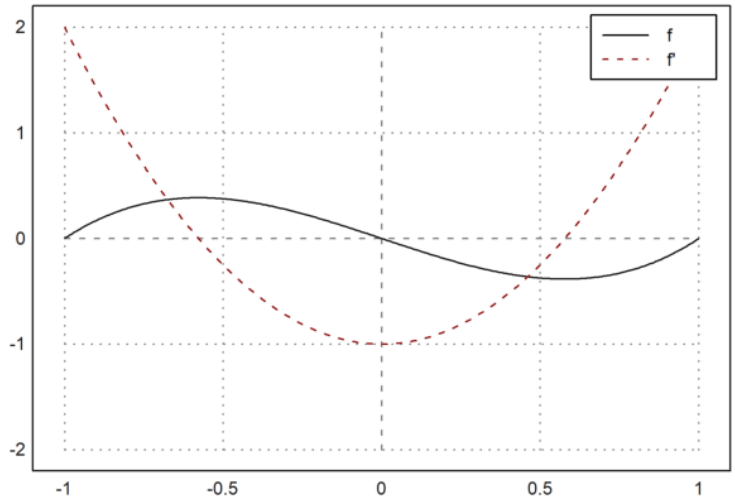
```
                              3
                             x  - x
```

```
>plot2d(f,-1,1,>square);
>x0=fmin(f,0,1); // compute point of minimum
>label("Rel. Min.",x0,f(x0),pos="lc"): // add a label there
```

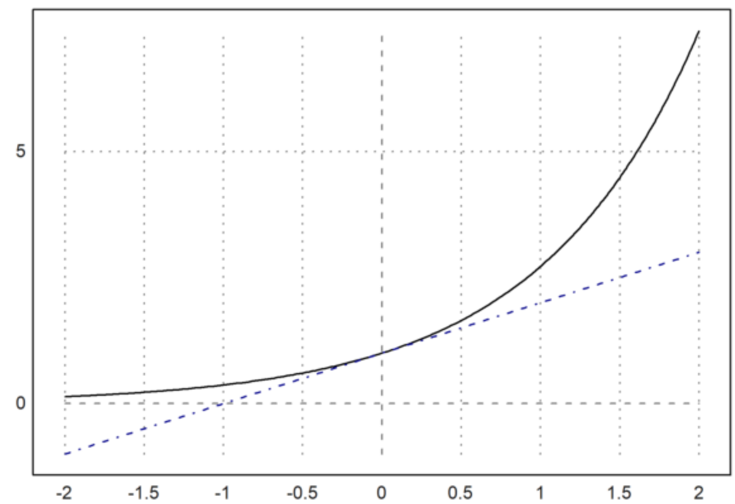


Ada juga kotak teks.

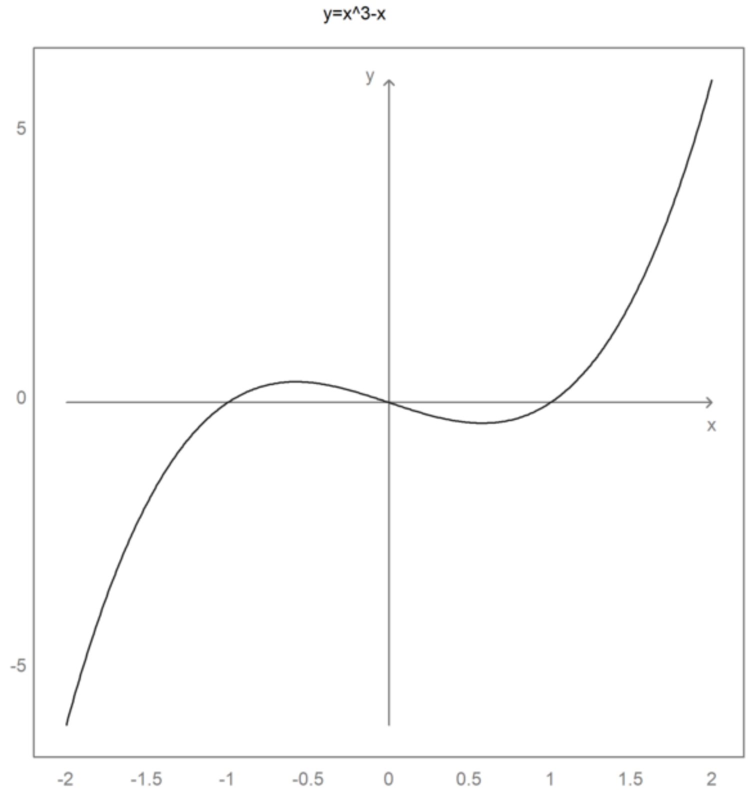
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
```




```
>plot2d(["exp(x)","1+x"],color=[black,blue],style=["-","-.-"]):
```

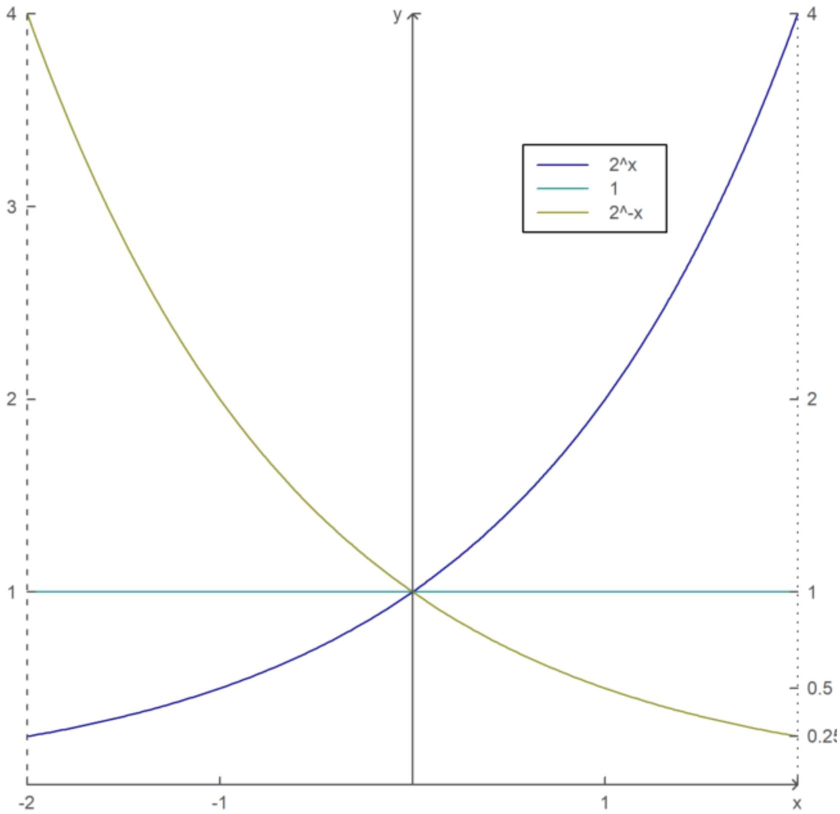



```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray);  ...
> plot2d("x^3-x",grid=1);   ...
> settitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray);  ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```



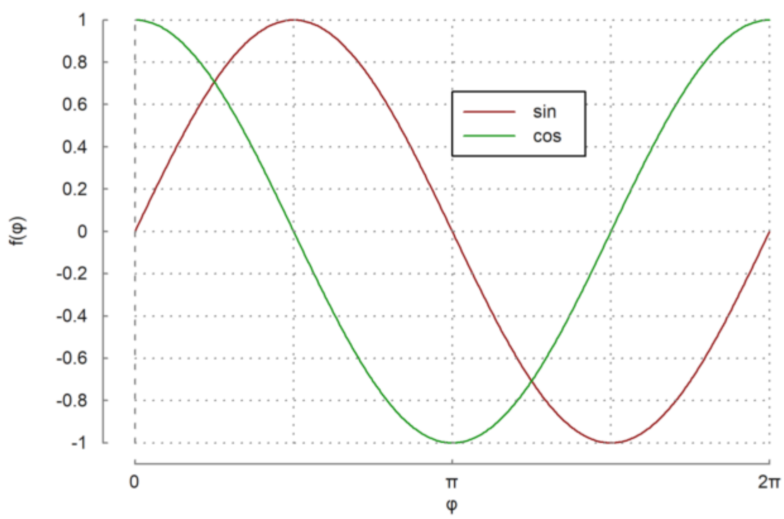
Untuk kontrol lebih, sumbu x dan sumbu y dapat dilakukan secara manual.
Perintah fullwindow() memperluas jendela plot karena kita tidak lagi membutuhkan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk mengatur ulang ke default.

```
>fullwindow; ...
> gridstyle(color=darkgray,textcolor=darkgray); ...
> plot2d(["2^x","1","2^(-x)"],a=-2,b=2,c=0,d=4,<grid,color=4:6,<frame); ...
> xaxis(0,-2:1,style="->"); xaxis(0,2,"x",<axis); ...
> yaxis(0,4,"y",style="->"); ...
> yaxis(-2,1:4,>left); ...
> yaxis(2,2^(-2:2),style=".",<left); ...
> labelbox(["2^x","1","2^-x"],colors=4:6,x=0.8,y=0.2); ...
> reset:
```



Berikut adalah contoh lain, di mana string Unicode digunakan dan sumbu di luar area plot.

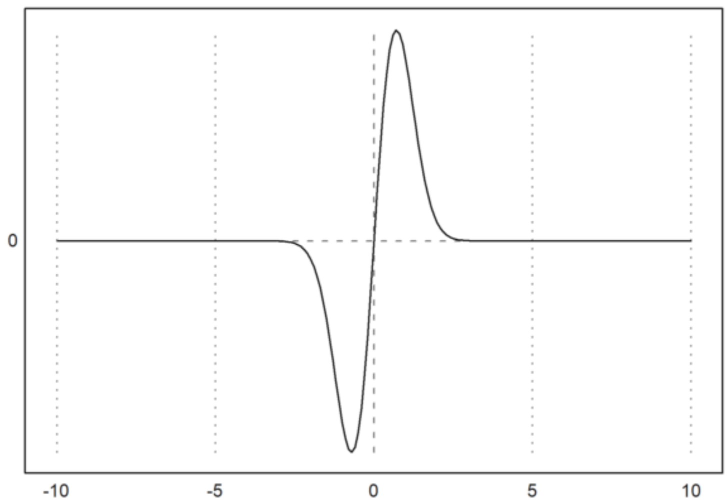
```
>aspect(1.5);
>plot2d(["sin(x)","cos(x)"],0,2pi,color=[red,green],<grid,<frame); ...
> xaxis(-1.1,(0:2)*pi,xt=["0",u"&pi;",u"2&pi;"],style="-",>ticks,>zero);  ...
> xgrid((0:0.5:2)*pi,<ticks); ...
> yaxis(-0.1*pi,-1:0.2:1,style="-",>zero,>grid); ...
> labelbox(["sin","cos"],colors=[red,green],x=0.5,y=0.2,>left); ...
> xlabel(u"&phi;"); ylabel(u"f(&phi;)"):
```



Merencanakan Data 2D

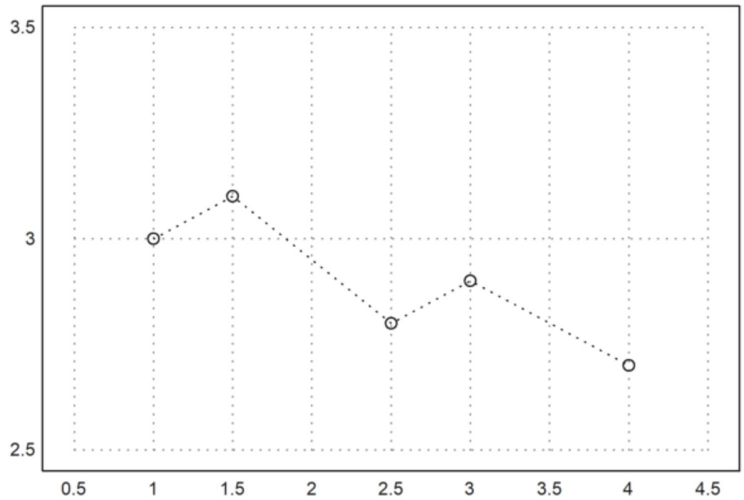
Jika x dan y adalah vektor data, data ini akan digunakan sebagai koordinat x dan y dari suatu kurva. Dalam hal ini, a, b, c, dan d, atau radius r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Atau, >persegi dapat diatur untuk menjaga rasio aspek persegi.
Memplot ekspresi hanyalah singkatan untuk plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai x, dan satu atau beberapa baris nilai y. Dari rentang dan nilai-x, fungsi plot2d akan menghitung data yang akan diplot, secara default dengan evaluasi fungsi yang adaptif. Untuk plot titik gunakan ">titik", untuk garis campuran dan titik gunakan ">tambahan".
Tetapi Anda dapat memasukkan data secara langsung.
- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y diplot baris demi baris.
Berikut adalah contoh dengan satu baris untuk x dan y.

```
>x=-10:0.1:10; y=exp(-x^2)*x; plot2d(x,y):
```

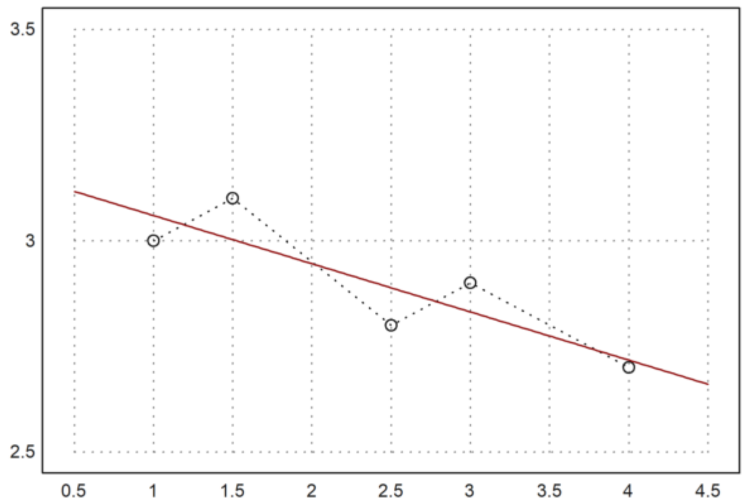


Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plotnya bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya.
- style="...": Pilih dari "[]", "<>", "o", ".", "..", "+", "*", "[]", "< >", "o", "..", "", "|".
Untuk memplot set poin gunakan >points. Jika warna adalah vektor warna, setiap titik
mendapat warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna berlaku untuk baris matriks.
Parameter >addpoints menambahkan titik ke segmen garis untuk plot data.

```
>xdata=[1,1.5,2.5,3,4]; ydata=[3,3.1,2.8,2.9,2.7]; // data
>plot2d(xdata,ydata,a=0.5,b=4.5,c=2.5,d=3.5,style="."); // lines
>plot2d(xdata,ydata,>points,>add,style="o"): // add points
```




```
>p=polyfit(xdata,ydata,1); // get regression line
>plot2d("polyval(p,x)",>add,color=red): // add plot of line
```

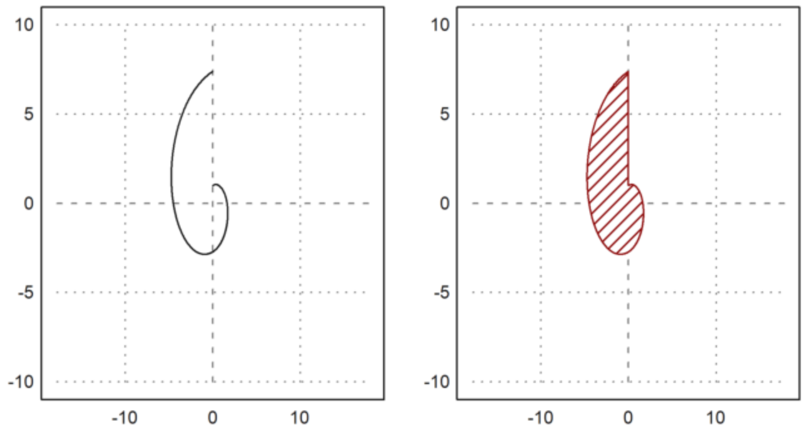


Menggambar Daerah Yang Dibatasi Kurva

Plot data benar-benar poligon. Kita juga dapat memplot kurva atau kurva terisi.
- terisi=true mengisi plot.
- style="...": Pilih dari "", "/", "\", "\/".
- fillcolor: Lihat di atas untuk warna yang tersedia.
Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional mencegah menggambar batas untuk semua gaya kecuali yang default.

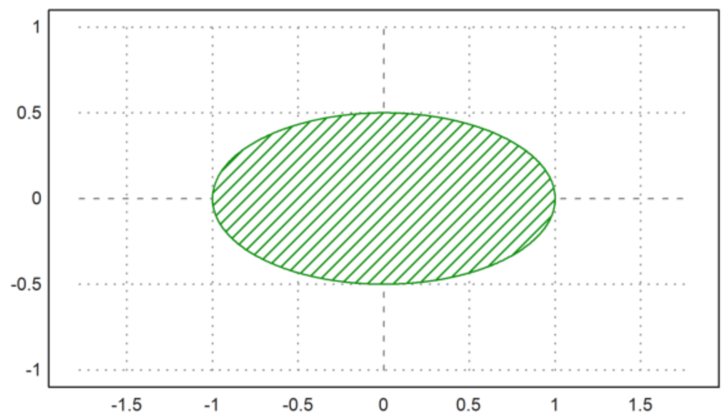
```
>t=linspace(0,2pi,1000); // parameter for curve
>x=sin(t)*exp(t/pi); y=cos(t)*exp(t/pi); // x(t) and y(t)
>figure(1,2); aspect(16/9)
>figure(1); plot2d(x,y,r=10); // plot curve
```

```
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // fill curve
>figure(0):
```

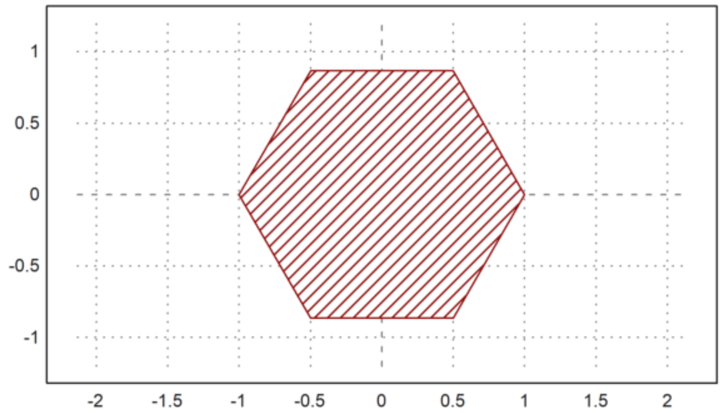


Dalam contoh berikut kami memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian berbeda.

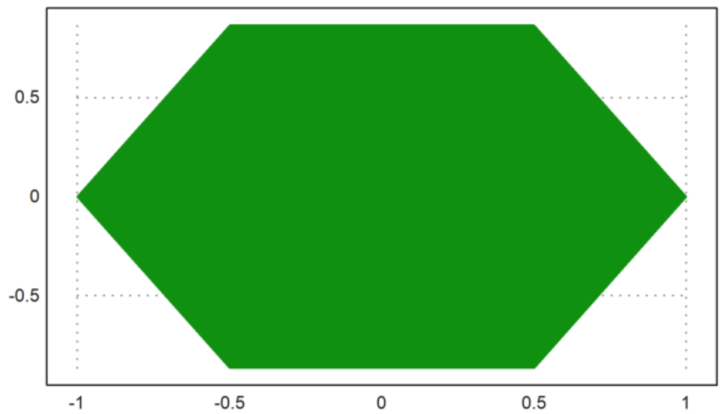
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```




```
>t=linspace(0,2pi,6); ...
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.2):
```

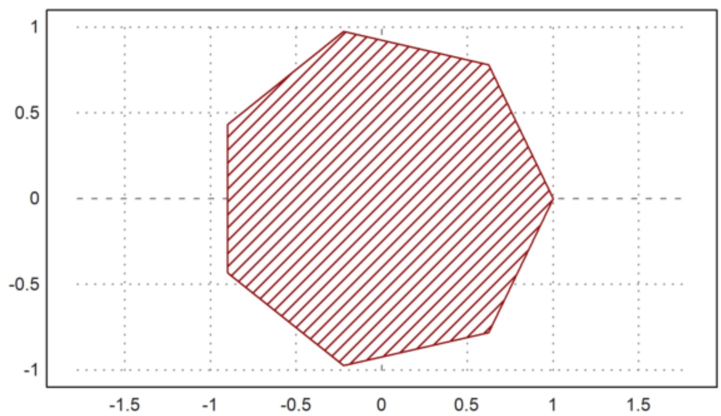



```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



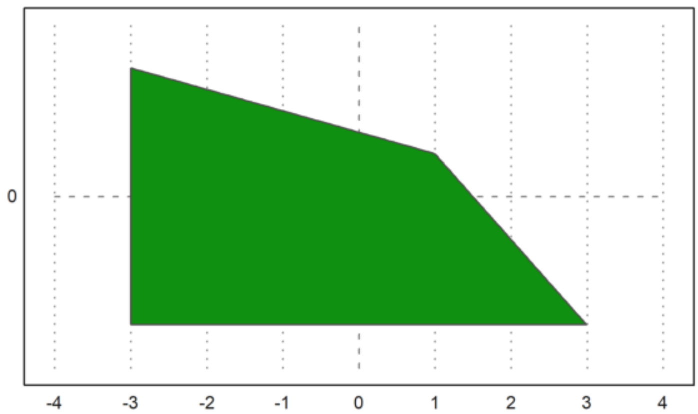
Contoh lainnya adalah segi empat, yang kita buat dengan 7 titik pada lingkaran satuan.

```
>t=linspace(0,2pi,7);  ...
> plot2d(cos(t),sin(t),r=1,>filled,style="/",fillcolor=red):
```



Berikut ini adalah himpunan nilai maksimal dari empat kondisi linier yang kurang dari atau sama dengan 3. Ini adalah A[k].v<=3 untuk semua baris A. Untuk mendapatkan sudut yang bagus, kita menggunakan n yang relatif besar.

```
>A=[2,1;1,2;-1,0;0,-1];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=4,level=[0;3],color=green,n=111):
```

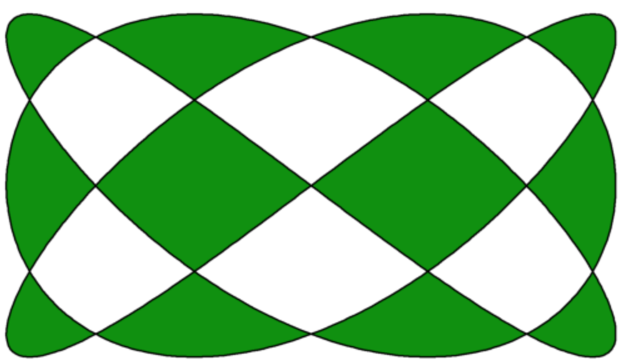


Poin utama dari bahasa matriks adalah memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);
```

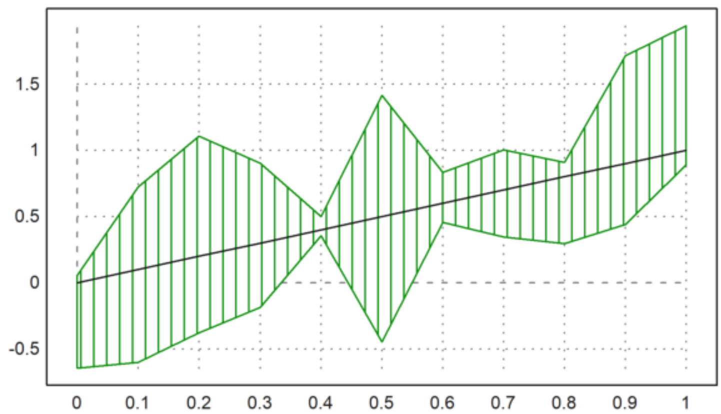
Kami sekarang memiliki vektor x dan y nilai. plot2d() dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plotnya bisa diisi. Pada kasus ini ini menghasilkan hasil yang bagus karena aturan lilitan, yang digunakan untuk isi.

```
>plot2d(x,y,<grid,<frame,>filled):
```



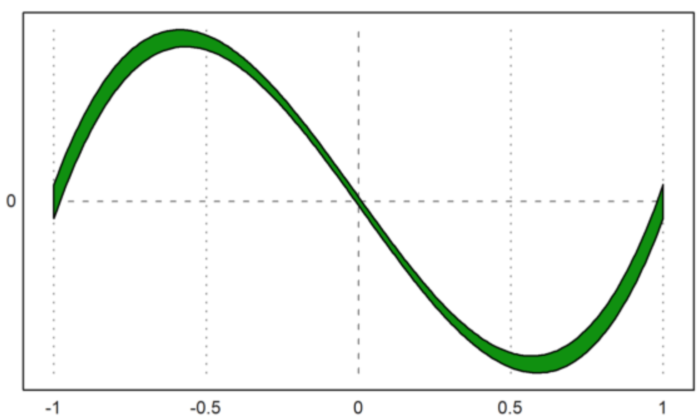
Sebuah vektor interval diplot terhadap nilai x sebagai daerah terisi
antara nilai interval bawah dan atas.
Ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa
juga digunakan untuk memplot kesalahan statistik.

```
>t=0:0.1:1; ...
> plot2d(t,interval(t-random(size(t)),t+random(size(t))),style="|");  ...
> plot2d(t,t,add=true):
```



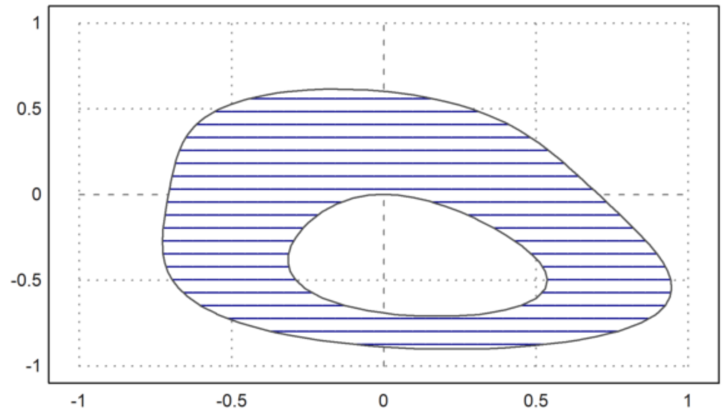
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

```
>t=-1:0.01:1; x=~t-0.01,t+0.01~; y=x^3-x;
>plot2d(t,y):
```



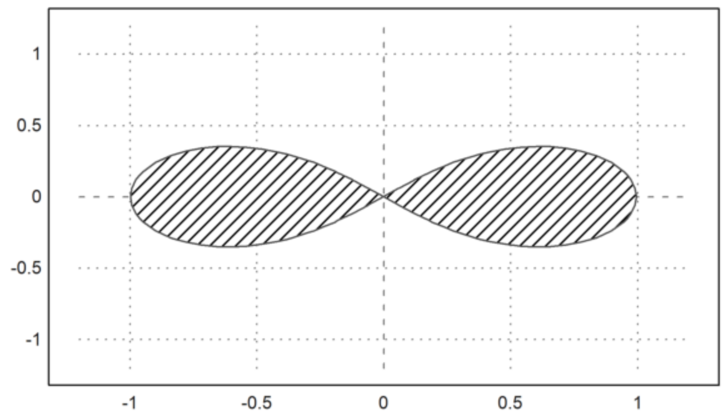
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk
ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah
dan baris kedua berisi batas atas.

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```

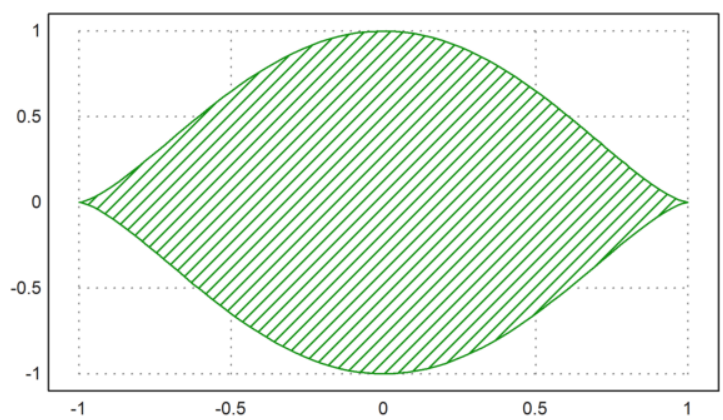


Kami juga dapat mengisi rentang nilai seperti
lateks: -1 \le (x^2+y^2)^2-x^2+y^2 \le 0.

```
>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/"):
```



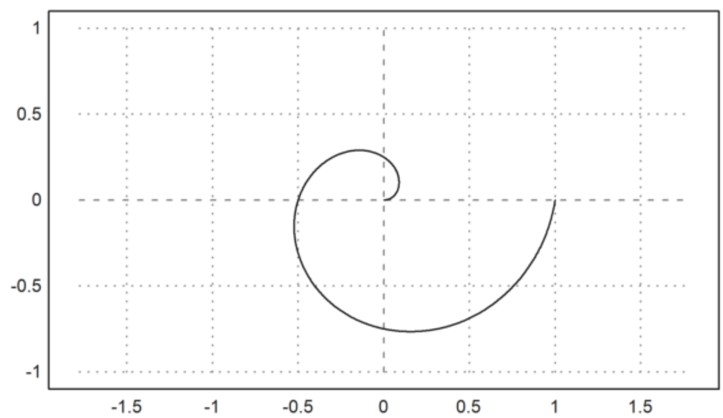

```
>plot2d("cos(x)","sin(x)^3",xmin=0,xmax=2pi,>filled,style="/"):
```



Menggambar Grafik Bilangan Kompleks

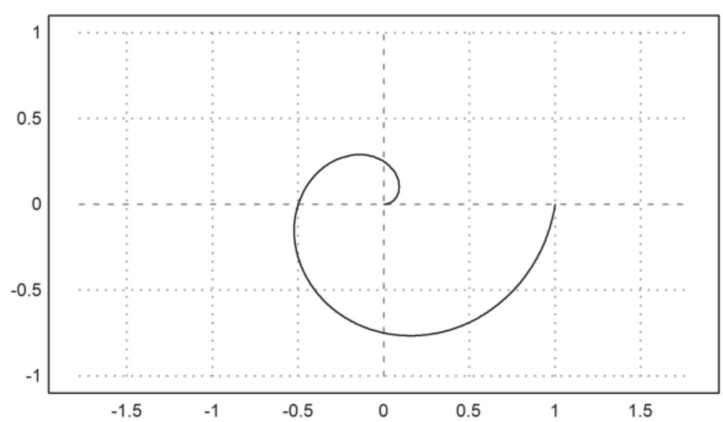
Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1x2) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.
Dalam contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

```
>t=linspace(0,1,1000); ...
>plot2d(t*cos(2*pi*t),t*sin(2*pi*t),r=1):
```

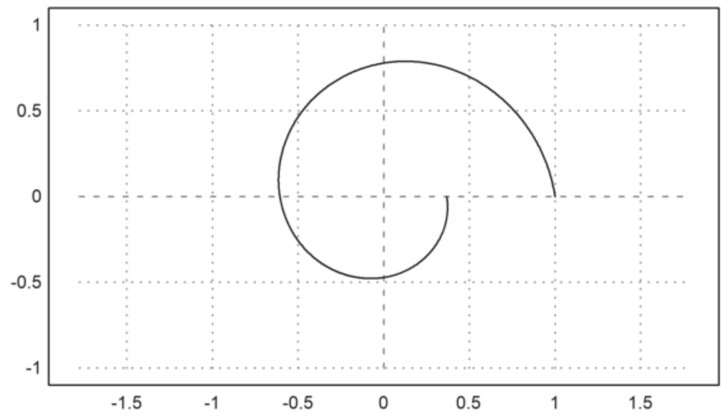


Atau, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

```
>plot2d("x*cos(2*pi*x)","x*sin(2*pi*x)",xmin=0,xmax=1,r=1):
```




```
>t=linspace(0,1,1000); r=exp(-t); x=r*cos(2pi*t); y=r*sin(2pi*t);
>plot2d(x,y,r=1):
```



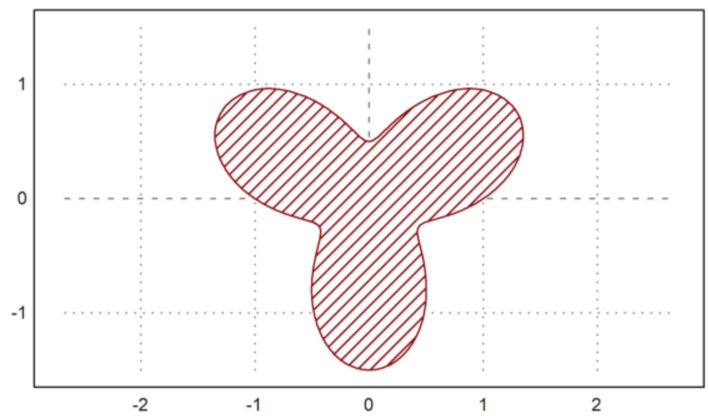
Dalam contoh, kami memplot lingkaran satuan dengan

$$\gamma(t) = (r(t)\cos(t), r(t)\sin(t))$$

with

$$r(t) = 1 + \frac{\sin(3t)}{2}.$$

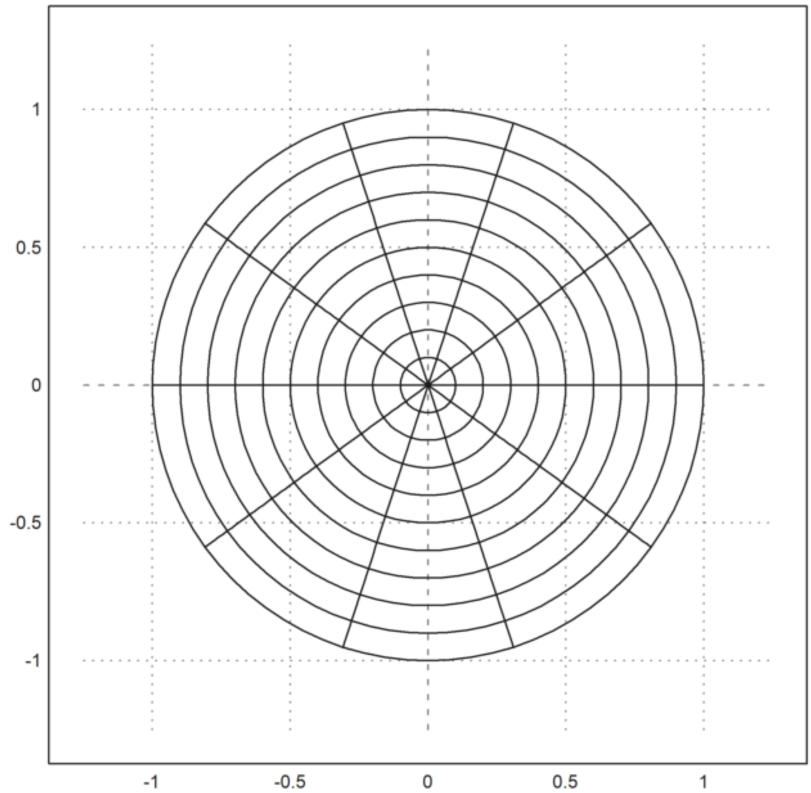
```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
>plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5):
```



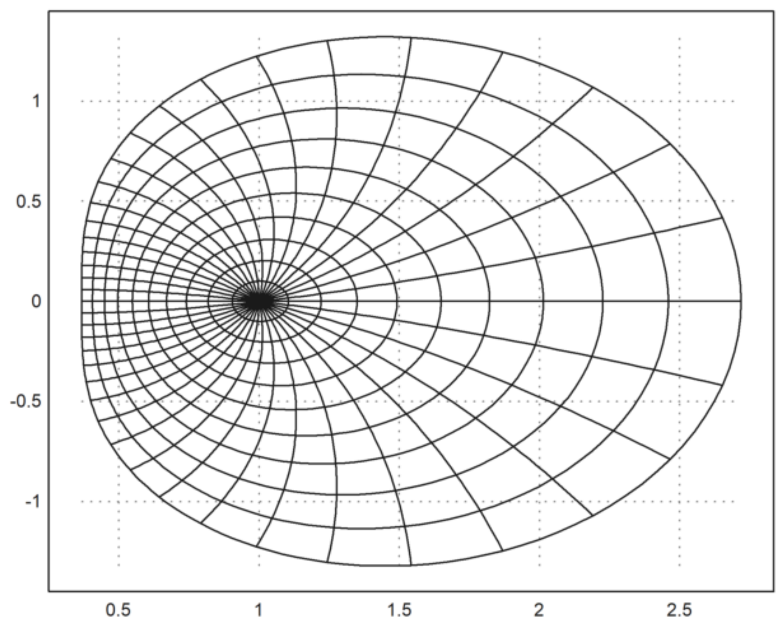
Menggambar Grafik Bilangan Kompleks

Array bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1x2) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.
Dalam contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

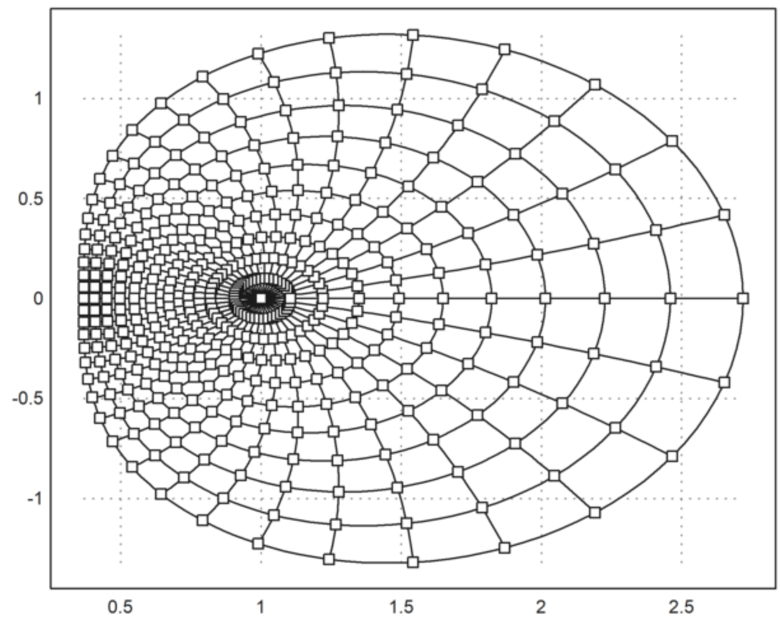
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```




```
>aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=r*exp(I*a);
>plot2d(exp(z),cgrid=[40,10]):
```



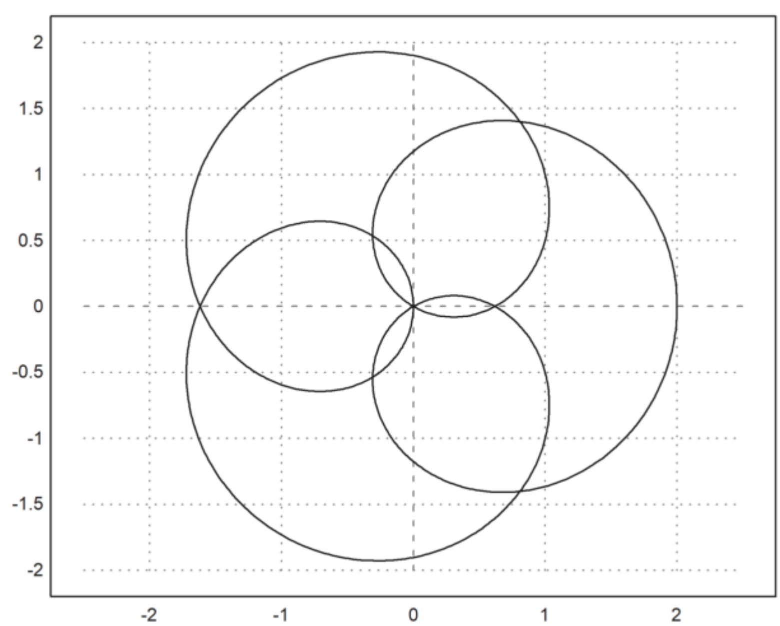
```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```



Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian real dan bagian imajiner.
Dalam contoh, kami memplot lingkaran satuan dengan

$$\gamma(t) = e^{it}$$

```
>t=linspace(0,2pi,1000); ...
>plot2d(exp(I*t)+exp(4*I*t),r=2):
```



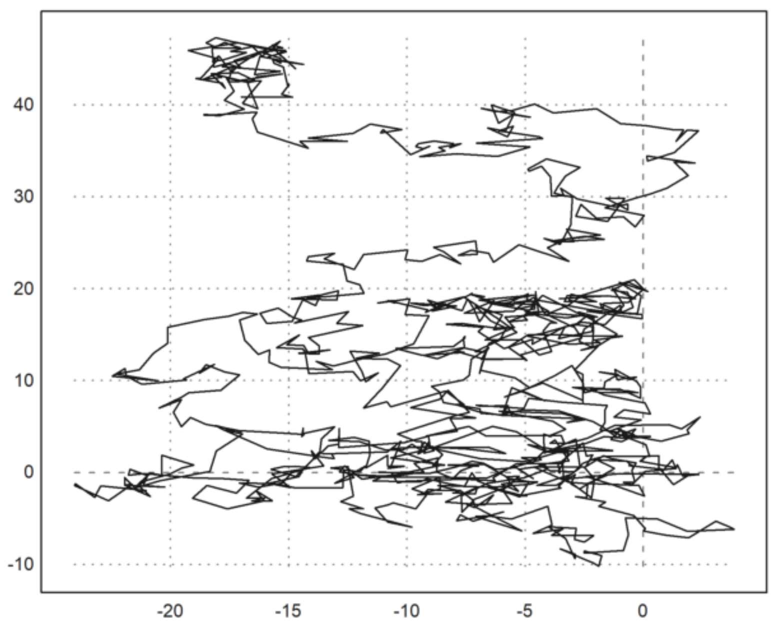
Plot Statistik

Ada banyak fungsi yang dikhususkan pada plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.
Jumlah kumulatif dari nilai terdistribusi 0-1-normal menghasilkan jalan acak.

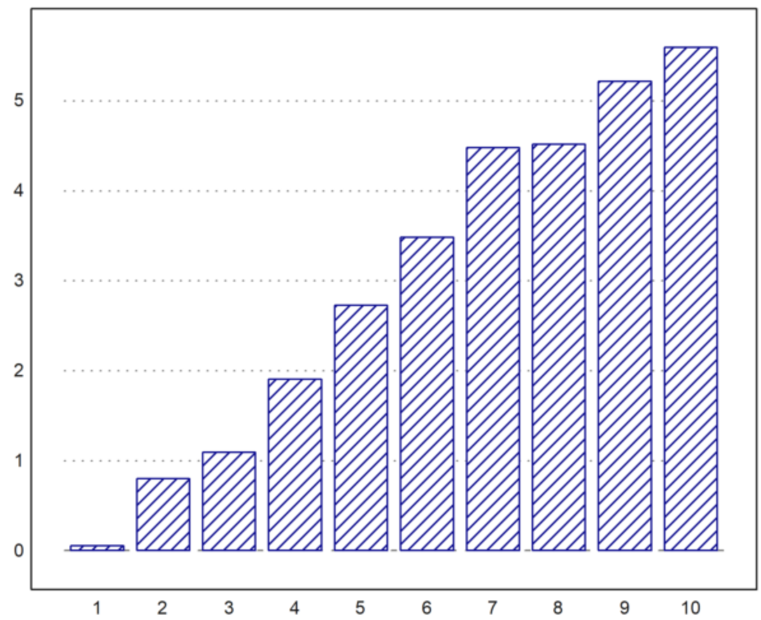
```
>plot2d(cumsum(randnormal(1,1000))):
```

Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

```
>X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
```

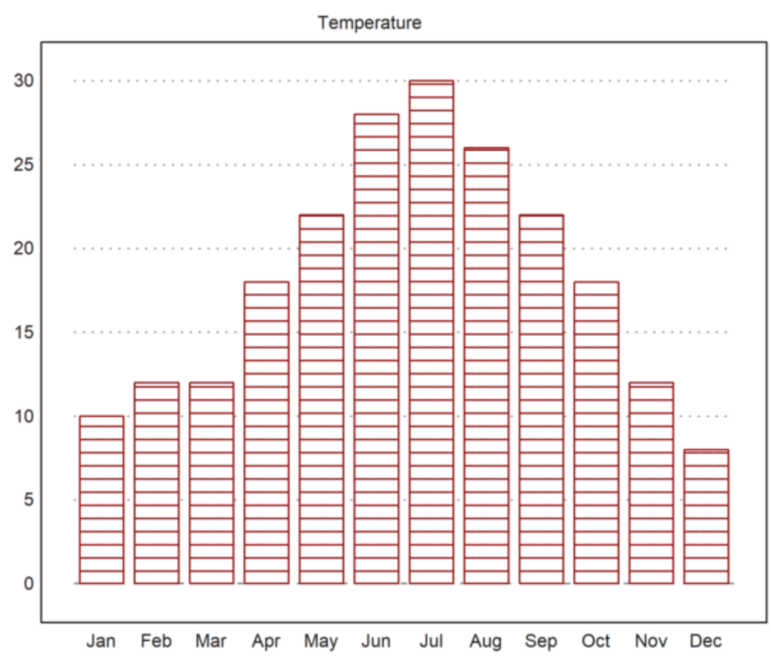


```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

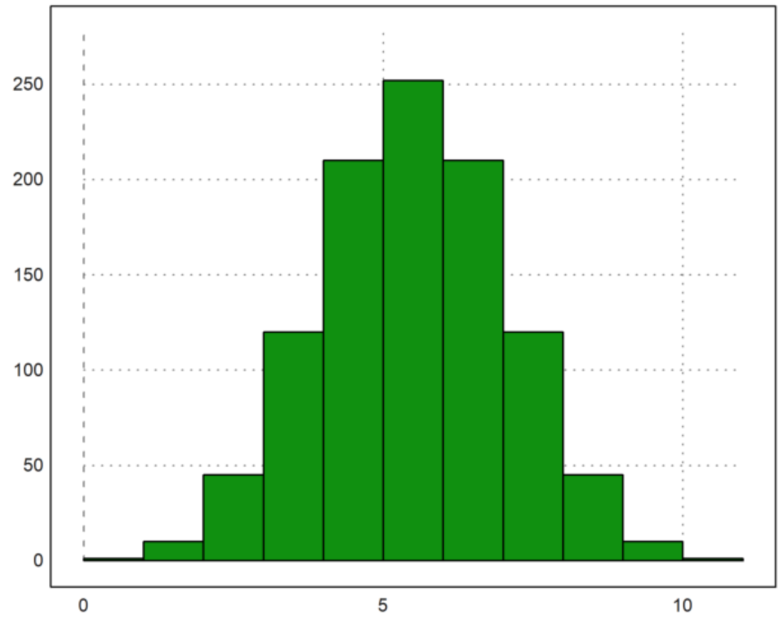


Itu juga dapat menampilkan string sebagai label.

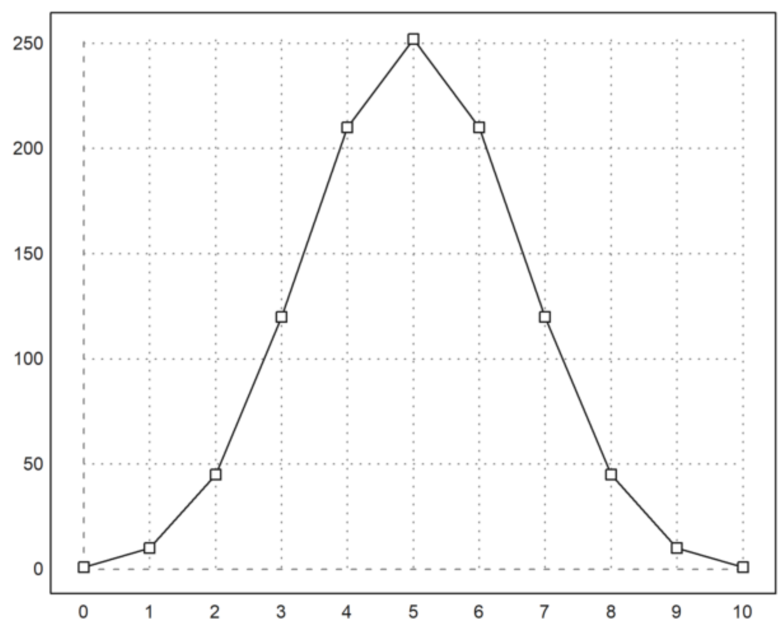
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
>  "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



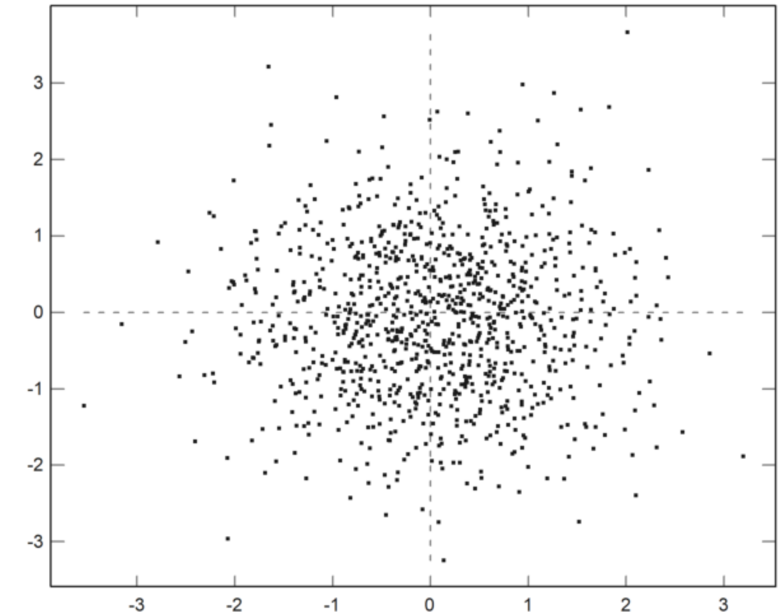
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



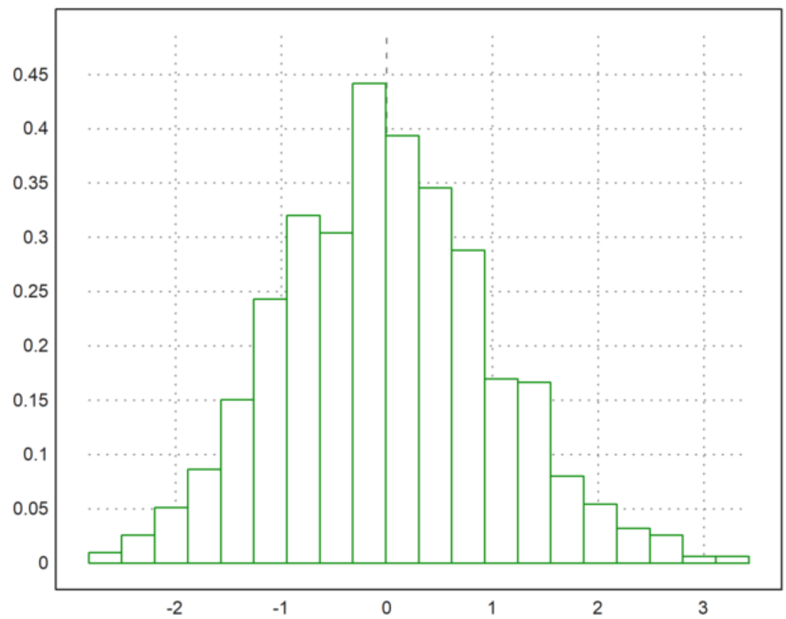

```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



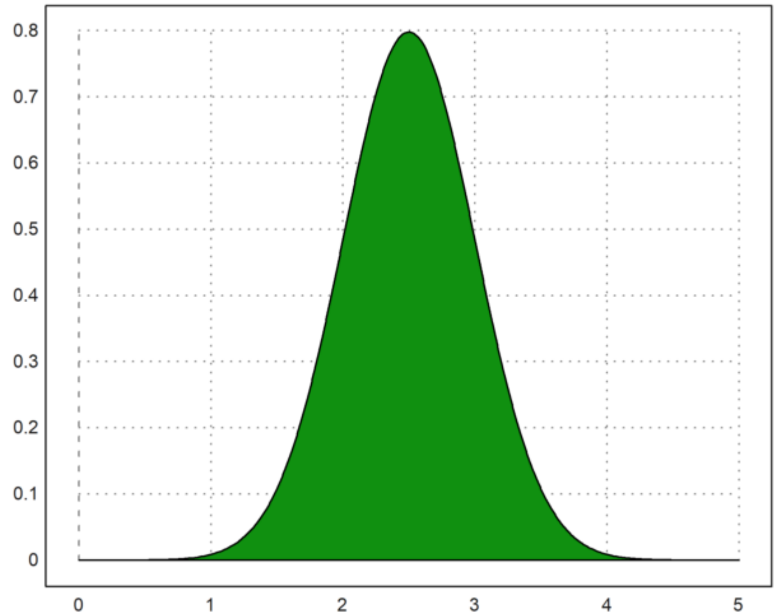

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```




```
>plot2d(normal(1,1000),>distribution,style="O"):
```

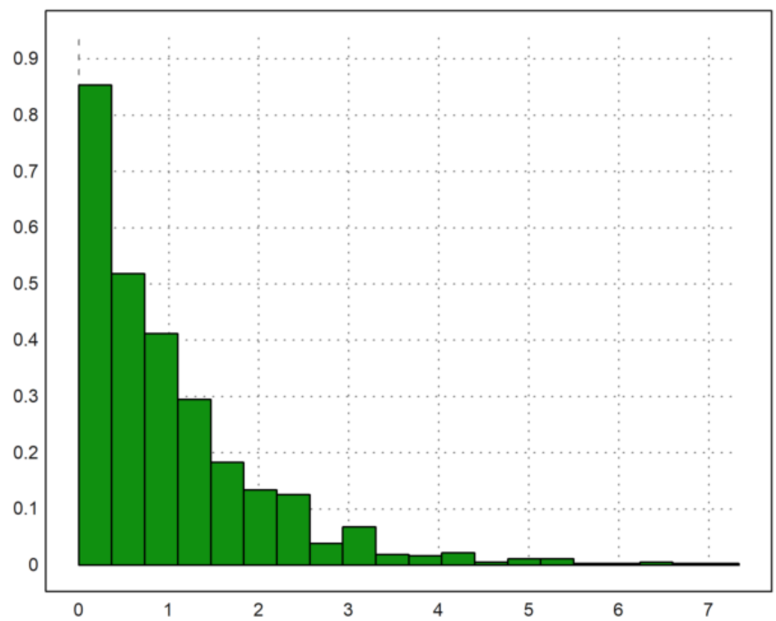



```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



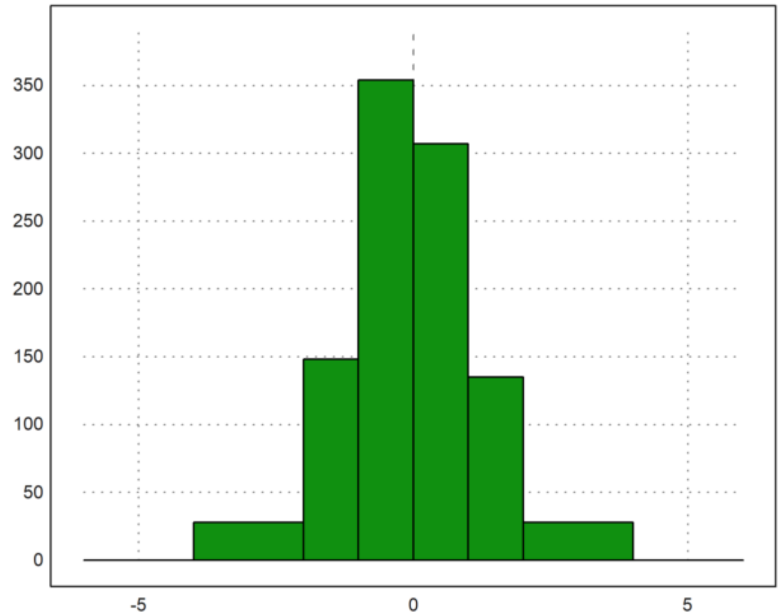
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

```
>w=randexponential(1,1000); // exponential distribution
>plot2d(w,>distribution): // or distribution=n with n intervals
```



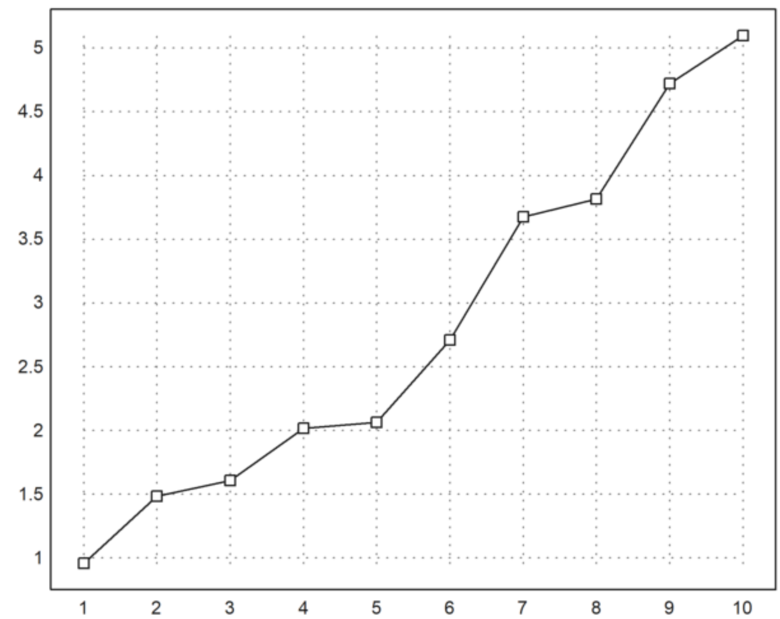
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom.

```
>w=normal(1000); // 0-1-normal distribution
>{x,y}=histo(w,10,v=[-6,-4,-2,-1,0,1,2,4,6]); // interval bounds v
>plot2d(x,y,>bar):
```

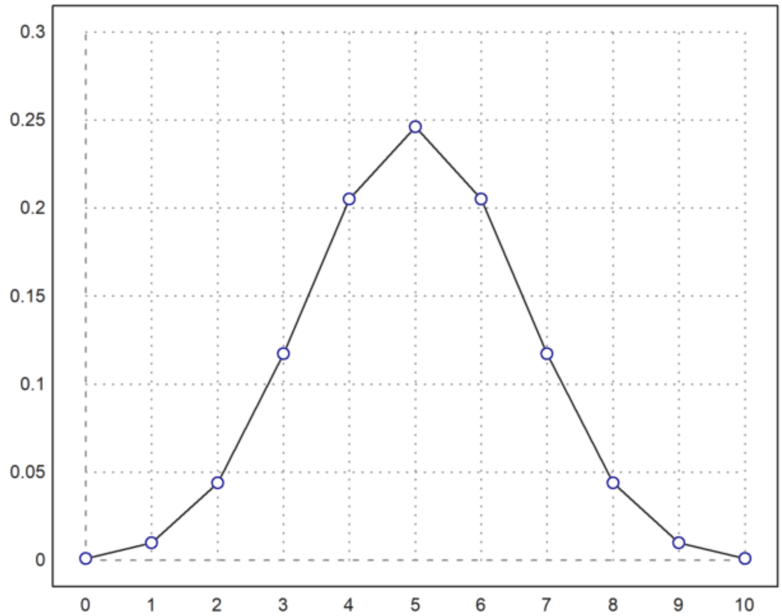


Fungsi statplot() menyetel gaya dengan string sederhana.

```
>statplot(1:10,cumsum(random(10)),"b"):
```

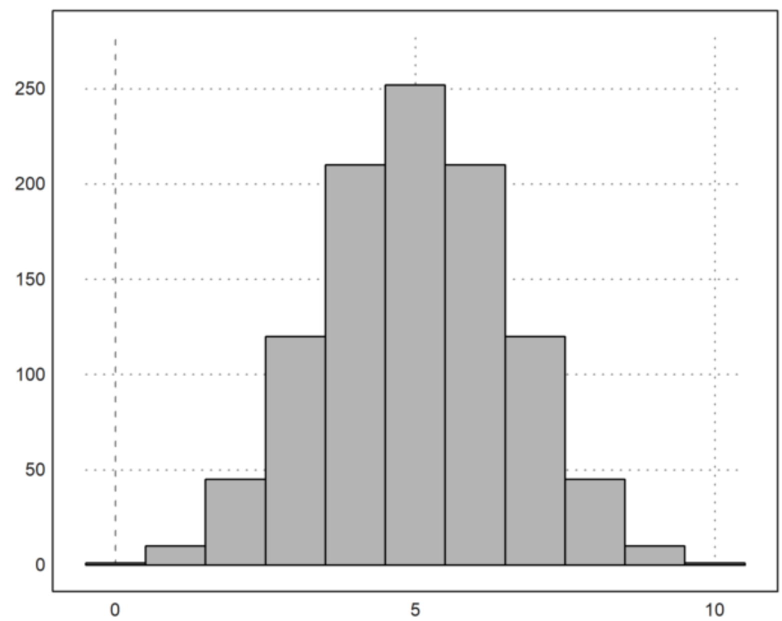



```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



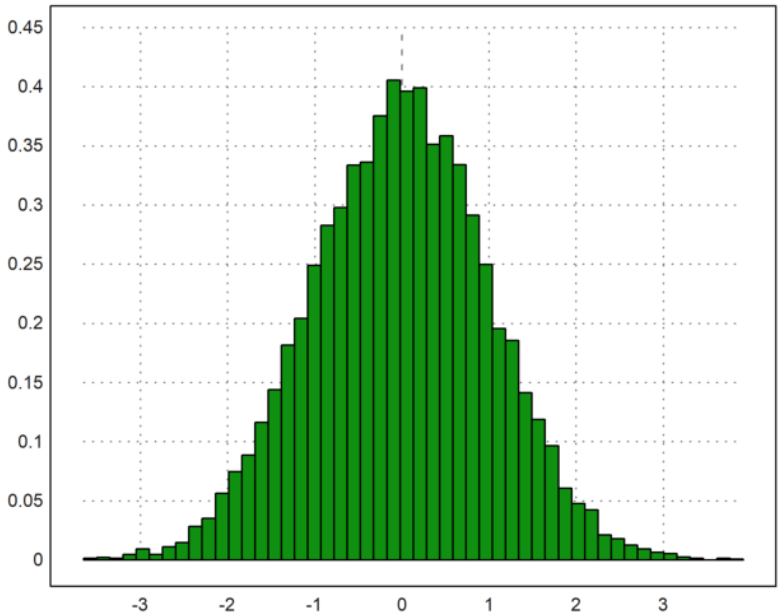
Selain itu, data dapat diplot sebagai batang. Dalam hal ini, x harus diurutkan dan satu elemen lebih panjang dari y. Bilah akan memanjang dari x[i] ke x[i+1] dengan nilai y[i]. Jika x memiliki ukuran yang sama dengan y, maka akan diperpanjang satu elemen dengan spasi terakhir.
Gaya isian dapat digunakan seperti di atas.

```
>n=10; k=bin(n,0:n); ...
>plot2d(-0.5:n+0.5,k,bar=true,fillcolor=lightgray):
```

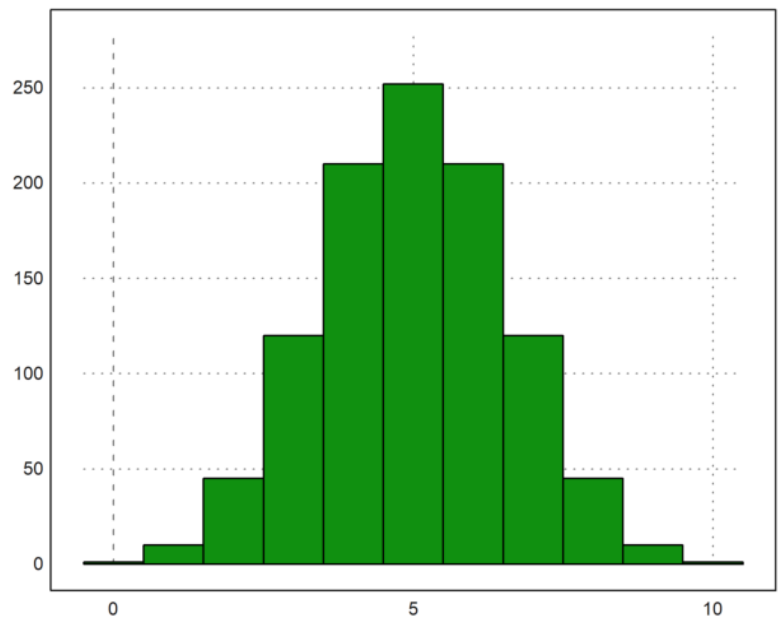


Data untuk plot batang (bar=1) dan histogram (histogram=1) dapat dinyatakan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribusi (atau distribusi=n). Histogram nilai xv akan dihitung secara otomatis dengan >histogram. Jika >genap ditentukan, nilai xv akan dihitung dalam interval bilangan bulat.

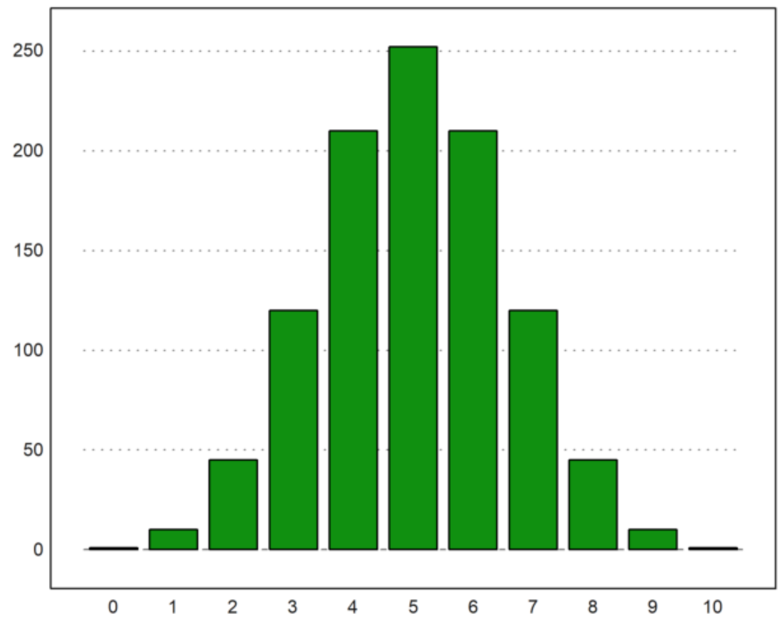
```
>plot2d(normal(10000),distribution=50):
```



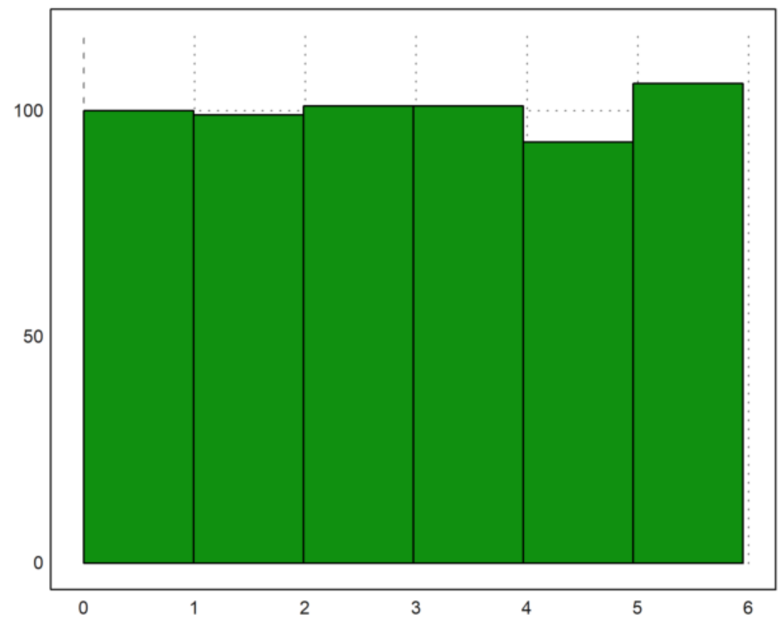

```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```




```
>columnsplot(m,k):
```

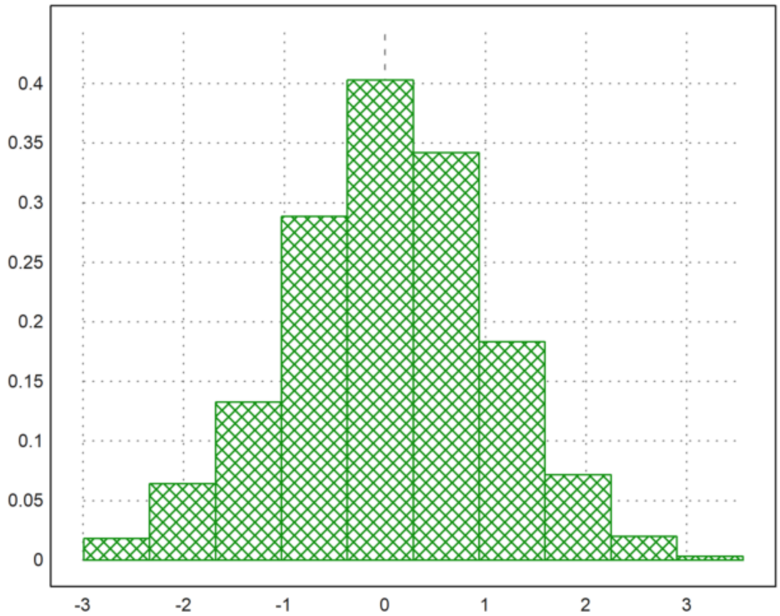



```
>plot2d(random(600)*6,histogram=6):
```



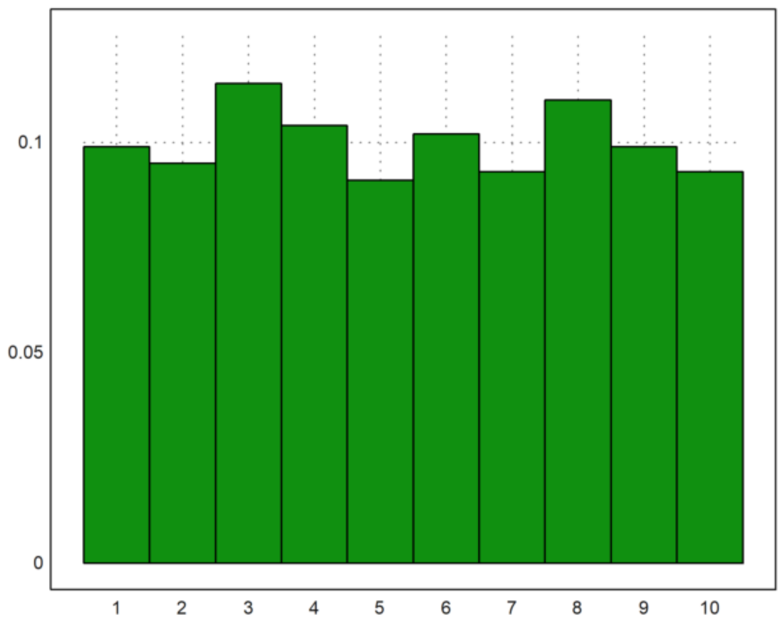
Untuk distribusi, ada parameter distribution=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

```
>plot2d(normal(1,1000),distribution=10,style="\/"):
```



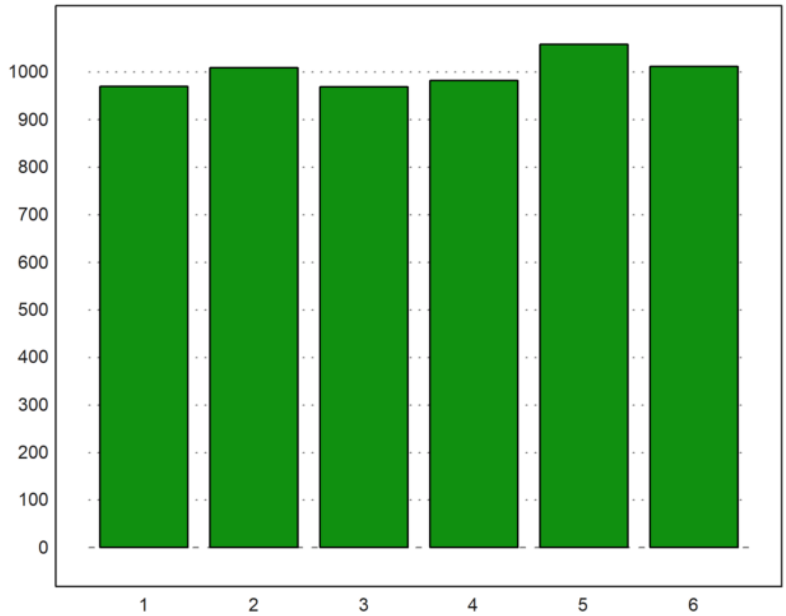
Dengan parameter even=true, ini akan menggunakan interval integer.

```
>plot2d(intrandom(1,1000,10),distribution=10,even=true):
```

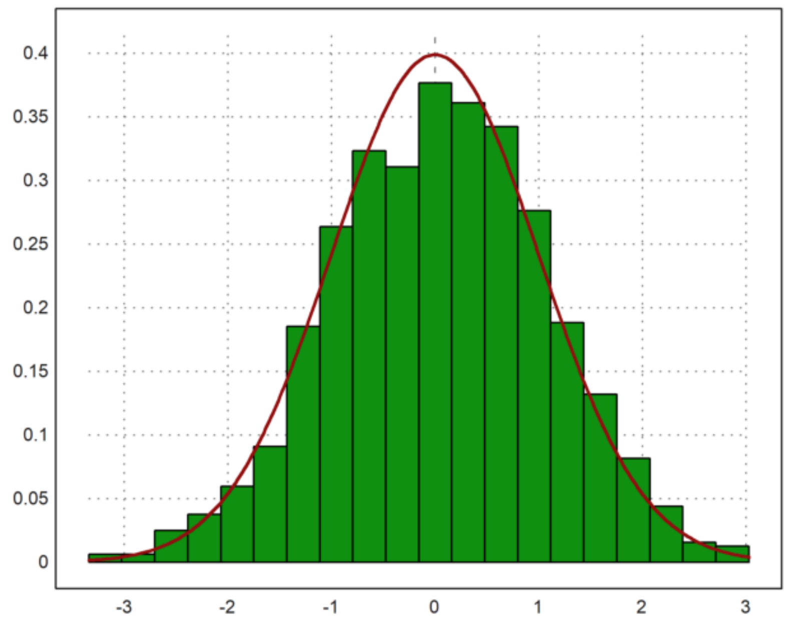


Perhatikan bahwa ada banyak plot statistik, yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

```
>columnsplot(getmultiplicities(1:6,intrandom(1,6000,6))):
```

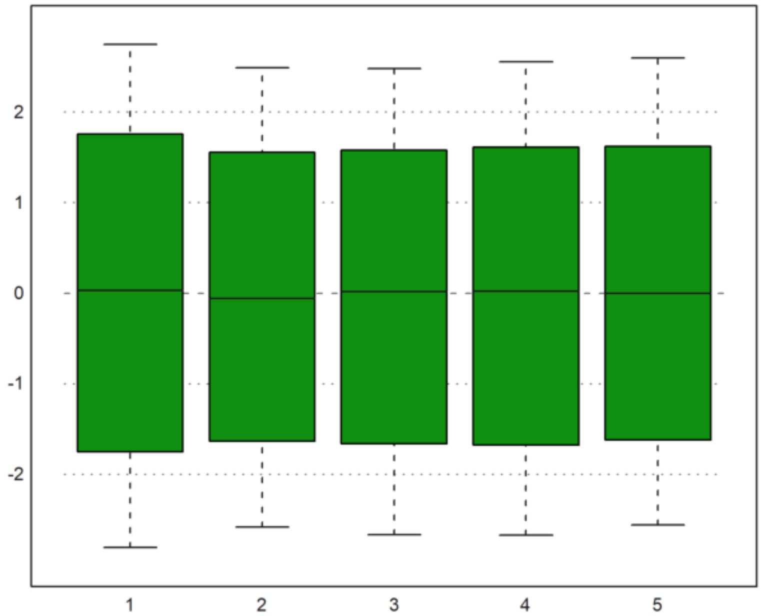



```
>plot2d(normal(1,1000),>distribution); ...
>  plot2d("qnormal(x)",color=red,thickness=2,>add):
```



Ada juga banyak plot khusus untuk statistik. Boxplot menunjukkan kuartil dari distribusi ini dan banyak outlier. Menurut definisi, outlier dalam boxplot adalah data yang melebihi 1,5 kali kisaran 50% tengah plot.

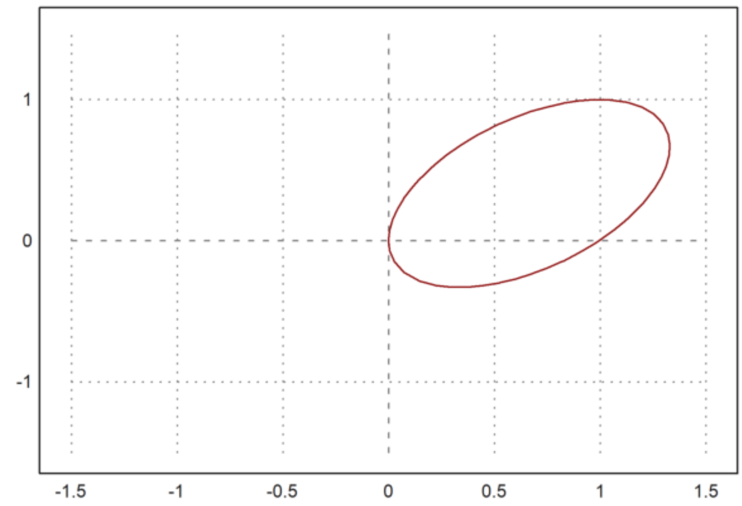
```
>M=normal(5,1000); boxplot(quartiles(M)):
```



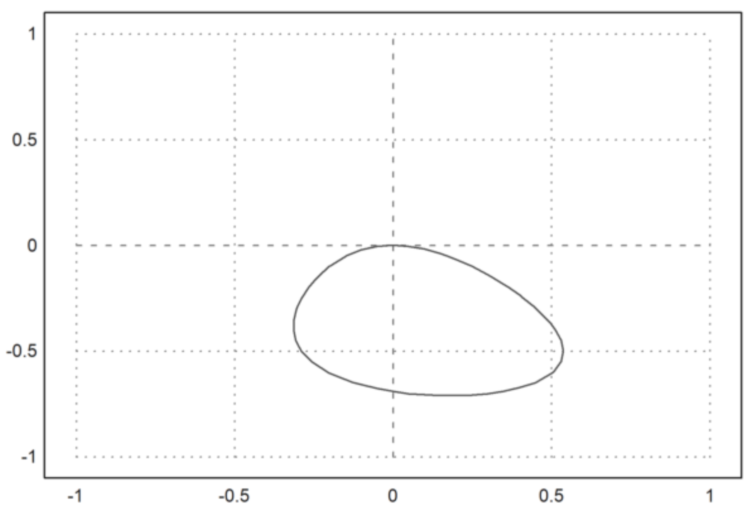
Fungsi Implisit

Plot implisit menunjukkan garis level yang menyelesaikan f(x,y)=level, di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika level="auto", akan ada garis level nc, yang akan menyebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan >hue untuk menunjukkan nilai fungsi. Untuk fungsi implisit, xv harus berupa fungsi atau ekspresi dari parameter x dan y, atau, sebagai alternatif, xv dapat berupa matriks nilai.
Euler dapat menandai garis level
lateks: f(x,y) = c
dari fungsi apapun.
Untuk menggambar himpunan f(x,y)=c untuk satu atau lebih konstanta c, Anda dapat menggunakan plot2d() dengan plot implisitnya di dalam bidang. Parameter untuk c adalah level=c, di mana c dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi untuk setiap titik dalam plot. Parameter "n" menentukan kehalusan plot.

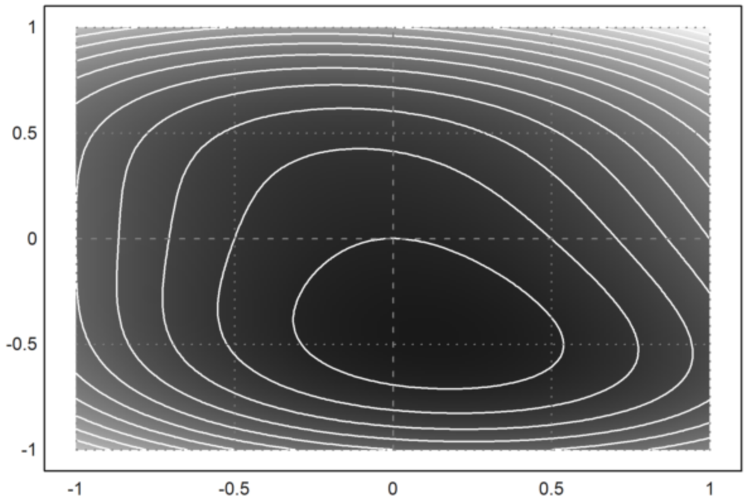
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



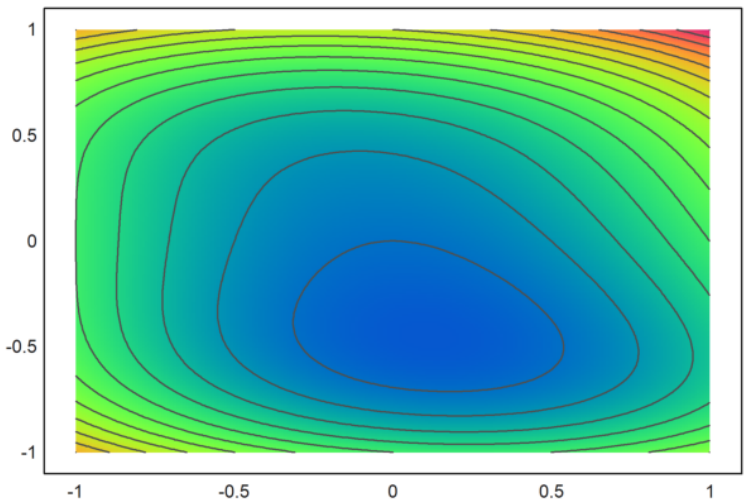
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```



```
>plot2d(expr,level=0:0.5:20,>hue,contourcolor=white,n=200): // nice
```

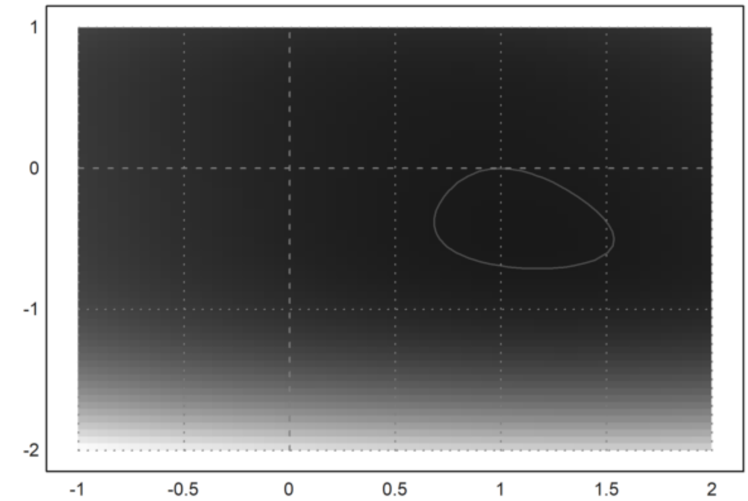



```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

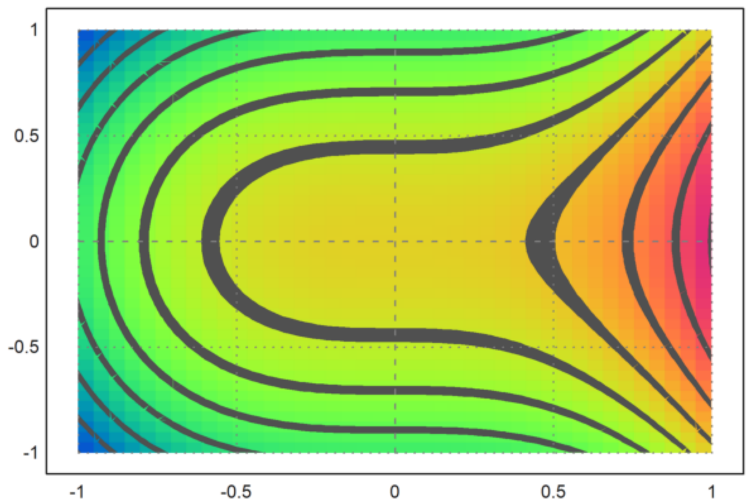


Ini berfungsi untuk plot data juga. Tetapi Anda harus menentukan rentangnya untuk label sumbu.

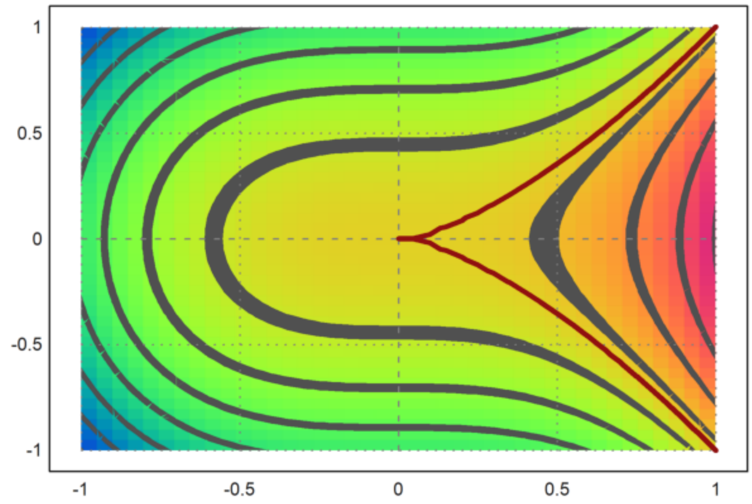
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



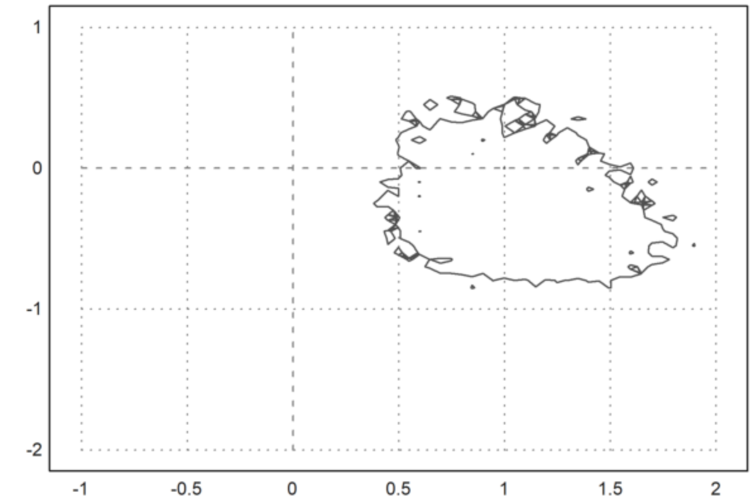

```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



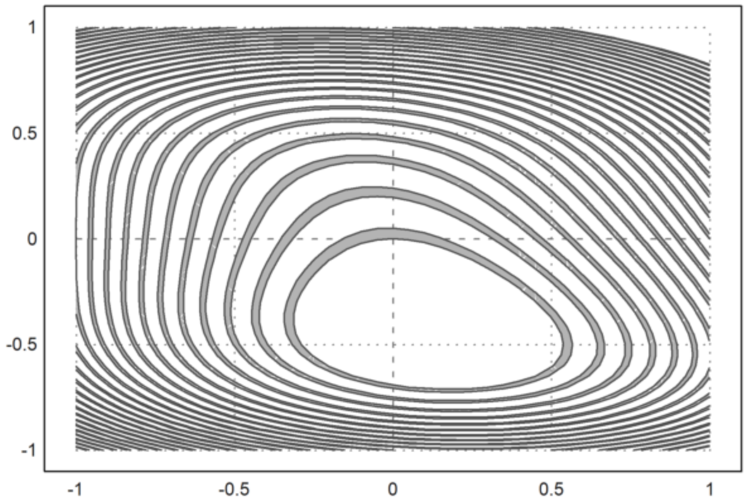

```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



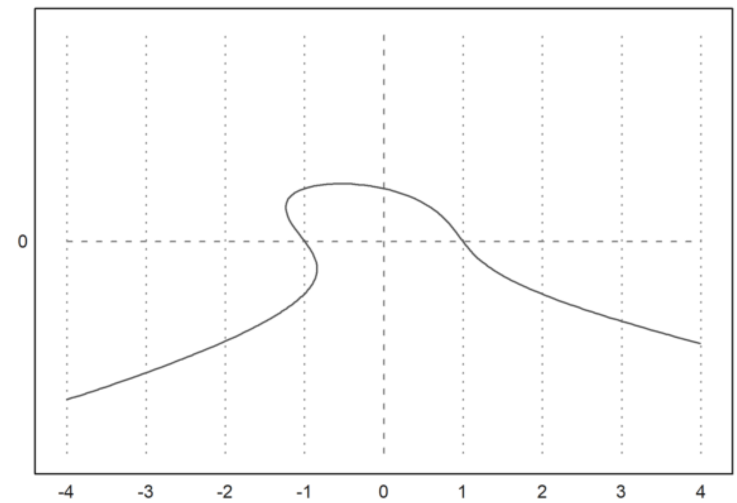

```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



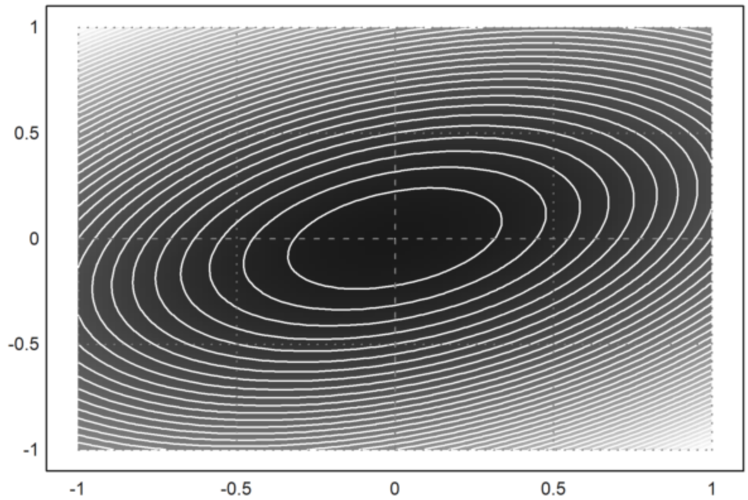

```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```




```
>plot2d("x^2+y^3+x*y",level=1,r=4,n=100):
```




```
>plot2d("x^2+2*y^2-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):
```

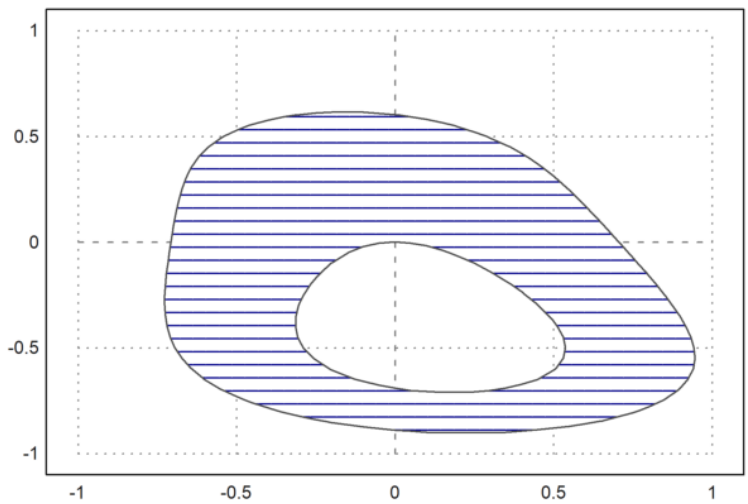


Dimungkinkan juga untuk mengisi set

$$a \leq f(x, y) \leq b$$

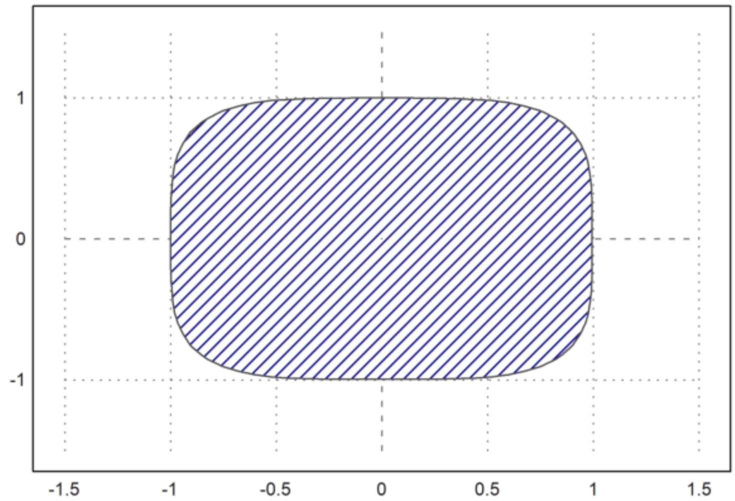
dengan rentang tingkat.
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

```
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```

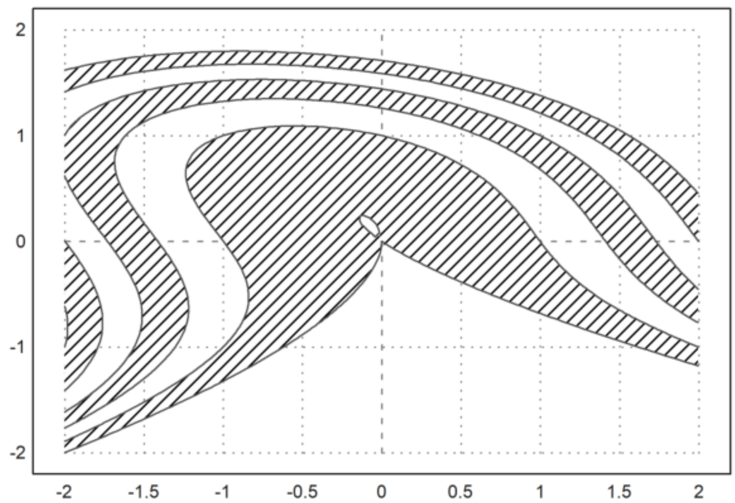


Plot implisit juga dapat menunjukkan rentang level. Kemudian level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Atau, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level ke interval.

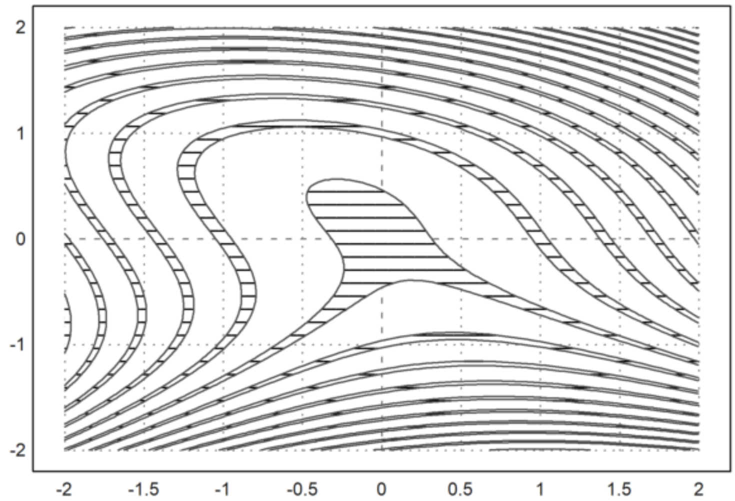
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



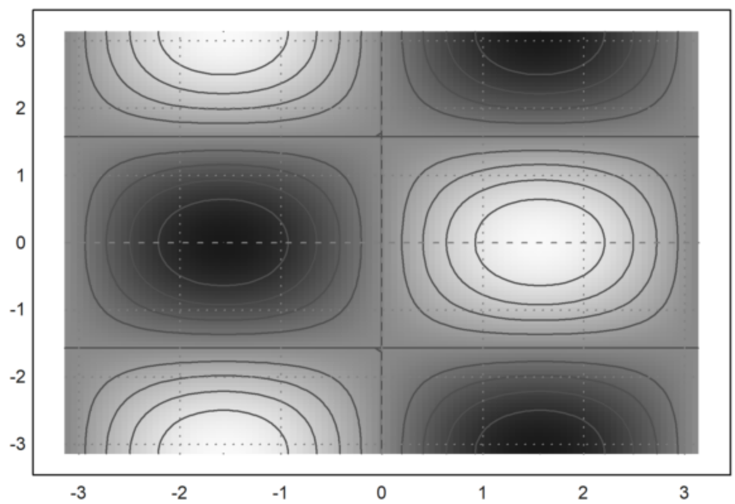

```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```




```
>plot2d("x^2+y^3+x*y",level=-10:20,r=2,style="-",dl=0.1,n=100):
```




```
>plot2d("sin(x)*cos(y)",r=pi,>hue,>levels,n=100):
```

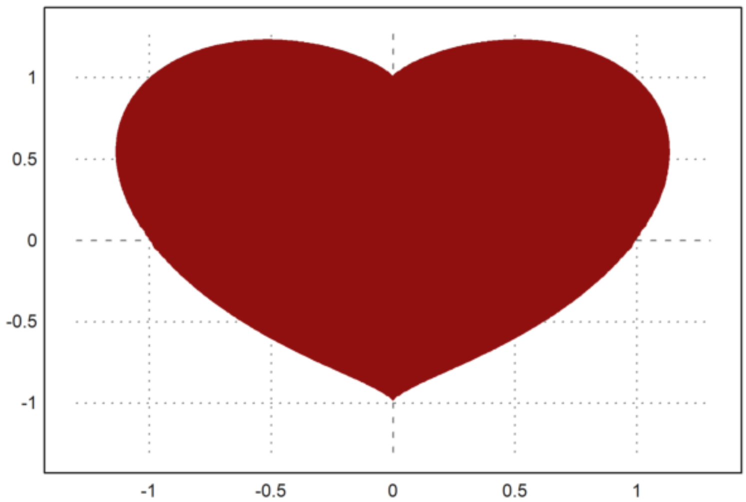


Dimungkinkan juga untuk menandai suatu wilayah

$$a \leq f(x,y) \leq b.$$

Ini dilakukan dengan menambahkan level dengan dua baris.

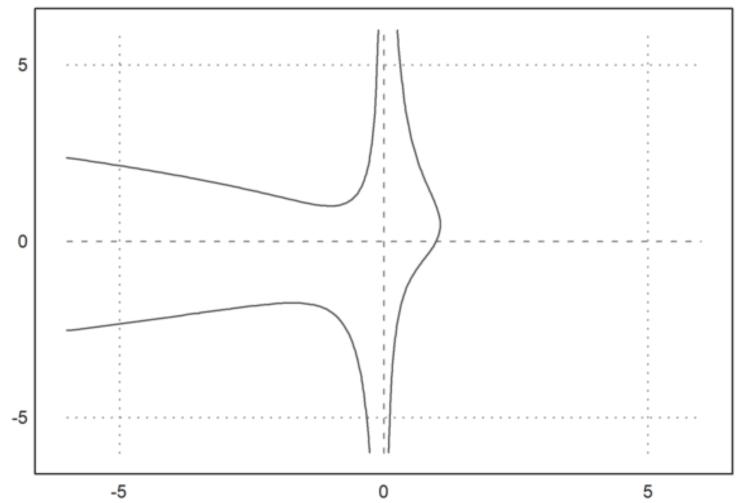
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>   style="#",color=red,<outline, ...
>   level=[-2;0],n=100):
```



Dimungkinkan untuk menentukan level tertentu. Misalnya, kita dapat memplot solusi persamaan seperti

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

```
>plot2d("x^3-x*y+x^2*y^2",r=6,level=1,n=100):
```



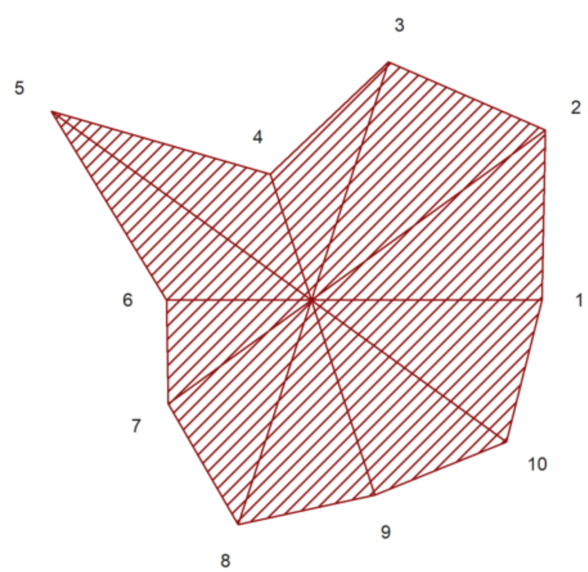
```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

```
    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 kotak atau sumbu kutu di sini. Selain itu, kami menggunakan jendela penuh untuk plot.
Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Ini tidak perlu, jika Anda yakin plot Anda berhasil.

```
>reset; starplot1(normal(1,10)+5,color=red,lab=1:10):
```



Terkadang, Anda mungkin ingin merencanakan sesuatu yang tidak dapat dilakukan plot2d, tetapi hampir.
Dalam fungsi berikut, kami melakukan plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

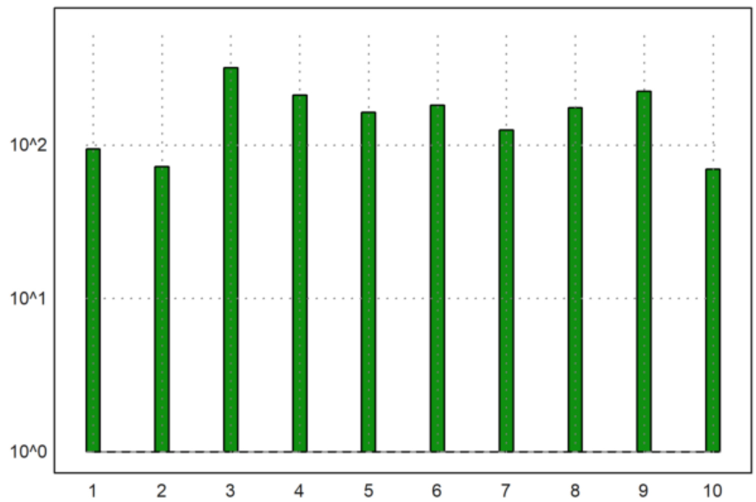
```
>function logimpulseplot1 (x,y) ...
```

```
  {x0,y0}=makeimpulse(x,log(y)/log(10));
  plot2d(x0,y0,>bar,grid=0);
  h=holding(1);
```

```
    frame();
    xgrid(ticks(x));
    p=plot();
    for i=-10 to 10;
      if i<=p[4] and i>=p[3] then
         ygrid(i,yt="10^"+i);
      endif;
    end;
    holding(h);
  endfunction
```

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

```
>aspect(1.5); x=1:10; y=-log(random(size(x)))*200; ...
>logimpulseplot1(x,y):
```



Mari kita menganimasikan kurva 2D menggunakan plot langsung. Perintah plot(x,y) hanya memplot kurva ke jendela plot. setplot(a,b,c,d) mengatur jendela ini.
Fungsi wait(0) memaksa plot untuk muncul di jendela grafik. Jika tidak, menggambar ulang terjadi dalam interval waktu yang jarang.

```
>function animliss (n,m) ...
```

```
  t=linspace(0,2pi,500);
  f=0;
  c=framecolor(0);
  l=linewidth(2);
  setplot(-1,1,-1,1);
  repeat
    clg;
    plot(sin(n*t),cos(m*t+f));
    wait(0);
    if testkey() then break; endif;
    f=f+0.02;
  end;
  framecolor(c);
```

```
linewidth(1);
endfunction
```

Tekan sembarang tombol untuk menghentikan animasi ini.

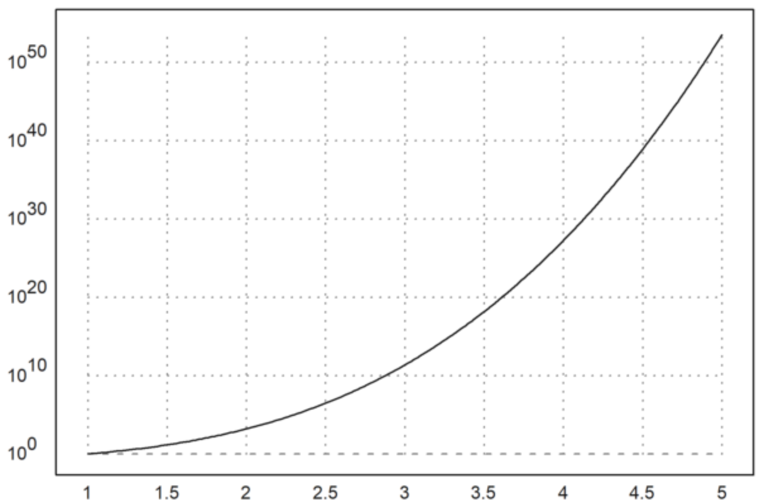
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Plot Logaritmik

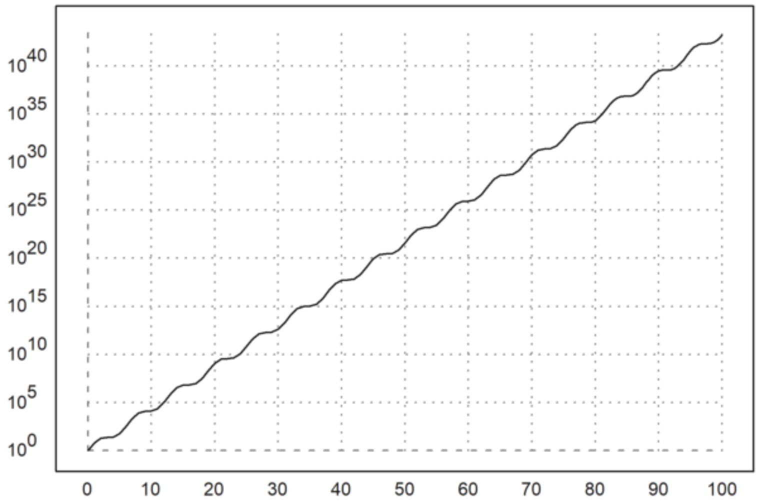
EMT menggunakan parameter "logplot" untuk skala logaritmik.
Plot logaritma dapat diplot baik menggunakan skala logaritma dalam y dengan logplot=1, atau menggunakan skala logaritma dalam x dan y dengan logplot=2, atau dalam x dengan logplot=3.

```
- logplot=1: y-logaritma
- logplot=2: x-y-logaritma
- logplot=3: x-logaritma
```

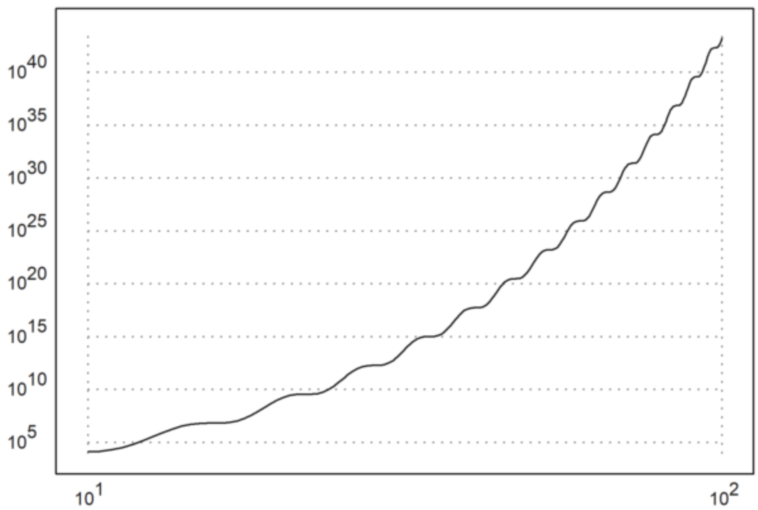
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



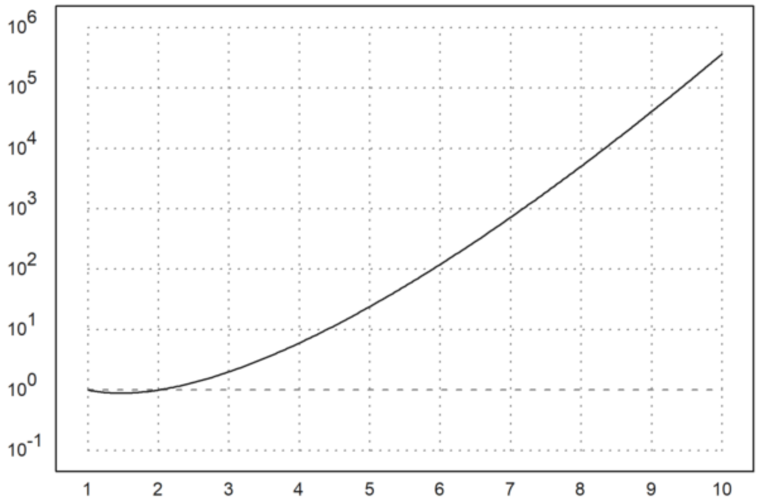

```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



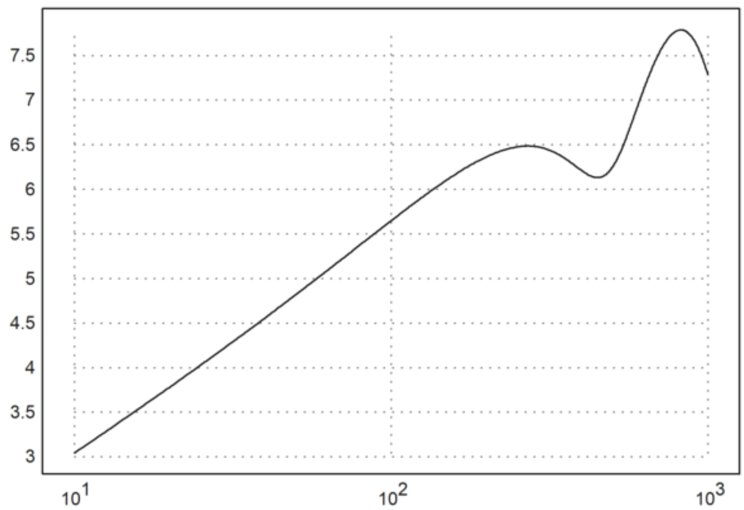

```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```




```
>plot2d("gamma(x)",1,10,logplot=1):
```

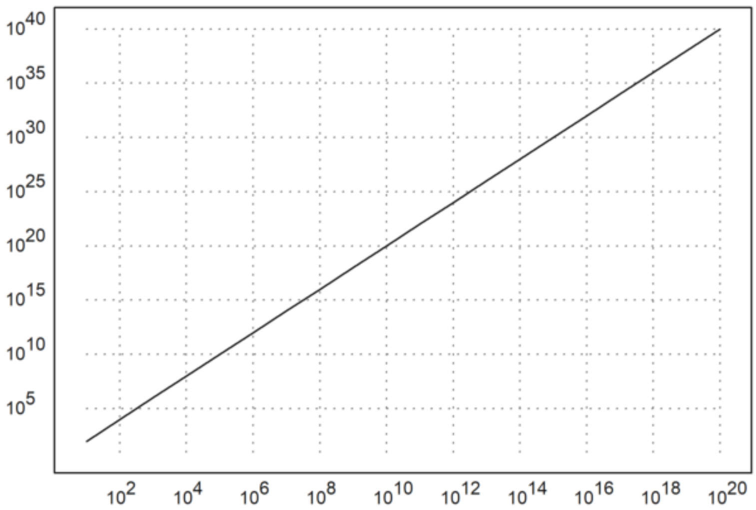



```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Ini juga berfungsi dengan plot data.

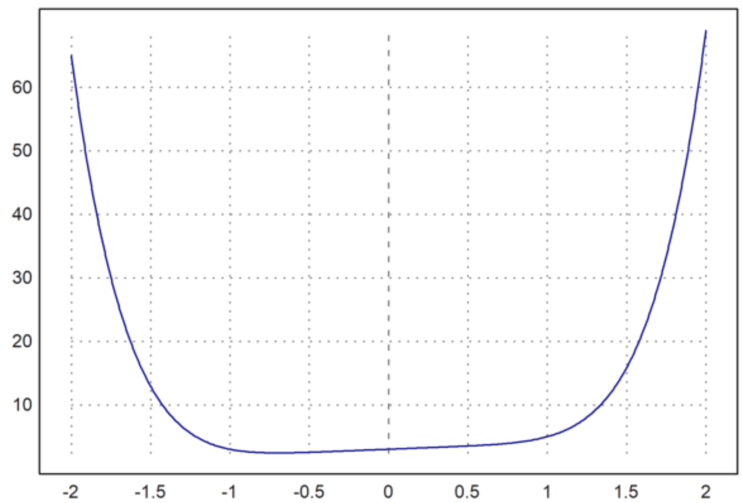
```
>x=10^(1:20); y=x^2-x;
>plot2d(x,y,logplot=2):
```



Latihan Soal

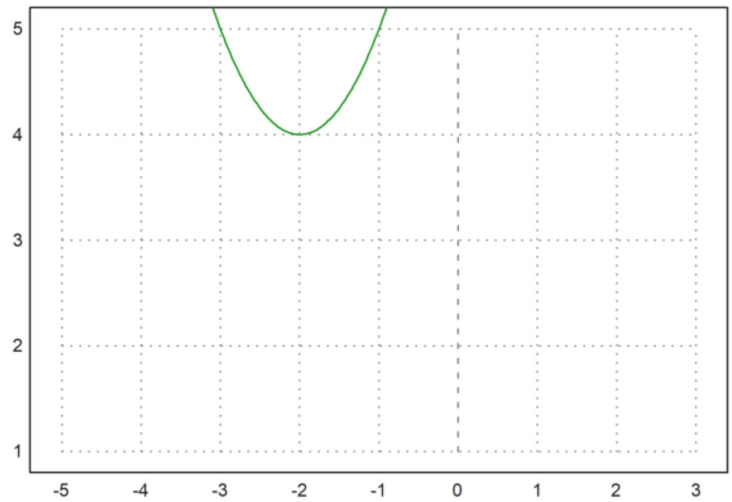
1. Gambarkan kurva fungsi f(x) = x^6 + x + 3!

```
>plot2d("x^6+x+3", color=blue):
```



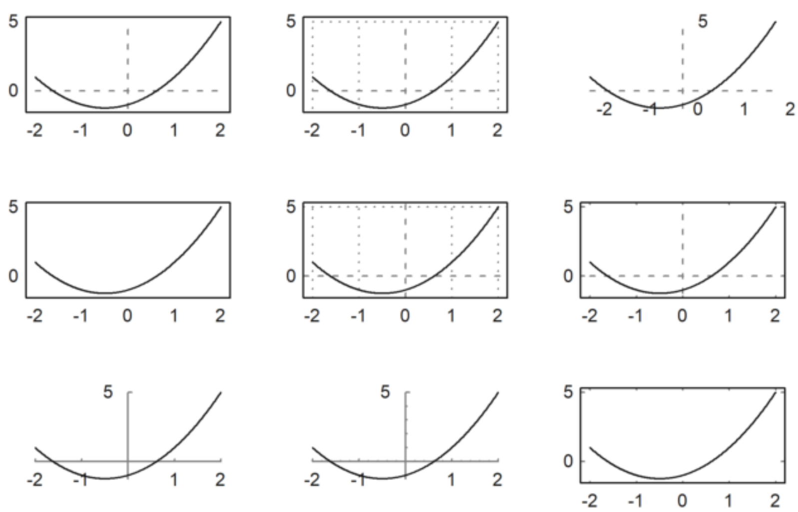
2. Gambarkan grafik f(x) = x^2 + 4x + 8 dan tentukan titik puncaknya!

```
>plot2d("x^2+4x+8",-5,3,1,5, color=green):
```



Berdasarkan grafik di atas, maka titik puncak dari fungsi tersebut berada pada titik (-2, 3).
3. Gambarkan f(x) = x^2 + x - 1 dalam beberapa model grafik!

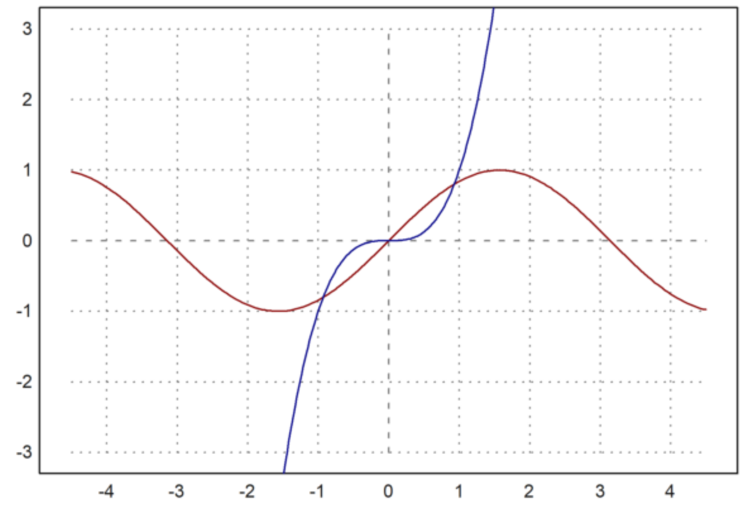
```
>figure(3,3);...
>for k=1:9; figure(k); plot2d("x^2+x-1",-2,2,grid=k); end;...
>figure(0):
```



4. Gambarkan kedua grafik berikut secara bersamaan!

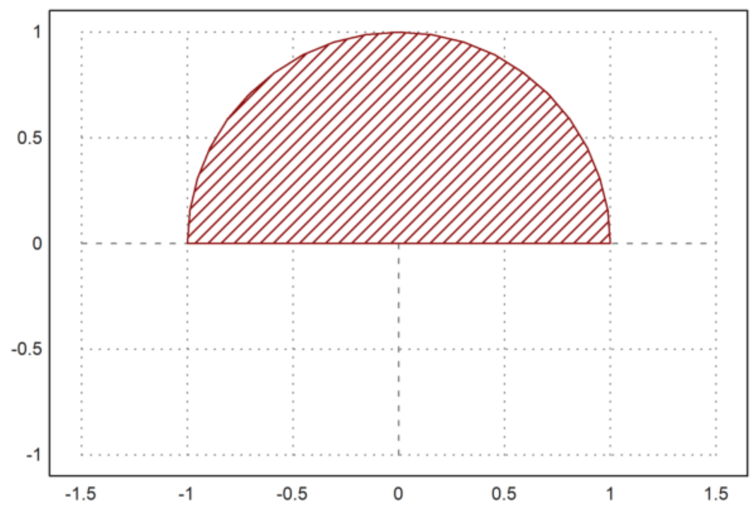
```
f(x) = sin(x)
f(x) = x^3
```

```
>plot2d("sin(x)",r=3, color=red); plot2d("x^3",color=blue,>add):
```



5. Gambarkan setengah lingkaran berwarna merah!

```
>t=linspace(0,pi,20); ...
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1):
```



Rujukan Lengkap Fungsi plot2d()

```
function plot2d (xv, yv, btest, a, b, c, d, xmin, xmax, r, n,  ..
logplot, grid, frame, framecolor, square, color, thickness, style, ..
auto, add, user, delta, points, addpoints, pointstyle, bar, histogram,  ..
distribution, even, steps, own, adaptive, hue, level, contour,  ..
nc, filled, fillcolor, outline, title, xl, yl, maps, contourcolor, ..
contourwidth, ticks, margin, clipping, cx, cy, insimg, spectral,  ..
cgrid, vertical, smaller, dl, niveau, levels)
```

Multipurpose plot function for plots in the plane (2D plots). This function can do plots of functions of one variables, data plots, curves in the plane, bar plots, grids of complex numbers, and implicit plots of functions of two variables.
Parameters
x,y : equations, functions or data vectors
a,b,c,d : Plot area (default a=-2,b=2)
r : if r is set, then a=cx-r, b=cx+r, c=cy-r, d=cy+r

```
r can be a vector [rx,ry] or a vector [rx1,rx2,ry1,ry2].
```

xmin,xmax : range of the parameter for curves
auto : Determine y-range automatically (default)
square : if true, try to keep square x-y-ranges
n : number of intervals (default is adaptive)
grid : 0 = no grid and labels,

```
1 = axis only,
2 = normal grid (see below for the number of grid lines)
3 = inside axis
4 = no grid
5 = full grid including margin
6 = ticks at the frame
7 = axis only
8 = axis only, sub-ticks
```

frame : 0 = no frame
framecolor: color of the frame and the grid
margin : number between 0 and 0.4 for the margin around the plot
color : Color of curves. If this is a vector of colors,

```
it will be used for each row of a matrix of plots. In the case of
point plots, it should be a column vector. If a row vector or a
full matrix of colors is used for point plots, it will be used for
each data point.
```

thickness : line thickness for curves

```
This value can be smaller than 1 for very thin lines.
```

style : Plot style for lines, markers, and fills.

```
For points use
"[]", "<>", ".", "..", "...",
"*", "+", "|", "-", "o"
"[]#", "<>#", "o#" (filled shapes)
"[]w", "<>w", "ow" (non-transparent)
For lines use
"-", "--", "-.", ".", ".-.", "-.-", "->"
For filled polygons or bar plots use
"#", "#O", "O", "/", "\", "\/",
"+", "|", "-", "t"
```

points : plot single points instead of line segments
addpoints : if true, plots line segments and points
add : add the plot to the existing plot
user : enable user interaction for functions
delta : step size for user interaction
bar : bar plot (x are the interval bounds, y the interval values)
histogram : plots the frequencies of x in n subintervals
distribution=n : plots the distribution of x with n subintervals
even : use inter values for automatic histograms.
steps : plots the function as a step function (steps=1,2)
adaptive : use adaptive plots (n is the minimal number of steps)
level : plot level lines of an implicit function of two variables
outline : draws boundary of level ranges.
If the level value is a 2xn matrix, ranges of levels will be drawn
in the color using the given fill style. If outline is true, it
will be drawn in the contour color. Using this feature, regions of
f(x,y) between limits can be marked.
hue : add hue color to the level plot to indicate the function

```
value
```

contour : Use level plot with automatic levels
nc : number of automatic level lines
title : plot title (default "")
xl, yl : labels for the x- and y-axis
smaller : if >0, there will be more space to the left for labels.
vertical :

```
Turns vertical labels on or off. This changes the global variable
verticallabels locally for one plot. The value 1 sets only vertical
text, the value 2 uses vertical numerical labels on the y axis.
```

filled : fill the plot of a curve
fillcolor : fill color for bar and filled curves
outline : boundary for filled polygons
logplot : set logarithmic plots

```
1 = logplot in y,
2 = logplot in xy,
3 = logplot in x
```

own :

```
A string, which points to an own plot routine. With >user, you get
the same user interaction as in plot2d. The range will be set
before each call to your function.
```

maps : map expressions (0 is faster), functions are always mapped.
contourcolor : color of contour lines
contourwidth : width of contour lines
clipping : toggles the clipping (default is true)
title :

```
This can be used to describe the plot. The title will appear above
the plot. Moreover, a label for the x and y axis can be added with
xl="string" or yl="string". Other labels can be added with the
functions label() or labelbox(). The title can be a unicode
string or an image of a Latex formula.
```

cgrid :

```
Determines the number of grid lines for plots of complex grids.
Should be a divisor of the the matrix size minus 1 (number of
subintervals). cgrid can be a vector [cx,cy].
```

Overview

The function can plot

- expressions, call collections or functions of one variable,
- parametric curves,
- x data against y data,
- implicit functions,
- bar plots,
- complex grids,
- polygons.

If a function or expression for xv is given, plot2d() will compute values in the given range using the function or expression. The expression must be an expression in the variable x. The range must be defined in the parameters a and b unless the default range should be used. The y-range will be computed automatically, unless c and d are specified, or a radius r, which yields the range r,r for x and y. For plots of functions, plot2d will use an adaptive evaluation of the function by default. To speed up the plot for complicated functions, switch this off with <adaptive, and optionally decrease the number of intervals n. Moreover, plot2d() will by default use mapping. I.e., it will compute the plot element for element. If your expression or your functions can handle a vector x, you can switch that off with <maps for faster evaluation. Note that adaptive plots are always computed element for element. If functions or expressions for both xv and for yv are specified, plot2d() will compute a curve with the xv values as x-coordinates and the yv values as y-coordinates. In this case, a range should be defined for the parameter using xmin, xmax. Expressions contained in strings must always be expressions in the parameter variable x.

BAB 3

MENGGUNAKAN EMT UNTUK MENGGAMBAR GRAFIK 3 DIMENSI (3D)

[a4paper,10pt]article eumat

Nama : Anisah Daffa Citra Nareswari
Kelas : Matematika E 2022
NIM : 22305141044

Menggambar Plot 3D dengan EMT adalah pengenalan plot 3D di EMT

Kami membutuhkan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

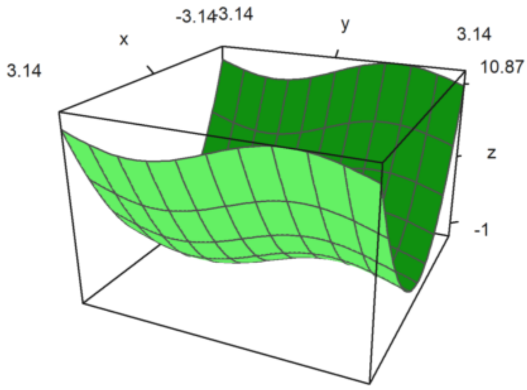
Euler menggambar fungsi seperti itu menggunakan algoritma pengurutan untuk menyembunyikan bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Defaultnya adalah dari kuadran x-y positif ke arah asal x = y = z = 0, tetapi sudut = 0 ? melihat dari arah sumbu y. Sudut pandang dan ketinggian dapat diubah.

Euler bisa merencanakan

- permukaan dengan bayangan dan garis level atau rentang level,
- awan poin,
- kurva parametrik,
- permukaan implisit.

Plot 3D dari suatu fungsi menggunakan plot3d. Cara termudah adalah dengan memplot ekspresi dalam x dan y. Parameter r mengatur kisaran plot sekitar (0,0).

```
>aspect(1.5); plot3d("x^2+sin(y)",r=pi):
```



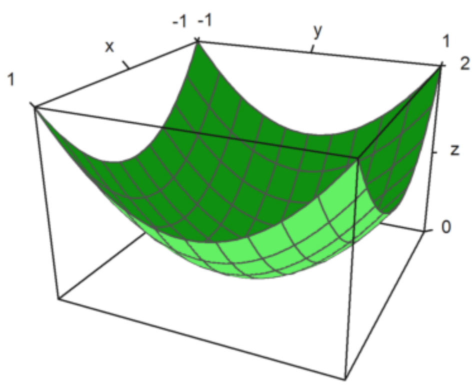
Fungsi Dua Variabel

Untuk grafik suatu fungsi, gunakan
- ekspresi sederhana dalam x dan y,
- nama fungsi dari dua variabel l
- atau matriks data.

Defaultnya adalah kisi kawat yang diisi dengan warna berbeda di kedua sisi. Perhatikan bahwa jumlah default interval kisi adalah 10, tetapi plot menggunakan jumlah default persegi panjang 40x40 untuk membuat permukaan. Ini bisa diubah.
- n = 40, n = [40,40]: jumlah garis grid di setiap arah
- grid= 10, grid = [10,10]: jumlah garis kisi di setiap arah.

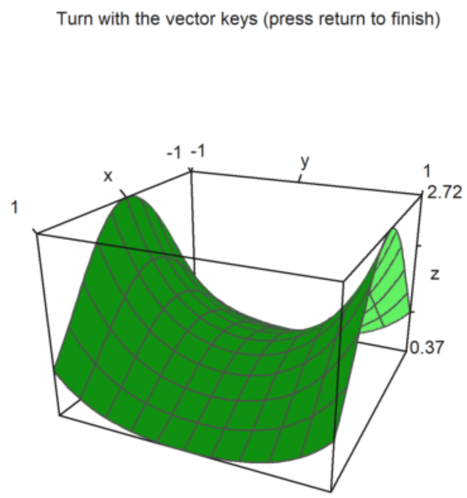
Kami menggunakan default n = 40 dan grid = 10.

```
>plot3d("x^2+y^2"):
```



Interaksi pengguna dimungkinkan dengan> user parameter. Pengguna dapat menekan tombol berikut.
- left,right,up,down: putar sudut pandang
- +, -: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l: beralih memutar sumber cahaya (lihat di bawah)
- space: reset ke default
- return: interaksi akhir

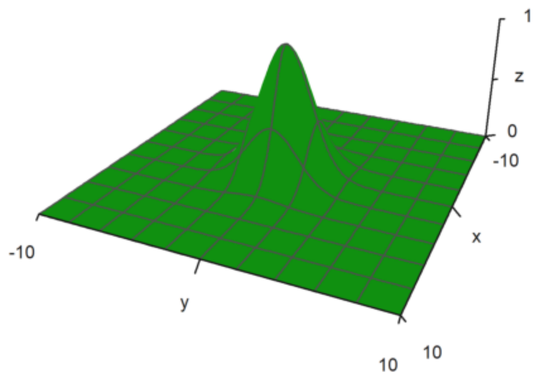
```
>plot3d("exp(-x^2+y^2)",>user, ...
>  title="Turn with the vector keys (press return to finish)"):
```



Rentang plot untuk fungsi dapat ditentukan dengan
- a, b: rentang-x
- c, d: rentang y
- r: persegi simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.
fscale: menskalakan ke nilai fungsi (defaultnya adalah <fscale).
scale: angka atau vektor 1x2 untuk skala ke arah x dan y.
frame: jenis bingkai (default 1)

```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3):
```



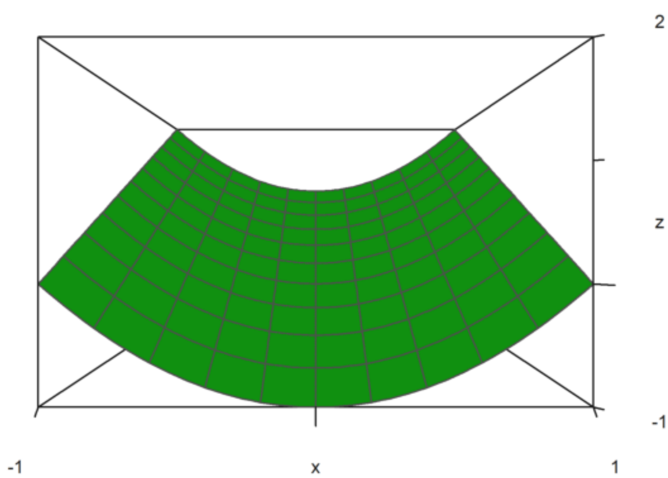
Tampilan dapat diubah dengan berbagai cara.
- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut ke sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.
Nilai default bisa diperiksa atau diubah dengan fungsi view (). Ini mengembalikan parameter dalam urutan di atas.

```
>view
```

```
[5,  2.6,  2,  0.4]
```

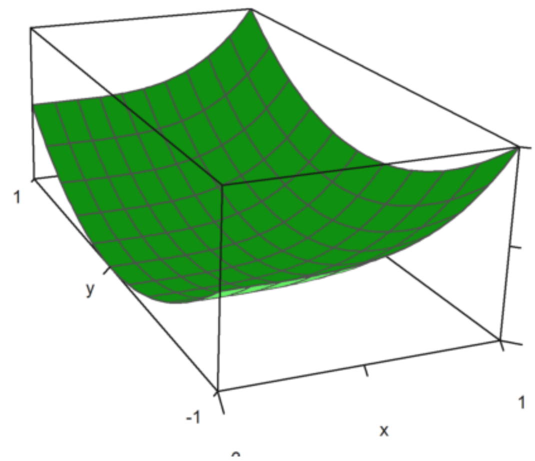
Jarak yang lebih dekat membutuhkan lebih sedikit zoom. Efeknya lebih seperti lensa sudut lebar.
Dalam contoh berikut, sudut = 0 dan tinggi = 0 dilihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^2+y",distance=3,zoom=2,angle=0,height=0):
```



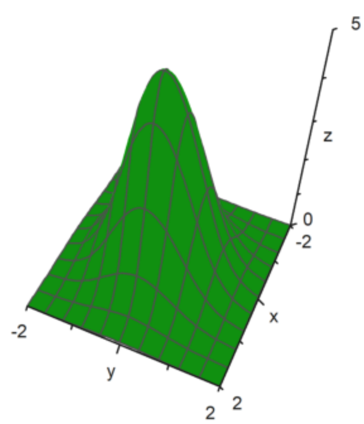
Plot selalu terlihat ke tengah plot kubus. Anda dapat memindahkan pusat dengan parameter tengah.

```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
>  center=[0.4,0,0],zoom=5):
```



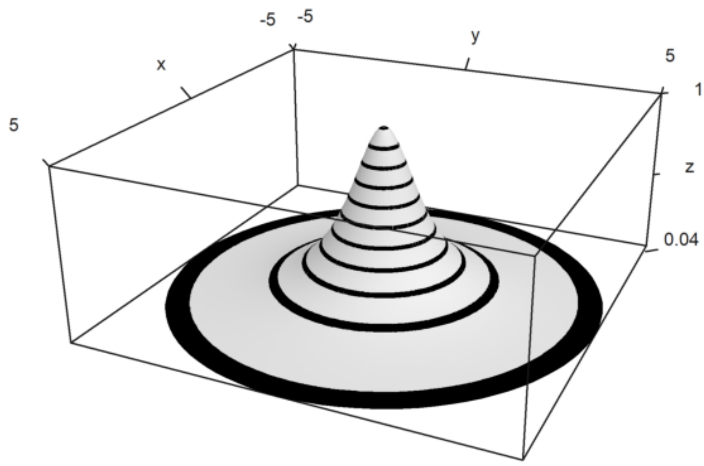
Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung ukuran plot. Namun, label mengacu pada ukuran sebenarnya.
Jika Anda mematikannya dengan scale = false, Anda harus berhati-hati, bahwa plot masih pas dengan jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan pusat.

```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
>  center=[0,0,-2],frame=3):
```

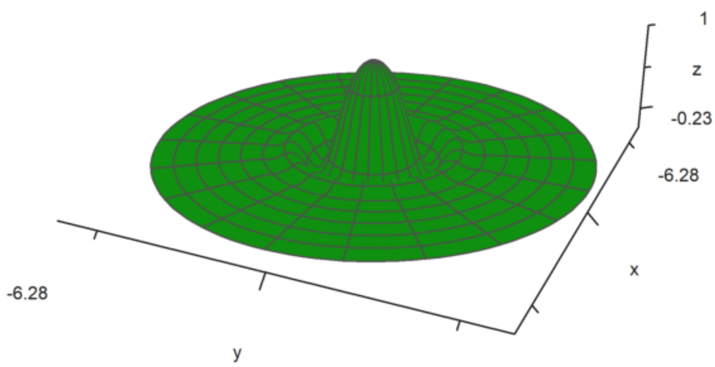


Plot kutub juga tersedia. The parameter polar= true menggambarkan plot kutub. Fungsi tersebut harus tetap merupakan fungsi dari x dan y. Parameter "fscale" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi diskalakan agar sesuai dengan kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=gray):
```

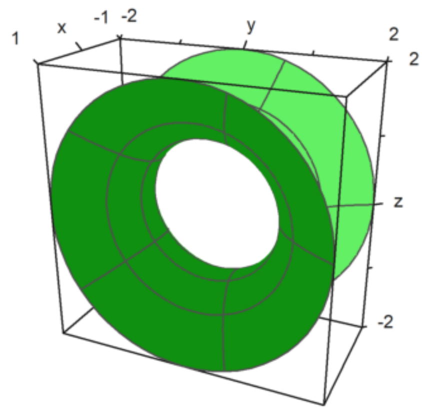



```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=2pi,frame=3,zoom=4):
```



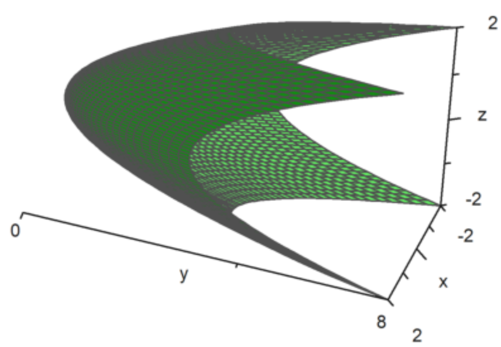
Parameter memutar memutar fungsi dalam x di sekitar sumbu x.
- rotate = 1: Menggunakan sumbu x
- rotate = 2: Menggunakan sumbu z

```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



Berikut adalah plot dengan tiga fungsi.

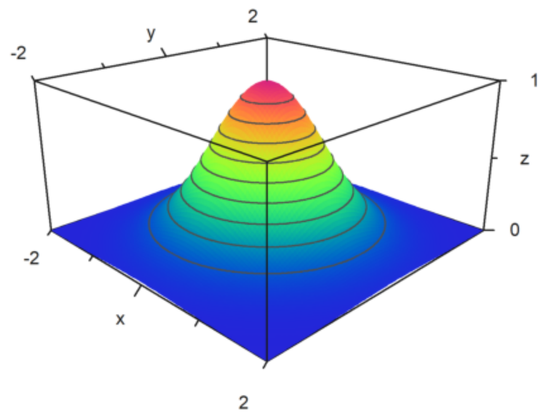
```
>plot3d("x","x^2+y^2","y",r=2,zoom=3.5,frame=3):
```



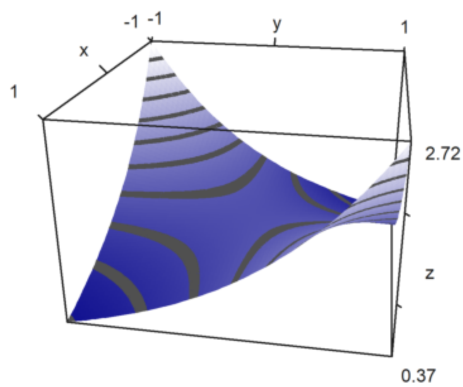
Plot Kontur

Untuk plot, Euler menambahkan garis kisi. Alih-alih, dimungkinkan untuk menggunakan garis level dan corak satu warna atau corak warna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan shading. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah / cyan.
-> hue: Mengaktifkan bayangan terang, bukan kabel.
-> contour: Membuat plot garis kontur otomatis pada plot.
- level = ... (atau level): Vektor nilai untuk garis kontur.
Standarnya adalah level = "auto", yang menghitung beberapa baris level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.
Gaya default dapat diubah. Untuk plot kontur berikut, kami menggunakan kisi yang lebih halus untuk 100x100 titik, menskalakan fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...
> >contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):
```

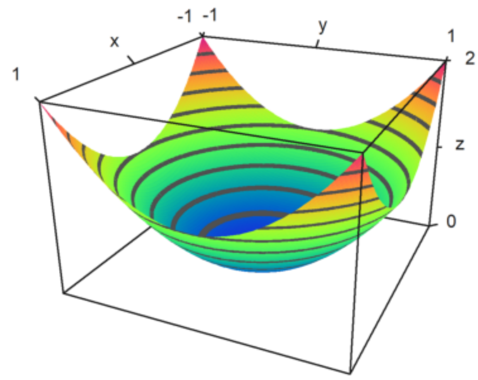



```
>plot3d("exp(x*y)",angle=100°,>contour,color=blue):
```



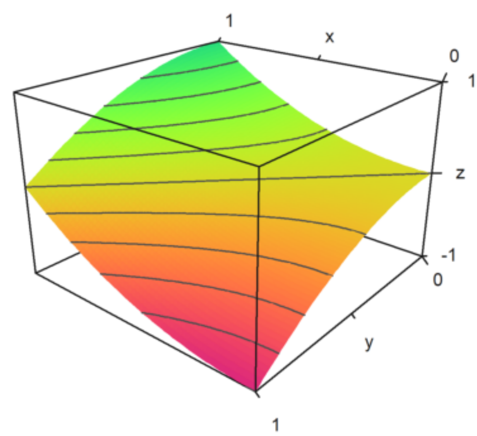
Bayangan default menggunakan warna abu-abu. Tetapi berbagai spektrum warna juga tersedia.
-> spectral: Menggunakan skema spektral default
- color = ...: Menggunakan warna khusus atau skema spektral
Untuk plot berikut, kami menggunakan skema spektral default dan menambah jumlah poin untuk mendapatkan tampilan yang sangat mulus.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```



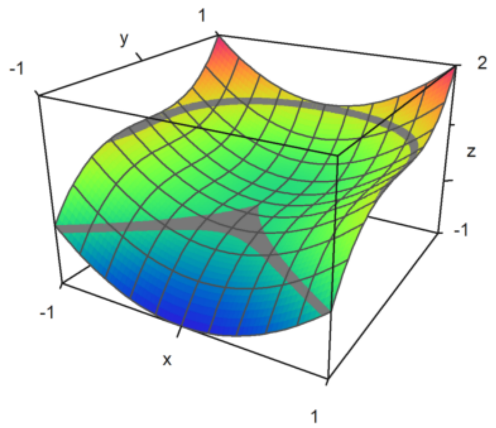
Alih-alih garis level otomatis, kami juga dapat mengatur nilai garis level. Ini akan menghasilkan garis level tipis, bukan rentang level.

```
>plot3d("x^2-y^2",0,1,0,1,angle=220°,level=-1:0.2:1,color=redgreen):
```



Dalam plot berikut, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas level sebagai kolom.
Selain itu, kami melapisi kisi dengan 10 interval di setiap arah.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...
>  >spectral,angle=30°,grid=10,contourcolor=gray):
```

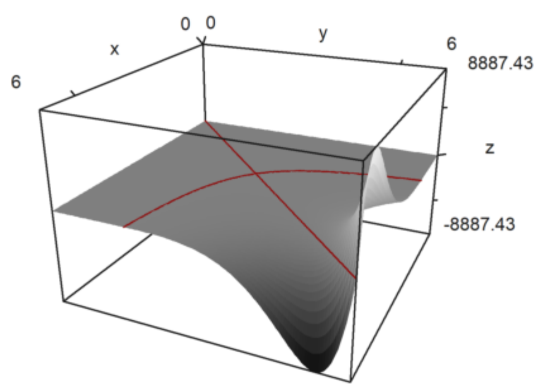


Dalam contoh berikut, kami memplot set, di mana

$$f(x,y) = x^y - y^x = 0$$

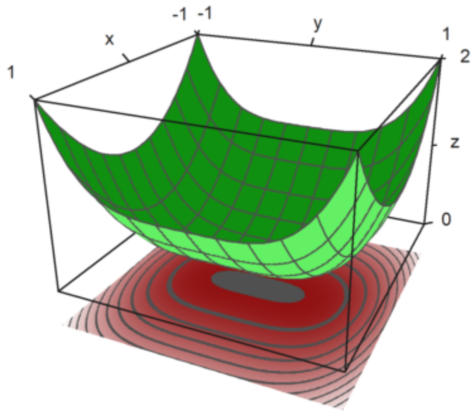
Kami menggunakan satu garis tipis untuk garis level.

```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```



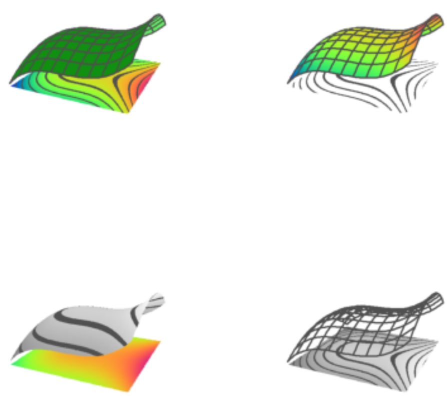
Dimungkinkan untuk menunjukkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

```
>plot3d("x^2+y^4",>cp,cpcolor=red,cpdelta=0.2):
```



Berikut beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi.

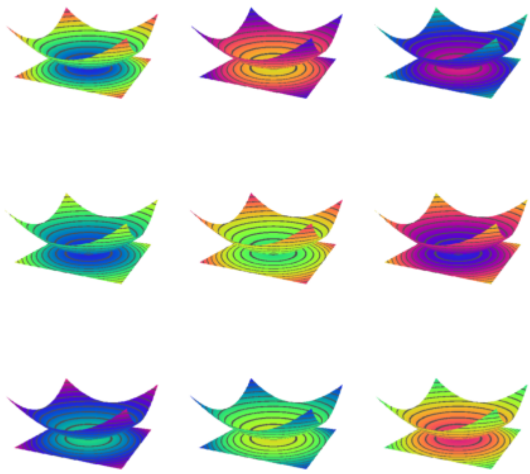
```
>figure(2,2); ...
>expr="y^3-x^2"; ...
>figure(1);  ...
>  plot3d(expr,<frame,>cp,cpcolor=spectral); ...
>figure(2);  ...
>  plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
>figure(3);  ...
>  plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
>figure(4);  ...
>  plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...
>figure(0):
```



Ada beberapa skema spektral lain, diberi nomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan color=value, di mana nilai
- spectral: untuk rentang dari biru hingga merah
- white: untuk rentang yang lebih redup

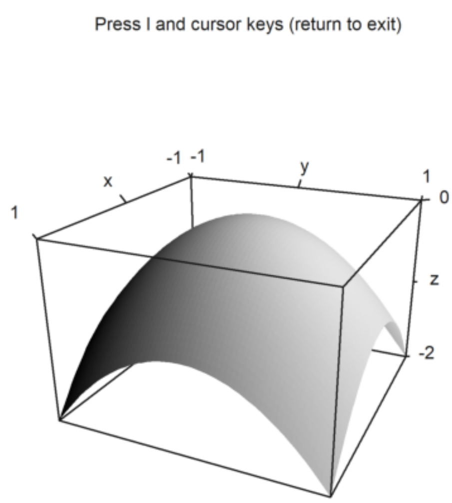
- yellowblue,purplegreen,blueyellow,greenred
- blueyellow, greenpurple,yellowblue,redgreen

```
>figure(3,3); ...
>for i=1:9;  ...
>  figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4);  ...
>end; ...
>figure(0):
```



Sumber cahaya dapat diubah dengan l dan tombol kursor selama interaksi pengguna. Itu juga dapat diatur dengan parameter.
-light : arah cahaya
- amb: cahaya ambient antara 0 dan 1
Perhatikan bahwa program tidak membuat perbedaan antara sisi-sisi plot. Tidak ada bayangan. Untuk ini, Anda membutuhkan Povray.

```
>plot3d("-x^2-y^2", ...
>  hue=true,light=[0,1,1],amb=0,user=true, ...
>  title="Press l and cursor keys (return to exit)"):
```

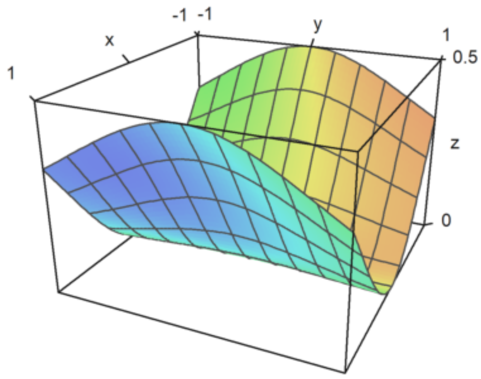


Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
>  zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01):
```

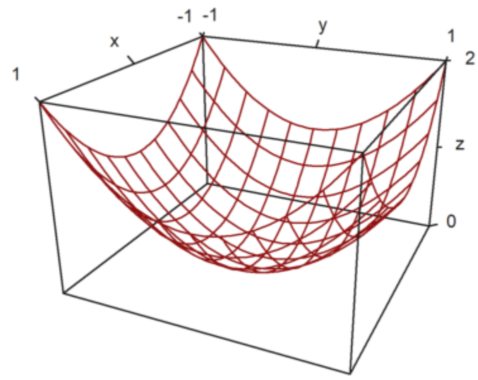
Warna 0 memberikan efek pelangi khusus.

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



Permukaannya juga bisa transparan.

```
>plot3d("x^2+y^2",>transparent,grid=10,wirecolor=red):
```



Plot Implisit

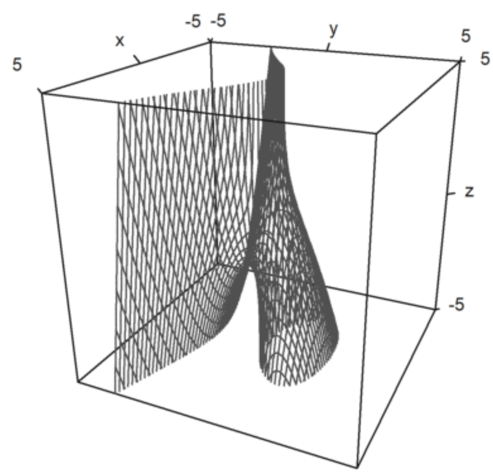
Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan pemotongan melalui objek. Fitur plot3d termasuk plot implisit. Plot ini menunjukkan himpunan nol fungsi dalam tiga variabel.
Solusi dari

$$f(x,y,z)=0$$

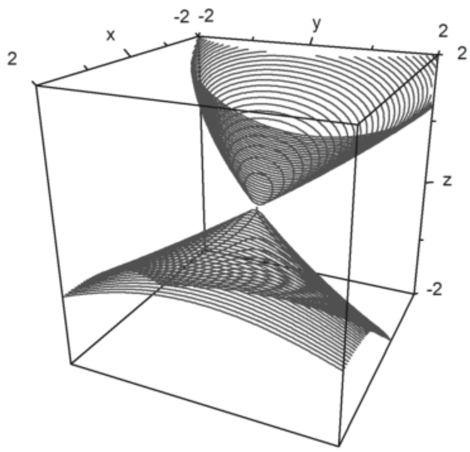
dapat divisualisasikan dalam potongan sejajar dengan bidang x-y-, x-z-, dan y-z.
- implisit = 1: potong sejajar bidang y-z
- implisit = 2: potong sejajar dengan bidang x-z
- implisit = 4: potong sejajar bidang x-y
Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh yang kami plot

$$M=\{(x,y,z):x^2+y^3+zy=1\}$$

```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```



```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):
```



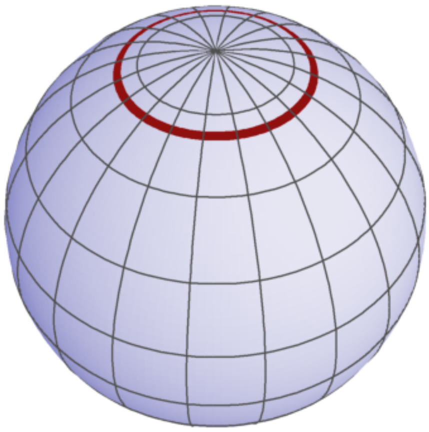
Plotting 3D Data

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu memberikan matriks nilai x, y dan z, atau tiga fungsi atau ekspresi fx (x, y), fy (x, y), fz (x, y).

$$\gamma(t,s) = (x(t,s), y(t,s), z(t,s))$$

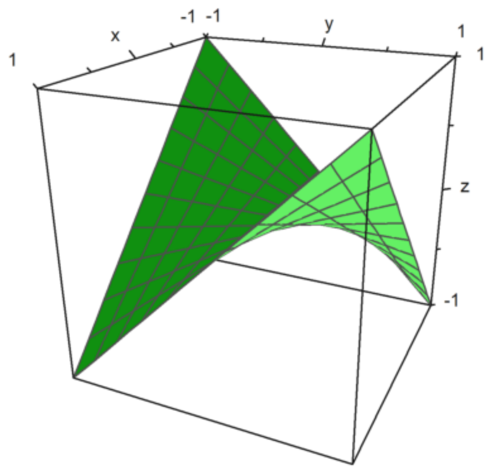
Karena x, y, z adalah matriks, kita asumsikan bahwa (t, s) berjalan melalui kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di luar angkasa.
Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.
Dalam contoh berikut, kami menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar kita bisa menandai daerah, dalam kasus kita daerah kutub.

```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...
>plot3d(x,y,z,>hue, ...
>color=blue,<frame,grid=[10,20], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.4,height=50°):
```



Berikut adalah contoh grafik dari suatu fungsi.

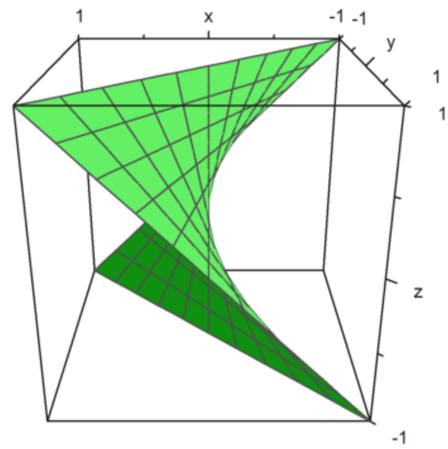
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



Namun, kita bisa membuat semua jenis permukaan. Ini adalah permukaan yang sama sebagai suatu fungsi

$$x = y\,z$$

```
>plot3d(t*s,t,s,angle=180°,grid=10):
```



Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.
Dalam contoh berikut, kami membuat tampilan berbayang dari bola yang terdistorsi. Koordinat biasa untuk bola adalah

$$\gamma(t,s) = (\cos(t)\cos(s), \sin(t)\sin(s), \cos(s))$$

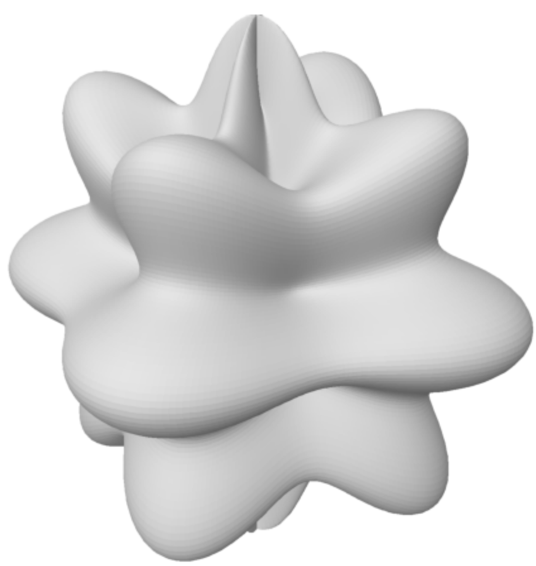
dengan

$$0 \le t \le 2\pi, \quad \frac{-\pi}{2} \le s \le \frac{\pi}{2}.$$

Kami menyimpangkan ini dengan sebuah faktor

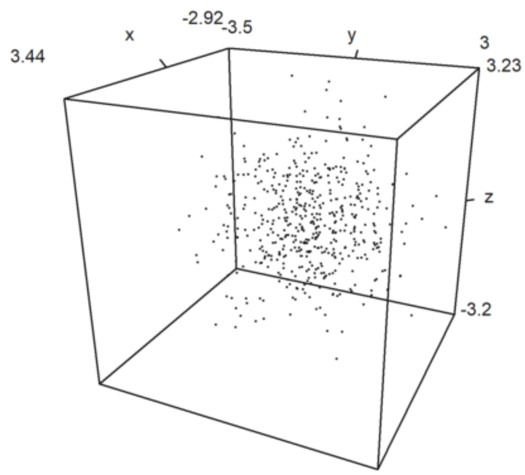
$$d(t,s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
>  light=[1,0,1],frame=0,zoom=5):
```



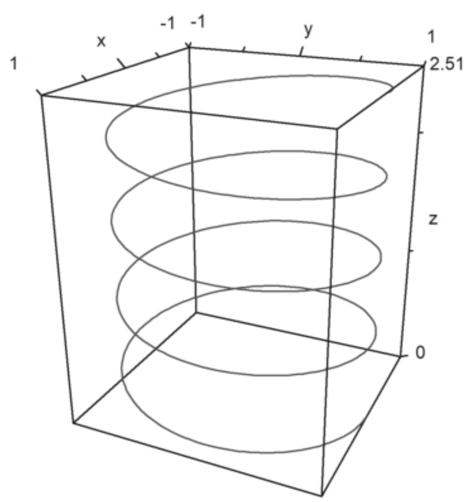
Tentu saja, point cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor sebagai koordinat titik.
Gayanya sama seperti di plot2d dengan points = true;

```
>n=500;  ...
>  plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

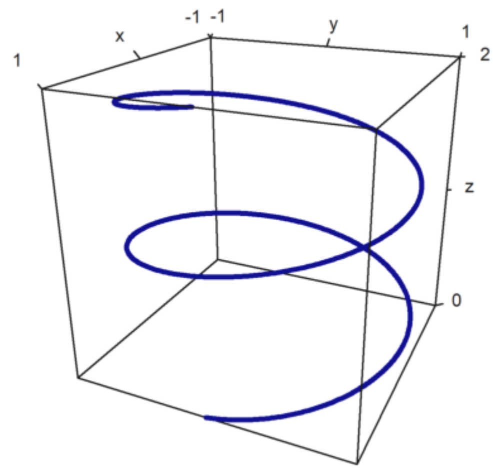


Juga dimungkinkan untuk memplot kurva dalam 3D. Dalam kasus ini, lebih mudah untuk menghitung sebelumnya titik-titik kurva. Untuk kurva di bidang kita menggunakan urutan koordinat dan parameter wire = true.

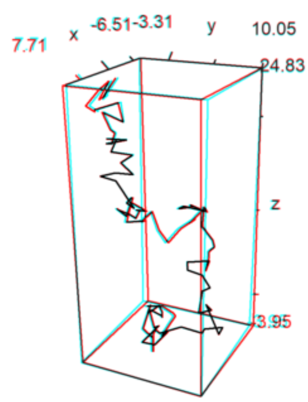
```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
>linewidth=3,wirecolor=blue):
```

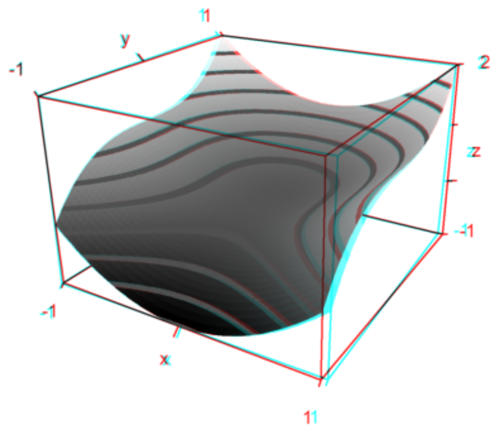



```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



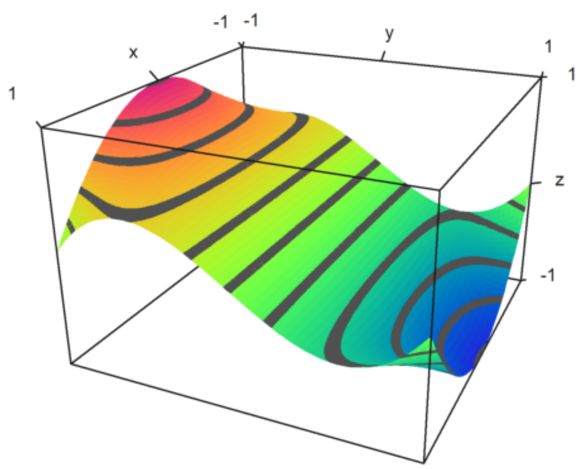
EMT juga dapat memplot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda membutuhkan red/cyan glasses.

```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```



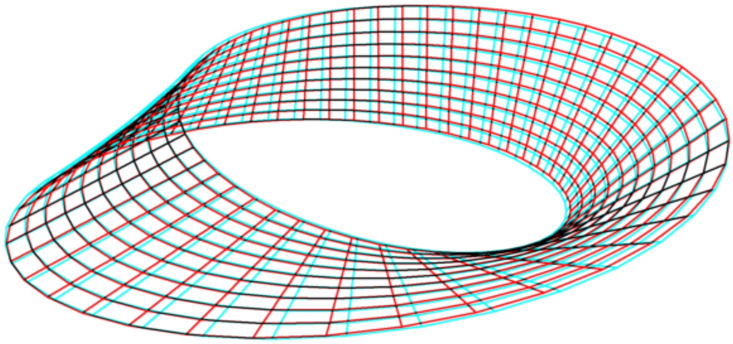
Seringkali, skema warna spektral digunakan untuk plot. Ini menekankan ketinggian fungsinya.

```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```



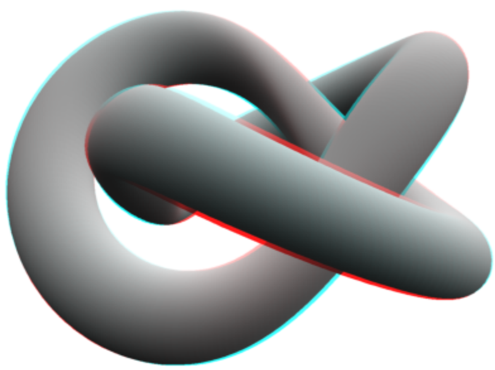
Euler juga dapat memplot permukaan berparameter, jika parameternya adalah nilai x, y, dan z dari gambar kisi persegi panjang di dalam ruang.
Untuk demo berikut, kami menyiapkan parameter u- dan v-, dan menghasilkan koordinat ruang dari ini.

```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...
>X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```



Berikut adalah contoh yang lebih rumit, yang megah dengan red/cyan glasses.

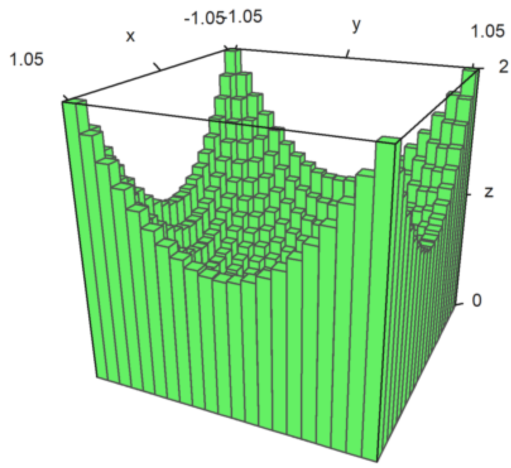
```
>u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)';  ...
>x:=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...
>y:=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...
> z=sin(u)+2*cos(3*v); ...
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```



*Plot Statistik
Plot batang juga dimungkinkan. Untuk ini, kami harus menyediakan
- x: vektor baris dengan n + 1 elemen
- y: vektor kolom dengan n + 1 elemen
- z: nxn matriks nilai.

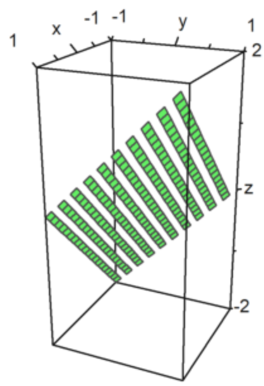
z bisa lebih besar, tetapi hanya nilai nxn yang akan digunakan.
Dalam contoh, pertama-tama kita menghitung nilainya. Kemudian kita menyesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...
>plot3d(xa,ya,z,bar=true):
```



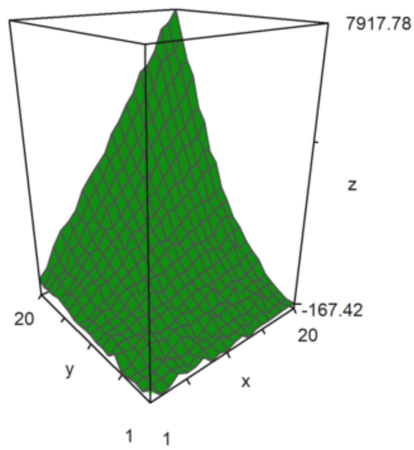
Dimungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20):
```

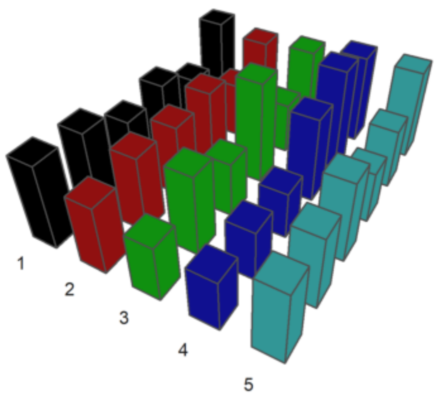


Jika memuat atau membuat matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan skala (M), atau menskalakan matriks dengan> zscale. Ini dapat dikombinasikan dengan faktor penskalaan individu yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8):
```

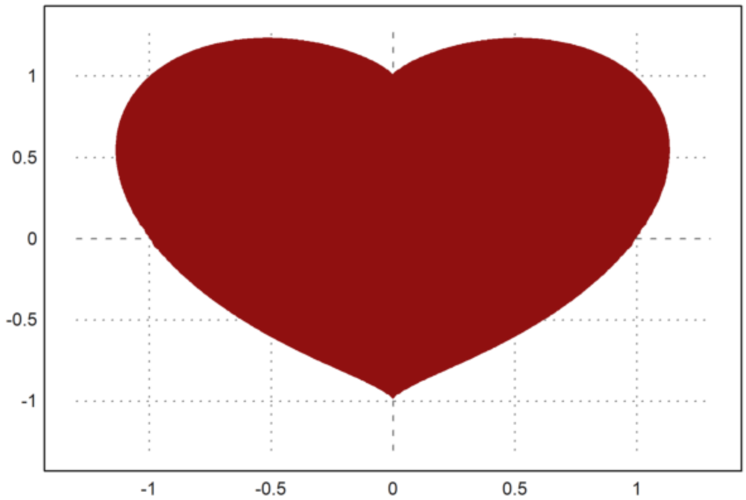



```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v',scols=1:5,ccols=[1:5]):
```



Permukaan Benda Putar

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>style="#",color=red,<outline, ...
>level=[-2;0],n=100):
```




```
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

$$\left(y^2+x^2-1\right)^3-x^2\,y^3$$

Kami ingin memutar kurva jantung di sekitar sumbu y. Inilah ungkapan yang mendefinisikan hati:

$$f(x,y)=(x^2+y^2-1)^3-x^2.y^3.$$

Selanjutnya kita atur

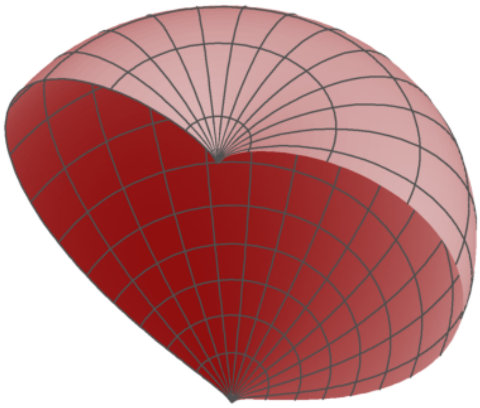
$$x=r.cos(a),\quad y=r.sin(a).$$

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

$$\left(r^2-1\right)^3+\frac{\left(\sin\left(5\,a\right)-\sin\left(3\,a\right)-2\,\sin a\right)\,r^5}{16}$$

Hal ini memungkinkan untuk menentukan fungsi numerik, yang menyelesaikan r, jika a diberikan. Dengan fungsi itu kita dapat memplot jantung yang berubah sebagai permukaan parametrik.

```
>function map f(a) := bisect("fr",0,2;a); ...
>t=linspace(-pi/2,pi/2,100); r=f(t);  ...
>s=linspace(pi,2pi,100)'; ...
>plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```

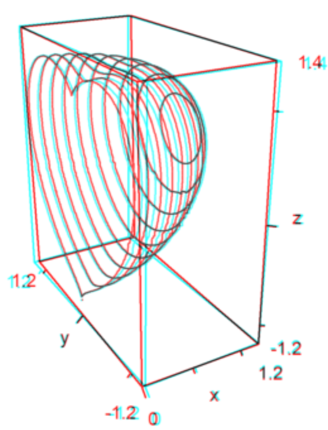


Berikut ini adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kami mendefinisikan fungsi, yang mendeskripsikan objek.

```
>function f(x,y,z) ...
```

```
  r=x^2+y^2;
  return (r+z^2-1)^3-r*z^3;
   endfunction
```

```
>plot3d("f(x,y,z)", ...
>xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...
>implicit=1,angle=-30°,zoom=2.5,n=[10,60,60],>anaglyph):
```



Special 3D Plots

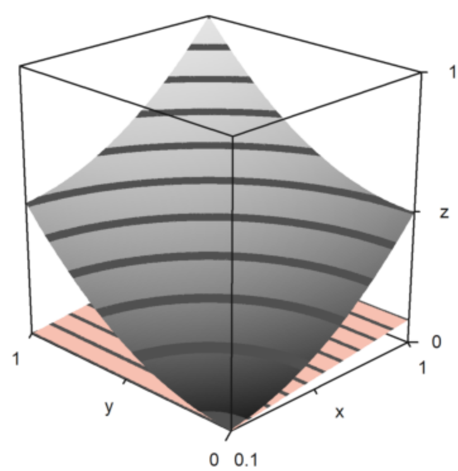
Fungsi plot3d bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda bisa mendapatkan plot berbingkai dari objek apa pun yang Anda suka.
Meskipun Euler bukan program 3D, Euler dapat menggabungkan beberapa objek dasar. Kami mencoba untuk memvisualisasikan paraboloid dan garis singgung-nya.

```
>function myplot ...
```

```
    y=0:0.01:1; x=(0.1:0.01:1)';
    plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ..
      hues=0.5,>contour,color=orange);
    h=holding(1);
    plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
    holding(h);
  endfunction
```

Sekarang framedplot () menyediakan bingkai, dan menyetel tampilan.

```
>framedplot("myplot",[0.1,1,0,1,0,1],angle=-45°, ...
>  center=[0,0,-0.7],zoom=6):
```

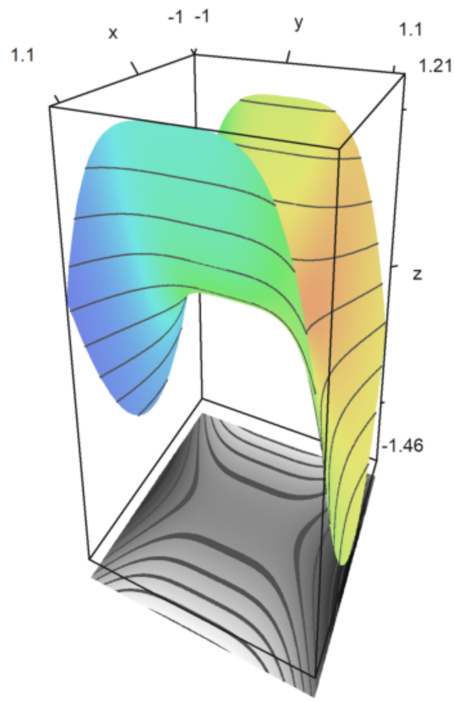


Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d () menyetel jendela ke fullwindow () secara default, tetapi plotcontourplane () mengasumsikannya.

```
>x=-1:0.02:1.1; y=x'; z=x^2-y^4;
>function myplot (x,y,z) ...
```

```
    zoom(2);
    wi=fullwindow();
    plotcontourplane(x,y,z,level="auto",<scale);
    plot3d(x,y,z,>hue,<scale,>add,color=white,level="thin");
    window(wi);
    reset();
  endfunction
```

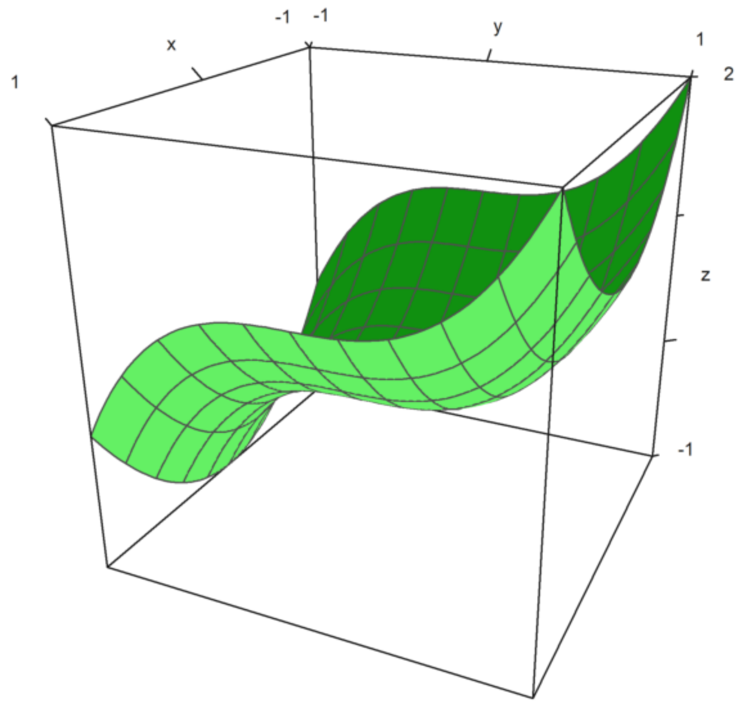
```
>myplot(x,y,z):
```



Animasi

Salah satu fungsi yang memanfaatkan teknik ini adalah memutar. Itu dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi tersebut memanggil addpage () untuk setiap plot baru. Akhirnya itu menjiwai plot.
Harap pelajari sumber rotasi untuk melihat lebih detail.

```
>function testplot () := plot3d("x^2+y^3"); ...
>rotate("testplot"); testplot():
```



Menggambar Povray

Dengan bantuan file Euler povray.e, Euler dapat menghasilkan file Povray. Hasilnya sangat bagus untuk dilihat.
Anda perlu menginstal Povray (32bit atau 64bit)dari
http://www.povray.org/,
dan meletakkan sub-direktori "bin" Povray ke dalam jalur lingkungan, atau menyetel variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe".
Antarmuka Povray dari Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah current.pov, dan direktori default adalah eulerhome (), biasanya c: \ Users \ Username \ Euler. Povray menghasilkan file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan file-file ini, gunakan povclear ().
Fungsi pov3d memiliki semangat yang sama dengan plot3d. Ini dapat menghasilkan grafik fungsi f (x, y), atau permukaan dengan koordinat X, Y, Z dalam matriks, termasuk garis level opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat pemandangan ke dalam notebook Euler.
Selain pov3d (), ada banyak fungsi yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string, berisi kode Povray untuk objek. Untuk menggunakan fungsi ini, mulai file Povray dengan povstart (). Kemudian gunakan writeln (...) untuk menulis objek ke file adegan. Terakhir, akhiri file dengan povend (). Secara default, raytracer akan mulai, dan PNG akan dimasukkan ke dalam notebook Euler.
Fungsi objek memiliki parameter yang disebut "look", yang membutuhkan string dengan kode Povray untuk tekstur dan penyelesaian objek. Fungsi povlook () dapat digunakan untuk menghasilkan string ini. Ini memiliki parameter untuk warna, transparansi, Phong Shading dll.
Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk ke atas secara vertikal, sumbu a nd x, y, z di tangan kanan.
Anda perlu memuat file povray.

```
> load povray;
```

Pastikan, direktori bin Povray ada di jalurnya. Jika tidak, edit variabel berikut sehingga berisi path ke povray yang dapat dieksekusi.

```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Untuk pertamakali, kami memplot fungsi sederhana. Perintah berikut menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk menelusuri file ini.

Jika Anda memulai perintah berikut, GUI Povray akan terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya, apakah Anda ingin mengizinkan file exe dijalankan. Anda dapat menekan batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK di jendela Povray untuk mengetahui dialog start-up Povray.

```
>pov3d("x^2+y^2",zoom=3);
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
pov3d:
    if povray then povray(currentfile,w,h,w/h); endif;
```

Kita bisa membuat fungsinya transparan dan menambahkan hasil akhir lainnya. Kita juga bisa menambahkan garis level ke plot fungsi.

```
>pov3d("x^2+y^3",axiscolor=red,angle=20°, ...
>  look=povlook(blue,0.2),level=-1:0.5:1,zoom=3.8);
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
pov3d:
    if povray then povray(currentfile,w,h,w/h); endif;
```

Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.

Kami memplot himpunan titik di bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=1.5, ...
>  angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=45°,n=50, ...
>  <fscale,zoom=3.8);
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
```

```
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
pov3d:
    if povray then povray(currentfile,w,h,w/h); endif;
```

Merencanakan dengan Koordinat

Alih-alih fungsi, kita bisa memplot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objeknya.
Dalam contoh ini, kita memutar fungsi di sekitar sumbu z.

```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,8)'; ...
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,height=20°,axis=0,zoom=4,light=[10,-5,5]);
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
pov3d:
    if povray then povray(currentfile,w,h,w/h); endif;
```

Dalam contoh berikut, kami memplot gelombang teredam. Kami menghasilkan gelombang dengan bahasa matriks Euler.
Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut. Perhatikan bahwa plot3d menskalakan plot, sehingga sesuai dengan kubus satuan

```
>r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ...
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3;  ...
>pov3d(x,y,z,zoom=5,axis=0,add=povsphere([0,0,0.5],0.1,povlook(green)), ...
>  w=500,h=300);
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
pov3d:
    if povray then povray(currentfile,w,h,w/h); endif;
```

Dengan metode naungan lanjutan Povray, sangat sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya di perbatasan dan dalam bayangan, triknya mungkin menjadi jelas.
Untuk ini, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>Z &= x^2*y^3
```

$$x^2 y^3$$

Persamaan permukaannya adalah [x, y, Z]. Kami menghitung dua turunan menjadi x dan y dari ini dan mengambil produk silang sebagai normal.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

Kami mendefinisikan normal sebagai produk silang dari turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

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

We use only 25 points.

```
>x=-1:0.5:1; y=x';
>pov3d(x,y,Z(x,y),angle=10°, ...
>  xv=NX(x,y),yv=NY(x,y),zv=NZ(x,y),<shadow);
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
pov3d:
    if povray then povray(currentfile,w,h,w/h); endif;
```

Berikut ini adalah simpul Trefoil yang dilakukan oleh A. Busser di Povray. Ada versi perbaikannya dalam contoh.

See: Examples\Trefoil Knot | Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normal bagi kami. Pertama, tiga fungsi koordinat sebagai ekspresi simbolik.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
>Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
>Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunannya menjadi x dan y.

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang normal, yang merupakan produk persilangan dari dua turunannya.

```
>dn &= crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

Vektor normal adalah evaluasi dari ekspresi simbolik dn [i] untuk i = 1,2,3. Sintaks untuk ini adalah & "expresi" (parameter). Ini adalah alternatif metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),axis=0,zoom=5,w=450,h=350, ...
>  <shadow,look=povlook(gray), ...
>  xv=&"dn[1]"(x,y), yv=&"dn[2]"(x,y), zv=&"dn[3]"(x,y));
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
pov3d:
    if povray then povray(currentfile,w,h,w/h); endif;
```

Kami juga dapat membuat grid dalam 3D.

```
>povstart(zoom=4); ...
>x=-1:0.5:1; r=1-(x+1)^2/6; ...
>t=(0°:30°:360°)'; y=r*cos(t); z=r*sin(t); ...
>writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...
>povend();
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Dengan povgrid (), kurva dimungkinkan.

```
>povstart(center=[0,0,1],zoom=3.6); ...
>t=linspace(0,2,1000); r=exp(-t); ...
>x=cos(2*pi*10*t)*r; y=sin(2*pi*10*t)*r; z=t; ...
>writeln(povgrid(x,y,z,povlook(red))); ...
>writeAxis(0,2,axis=3); ...
>povend();
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Objek Povray

Di atas, kami menggunakan pov3d untuk memplot permukaan. Antarmuka povray di Euler juga dapat menghasilkan objek Povray. Objek-objek ini disimpan sebagai string di Euler, dan perlu ditulis ke file Povray.
Kami memulai output dengan povstart ().

```
>povstart(zoom=4);
```

Pertama kita tentukan tiga silinder, dan simpan dalam string di Euler.
Fungsi povx () dll. Hanya mengembalikan vektor [1,0,0], yang bisa digunakan sebagai gantinya.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...
>c2=povcylinder(-povy,povy,1,povlook(green)); ...
>c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

```
>c1
```

```
cylinder { <-1,0,0>, <1,0,0>, 1
 texture { pigment { color rgb <0.564706,0.0627451,0.0627451> }  }
 finish { ambient 0.2 }
 }
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.
Itu dilakukan oleh povlook (), yang mengembalikan string dengan kode Povray yang relevan. Kita dapat menggunakan warna Euler default, atau menentukan warna kita sendiri. Kami juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

```
 texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> }  }
 finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek interseksi, dan menulis hasilnya ke file.

```
>writeln(povintersection([c1,c2,c3]));
```

Perpotongan tiga silinder sulit untuk divisualisasikan, jika Anda belum pernah melihatnya sebelumnya.

```
>povend;
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Fungsi berikut menghasilkan fraktal secara rekursif.
Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana. Fungsi povbox () mengembalikan string, berisi koordinat kotak, tekstur, dan hasil akhir.

```
>function onebox(x,y,z,d) := povbox([x,y,z],[x+d,y+d,z+d],povlook());
>function fractal (x,y,z,h,n) ...
```

```
   if n==1 then writeln(onebox(x,y,z,h));
   else
     h=h/3;
     fractal(x,y,z,h,n-1);
     fractal(x+2*h,y,z,h,n-1);
     fractal(x,y+2*h,z,h,n-1);
     fractal(x,y,z+2*h,h,n-1);
     fractal(x+2*h,y+2*h,z,h,n-1);
     fractal(x+2*h,y,z+2*h,h,n-1);
     fractal(x,y+2*h,z+2*h,h,n-1);
     fractal(x+2*h,y+2*h,z+2*h,h,n-1);
     fractal(x+h,y+h,z+h,h,n-1);
   endif;
  endfunction
```

```
>povstart(fade=10,<shadow);
>fractal(-1,-1,-1,2,4);
>povend();
```

```
  exec:
      return _exec(program,param,dir,print,hidden,wait);
  povray:
      exec(program,params,defaulthome);
  Try "trace errors" to inspect local variables after errors.
  povend:
      povray(file,w,h,aspect,exit);
```

Perbedaan memungkinkan pemotongan satu objek dari yang lain. Seperti persimpangan, ada bagian dari objek CSG Povray.

```
>povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kami mendefinisikan sebuah objek di Povray, daripada menggunakan string di Euler.
Definisi segera ditulis ke file.
Koordinat kotak -1 berarti [-1, -1, -1].

```
>povdefine("mycube",povbox(-1,1));
```

Kita bisa menggunakan objek ini di povobject (), yang mengembalikan string seperti biasa.

```
>c1=povobject("mycube",povlook(red));
```

Kami menghasilkan kubus kedua, dan memutar serta menskalakannya sedikit.

```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...
>  rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita ambil perbedaan kedua objek tersebut.

```
>writeln(povdifference(c1,c2));
```

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();
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Fungsi Implisit

Povray dapat memplot himpunan di mana f (x, y, z) = 0, seperti parameter implisit di plot3d. Namun, hasilnya terlihat jauh lebih baik.
Sintaks untuk fungsinya sedikit berbeda. Anda tidak dapat menggunakan keluaran ekspresi Maxima atau Euler.

```
>povstart(angle=70°,height=50°,zoom=4);
```

Buat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi tersebut.

```
>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...
>writeAxes(); ...
>povend();
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Objek Jaring

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.
Kami ingin memaksimalkan xy di bawah kondisi x + y = 1 dan mendemonstrasikan sentuhan tangensial dari garis level.

```
>povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kami tidak dapat menyimpan objek dalam string seperti sebelumnya, karena terlalu besar. Jadi kami mendefinisikan objek dalam file Povray menggunakan declare. Fungsi povtriangle () melakukan ini secara otomatis. Ia dapat menerima vektor normal seperti pov3d ().
Yang berikut ini mendefinisikan objek mesh, dan langsung menulisnya ke dalam file.

```
>x=0:0.02:1; y=x'; z=x*y; vx=-y; vy=-x; vz=1;
>mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kami mendefinisikan dua disk, yang akan berpotongan dengan permukaan.

```
>cl=povdisc([0.5,0.5,0],[1,1,0],2); ...
>ll=povdisc([0,0,1/4],[0,0,1],2);
```

Tulis permukaan dikurangi dua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tuliskan dua persimpangan tersebut.

```
>writeln(povintersection([mesh,cl],povlook(red))); ...
>writeln(povintersection([mesh,ll],povlook(gray)));
```

Tulis titik maksimal.

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));
```

Tambahkan sumbu dan selesai.

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...
>povend();
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Anaglyph di Povray

Untuk menghasilkan anaglyph untuk kacamata merah / cyan, Povray harus dijalankan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph ().

Tentu saja, Anda memerlukan kaca mata merah / cyan untuk melihat contoh berikut dengan benar.

Fungsi pov3d () memiliki tombol sederhana untuk menghasilkan anaglyph.

```
>pov3d("-exp(-x^2-y^2)/2",r=2,height=45°,>anaglyph, ...
>  center=[0,0,0.5],zoom=3.5);
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
pov3d:
    if povray then povray(currentfile,w,h,w/h); endif;
```

Jika Anda membuat adegan dengan objek, Anda perlu memasukkan pembuatan adegan ke dalam fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

```
>function myscene ...
```

```
  s=povsphere(povc,1);
  cl=povcylinder(-povz,povz,0.5);
  clx=povobject(cl,rotate=xrotate(90°));
  cly=povobject(cl,rotate=yrotate(90°));
  c=povbox([-1,-1,0],1);
  un=povunion([cl,clx,cly,c]);
  obj=povdifference(s,un,povlook(red));
  writeln(obj);
  writeAxes();
endfunction
```

Fungsi povanaglyph () melakukan semua ini. Parameternya seperti di povstart () dan povend () digabungkan.

```
>povanaglyph("myscene",zoom=4.5);
```

```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povanaglyph:
    povray(currentfile,w,h,aspect,exit);
```

Mendefinisikan Objek Sendiri

Antarmuka povray Euler berisi banyak objek. Tetapi Anda tidak dibatasi untuk ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau merupakan objek yang sama sekali baru.

Kami mendemonstrasikan torus. Perintah Povray untuk ini adalah "torus". Jadi kami mengembalikan string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat pada asalnya.

```
>function povdonat (r1,r2,look="") ...
```

```
    return "torus {"+r1+","+r2+look+"}";
  endfunction
```

Here is our first torus.

```
>t1=povdonat(0.8,0.2)
```

```
  torus {0.8,0.2}
```

Mari kita gunakan objek ini untuk membuat torus kedua, diterjemahkan dan diputar.

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
  object { torus {0.8,0.2}
   rotate 90 *x
   translate <0.8,0,0>
   }
```

Sekarang kami menempatkan objek ini ke dalam sebuah adegan. Untuk tampilan, kami menggunakan Phong Shading.

```
>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...
>writeln(povobject(t1,povlook(green,phong=1))); ...
>writeln(povobject(t2,povlook(green,phong=1))); ...
```

> povend ();
memanggil program Povray. Namun, jika terjadi kesalahan, itu tidak menampilkan kesalahan. Karena itu Anda harus menggunakan

```
 > povend (<exit);
```

jika ada yang tidak berhasil. Ini akan membuat jendela Povray terbuka.

```
>povend (h=320,w=480);
```

```
  Command was not allowed!
  exec:
      return _exec(program,param,dir,print,hidden,wait);
  povray:
      exec(program,params,defaulthome);
  Try "trace errors" to inspect local variables after errors.
  povend:
      povray(file,w,h,aspect,exit);
```

Berikut adalah contoh yang lebih lengkap. Kami menyelesaikannya

$$Ax \leq b, \quad x \geq 0, \quad c.x \to \text{Max.}$$

dan menunjukkan poin yang layak dan optimal dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
>b=[10,10,10,10]';
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini memiliki solusi.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0,  1,  0.5]
```

Yes, it has.
Next we define two objects. The first is the plane

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look="") ...
```

```
    return povplane(a,b,look)
  endfunction
```

Kemudian kami mendefinisikan perpotongan dari semua setengah spasi dan sebuah kubus.

```
>function adm (A, b, r, look="") ...
```

```
    ol=[];
    loop 1 to rows(A); ol=ol|oneplane(A[#],b[#]); end;
    ol=ol|povbox([0,0,0],[r,r,r]);
    return povintersection(ol,look);
  endfunction
```

We can now plot the scene.

```
>povstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); ...
>writeln(adm(A,b,2,povlook(green,0.4))); ...
>writeAxes(0,1.3,0,1.6,0,1.5); ...
```

The following is a circle around the optimum.
–Terjemahan
Berikut ini adalah lingkaran di sekitar optimal.

```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
>  povlook(red,0.9)));
```

Dan ada kesalahan di arah optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

```
>writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=125°)); ...
>povend();
```

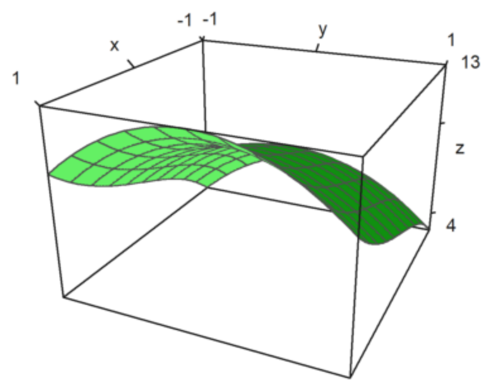
```
exec:
    return _exec(program,param,dir,print,hidden,wait);
povray:
    exec(program,params,defaulthome);
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

Latihan Soal

1. Buatkan grafik

$$f(x) = x^3 + 2x - 3y^2 + 10$$

```
>aspect(1.5); plot3d("x^3+2x-3*y^2+10"):
```

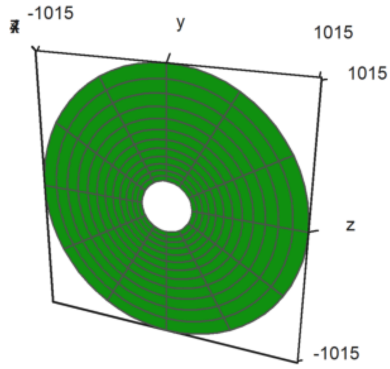


2. Buatkan grafik

$$f(x) = 3x^3 - 5x + 21$$

dengan a=4, b=7, dan grid=13

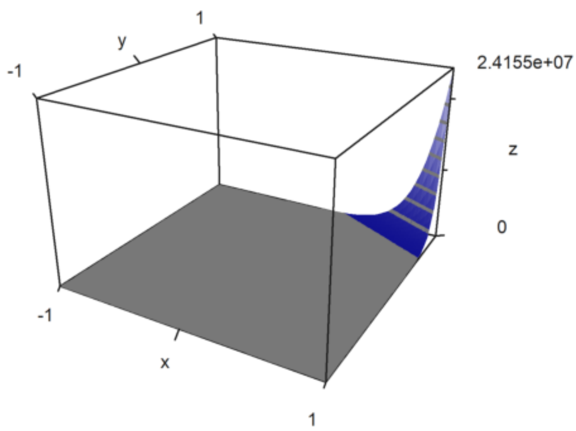
```
>aspect(1.5) ; plot3d("3*x^3-5*x+21",a=4,b=7,rotate=true,grid=13):
```



3. Buatlah grafik plot kontur

$$f(x) = e^{6x+11y}$$

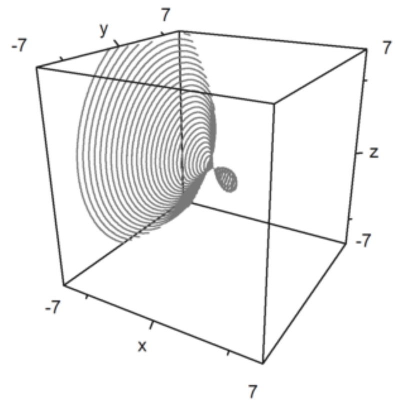
```
>plot3d("exp(6*x+11y)",angle=30°,>contour,color=blue):
```



4. Buat grafik plot implisit

$$x^3 + 5 * y^2 + 7xy + 3z^3$$

```
>plot3d("x^3+5*y^2+7*x*y+6*z^2",>implicit,r=7,zoom=2,angle=30°):
```



-7
y
7
7
z
-7
-7
x
7

BAB 4

MENGGUNAKAN EMT UNTUK KALKULUS

[a4paper,10pt]article eumat

Identitas Diri

Nama : Anisah Daffa Citra Nareswari
Kelas : Matematika E 2022
NIM : 22305141044

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 adalah beberapa contoh cara mendefinisikan fungsi.

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
```

```
1
4.31977682472
20.7392088022
```

```
>function g(x) := sqrt(x^2-3*x)/(x+1)
>g(3)
```

```
0
```

```
>g(0)
```

```
0
```

```
>g(1)
```

```
Floating point error!
Error in sqrt
Try "trace errors" to inspect local variables after errors.
g:
    useglobal; return sqrt(x^2-3*x)/(x+1)
Error in:
g(1) ...
    ^
```

```
>f(g(5)) // komposisi fungsi
```

```
2.20920171961
```

```
>g(f(5))
```

```
0.950898070639
```

```
>f(0:10) // nilai-nilai f(1), f(2), ..., f(10)
```

```
[1,  4.31978,  10.4826,  19.1516,  32.4692,  50.3833,  72.7562,
99.929,  130.69,  163.51,  200.58]
```

```
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
```

```
[1,  4.31978,  10.4826,  19.1516,  32.4692,  50.3833,  72.7562,
99.929,  130.69,  163.51,  200.58]
```

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \le 0. \end{cases}$$

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
```

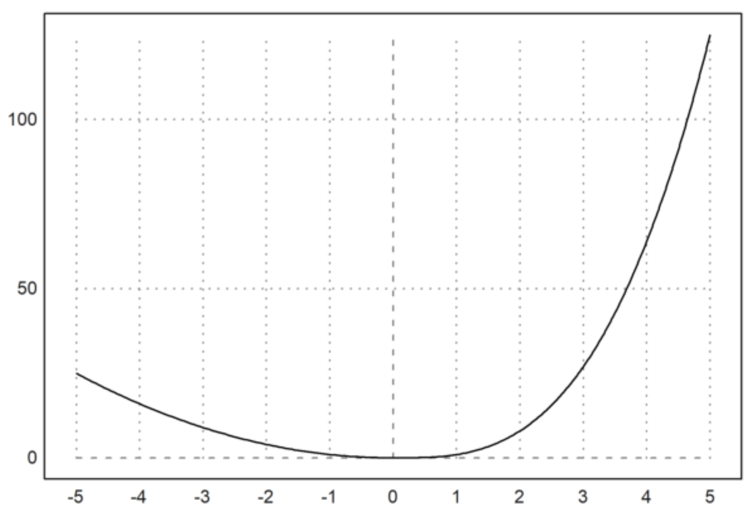
```
>f(-2)
```

```
  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
```

```
                x
              2 E
```

```
>function g(x) &= 3*x+1
```

```
                3 x + 1
```

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

```
                3 x + 1
              2 E
```

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik tersebut.
Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

Fungsi 1 Variabel

1. Fungsi 1

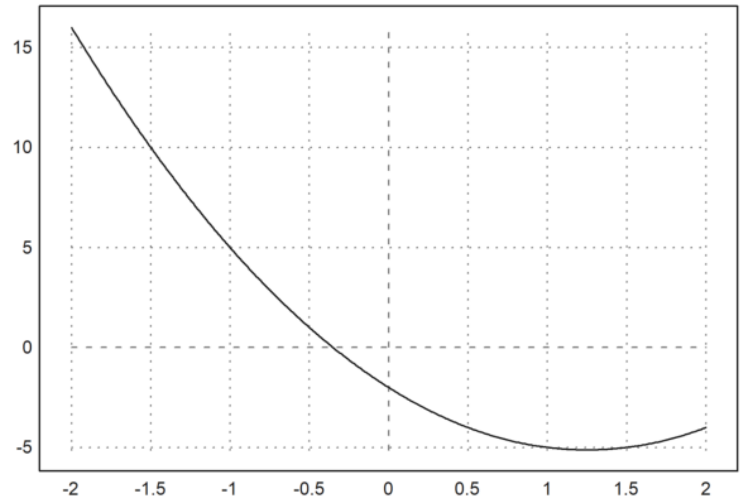
```
>function p(x) := 2*x^2-5x-2
>p(0), p(1), p(2)
```

```
-2
-5
-4
```

```
>pmap(0:3)
```

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

```
>plot2d("p(x)"):
```



2. Fungsi 2

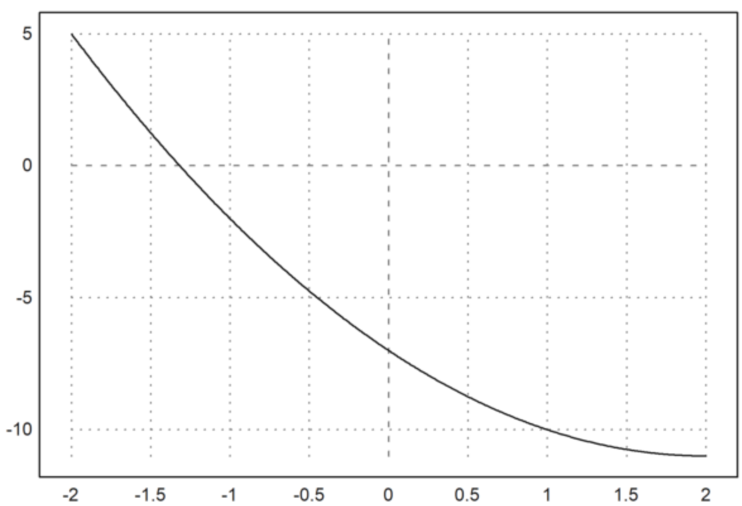
```
>function a(x) := x^2-4x-7
>a(1), a(5), a(-2)
```

```
-10
-2
5
```

```
>amap(-5:5)
```

```
[38,  25,  14,  5,  -2,  -7,  -10,  -11,  -10,  -7,  -2]
```

```
>plot2d("a(x)"):
```



3. Fungsi 3

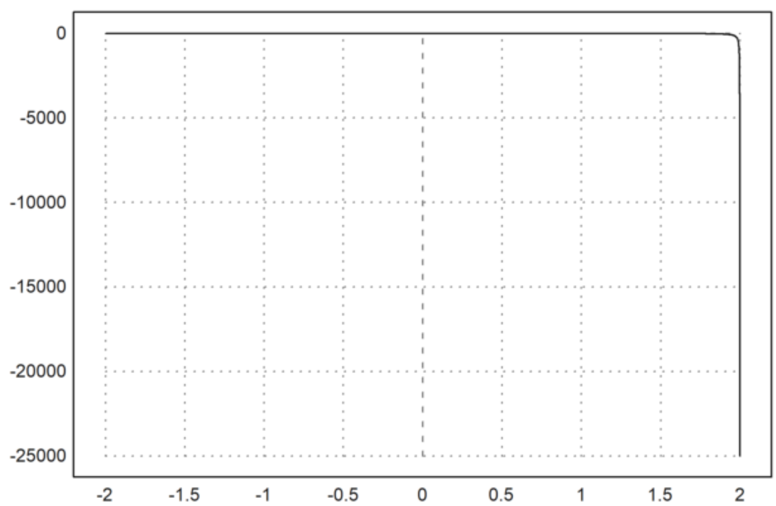
```
>function b(x) := (x+3)/(x-2)
>b(-3), b(1), b(10)
```

```
0
-4
1.625
```

```
>bmap(1:50)
```

```
Floating point error!
Try "trace errors" to inspect local variables after errors.
b:
    useglobal; return (x+3)/(x-2)
Error in map.
Error in:
bmap(1:50)  ...
          ^
```

```
>plot2d("b(x)"):
```




```
>
```

4.Fungsi 4

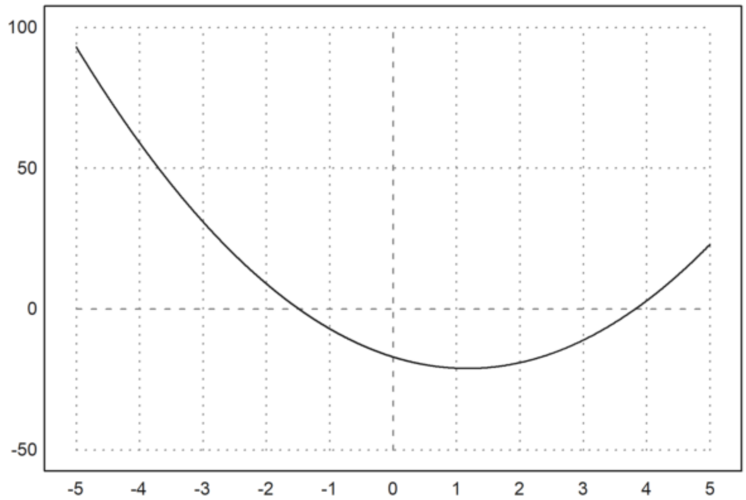
```
> function c(x) := 3*x^2-7x-17
>c(0), c(-1), c(1)
```

```
-17
-7
-21
```

```
>cmap(-5:5)
```

```
[93,  59,  31,  9,  -7,  -17,  -21,  -19,  -11,  3,  23]
```

```
>plot2d("c(x)",-5,5,-50,100):
```



5. Fungsi 5

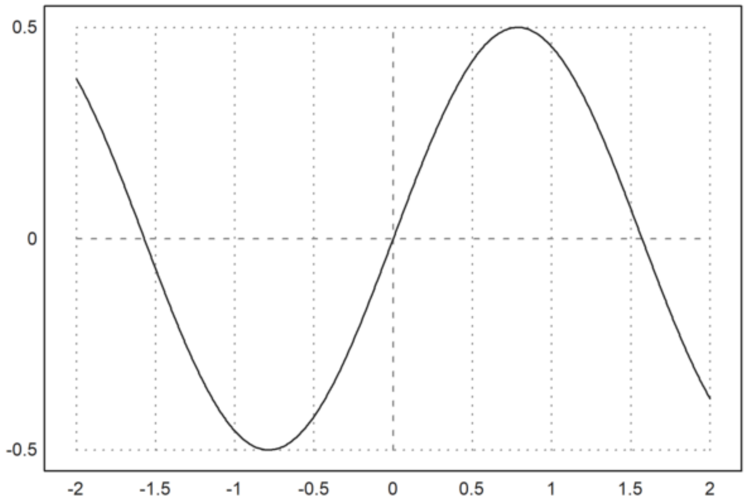
```
>function d(x) := (sin(x))*(cos(x))
>d(0), d(pi/5), d(pi/10)
```

```
0
0.475528258148
0.293892626146
```

```
>dmap(0:5pi)
```

```
[0,  0.454649,  -0.378401,  -0.139708,  0.494679,  -0.272011,
-0.268286,  0.495304,  -0.143952,  -0.375494,  0.456473,  -0.00442565,
-0.452789,  0.381279,  0.135453,  -0.494016]
```

```
>plot2d("d(x)"):
```



Fungsi 2 Variabel

```
>
```

1. Fungsi 1

```
>function e(x,y) ...
```

```
return x^3+y^2
endfunction
```

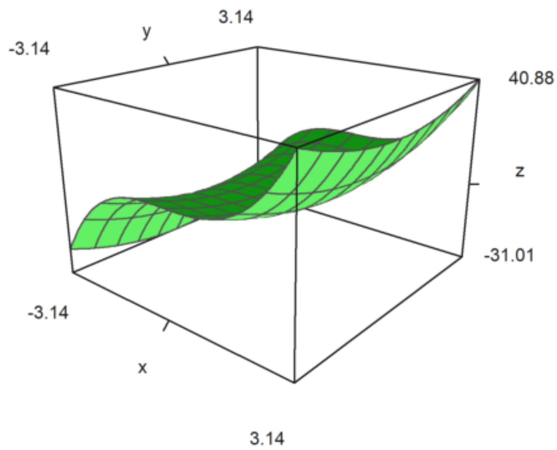
```
>e(0,2), e(1,4), e(2,1)
```

```
4
17
9
```

```
>emap(0:3,5:8)
```

```
[25,   37,   57,   91]
```

```
>aspect=1.5; plot3d("e(x,y)",a=-100,b=100,c=-80,d=80,angle=40°,height=20°,r=pi,n=100):
```



2. Fungsi 2

```
>function f(x,y) ...
```

```
return sqrt(x^2+y^2)
endfunction
```

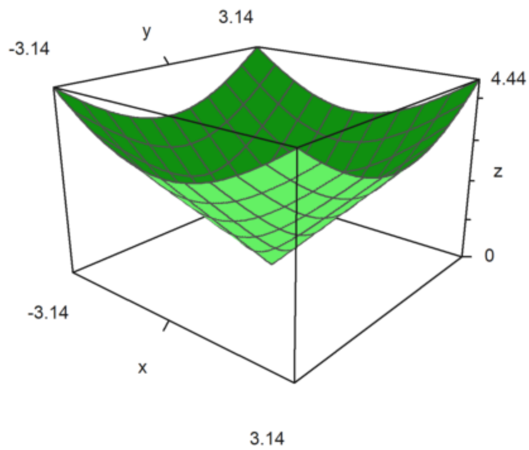
```
>f(1,5), f(4,6), f(-1,4)
```

```
5.09901951359
7.21110255093
4.12310562562
```

```
>fmap(-3:0,1:4)
```

```
[3.16228,   2.82843,   3.16228,   4]
```

```
>aspect=1.5; plot3d("f(x,y)",a=-100,b=100,c=-80,d=80,angle=40°,height=20°,r=pi,n=100):
```



Menghitung Limit

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi "plus" atau "minus". Hasil limit dapat berupa nilai, "und' (tak definisi), "ind" (tak tentu namun terbatas), "infinity" (kompleks tak hingga).
Perhatikan beberapa contoh berikut. Perhatikan cara menampilkan perhitungan secara lengkap, tidak hanya menampilkan hasilnya saja.

```
>$showev('limit(1/(2*x-1),x,0))
```

$$\lim_{x\rightarrow 0}{\frac{1}{2\,x-1}}=-1$$

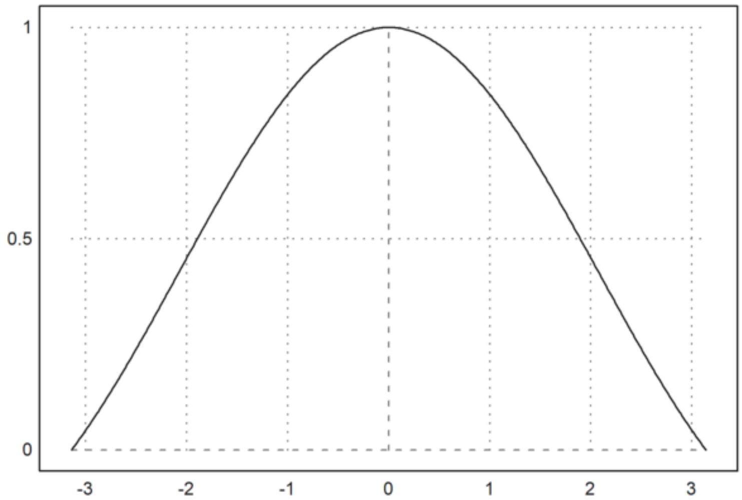
```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

$$\lim_{x\rightarrow 5}{\frac{x^2-3\,x-10}{x-5}}=7$$

```
>$showev('limit(sin(x)/x,x,0))
```

$$\lim_{x\rightarrow 0}{\frac{\sin x}{x}}=1$$

```
>plot2d("sin(x)/x",-pi,pi):
```




```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x \to 0} \frac{\sin x^3}{x} = 0$$

```
>$showev('limit(log(x), x, minf))
```

$$\lim_{x \to -\infty} \log x = \mathit{infinity}$$

```
>$showev('limit((-2)^x,x, inf))
```

$$\lim_{x \to \infty} (-2)^x = \mathit{infinity}$$

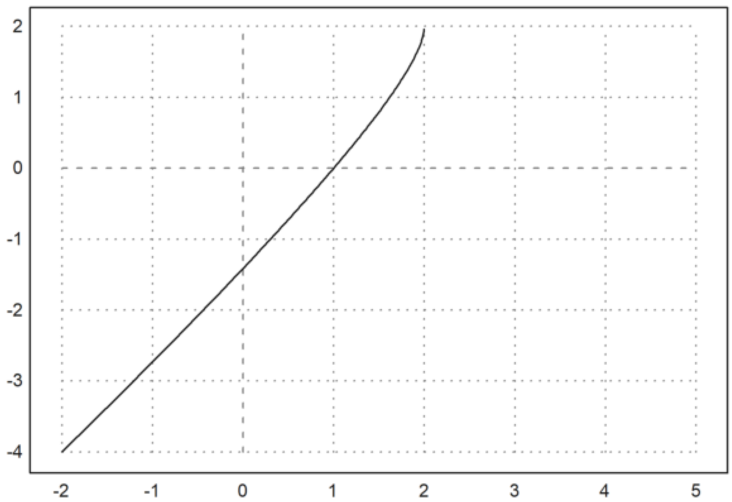
```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t \uparrow 2} t - \sqrt{2-t} = 2$$

```
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
```

$$\lim_{t \downarrow 5} t - \sqrt{2-t} = 5 - \sqrt{3}\, i$$

```
>plot2d("x-sqrt(2-x)",-2,5):
```




```
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
```

$$\lim_{x\to 3}\frac{x^2-9}{2\,x^2-5\,x-3}=\frac{6}{7}$$

```
>$showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x\to 0}\frac{1-\cos x}{x}=0$$

```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

$$\lim_{x\to 0}\frac{|x|+x^2}{x^2-|x|}=-1$$

```
>$showev('limit((1+1/x)^x,x,inf))
```

$$\lim_{x\to\infty}\left(\frac{1}{x}+1\right)^x=e$$

```
>$showev('limit((1+k/x)^x,x,inf))
```

$$\lim_{x\to\infty}\left(\frac{k}{x}+1\right)^x=e^k$$

```
>$showev('limit((1+x)^(1/x),x,0))
```

$$\lim_{x\to 0}\left(x+1\right)^{\frac{1}{x}}=e$$

```
>$showev('limit((x/(x+k))^x,x,inf))
```

$$\lim_{x\to\infty}\left(\frac{x}{x+k}\right)^{x}=e^{-k}$$

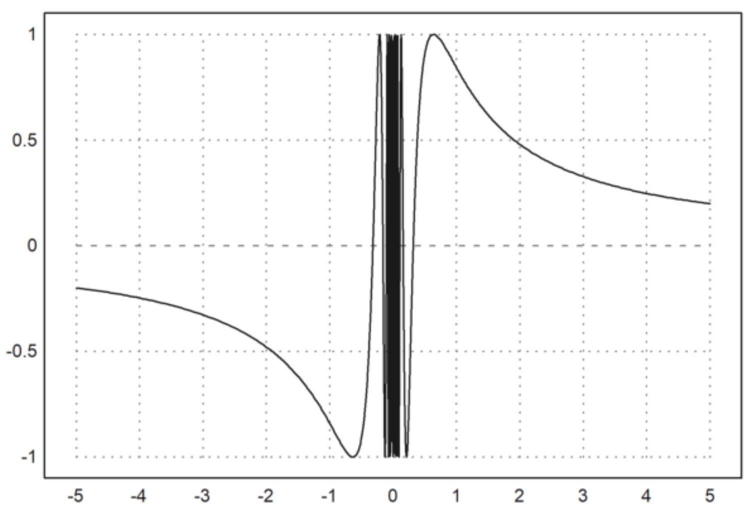
```
>$showev('limit(sin(1/x),x,0))
```

$$\lim_{x\to 0}\sin\left(\frac{1}{x}\right)=ind$$

```
>$showev('limit(sin(1/x),x,inf))
```

$$\lim_{x\to\infty}\sin\left(\frac{1}{x}\right)=0$$

```
>plot2d("sin(1/x)",-5,5):
```



Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

1. Fungsi 1

$$f'(x)=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>$showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
```

$$\lim_{h\to 0}{\frac{\left(x+h\right)^{n}-x^{n}}{h}}=n\,x^{n-1}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.
Sebagai petunjuk, ekspansikan (x+h)^n dengan menggunakan teorema binomial.

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

$$\lim_{h\to 0}{\frac{\sin \left(x+h\right)-\sin x}{h}}=\cos x$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.
Sebagai petunjuk, ekspansikan sin(x+h) dengan menggunakan rumus jumlah dua sudut.

```
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
```

$$\lim_{h\to 0}{\frac{\log \left(x+h\right)-\log x}{h}}=\frac{1}{x}$$

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
```

$$\lim_{h\to 0}{\frac{\frac{1}{x+h}-\frac{1}{x}}{h}}=-\frac{1}{x^2}$$

```
>$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x
```

```
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
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:

$$\lim_{h\to 0} \frac{e^{x+h} - e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h\to 0} \frac{e^h - 1}{h} = 1$$

```
>$factor(E^(x+h)-E^x)
```

$$\left(e^h - 1\right)\, e^x$$

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

$$\left(\lim_{h\to 0} \frac{e^h - 1}{h}\right)\, e^x = e^x$$

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

```
                                  x
                                 x
```

```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h\to 0} \frac{(x+h)^{x+h} - x^x}{h} = \mathit{infinity}$$

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h\to 0} \frac{(x+h)^{x+h} - x^x}{h}.$$

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
```

$$\lim_{h\to 0} \frac{(x+h)^{x+h} - x^x}{h} = x^x\, (\log x + 1)$$

```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

```
[x > 0]
```

```
>&forget(x<0)
```

```
[x < 0]
```

```
>&facts()
```

```
[]
```

```
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
```

$$\lim_{h\to 0}\frac{\arcsin\left(x+h\right)-\arcsin x}{h}=\frac{1}{\sqrt{1-x^2}}$$

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

$$\lim_{h\to 0}\frac{\tan\left(x+h\right)-\tan x}{h}=\frac{1}{\cos^2 x}$$

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

```
sinh(x)
```

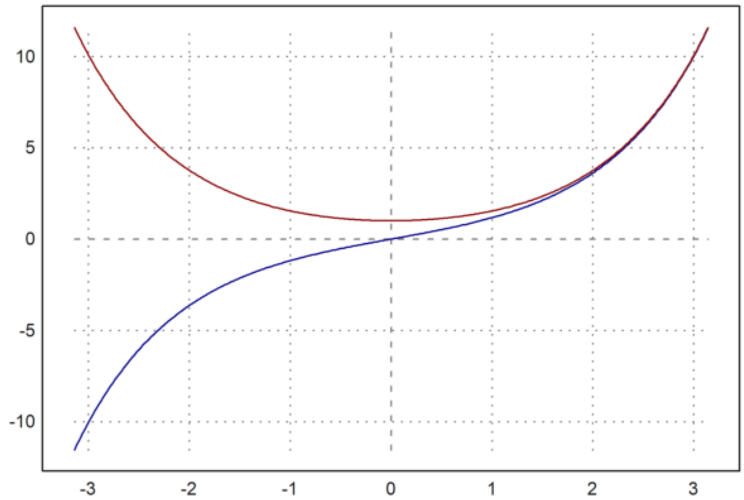
```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

$$\frac{e^{-x}\,\left(e^{2\,x}+1\right)}{2}$$

Hasilnya adalah cosh(x), karena

$$\frac{e^x+e^{-x}}{2}=\cosh(x).$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```



Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), seperti contoh-contoh tersebut. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.
1. Fungsi 1

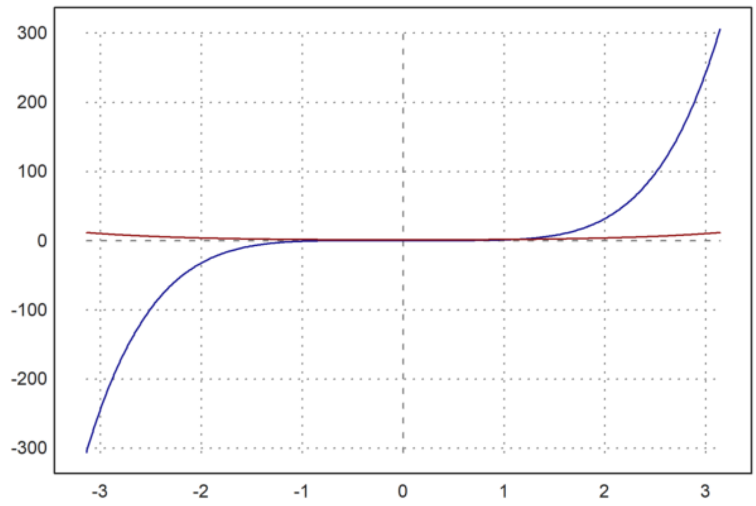
```
>function f(x) := (x^5)
>$showev('limit((((x+h)^5)-x^5)/h,h,0)) // Turunan x^5
```

$$\lim_{h\to 0}{\frac{\left(x+h\right)^5-x^5}{h}}=5\,x^4$$

```
>function df(x) &= limit((((x+h)^5)-x^5)/h,h,0): $df(x)// df(x) = f'(x)
```

```
 - x    2 x
E    (E     + 1)
---------------
       2
```

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```



2. Fungsi 2

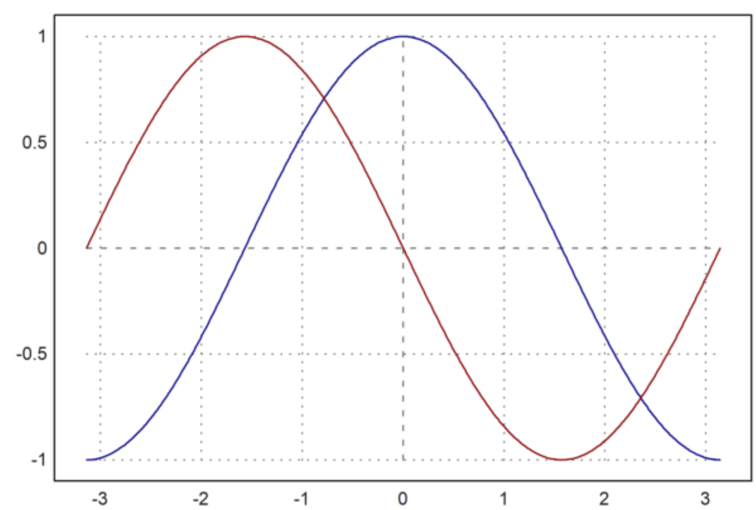
```
>function f(x) := cos(x)
>$showev('limit((cos(x+h)-cos(x))/h,h,0))
```

$$\lim_{h\to 0}{\frac{\cos \left(x+h\right)-\cos x}{h}}=-\sin x$$

```
>function df(x) &= limit((cos(x+h)-cos(x))/h,h,0);  $df(x)// df(x) = f'(x)
```

$$-\sin x$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```



3. Fungsi 3

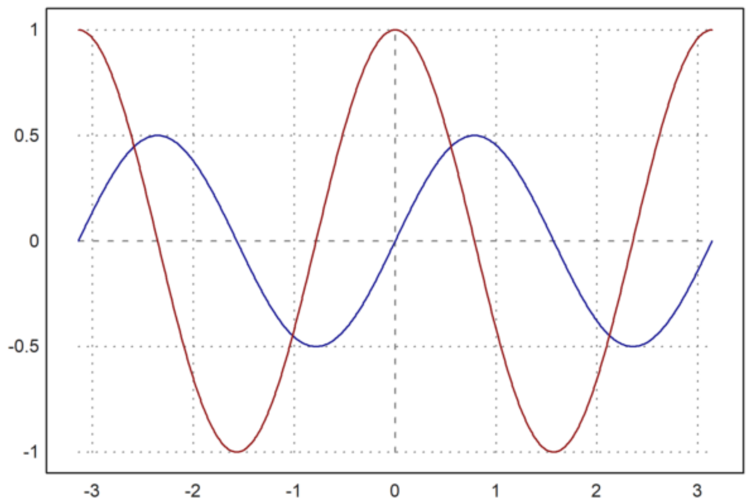
```
>function f(x) := sin(x)*cos(x)
>$showev('limit((sin(x+h)*cos(x+h)-sin(x)*cos(x))/h,h,0))
```

$$\lim_{h\to 0}\frac{\cos\left(x+h\right)\,\sin\left(x+h\right)-\cos x\,\sin x}{h}=\cos^2 x-\sin^2 x$$

```
>function df(x) &= limit((sin(x+h)*cos(x+h)-sin(x)*cos(x))/h,h,0);  $df(x)// df(x) = f'(x)
```

$$\cos^2 x-\sin^2 x$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```



4. Fungsi 4

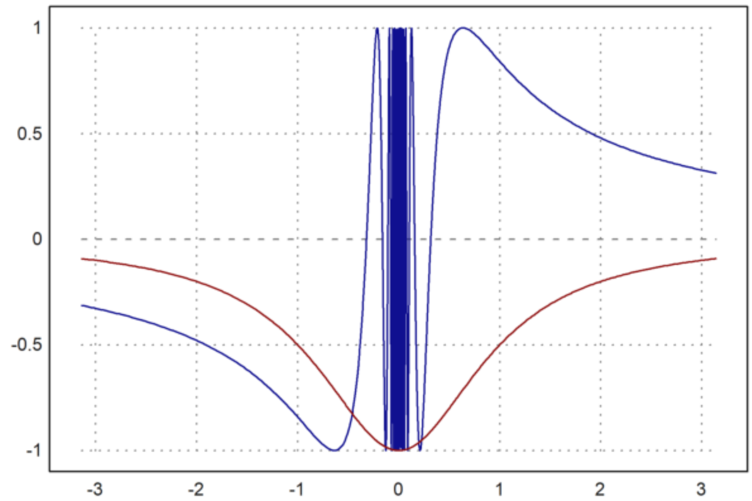
```
>function f(x) := sin(1/x)
>$showev('limit((sin(1/(x+h))-sin(1/x))/h,h,0))
```

$$\lim_{h\to 0}\frac{\sin\left(\frac{1}{x+h}\right)-\sin\left(\frac{1}{x}\right)}{h}=-\frac{\cos\left(\frac{1}{x}\right)}{x^2}$$

```
>function df(x) &= limit((atan(1/(x+h))-atan(1/x))/h,h,0);$df(x)// df(x) = f'(x)
```

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

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```



5. Fungsi 5

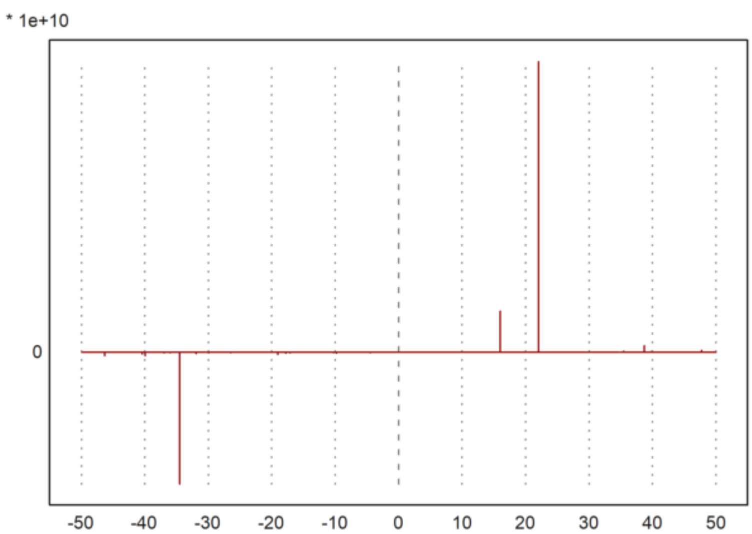
```
>function f(x) := tan(x^2)
>$showev('limit(((tan((x+h)^2))-tan(x^2))/h,h,0))
```

$$\lim_{h\to 0}\frac{\tan\left(x+h\right)^2-\tan x^2}{h}=\frac{2\,x}{\cos^2 x^2}$$

```
>function df(x) &= limit((tan((x+h)^2)-tan(x^2))/h,h,0); $df(x)// df(x) = f'(x)
```

$$\frac{2\,x}{\cos^2 x^2}$$

```
>plot2d(["f(x)","df(x)"],-50,50,color=[blue,red]):
```



Integral

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).
Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

$$\int_a^b f(x)\,dx = F(b) - F(a), \quad \text{dengan } F'(x) = f(x).$$

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak integrabel, kita tidak akan menggunakan metode ini).

```
>$showev('integrate(x^n,x))
```

```
Answering "Is n equal to -1?" with "no"
```

$$\int x^n\,dx = \frac{x^{n+1}}{n+1}$$

```
>$showev('integrate(1/(1+x),x))
```

$$\int \frac{1}{x+1}\,dx = \log\left(x+1\right)$$

```
>$showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2+1}\,dx = \arctan x$$

```
>$showev('integrate(1/sqrt(1-x^2),x))
```

$$\int \frac{1}{\sqrt{1-x^2}}\,dx = \arcsin x$$

```
>$showev('integrate(sin(x),x,0,pi))
```

$$\int_0^\pi \sin x\,dx = 2$$

```
>$showev('integrate(sin(x),x,a,b))
```

$$\int_a^b \sin x\,dx = \cos a - \cos b$$

```
>$showev('integrate(x^n,x,a,b))
```

```
Answering "Is n positive, negative or zero?" with "positive"
```

$$\int_{a}^{b}{x^{n}\;dx}=\frac{b^{n+1}}{n+1}-\frac{a^{n+1}}{n+1}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
```

$$\int {x^2\,\sqrt{2\,x+1}\;dx}=\frac{\left(2\,x+1\right)^{\frac{7}{2}}}{28}-\frac{\left(2\,x+1\right)^{\frac{5}{2}}}{10}+\frac{\left(2\,x+1\right)^{\frac{3}{2}}}{12}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
```

$$\int_{0}^{2}{x^2\,\sqrt{2\,x+1}\;dx}=\frac{2\,5^{\frac{5}{2}}}{21}-\frac{2}{105}$$

```
>$ratsimp(%)
```

$$\int_{0}^{2}{x^2\,\sqrt{2\,x+1}\;dx}=\frac{2\,5^{\frac{7}{2}}-2}{105}$$

```
>$showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
```

$$\int_{0}^{\pi^2}{\frac{\sin \left(\sqrt{x}+a\right)\,e^{\sqrt{x}}}{\sqrt{x}}\;dx}=\left(-e^{\pi}-1\right)\,\sin a+\left(e^{\pi}+1\right)\,\cos a$$

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

$$\int_{0}^{\pi^2}{\frac{\sin \left(\sqrt{x}+a\right)\,e^{\sqrt{x}}}{\sqrt{x}}\;dx}=\left(-e^{\pi}-1\right)\,\left(\sin a-\cos a\right)$$

```
>function map f(x) &= E^(-x^2)
```

```
            2
         - x
        E
```

```
>$showev('integrate(f(x),x))
```

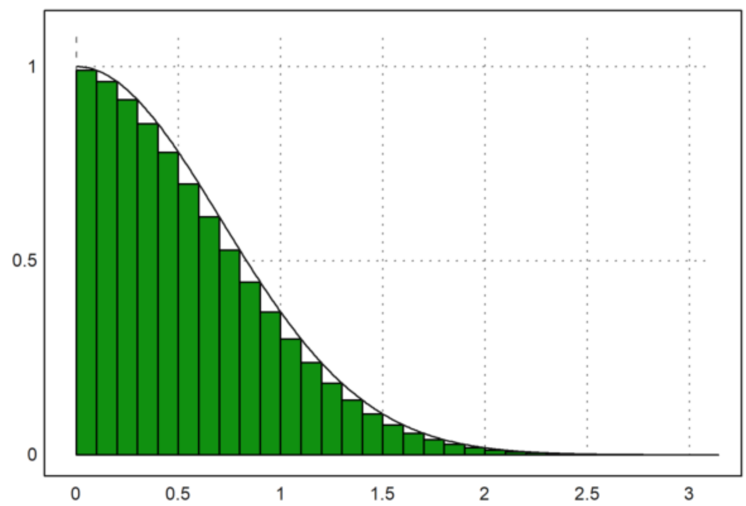
$$\int {e^ {- x^2 }\;dx}=\frac{\sqrt{\pi}\,\mathrm{erf}\left(x\right)}{2}$$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

$$erf(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}}\,dx.$$

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, ..., 3.2.

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

```
0.836219610253
```

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)
Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu

keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```

$$\int_{0}^{\infty}e^{-x^2}\;dx=\frac{\sqrt{\pi}}{2}$$

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

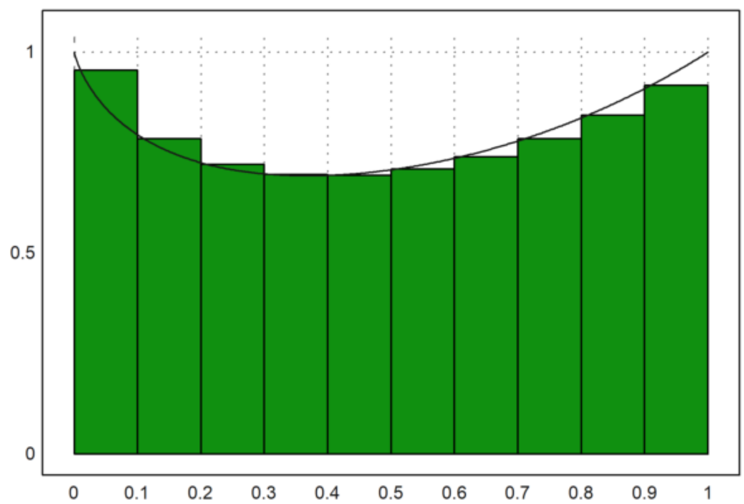
```
>function f(x) &= x^x
```

```
                                  x
                                 x
```

```
>$showev('integrate(f(x),x,0,1))
```

$$\int_{0}^{1}x^{x}\;dx=\int_{0}^{1}x^{x}\;dx$$

```
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```



Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
```

maxima: 'integrate(f(x),x,0,1) = 0.01*sum(fx[i],i,1,length(fx))
Apakah hasil tersebut cukup baik? perhatikan gambarnya. **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.)
1. Fungsi 1

```
>function f(x) &= x^2-2*x; $f(x)
```

$$x^2 - 2\,x$$

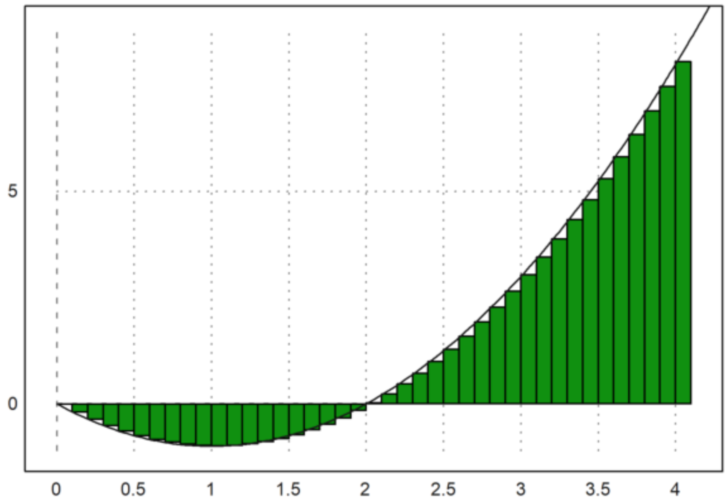
```
>$showev('integrate(f(x),x))
```

$$\int x^2 - 2\,x\,dx = \frac{x^3}{3} - x^2$$

```
>$showev('integrate(f(x),x,0,5))
```

$$\int_0^5 x^2 - 2\,x\,dx = \frac{50}{3}$$

```
>x=0:0.1:4; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,5,>add):
```




```
>
```

2. Fungsi 2

```
>function f(x) &= (cos(x))^2+1; $f(x)
```

$$\cos^2 x + 1$$

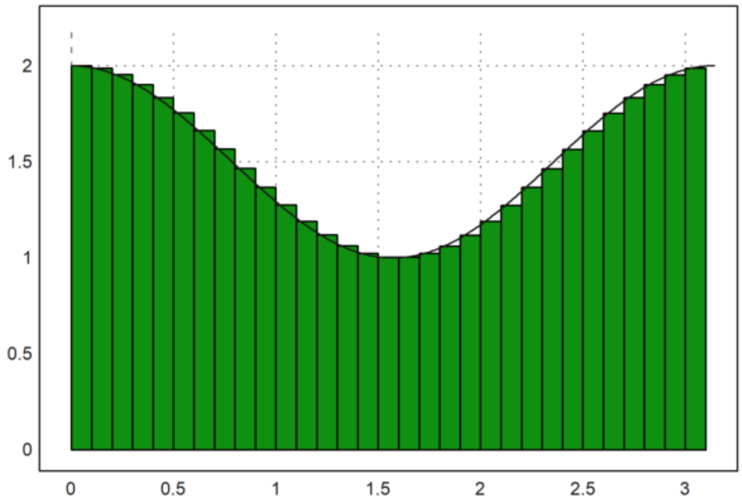
```
>$showev('integrate(f(x),x))
```

$$\int \cos^2 x + 1\, dx = \frac{\frac{\sin(2\, x)}{2} + x}{2} + x$$

```
>$showev('integrate(f(x),x,0,pi/2))
```

$$\int_0^{\frac{\pi}{2}} \cos^2 x + 1\, dx = \frac{3\,\pi}{4}$$

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.02),>bar); plot2d("f(x)",0,pi,>add):
```



3. Fungsi 3

```
>function f(x) &= sqrt(x^2+5); $f(x)
```

$$\sqrt{x^2+5}$$

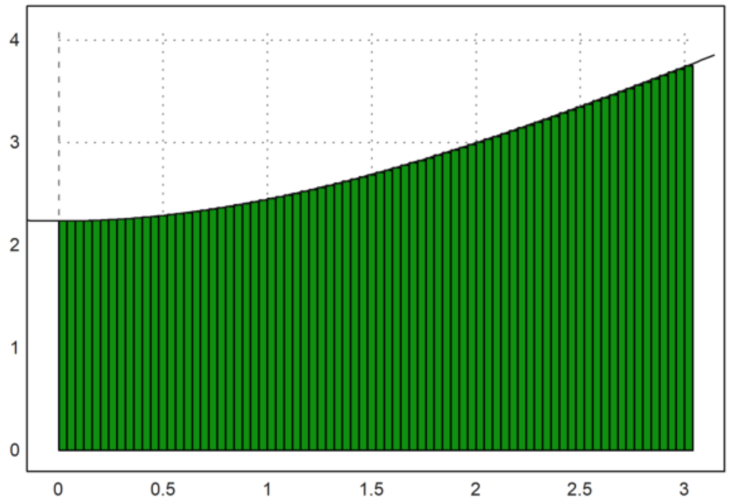
```
>$showev('integrate(f(x),x))
```

$$\int \sqrt{x^2+5}\,dx = \frac{5\,\mathrm{asinh}\left(\frac{x}{\sqrt{5}}\right)}{2} + \frac{x\,\sqrt{x^2+5}}{2}$$

```
>$showev('integrate(f(x),x,-1,1))
```

$$\int_{-1}^{1} \sqrt{x^2+5}\,dx = 5\,\mathrm{asinh}\left(\frac{1}{\sqrt{5}}\right) + \sqrt{6}$$

```
>x=0:0.04:3; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",-pi,pi,>add):
```



4. Fungsi 4

```
>function f(x) &= (x+5); $f(x)
```

$$x+5$$

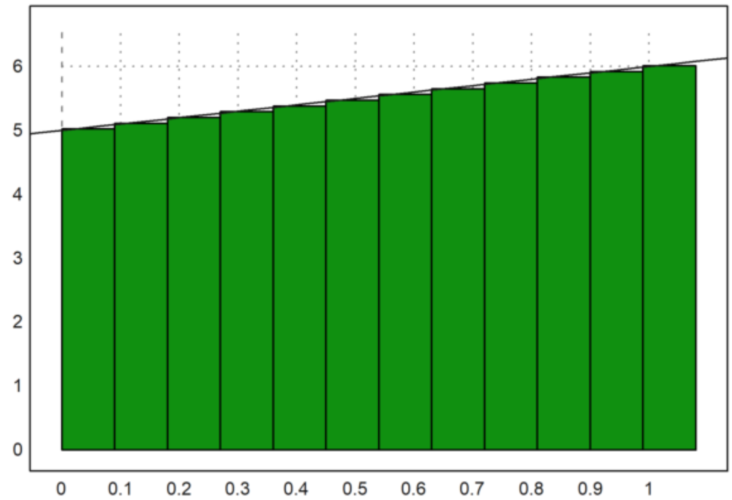
```
>$showev('integrate(f(x),x))
```

$$\int x+5\,dx=\frac{x^2}{2}+5\,x$$

```
>$showev('integrate(f(x),x,-2,5))
```

$$\int_{-2}^{5}x+5\,dx=\frac{91}{2}$$

```
>x=0:0.09:1; plot2d(x,f(x+0.02),>bar); plot2d("f(x)",-2,5,>add):
```



5. Fungsi 5

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

$$-3\,\sin^2 x$$

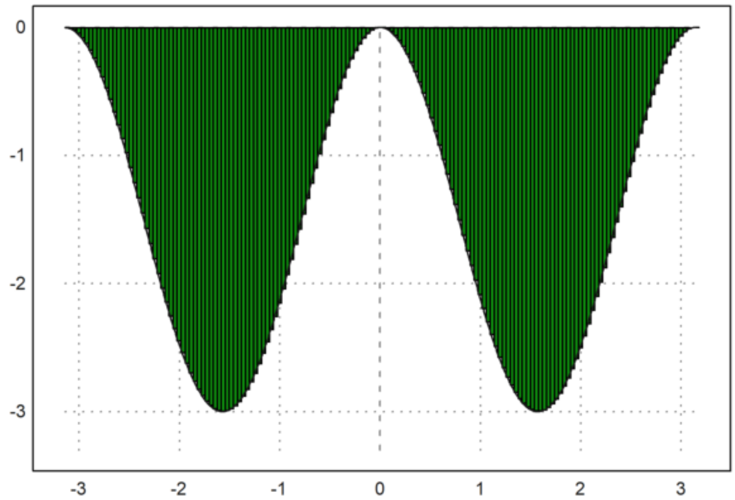
```
>$showev('integrate(f(x),x))
```

$$-3\int{\sin^2 x\;dx}=-\frac{3\,\left(x-\frac{\sin \left(2\,x\right)}{2}\right)}{2}$$

```
>$showev('integrate(f(x),x,-pi,pi))
```

$$-3\,\int_{-\pi}^{\pi}{\sin^2 x\;dx}=-3\,\pi$$

```
>x=-pi:0.04:pi; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",-pi,pi,>add):
```

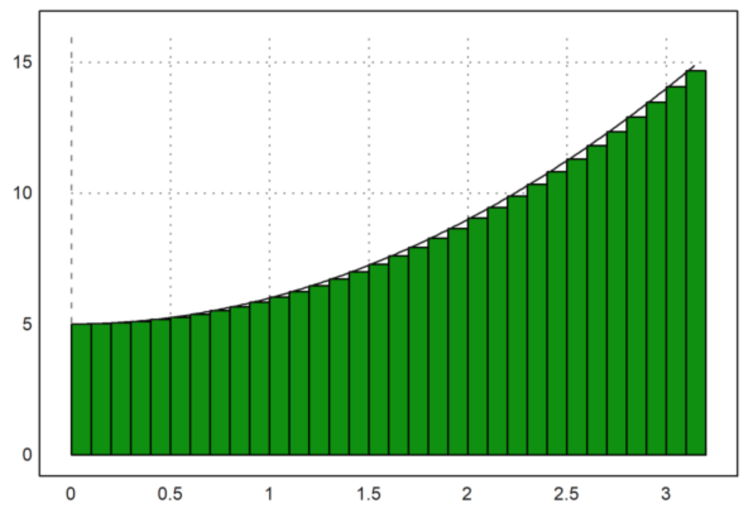


6. Fungsi 6

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
>function f(x) &= x^2+5; $f(x)
```

$$x^2 + 5$$

```
>x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,pi,>add):
```




```
>0.01*sum(f(x+0.01))
```

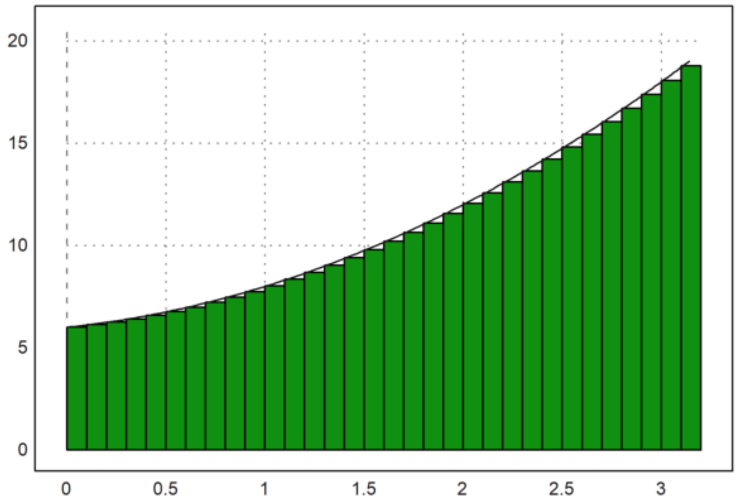
```
2.651552
```

7. Fungsi 7

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
>function f(x) &= (x^2+x+6); $f(x)
```

$$x^2 + x + 6$$

```
>x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,pi,>add):
```




```
>0.01*sum(f(x+0.01))
```

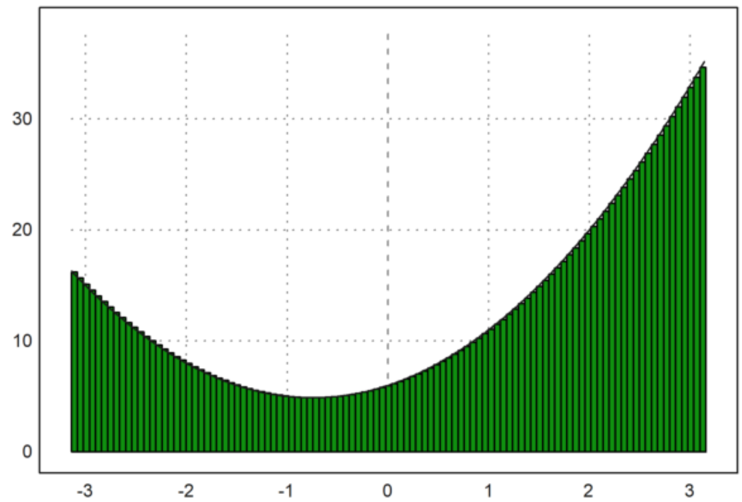
```
3.470752
```

8. Fungsi 8

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
>function f(x) &= 2*x^2+3*x+6; $f(x)
```

$$2\,x^2 + 3\,x + 6$$

```
>x=-pi:0.06:pi-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",-pi,pi,>add):
```




```
>0.01*sum(f(x+0.01))
```

```
13.2088853594
```

Luas Dibatasi 2 Kurva

1. Fungsi 1

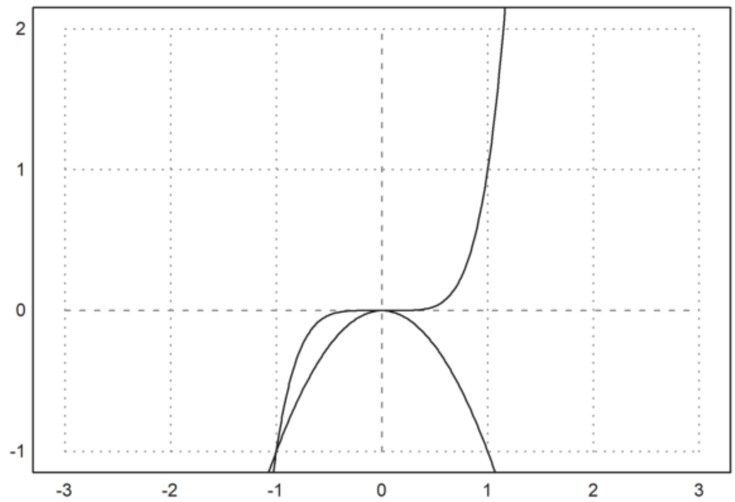
```
>function f(x) &= -x^2; $f(x)
```

$$-x^2$$

```
>function g(x) &= x^5; $g(x)
```

$$x^5$$

```
>plot2d(["-x^2","x^5"],-3,3,-1,2):
```




```
>function h(x) &= f(x)-g(x); $h(x)
```

$$-x^5 - x^2$$

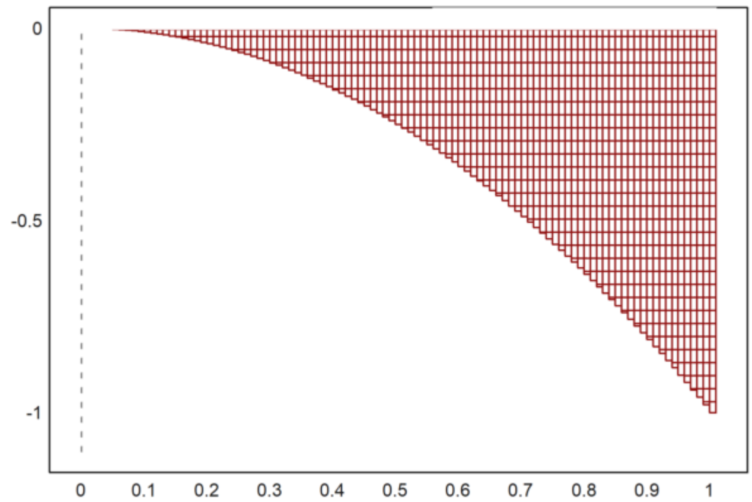
```
>$&solve(f(x)=g(x))
```

$$\left[x = \frac{1 - \sqrt{3}\,i}{2}, x = \frac{\sqrt{3}\,i + 1}{2}, x = -1, x = 0\right]$$

```
>$showev('integrate(h(x),x,0,1))
```

$$\int_{0}^{1}{-x^5 - x^2\;dx} = -\frac{1}{2}$$

```
>x=0:0.01:1; plot2d(x,f(x),>bar,>filled,style="-",fillcolor=red,>grid); plot2d(x,g(x),>bar
```



2. Fungsi 2

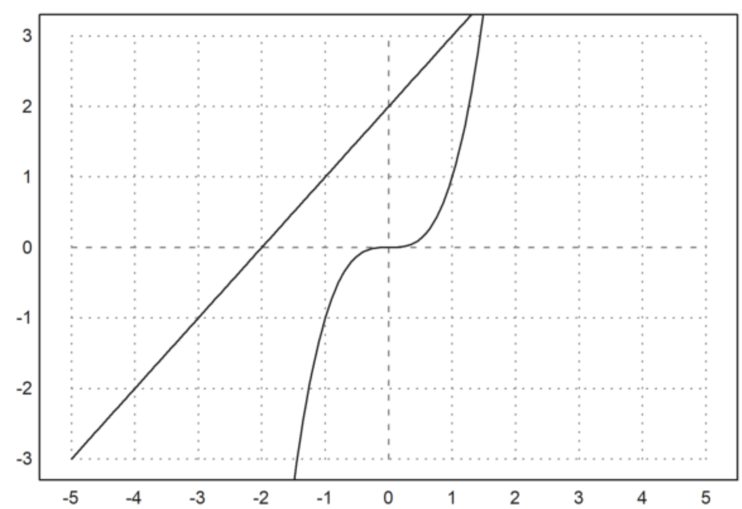
```
>function f(x) &= x+2; $f(x)
```

$$x + 2$$

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

$$x^3$$

```
>plot2d(["x+2","x^3"],-5,5,-3,3):
```



```
>function h(x) &= f(x)-g(x); $h(x)
```

$$-x^3 + x + 2$$

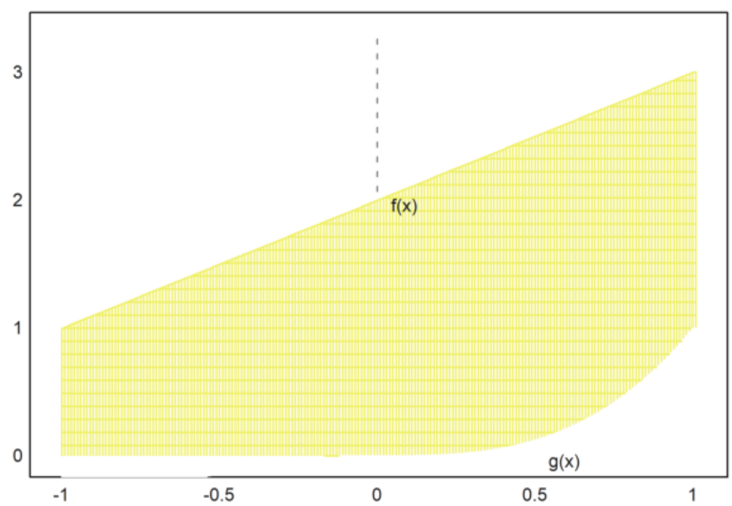
```
>$&solve(f(x)=g(x))
```

$$\left[x = \frac{\frac{\sqrt{3}\,i}{2} - \frac{1}{2}}{3\left(\frac{\sqrt{26}}{3^{\frac{3}{2}}} + 1\right)^{\frac{1}{3}}} + \left(\frac{\sqrt{26}}{3^{\frac{3}{2}}} + 1\right)^{\frac{1}{3}}\left(-\frac{\sqrt{3}\,i}{2} - \frac{1}{2}\right), x = \left(\frac{\sqrt{26}}{3^{\frac{3}{2}}} + 1\right)^{\frac{1}{3}}\left(\frac{\sqrt{3}\,i}{2} - \frac{1}{2}\right) + \frac{-\frac{\sqrt{3}\,i}{2} - \frac{1}{2}}{3\left(\frac{\sqrt{26}}{3^{\frac{3}{2}}} + 1\right)^{\frac{1}{3}}}, x = \left(\frac{\sqrt{26}}{3^{\frac{3}{2}}} + 1\right)^{\frac{1}{3}} +\right.$$

```
>$showev('integrate(h(x),x,-1,1))
```

$$\int_{-1}^{1} -x^3 + x + 2\,dx = 4$$

```
>x=-1:0.01:1; plot2d(x,f(x),>bar,>filled,style="-",fillcolor=yellow,>grid); plot2d(x,g(x),
```



Volume Benda Putar

Menghitung volume hasil perputaran kurva

$$m(x) = x^2 + 2$$

dari x=0 sampai x=1. Diputar terhadap sumbu-x.
Jawab:

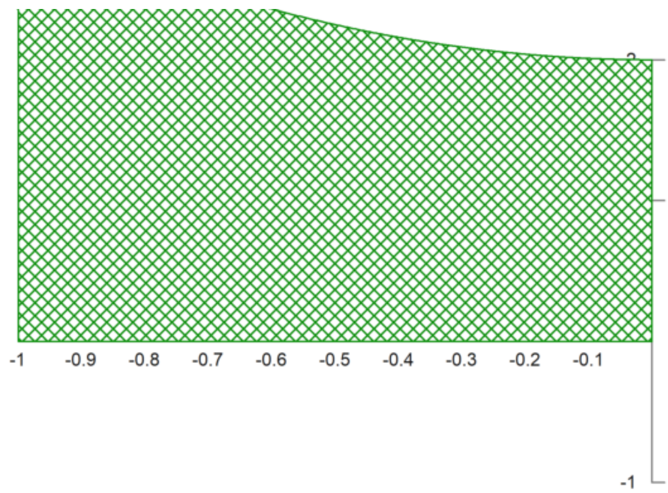
```
>function m(x) &= x^2+2; $m(x)
```

$$x^2 + 2$$

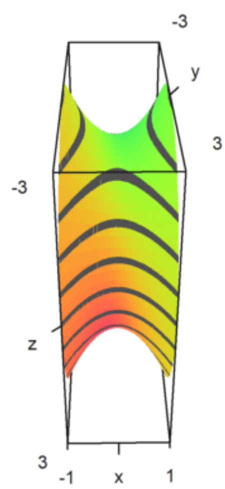
```
>$showev('integrate(m(x),x,-1,1))
```

$$\int_{-1}^{1} x^2 + 2\,dx = \frac{14}{3}$$

```
>plot2d("m(x)",-1,0,-1,2,grid=7,>filled, style="/\"):
```




```
>plot3d("m(x)",-1,1,-1,1,>rotate,angle=6.3,>hue,>contour,color=redgreen,height=10):
```



Barisan dan Deret

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh pengguanaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan (menggunakan titik dua ":");
- menggunakan perintah "sequence" dan rumus barisan (suku ke -n);
- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create_list" atau "makelist" untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana
```

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

```
>1:2:30
```

```
[1,  3,  5,  7,  9,  11,  13,  15,  17,  19,  21,  23,  25,  27,  29]
```

```
>sum(1:2:30), sum(1/(1:2:30))
```

```
225
2.33587263431
```

```
>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:menghitung deret
```

$$\sum_{k=1}^{n}{k}=\frac{n\,\left(n+1\right)}{2}$$

```
>$'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))
```

$$\sum_{k=0}^{\infty }{\frac{1}{3^{k}+k}}=\sum_{k=0}^{\infty }{\frac{1}{3^{k}+k}}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai e
```

$$\sum_{x=1}^{\infty}{\frac{1}{x^2}}=\frac{\pi^2}{6}$$

```
>$'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty}{\frac{\left(-1\right)^{k-1}}{k}}=-\sum_{x=1}^{\infty}{\frac{\left(-1\right)^{x}}{x}}$$

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)
```

$$\sum_{k=1}^{\infty}{\frac{\left(-1\right)^{k}}{2\,k-1}}=\sum_{k=1}^{\infty}{\frac{\left(-1\right)^{k}}{2\,k-1}}$$

```
>$ev(sum(1/n!, n, 0, inf),simpsum=true)
```

$$\sum_{n=0}^{\infty}{\frac{1}{n!}}$$

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
>&assume(abs(x)<1); $'sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forge
```

$$a\,\sum_{k=0}^{\infty}{x^{k}}=\frac{a}{1-x}$$

Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1.

Deret Taylor

Deret Taylor suatu fungsi f yang diferensiabel sampai tak hingga di sekitar x=a adalah:

$$f(x)=\sum_{k=0}^{\infty}\frac{(x-a)^k f^{(k)}(a)}{k!}.$$

```
>$'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai suku ke-11
```

$$e^{x}=\frac{x^{10}}{3628800}+\frac{x^9}{362880}+\frac{x^8}{40320}+\frac{x^7}{5040}+\frac{x^6}{720}+\frac{x^5}{120}+\frac{x^4}{24}+\frac{x^3}{6}+\frac{x^2}{2}+x+1$$

```
>$'log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1
```

$$\log x=x-\frac{\left(x-1\right)^{10}}{10}+\frac{\left(x-1\right)^9}{9}-\frac{\left(x-1\right)^8}{8}+\frac{\left(x-1\right)^7}{7}-\frac{\left(x-1\right)^6}{6}+\frac{\left(x-1\right)^5}{5}-\frac{\left(x-1\right)^4}{4}+\frac{\left(x-1\right)^3}{3}-\frac{\left(x-1\right)^2}{2}-1$$

BAB 5

MENGGUNAKAN EMT UNTUK GEOMETRI

[a4paper,10pt]article eumat

MATERI GEOMETRI KELAS MATEMATIKA E 2022

Anggota Kelompok

1. Anisah Daffa Citra Nareswari (22305141044/sub bab 1 dan 2)
2. Abigail Dianingtyas Paramitha (22305141018/sub bab 3 dan 10)
3. Aulina Mafazaturrahmah (22305144043/sub bab 4)
4. Shinta Nailil Mufida Rahayu (22305141006/sub bab 5)
5. Rizqi Amallia Rahma Dani (22305141035/sub bab 6 dan 7)
6. Muhammad Nurcahyo Eko Saputra (22305141043/sub bab 8 dan 11)
7. Hilyatun Nafida Badari (22305144035/sub bab 9)

Materi 1

Memangg

Program "geometry.e" untuk Menggunakan Perintah-perintah atau Fungsi-fungsi Geometri

Pada Euler kita dapat melakukan visualisasi dan perhitungan geometri secara numerik dan secara analitik dengan menggunakan Maxima.
Untuk melakukan visualisasi dan perhitungan geometri, kita harus memanggil program "geometry.e" terlebih dahulu agar perhitungan dan visualisasi dapat tereksekusi.

```
>load geometry
```

```
Numerical and symbolic geometry.
```

Terdapat fungsi-fungsi geometri pada Euler sebagai berikut:

Fungsi-fungsi Geometri

Berikut adalah fungsi-fungsi untuk menggambar objek geometri:
Fungsi yang digunakan untuk membuat bidang koordinat adalah sebagai berikut:

```
setPlotRange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang
```

koordinat. Keterangan:

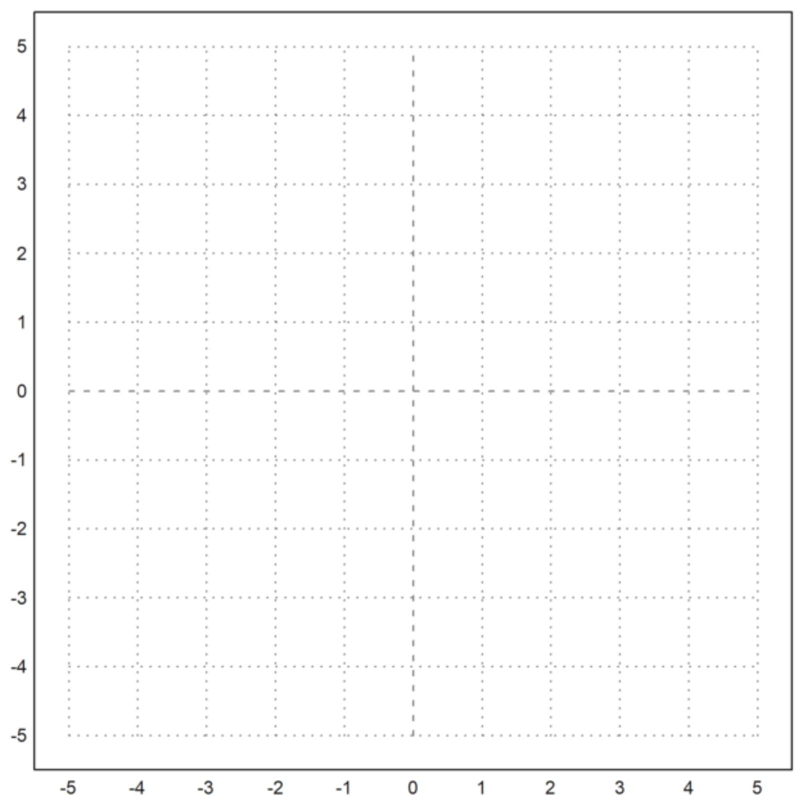
```
   x1 adalah batas terkecil sumbu-x
   x2 adalah batas terbesar sumbu-x
   y1 adalah batas terkecil sumbu-y
   y2 adalah batas terbesar sumbu-y
setPlotRange(m): pusat bidang koordinat (0,0) dan batas-batas
```

sumbu-x dan sumbu-y adalah -m sampai dengan m.

```
>setPlotRange(-4,5,0,5)
```

```
[-4,  5,  0,  5]
```

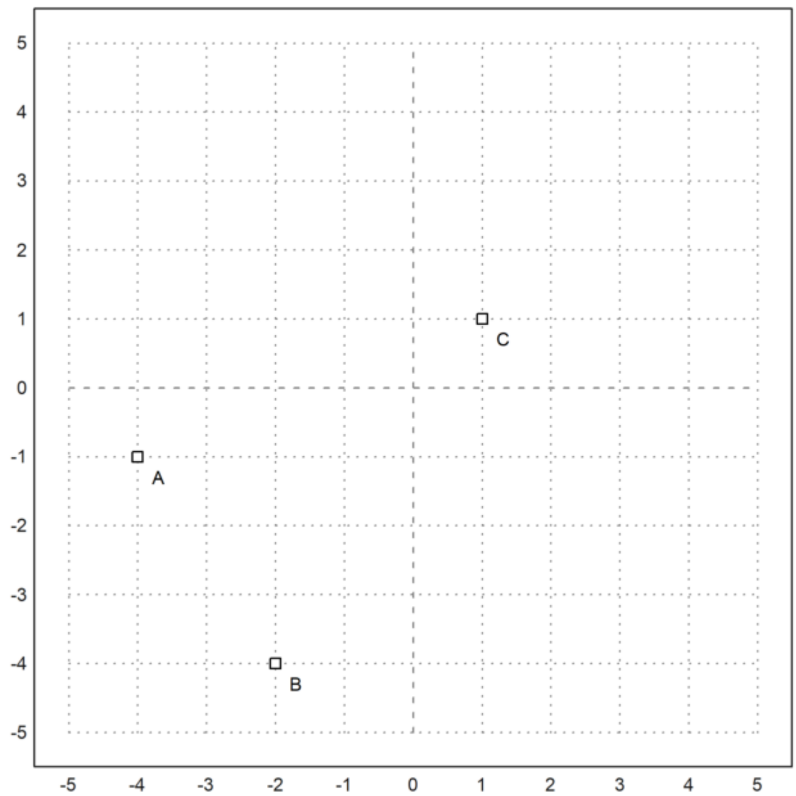
```
>setPlotRange(5):
```



```
plotPoint (A, "A"): menggambarkan titik P dan diberi label "A".
```

Sebelum itu, kita harus mendefinisikan A sebagai suatu titik koordinat terlebih dahulu atau dapat dituliskan dengan A=[x,y]

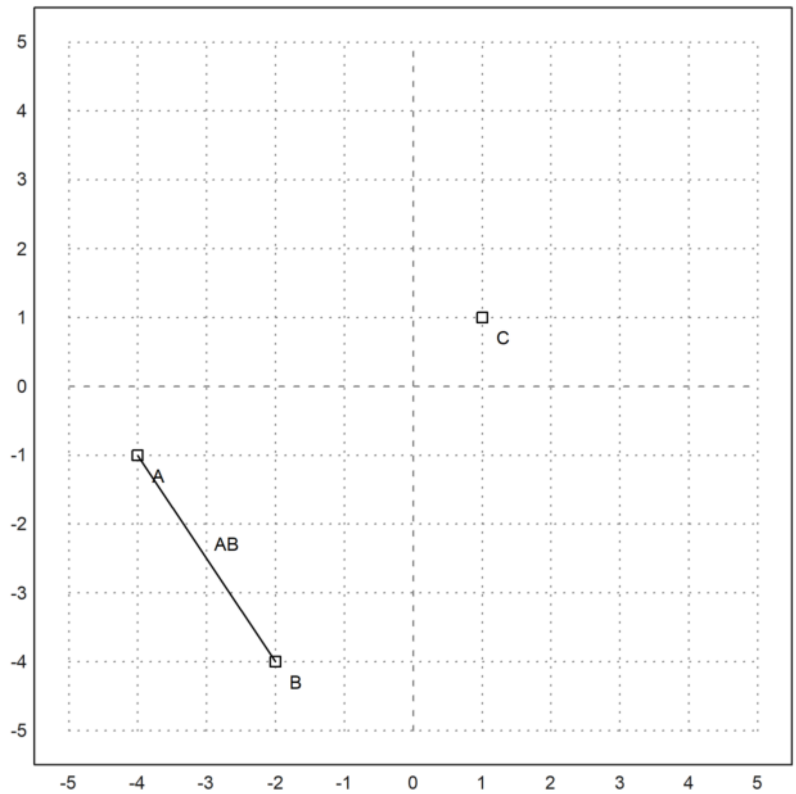
```
>A=[-4,-1]; B=[-2,-4]; C=[1,1];
>plotPoint(A, "A"); plotPoint(B, "B"); plotPoint(C, "C"):
```




```
plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label
```

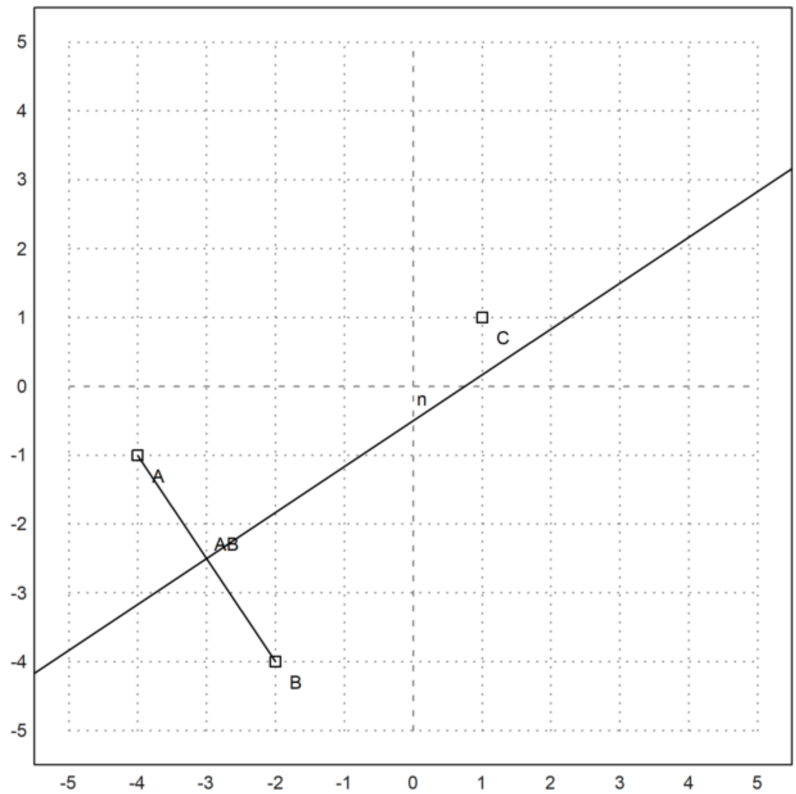
"AB" sejauh nilai d. Sebelum menggambar ruas garis AB, kita harus mendefinisikan titik A dan B terlebih dahulu.

```
>plotSegment(A,B, "AB", 30):
```



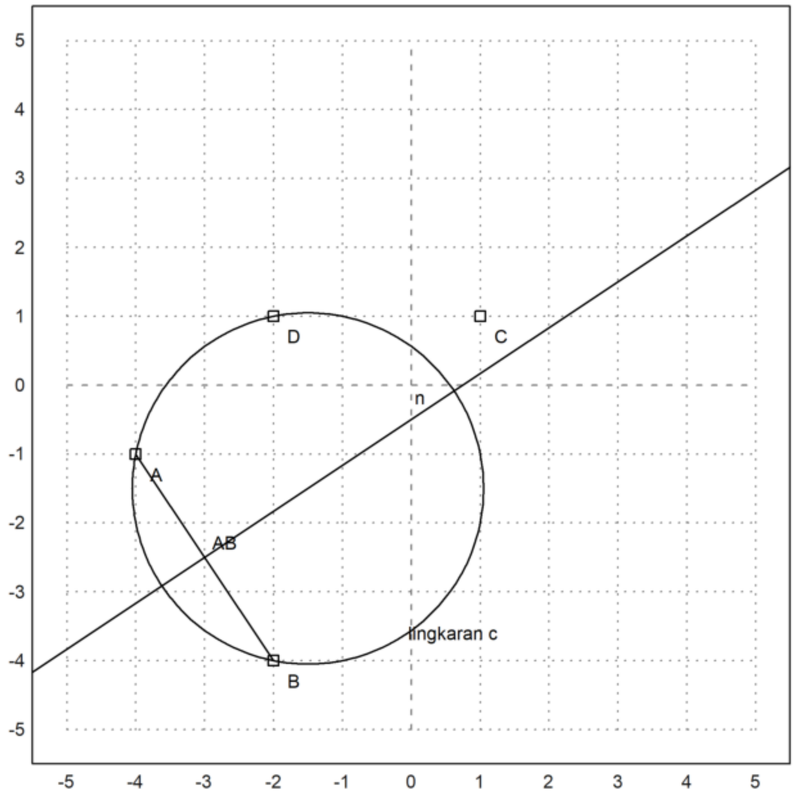
plotLine (n, "n", d): menggambar garis n diberi label "n" sejauh d

```
>n=middlePerpendicular(A,B);
>plotLine(n, "n", 15):
```



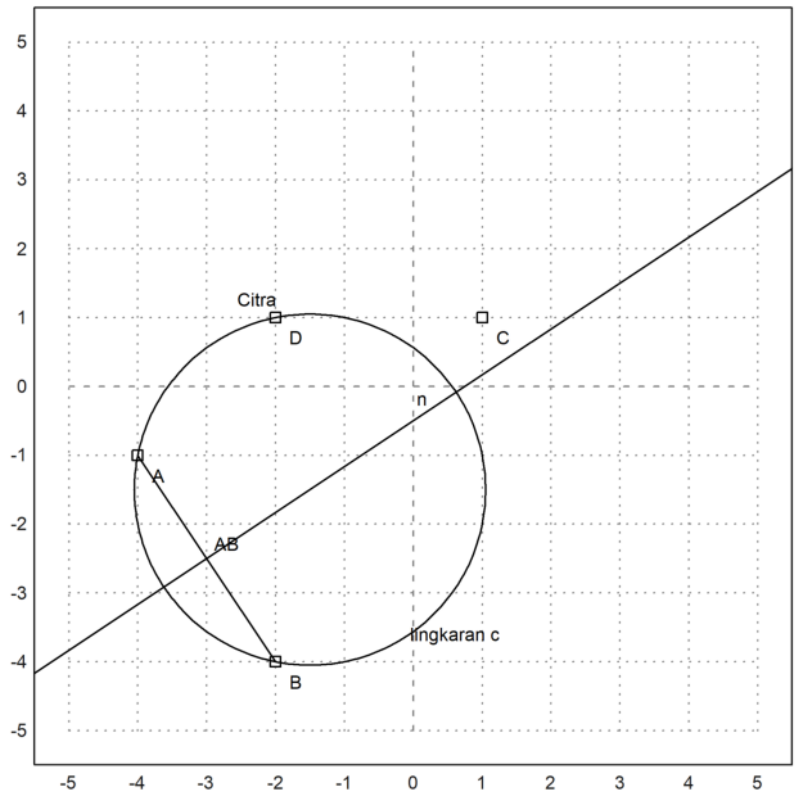
plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"

```
>D=[-2,1]; plotPoint(D,"D");
>c=circleThrough(A,B,D);
>plotCircle(c,"lingkaran c"):
```



plotLabel (label, P, V, d): menuliskan label pada posisi P

```
>plotLabel("Citra", D, [-2,2], 30):
```



Fungsi-fungsi Geometri Analitik (numerik maupun simbolik)

norm(A-B): jarak antara titik A dan titik B
turn(v, phi): memutar vektor v sejauh phi
turnLeft(v): memutar vektor v ke kiri
turnRight(v): memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang vektor v dan w.
lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh. ax+by=c.
lineWithDirection(A,v): garis melalui A searah vektor v

```
>norm(A-C)
```

```
5.38516480713
```

```
>sqrt(29)
```

```
5.38516480713
```

```
>lineThrough(A,D)
```

```
[-2,  2,  6]
```

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

```
[-1,  -1]
```

```
>lineWithDirection(A,v)
```

```
[1,  -1,  -3]
```

getLineDirection(v): vektor arah (gradien) garis v
getNormal(v): vektor normal (tegak lurus) garis v

```
>getLineDirection(v)
```

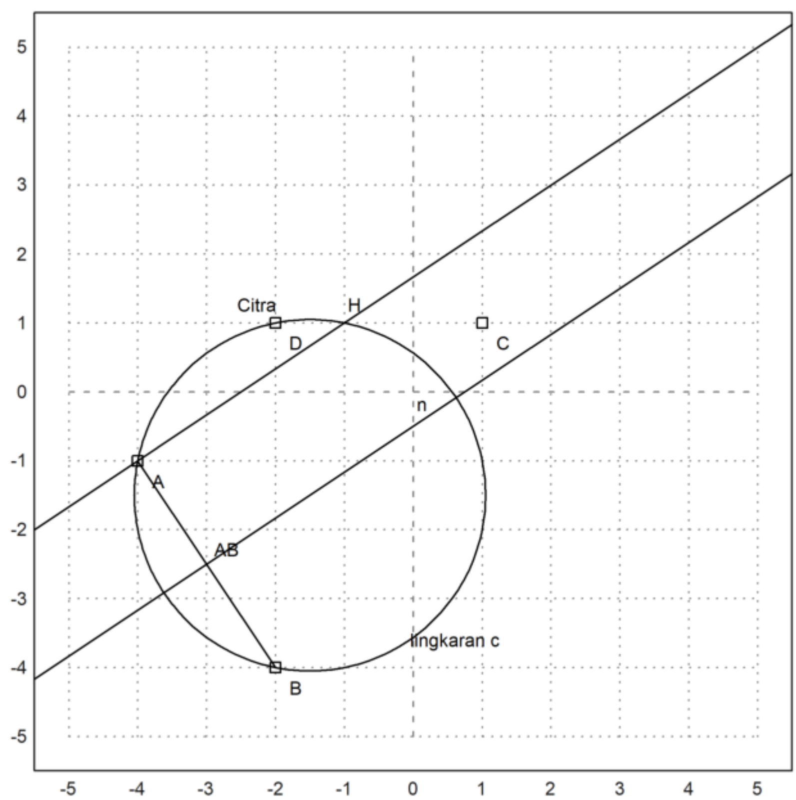
```
[-1,  1]
```

```
>getNormal(v)
```

```
-1
```

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

```
>h=perpendicular(A, lineThrough(A,B)); // garis melalui A dan tegak lurus AB
>plotLine(h, "H", 10):
```



```
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
```

```
>distance(A, C)
```

```
5.38516480713
```

```
>distanceSquared(A, C)
```

```
29
```

```
>quadrance(A, C)
```

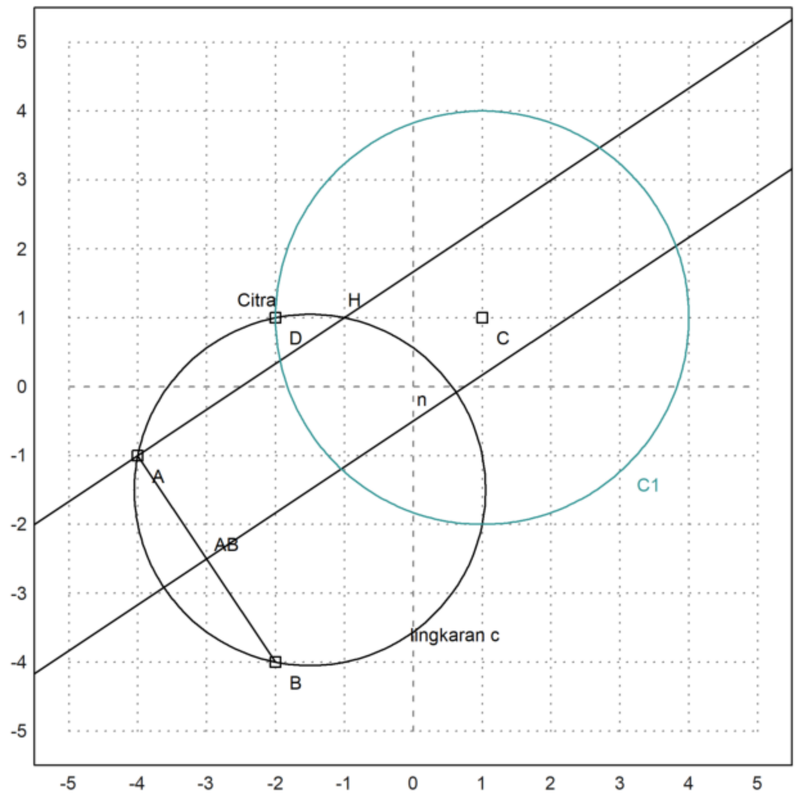
```
29
```

```
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
```

```
>areaTriangle(A, B, C)
```

```
9.5
```

```
>c1=circleWithCenter(C, 3);
>color(5); plotCircle(c1, "C1"):
```




```
getCircleCenter(c):  pusat lingkaran c
getCircleRadius(c):  jari-jari lingkaran c
circleThrough(A,B,C):  lingkaran melalui A, B, C
```

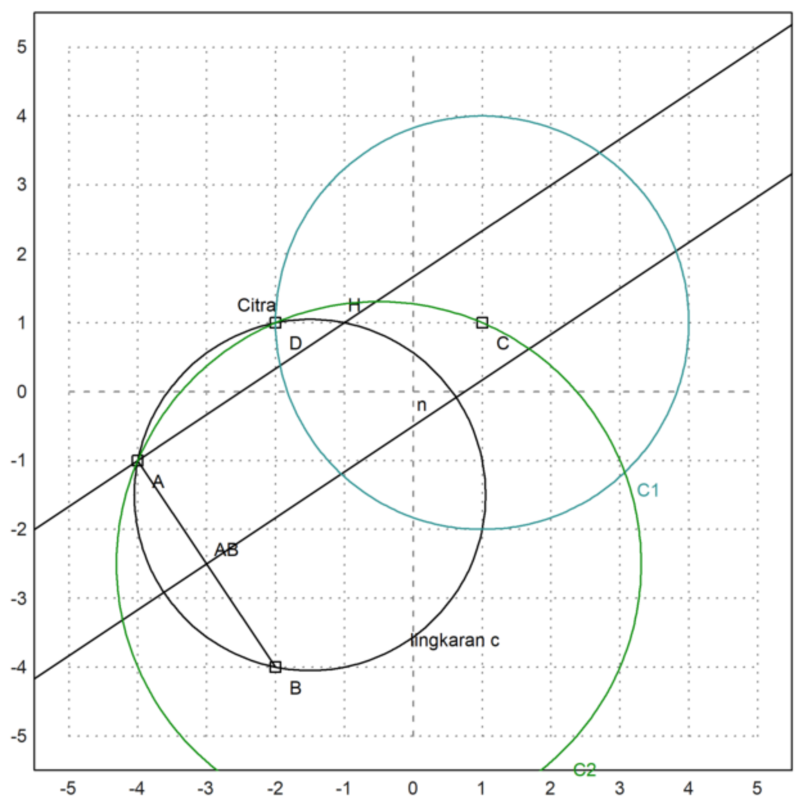
```
>getCircleCenter(c1)
```

```
[1,  1]
```

```
>getCircleRadius(c1)
```

```
3
```

```
>c2=circleThrough(A, D, C);
>color(3); plotCircle(c2, "C2"):
```

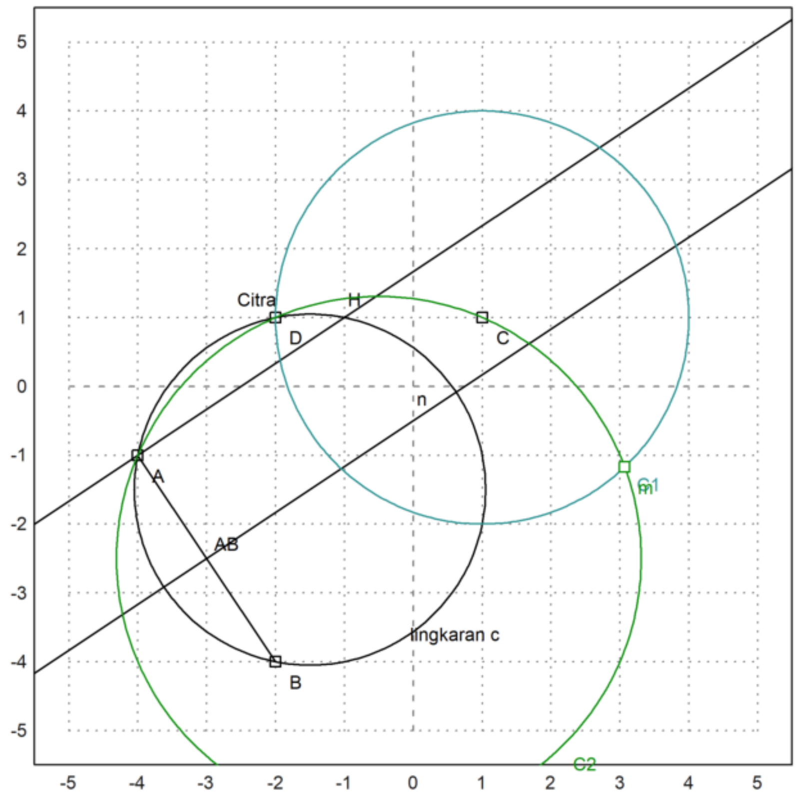



```
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan lingkran c
circleCircleIntersections (c1, c2):  titik potong lingkaran c1 dan
```

c2

```
planeThrough(A, B, C):  bidang melalui titik A, B, C
```

```
>m=circleCircleIntersections(c1,c2);
>plotPoint(m,"m"):
```



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. Bentuk hesse garis merupakan cara untuk menyatakan garis dalam bentuk matematika yang lebih umum.

```
>M &= [1,1]; N &= [2,2];
>g &= lineThrough(M,N)
```

[- 1, 1, 0]

```
>&getLineEquation(g,x,y)
```

y - x = 0

```
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.

```
>a &= 8^2; b &= 15^2; c &= 17^2; &a+b=c
```

```
289 = 289
```

```
>&spread(a,b,c)
```

```
64
---
289
```

Materi 2 ** Mengatur Luas Bidang Koordinat untuk Menggambar Objek-objek Geometri

Untuk menggambarkan objek geometri yang diinginkan pada Euler, kita harus menggambarkan luas bidang koordinat dengan menentukan rentang sumbu-sumbu koordinatnya terlebih dahulu. Seluruh objek geometri yang kita gambarkan, nantinya dapat muncul pada satu bidang koordinat yang sama sampai kita mendefinisikan bidang koordinat yang baru.
Untuk mengatur luas bidang koordinat dapat menggunakan fungsi setPlotRange.

setPlotRange(n):

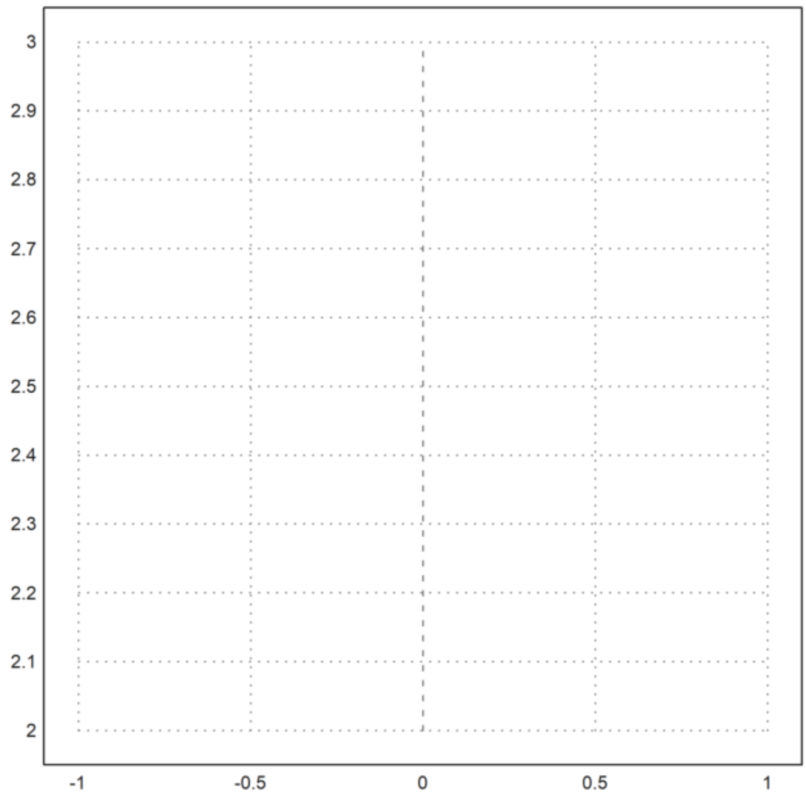
merupakan fungsi yang menggambarkan luas bidang koordinat dengan pusat bidang koordinat (0,0) dan batas-batas sumbu-x dan sumbu-y adalah -n sampai dengan n.

setPlotRange(x1,x2,y1,y2):

Keterangan:

```
x1 adalah batas terkecil sumbu-x
x2 adalah batas terbesar sumbu-x
y1 adalah batas terkecil sumbu-y
y2 adalah batas terbesar sumbu-y
```

```
>setPlotRange(5);
>setPlotRange(-1,1,2,3):
```



Bonus alias Latihan!

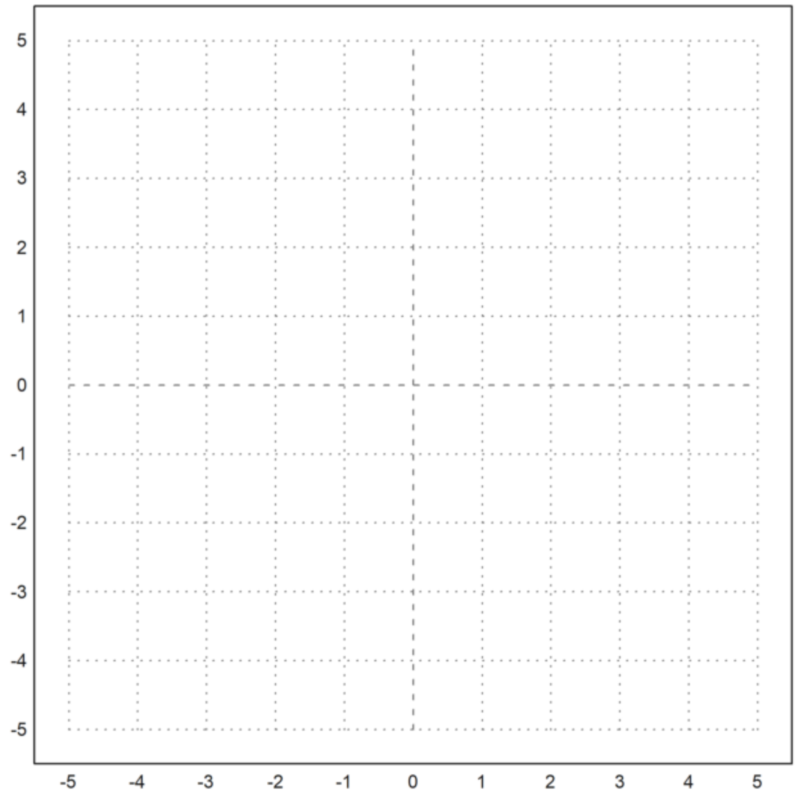
Latihan

soal berisi dari penggabungan

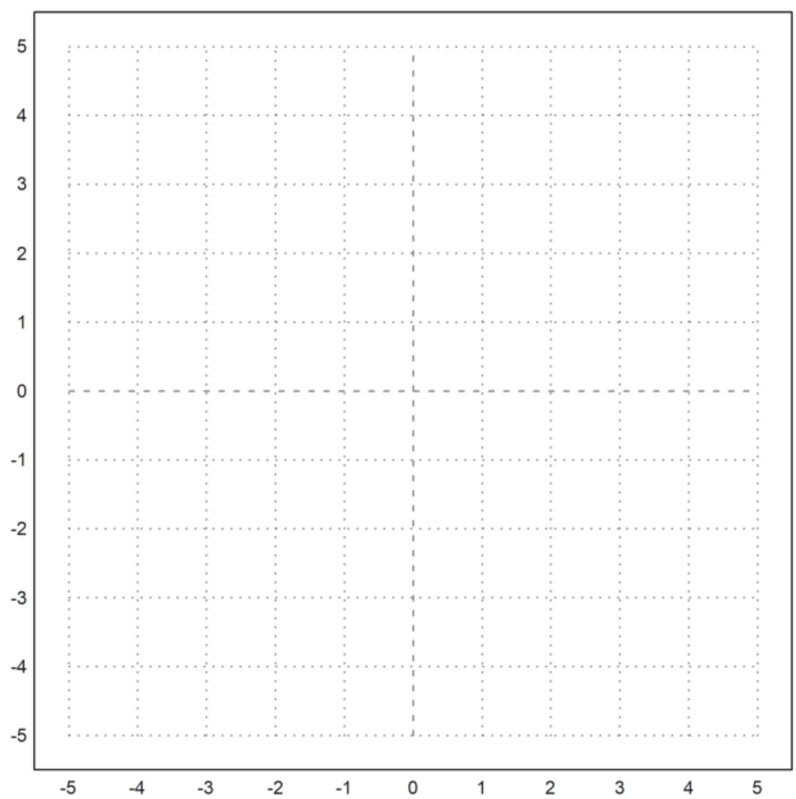
Materi 1 dan Materi 2

Kita akan menghitung luas segitiga ABC dan lingkaran luar ABC. Sebelum itu, kita perlu menggambarkan objek segitiga ABC pada bidang koordinat terlebih dahulu.
Berikut adalah langkah-langkah menggambarkan segitiga siku-siku.
Langkah 1: Menentukan luas bidang koordinat
Langkah 1: Menghitung panjang ruas garis BC sebagai alas segitiga

```
>setPlotRange(5):
```

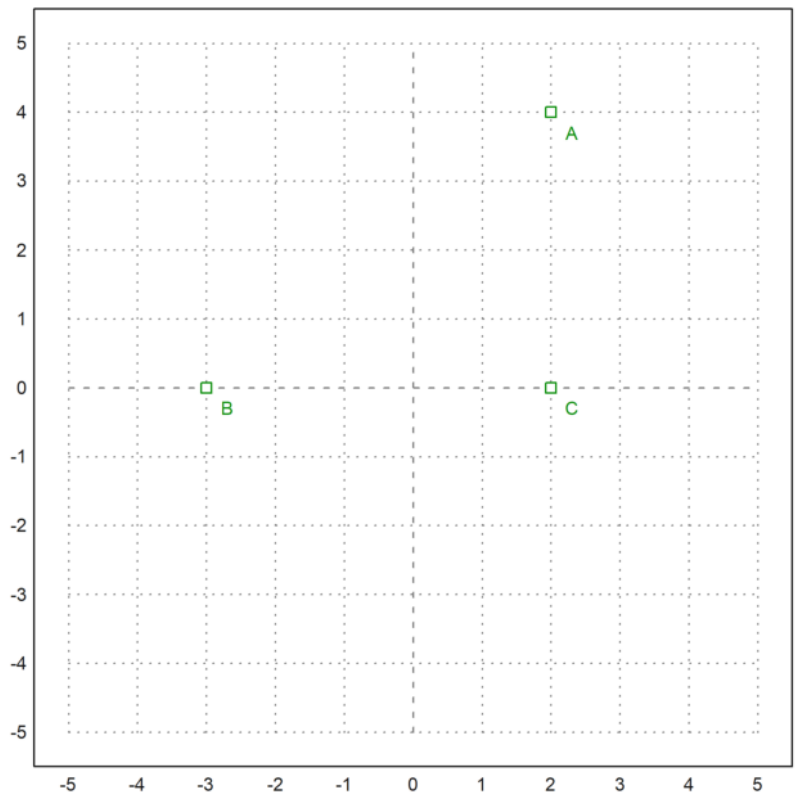



```
>setPlotRange(5):
```



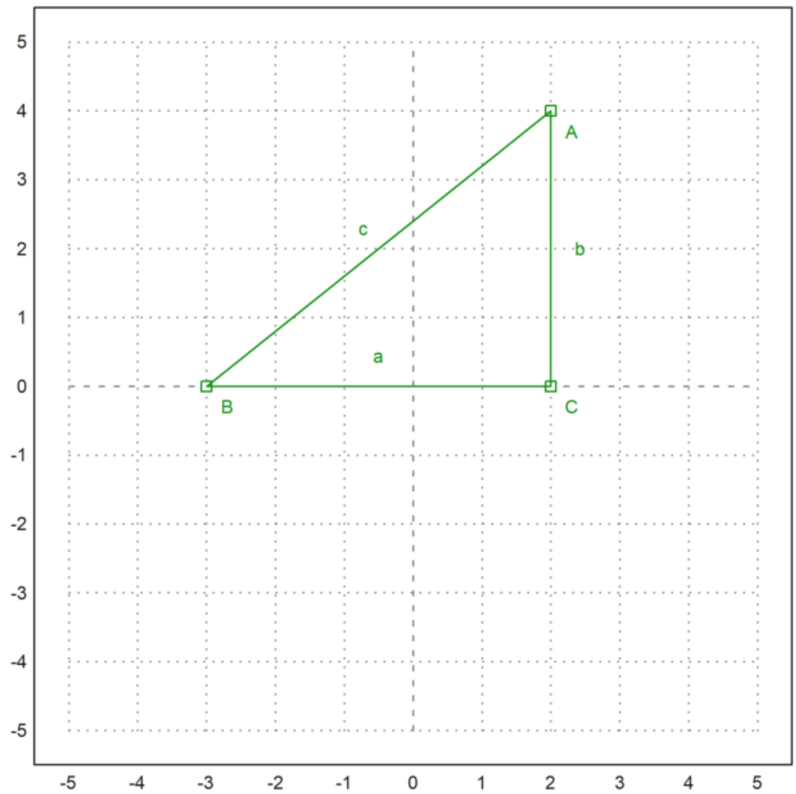
Langkah 2: Menentukan titik-titik pada bidang koordinat

```
>A=[2,4]; plotPoint(A,"A");
>B=[-3,0]; plotPoint(B,"B");
>C=[2,0]; plotPoint(C,"C"):
```



Langkah 3: Menggambarkan ruas garis pada titik A, B, dan C

```
>plotSegment(A,B,"c",-30);
>plotSegment(B,C,"a");
>plotSegment(A,C,"b"):
```



Setelah menggambarkan objek geometri yang diiingikan, kita akan menghitung luas dari segitiga siku-siku tersebut menggunakan fungsi geometri yang terdapat pada Euler.

```
>areaTriangle(A,B,C)
```

```
10
```

Untuk membuktikan hitungan menggunakan fungsi geometri Euler bernilai benar, kita akan membandingkannya dengan hasil luas segitiga siku-siku yang dihitung secara manual.
Rumus Luas Segitiga:

$$L_{\triangle ABC} = \frac{1}{2}BC.AC$$

```
>norm(B-C)
```

```
5
```

Langkah 2: Menghitung panjang ruas garis AC sebagai tinggi segitiga

```
>norm(A-C)
```

```
4
```

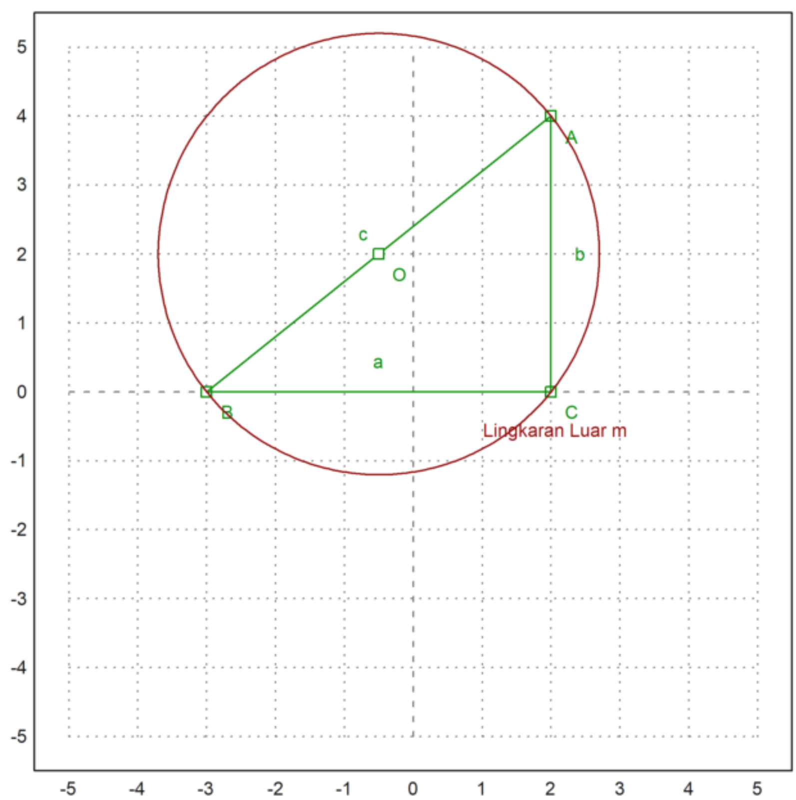
Langkah 3: Menghitung luas segitiga siku-siku menggunakan rumus luas segitiga

```
>norm(B-C)*norm(A-C)/2
```

```
10
```

Jadi, hasil dari perhitungan secara manual dan menggunakan fungsi areaTriangle(A,B,C) bernilai sama. Luas segitiga ABC bernilai 10 satuan.
Selanjutnya, kita akan menggambarkan suatu lingkaran yang mengelilingi segitiga siku-siku ABC. Lingkaran tersebut kita beri nama lingkaran luar m.

```
>m=circleThrough(A,B,C);
>R=getCircleRadius(m);
>O=getCircleCenter(m);
>plotPoint(O,"O");
>color(2); plotCircle(m,"Lingkaran Luar m"):
```



Untuk menghitung luas lingkaran, kita akan mencari jari-jari dari lingkaran luar m.

```
>O, R
```

```
[-0.5,  2]
3.20156211872
```

Selanjutnya, kita akan menghitung luas lingkaran luar m secara manual menggunakan rumus:

$$L_m = \pi \times r^2$$

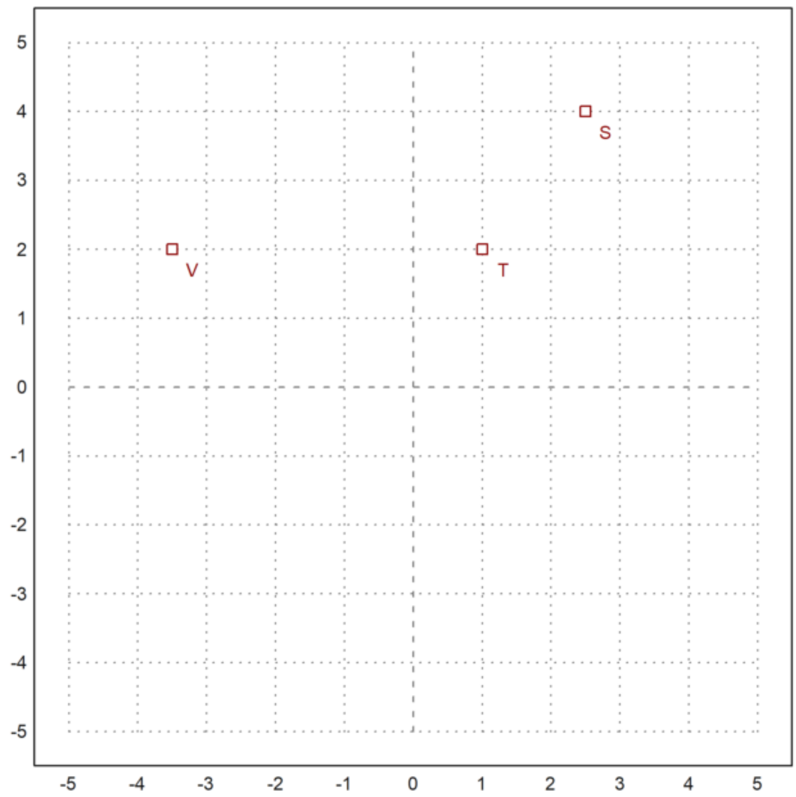
```
>pi*(3.20156211872)^2
```

```
32.2013246994
```

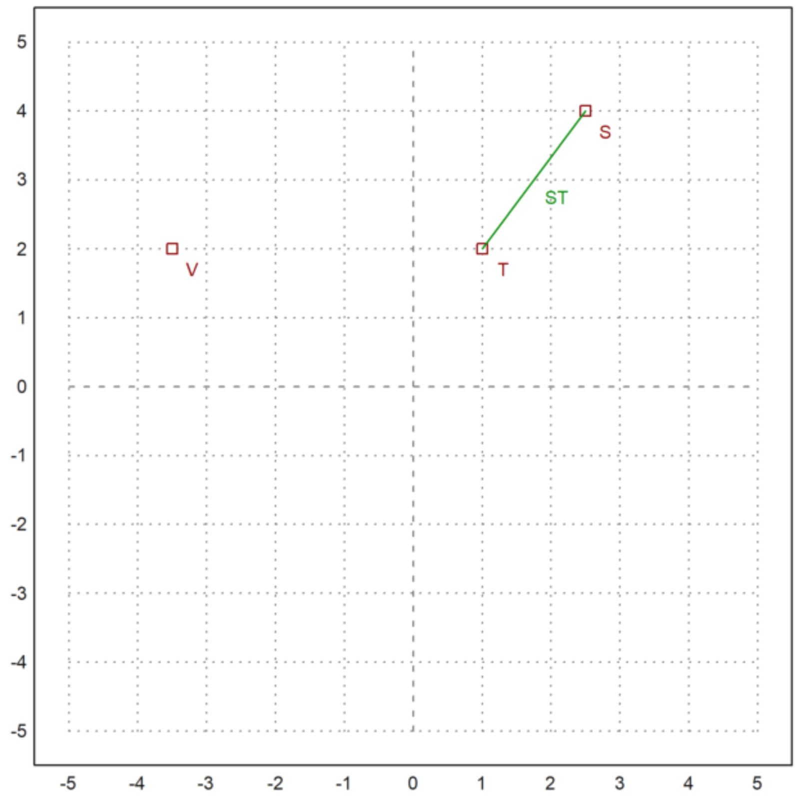
Materi 3 Menggambar titik, ruas garis, garis, segi banyak, lingkaran, ellips, parabola, hiperbola, dll.

Menggambarkan titik pada bidang koordinat.

```
>setPlotRange(-5,5,-5,5);
>S=[2.5,4]; plotPoint(S,"S");
>V=[-3.5,2]; plotPoint(V,"V");
>T=[1,2]; plotPoint(T,"T"):
```

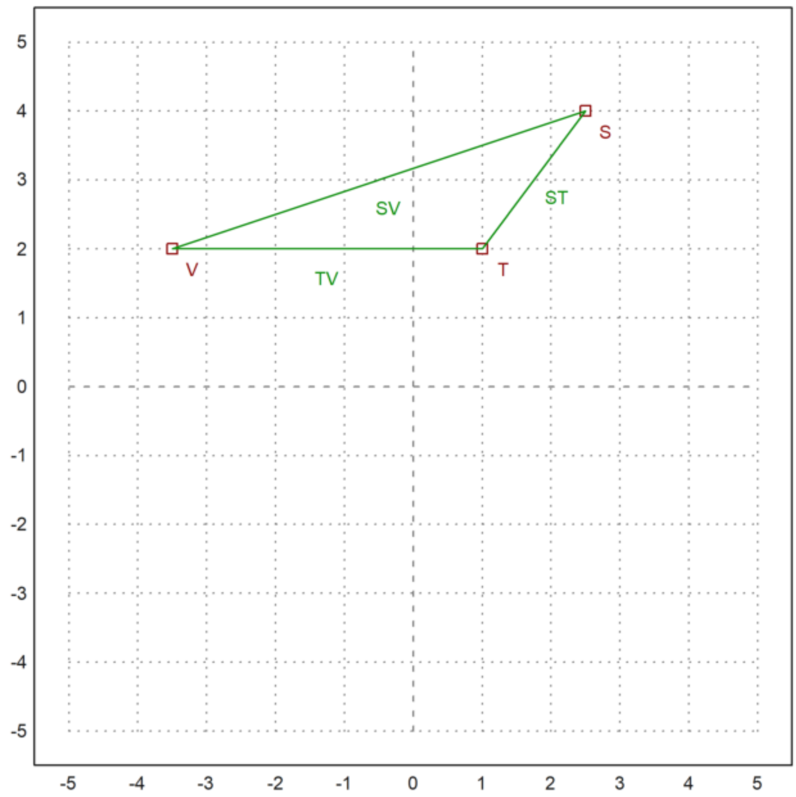



```
>color(3); plotSegment(S,T,"ST"):
```

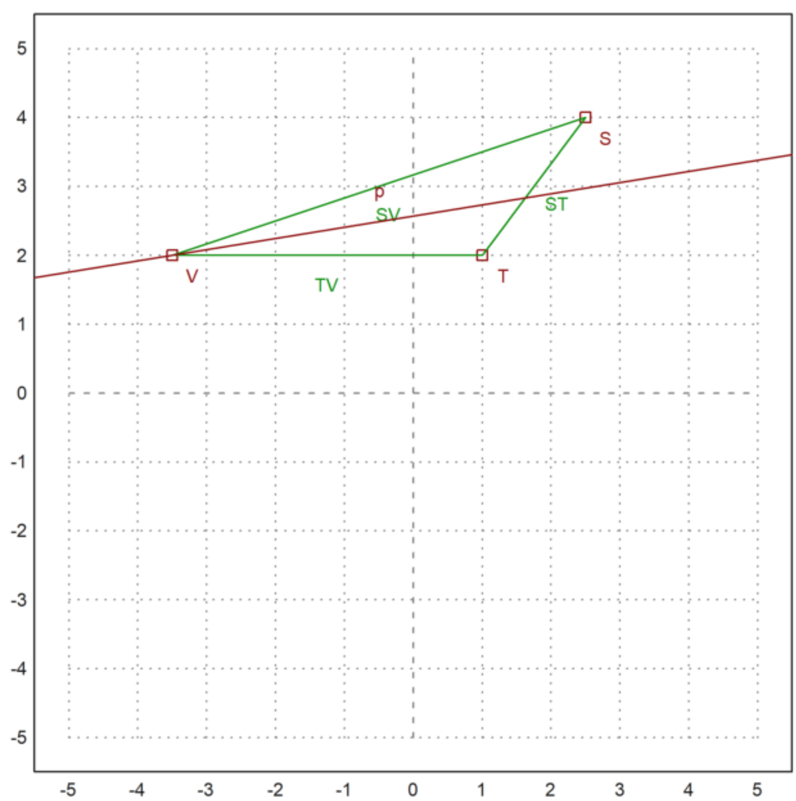


Menggambar ruas garis.

```
>color(3); plotSegment(S,V,"SV");
>color(3); plotSegment(T,V,"TV"):
```

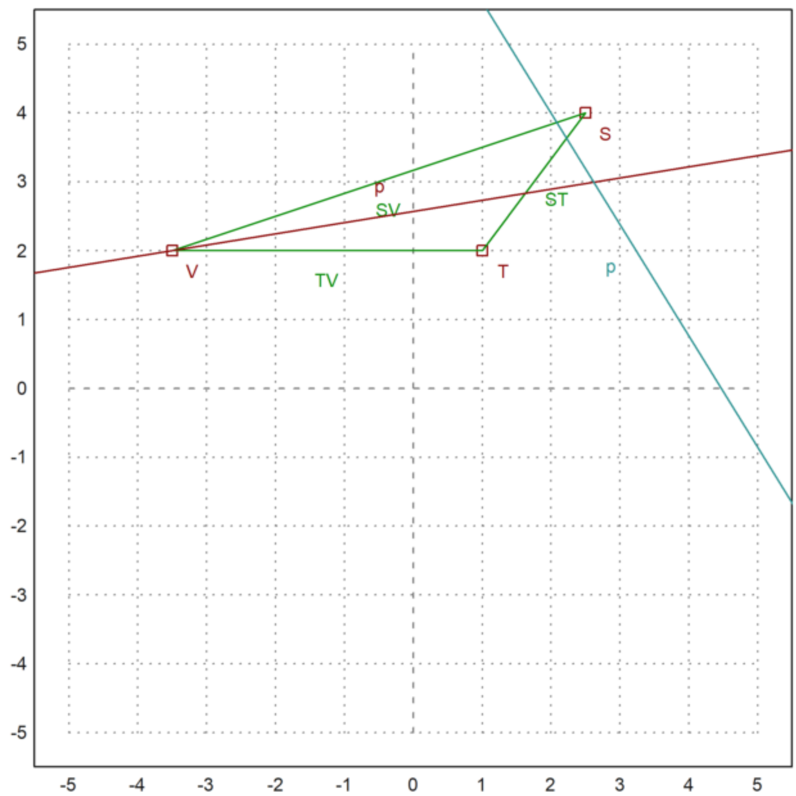



```
>u=angleBisector(S,V,T); color(2); plotLine(u):
```



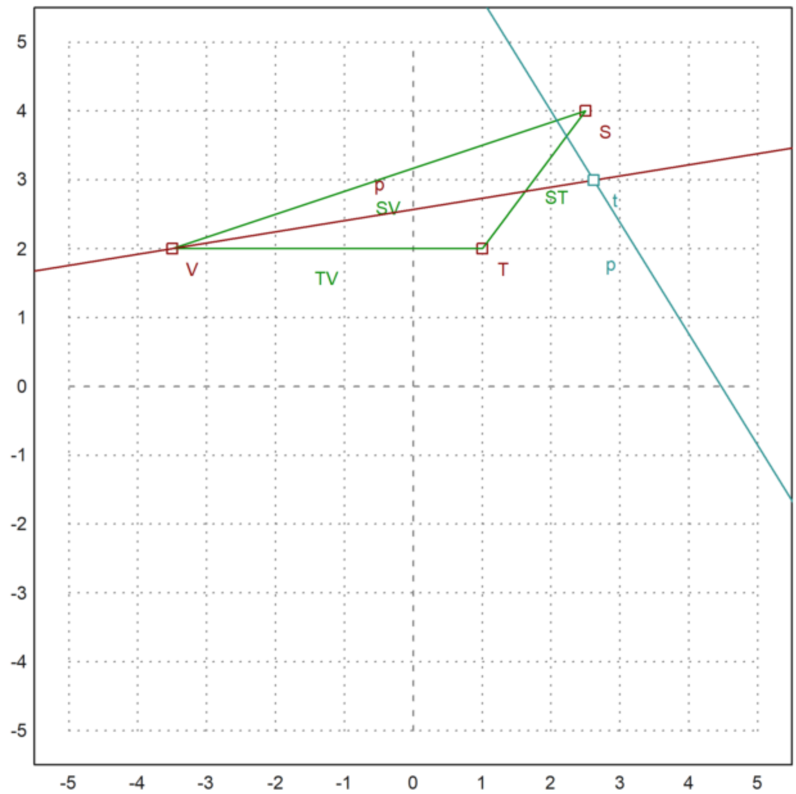
Menggambar garis.

```
>n=angleBisector(T,A,S); color(5); plotLine(n):
```



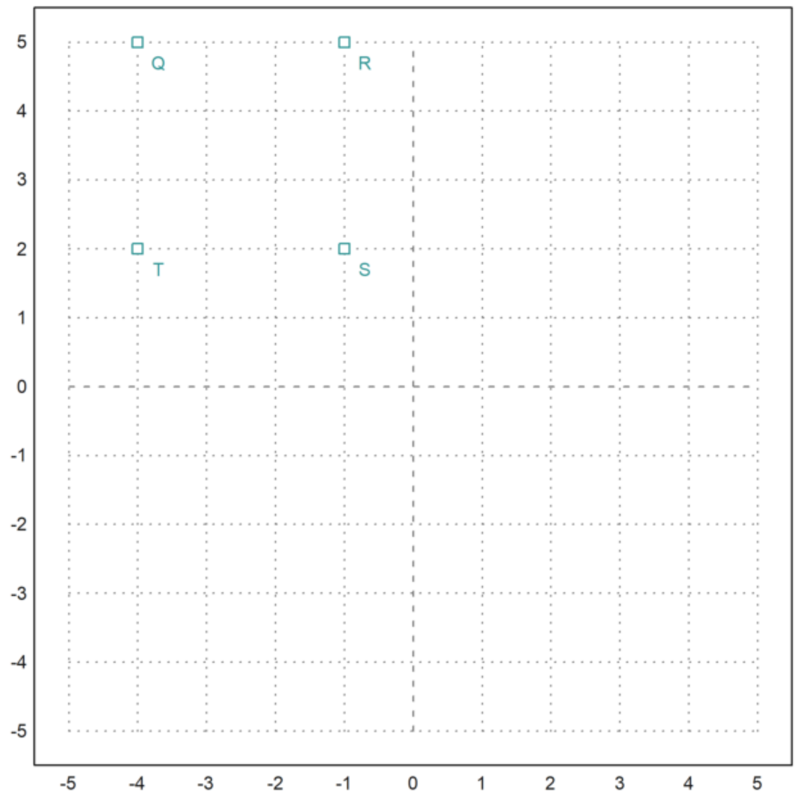
Menggambarkan titik potong garis u dan garis n.

```
>t=lineIntersection(u,n); plotPoint(t):
```



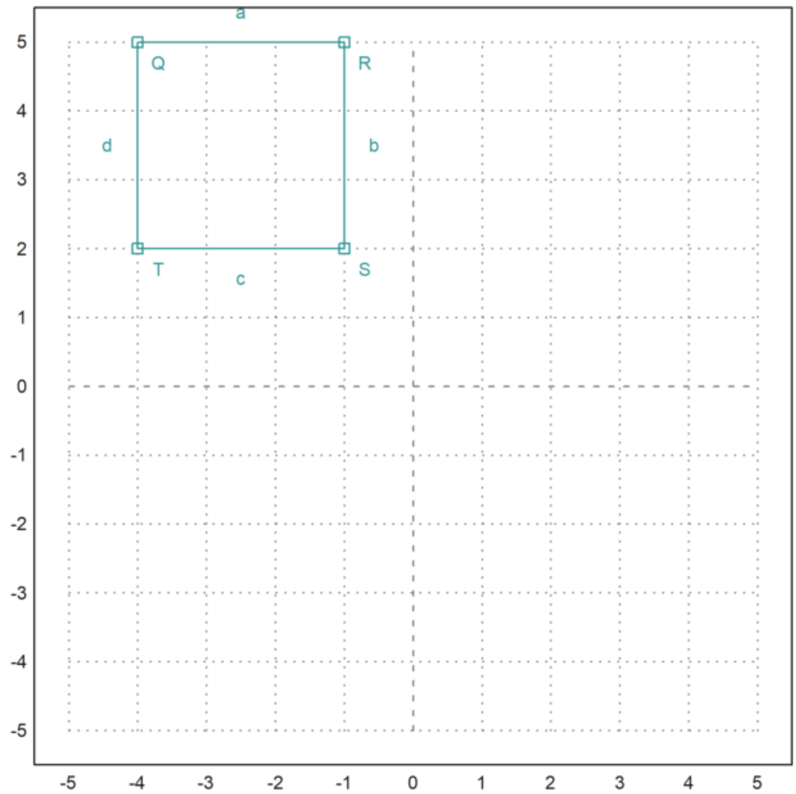
Menggambar segibanyak.
Berikut ini adalah penggambaran objek geometri berupa segi-4:
Persegi
Langkah 1: Menggambarkan titik-titik koordinat Q, R, S, T.

```
>setPlotRange(-5,5,-5,5);
>Q=[-4,5]; plotPoint(Q,"Q");
>R=[-1,5]; plotPoint(R, "R");
>S=[-1,2]; plotPoint(S, "S");
>T=[-4,2]; plotPoint(T, "T"):
```



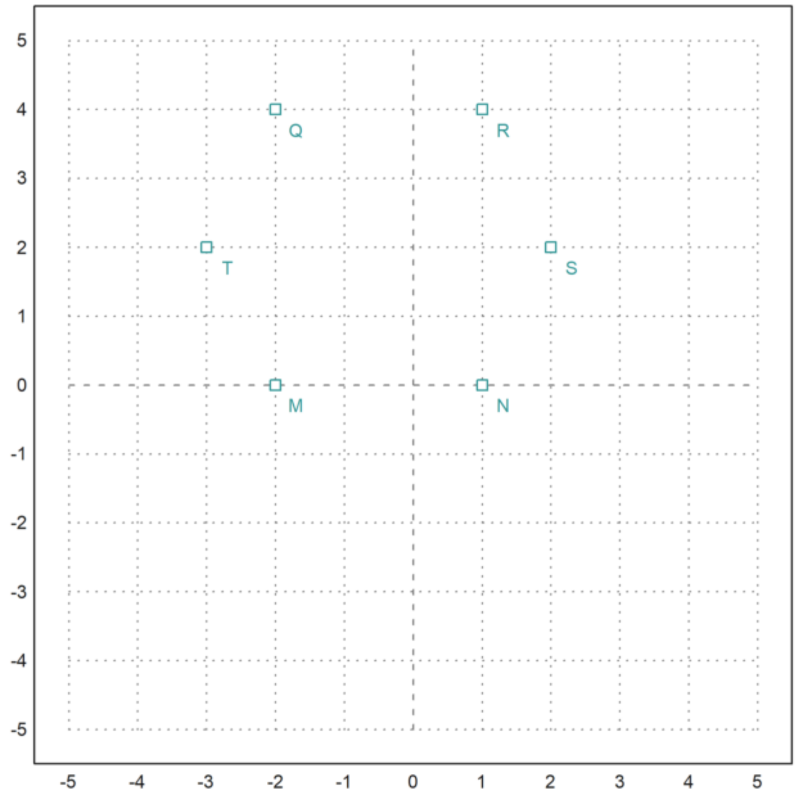
Menghubungkan titik-titik koordinat Q, R, S, T dengan ruas garis.

```
>plotSegment(Q,R,"a");
>plotSegment(R,S,"b");
>plotSegment(S,T,"c");
>plotSegment(T,Q,"d"):
```

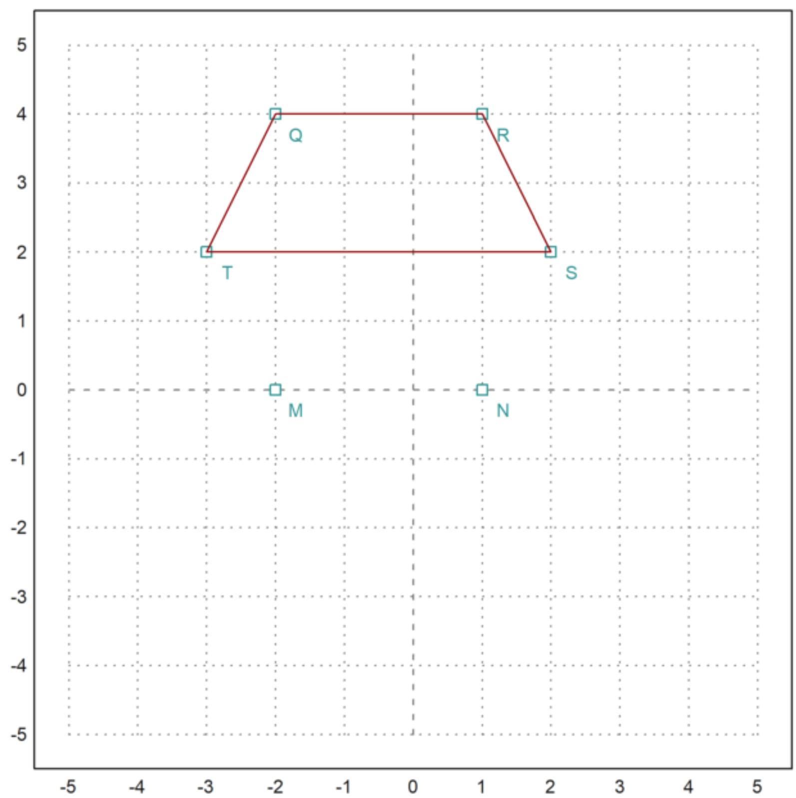


Trapesium Sama Kaki
Digambarkan enam titik, yaitu Q, R, S, T, M, N. Keenam titik ini jika dihubungkan dapat membentuk dua bangun datar, yaitu trapesium sama kaki dan segienam.

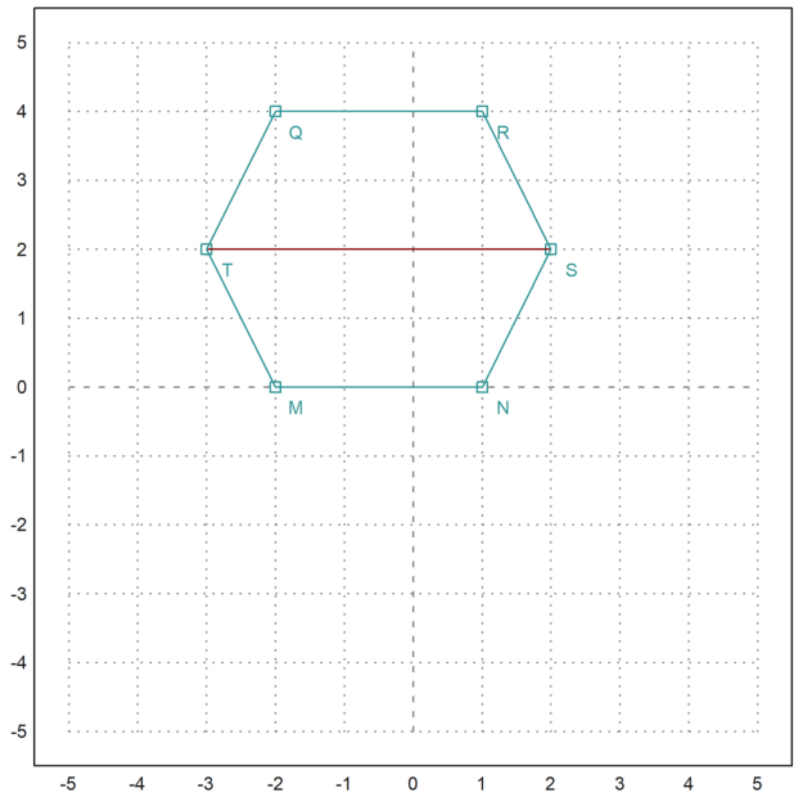
```
>setPlotRange(-5,5,-5,5);
>Q=[-2,4]; plotPoint(Q,"Q");
>R=[1,4]; plotPoint(R, "R");
>S=[2,2]; plotPoint(S, "S");
>T=[-3,2]; plotPoint(T, "T");
>M=[-2,0]; plotPoint(M, "M");
>N=[1,0]; plotPoint(N, "N"):
```




```
>color(2); plotSegment(Q,R," "); plotSegment(S,R," ");  plotSegment(S,T," ");  plotSegment
```

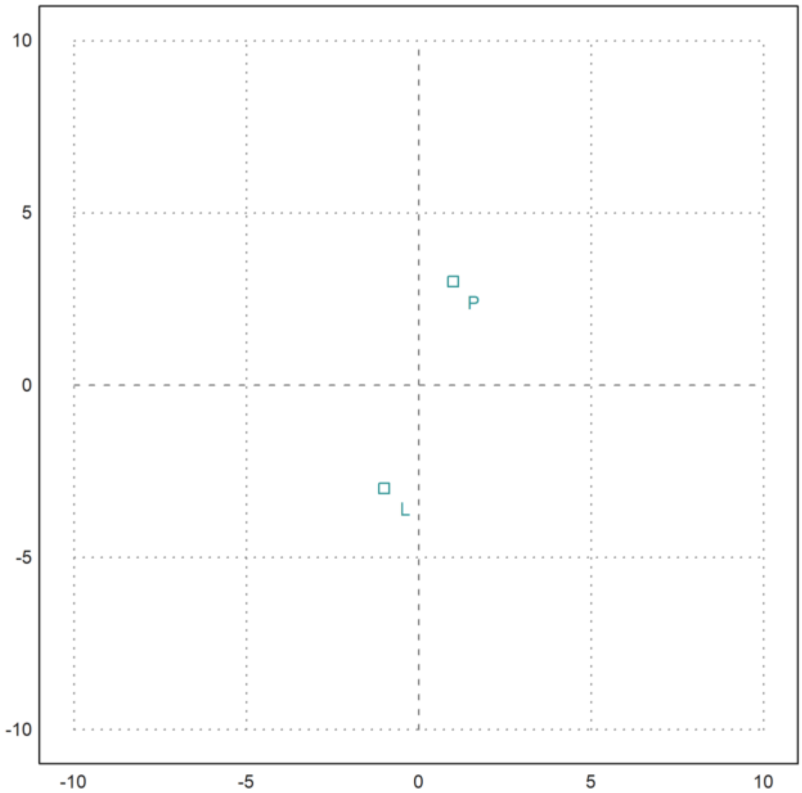


```
>color(5); plotSegment(Q,R," "); plotSegment(R,S," ");  plotSegment(S,N," ");  plotSegment
```

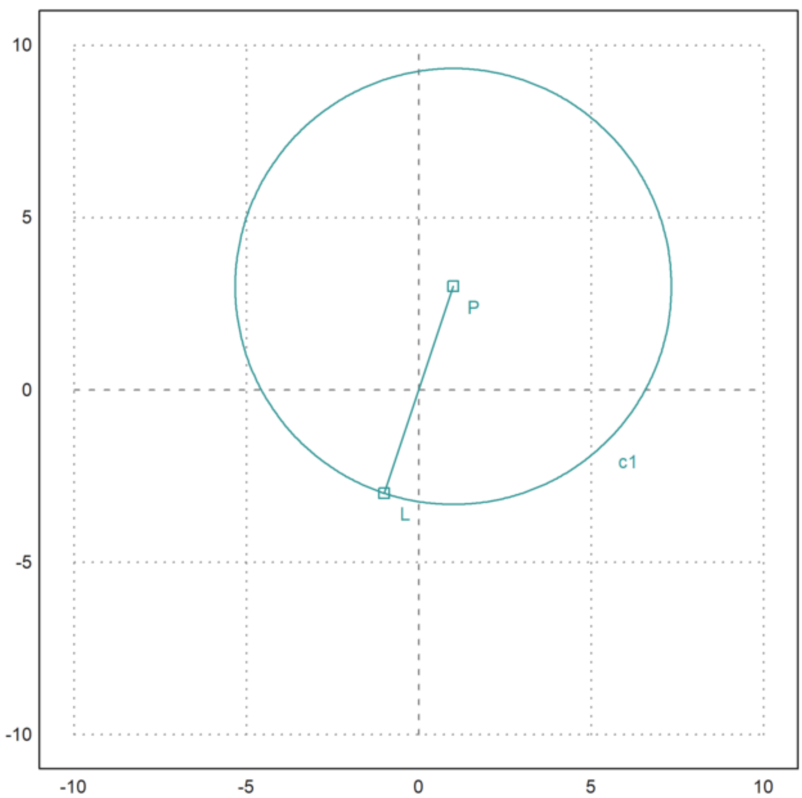


Menggambarkan lingkaran dengan pusat titik P. (Lingkaran akan digambar menggunakan fungsi geometri yang disediakan oleh Euler)
P adalah titik pusat lingkaran.
PL adalah jari-jari lingkaran.

```
>setPlotRange(-10,10,-10,10);
>P=[1,3]; plotPoint(P,"P");
>L=[-1,-3]; plotPoint(L,"L"):
```



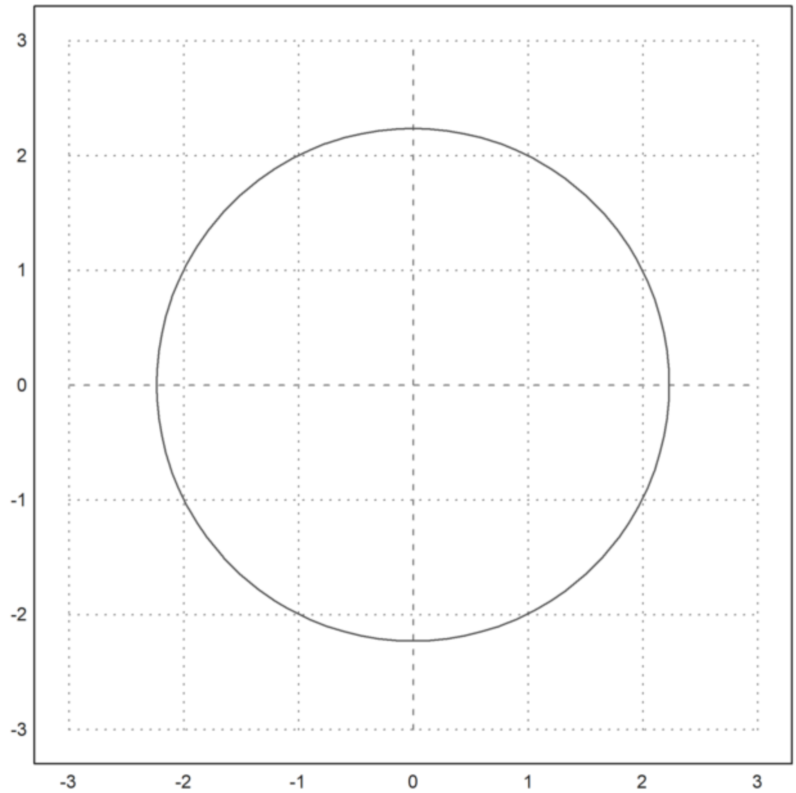

```
>c1=circleWithCenter(P,distance(P,L)); plotCircle(c1);
>plotSegment(P,L," "):
```



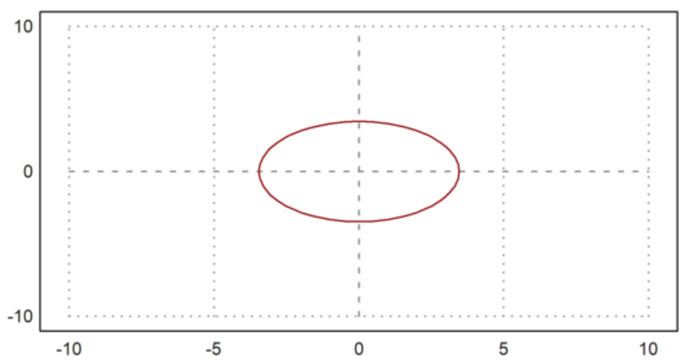
Sekarang kita akan menggambarkan lingkaran menggunakan Plot 2D dengan menggunakan rumus umum lingkaran, yaitu:

$$x^2 + y^2 = r^2$$

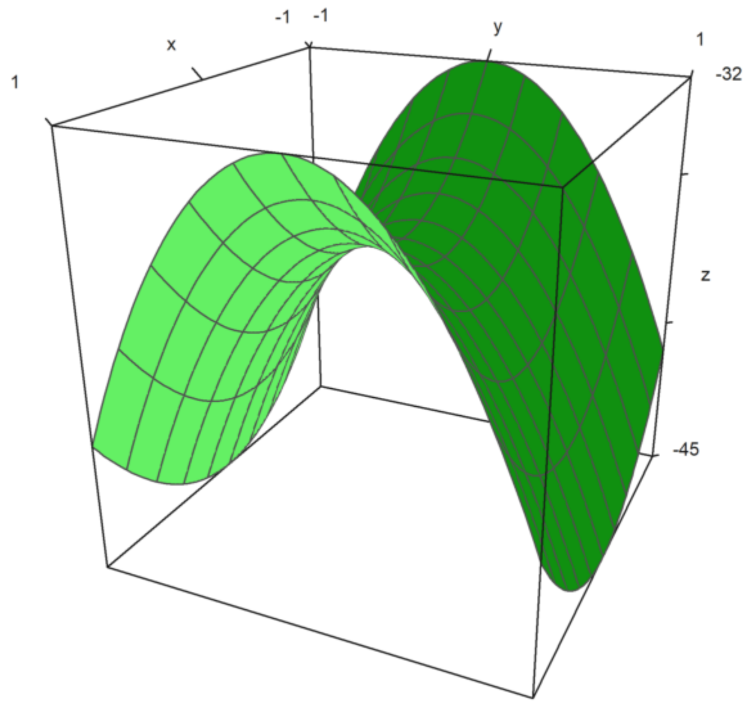
```
>aspect(1);
>plot2d("x^2+y^2-4",r=3,level=1):
```



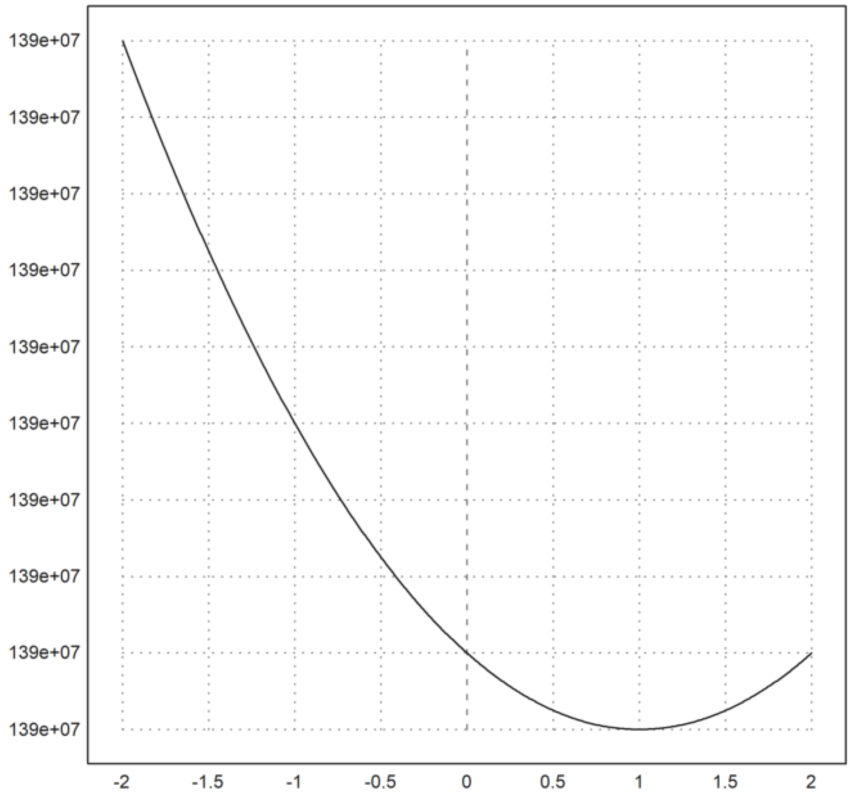

```
>aspect(2);
>plot2d("((x^2)/4)+((y^2)/4)-1",r=10,level=2,contourcolor=red):
```



```
>aspect(1);
>plot3d("4x^2-9y^2-36"): //hiperbola
```




```
>aspect(1);
>plot2d("x^2-2x-2y+5"): //parabola
```



Materi 4

a. Menentukan titik tengah suatu ruas garis
b. Menentukan titik potong dua garis
c. Menentukan titik potong garis dan lingkaran
d. Menentukan titik potong dua lingkaran

Menentukan Titik Tengah Suatu Ruas Garis

Titik tengah suatu garis adalah titik yang berada tepat di tengah garis tersebut, sehingga jarak dari titik ini ke ujung-ujung garis tersebut adalah sama. Dalam Matematika, untuk menemukan titik tengah suatu garis, kita dapat menggunakan formula berikut:
Jika kita memiliki dua koordinat ujung garis (x1,y1) maka koordinat titik tengahnya (x,y) dapat dihitung sebagai berikut:

$$(x, y)$$

dengan

$$x = \frac{(x1 + x2)}{2}$$

dan

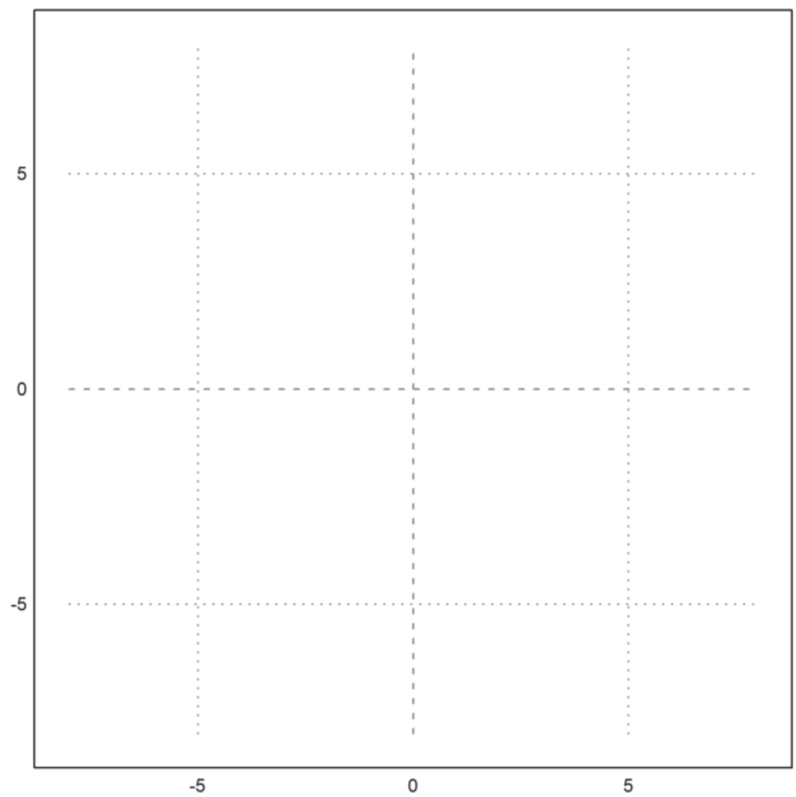
$$y = \frac{(y1 + y2)}{2}$$

maka dapat dituliskan

$$(\frac{(x1+x2)}{2}, \frac{(y1+y2)}{2})$$

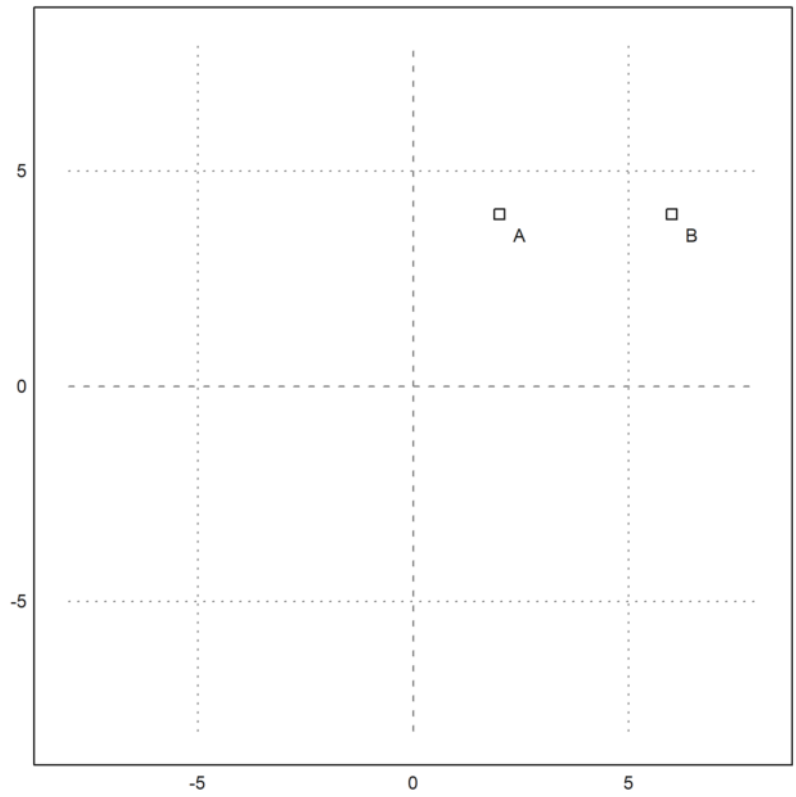
Ini berarti Kita menjumlahkan koordinat x dari kedua ujung garis dan kemudian membaginya dengan 2 untuk menemukan koordinat x titik tengah.
Demikian juga, Kita menjumlahkan koordinat y dari kedua ujung garis dan membaginya dengan 2 untuk menemukan koordinat y titik tengah.
Titik tengah suatu garis juga dapat ditemukan dengan metode geometri. Kita dapat menggambar dua garis diagonal dari sudut-sudut yang berlawanan pada segiempat yang dibentuk oleh garis tersebut. Titik pertemuan kedua diagonal ini adalah titik tengah garis.
Jadi, titik tengah suatu garis adalah titik yang terletak persis di tengah garis tersebut, dengan jarak yang sama dari kedua ujung garisnya. Ini adalah konsep dasar dalam geometri dan matematika yang sering digunakan dalam berbagai konteks, seperti pembuatan grafik, pemodelan geometri, dan banyak aplikasi lainnya.
CONTOH :
Diberikan ruas garis AB yang melalui dua titik A dan titik B. Dengan koordinat titik A dan titik B berturut-turut (2,4) dan (6,4). Tentukan koordinat titik tengah ruas garis AB

```
>setPlotRange(8):
```

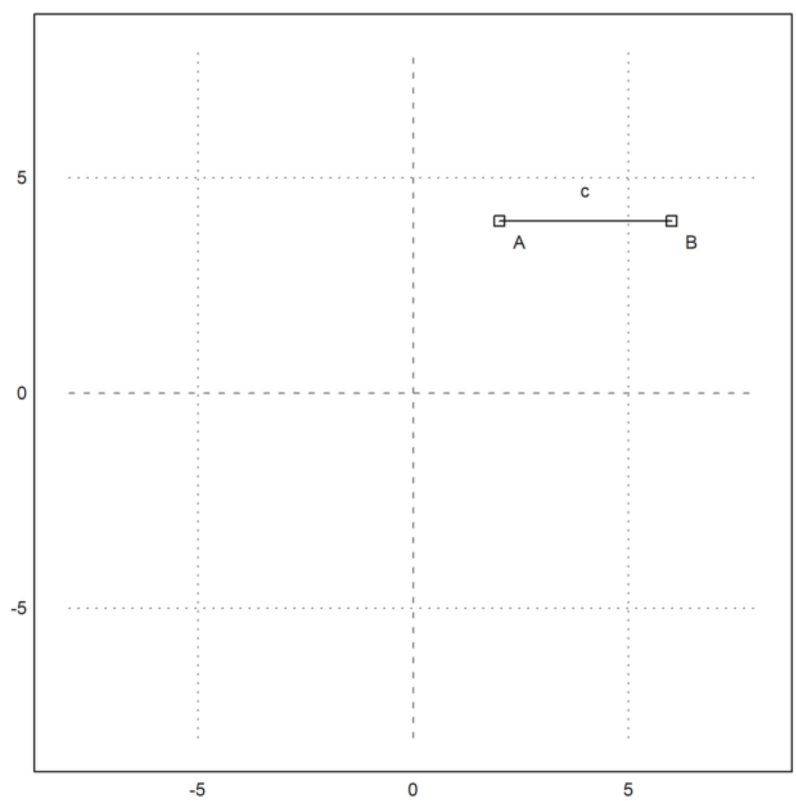


Selanjutnya buat ruas garis yang melalui 2 titik A dan B yang akan kita cari titik tengahnya pada bidang kartesius

```
>A=[2,4]; plotPoint(A,"A");
>B=[6,4]; plotPoint(B,"B"):
```

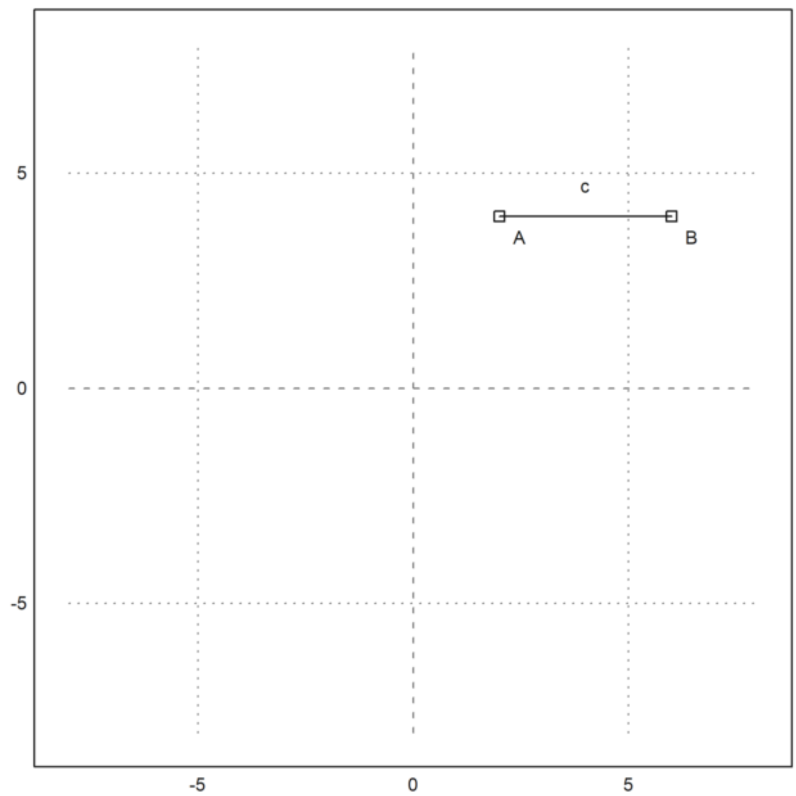



```
>plotSegment(A,B,"c"):
```



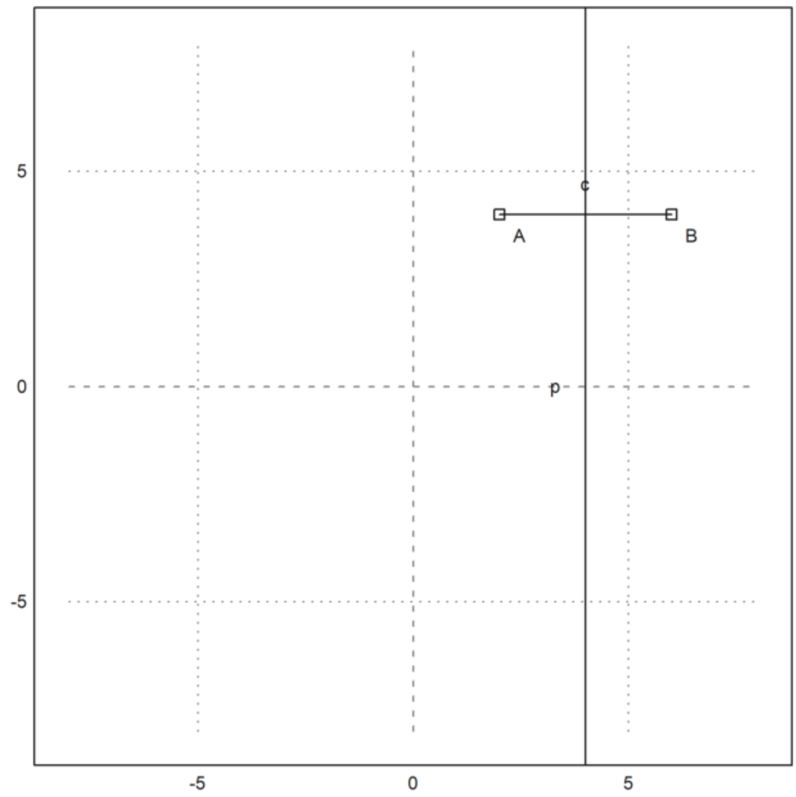
Berikutnya buat garis h yang tegak lurus ruas garis AB dan memotong tepat di tengah ruas garis AB

```
>h = middlePerpendicular(A,B):
```

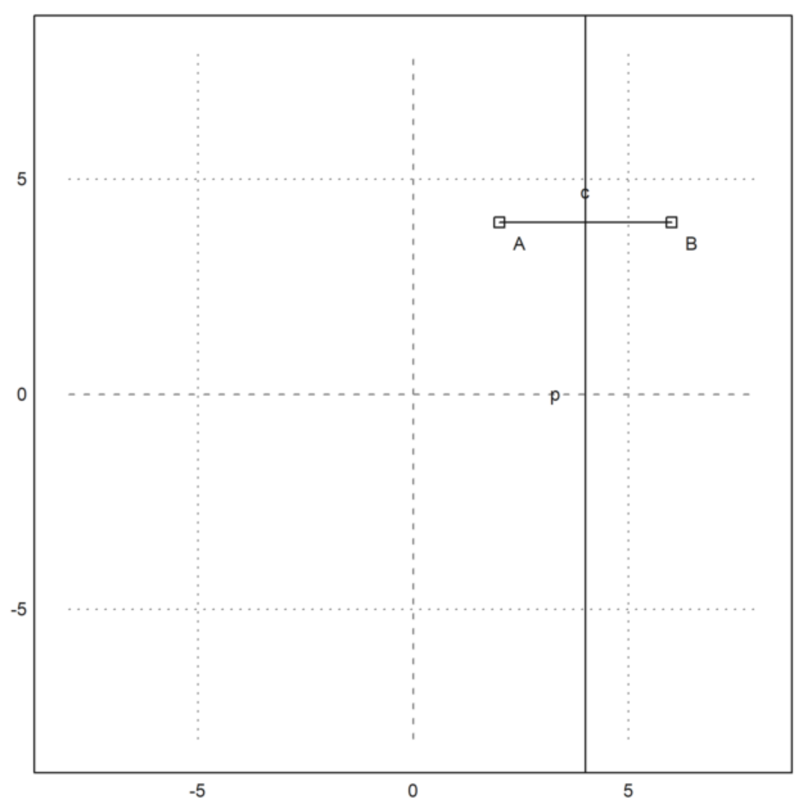


kemudian kita menentukan koordinat titik D yaitu perpotongan garis h dan ruas garis AB. titik D adalah titik tengah ruas garis AB

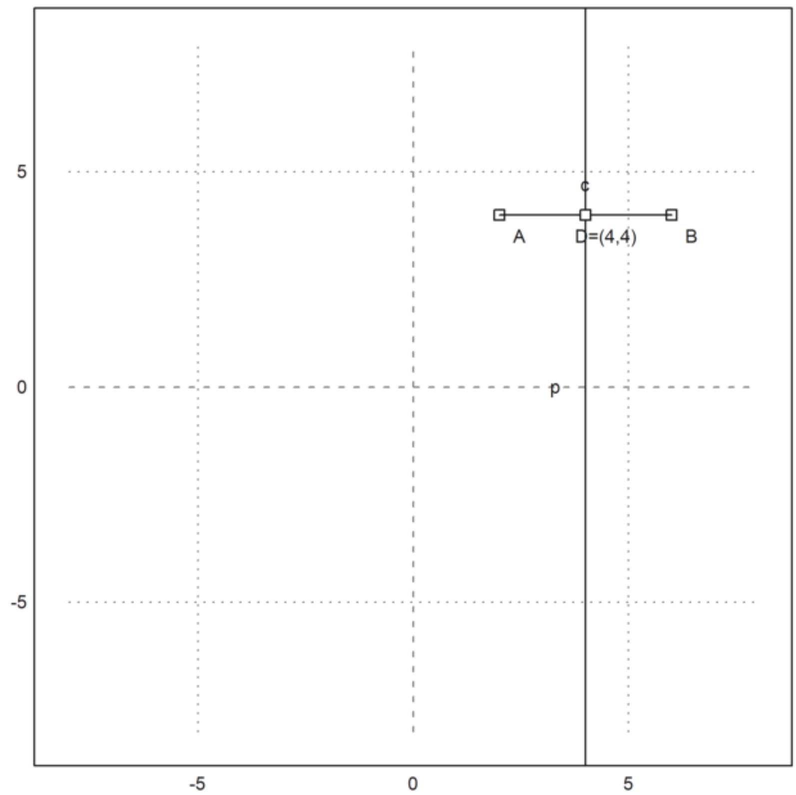
```
>plotLine(h):
```




```
>D=lineIntersection(h,lineThrough(A,B)):
```



```
>plotPoint(D,value=1):
```



Pembuktian menggunakan rumus matematika

$$(\frac{x1+x2}{2},\frac{y1+y2}{2})$$

$$=(\frac{2+6}{2},\frac{4+4}{2})$$

$$=(\frac{8}{2},\frac{8}{2})$$

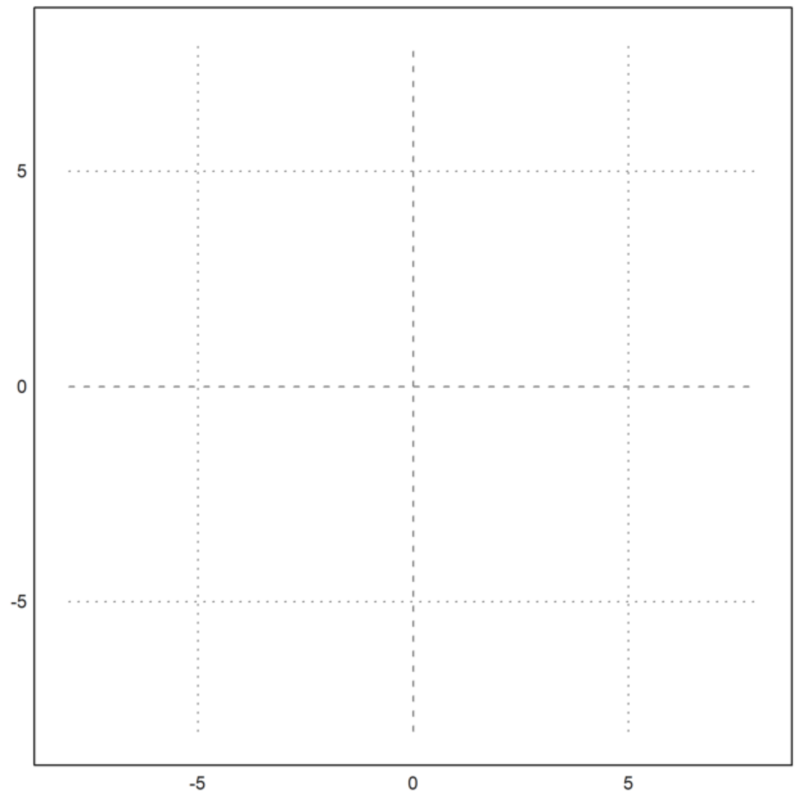
$$=(4,4)$$

Jadi, terbukti benar bahwa titik tengah ruas garis AB adalah(4,4)

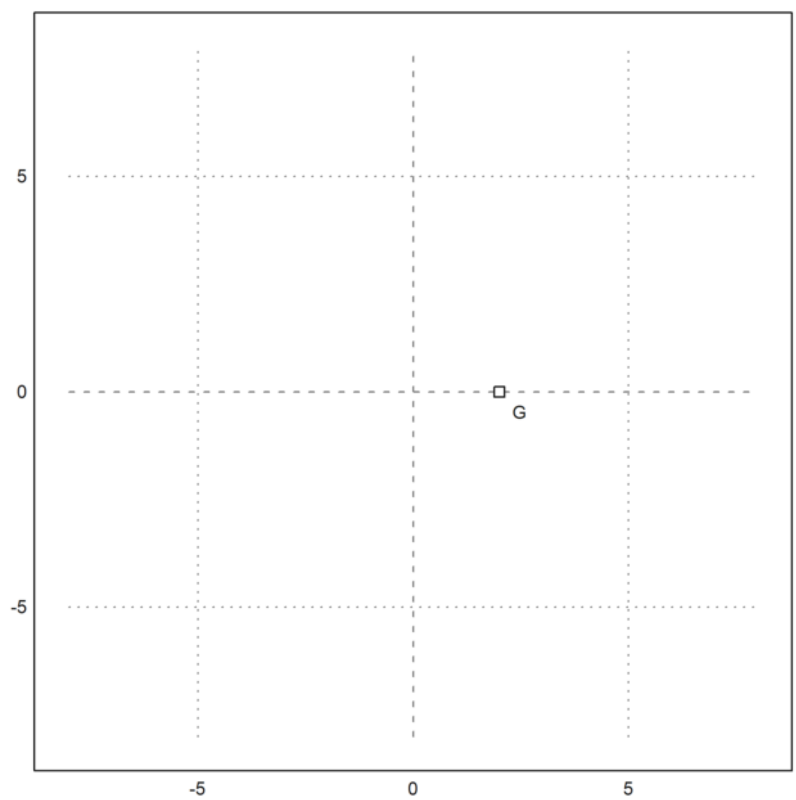
Latihan

1. Diberikan ruas garis l yang melalui dua titik F dan titik G. Dengan koordinat titik F dan titik G berturut-turut (2,0) dan (2,6). tentukan koordinat titik tengah ruas garis l

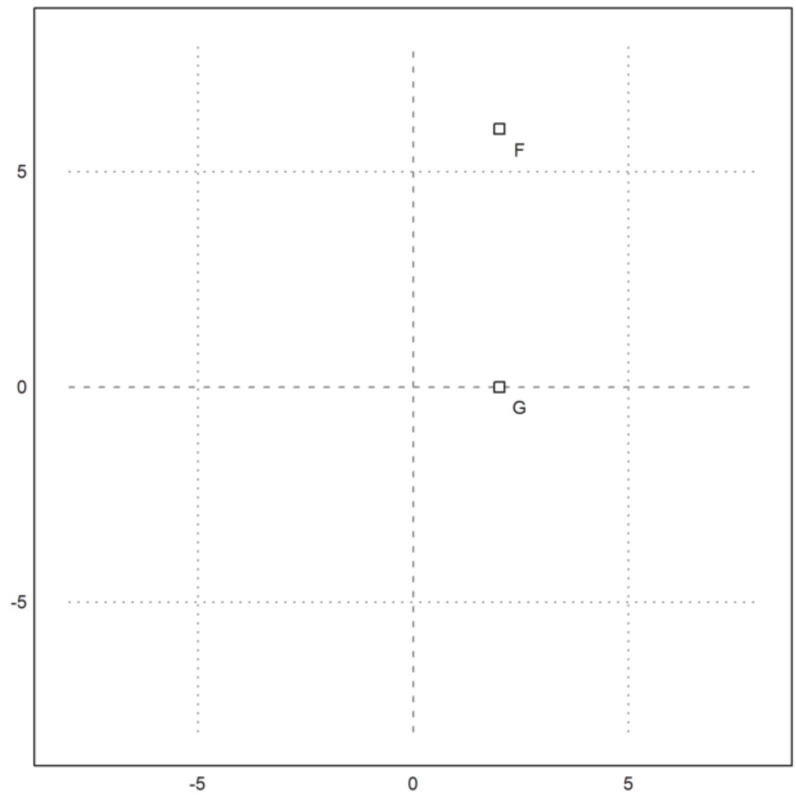
```
>
>setPlotRange(8):
```



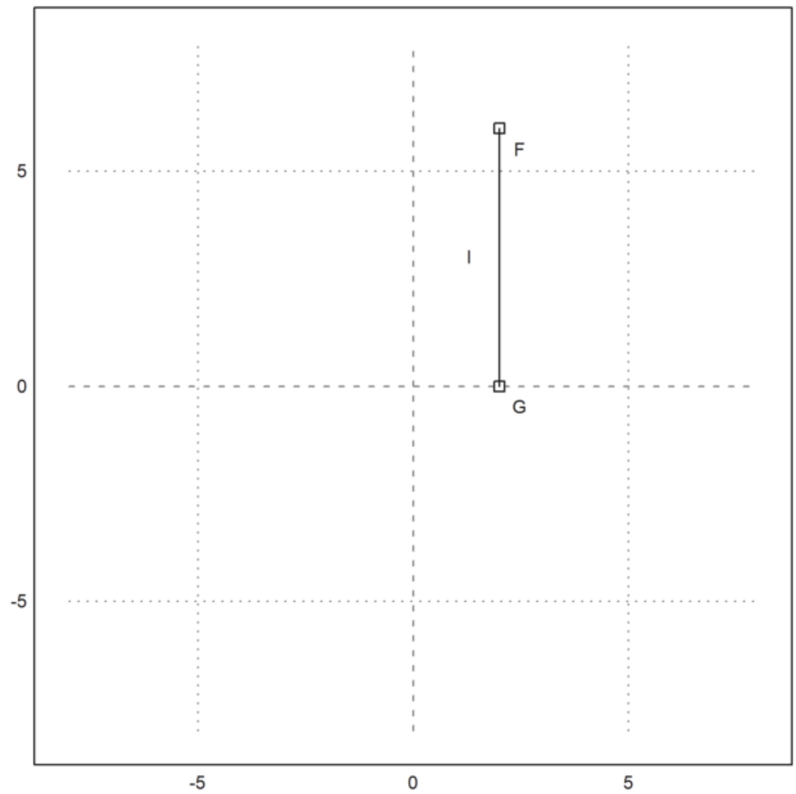

```
>G=[2,0]; plotPoint(G,"G"):
```



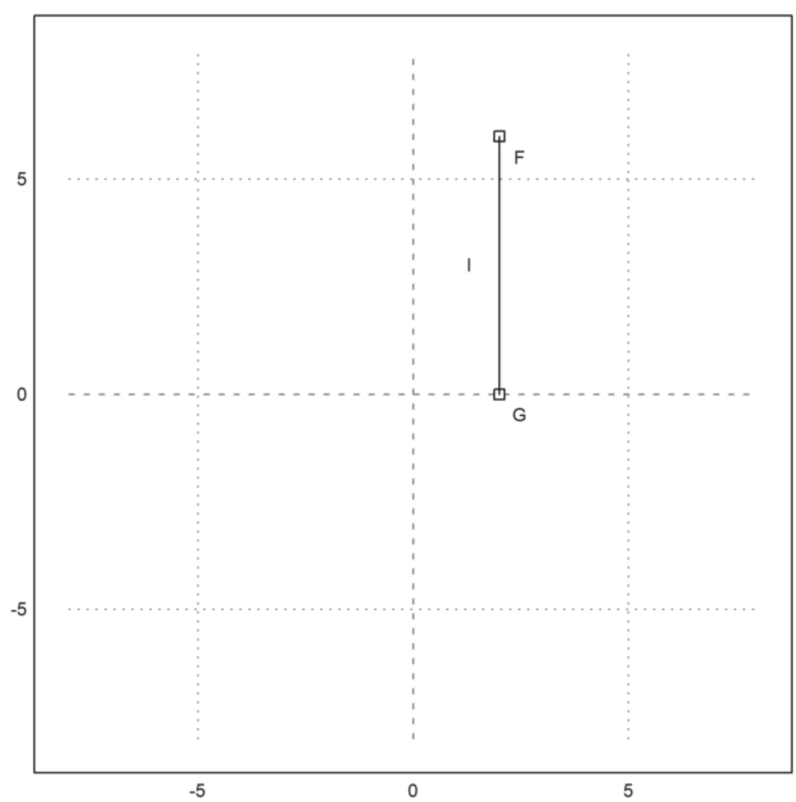
```
>F=[2,6]; plotPoint(F,"F"):
```



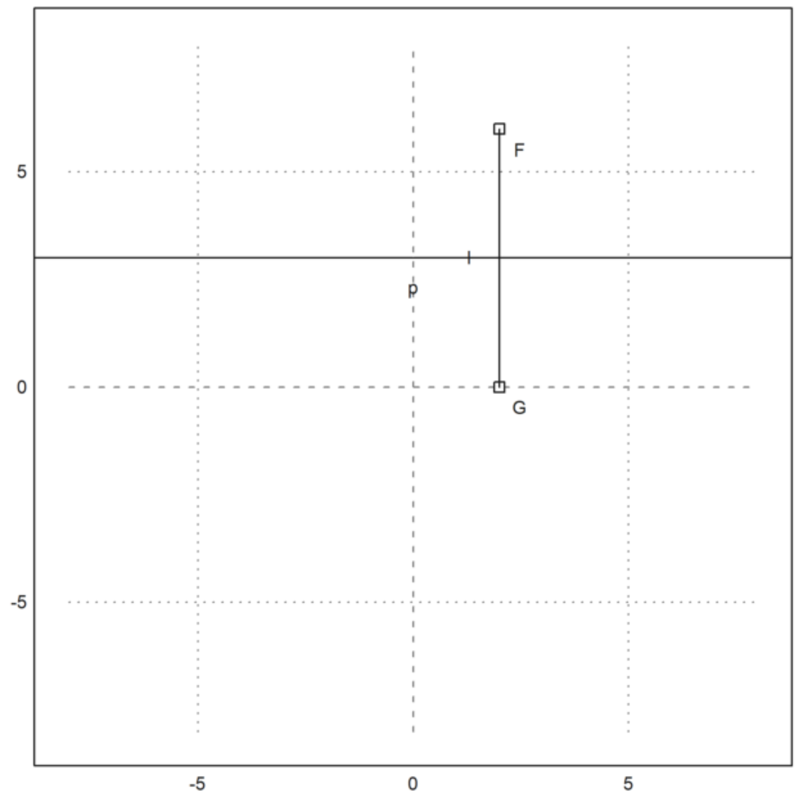

```
>plotSegment(G,F,"l"):
```



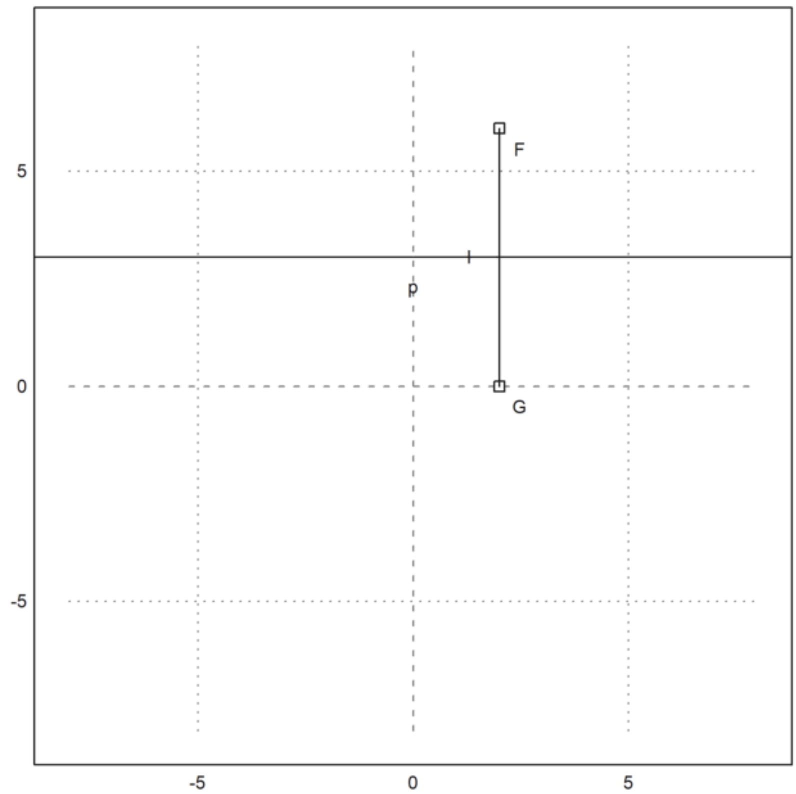

```
>h= middlePerpendicular(G,F):
```



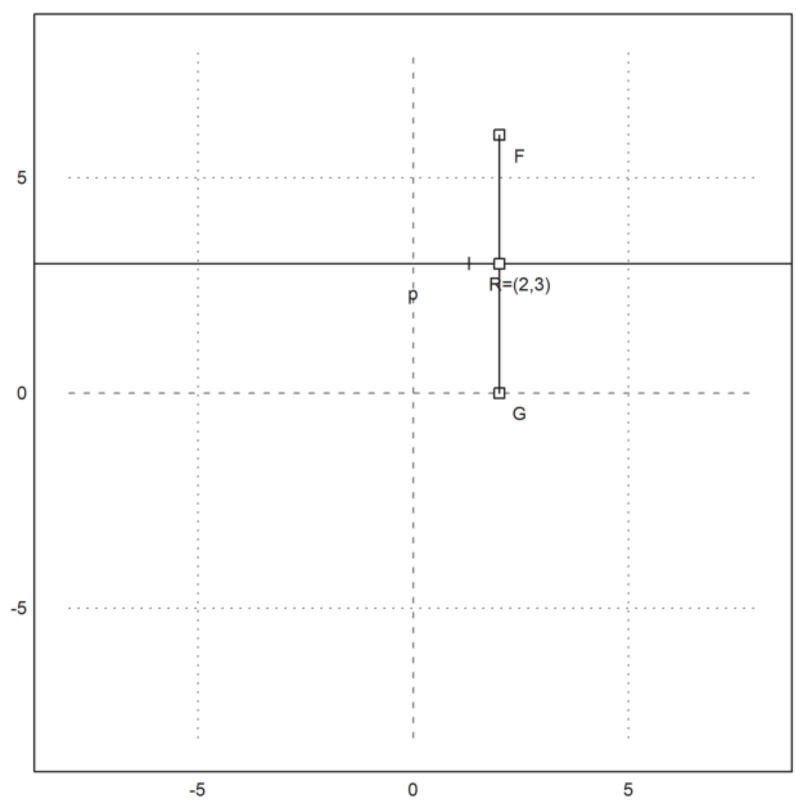
```
>plotLine(h):
```




```
>R=lineIntersection(h,lineThrough(G,F)):
```




```
>plotPoint(R,value=1):
```



Pembuktian menggunakan rumus matematika

$$(\frac{x1+x2}{2},\frac{y1+y2}{2})$$

$$=(\frac{2+2}{2},\frac{0+6}{2})$$

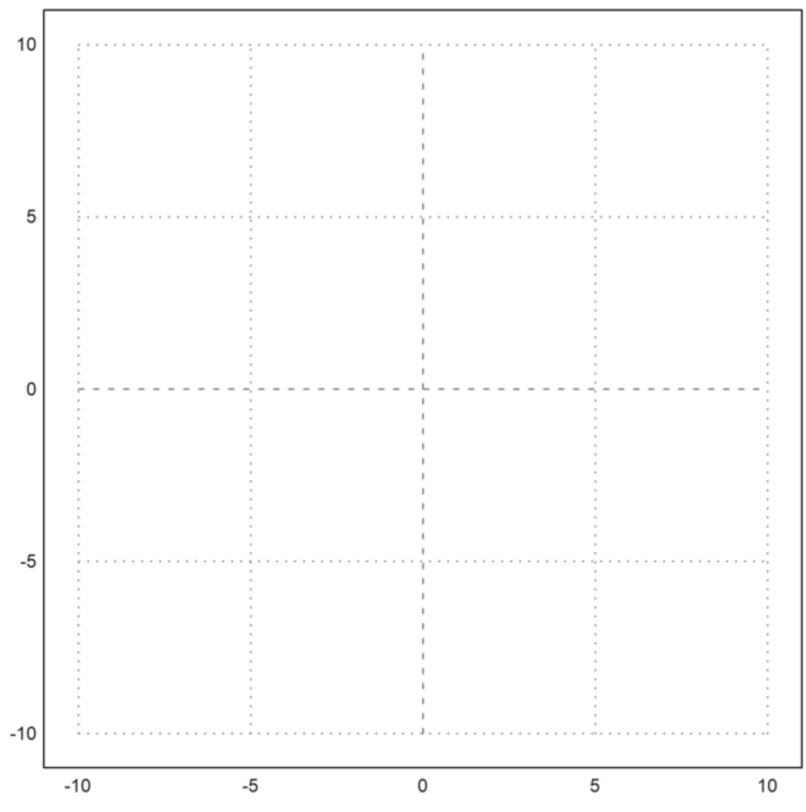
$$=(\frac{4}{2},\frac{6}{2})$$

$$=(2,3)$$

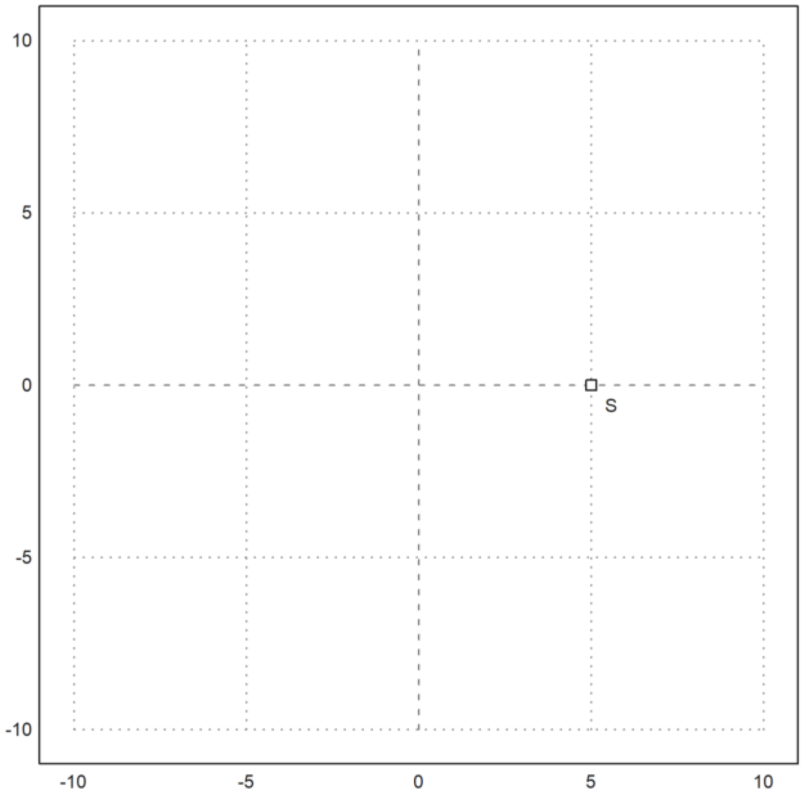
Jadi, terbukti benar bahwa titik tengah ruas garis FG adalah(2,3)

2. Diberikan ruas garis k yang melalui dua titik S dan titik R. Dengan koordinat titik S dan titik R berturut-turut (5,0) dan (3,6). Tentukan koordinat titik tengah ruas garis k

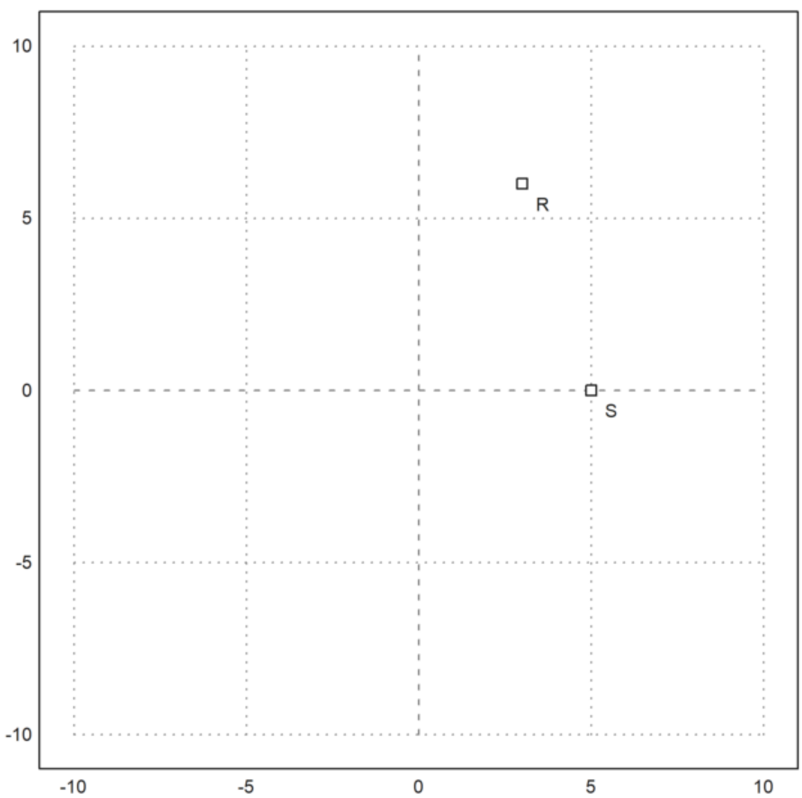
```
>setPlotRange(10):
```



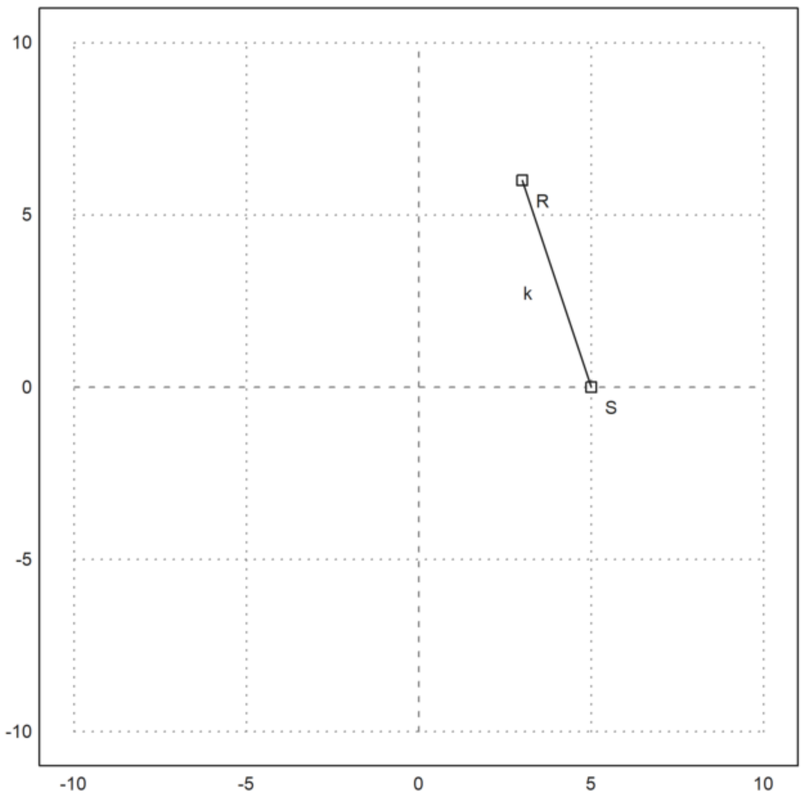

```
>S=[5,0]; plotPoint(S,"S"):
```



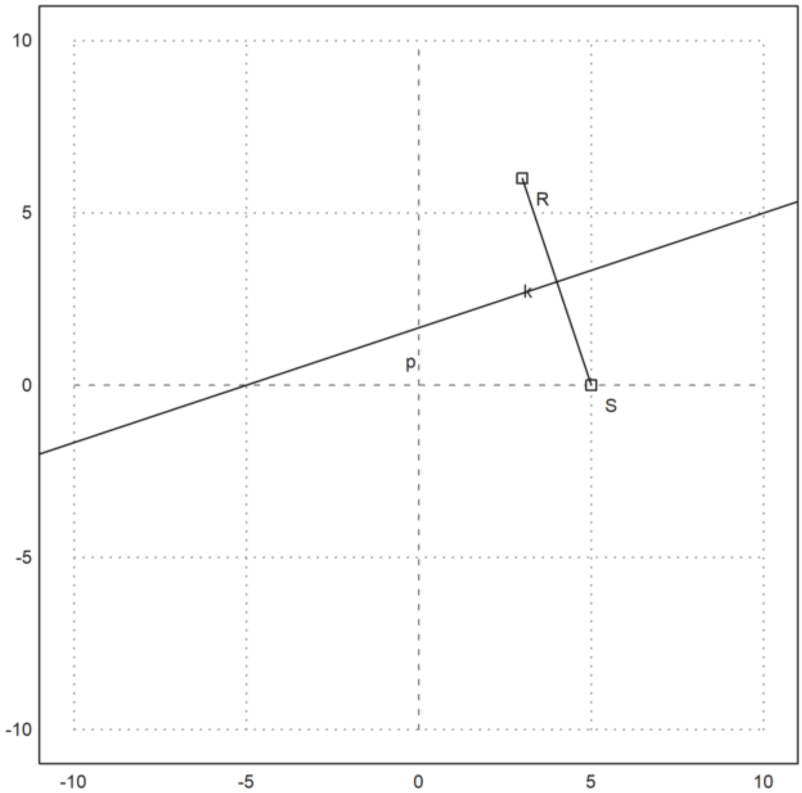

```
>R=[3,6]; plotPoint(R,"R"):
```



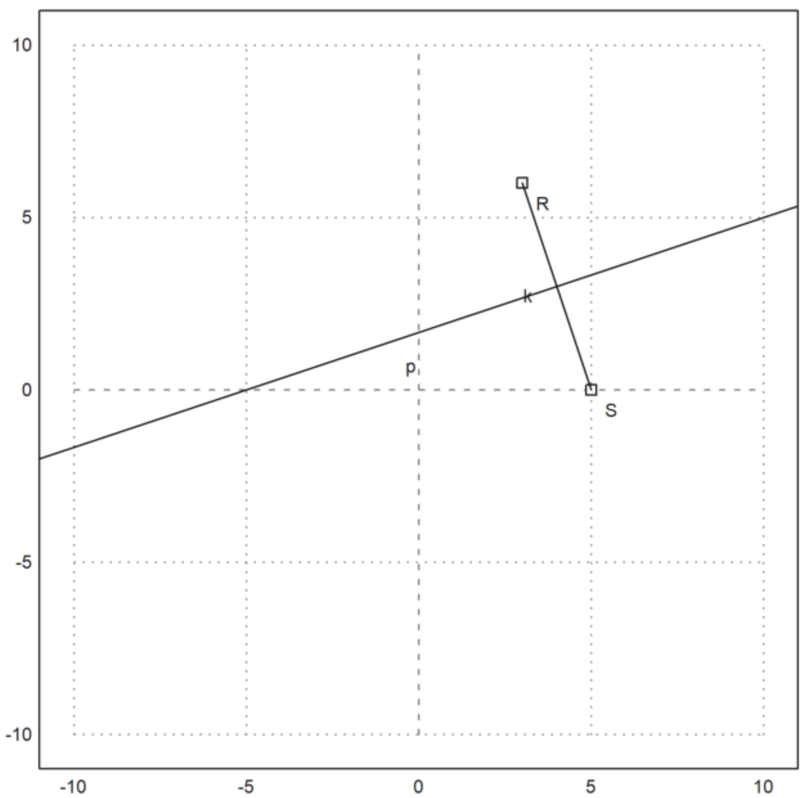
```
>plotSegment(S,R,"k"):
```



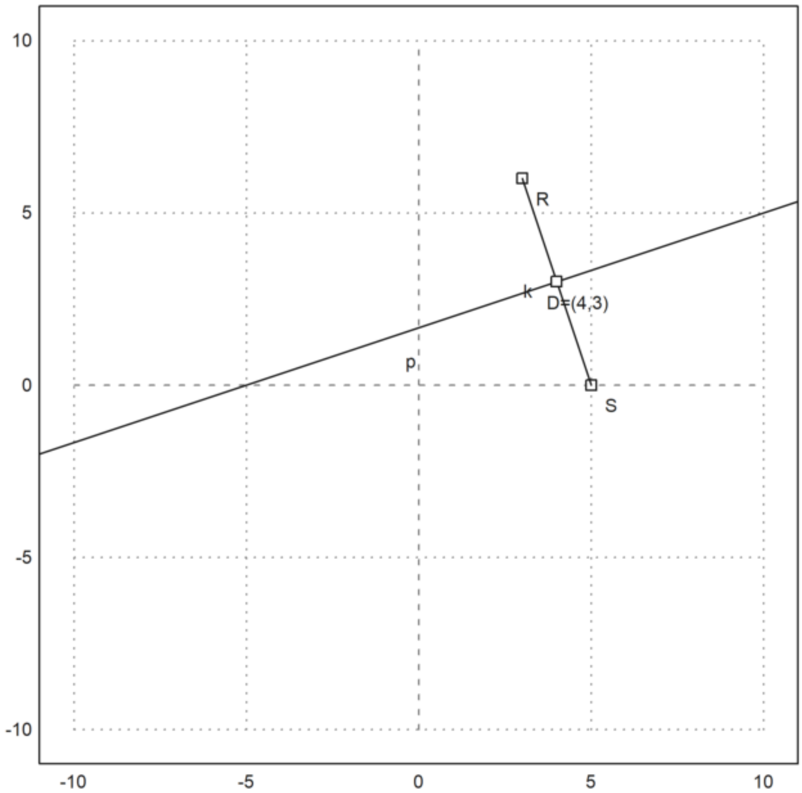

```
>h = middlePerpendicular(S,R);
>plotLine(h):
```




```
>D=lineIntersection(h,lineThrough(S,R)):
```



```
>plotPoint(D,value=1):
```



Pembuktian menggunakan rumus matematika

$$(\frac{x1+x2}{2},\frac{y1+y2}{2})$$
$$=(\frac{5+3}{2},\frac{0+6}{2})$$
$$=(\frac{8}{2},\frac{6}{2})$$
$$=(4,3)$$

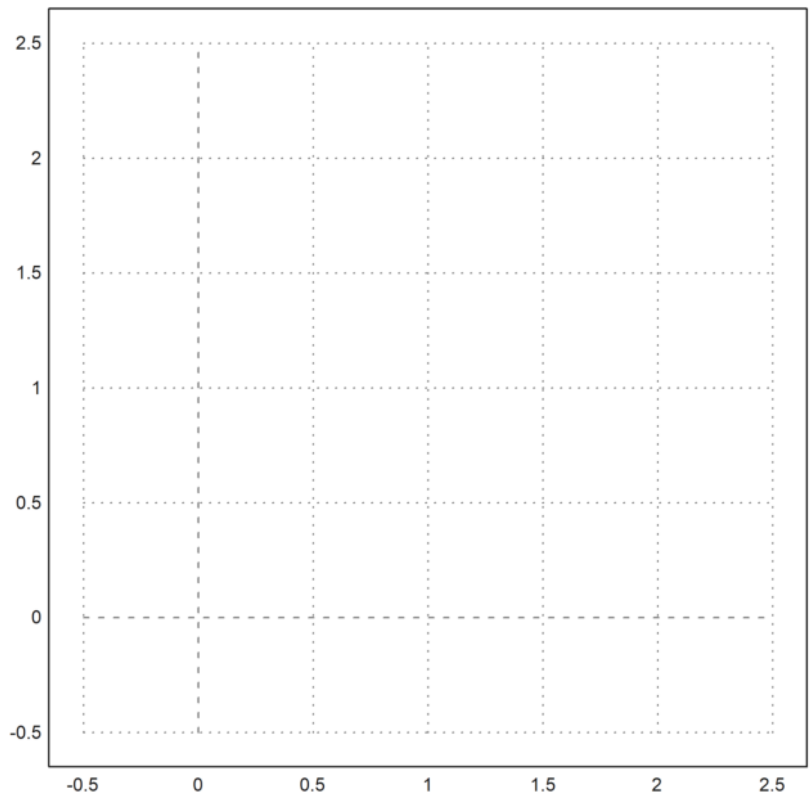
Jadi, terbukti benar bahwa titik tengah ruas garis SR adalah(4,3)

Menentukan titik potong dua garis

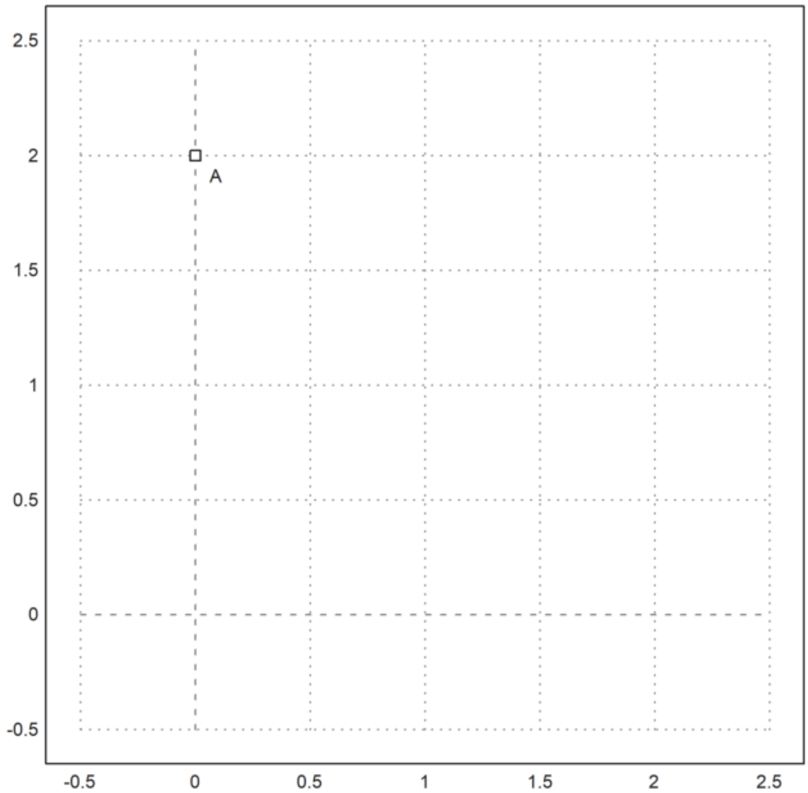
Titik potong dua garis adalah titik dimana dua garis lurus berpotongan satu sama lain. Dalam konteks geometri atau matematika, dua garis yang berpotongan akan memiliki satu titik potong, yang merupakan titik yang terletak pada kedua garis tersebut. Titik potong ini memiliki koordinat yang spesifik dalam sistem koordinat yang digunakan.
Dalam geometri Euclidean, jika dua garis sejajar(paralel), maka mereka tidak akan memiliki titik potong. Namun, jika dua garis tidak sejajar, maka mereka akan memiliki satu titik potong yang dapat dihitung atau ditentukan.
Secara matematis, untuk menemukan titik potong dua garis, kita dapat menggunakan sistem persamaan linier. Jika kita memiliki dua persamaan linier yang mewakili dua garis, Kita dapat menyelesaikan sistem persamaan ini untuk menemukan nilai koordinat titik potong. Biasanya, kita akan menggunakan metode substitusi, eliminasi, atau grafik untuk menemukan titik potongnya.

CONTOH :
Diberikan titik A(0,2), B(2,0), C(0,0), dan D(2,2). c merupakan ruas garis AB dan b merupakan Ruas garis CD. tentukan koordinat titik potong dua ruas garis tersebut!
Jawab:
langkah pertama kita menentukan luas bidang kartesius terlebih dahulu

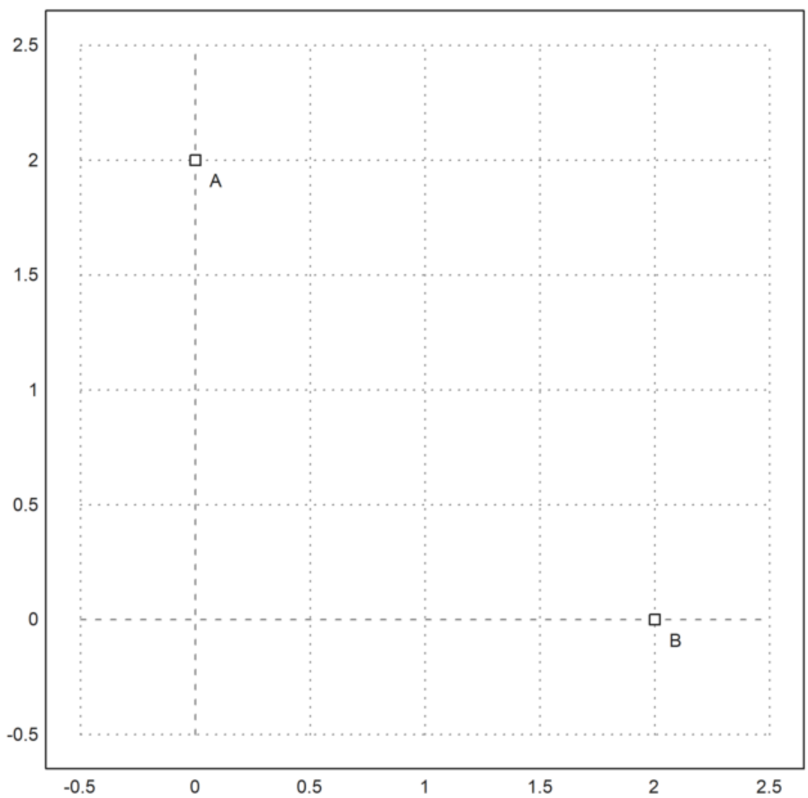
```
>setPlotRange(-0.5,2.5,-0.5,2.5): // mendefinisikan bidang koordinat baru
```



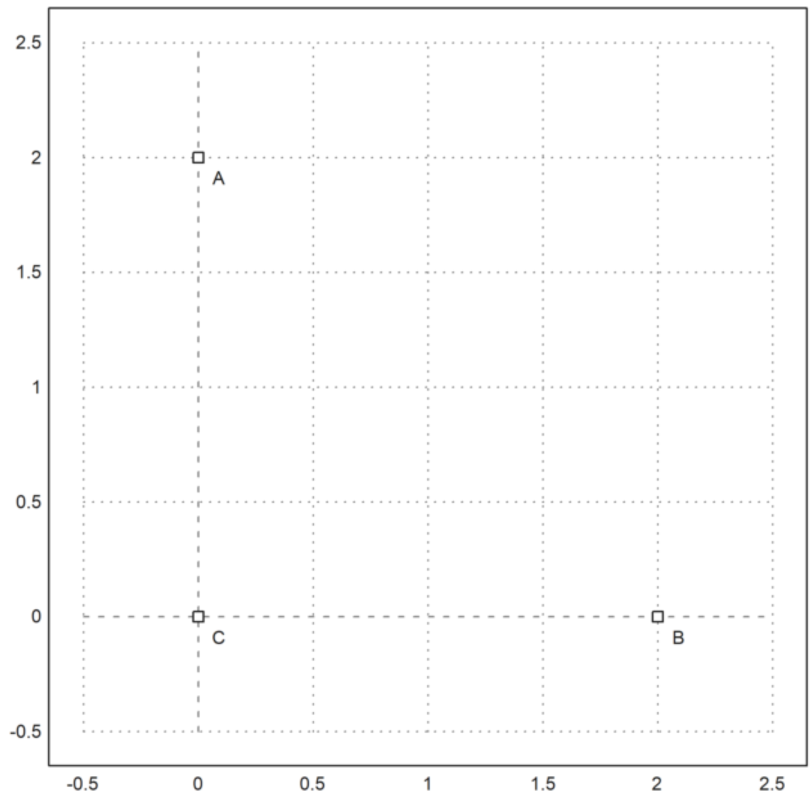

```
>A=[0,2]; plotPoint(A,"A"):
```



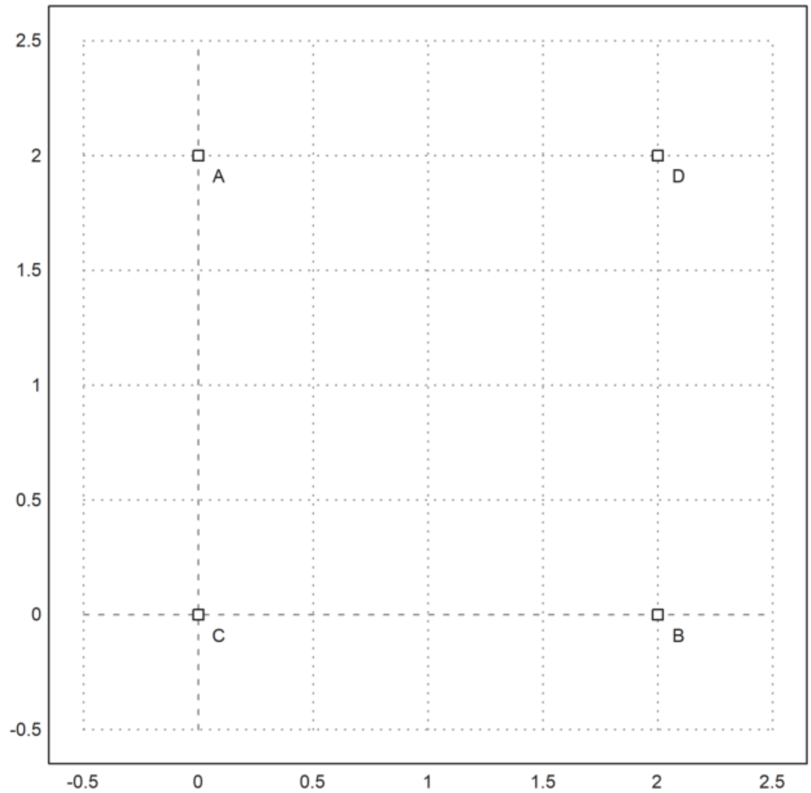

```
>B=[2,0]; plotPoint(B,"B"):
```



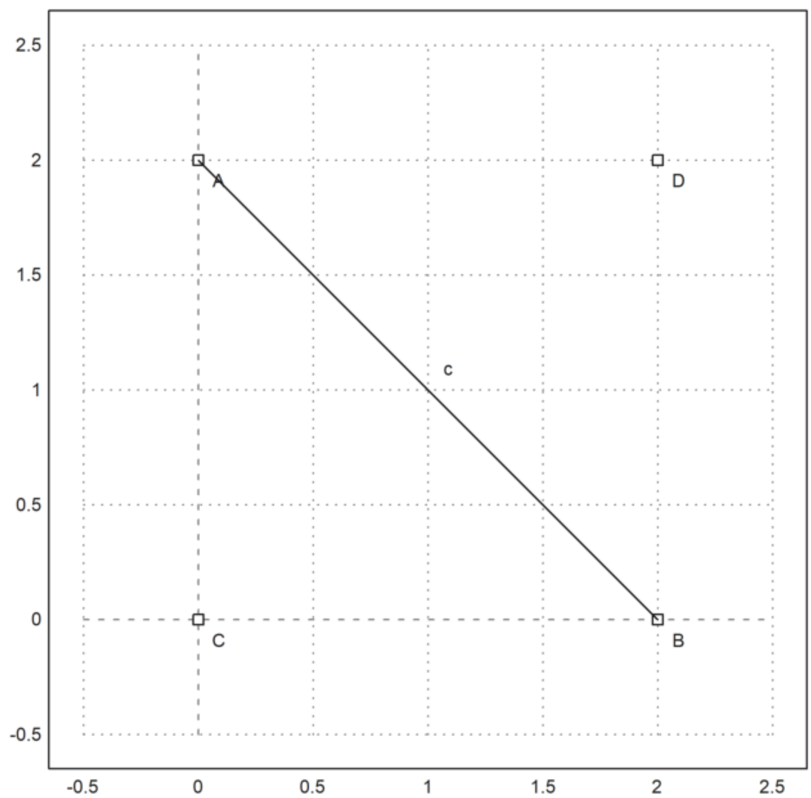
```
>C=[0,0]; plotPoint(C,"C"):
```



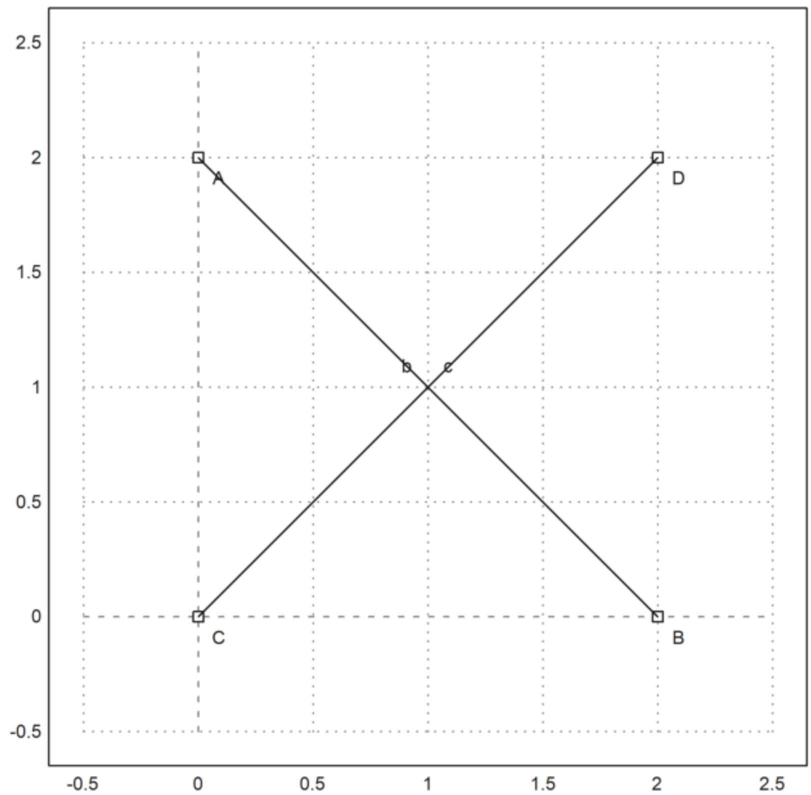

```
>D=[2,2]; plotPoint(D,"D"):
```




```
>plotSegment(A,B,"c"): // c=AB
```

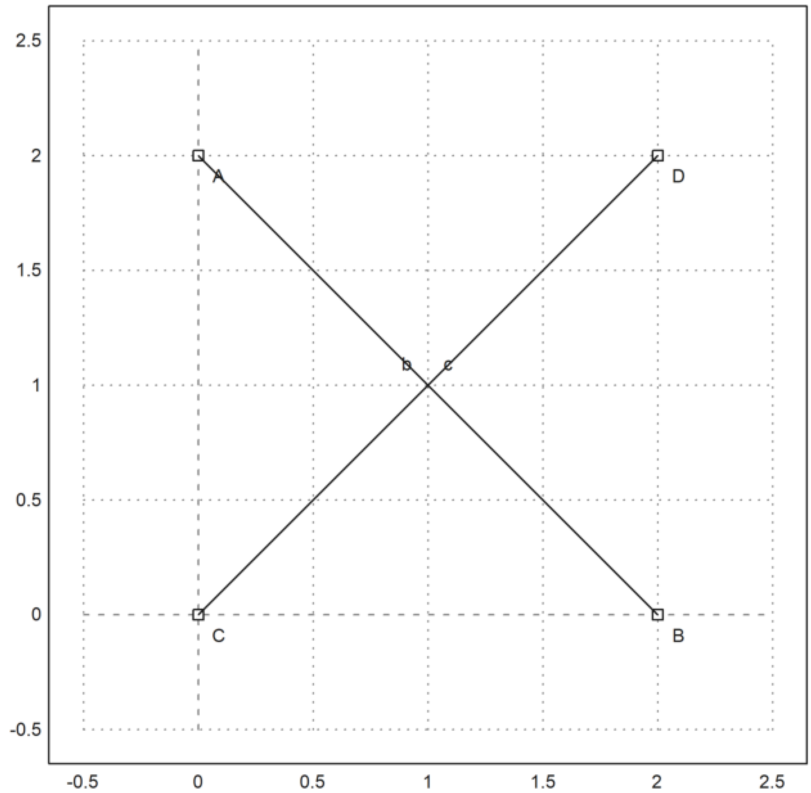


```
>plotSegment(C,D,"b"): // b=CD
```

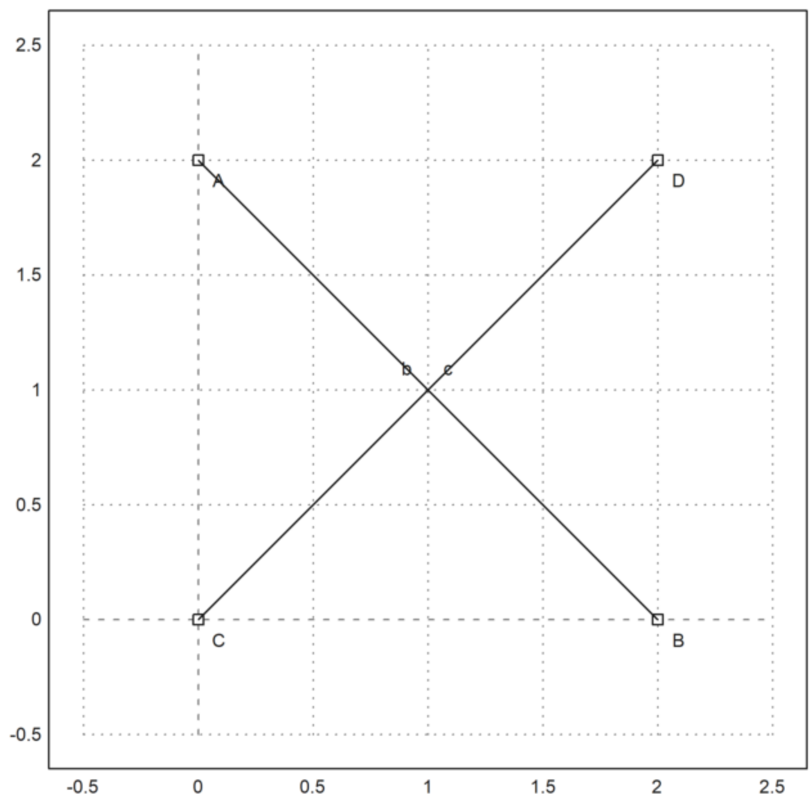


berikutnya kita akan menentukan koordinat titik potongnya

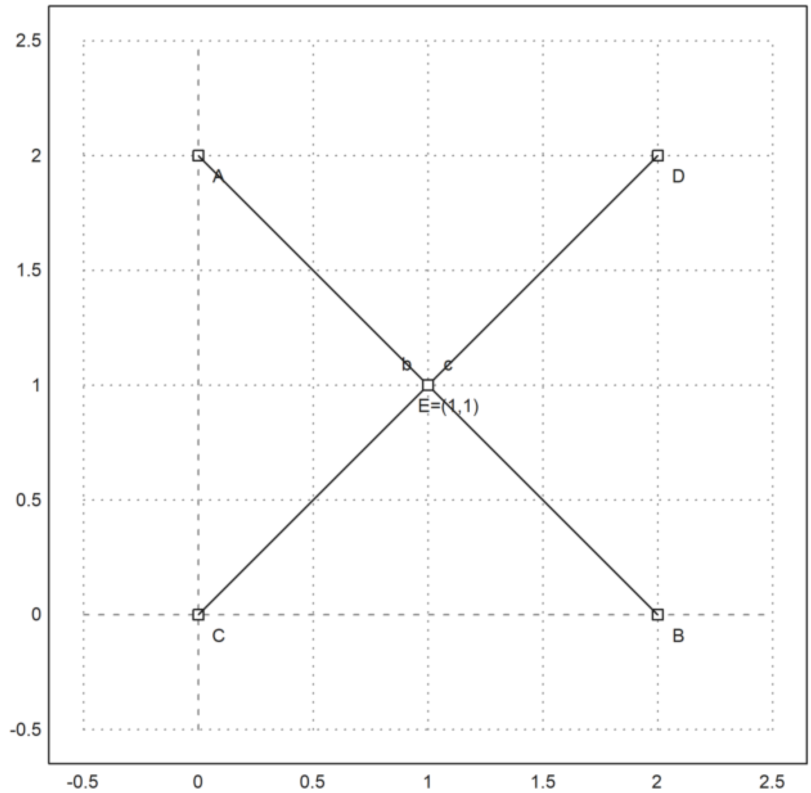
```
>lineThrough(A,B): // garis yang melalui A dan B
```




```
>lineThrough(C,D): // garis yang melalui C dan D
```



```
>E=lineIntersection(lineThrough(A,B),lineThrough(C,D)); // E adalah titik potong c dan b
>plotPoint(E,value=1): // koordinat E ditampilkan
```



Pembuktian dengan rumus matematika:
1. Mencari persamaan garis AB

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$
$$\frac{y - 2}{0 - 2} = \frac{x - 0}{2 - 0}$$
$$\frac{y - 2}{-2} = \frac{x}{2}$$

$$2y - 4 = -2x$$

$$y = -x + 2$$

2. Mencari persamaan garis CD

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$
$$\frac{y - 0}{2 - 0} = \frac{x - 0}{2 - 0}$$
$$\frac{y}{2} = \frac{x}{2}$$

$$2y = 2x$$

$$y = x$$

3. Subsitusi y pada CD dengan -x+2 pada AB

$$y = -x + 2$$

$$x = -x + 2$$

$$2x = 2$$

$$x = 1$$

4. Subsitusi x=1 ke persamaan CD

$$y = x$$

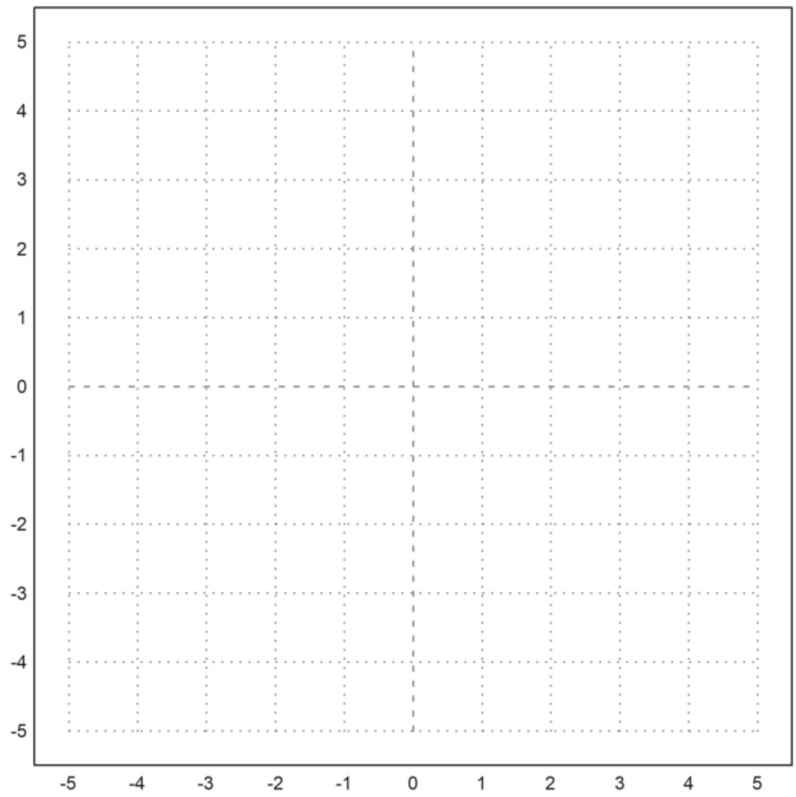
$$y = 1$$

Jadi, titik koordinatnya (1,1)
Terbukti **Latihan**

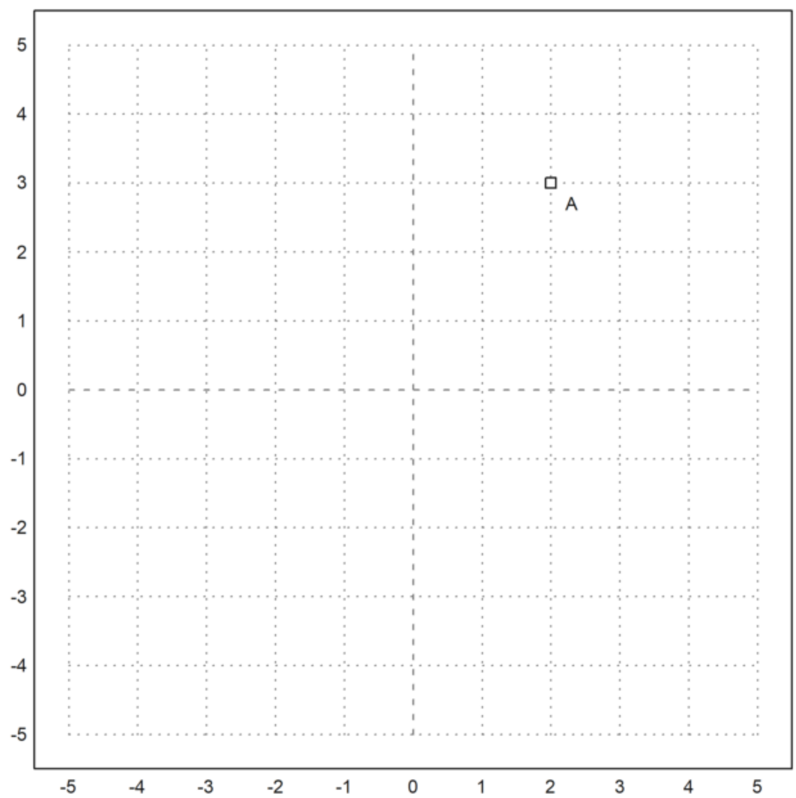
---

1. Diberikan dua garis yang melewati titik koordinat sebagai berikut:
garis AB : titik A(2,3)dan titik B(0,1)
garis CD : titik C(0,3) dan titik D(3,0)
tentukan koordinat titik potong kedua garis tersebut!

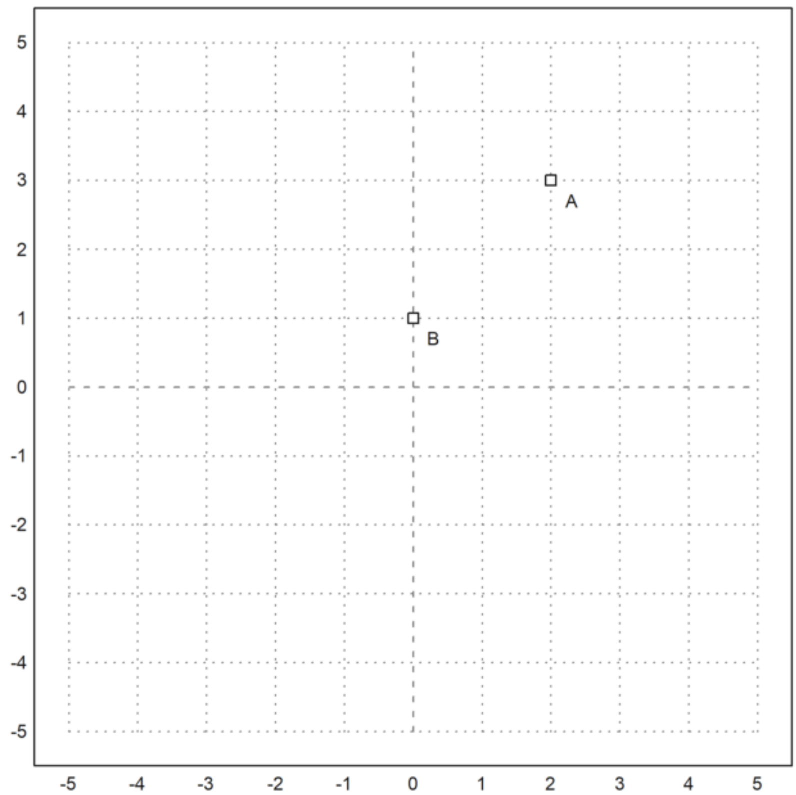
```
>setPlotRange(5):
```



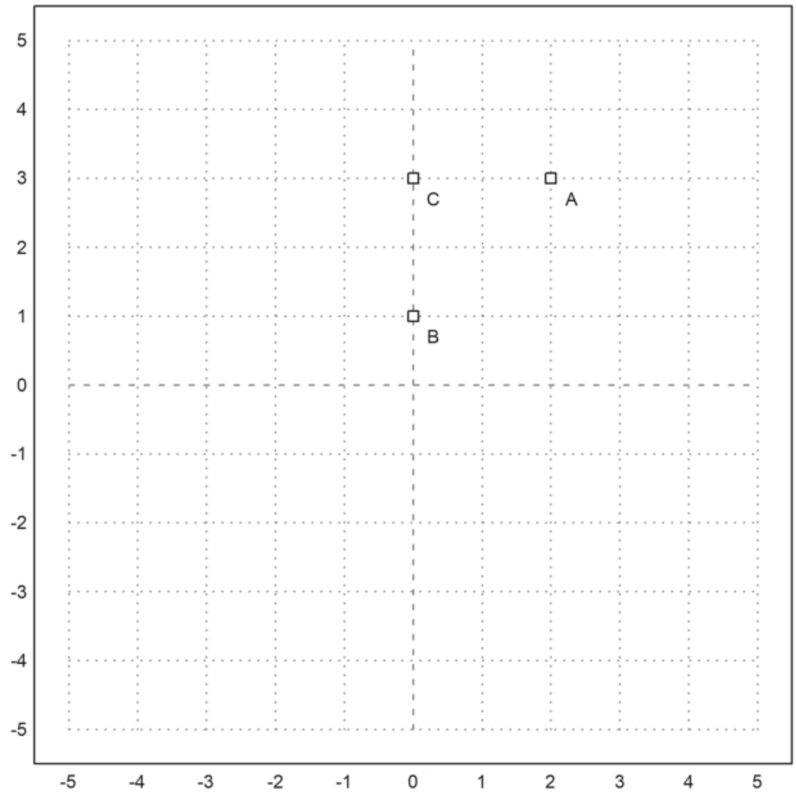

```
>A=[2,3]; plotPoint(A,"A"):
```



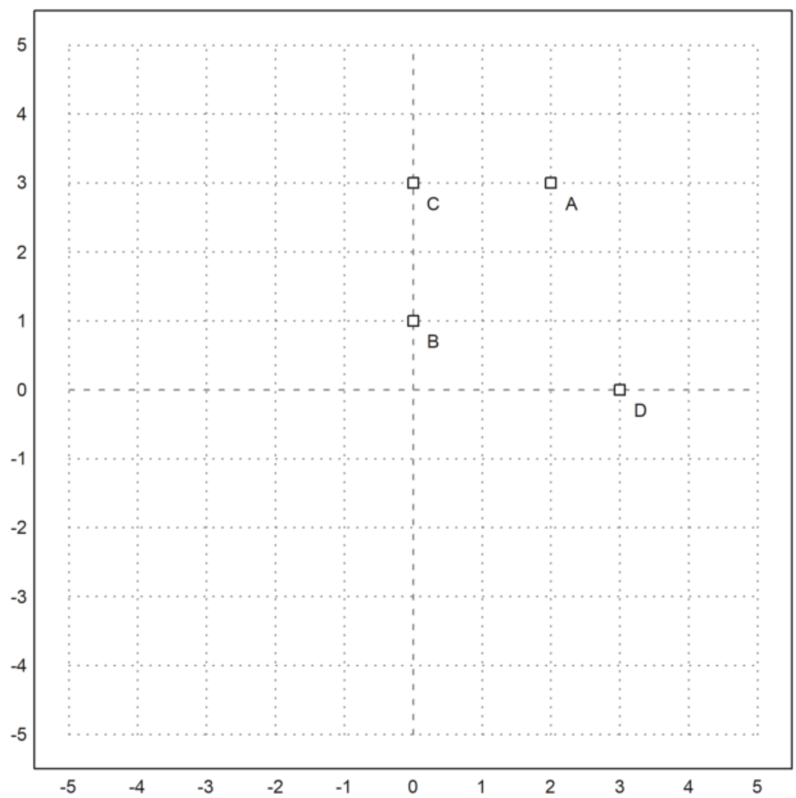
```
>B=[0,1]; plotPoint(B,"B"):
```



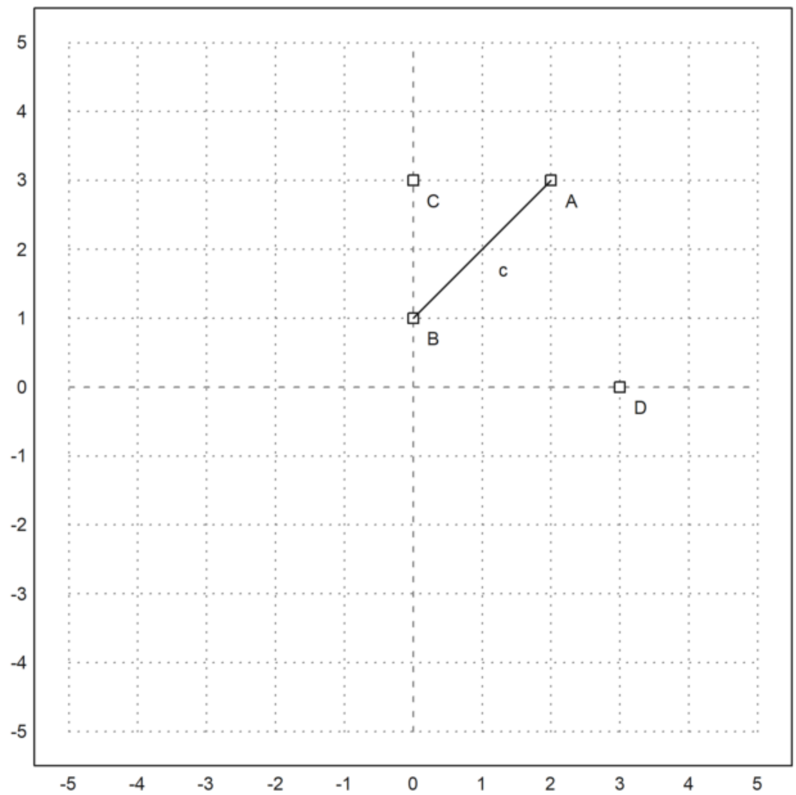

```
>C=[0,3]; plotPoint(C,"C"):
```



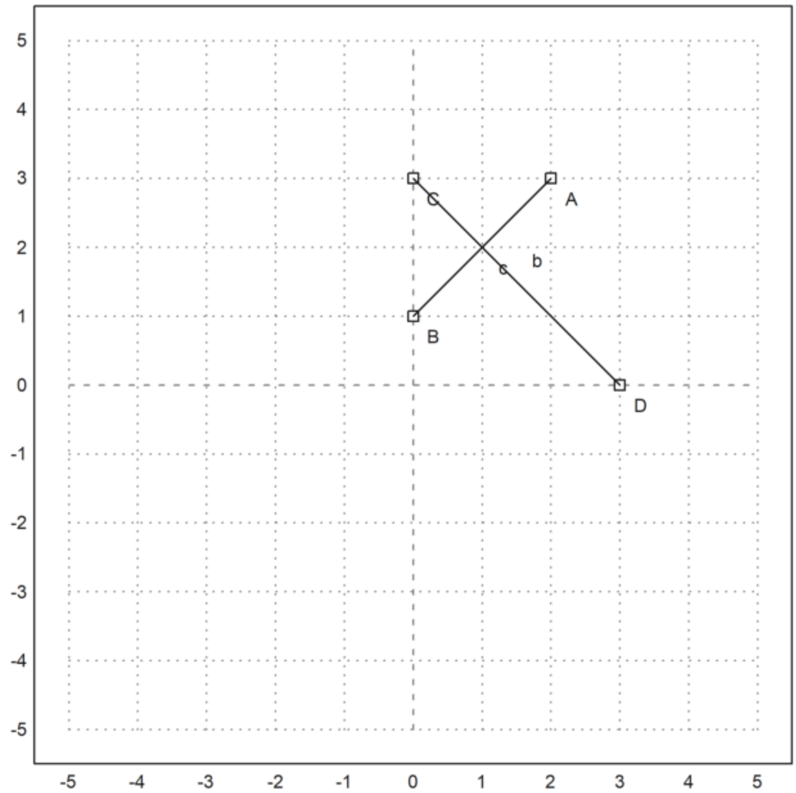

```
>D=[3,0]; plotPoint(D,"D"):
```



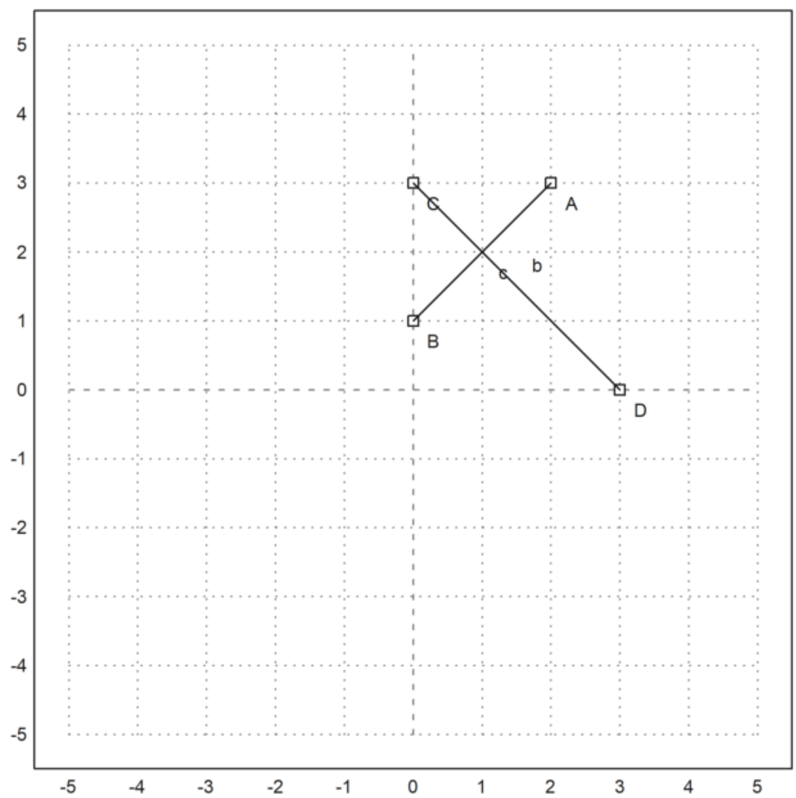
```
>plotSegment(A,B,"c"): // c=AB
```



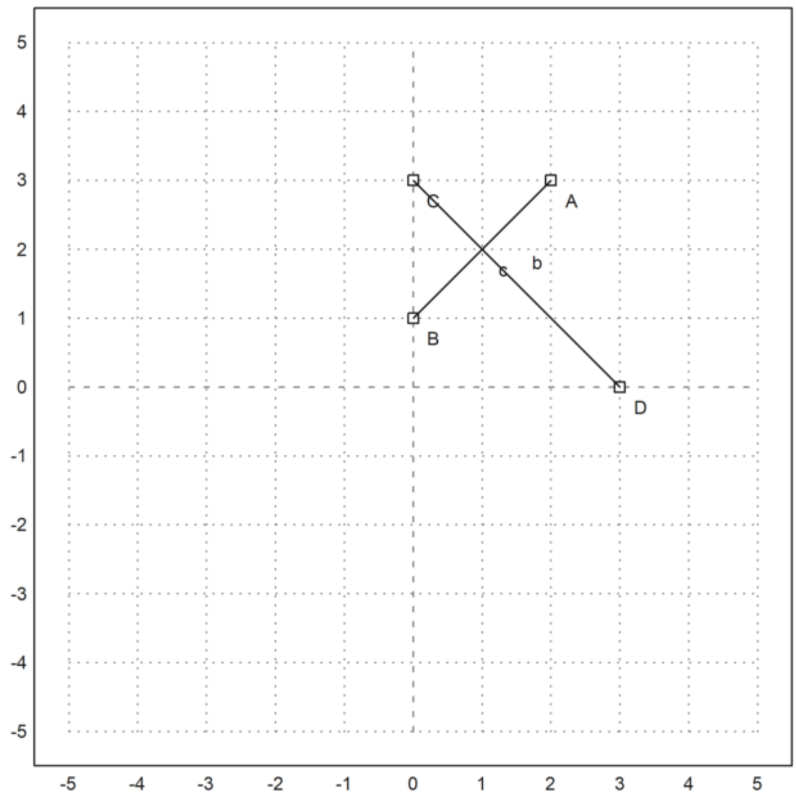

```
>plotSegment(C,D,"b"): // b=CD
```



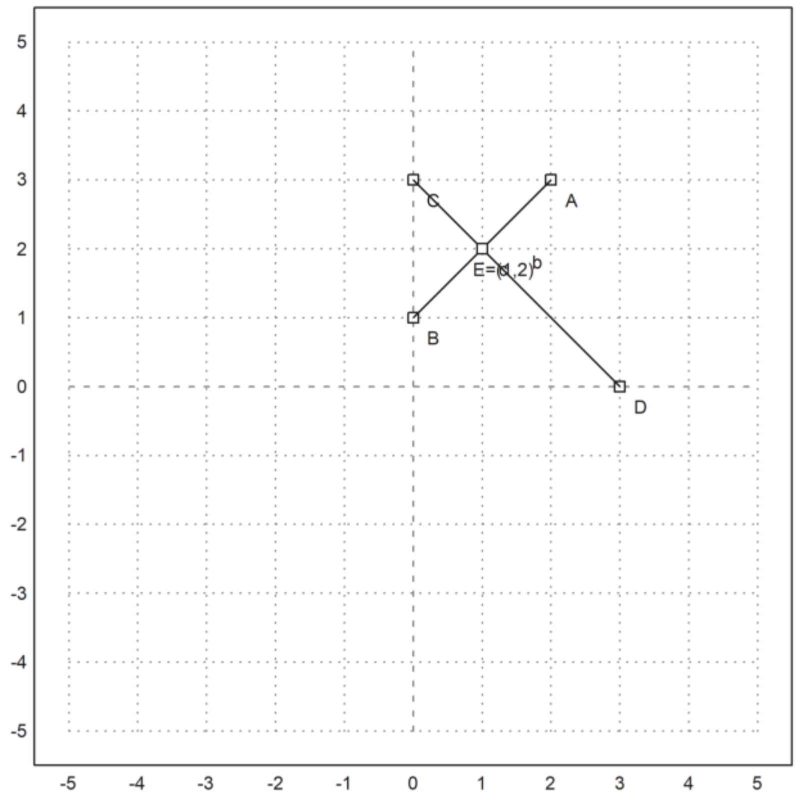

```
>lineThrough(A,B):
```



```
>lineThrough(C,D):
```




```
>E=lineIntersection(lineThrough(A,B),lineThrough(C,D));
>plotPoint(E,value=1):
```



Menentukan titik potong garis dan lingkaran

Titik potong antara garis dan lingkaran adalah titik di mana sebuah garis lurus memotong lingkaran. Untuk memahami konsep ini secara lengkap, perlu dipahami beberapa hal terkait dengan garis dan lingkaran.

Garis: Garis adalah himpunan tak terbatas titik yang terletak sepanjang jalur yang sama, dan garis ini dapat dinyatakan dalam berbagai bentuk persamaan matematis, seperti persamaan linier. Sebuah garis memiliki kemiringan (gradien) dan titik potong sumbu y (intersep) yang memungkinkan kita untuk menggambarkannya atau menganalisanya dalam sistem koordinat.

Lingkaran: Lingkaran adalah himpunan semua titik yang berjarak sama dari satu titik tertentu yang disebut sebagai pusat lingkaran. Jarak ini disebut sebagai jari-jari lingkaran. Persamaan matematis dari lingkaran adalah

$$(x-a)^2 + (y-b)^2 = r^2$$

di mana (a, b) adalah koordinat pusat lingkaran dan r adalah jari-jari lingkaran.

Ketika kita berbicara tentang titik potong antara garis dan lingkaran, ada beberapa kemungkinan:

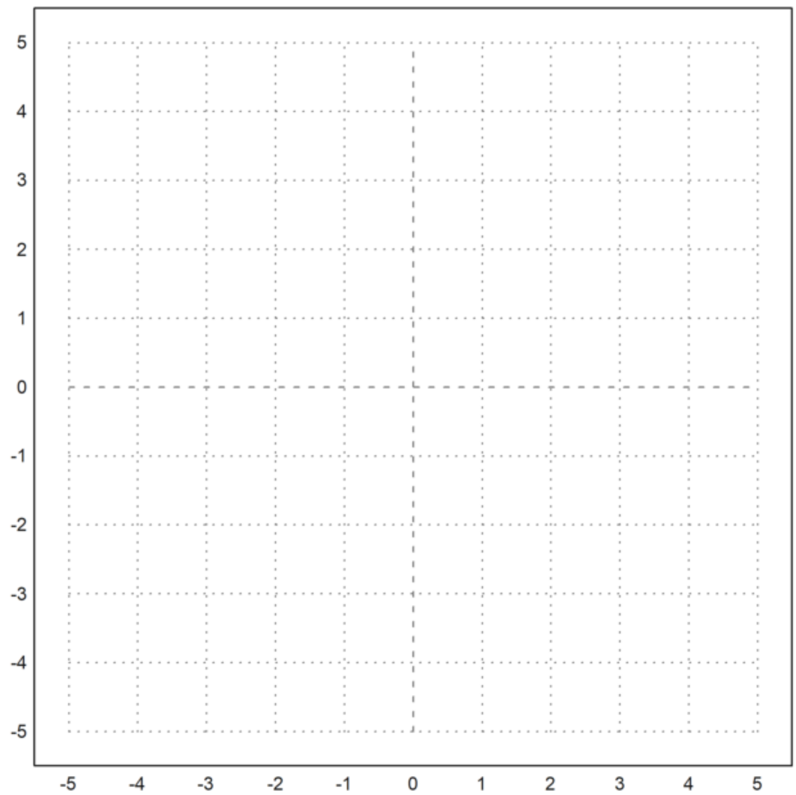
Tidak Ada Titik Potong: Garis tersebut mungkin tidak memotong lingkaran sama sekali jika jarak antara garis dan pusat lingkaran lebih besar dari jari-jari lingkaran.

Satu Titik Potong: Garis mungkin hanya memotong lingkaran di satu titik jika garis tersebut menyentuh lingkaran secara tepat pada satu titik, dengan jarak antara garis dan pusat lingkaran sama dengan jari-jari lingkaran.

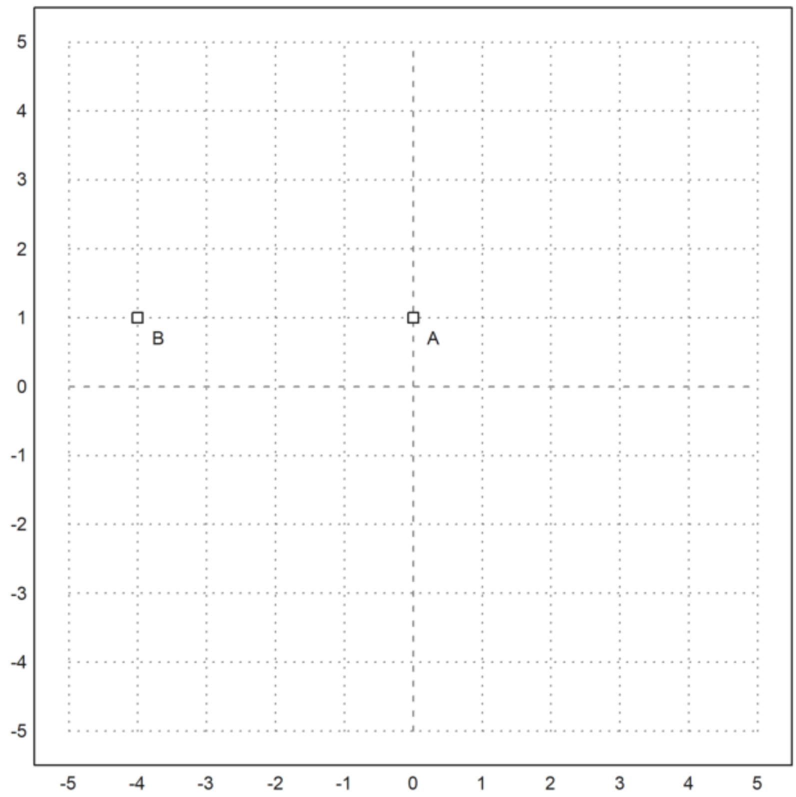
Dua Titik Potong: Garis bisa memotong lingkaran di dua titik berbeda jika garis melintasi lingkaran.

Untuk menentukan titik potong antara garis dan lingkaran, kita perlu menyelesaikan sistem persamaan antara persamaan garis (yang bisa berupa persamaan linier) dan persamaan lingkaran. Ini bisa menghasilkan titik potong yang merupakan koordinat di mana garis memotong lingkaran. Dalam kasus lingkaran, ini sering digunakan dalam masalah geometri dan trigonometri untuk menemukan titik potong dan berbagai sifat lainnya dari bangun geometri.

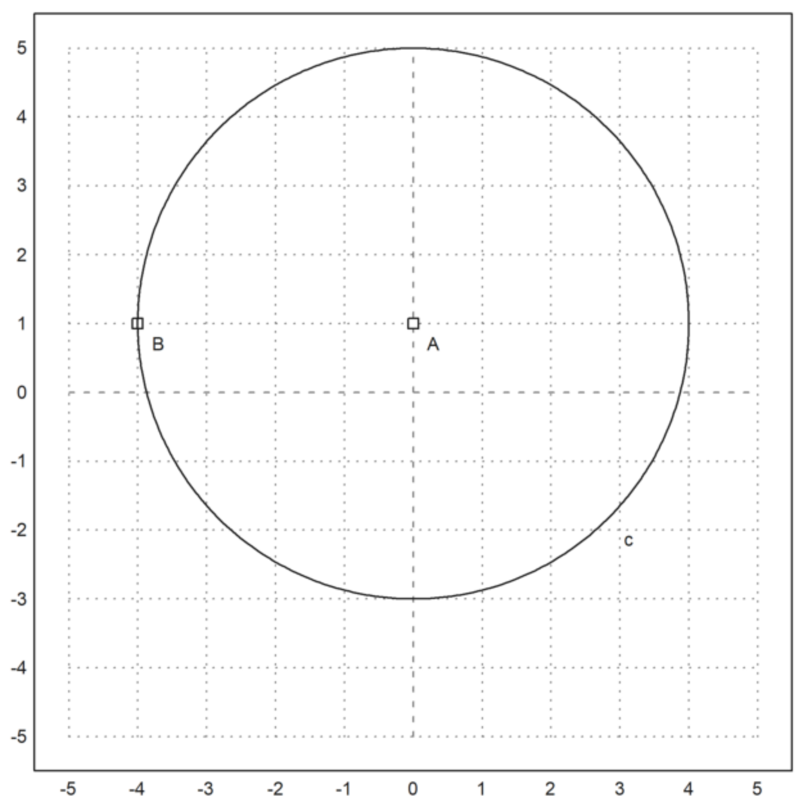
```
>setPlotRange(5):
```




```
>A=[0,1]; plotPoint(A,"A"); B=[-4,1]; plotPoint(B,"B"):
```




```
>c=circleWithCenter(A,distance(A,B)); plotCircle(c):
```



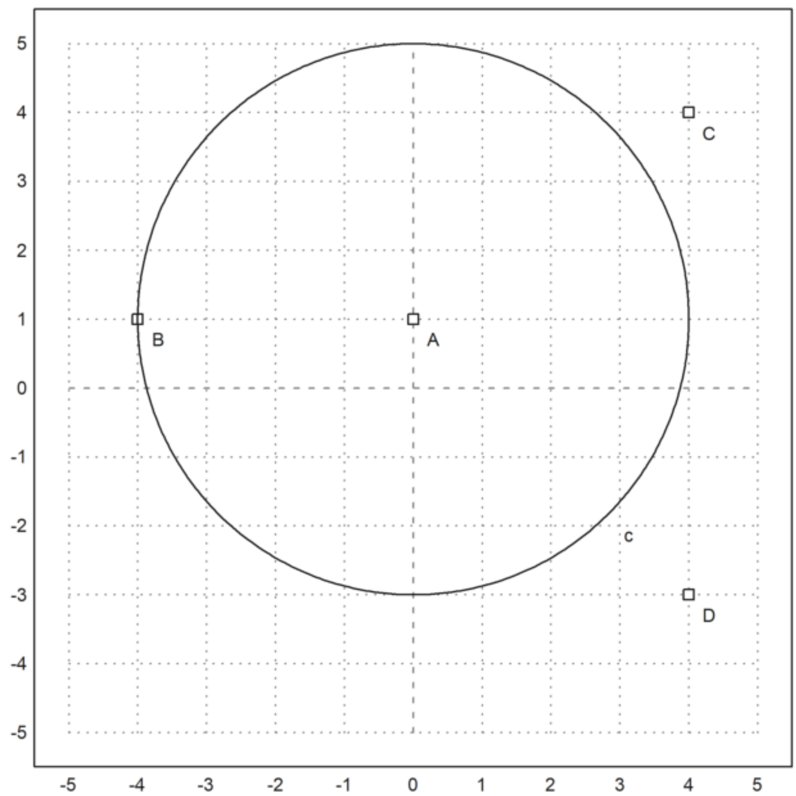
```
>A
```

```
[0,  1]
```

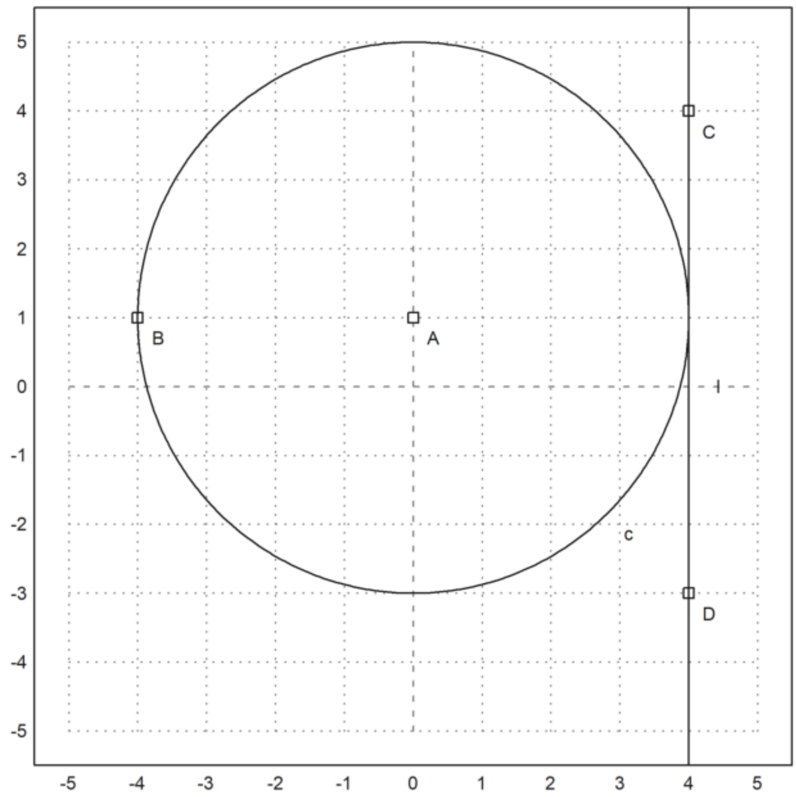
```
>distance(A,B)
```

```
4
```

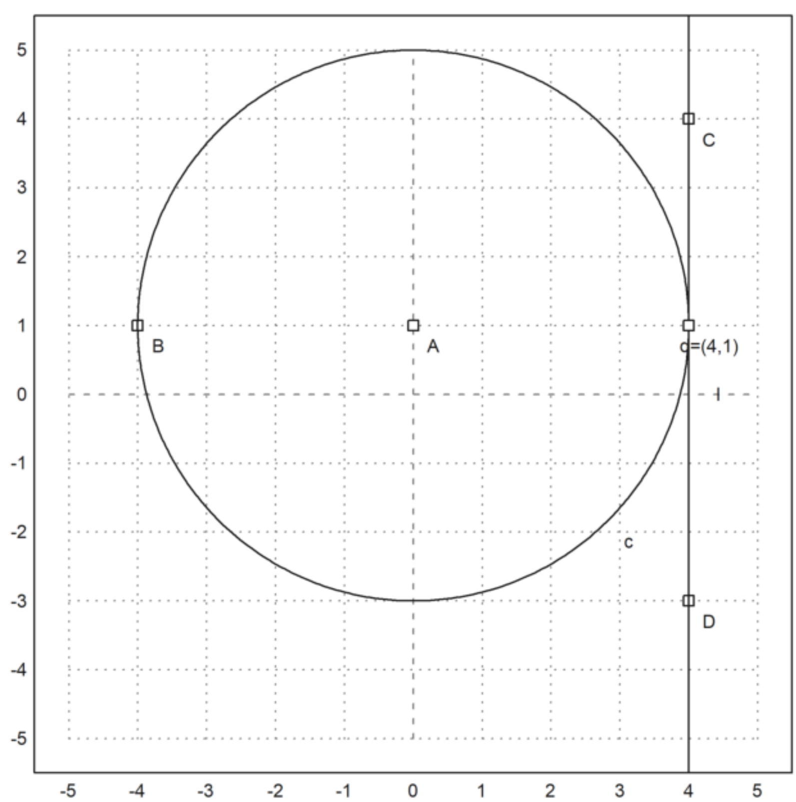
```
>C=[4,4]; plotPoint(C,"C"); D=[4,-3]; plotPoint(D,"D"):
```




```
>l=lineThrough(C,D); plotLine(l,"l"):
```




```
>q=lineCircleIntersections(l,c); plotPoint(q, value=1):
```



```
>C&=[4,4]
```

[4, 4]

```
>D&=[4,-3]
```

[4, - 3]

Pembuktian dengan matematika:
Diketahui:

$$r = 4, A(0, 1),$$

$$C(4, 4), D(4, -3)$$

Penyelesaian:
1. Cari persamaan lingkarannya

$$(x - a)^2 + (y - b)^2 = r^2$$

$$(x - 0)^2 + (y - 1)^2 = 4^2$$

$$x^2 + y^2 - 2y + 1 = 16$$

$$x^2 + y^2 - 2y - 15 = 0$$

2. Cari persamaan garis

$$\frac{y - 4}{(-3) - 4} = \frac{x - 4}{4 - 4}$$
$$\frac{y - 4}{-7} = \frac{x - 4}{0}$$

$$-7x + 28$$

$$x = 4$$

3. Subsitusi x=4 ke persamaan lingkaran

$$(4)^2 + y^2 - 2y - 15 = 0$$

$$16 + y^2 - 2y - 15 = 0$$

$$y^2 - 2y + 1 = 0$$

$$(y - 1)(y - 1) = 0$$

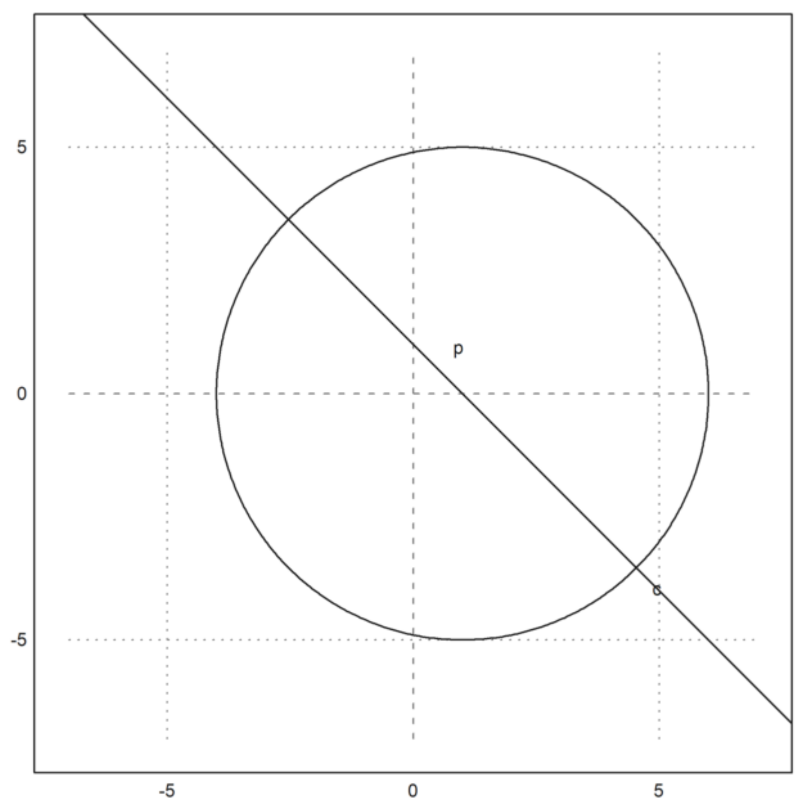
$$y = 1$$

Jadi, titik koordinat perpotongan garis dengan lingkarannya (4,1)
Terbukti **Latihan**

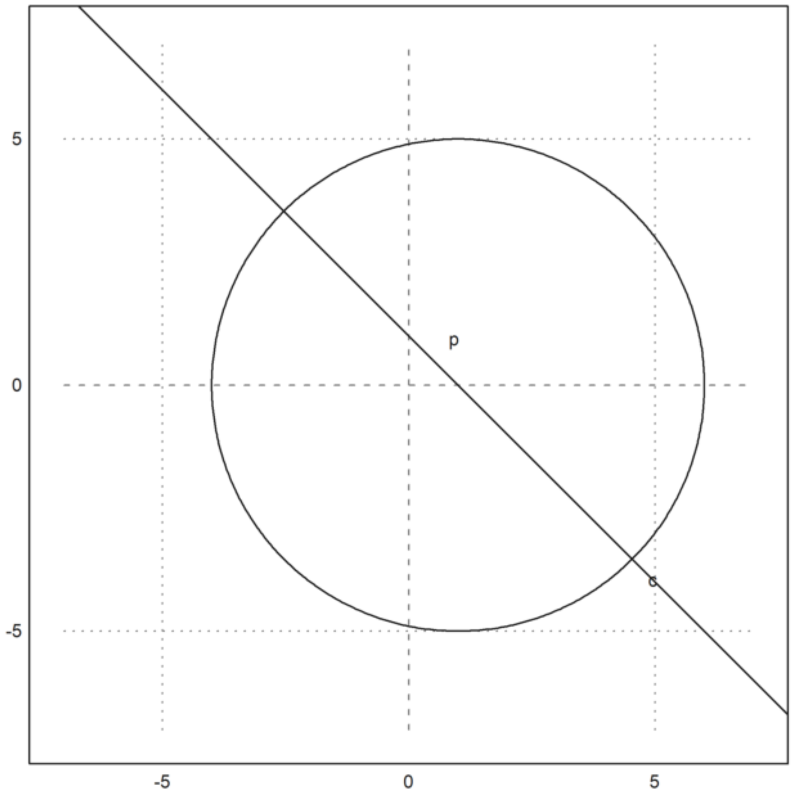
---

1. Diberikan lingkaran berjari jari 5 dengan titik pusat di J(1,0). diberikan dua titik M dan N dengan koordinat berturut-turut (-1,2) dan (3,-2). Apabila dibuat garis yang melalui titik M dan N serta memotong lingkran,tentukan koordinat titik potong tersebut!

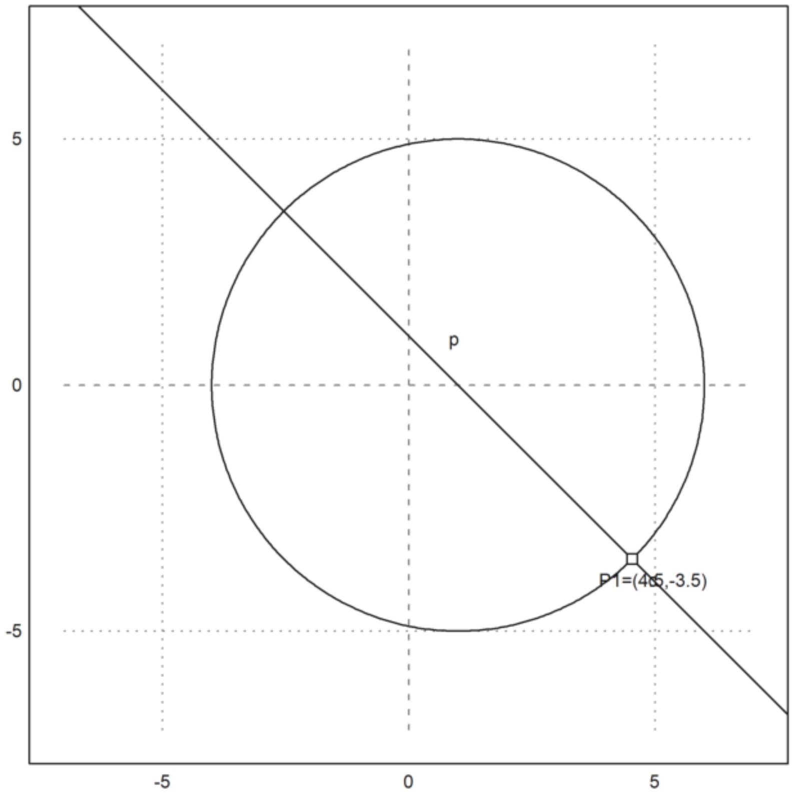
```
>J &:= [1,0]; plotPoint(J,"J"); c=circleWithCenter(J,5);
>M &:= [-1,2]; N &:= [3,-2]; l=lineThrough(M,N);
>setPlotRange(7); plotCircle(c); plotLine(l):
```



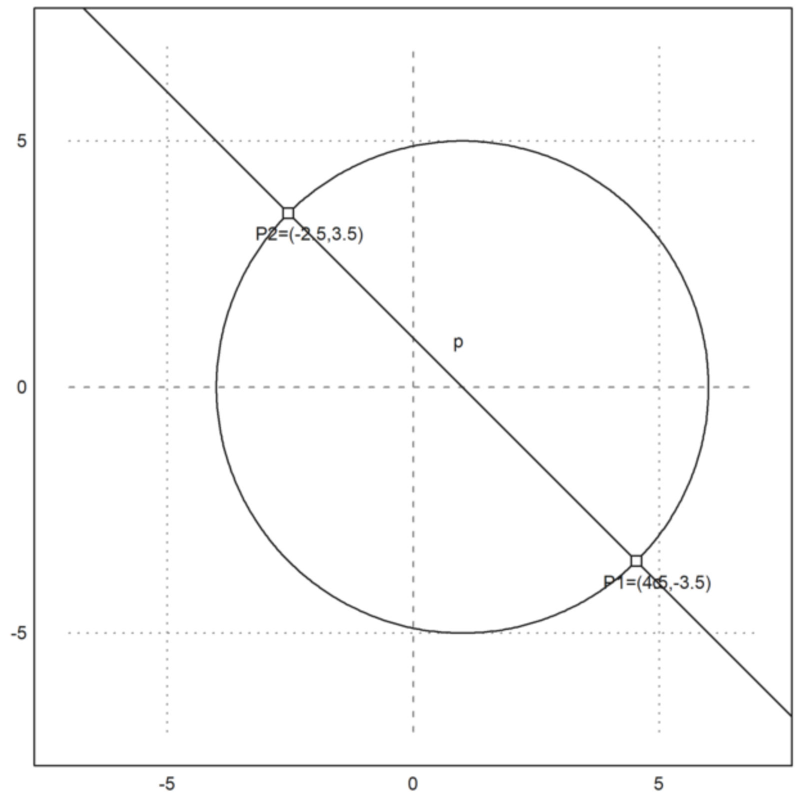

```
>{P1,P2,f}=lineCircleIntersections(l,c):
```




```
>P1; P2;
>plotPoint(P1, value=1):
```



```
>plotPoint(P2, value=1):
```



Menentukan titik potong dua lingkaran

Titik potong dua lingkaran adalah titik atau titik-titik di mana dua lingkaran berpotongan, yang berarti bahwa titik-titik tersebut terletak pada kedua lingkaran sekaligus. Titik potong ini dapat ditemukan jika dua lingkaran saling bersilangan, menyilangi, atau bersinggungan satu sama lain.

Dalam kasus titik potong dua lingkaran, ada beberapa skenario yang mungkin terjadi:

Dua Lingkaran Saling Bersilangan: Ini berarti bahwa dua lingkaran sepenuhnya tumpang tindih satu sama lain, dan titik potongnya akan ada di setiap titik tumpang tindih.

Dua Lingkaran Saling Menyilangi: Dalam situasi ini, dua lingkaran saling melintasi satu sama lain, tetapi tidak sepenuhnya tumpang tindih. Oleh karena itu, akan ada dua titik potong yang berbeda di mana kedua lingkaran berpotongan.

Dua Lingkaran Bersinggungan: Jika dua lingkaran bersinggungan, maka mereka hanya memiliki satu titik potong yang merupakan titik sentuh antara kedua lingkaran. Jika lingkaran tersebut bersinggungan di dalam, maka mereka akan memiliki satu titik potong di dalam satu lingkaran, dan jika bersinggungan di luar, maka titik potongnya berada di luar kedua lingkaran.

Untuk menentukan titik potong antara dua lingkaran, kita dapat menggunakan prinsip geometri dan aljabar. Secara matematis, kita dapat menyelesaikan sistem persamaan antara persamaan lingkaran pertama dan lingkaran kedua. Persamaan umum lingkaran adalah

$$(x-h)^2 + (y-k)^2 = r^2$$

di mana (h, k) adalah pusat lingkaran dan r adalah jari-jari lingkaran. Dengan menggabungkan persamaan lingkaran pertama dan kedua, kita dapat menemukan titik potongnya.

Dalam beberapa kasus, bisa tidak ada titik potong jika kedua lingkaran berada pada posisi yang tidak saling bersilangan, menyilangi, atau bersinggungan. Pemahaman tentang titik potong dua lingkaran penting dalam berbagai aplikasi matematika dan ilmu pengetahuan, terutama dalam geometri dan rekayasa.

CONTOH:

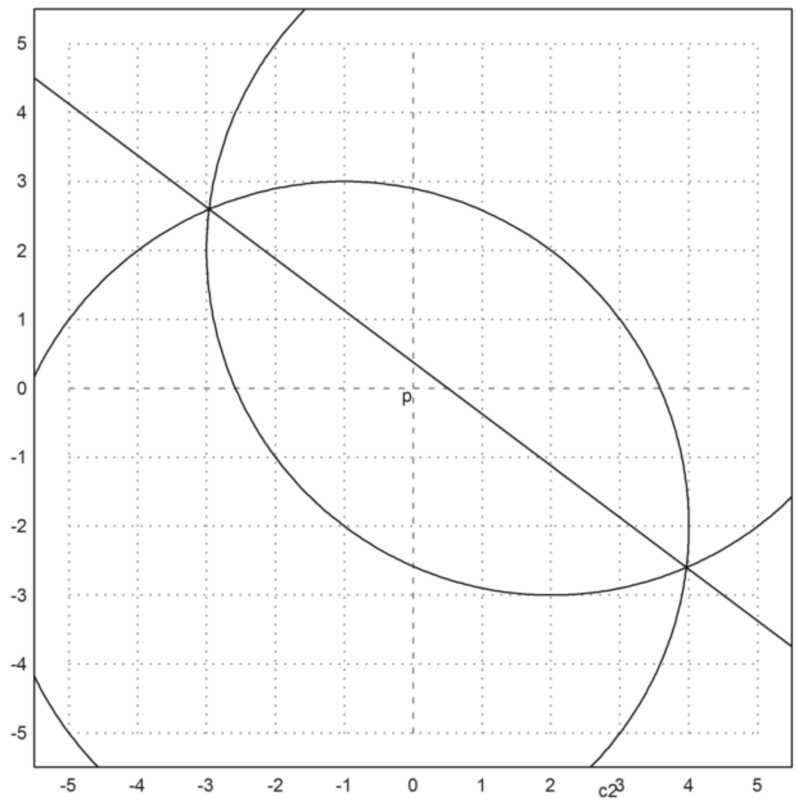
Diberikan 2 lingkaran berjari jari sepanjang ruas garis AB dengan titik A(2,2) merupakan titik pusat lingkaran petama dan titik B (-1,-2) merupakan titik pusat lingkaran kedua.kedua lingkaran tersebut berpotongan pada dua titik p1 dan p2. tentukan koordinat p1 dan p2!

langkah pertama, kita membuat 2 lingkaran tersebut

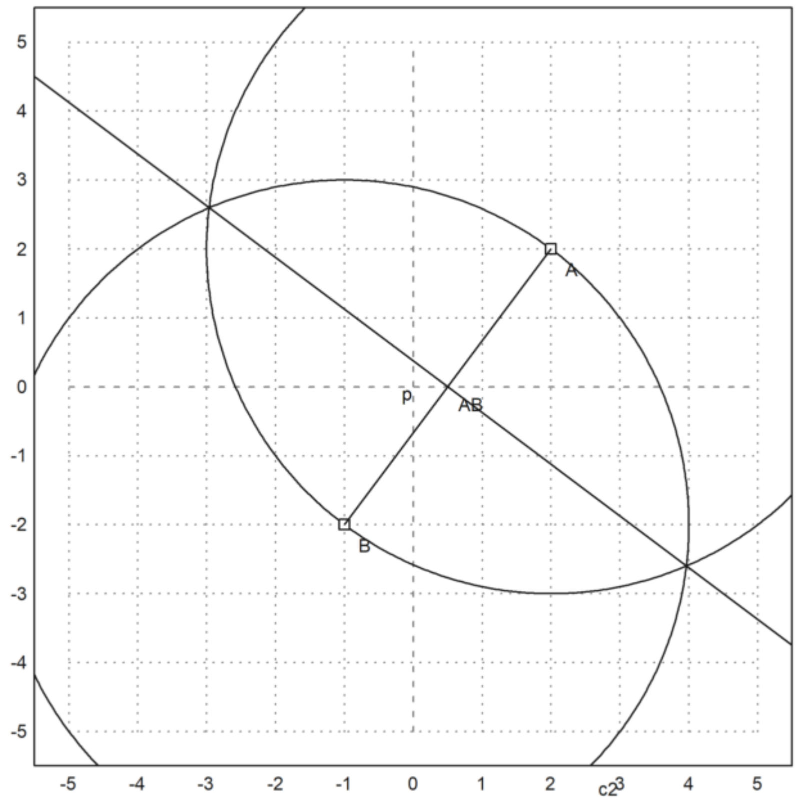
```
>A=[2,2]; B=[-1,-2];
>c1=circleWithCenter(A,distance(A,B)); //lingkaran 1 dengan titik pusat di A dan berjari-j
>c2=circleWithCenter(B,distance(A,B)); // lingkaran 2 dengan titik pusat di B dan berjari-
```

Berikutnya kita menggambar garis yang melalui perpotongan dua lingkaran tersebut

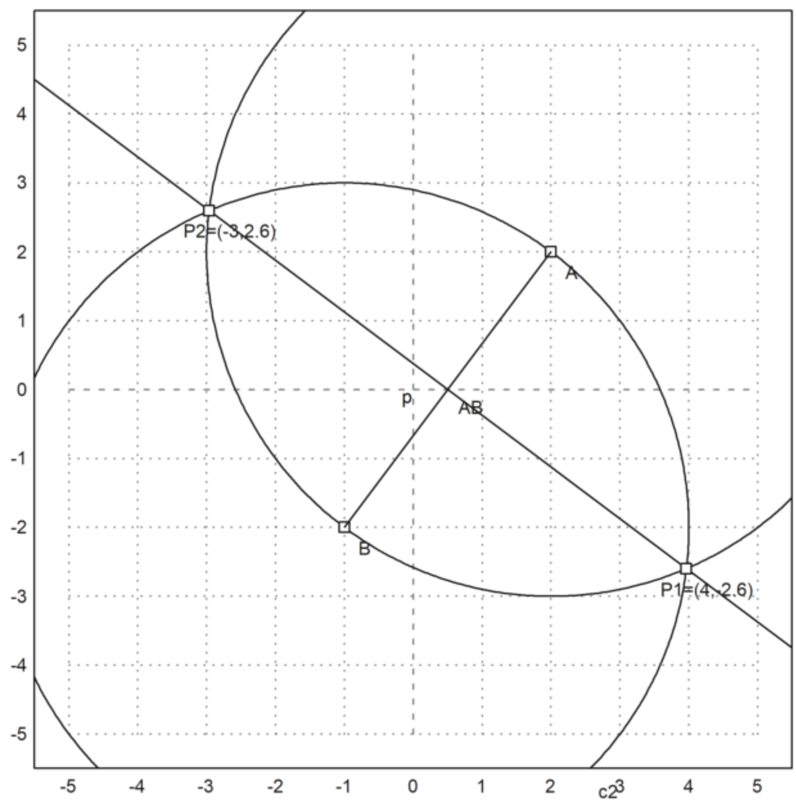
```
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(5); plotCircle(c1); plotCircle(c2);  plotLine(l):
```




```
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):
```




```
>plotPoint(P1,value=1); plotPoint(P2,value=1): // koordinat titik potong 2 lingkaran
```



```
>distance(A,B)
```

5

Pembuktian dengan rumus matematika:
1. Cari persamaan lingkaran 1

$$(x-2)^2+(y-2)^2=5^2$$

$$x^2-4x+4+y^2-4y+4=25$$

$$x^2+y^2-4x-4y-17=0$$

2. Cari persamaan lingkaran 2
(-1,-2)

$$(x-(-1))^2+(y-(-2))^2=5^2$$

$$(x+1)^2+(y+2)^2=5^2$$

$$x^2+2x+1+y^2+4y+4=25$$

$$x^2+y^2+2x+4y-20=0$$

3. Mengurangkan persamaan lingkaran 1 dengan lingkaran 2

$$(x^2+y^2-4x-4y-17=0)-(x^2+y^2+2x+4y-20=0)$$

$$-6x-8y+3=0$$

$$x=-\frac{8}{6}y+\frac{1}{2}$$

4. Subsitusi x ke persamaan lingkaran 2

$$x^2+y^2-2x+4y-20=0$$

$$(-\frac{8}{6}y+\frac{1}{2})^2+y^2-2(-\frac{8}{6}y+\frac{1}{2})+4y-20=0$$

$$\frac{64}{36}y^2-\frac{16}{12}y+\frac{1}{4}+y^2+\frac{8}{3}y-1+4y-20=0$$

$$64y^2-48y+9+36y^2+96y-36+144y-20=0$$

$$100y^2+192y-47=0$$

dengan menggunakan rumus ABC:

$$\frac{b^+_-\sqrt{b^2-4ac}}{2a}$$

$$\frac{192^+_-\sqrt{192^2-4(100)(-47)}}{2(100)}$$

$$\frac{192 \pm \sqrt{36864 + 18800}}{200}$$

$$\frac{192 \pm \sqrt{55664}}{200}$$

$$\frac{192 \pm 235,9}{200}$$

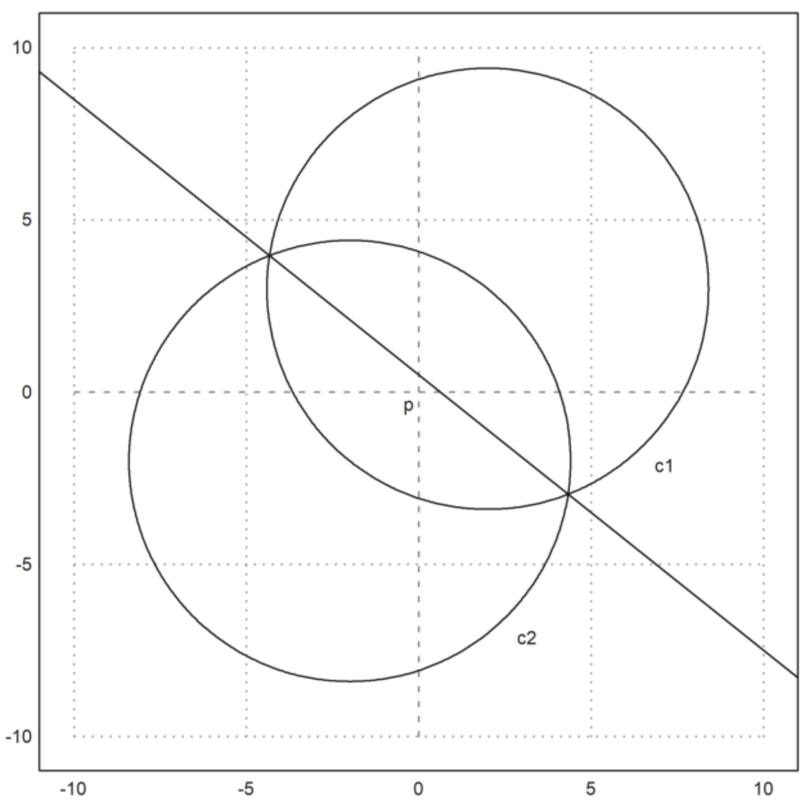
$$\left[y1 = \frac{192 + 235,9}{200}, y2 = \frac{192 - 235,9}{200}\right]$$

$$\left[y1 = \frac{427,9}{200}, y2 = \frac{-43,9}{200}\right]$$

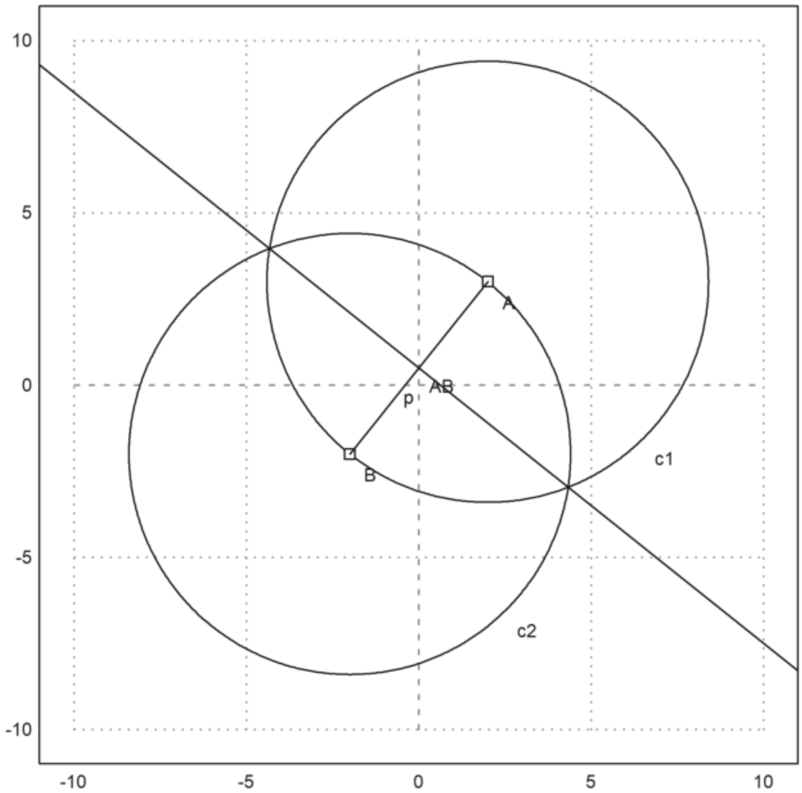
Latihan

1. Diberikan 2 lingkaran berjari jari sepanjang ruas garis AB dengan titik A(2,3) dan titik B (-2,-2).kedua lingkaran tersebut berpotongan pada dua titik p1 dan p2. tentukan koordinat p1 dan p2!

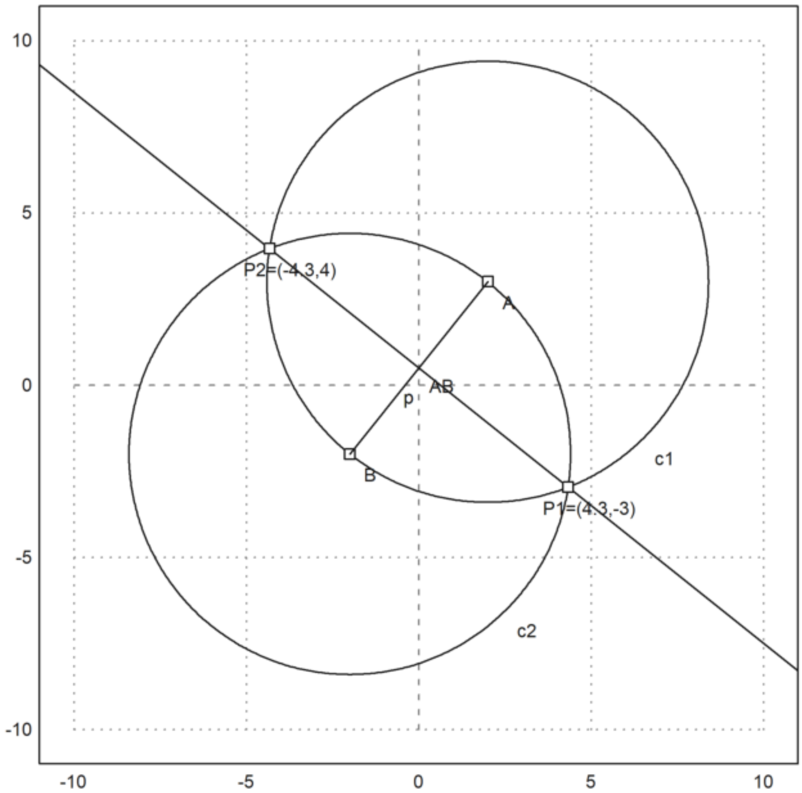
```
>A=[2,3]; B=[-2,-2];
>c1=circleWithCenter(A,distance(A,B)); //lingkaran 1 dengan titik pusat di A dan berjari-j
>c2=circleWithCenter(B,distance(A,B)); // lingkaran 2 dengan titik pusat di B dan berjari-
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(10); plotCircle(c1); plotCircle(c2);  plotLine(l):
```



```
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):
```




```
>plotPoint(P1,value=1); plotPoint(P2,value=1): // koordinat titik potong 2 lingkaran
```



MATERI 5

a. Menentukan persamaan garis yang melalui dua titik
b. Menentukan persamaan Garis Sumbu suatu ruas garis
c. Menentukan persamaan Garis Bagi sudut
d. Menentukan persamaan Garis Berat
e. Menentukan persamaan Garis Tinggi

Menentukan Persamaan Garis yang Melalui Dua Titik

Dengan konsep matematika, persamaan garis yang diketahui dua titiknya misalnya titik

$$P_1(x_1, y_1) dan P_2(x_2, y_2).$$

Akan dicari persamaa garis l yang melalui dua titik tersebut. Ambil sebarang titik P(x,y) pada l. Maka persamaan garis l dapat ditentukan dari

$$mPP_1 = mP_1P_2$$

yaitu:

$$\frac{(y - y_1)}{(x - x_1)} = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

atau

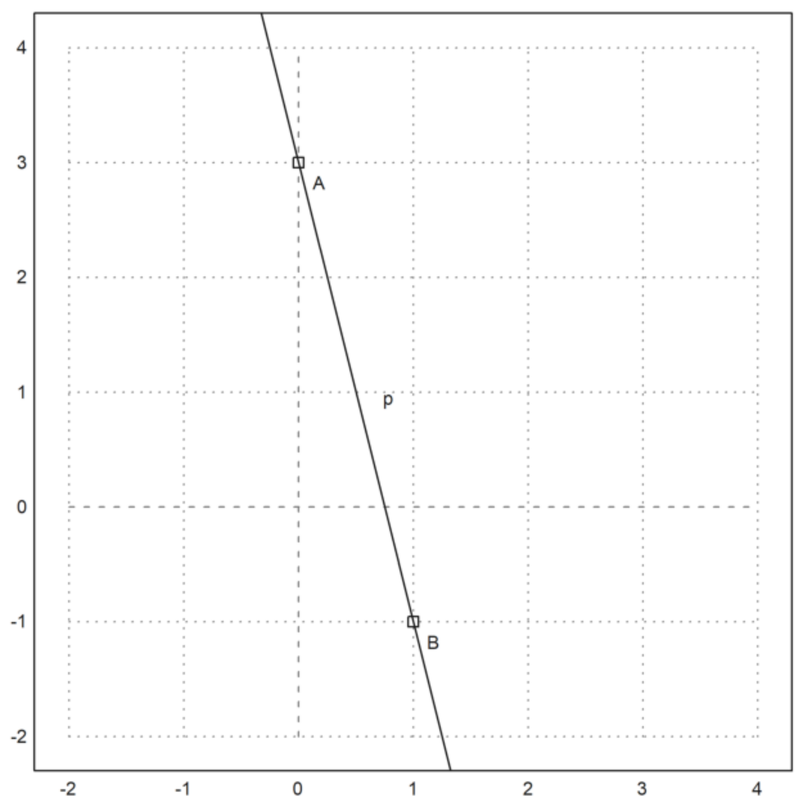
$$\frac{(y - y_1)}{(y_2 - y_1)} = \frac{(x - x_1)}{(x_2 - x_1)}$$

Dengan EMT persamaan garis yang melalui dua titik dapat dicari dengan cara mendefinisikan kedua titik koordinat kemudian menggunakan fungsi lineThrough(titik1, titik2), lalu menggunakan fungsi perintah getLineEquation(lineThrough(P1,P2),x,y).
Contoh soal :
1. tentukan persamaan garis yang melalui titik A(0,3) dan B(1,-1)

```
>load geometry
```

```
Numerical and symbolic geometry.
```

```
>setPlotRange(-2,4,-2,4); // mendefinisikan bidang koordinat baru
>A=[0,3]; plotPoint(A,"A");
>B=[1,-1]; plotPoint(B,"B");
>AB=lineThrough(A,B); plotLine(AB):
```



Mencari persamaan garis AB

```
>A&=[0,3]; B&=[1,-1];
>$getLineEquation(lineThrough(A,B),x,y)
```

$$y + 4\,x = 3$$

```
>$solve(%,y)|expand
```

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

Untuk membuktikan hasil tersebut kita dapat menghitung dengan rumus:

$$\frac{(y-y_1)}{(y_2-y_1)} = \frac{(x-x_1)}{(x_2-x_1)}$$

maka didapatkan

$$\frac{(y-3)}{(-1-3)} = \frac{(x-0)}{(1-0)}$$
$$\frac{(y-3)}{(-4)} = \frac{(x)}{(1)}$$
$$(y-3)(1) = (-4)(x)$$
$$y-3 = -4x$$
$$4x + y = 3$$
$$y + 4x = 3$$

jadi terbukti bahwa persamaan garisnya

$$y + 4x = 3$$

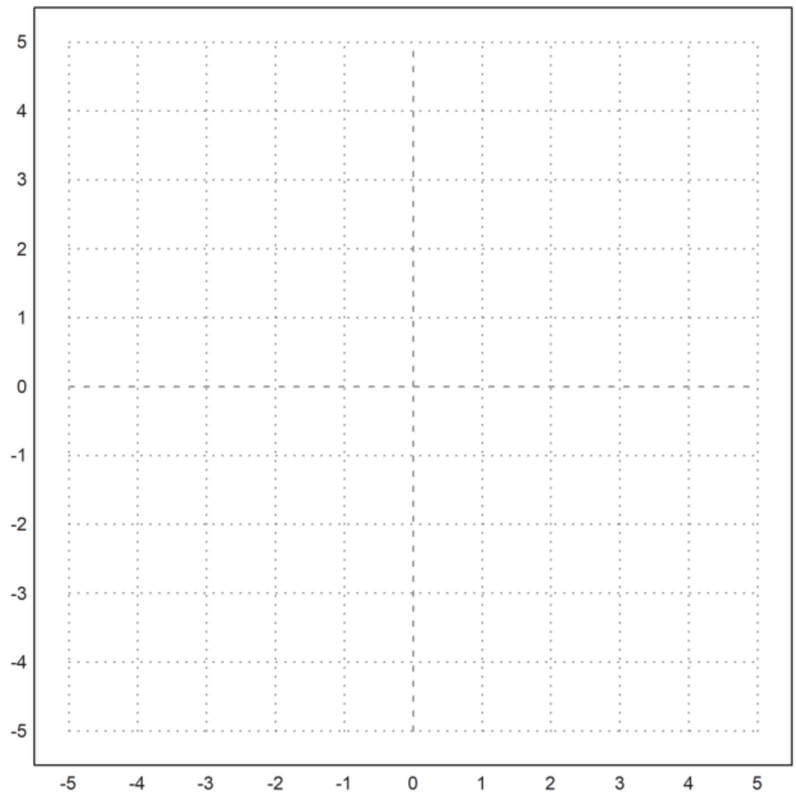
Latihan A

1. Tentukan persamaan garis yang melalui P[-3,2] dan Q[4,1]!

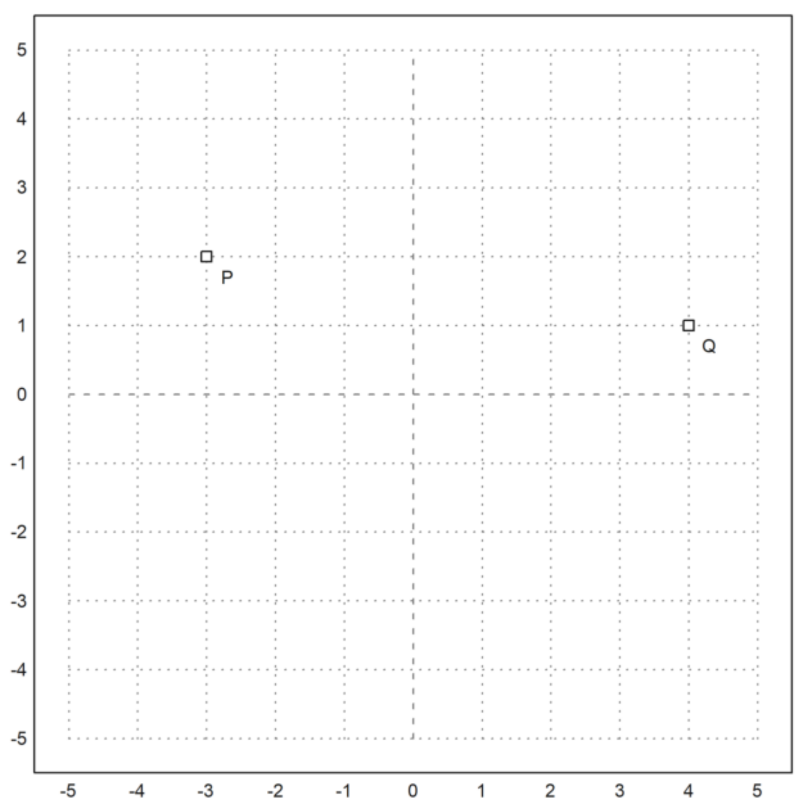
```
>load geometry
```

```
Numerical and symbolic geometry.
```

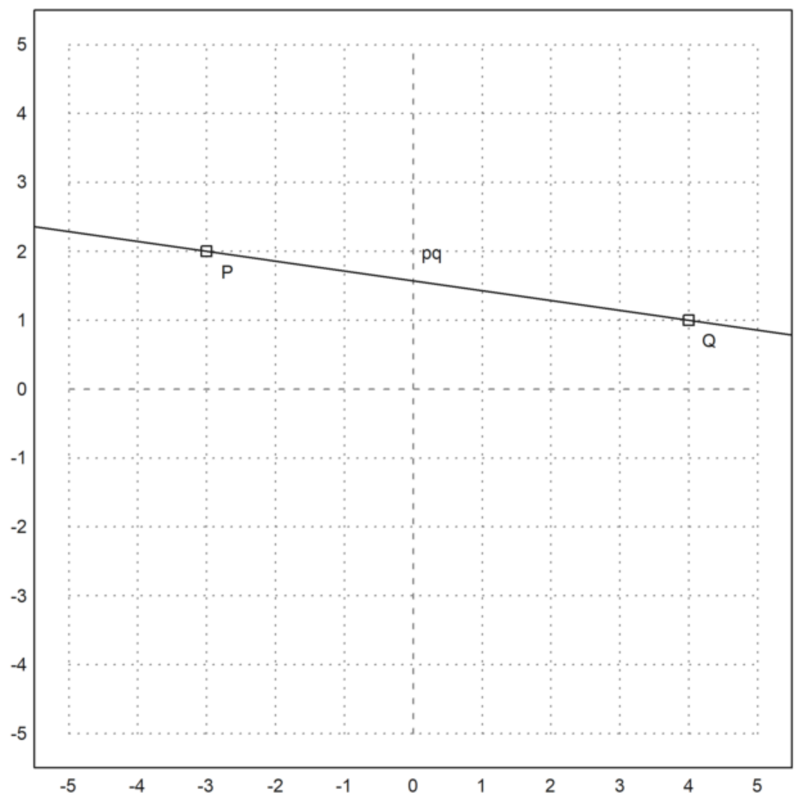
```
>setPlotRange(5):
```




```
>P=[-3,2]; Q=[4,1]; plotPoint(P,"P"); plotPoint(Q,"Q"):
```



```
>plotLine(lineThrough(P,Q),"pq"):
```




```
>P&=[-3,2]
```

[- 3, 2]

```
>Q&=[4,1]
```

[4, 1]

```
>$getLineEquation(lineThrough(P,Q),x,y)
```

$$7\,y+x=11$$

```
>$solve(%,y) | expand
```

$$\left[y=\frac{11}{7}-\frac{x}{7}\right]$$

2. Tentukan persamaan garis yang melalui

$$A(5,1) dan B(-2,-7)$$

```
>A&=[5,1]
```

[5, 1]

```
>B&=[-2,-7]
```

[- 2, - 7]

```
>$getLineEquation(lineThrough(A,B),x,y)
```

$$8\,x-7\,y=33$$

```
>$solve(%,y) | expand
```

$$\left[y=\frac{8\,x}{7}-\frac{33}{7}\right]$$

3. Tentukan persamaan garis RS dan TU dengan

$$R(-5,6) S(6,8) T(2,0) U(8,9)$$

Apakah kedua garis tersebut sejajar?

```
>R&=[-5,6]
```

[- 5, 6]

```
>S&=[6,8]
```

[6, 8]

```
>T&=[2,0]
```

[2, 0]

```
>U&=[8,9]
```

[8, 9]

```
>$getLineEquation(lineThrough(R,S),x,y)
```

$$11\,y-2\,x=76$$

```
>$solve(%,y) | expand
```

$$\left[y=\frac{2\,x}{11}+\frac{76}{11}\right]$$

```
>$getLineEquation(lineThrough(S,T),x,y)
```

$$8\,x-4\,y=16$$

```
>$solve(%,y) | expand
```

$$[y=2\,x-4]$$

karena

$$m(RS)=\frac{2x}{11}$$

$$m(TU)=2x$$

maka, garis RS dan garis TU tidak sejajar.
Jadi, persamaan garis RS:

$$y=\frac{2x}{11}+\frac{76}{11}$$

persamaan garis TU:

$$y=2x-4$$

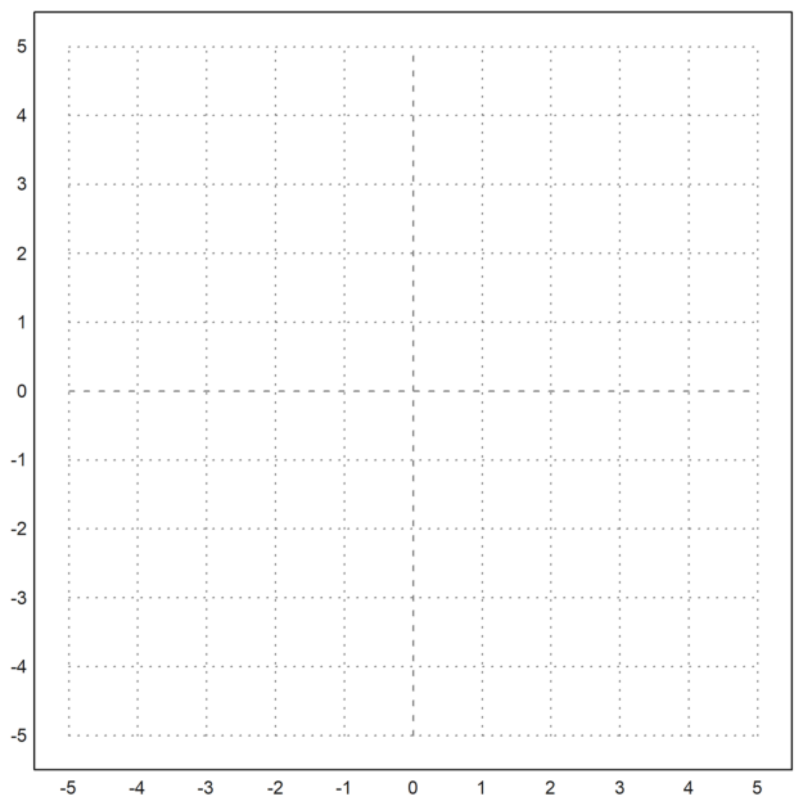
dan garis RS dan garis TU tidak sejajar. **Menentukan persamaan Garis sumbu suatu ruas garis**

Garis sumbu dalam sebuah segitiga adalah garis lurus yang menghubungkan satu titik pada segitiga dengan sisi dihadapannya dan membagi sisi tersebut menjadi dua bagian sama panjang secara tegak lurus.
Dengan menggunakan EMT kita dapat mencari persamaan garis sumbu suatu garis dengan cara mendefinisikan titik koordnat, menghubungkan titik sehingga membentuk segitiga, menggunakan middlePerpendicular untuk mencari titik tengah lalu mendefinisikan titik tengah tersebut menjadi garis sumbu kemudian mencari persamaan garis sumbu dengan perintah $getLineEquation(garis sumbu,x,y).
Dengan rumus/konsep matematika, kita dapat memperoleh persamaan Garis sumbu dengan cara mencari persamaan dan gradien garis yang berlawanan, mencari koordinat titik potong, lalu mencari persamaan garis sumbu dengan gradien dan titik koordinatnya.
Contoh terdapat

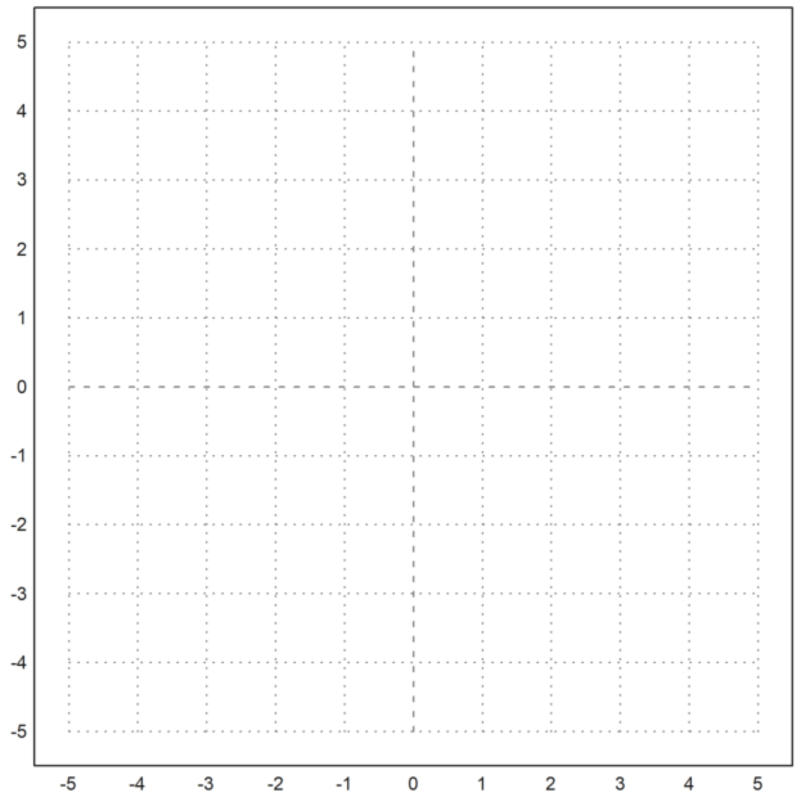
$$\triangle ABC,\ dengan\ A(2,5);\ B(-5,-4);\ dan\ C(4,-4),$$

Tentukan persamaan garis sumbu dari ruas garis BC!

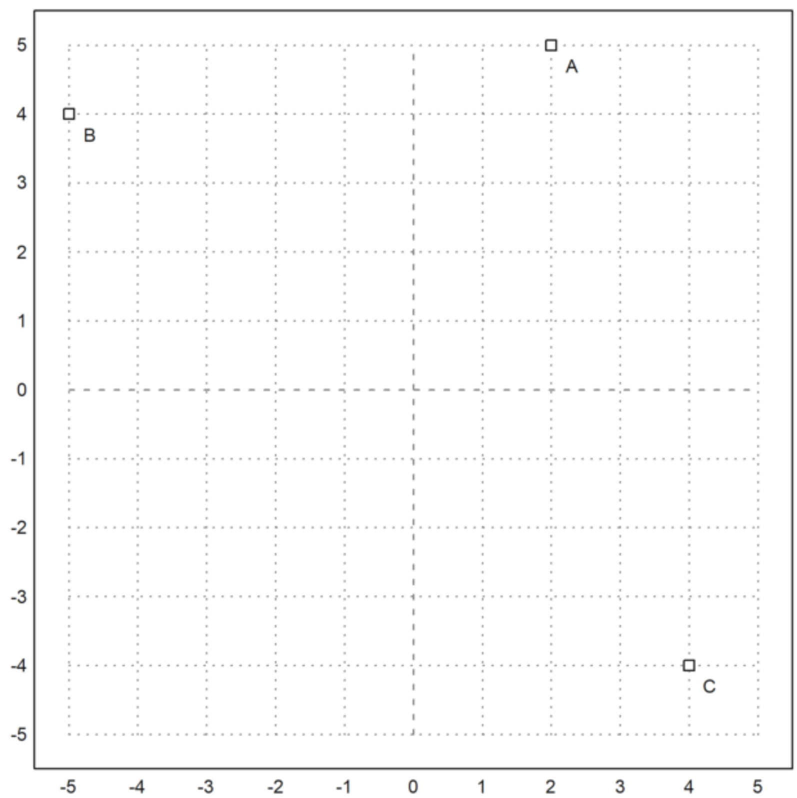
```
>setPlotRange(5):
```



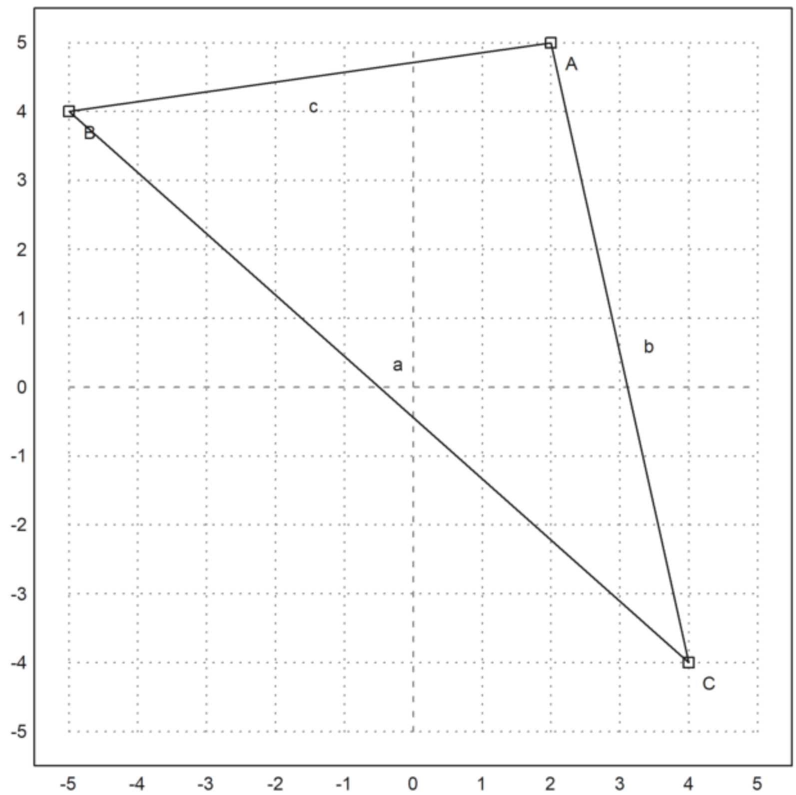
```
>A=[2,5]; B=[-5,4]; C=[4,-4]:
```



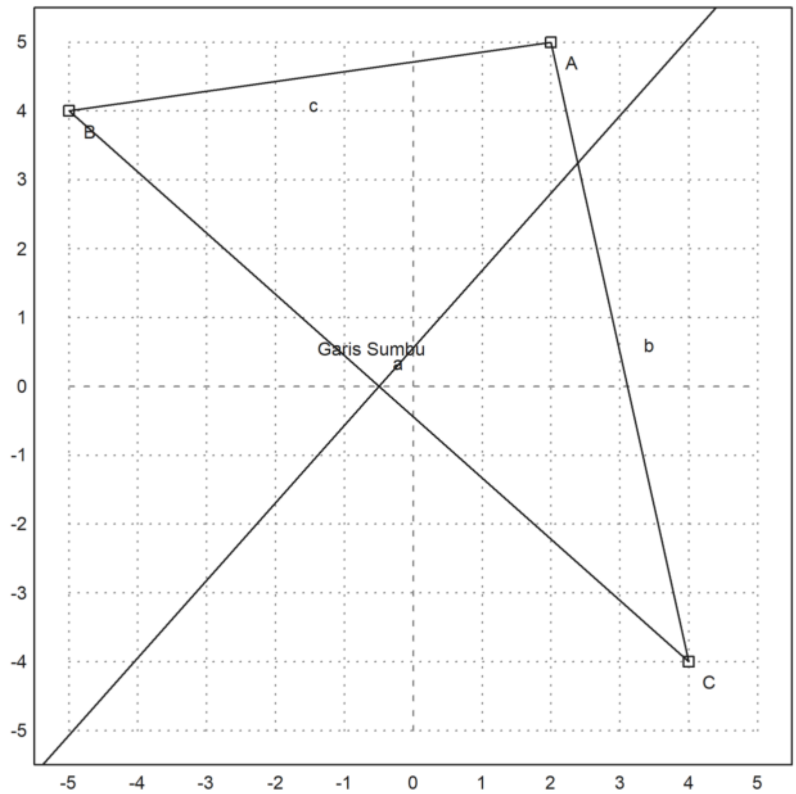

```
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"):
```



```
>plotSegment(A,B,"c"); plotSegment(B,C,"a"); plotSegment(A,C,"b"):
```




```
>s=middlePerpendicular(B,C); plotLine(s,"Garis Sumbu"):
```




```
>B&=[-5,4]
```

```
[- 5, 4]
```

```
>C&=[4,-4]
```

```
[4, - 4]
```

```
>s&=middlePerpendicular(B,C)
```

```
              9
[- 9, 8, -]
              2
```

```
>$getLineEquation(s,x,y)
```

$$8\,y-9\,x=\frac{9}{2}$$

```
>$solve(%,y) | expand
```

$$\left[y=\frac{9\,x}{8}+\frac{9}{16}\right]$$

Pembuktian dengan menggunakan rumus matematika:
1. Mencari persamaan garis BC

$$\frac{y-y_1}{y_2-y_1}=\frac{x-x_1}{x_2-x_1}$$
$$\frac{y-4}{(-4)-4}=\frac{x-(-5)}{4-(-5)}$$
$$\frac{y-4}{-8}=\frac{x+5}{9}$$

$$9y-36=-8x-40$$

$$9y=-8x-4$$

$$y=\frac{-8x}{9}-\frac{4}{9}$$

maka gradiennya

$$m=\frac{-8}{9}$$

karena tegak lurus dengan garis sumbu, maka gradien garis sumbu:

$$m=\frac{9}{8}$$

2. Mencari titik koordinat dari titik potong

$$\frac{x_1+x_2}{2},\frac{y_1+y_2}{2}$$
$$\frac{(-5)+4}{2},\frac{4+(-4)}{2}$$
$$\frac{-1}{2},0$$

Jadi, titik koordinatnya

$$(\frac{-1}{2},0)$$

3. Mencari persamaan garis sumbu
dengan gradien:

$$m=\frac{9}{8}$$

dan melalui titik:

$$(\frac{-1}{2},0)$$

$$(y-y_1)=m(x-x_1)$$

$$(y-0)=\frac{9}{8}(x-\frac{-1}{2})$$

$$y=\frac{9x}{8}-\frac{9}{16}$$

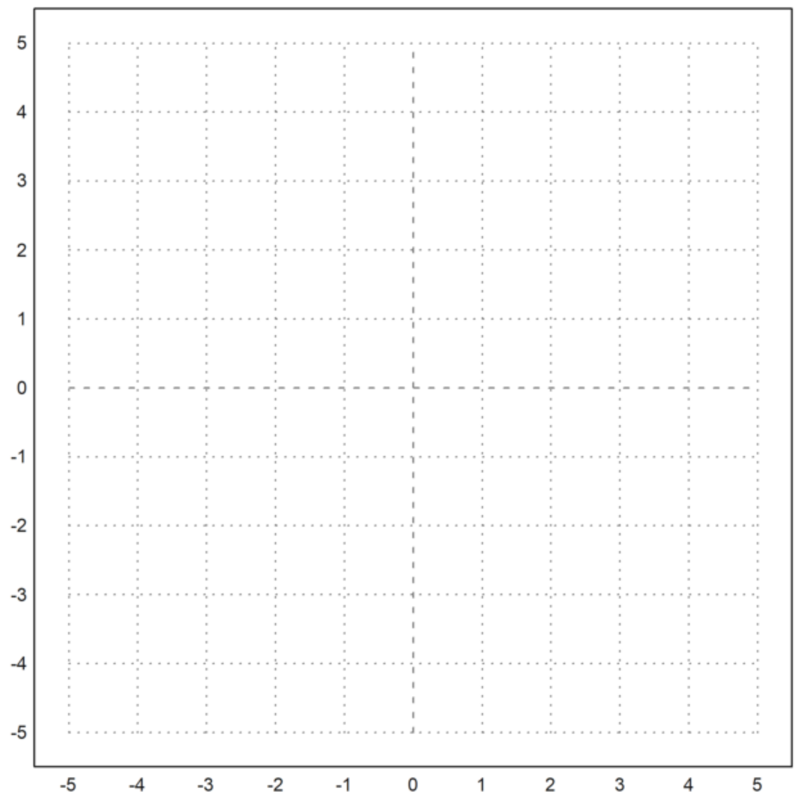
Terbukti **Latihan B**

---

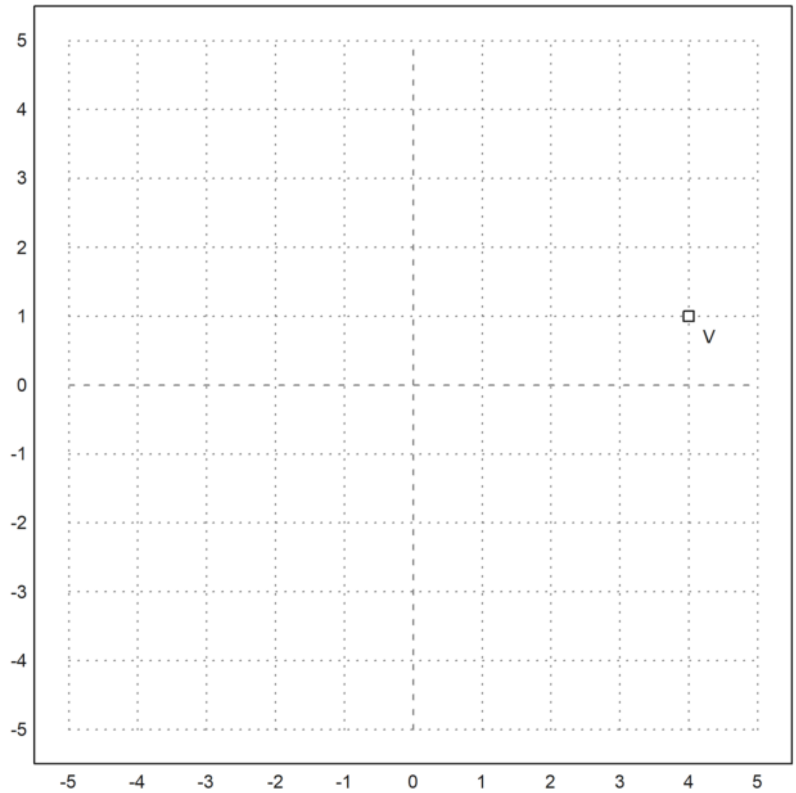
1. Tentukan persamaan garis sumbu

$$V(4,1)W(-3,5)$$

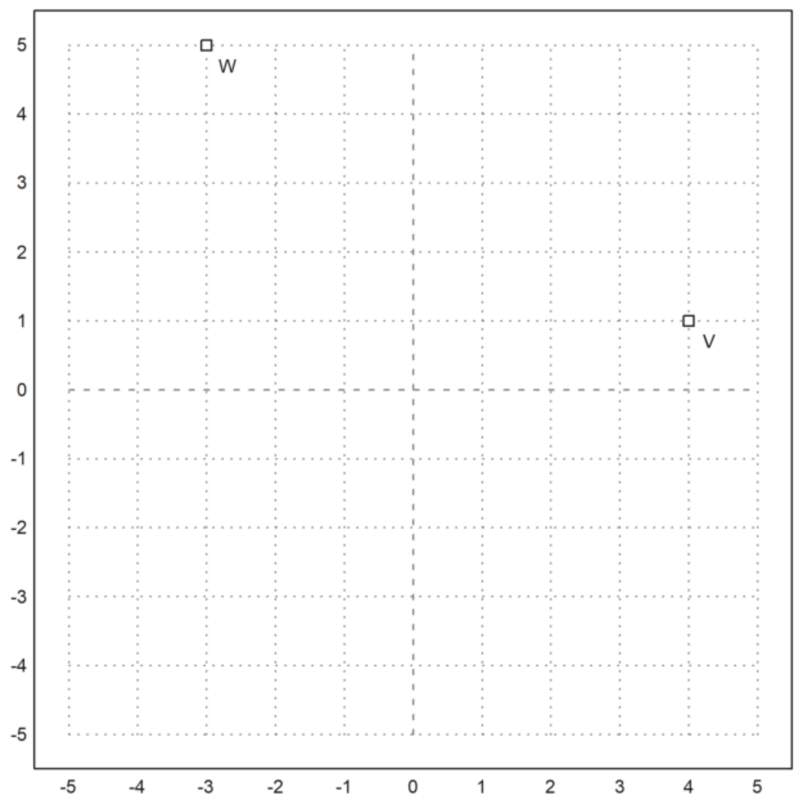
```
>setPlotRange(5):
```



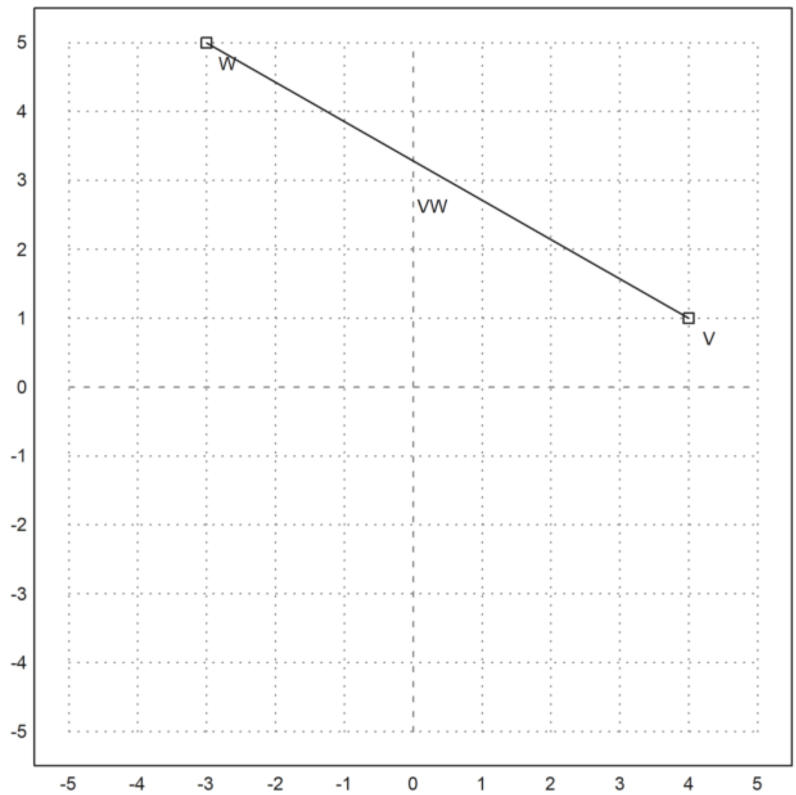
```
>V=[4,1]; plotPoint(V):
```



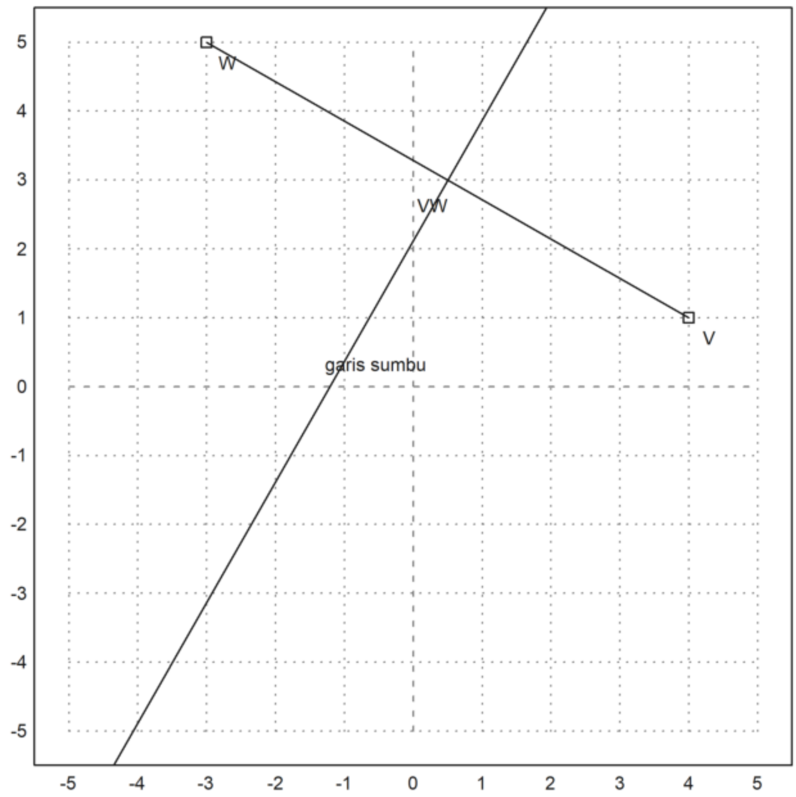

```
>W=[-3,5]; plotPoint(W):
```



```
>plotSegment(V,W,"VW"):
```




```
>p=middlePerpendicular(V,W); plotLine(p,"garis sumbu"):
```




```
>V&=[4,1]
```

```
[4, 1]
```

```
>W&=[-3,5]
```

```
[- 3, 5]
```

```
>p&=middlePerpendicular(V,W)
```

```
                17
[7, - 4, - --]
                2
```

```
>$getLineEquation(p,x,y); $solve(%,y) | expand
```

$$\left[y=\frac{7\,x}{4}+\frac{17}{8}\right]$$

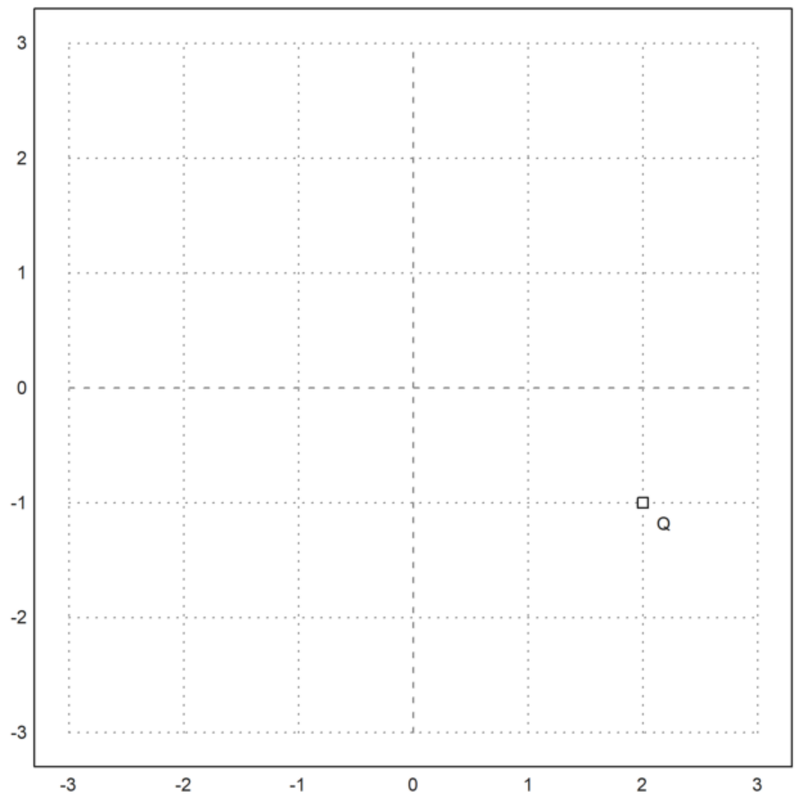
2. Tentukan garis sumbu dari

$$Q(2,-1)R(-1,2)$$

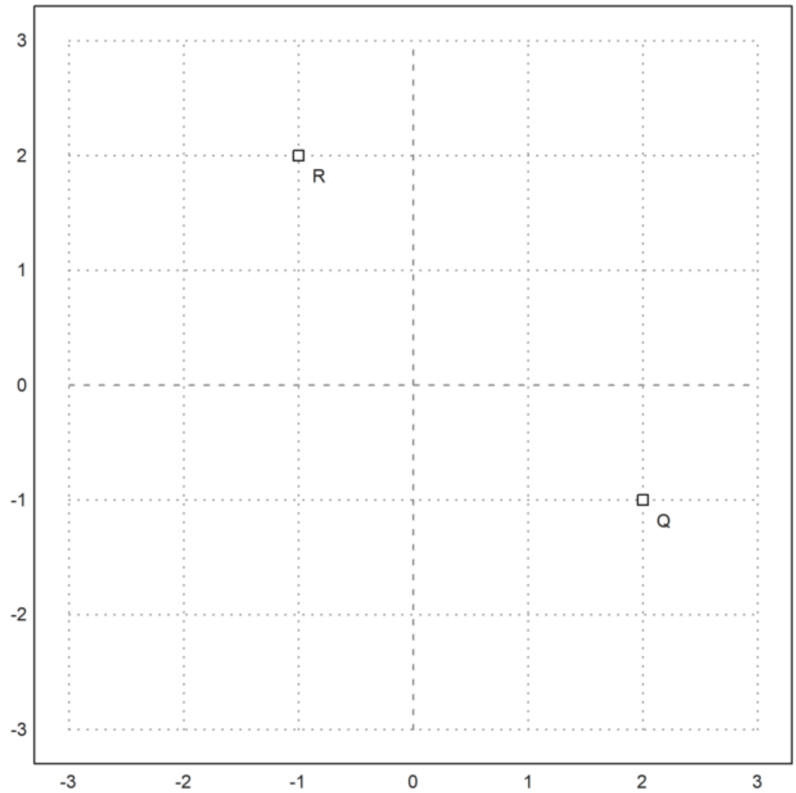
Apakah garis sumbu tersebut sejajar dengan

$$y=3x+\frac{19}{7}$$

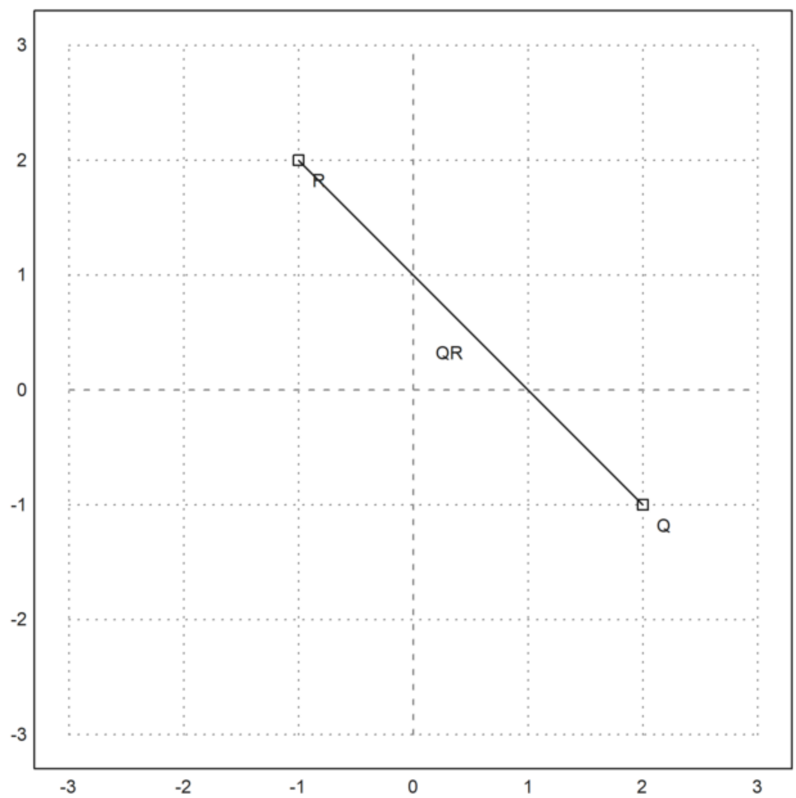
```
>setPlotRange(3);
>Q=[2,-1]; plotPoint(Q):
```



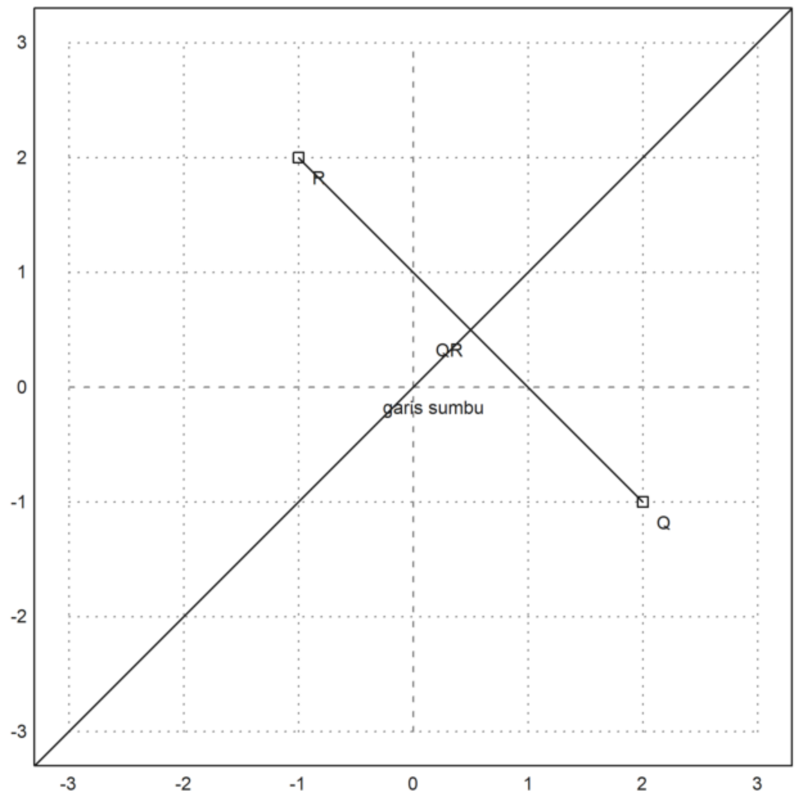

```
>R=[-1,2]; plotPoint(R):
```




```
>plotSegment(Q,R,"QR"):
```



```
>p=middlePerpendicular(Q,R); plotLine(p,"garis sumbu"):
```




```
>Q&=[2,1]
```

[2, 1]

```
>R&=[-1,2]
```

[- 1, 2]

```
>p&=middlePerpendicular(Q,R)
```

[3, - 1, 0]

```
>$getLineEquation(p,x,y); $solve(%,y) | expand
```

$$[y = 3\,x]$$

Jadi,persamaan garis sumbunya

$$y = 3x$$

dan sejajar dengan

$$y = 3x + \frac{19}{7}$$

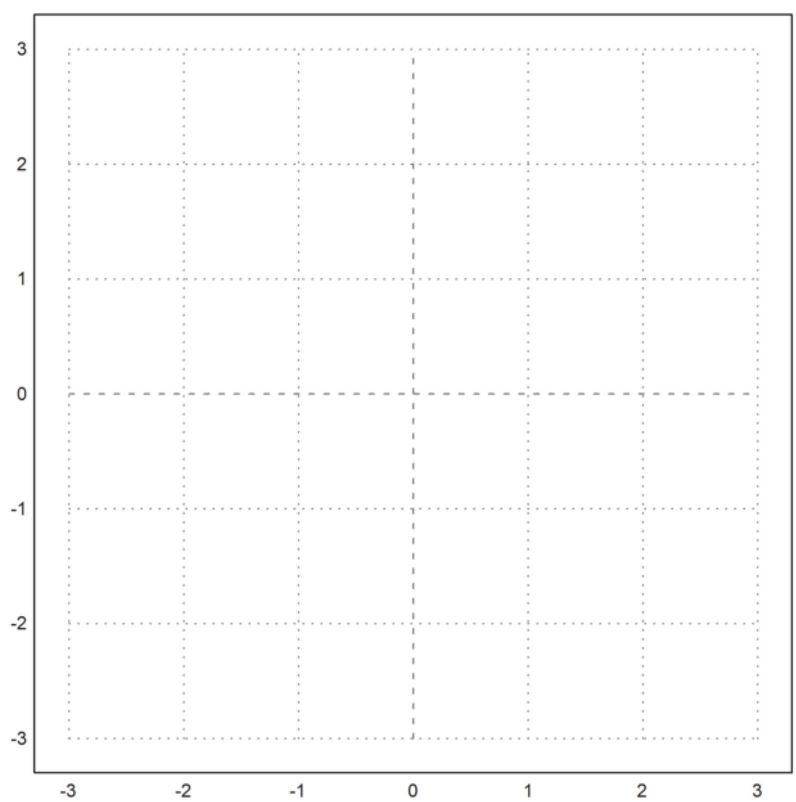
3. Tentukan garis sumbu dari

$$A(2,5)B(3,-2)$$

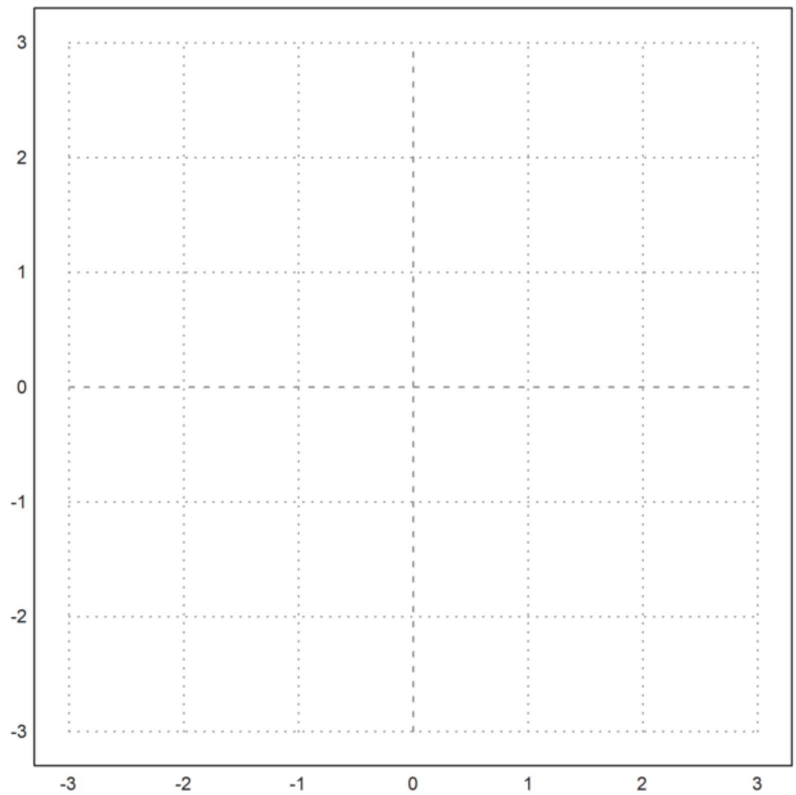
dan tunjukkan hubungan garis sumbu tersebut dengan garis

$$y = -7x + 23$$

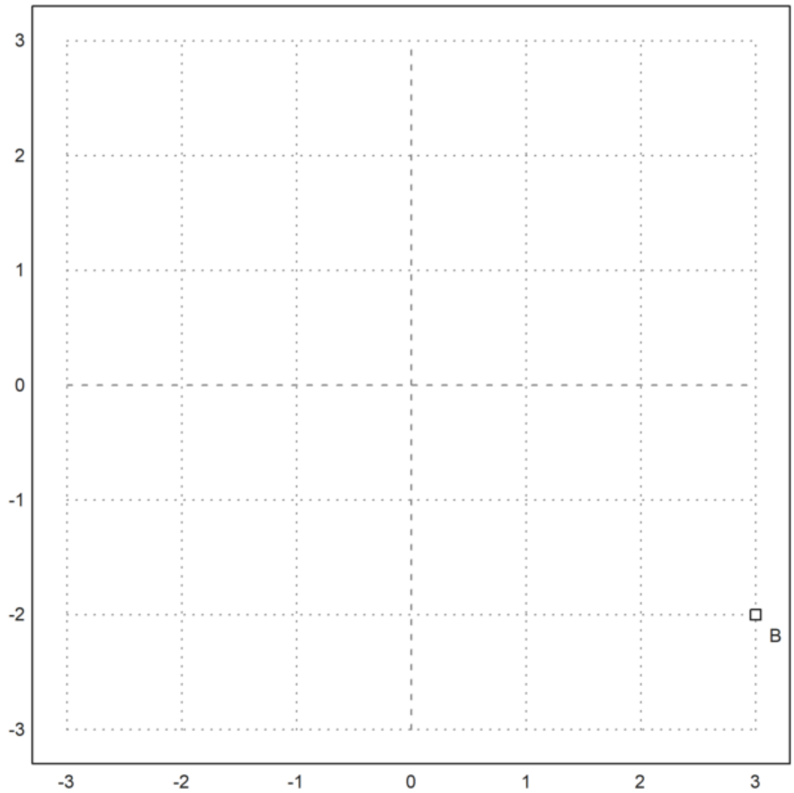
```
>setPlotRange(3):
```



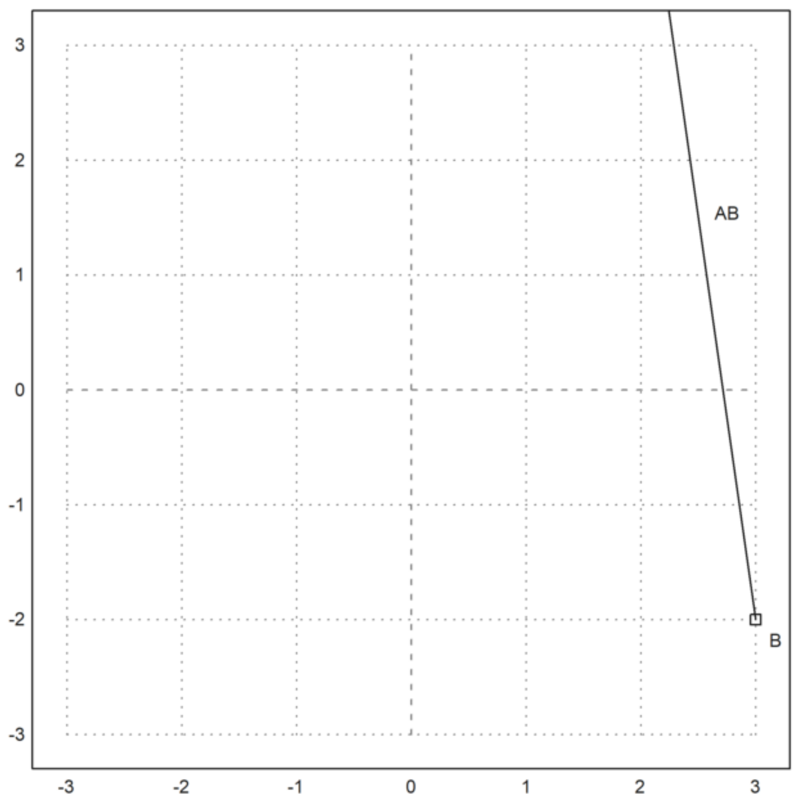
```
>A=[2,5]; plotPoint(A):
```



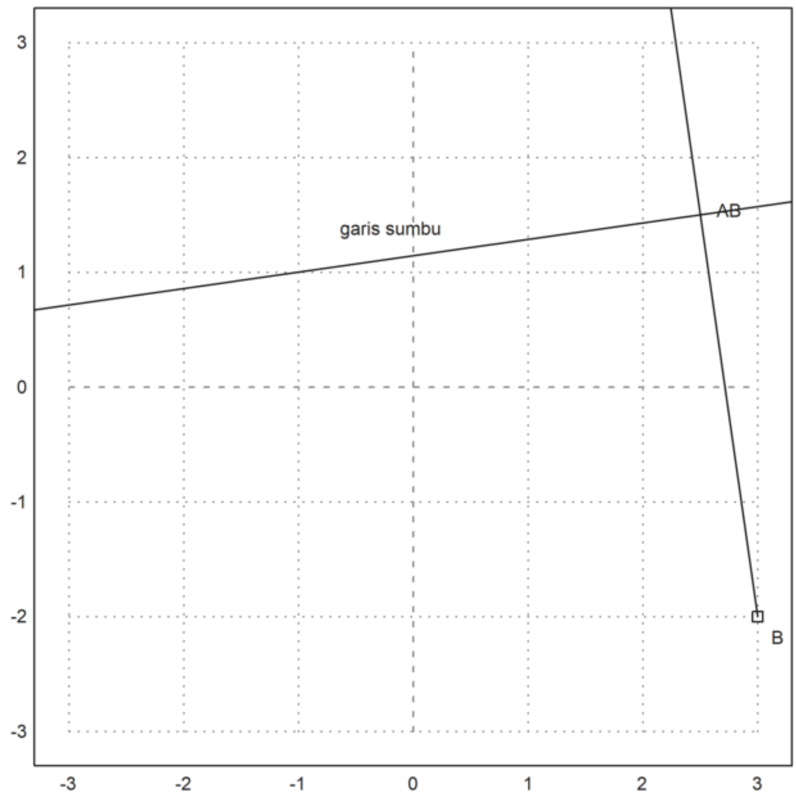

```
>B=[3,-2]; plotPoint(B):
```




```
>plotSegment(A,B,"AB"):
```



```
>p=middlePerpendicular(A,B); plotLine(p,"garis sumbu"):
```




```
>A&=[2,5]
```

[2, 5]

```
>B&=[3,-2]
```

[3, - 2]

```
>p&=middlePerpendicular(A,B)
```

[- 1, 7, 8]

```
>$getLineEquation(p,x,y); $solve(%,y) | expand
```

$$\left[y=\frac{x}{7}+\frac{8}{7}\right]$$

Jadi, persamaan garis sumbunya

$$y=\frac{x}{7}+\frac{8}{7}$$

dan tegak lurus dengan

$$y=-7x+23$$

Menentukan persamaan garis bagi sudut

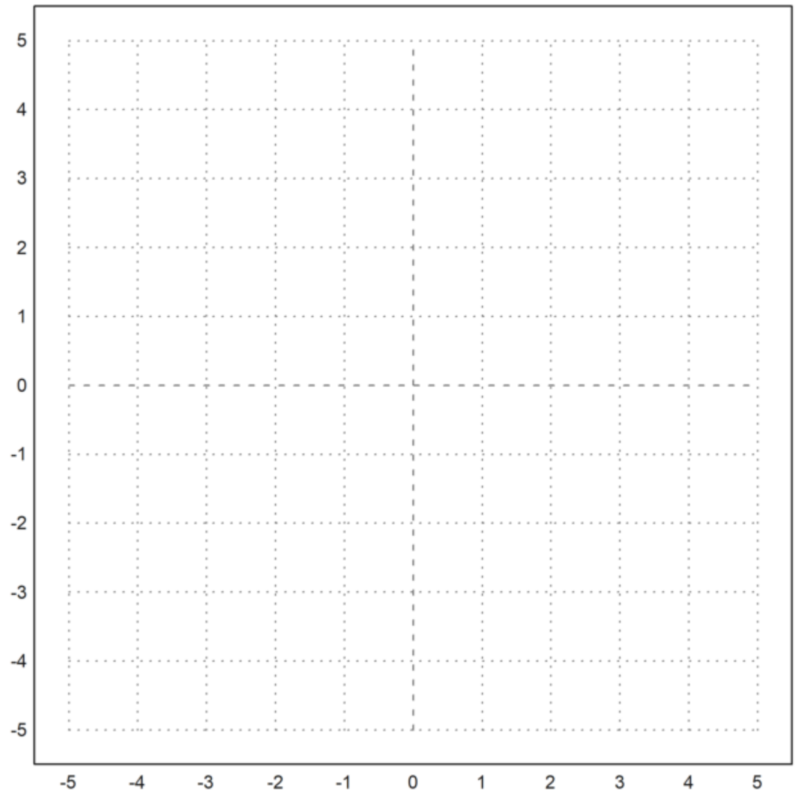
Garis bagi adalah garis yang membagi suatu sudut menjadi dua sudut yang sama besar. Dalam bahasa inggris, garis bagi disebut angle bisector.
Untuk mencari persamaan garis bagi sudut dalam EMT kita dapat menempuh cara-cara berikut:
1. Mendefinisikan titik-titik koordinat
2. Menggunakan perintah angleBisector(titik1,titik2,titik3)
3. Menggunakan getLineEquation
Contoh:

$$\triangle ABC, A(-1,2); B(3,1); C(-2,-4)$$

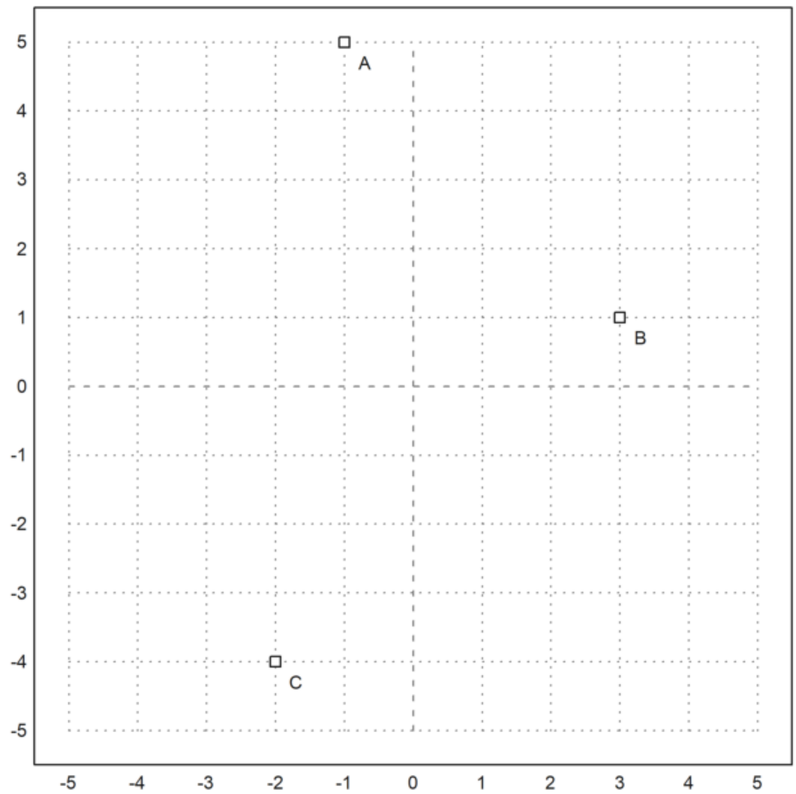
Tentukan garis bagi dari

$$\angle ABC$$

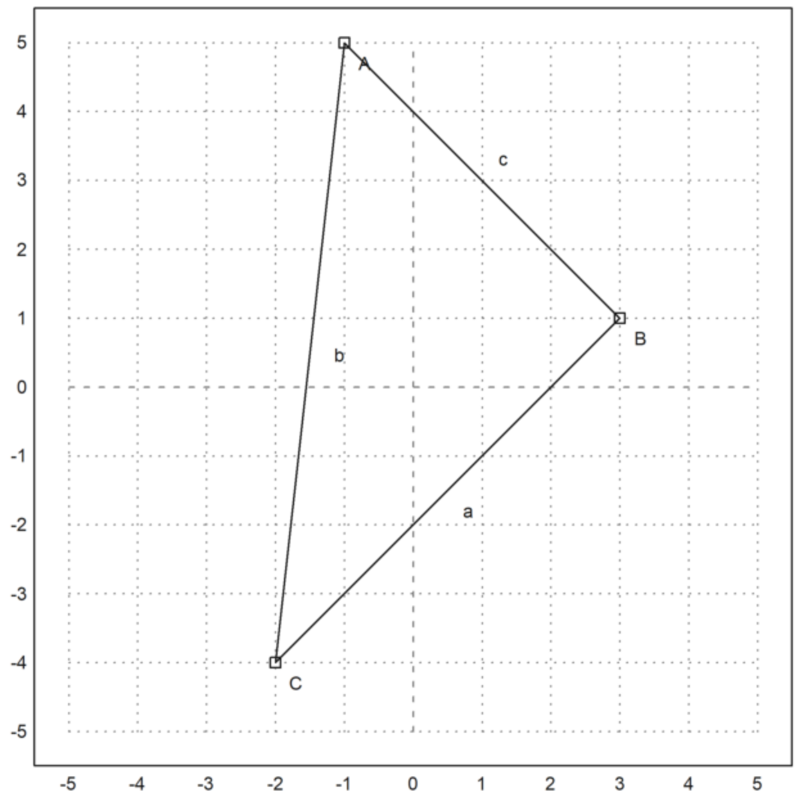
```
>setPlotRange(5):
```



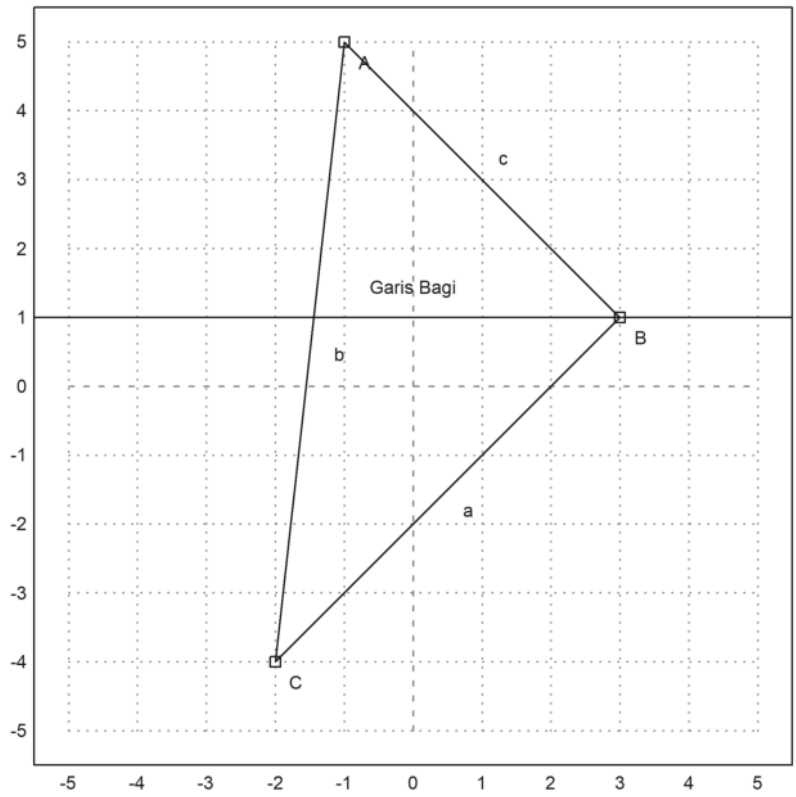

```
>A=[-1,5]; B=[3,1]; C=[-2,-4];
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"):
```



```
>plotSegment(A,B,"c"); plotSegment(B,C,"a"); plotSegment(A,C,"b"):
```




```
>w=angleBisector(A,B,C); plotLine(w,"Garis Bagi"):
```




```
>A&=[-1,5]
```

[- 1, 5]

```
>B&=[3,1]
```

[3, 1]

```
>C&=[-2,-4]
```

[- 2, - 4]

```
>w&=angleBisector(A,B,C)
```

[0, 8, 8]

```
>$getLineEquation(w,x,y)
```

$$8\,y=8$$

```
>$solve(%,y) | expand
```

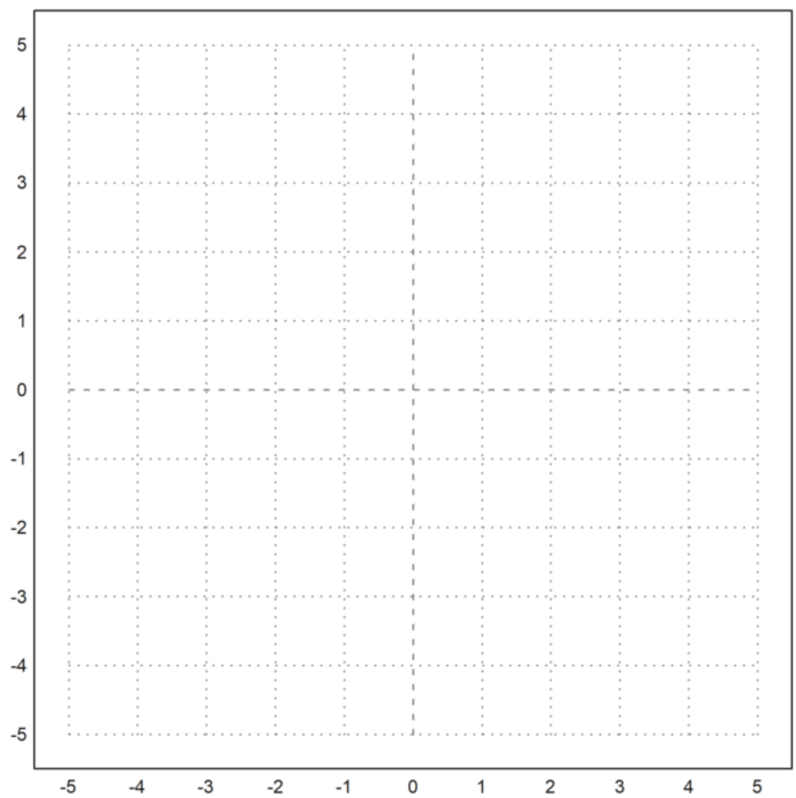
$$[y=1]$$

Latihan C

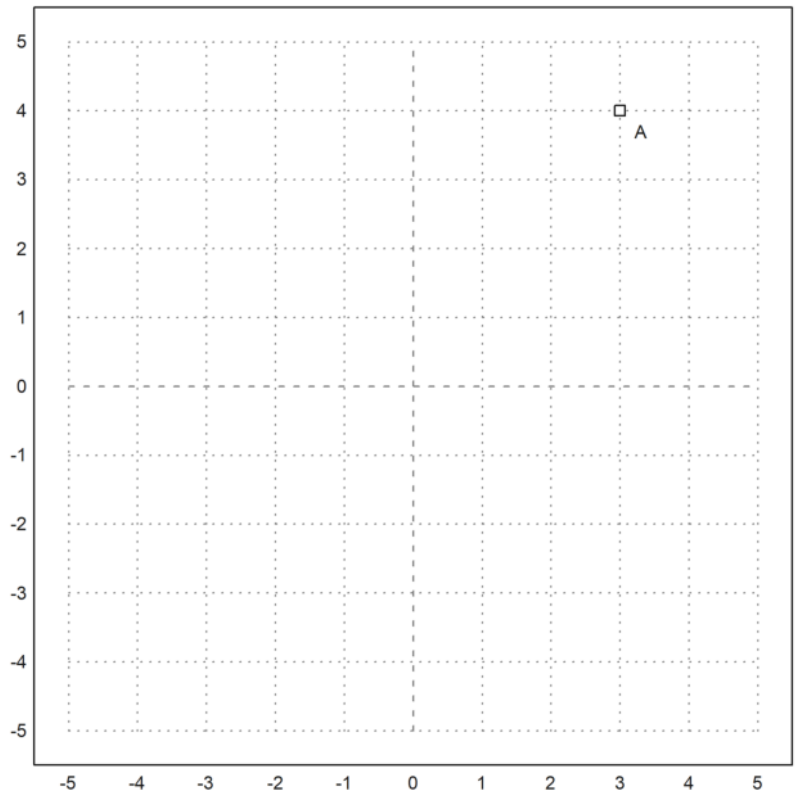
1. Tentukan persamaan garis bagi sudut BCA jika

$$A(3,4)B(-2,4)C(1,3)$$

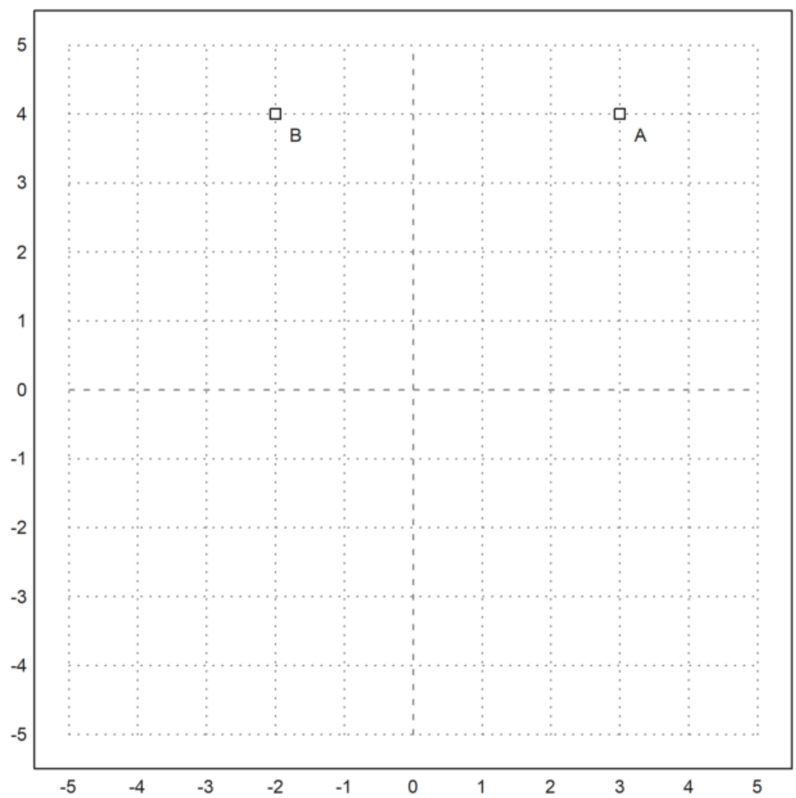
```
>setPlotRange(5):
```



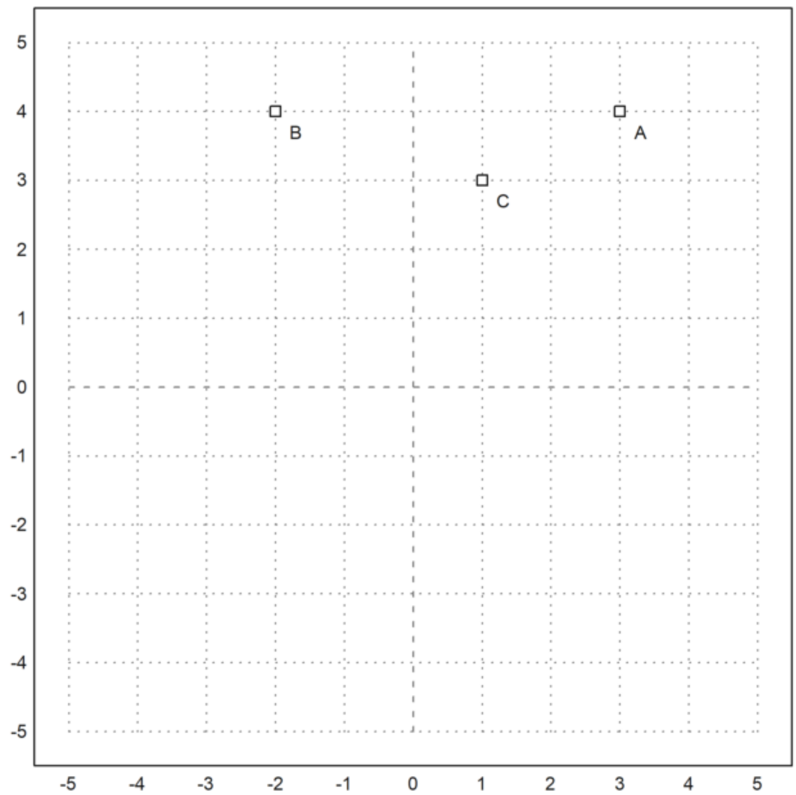

```
>A=[3,4]; plotPoint(A,"A"):
```



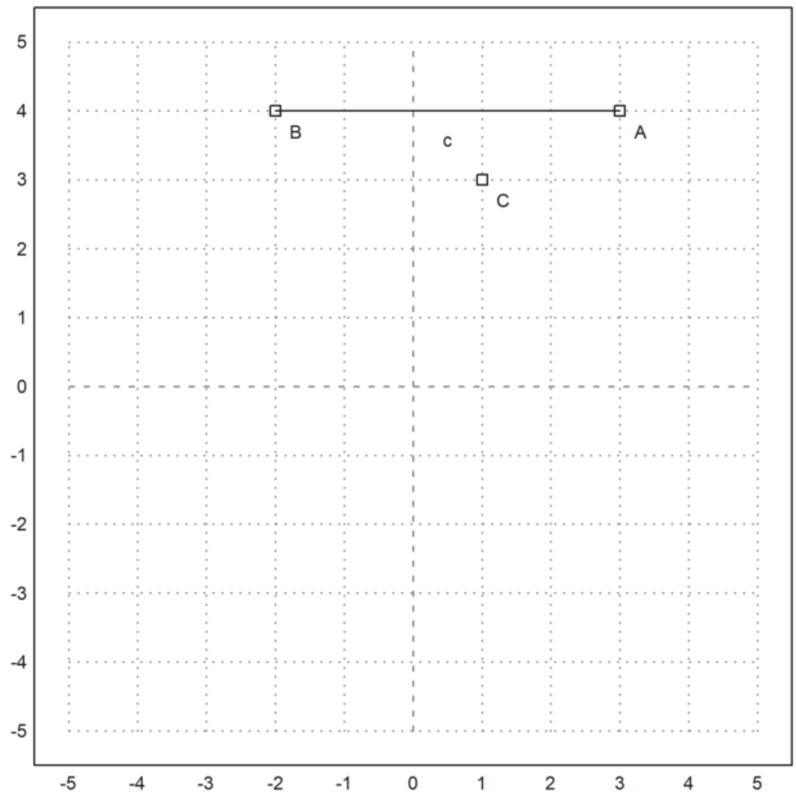

```
>B=[-2,4]; plotPoint(B,"B"):
```



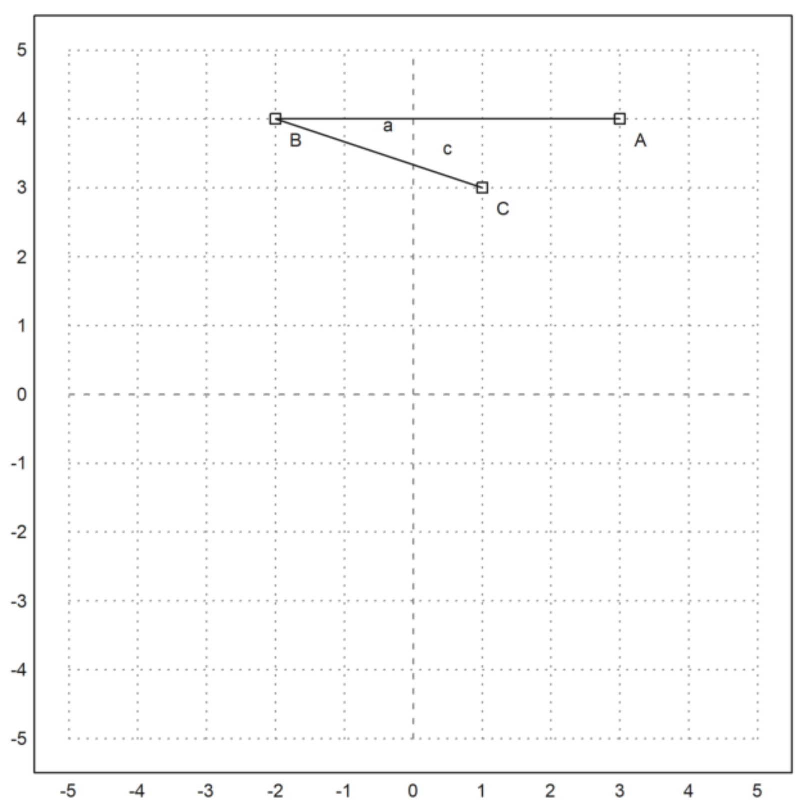
```
>C=[1,3]; plotPoint(C,"C"):
```



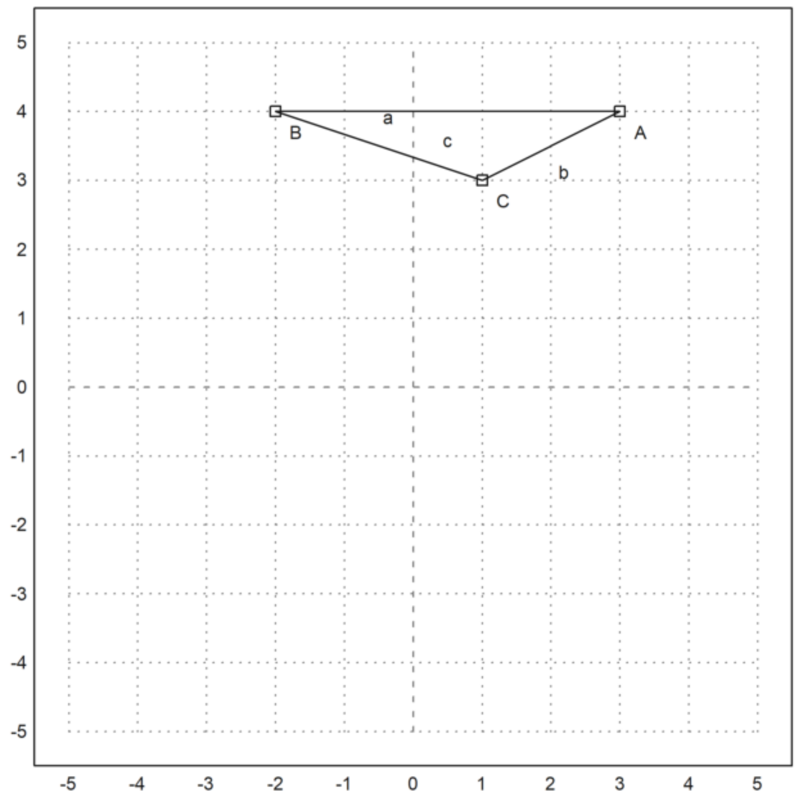

```
>plotSegment(A,B,"c"): // c=AB
```



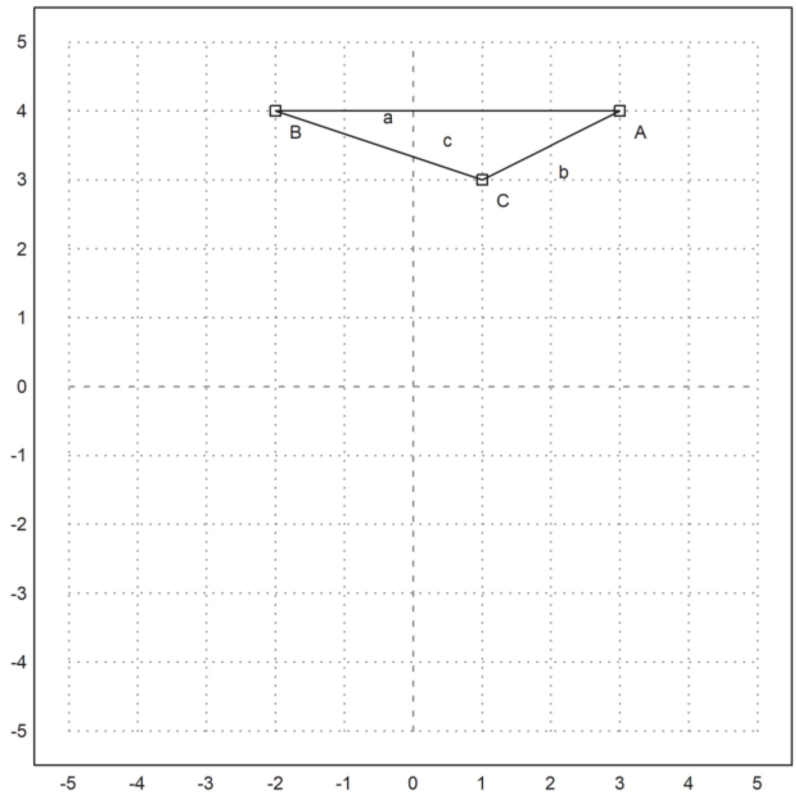

```
>plotSegment(B,C,"a"): // a=BC
```



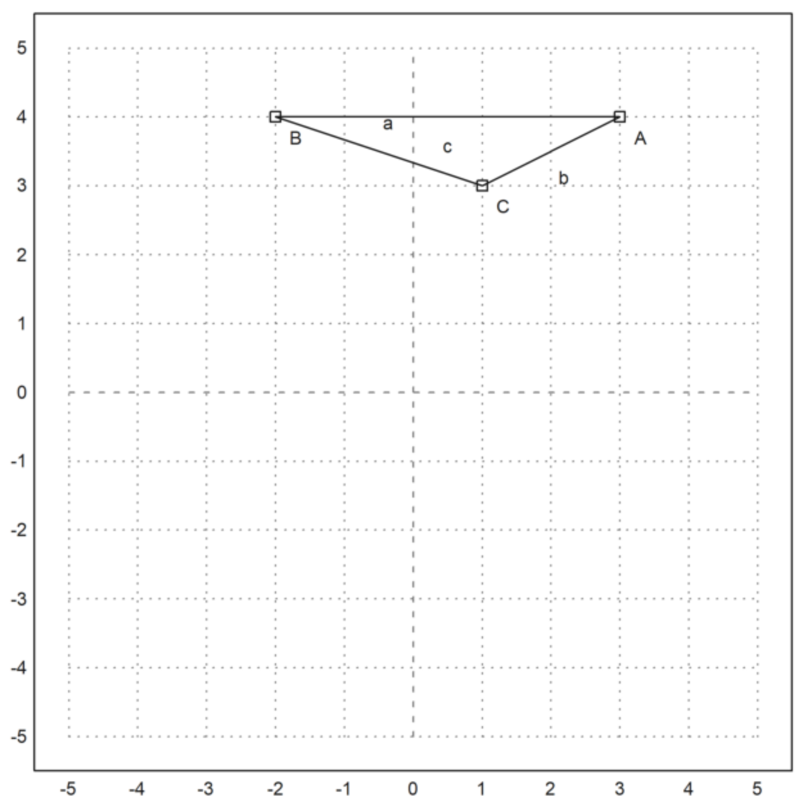
```
>plotSegment(A,C,"b"): // b=AC
```



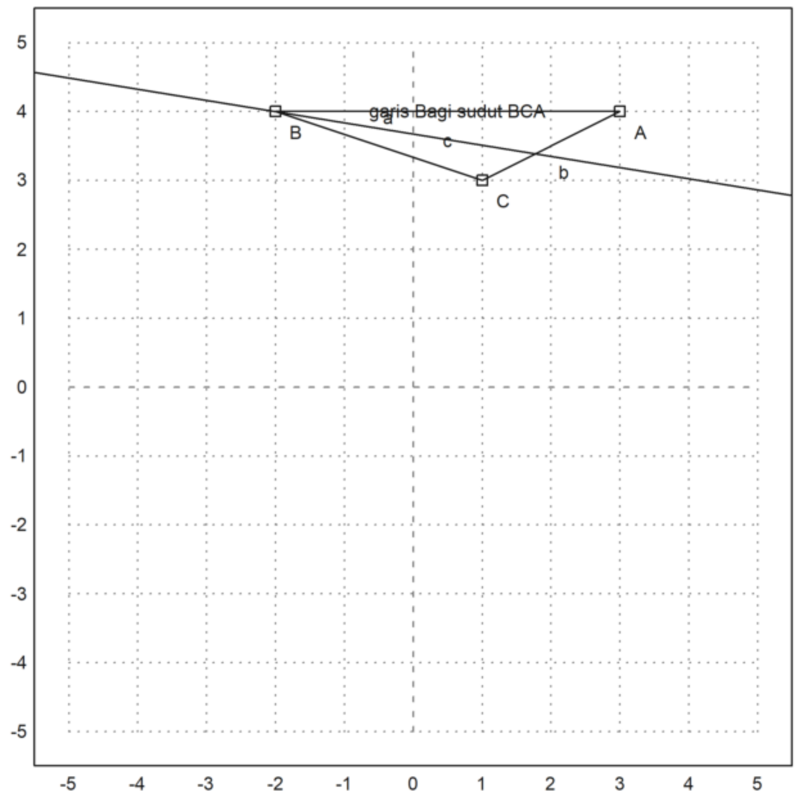

```
>t=angleBisector(A,B,C):
```




```
>o=lineIntersection(t,lineThrough(A,B)):
```



```
>plotLine(t,"garis Bagi sudut BCA"):
```




```
>A&=[3,4]
```

[3, 4]

```
>B&=[-2,4]
```

[- 2, 4]

```
>C&=[1,3]
```

[1, 3]

```
>t&=angleBisector(B,C,A)
```

```
          2 sqrt(10)           sqrt(10)
       [- ---------- - 3, 1 - --------,
           sqrt(5)             sqrt(5)
   2 sqrt(10)        2 sqrt(10)              sqrt(10)    sqrt(10)
(- ---------- - 3) (---------- - 1)    (1 - --------) (-------- + 7)
    sqrt(5)           sqrt(5)                sqrt(5)     sqrt(5)
----------------------------------- + -----------------------------]
                 2                                  2
```

```
>$getLineEquation(t,x,y)
```

$$\left(1-\frac{\sqrt{10}}{\sqrt{5}}\right)\,y+\left(-\frac{2\,\sqrt{10}}{\sqrt{5}}-3\right)\,x=\frac{\left(-\frac{2\,\sqrt{10}}{\sqrt{5}}-3\right)\,\left(\frac{2\,\sqrt{10}}{\sqrt{5}}-1\right)}{2}+\frac{\left(1-\frac{\sqrt{10}}{\sqrt{5}}\right)\,\left(\frac{\sqrt{10}}{\sqrt{5}}+7\right)}{2}$$

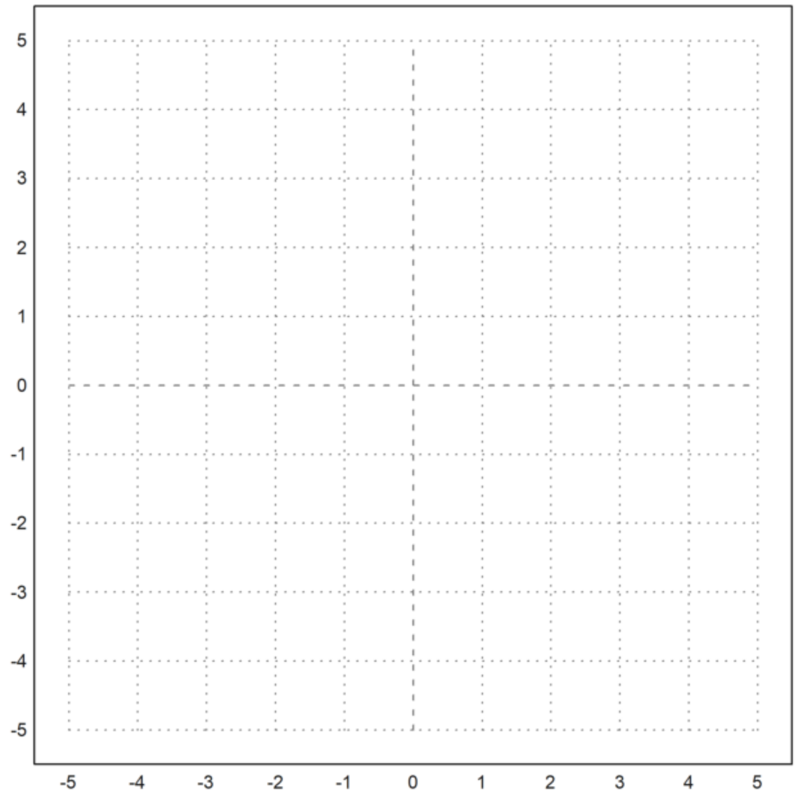
```
>$solve(%,y) | expand
```

$$\left[y=-\frac{2\,\sqrt{10}\,x}{\sqrt{10}-\sqrt{5}}-\frac{3\,\sqrt{5}\,x}{\sqrt{10}-\sqrt{5}}+\frac{5\,\sqrt{10}}{\sqrt{10}-\sqrt{5}}\right]$$

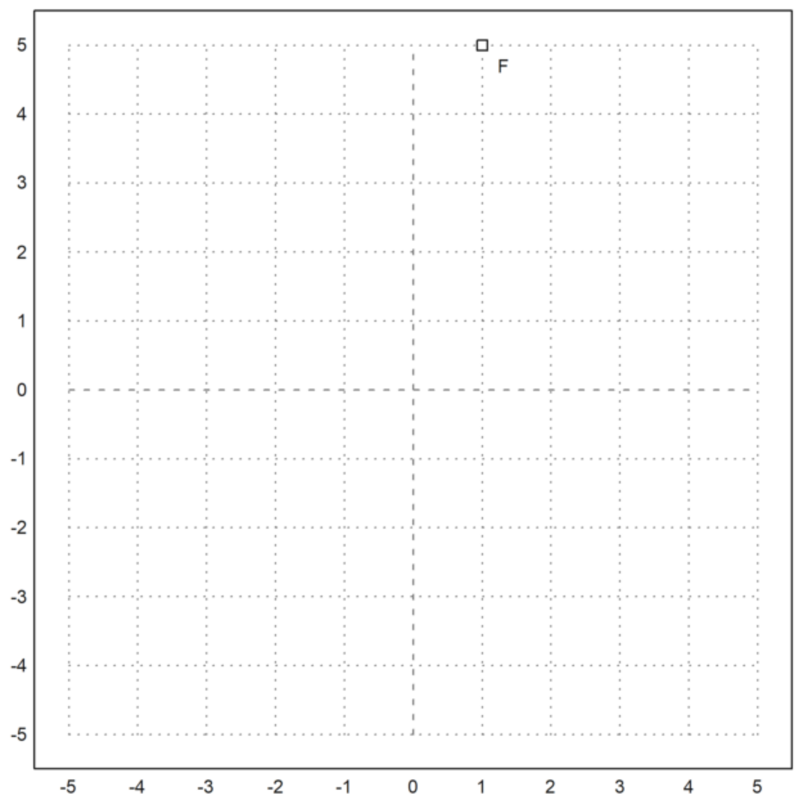
2. Tentukan persamaan garis bagi sudut FHG dengan

$$F(1,5)G(-4,3)H(0,-5)$$

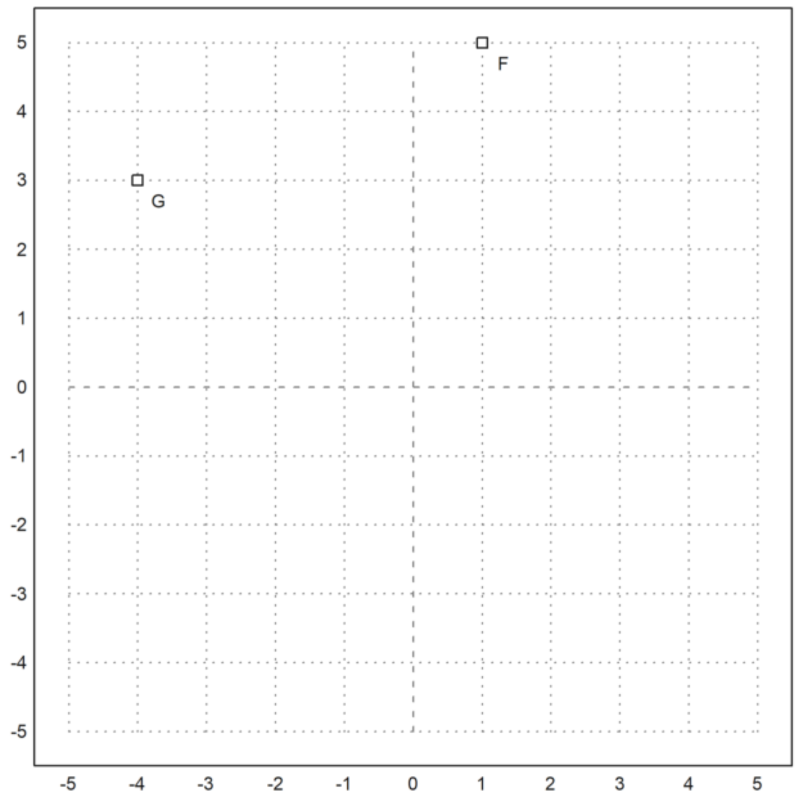
```
>setPlotRange(5):
```



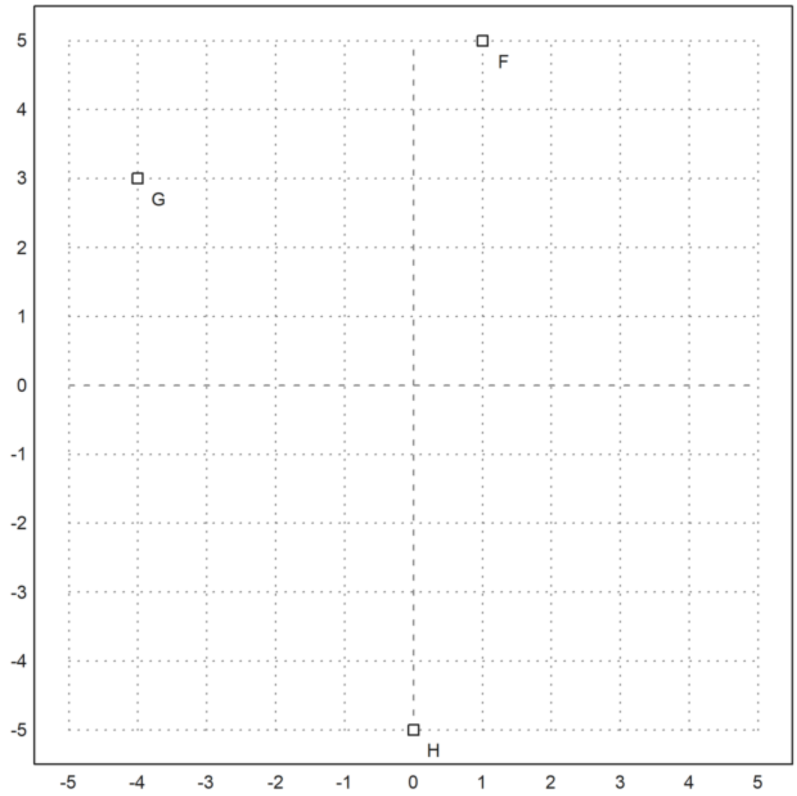

```
>F=[1,5]; plotPoint(F,"F"):
```



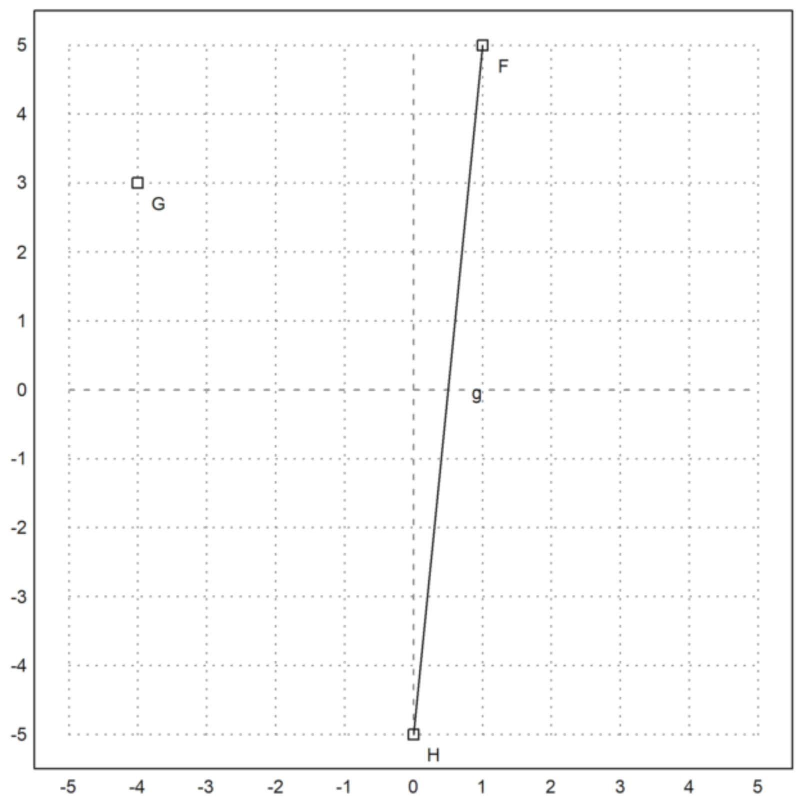
```
>G=[-4,3]; plotPoint(G,"G"):
```



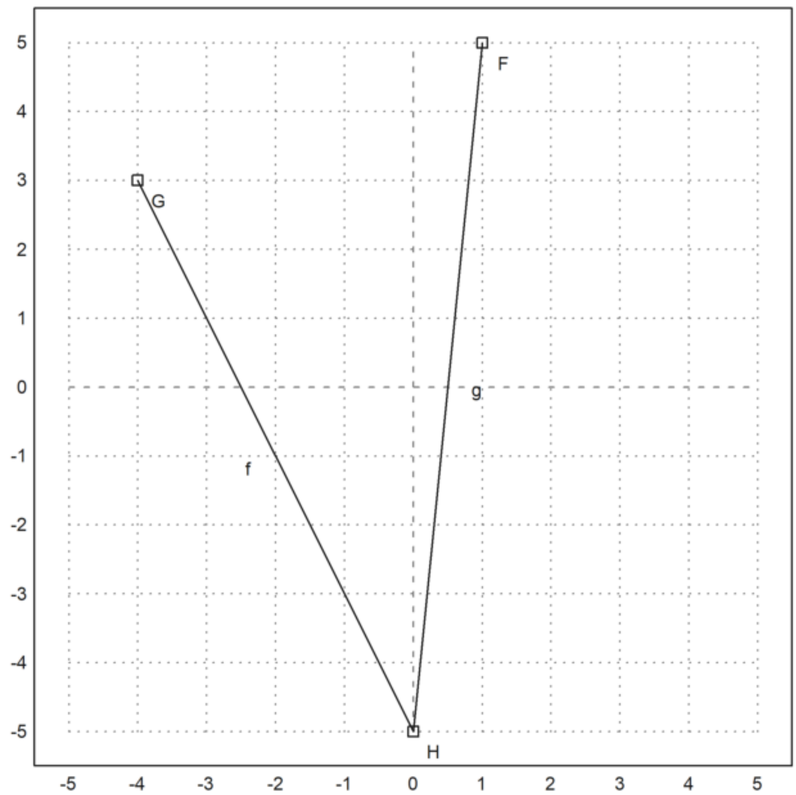

```
>H=[0,-5]; plotPoint(H,"H"):
```



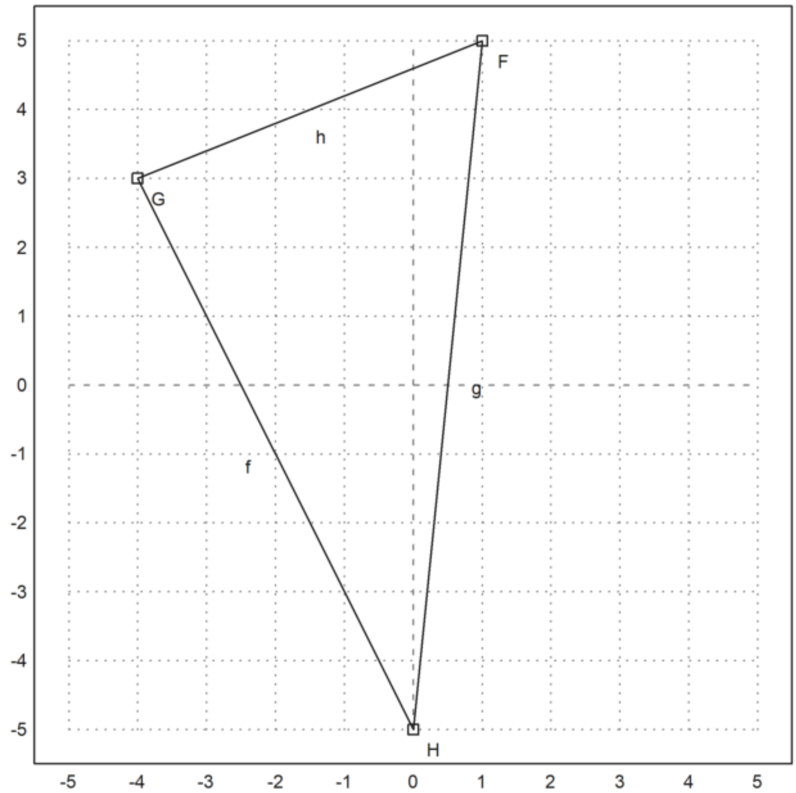

```
>plotSegment(F,H,"g"):
```



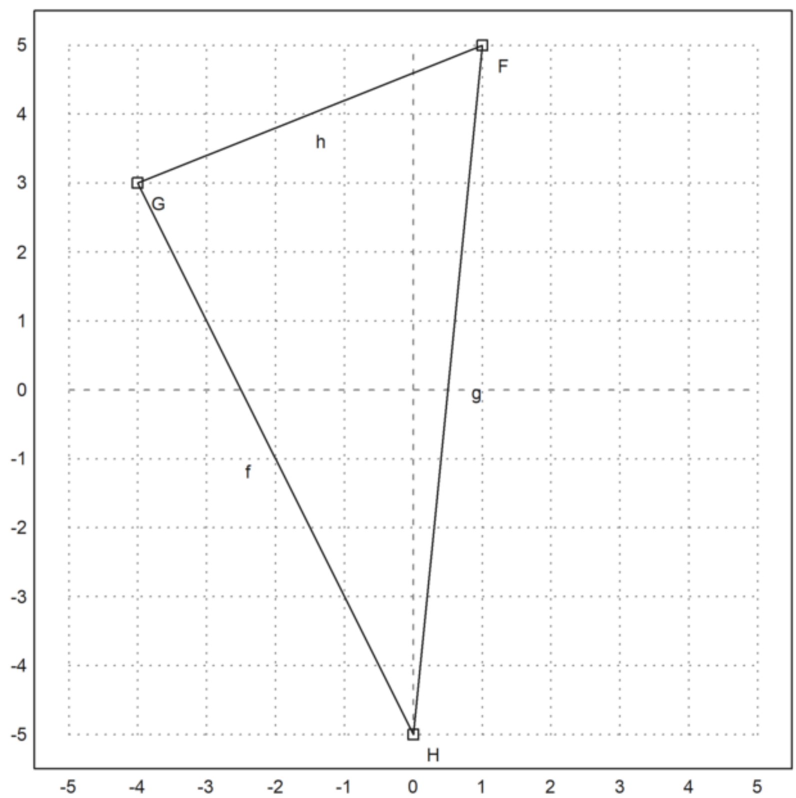
```
>plotSegment(H,G,"f"):
```



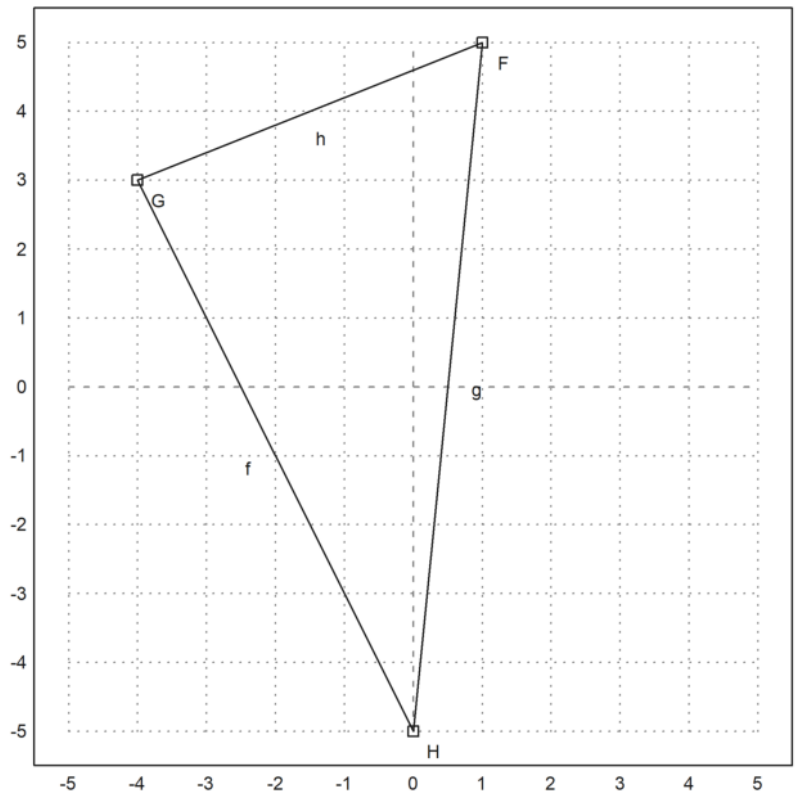

```
>plotSegment(F,G,"h"):
```



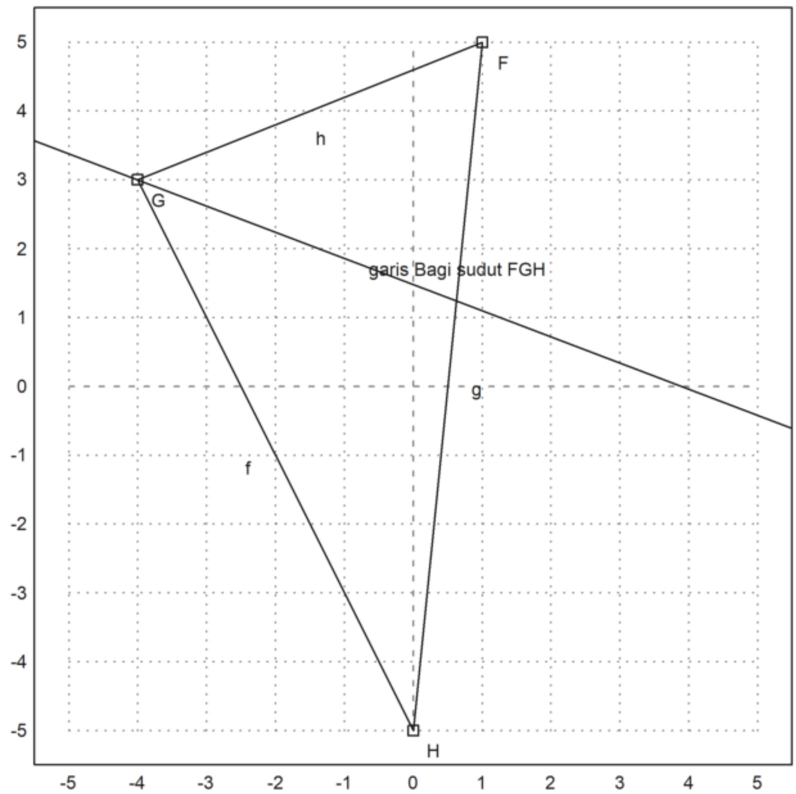

```
>t=angleBisector(F,G,H):
```



```
>o=lineIntersection(t,lineThrough(F,G)):
```




```
>plotLine(t,"garis Bagi sudut FGH"):
```




```
>F&=[1,5]
```

                                   [1, 5]

```
>G&=[-4,3]
```

                                  [- 4, 3]

```
>H&=[0,5]
```

                                   [0, 5]

```
>t&=angleBisector(F,H,G)
```

```
                 2          2
(1 - -------)  (------- + 1)
```

```
      2              1           sqrt(5)    sqrt(5)
[------- + 1, -------, ---------------------------
 sqrt(5)        sqrt(5)                 2
                                                        1
                                                 10 - -------
                                                      sqrt(5)
                                               + ------------]
                                                   2 sqrt(5)
```

```
>$getLineEquation(t,x,y)
```

$$\frac{y}{\sqrt{5}}+\left(\frac{2}{\sqrt{5}}+1\right)\,x=\frac{\left(1-\frac{2}{\sqrt{5}}\right)\,\left(\frac{2}{\sqrt{5}}+1\right)}{2}+\frac{10-\frac{1}{\sqrt{5}}}{2\,\sqrt{5}}$$

```
>$solve(%,y) | expand
```

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

Menentukan persamaan garis berat

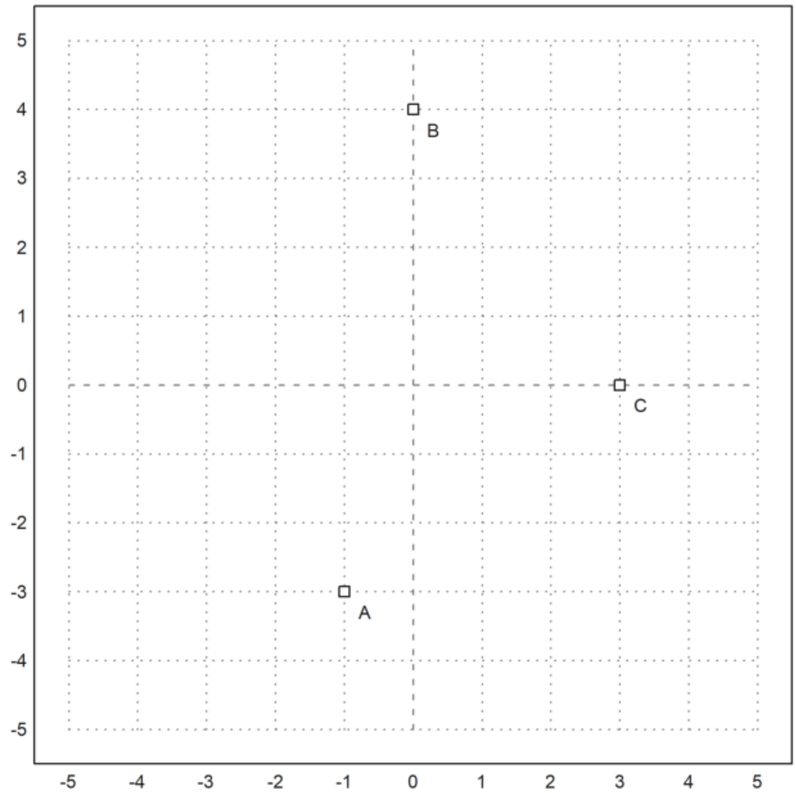
Garis berat pada sebuah segitiga adalah garis yang menghubungkan salah satu sudut segitiga dengan titik tengah sisi yang berlawanan. Untuk setiap sebuah segitiga, terdapat tiga garis berat yang menghubungkan salah satu sudut segitiga menuju titik tengah sisi yang berlawanan.
Cara mencari persamaan garis berat menggunakan EMT yaitu:
1. Mendefinisikan titik-titik koordinat
2. Menggunakan middlePerpendicullar(titik1,titik2) untuk mencari titik tengah pada sisi berlawanan
3. Mendefinisikan titik tengah
4. Menghubungkan titik tengah dengan titik sudut
5. menggunakan getLineEquation
Tentukan garis berat dari

$$\triangle ABC, A(-1,-3); B(0,4); C(3,0)$$

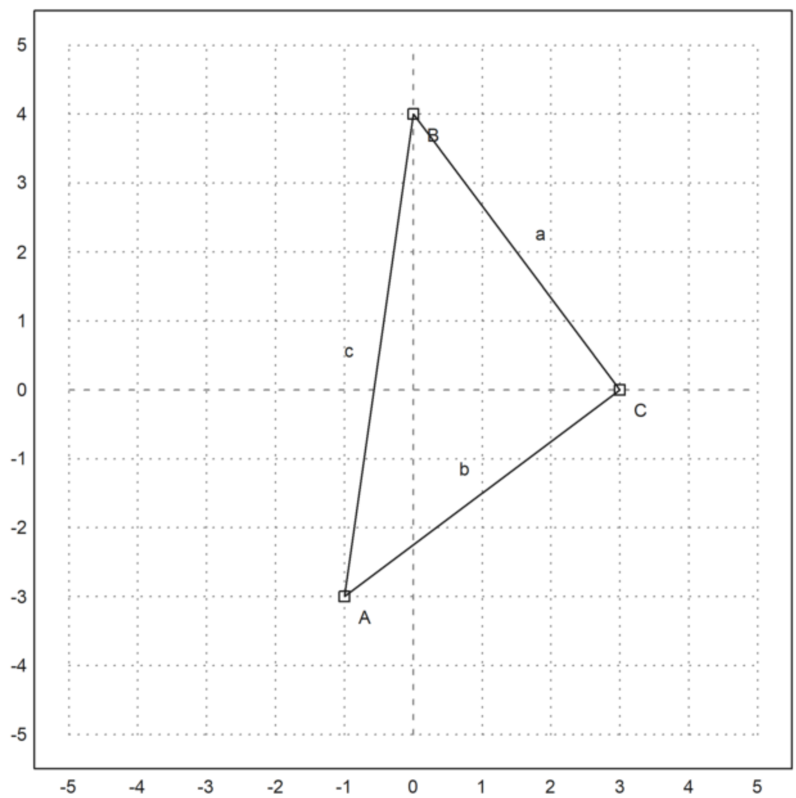
```
>setPlotRange(5)
```

```
[-5,  5,  -5,  5]
```

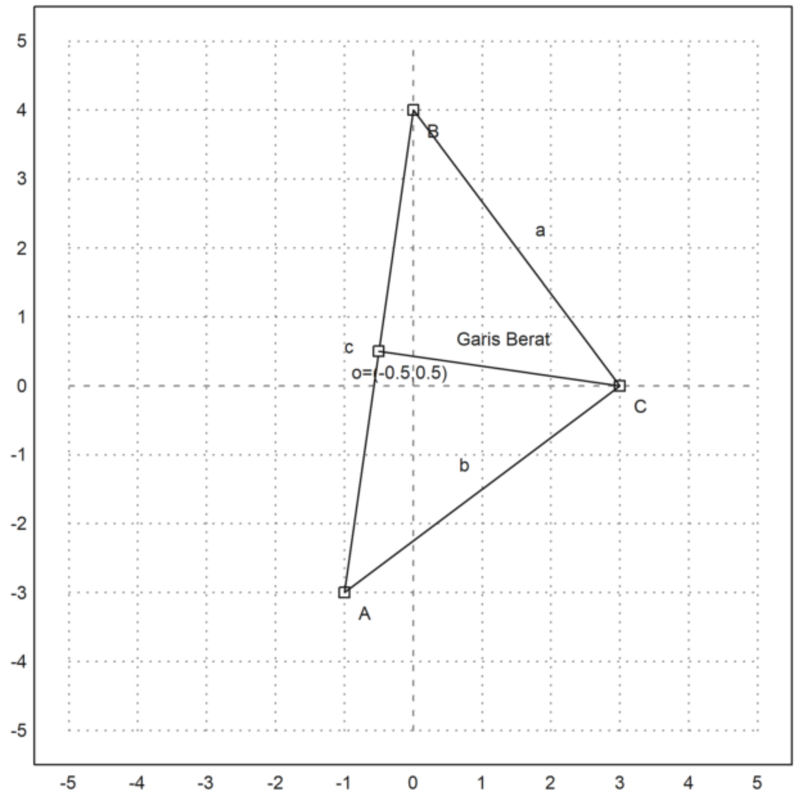
```
>A=[-1,-3]; B=[0,4]; C=[3,0];
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"):
```




```
>plotSegment(A,B,"c"); plotSegment(B,C,"a"); plotSegment(A,C,"b"):
```



```
>v=middlePerpendicular(A,B);
>o=lineIntersection(v, lineThrough(A,B)); plotPoint(o,value=1);
>plotSegment(o,C,"Garis Berat"):
```




```
>A&=[-1,-3]
```

[- 1, - 3]

```
>B&=[0,4]
```

[0, 4]

```
>C&=[3,0]
```

[3, 0]

```
>v&=middlePerpendicular(A,B)
```

```
[- 1, - 7, - 3]
```

```
>o&=lineIntersection(v,lineThrough(A,B))
```

```
    1   1
[- -, -]
    2   2
```

```
>z&=lineThrough(o,C)
```

```
 1   7   3
[-, -, -]
 2   2   2
```

```
>$getLineEquation(z,x,y)
```

$$\frac{7\,y}{2}+\frac{x}{2}=\frac{3}{2}$$

```
>$solve(%,y) | expand
```

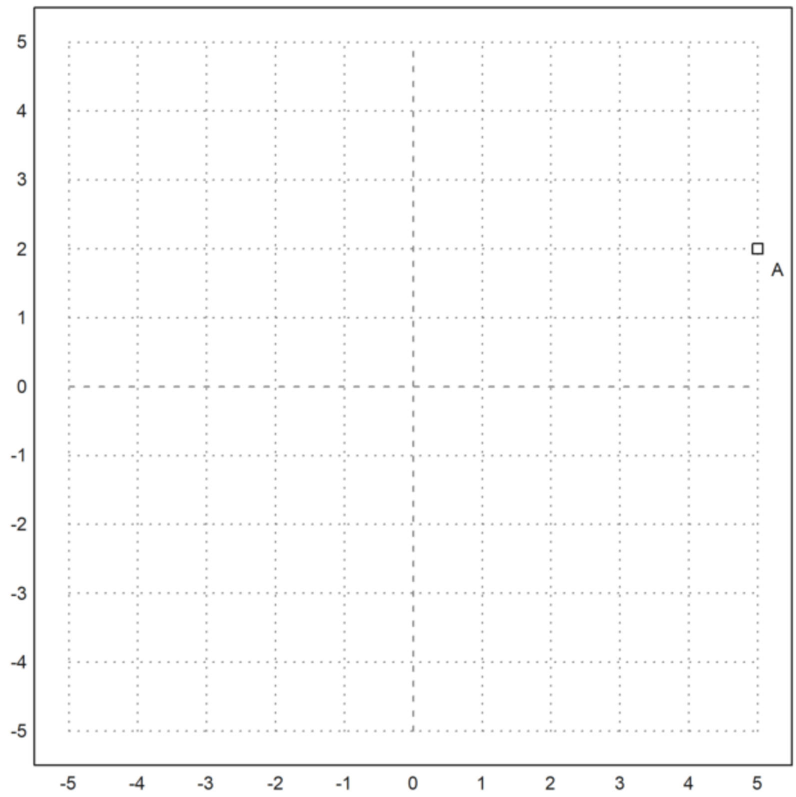
$$\left[y=\frac{3}{7}-\frac{x}{7}\right]$$

Latihan D

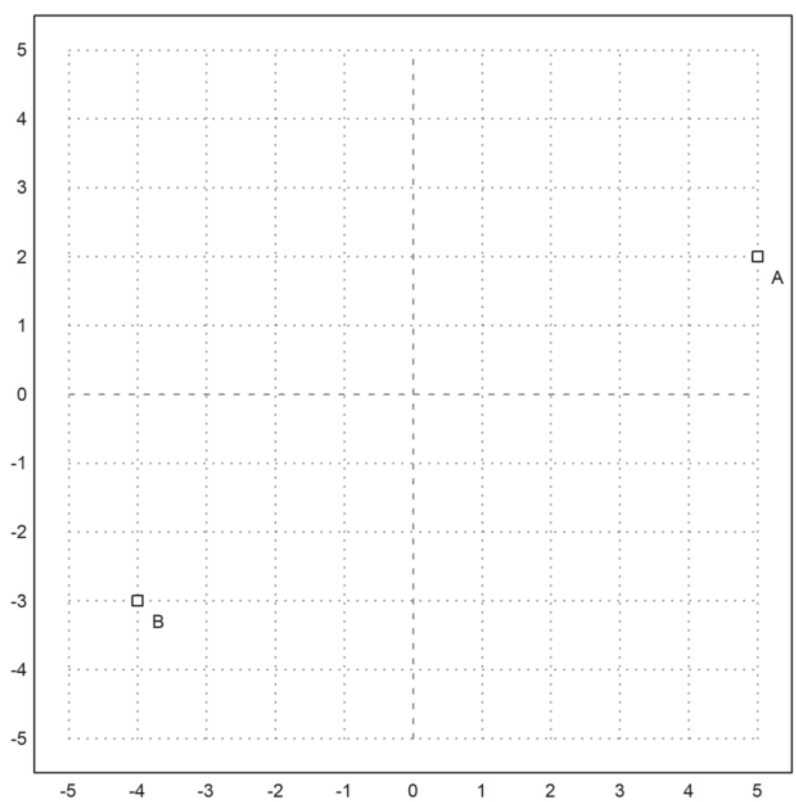
1. Tentukan persamaan garis berat dari titik C dengan

$$A(5,2), B(-4,-3), C(0,3)$$

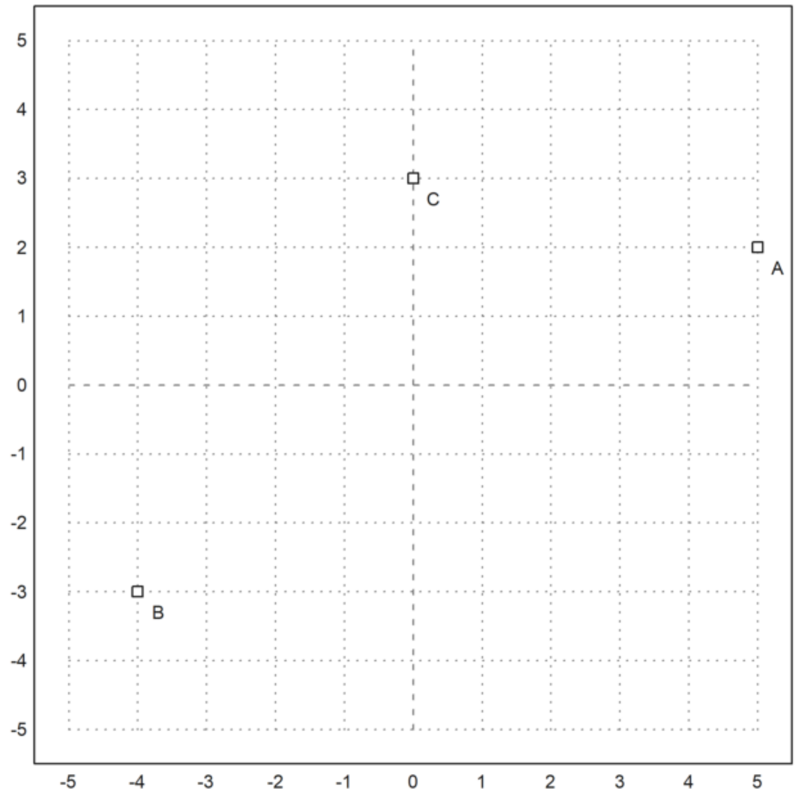
```
>setPlotRange(5);
>A=[5,2]; plotPoint(A,"A"):
```



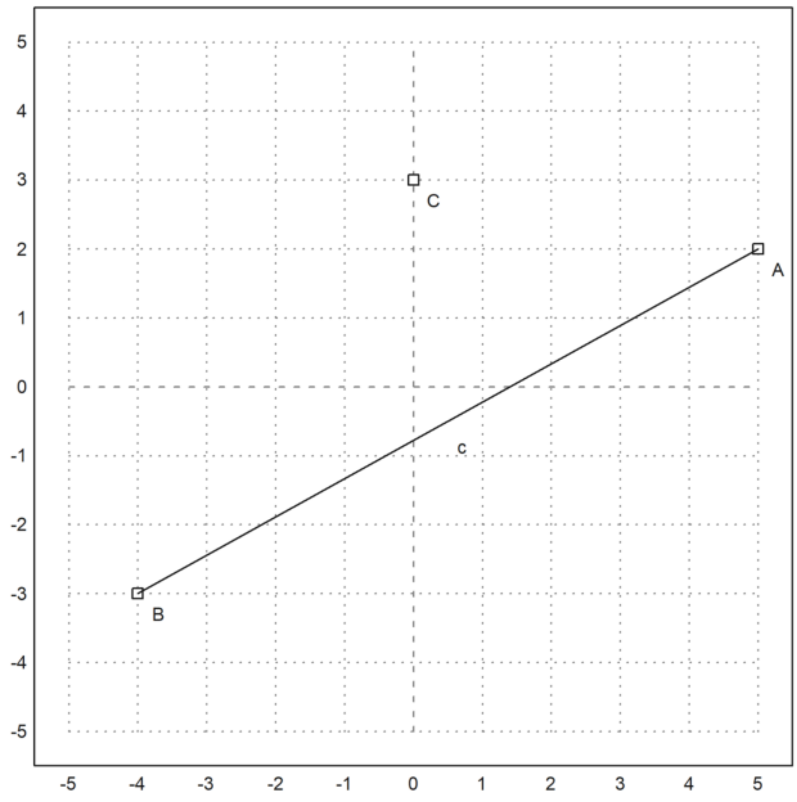

```
>B=[-4,-3]; plotPoint(B,"B"):
```



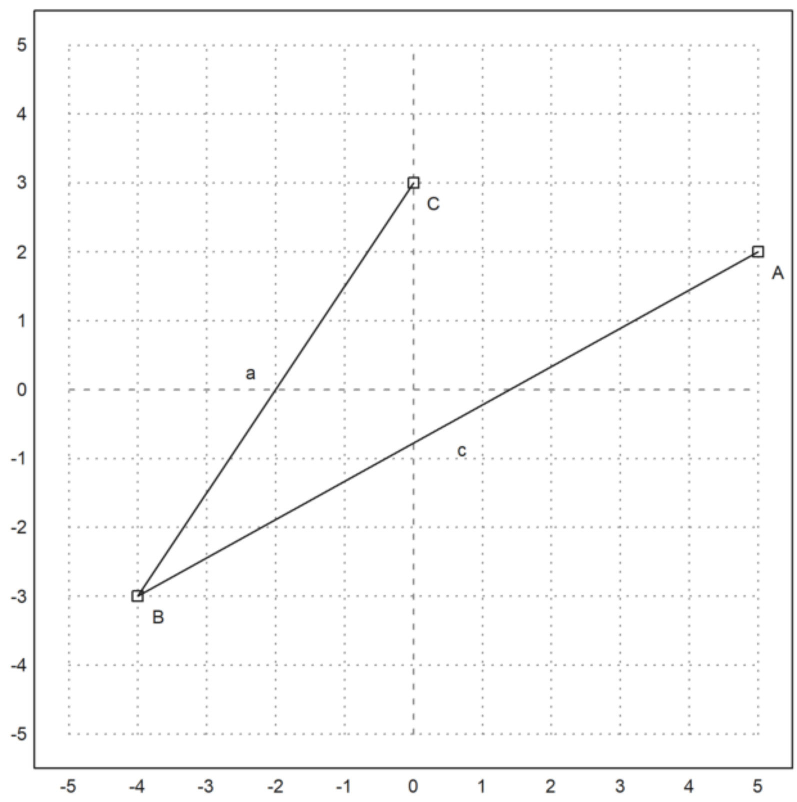
```
>C=[0,3]; plotPoint(C,"C"):
```



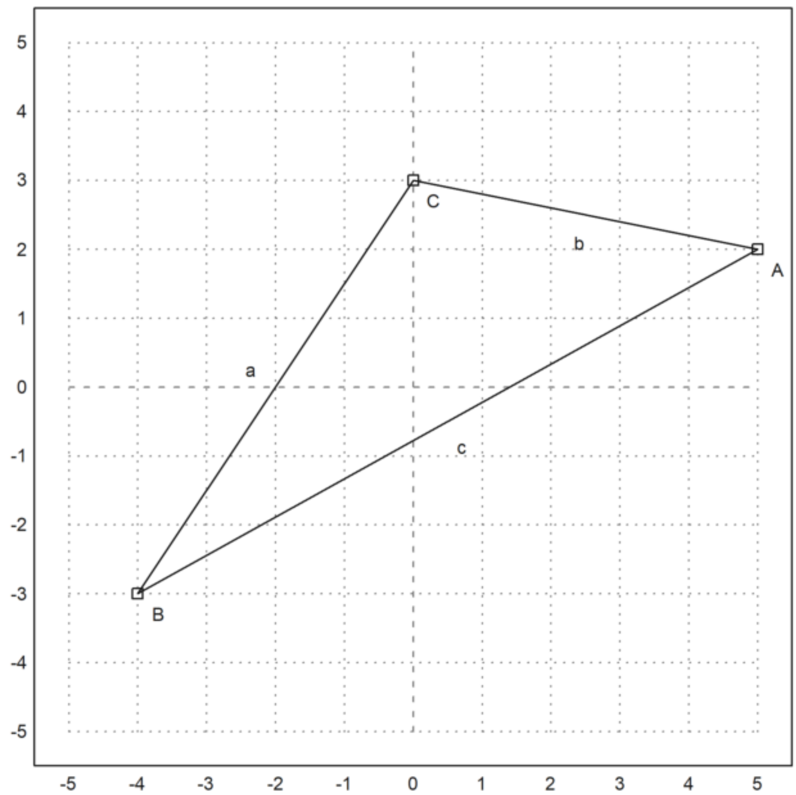

```
>plotSegment(A,B,"c"): // c=AB
```



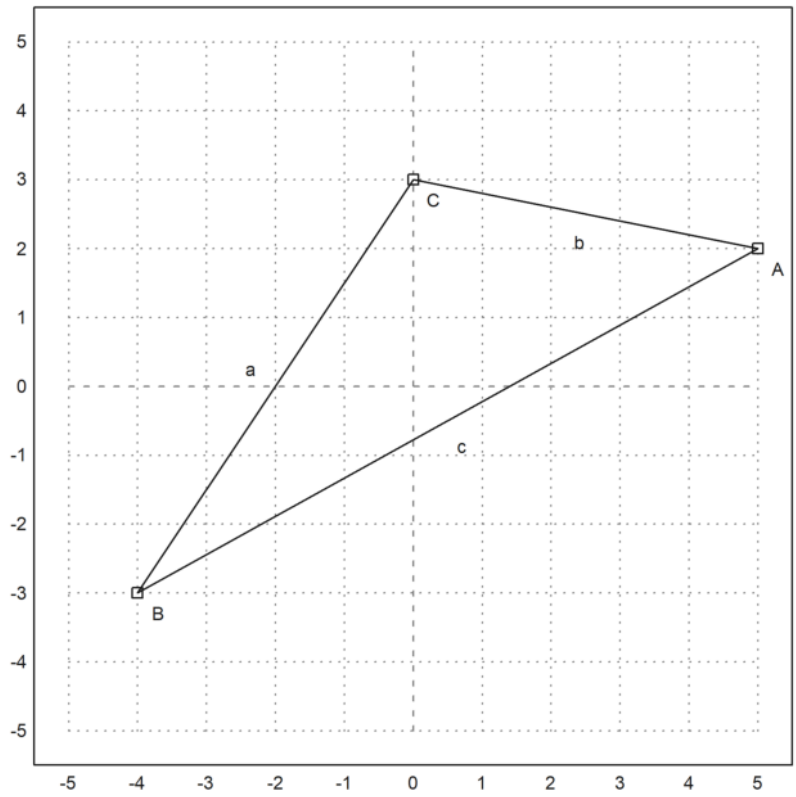

```
>plotSegment(B,C,"a"): // a=BC
```



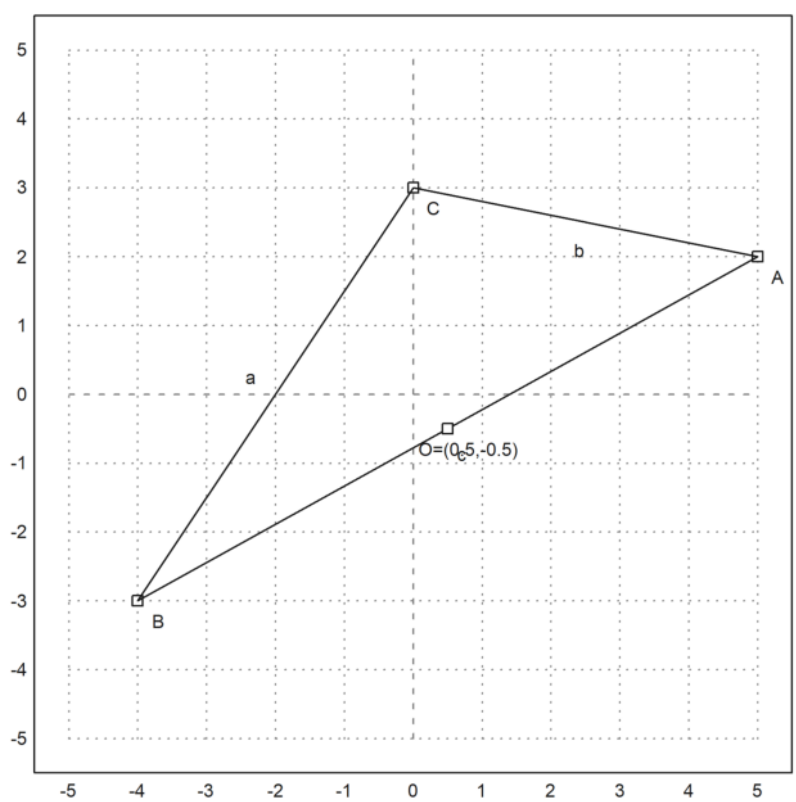
```
>plotSegment(A,C,"b"): // b=AC
```



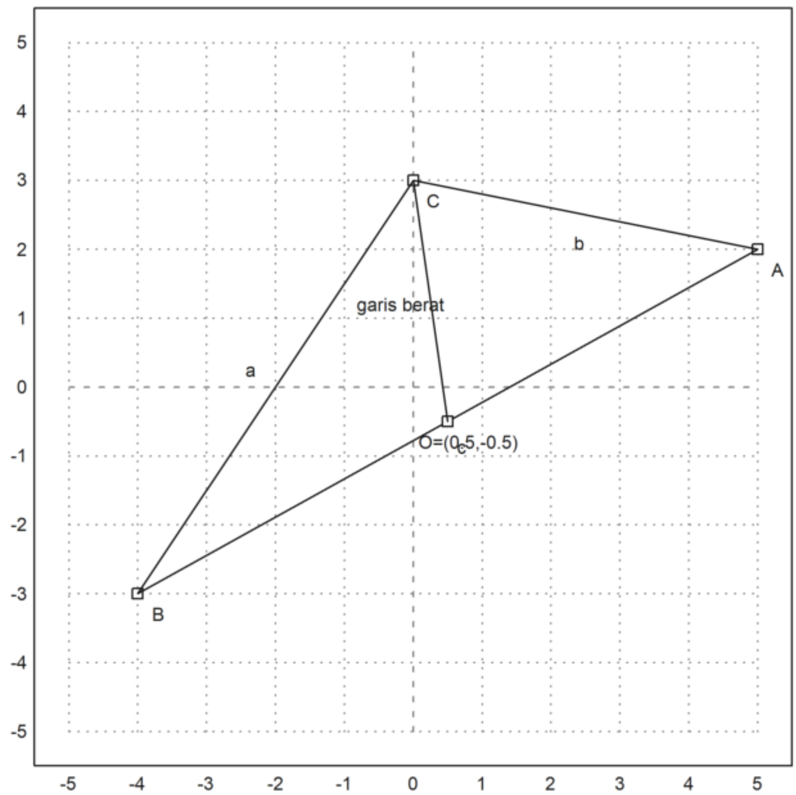

```
>t=middlePerpendicular(A,B):
```




```
>O=lineIntersection(t,lineThrough(A,B)); plotPoint(O,value=1):
```



```
>plotSegment(O,C,"garis berat"):
```




```
>A&=[5,2]
```

[5, 2]

```
>B&=[-4,-3]
```

[- 4, - 3]

```
>C&=[0,3]
```

[0, 3]

```
>t&=middlePerpendicular(A,B)
```

```
[9, 5, 2]
```

```
>O&=lineIntersection(t,lineThrough(A,B))
```

```
 1     1
[-, - -]
 2     2
```

```
>$getLineEquation(lineThrough(O,C),x,y)
```

$$-\frac{y}{2}-\frac{7\,x}{2}=-\frac{3}{2}$$

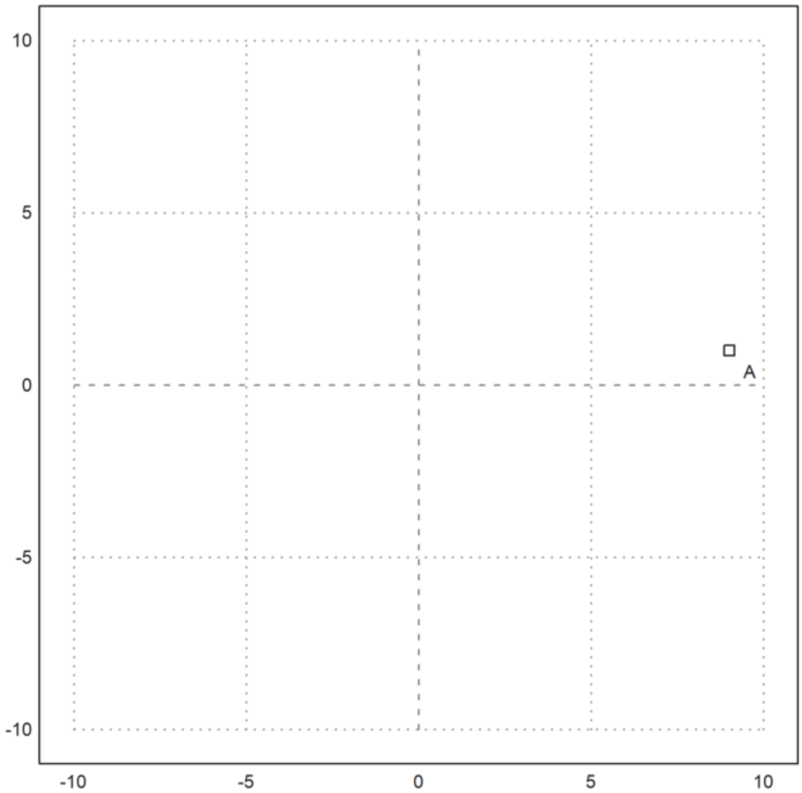
```
>$solve(%,y)| expand
```

$$[y=3-7\,x]$$

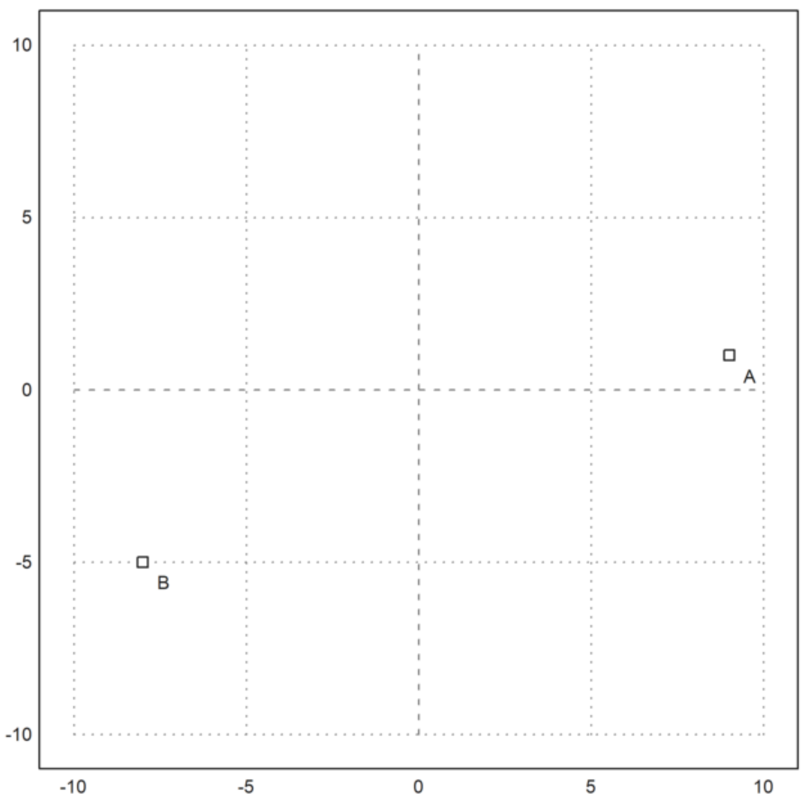
2. Tentukan persamaan garis berat yang melalui ruas garis AC dengan

$$A(9,1), B(-8,-5), C(0,5)$$

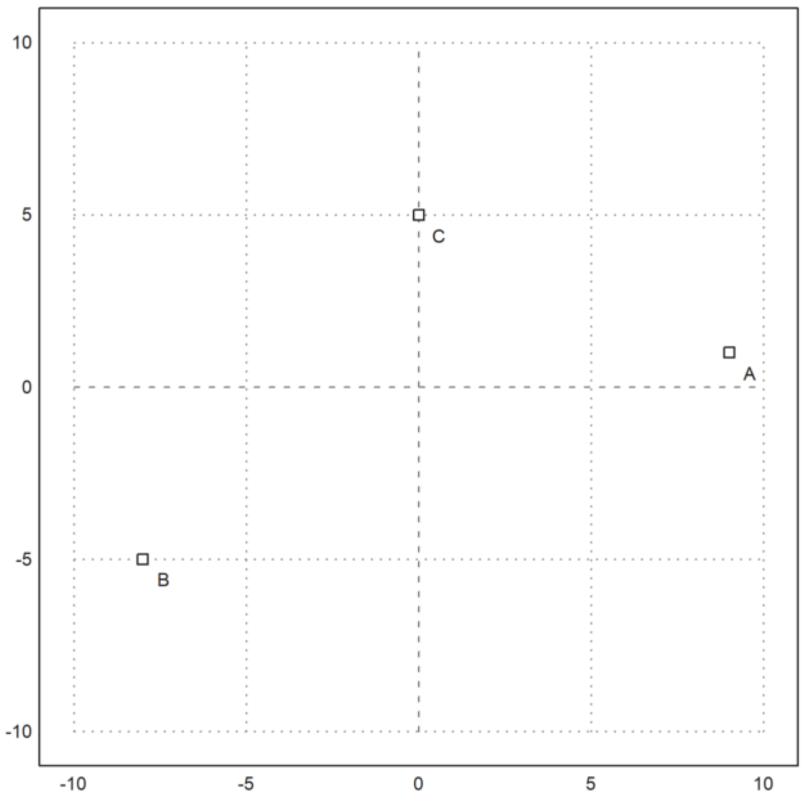
```
>setPlotRange(10);
>A=[9,1]; plotPoint(A,"A"):
```



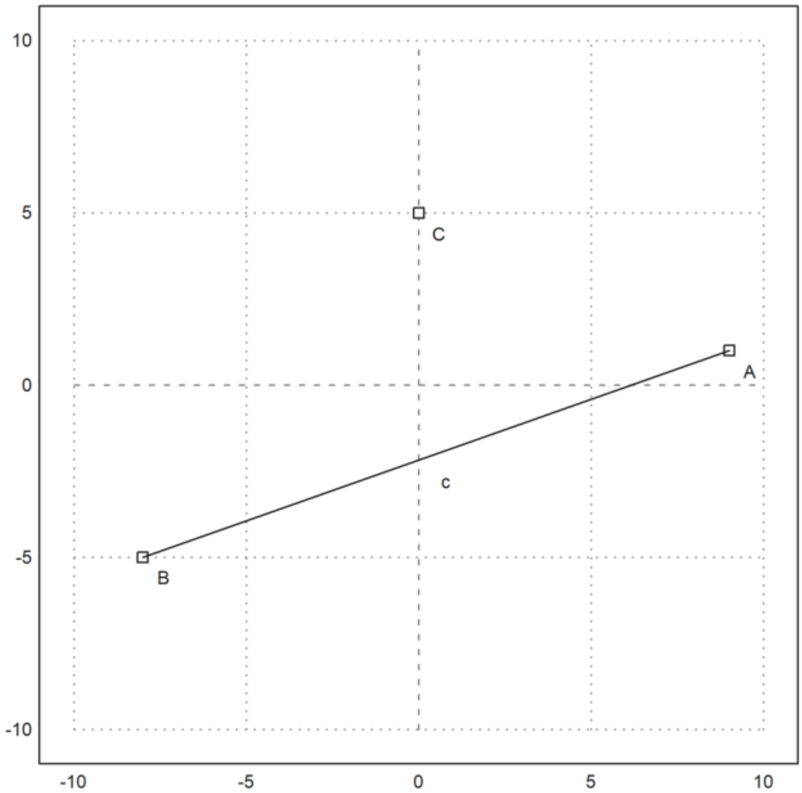

```
>B=[-8,-5]; plotPoint(B,"B"):
```



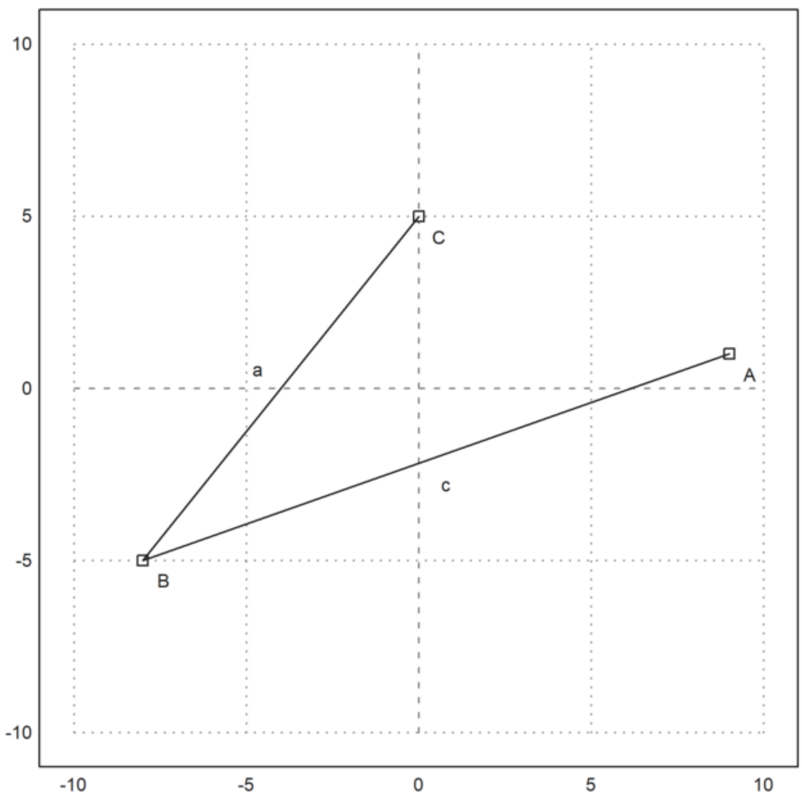
```
>C=[0,5]; plotPoint(C,"C"):
```



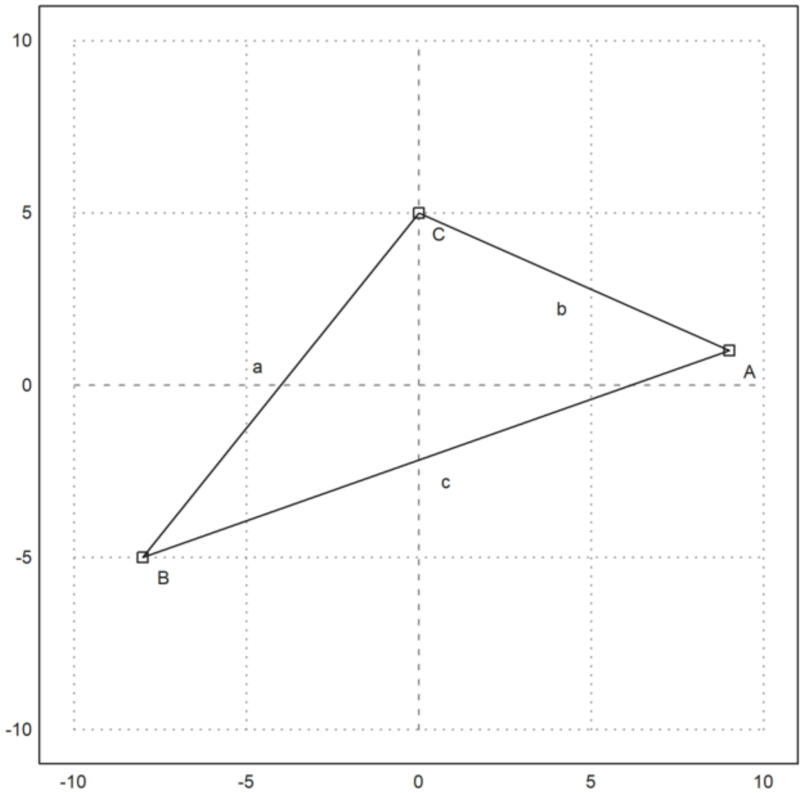

```
>plotSegment(A,B,"c"): // c=AB
```



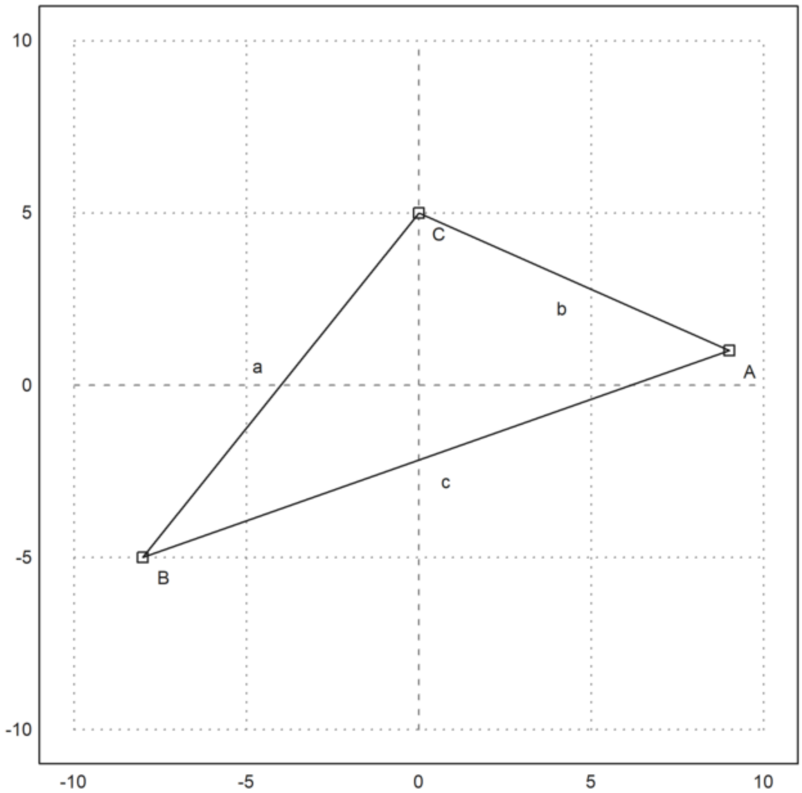

```
>plotSegment(B,C,"a"): // a=BC
```



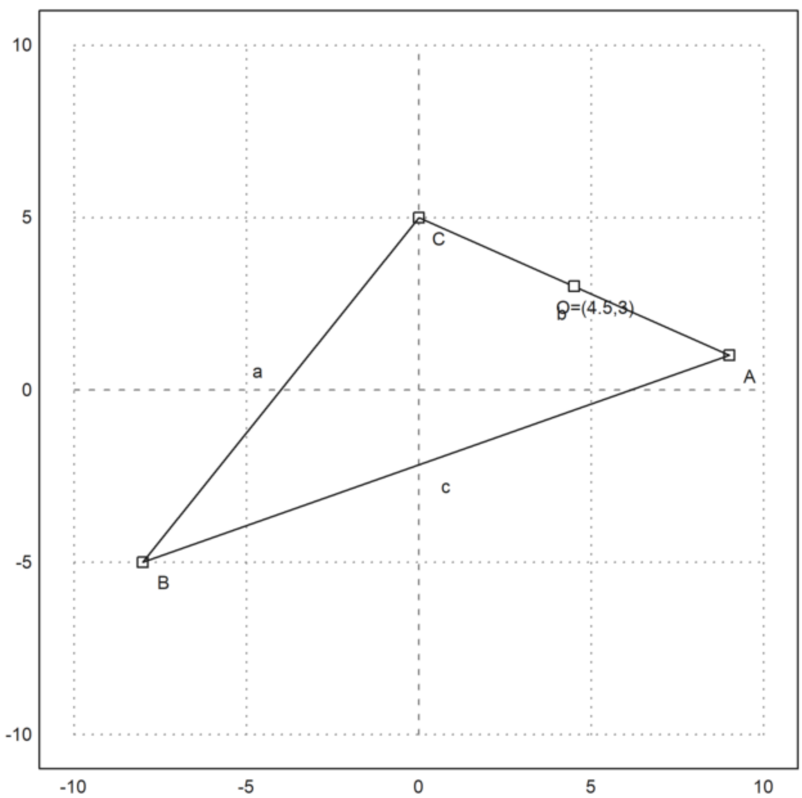
```
>plotSegment(A,C,"b"): // b=AC
```



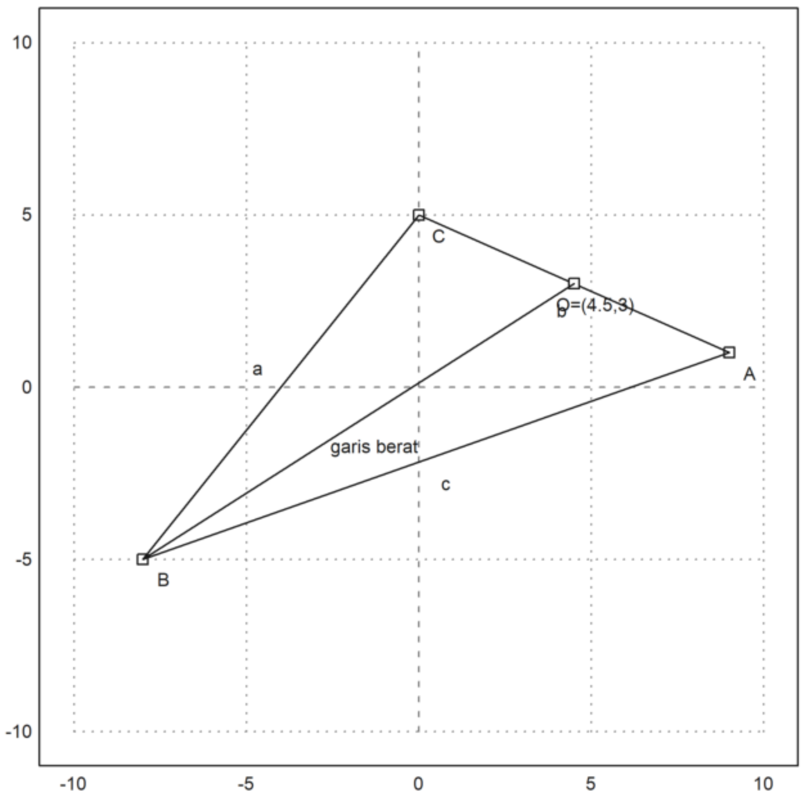

```
>t=middlePerpendicular(A,C):
```




```
>O=lineIntersection(t,lineThrough(A,C)); plotPoint(O,value=1):
```



```
>plotSegment(O,B,"garis berat"):
```




```
>A&=[9,1]
```

[9, 1]

```
>B&=[-8,-5]
```

[- 8, - 5]

```
>C&=[0,5]
```

[0, 5]

```
>t&=middlePerpendicular(A,C)
```

```
                    57
       [9, - 4, --]
                    2
```

```
>O&=lineIntersection(t,lineThrough(A,C))
```

```
       9
      [-, 3]
       2
```

```
>$getLineEquation(lineThrough(O,B),x,y)
```

$$8\,x-\frac{25\,y}{2}=-\frac{3}{2}$$

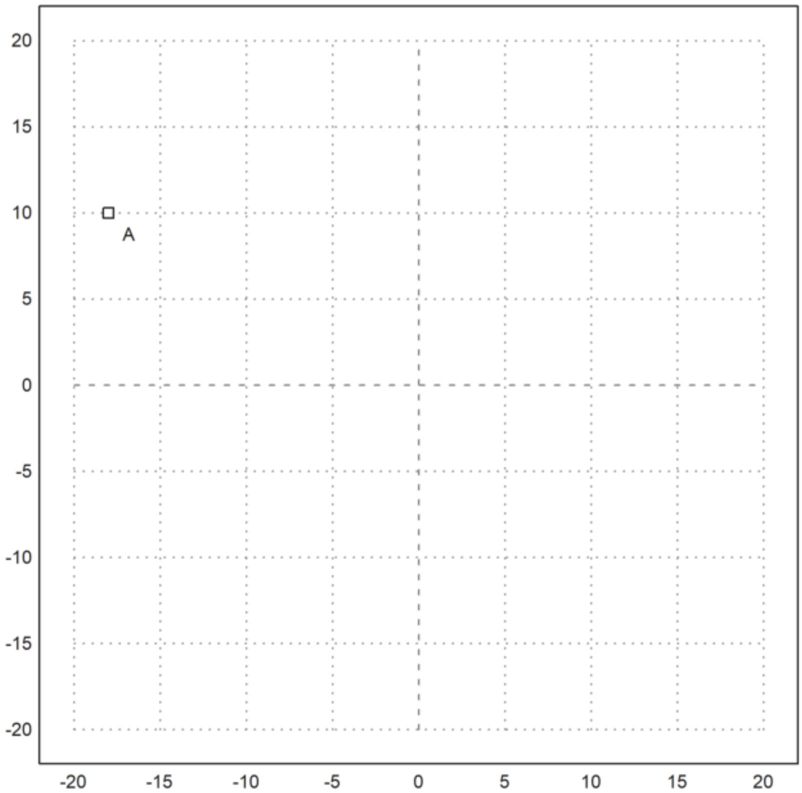
```
>$solve(%,y)| expand
```

$$\left[y=\frac{16\,x}{25}+\frac{3}{25}\right]$$

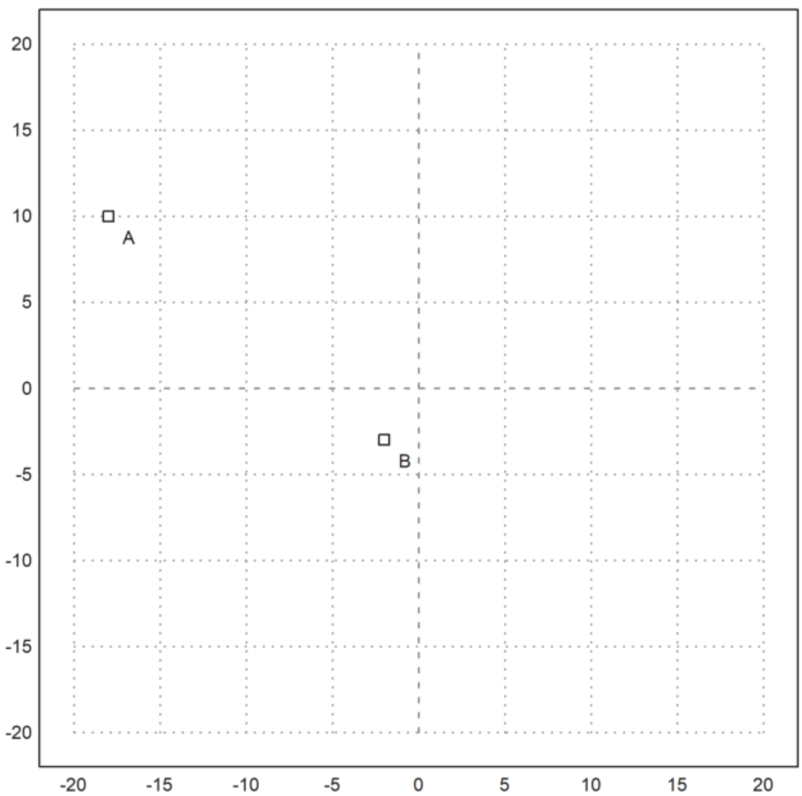
3. Tentukan persamaan garis berat dari semua titik sudut dengan

$$A(-18,10),B(-2,-3),C(17,7)$$

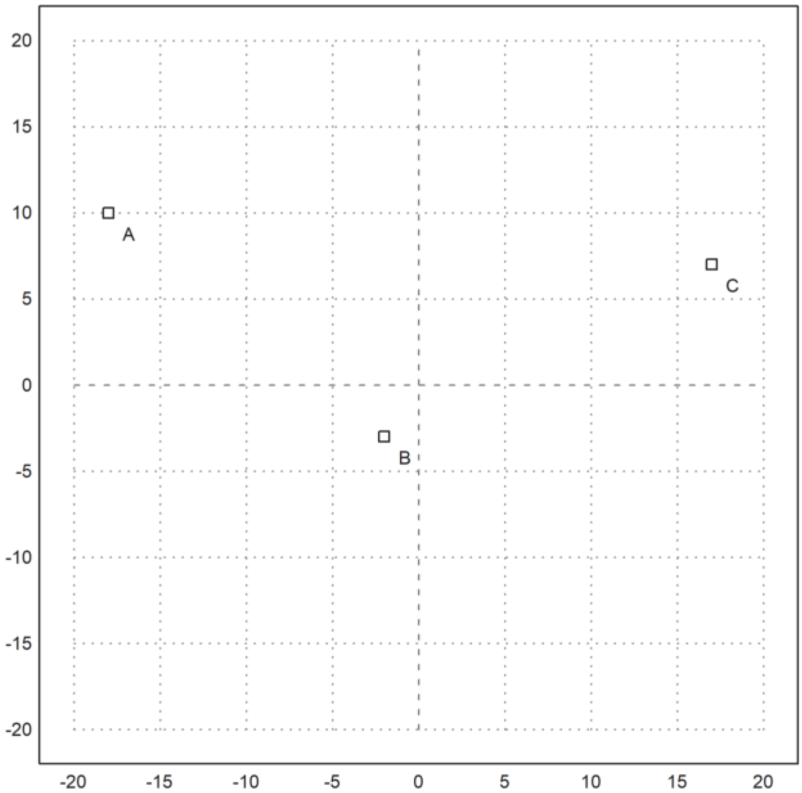
```
>setPlotRange(20);
>A=[-18,10]; plotPoint(A,"A"):
```



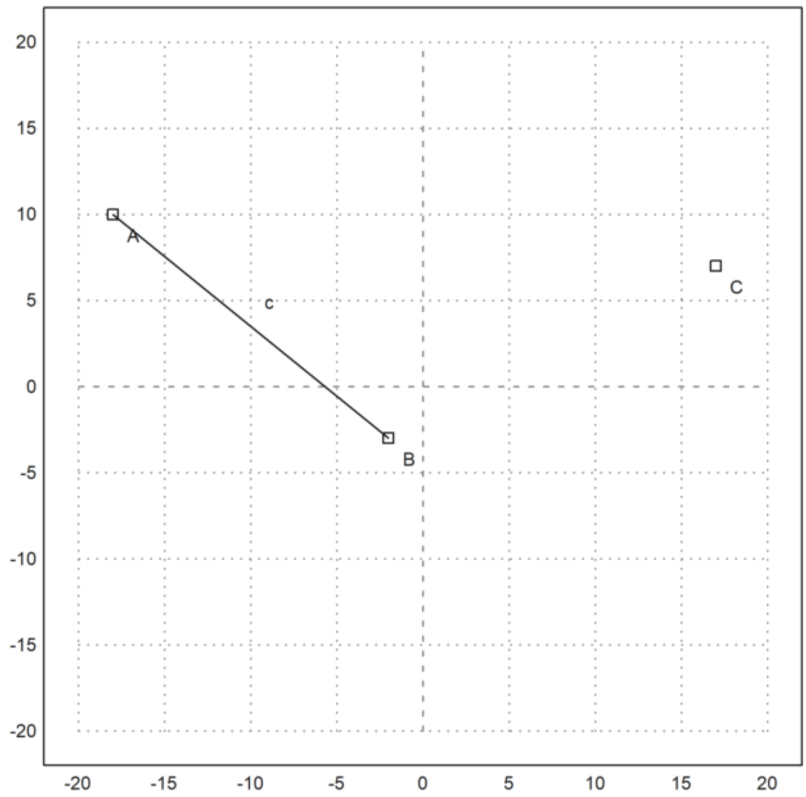

```
>B=[-2,-3]; plotPoint(B,"B"):
```



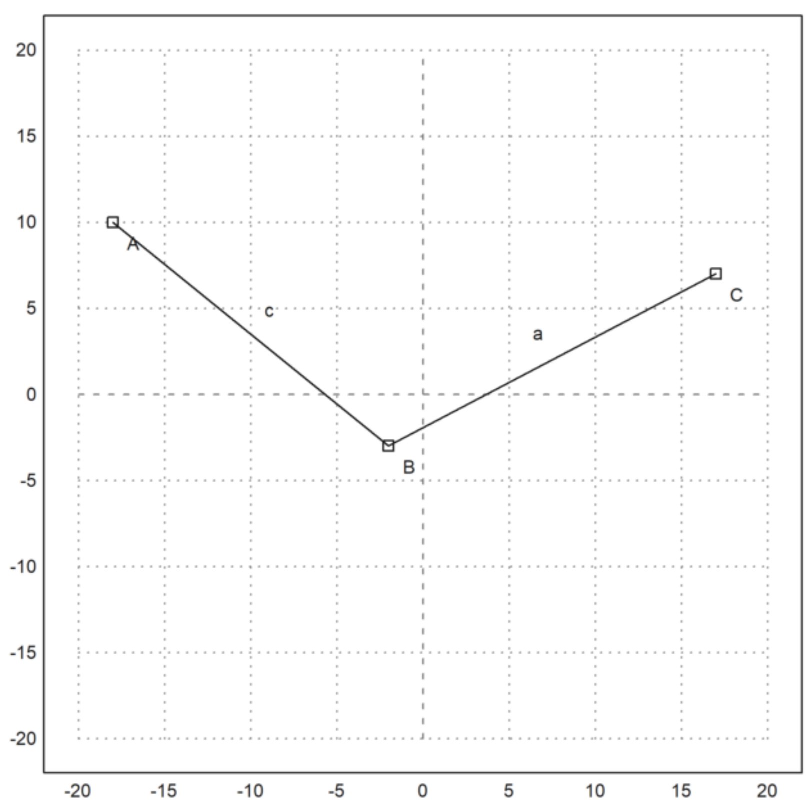
```
>C=[17,7]; plotPoint(C,"C"):
```



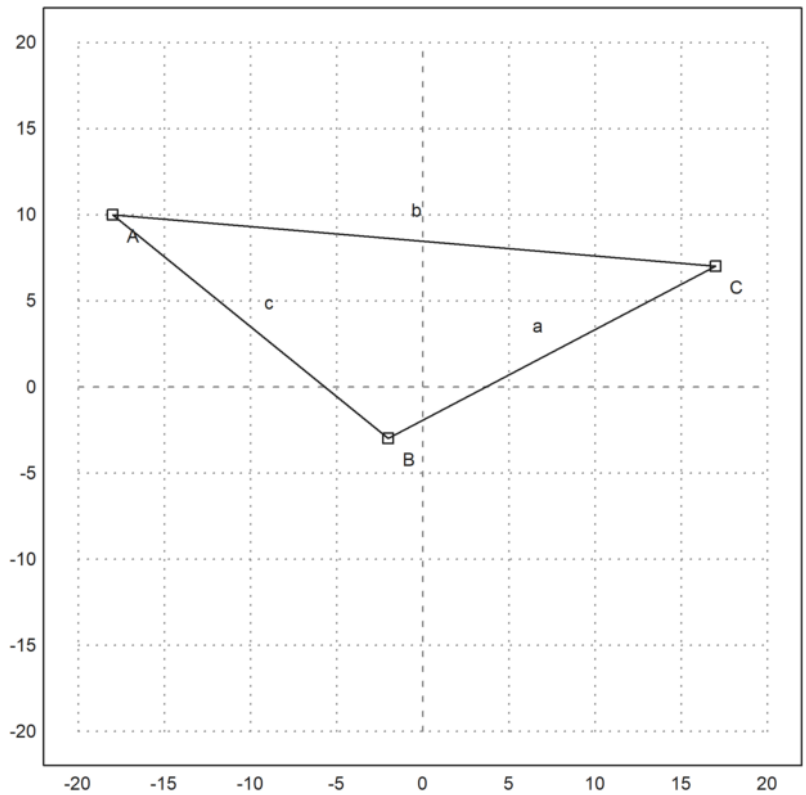

```
>plotSegment(A,B,"c"): // c=AB
```



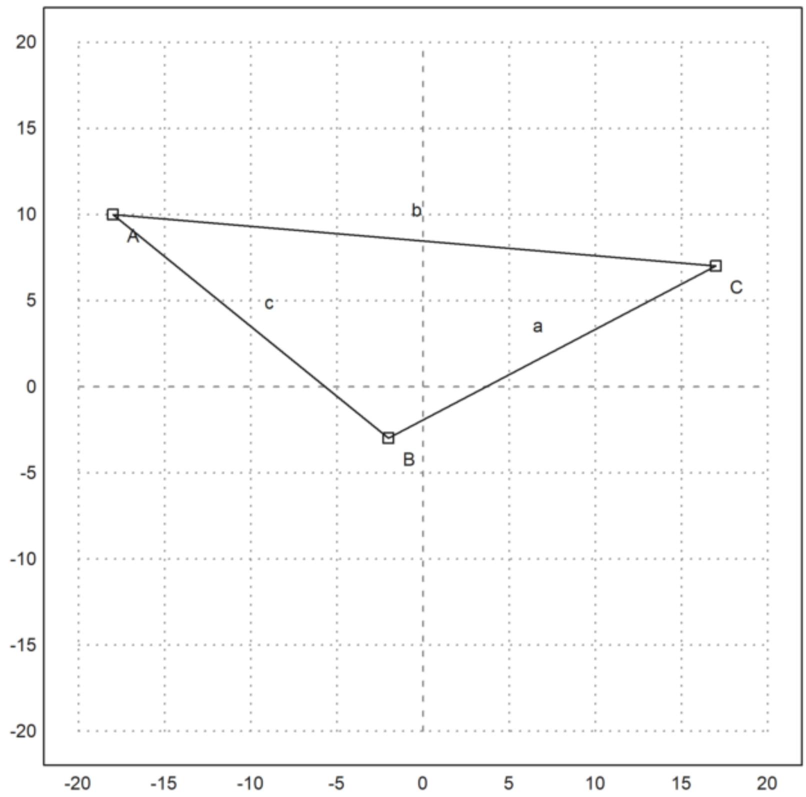

```
>plotSegment(B,C,"a"): // a=BC
```



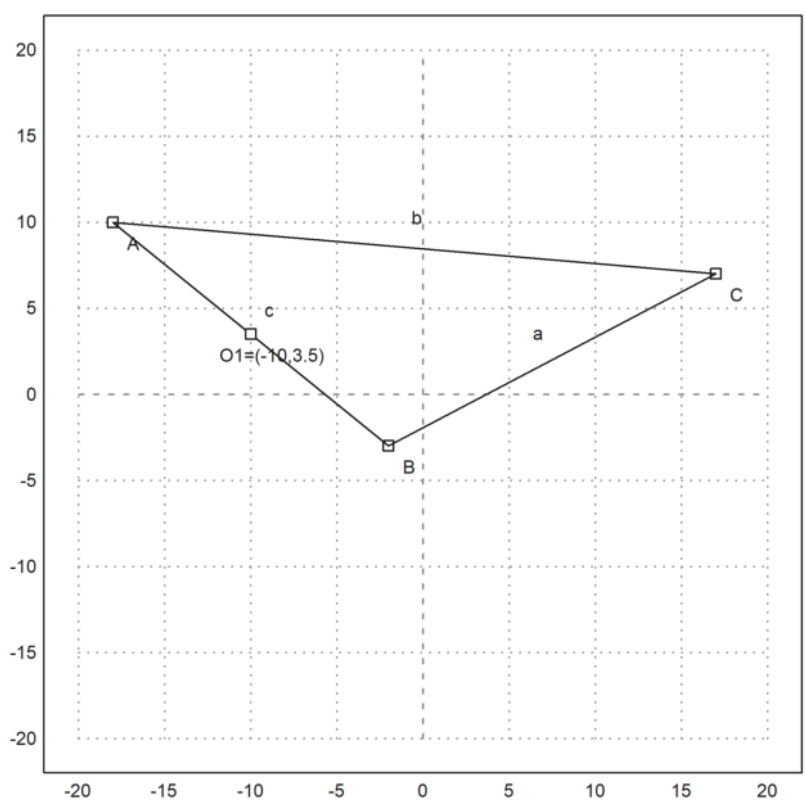
```
>plotSegment(A,C,"b"): // b=AC
```



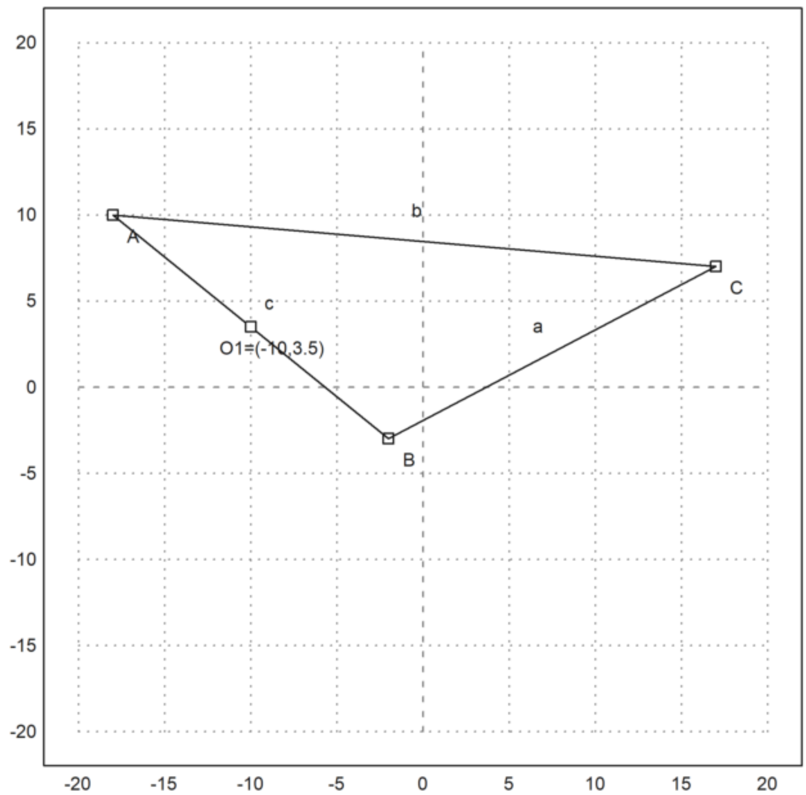

```
>t1=middlePerpendicular(A,B):
```



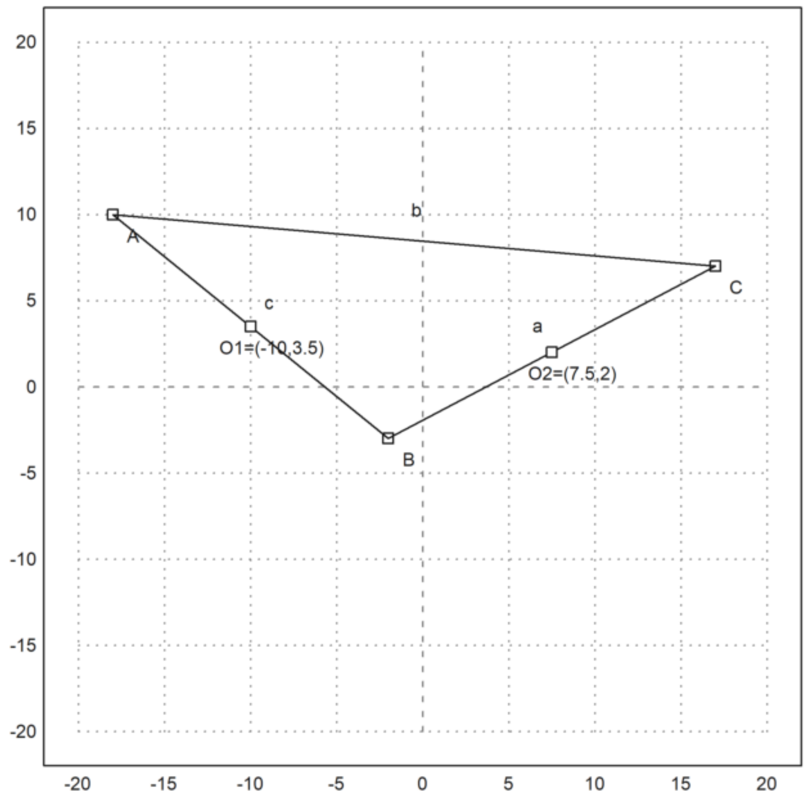

```
>O1=lineIntersection(t1,lineThrough(A,B)); plotPoint(O1,value=1):
```



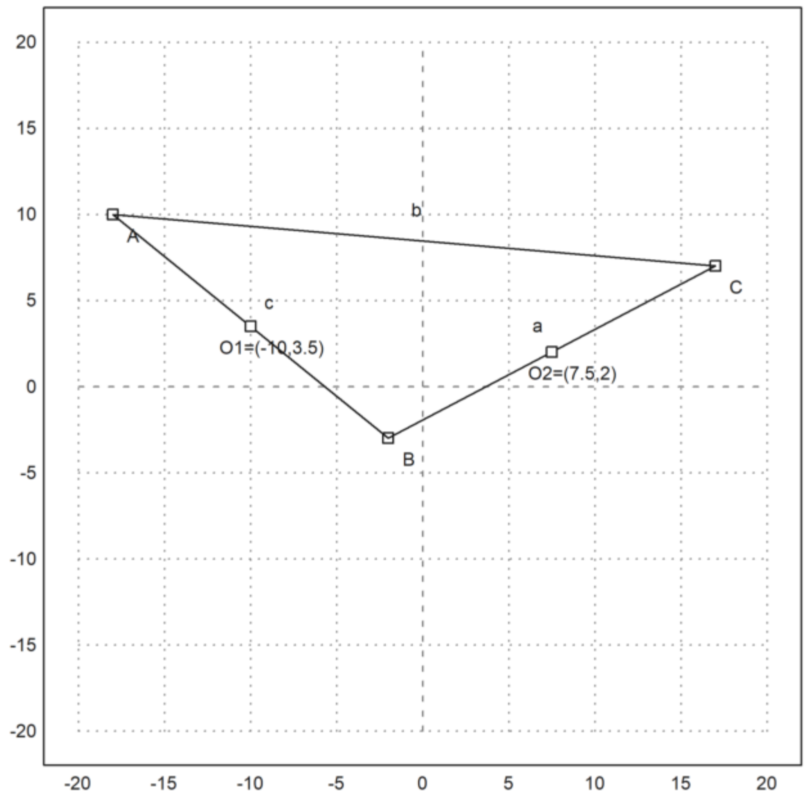
```
>t2=middlePerpendicular(C,B):
```



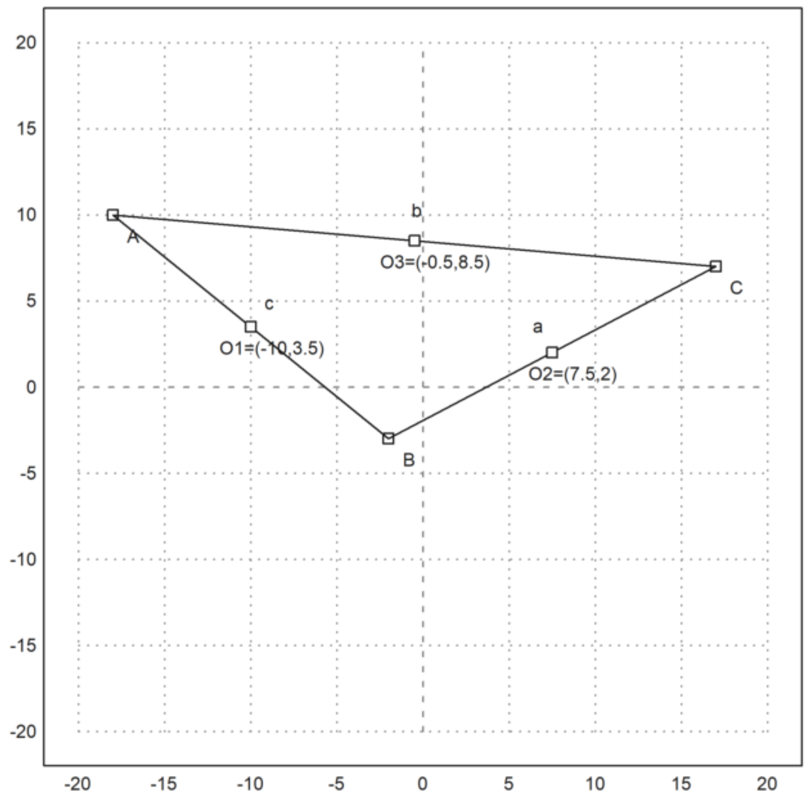

```
>O2=lineIntersection(t2,lineThrough(C,B)); plotPoint(O2,value=1):
```



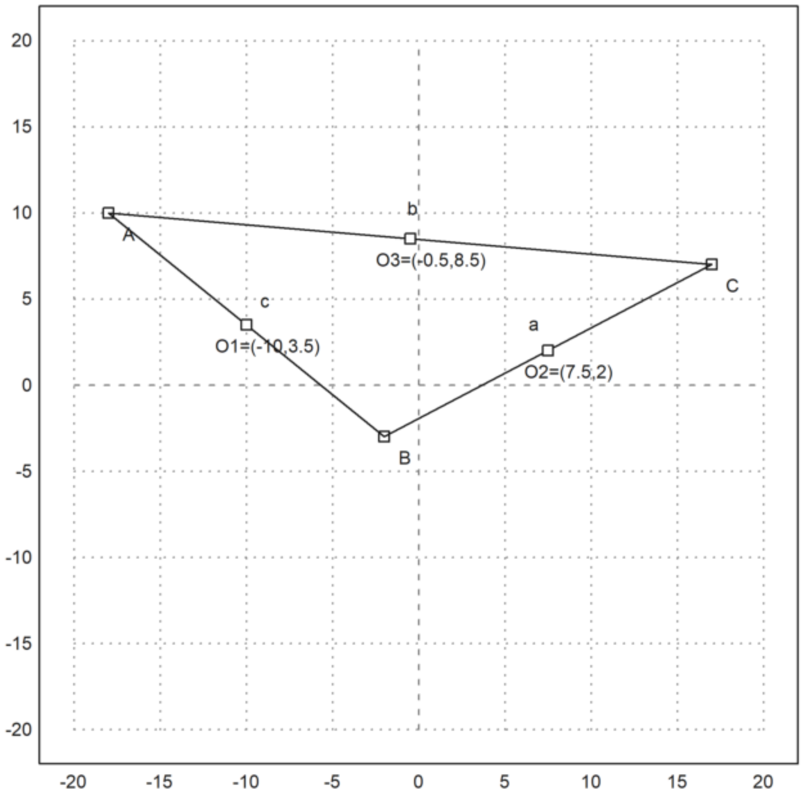

```
>t3=middlePerpendicular(A,C):
```



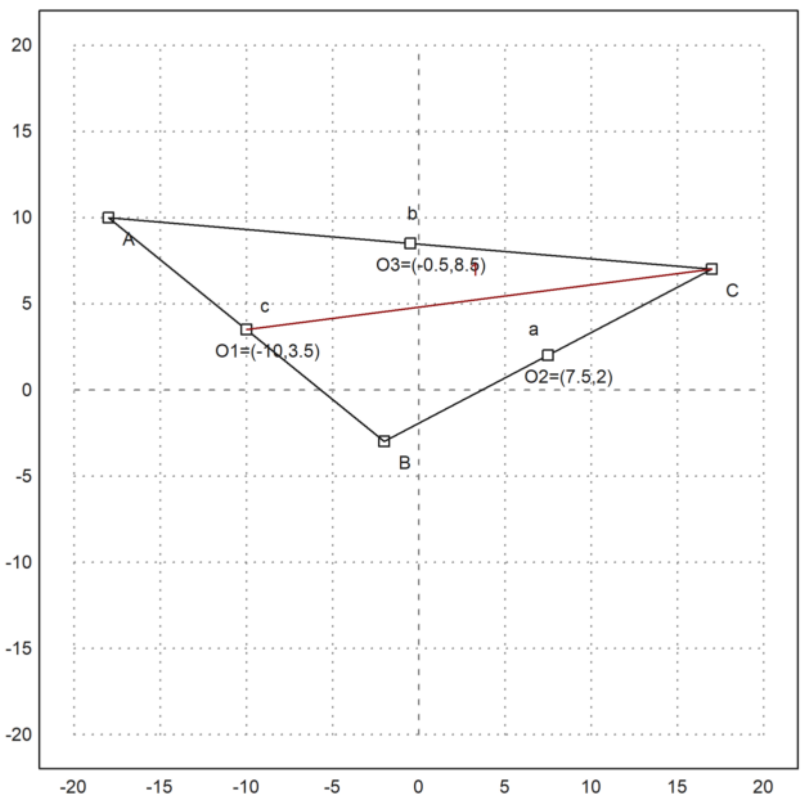
```
>O3=lineIntersection(t3,lineThrough(A,C)); plotPoint(O3,value=1):
```



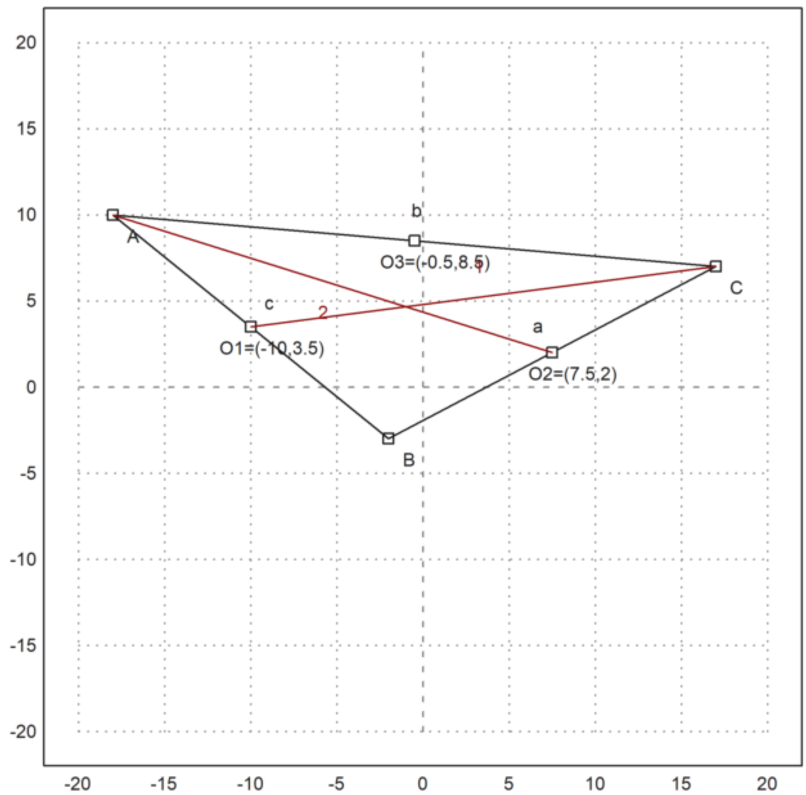

```
>color(2):
```



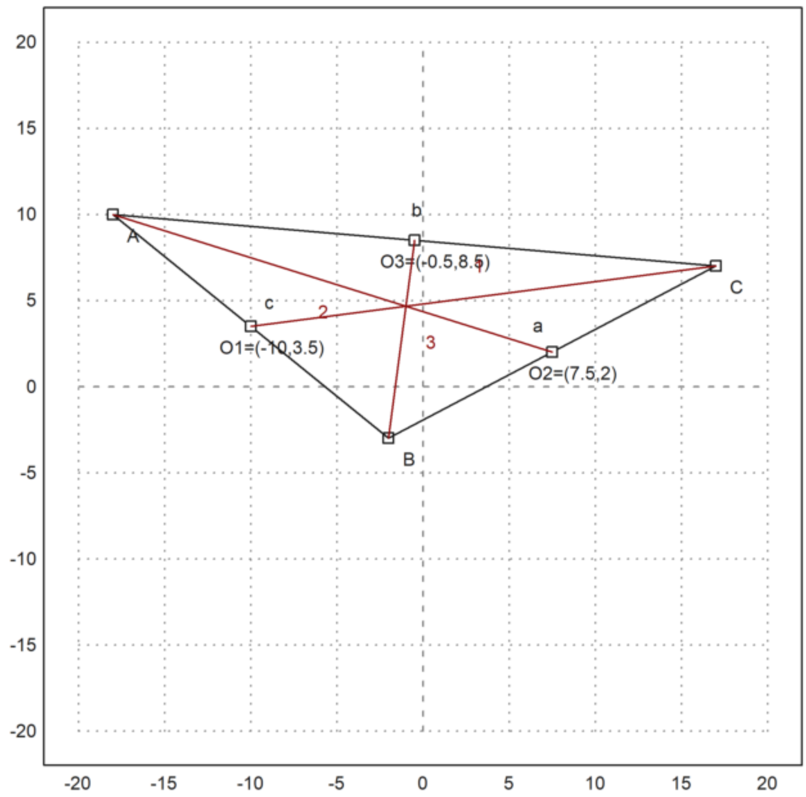

```
>plotSegment(O1,C,"1"):
```



```
>plotSegment(O2,A,"2"):
```

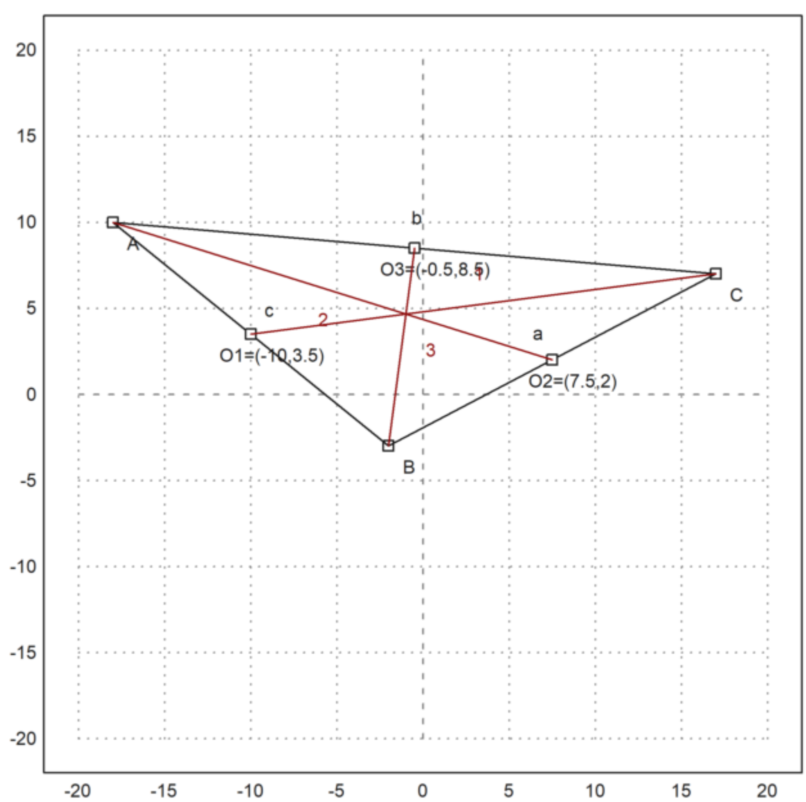



```
>plotSegment(O3,B,"3"):
```



A
b
O3=(-0.5,8.5)
1
C
c
2
a
3
O1=(-10,3.5)
O2=(7.5,2)
B


```
>color(1):
```



A
b
O3=(-0.5,8.5)
1
C
c
2
a
3
O1=(-10,3.5)
O2=(7.5,2)
B

```
>A&=[-18,10]
```

```
[- 18, 10]
```

```
>B&=[-2,-3]
```

```
[- 2, - 3]
```

```
>C&=[5,-16]
```

```
[5, - 16]
```

```
>t1&=middlePerpendicular(A,B)
```

```
                411
[- 16, 13, ---]
                 2
```

```
>O1&=lineIntersection(t1,lineThrough(A,B))
```

```
         7
[- 10, -]
         2
```

```
>t2&=middlePerpendicular(C,B)
```

```
[7, - 13, 134]
```

```
>O2&=lineIntersection(t2,lineThrough(C,B))
```

```
 3    19
[-, - --]
 2    2
```

```
>t3&=middlePerpendicular(A,C)
```

```
                    143
[- 23,  26,  ---]
                     2
```

```
>O3&=lineIntersection(t3,lineThrough(A,C))
```

```
      13
[-  --,  - 3]
      2
```

```
>$getLineEquation(lineThrough(O1,C),x,y)
```

$$15\,y+\frac{39\,x}{2}=-\frac{285}{2}$$

```
>$solve(%,y)| expand
```

$$\left[y=-\frac{13\,x}{10}-\frac{19}{2}\right]$$

```
>$getLineEquation(lineThrough(O2,A),x,y)
```

$$-\frac{39\,y}{2}-\frac{39\,x}{2}=156$$

```
>$solve(%,y)| expand
```

$$[y=-x-8]$$

```
>$getLineEquation(lineThrough(O3,B),x,y)
```

$$\frac{9\,y}{2}=-\frac{27}{2}$$

```
>$solve(%,y)| expand
```

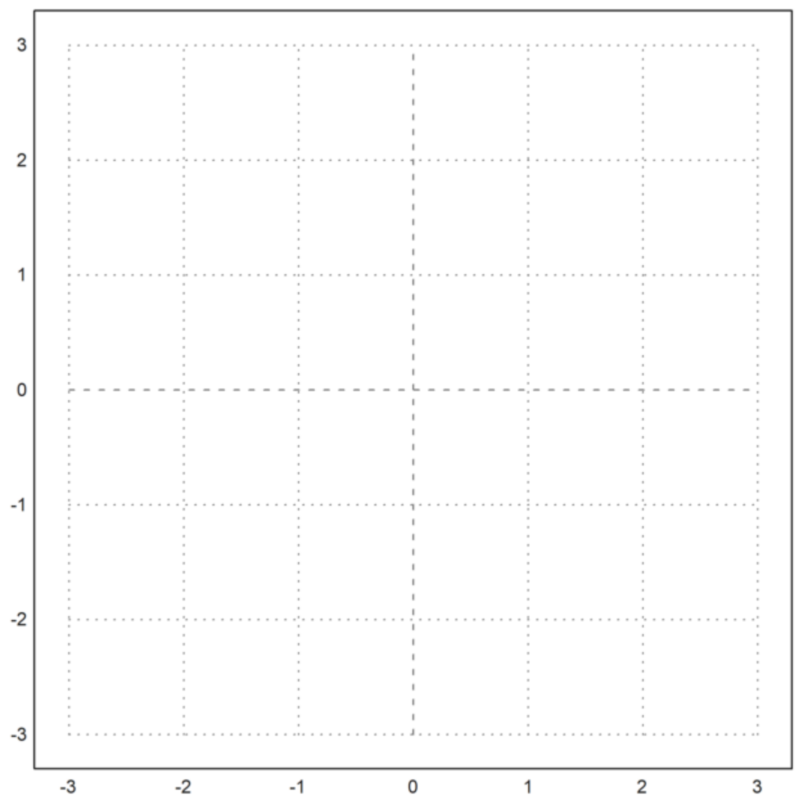
$$[y=-3]$$

Mencari persamaan garis tinggi

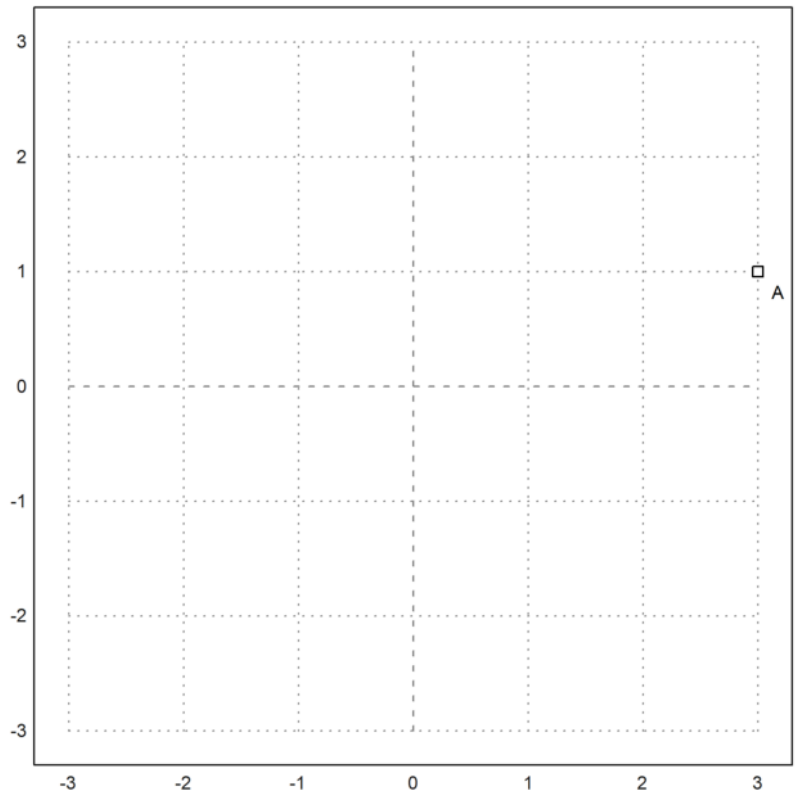
Garis tinggi pada segitiga adalah ruas garis yang ditarik dari sudut segitiga ke sisi yang berlawanan secara tegak lurus. Dengan kata lain, garis tinggi merupakan jarak vertikal dari salah satu sudut segitiga ke sisi yang berlawanan, dan garis ini membentuk sudut siku-siku dengan sisi tersebut.
Dengan EMT kita dapat mencari persamaan garis tinggi yaitu dengan cara:
1. Mendefinisikan titik koordinat
2. Mendefinisikan g dan menggunakan perintah lineThrough(B,A)
3. Mendefinisikan h dan menggunakan perintah perpendicular(C,g)
4. Menggunakan perintah getLineEquation(h,x,y)
5. Menggunakan perintah solve(%,y) | expand
Contoh:
Tentukan garis tinggi dari

$$\triangle ABC, A(3,1); B(0,0); C(2,-2)$$

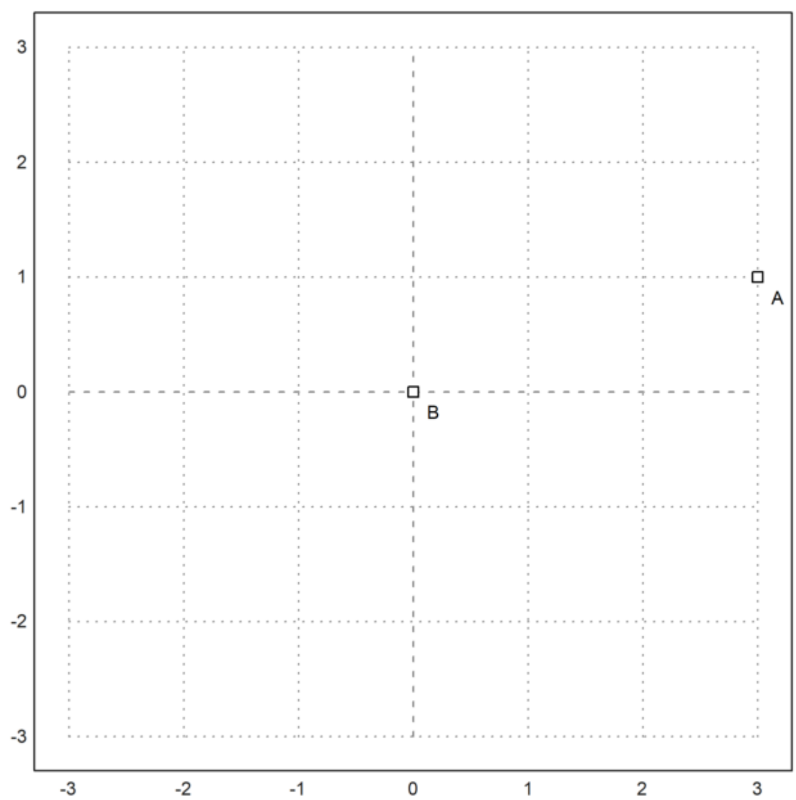
```
>setPlotRange(3):
```



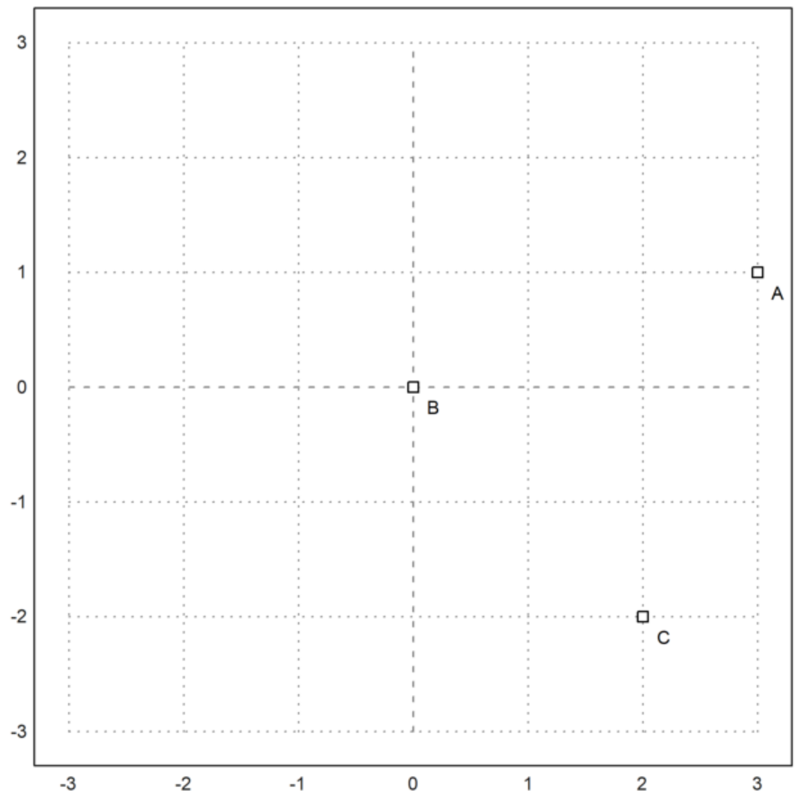

```
>A=[3,1]; plotPoint(A,"A"):
```



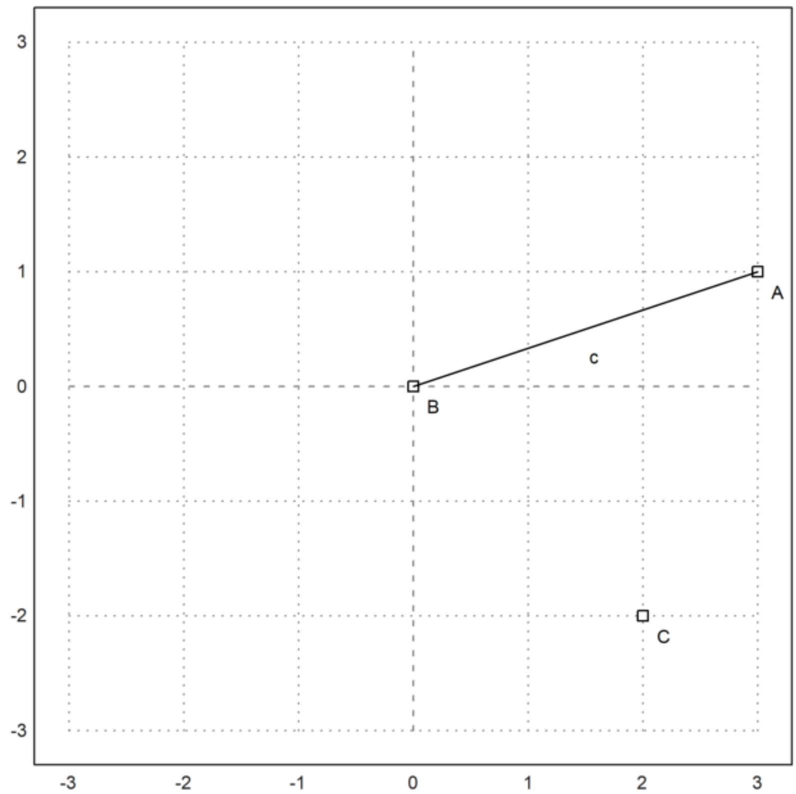

```
>B=[0,0]; plotPoint(B,"B"):
```



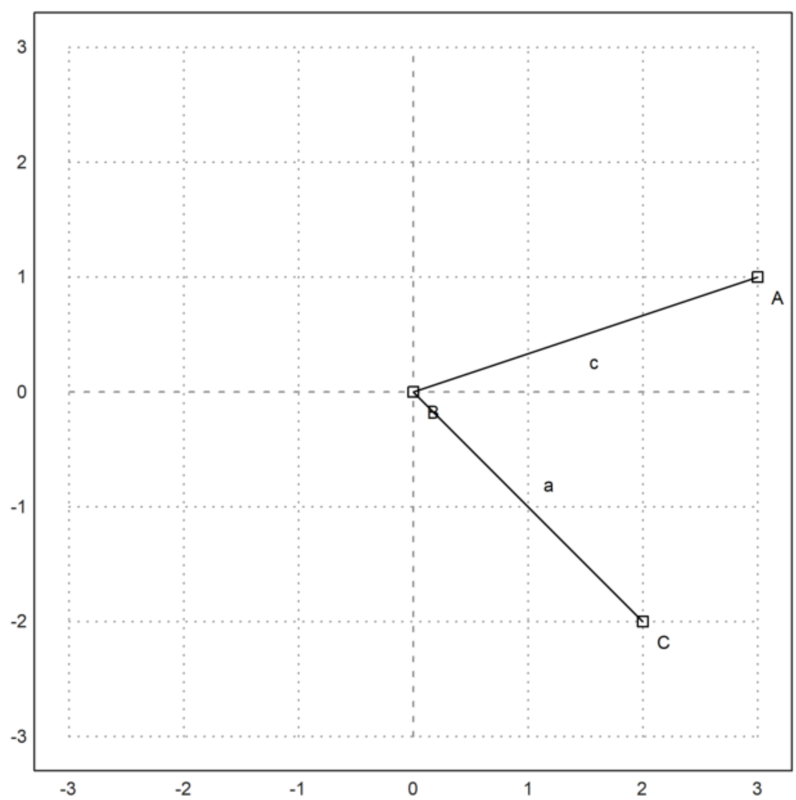
```
>C=[2,-2]; plotPoint(C,"C"):
```



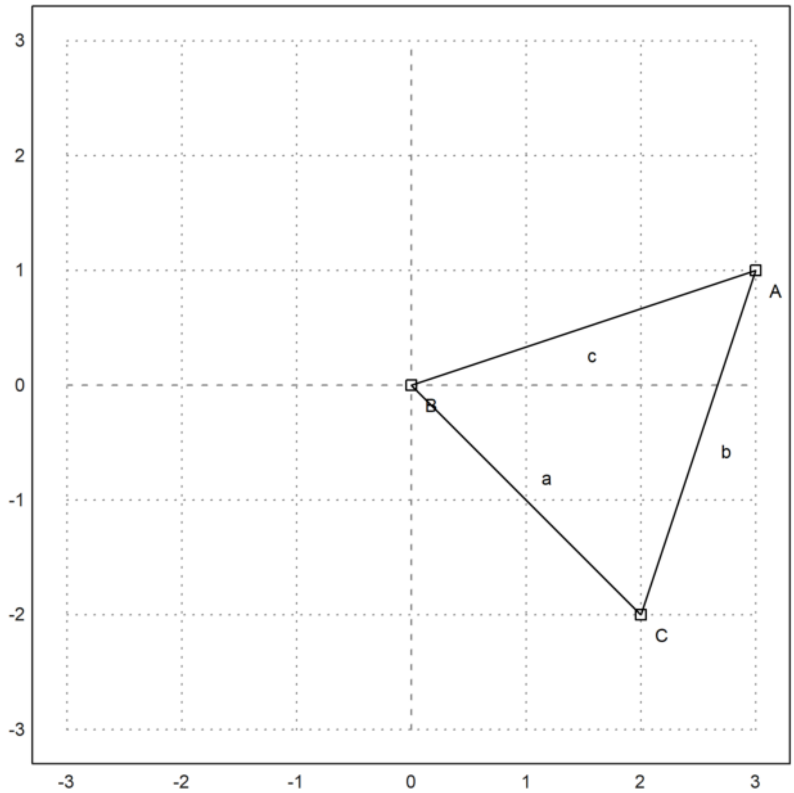

```
>plotSegment(A,B,"c"): // c=AB
```



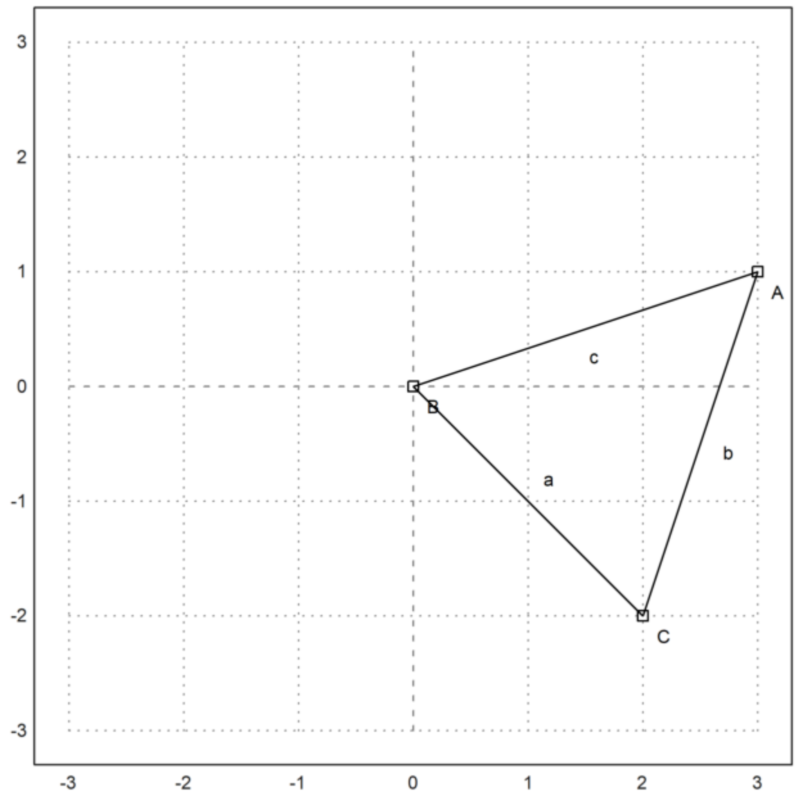

```
>plotSegment(B,C,"a"): // a=BC
```



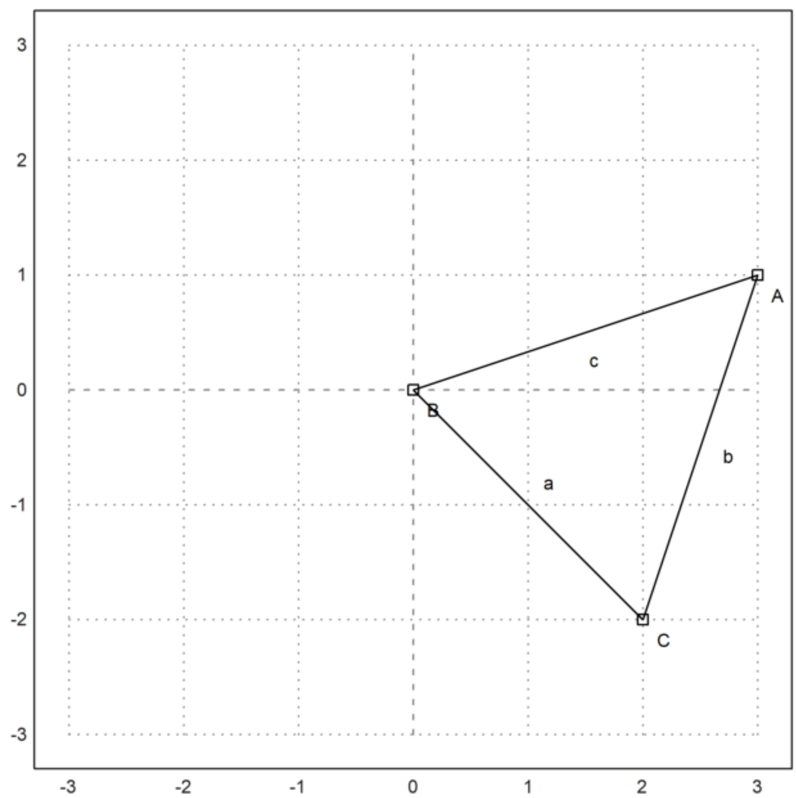
```
>plotSegment(A,C,"b"): // b=AC
```



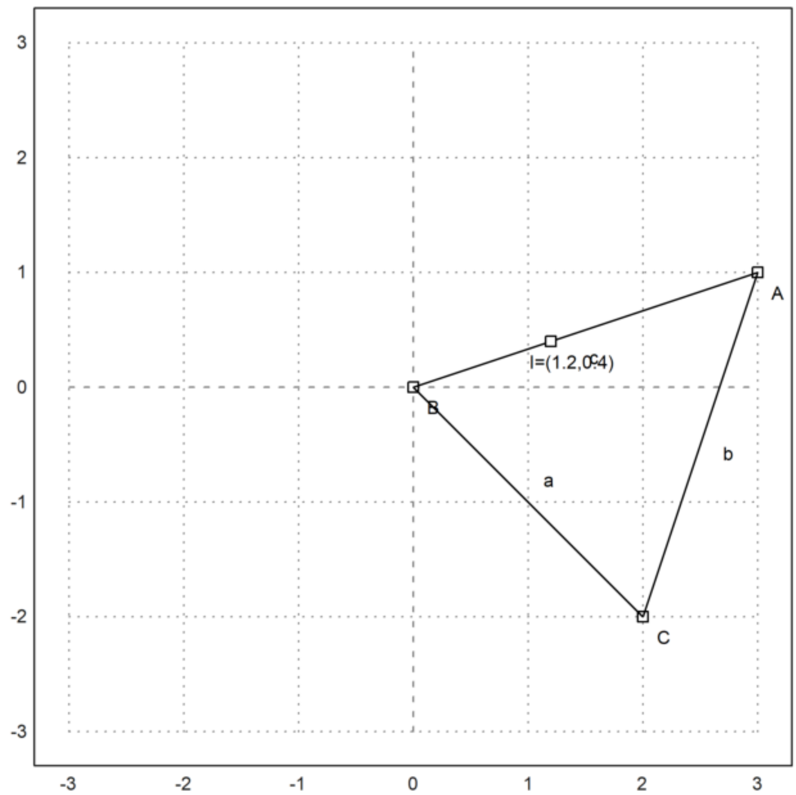

```
>g=lineThrough(B,A):
```



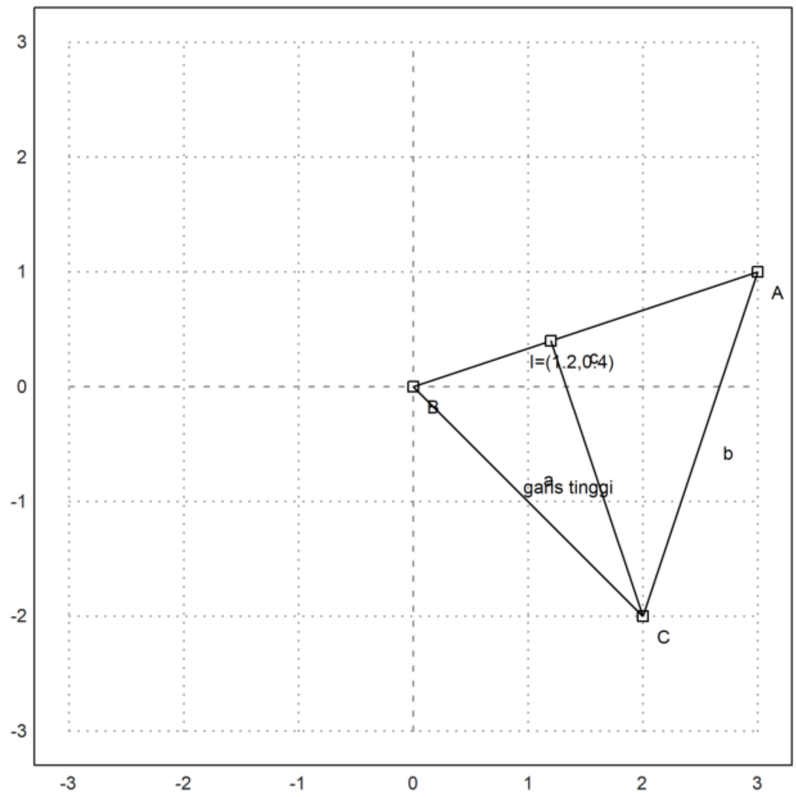

```
>h=perpendicular(C,g):
```



```
>I = lineIntersection(g,h); plotPoint(I,value=1):
```




```
>plotSegment(C,I,"garis tinggi"):
```




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

[3, 1]

```
>B&=[0,0]
```

[0, 0]

```
>C&=[2,-2]
```

[2, - 2]

```
>g&=lineThrough(B,A)
```

[- 1, 3, 0]

```
>h&=perpendicular(C,g)
```

[3, 1, 4]

```
>$getLineEquation(h,x,y)
```

$$y+3\,x=4$$

```
>$solve(%,y) | expand
```

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

Pembuktian menggunakan rumus matematika:
Pembuktian dengan menggunakan rumus matematika:
1. Mencari persamaan garis AB

$$\frac{y-1}{0-1}=\frac{x-3}{0-3}$$
$$\frac{y-1}{-1}=\frac{x-3}{-3}$$

$$-3y+3=-x-+3$$

$$3y=x$$

$$y=\frac{x}{3}$$

maka gradiennya

$$m=\frac{1}{3}$$

karena tegak lurus dengan garis tinggi, maka gradien garis tinggi:

$$m=-3$$

2. Mencari persamaan garis sumbu
dengan gradien:

$$m=-3$$

dan melalui titik:

$$(2,-2)$$

$$(y-y_1)=m(x-x_1)$$

$$(y-(-2))=-3(x-2)$$

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

$$y = -3x + 4$$

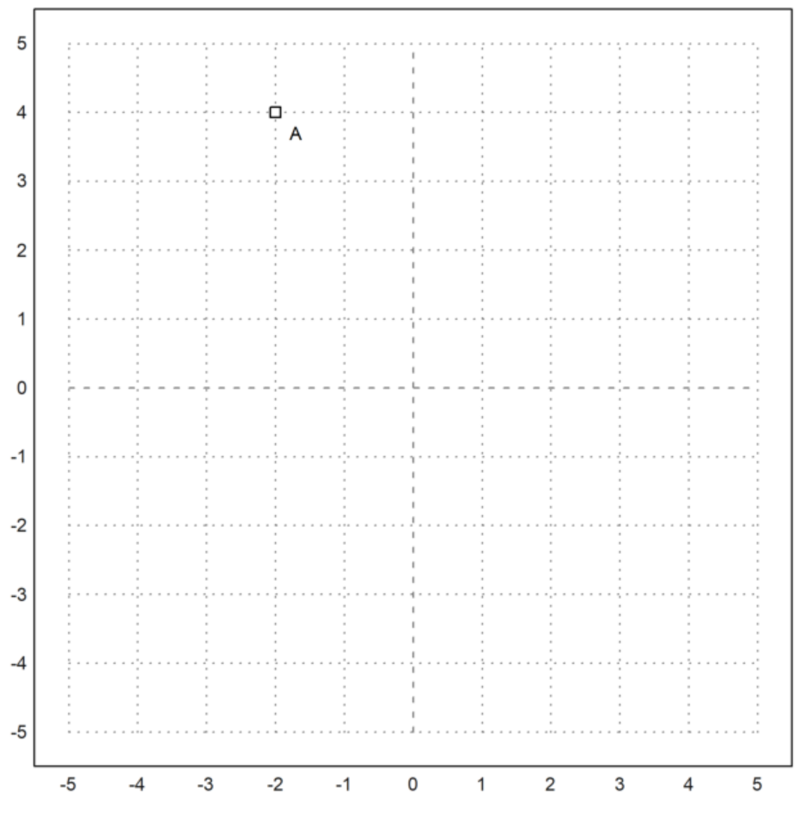
Terbukti **Latihan E**

---

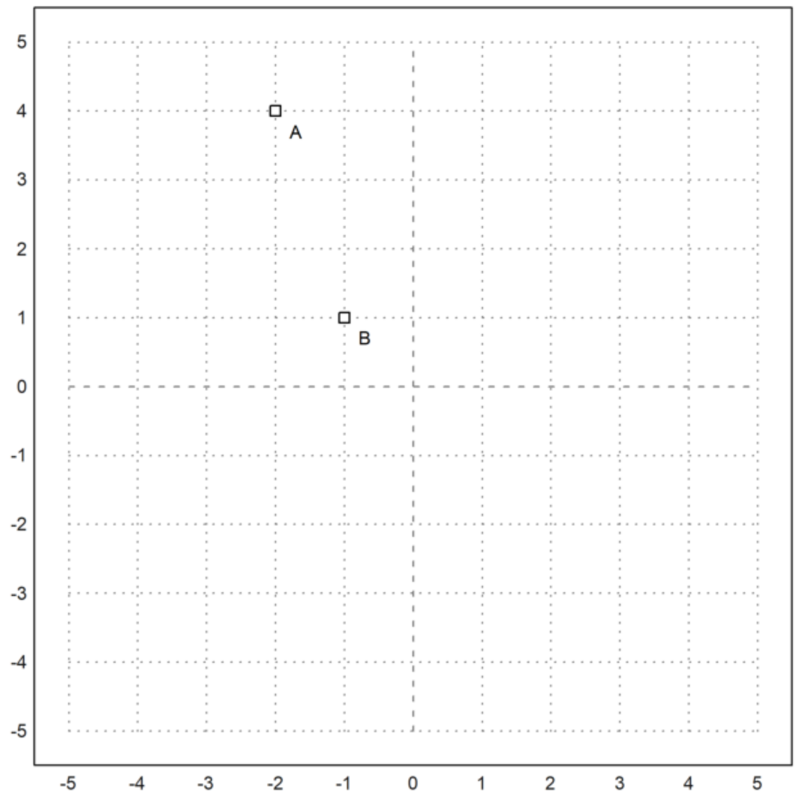
1. Tentukan persamaan tinggi yang diukur dari titik A

$$A(-2,4), B(-1,1), C(5,-4)$$

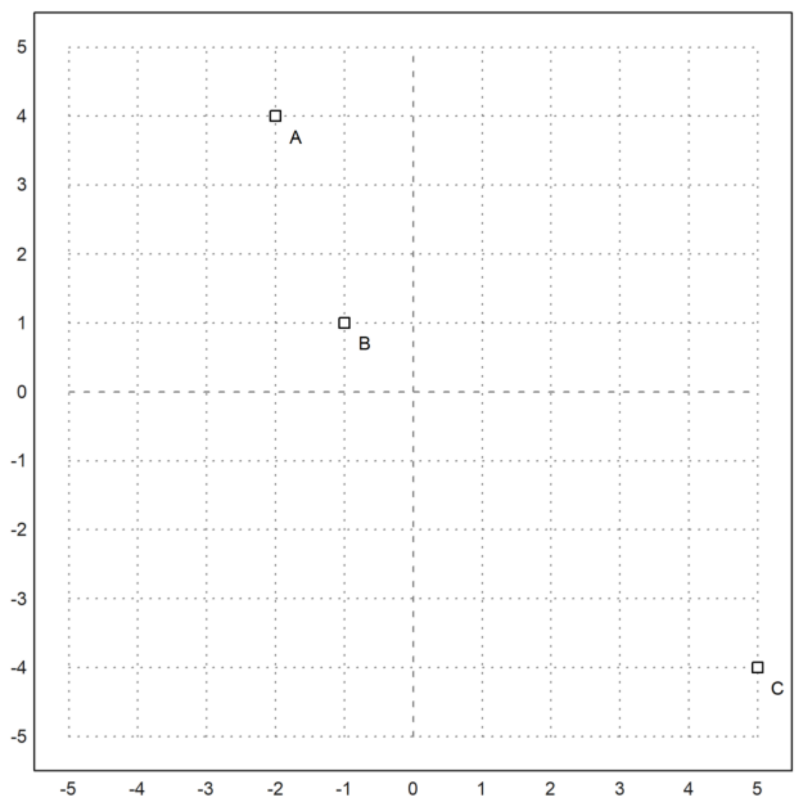
```
>setPlotRange(5);
>A=[-2,4]; plotPoint(A,"A"):
```



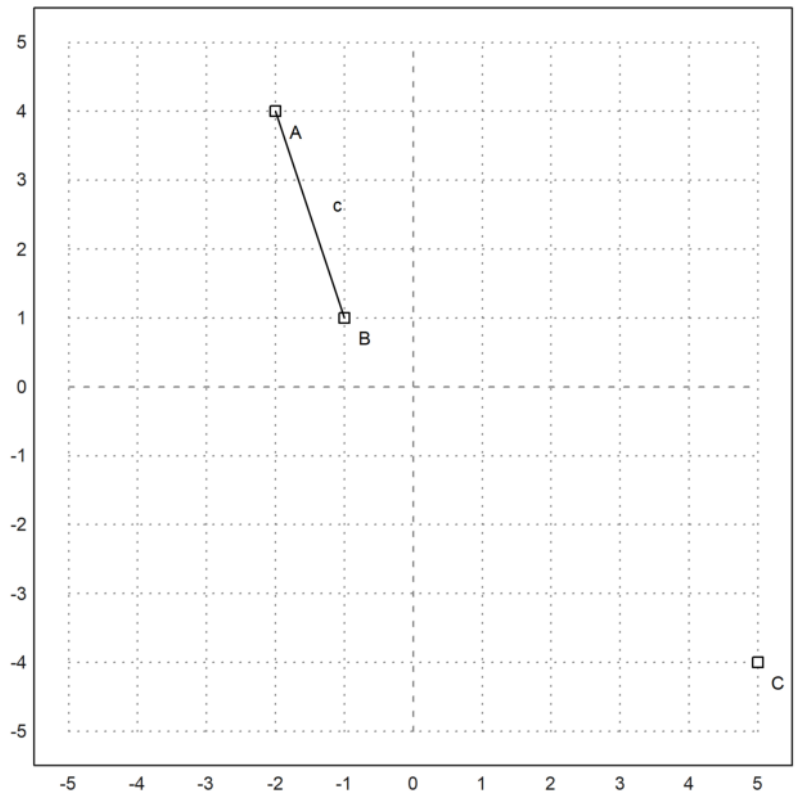

```
>B=[-1,1]; plotPoint(B,"B"):
```



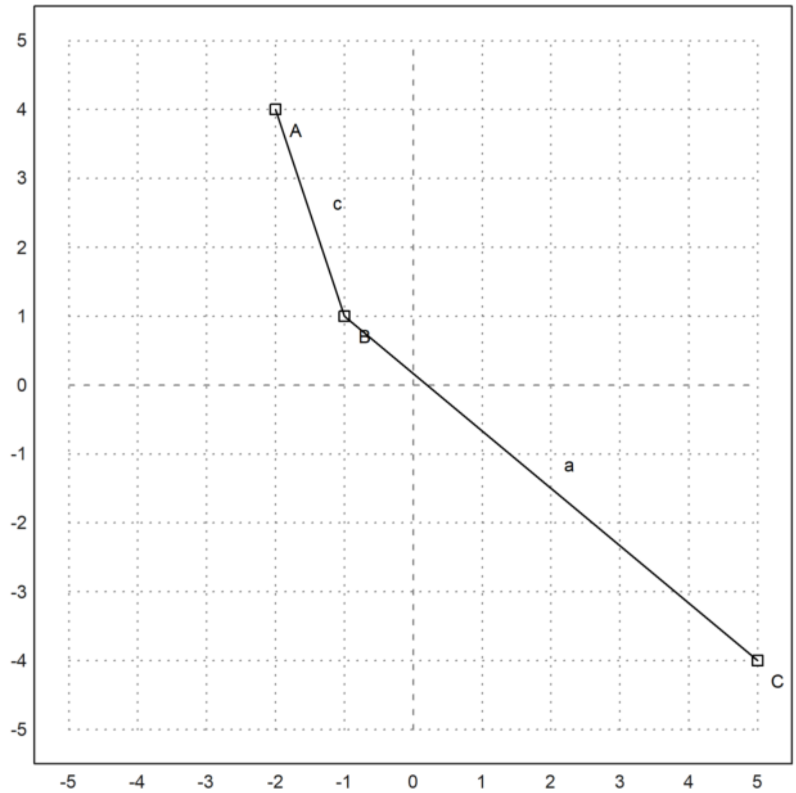

```
>C=[5,-4]; plotPoint(C,"C"):
```



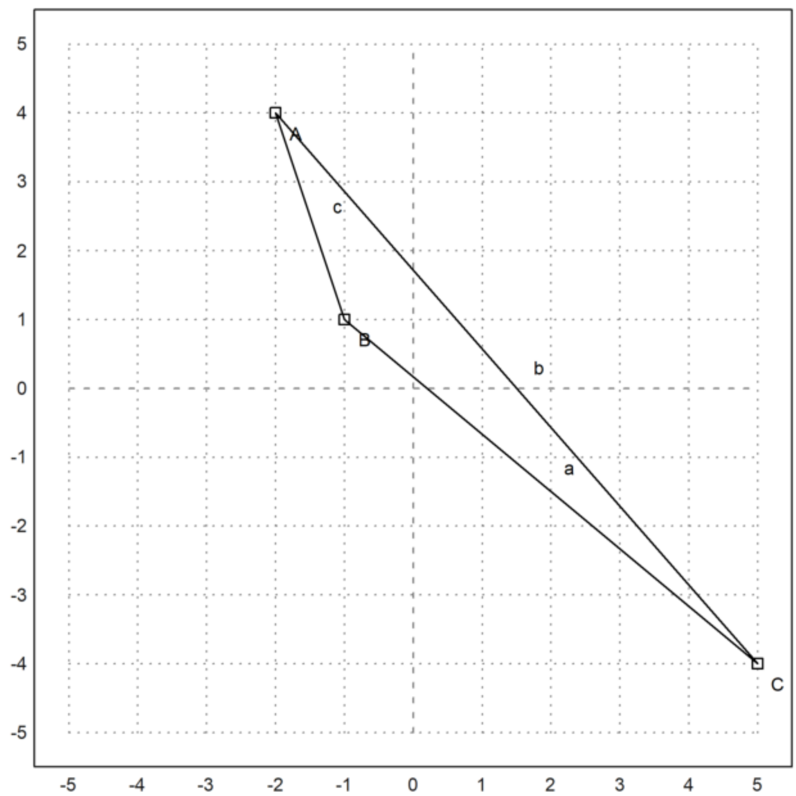
```
>plotSegment(A,B,"c"): // c=AB
```



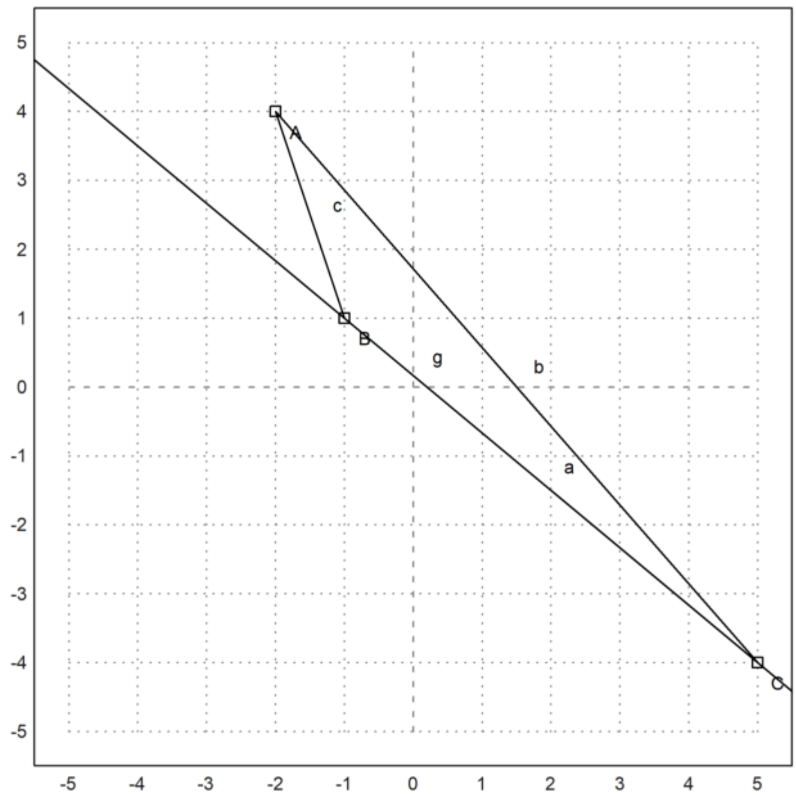

```
>plotSegment(B,C,"a"): // a=BC
```



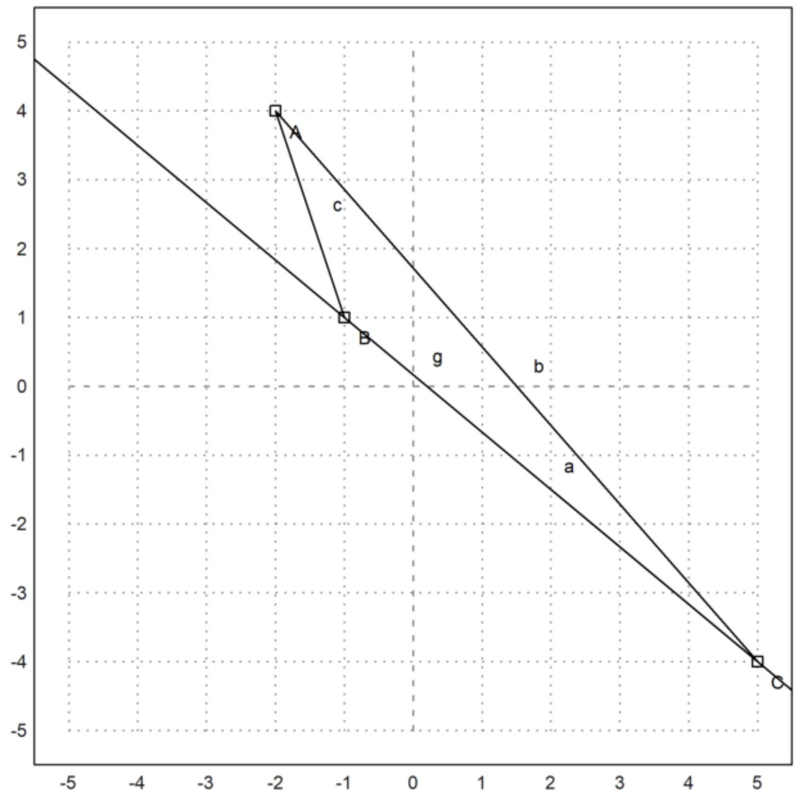

```
>plotSegment(A,C,"b"): // b=AC
```



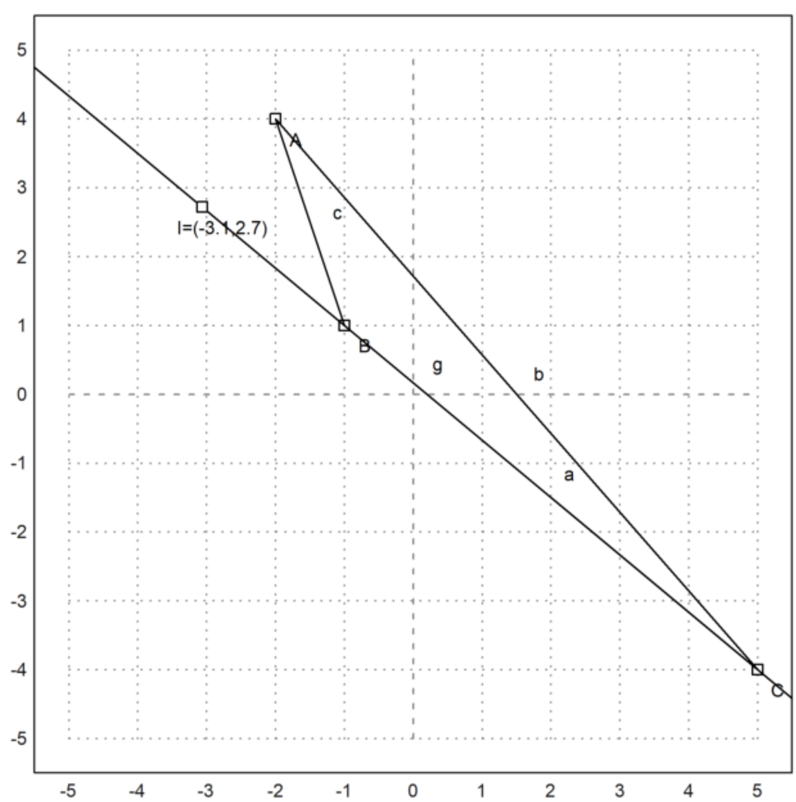
```
>g=lineThrough(B,C); plotLine(g,"g"):
```



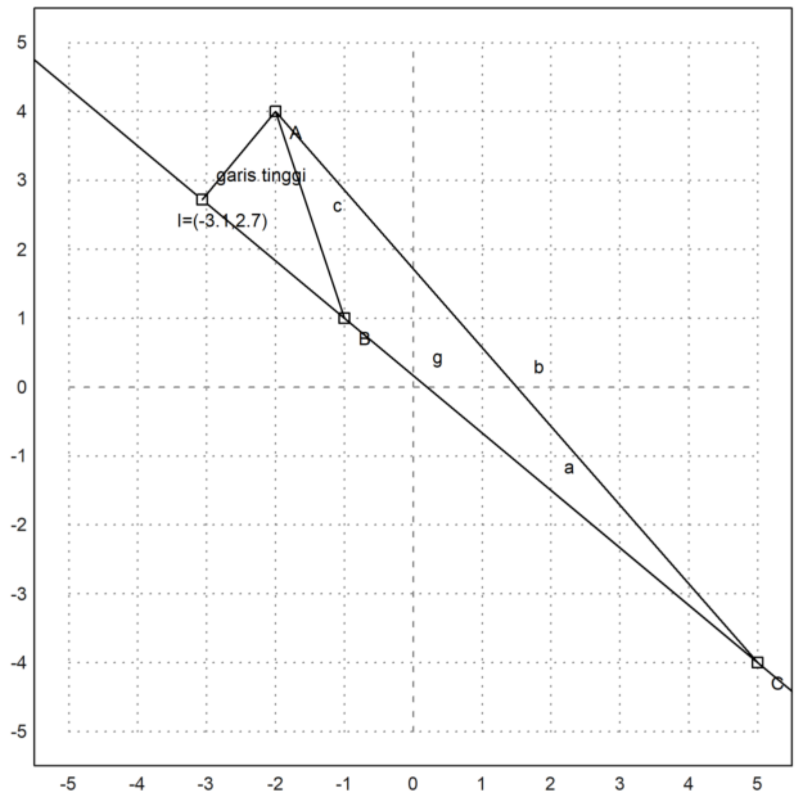

```
>h=perpendicular(A,g):
```




```
>I = lineIntersection(g,h); plotPoint(I,value=1):
```



```
>plotSegment(A,I,"garis tinggi"):
```




```
>A&=[-2,4]
```

[- 2, 4]

```
>B&=[-1,1]
```

[- 1, 1]

```
>C&=[5,-4]
```

[5, - 4]

```
>g&=lineThrough(B,C)
```

```
[5, 6, 1]
```

```
>h&=perpendicular(A,g)
```

```
[6, - 5, - 32]
```

```
>$getLineEquation(h,x,y)
```

$$6\,x-5\,y=-32$$

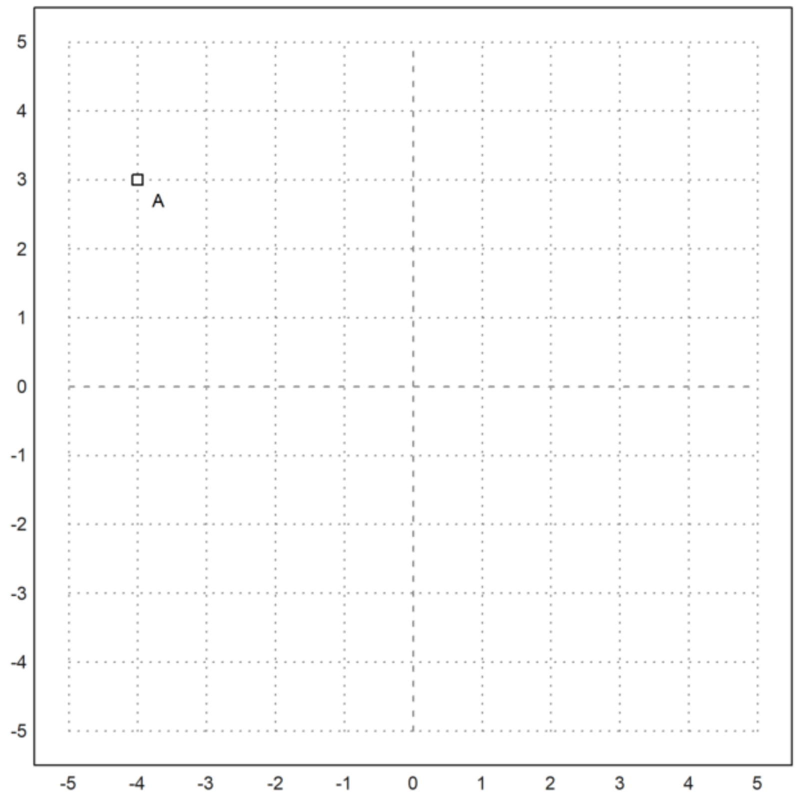
```
>$solve(%,y) | expand
```

$$\left[y=\frac{6\,x}{5}+\frac{32}{5}\right]$$

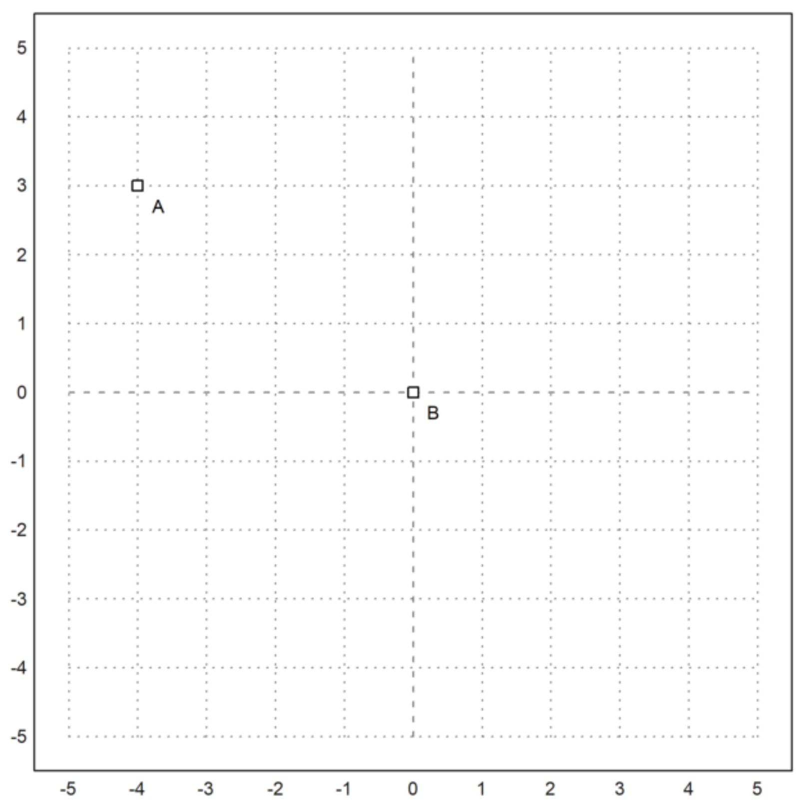
2. Tentukan tinggi segitiga ABC di titik B dengan

$$A(-4,3),B(0,0),C(2,2)$$

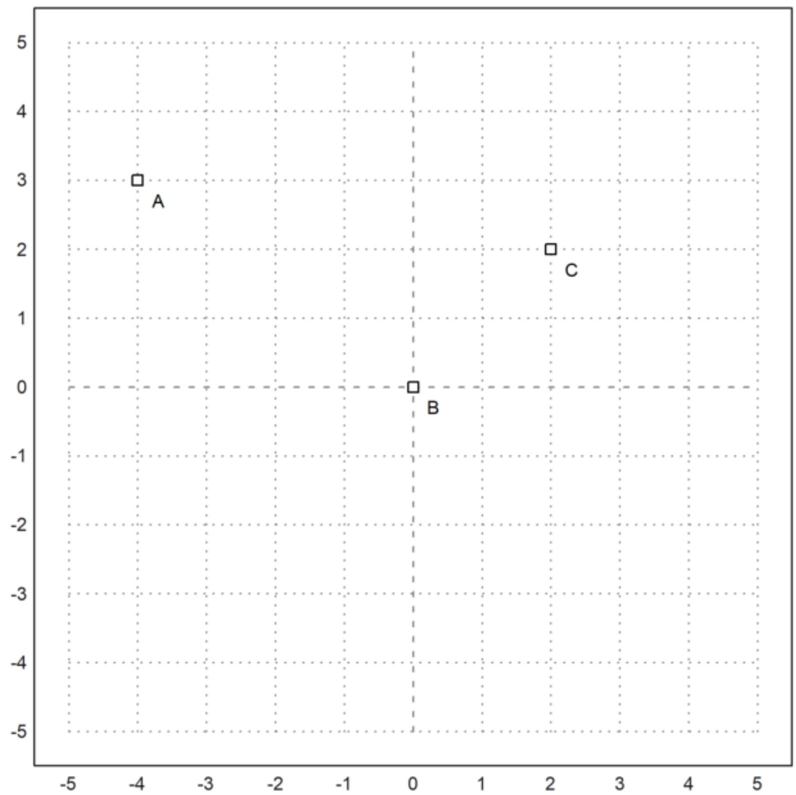
```
>setPlotRange(5);
>A=[-4,3]; plotPoint(A,"A"):
```



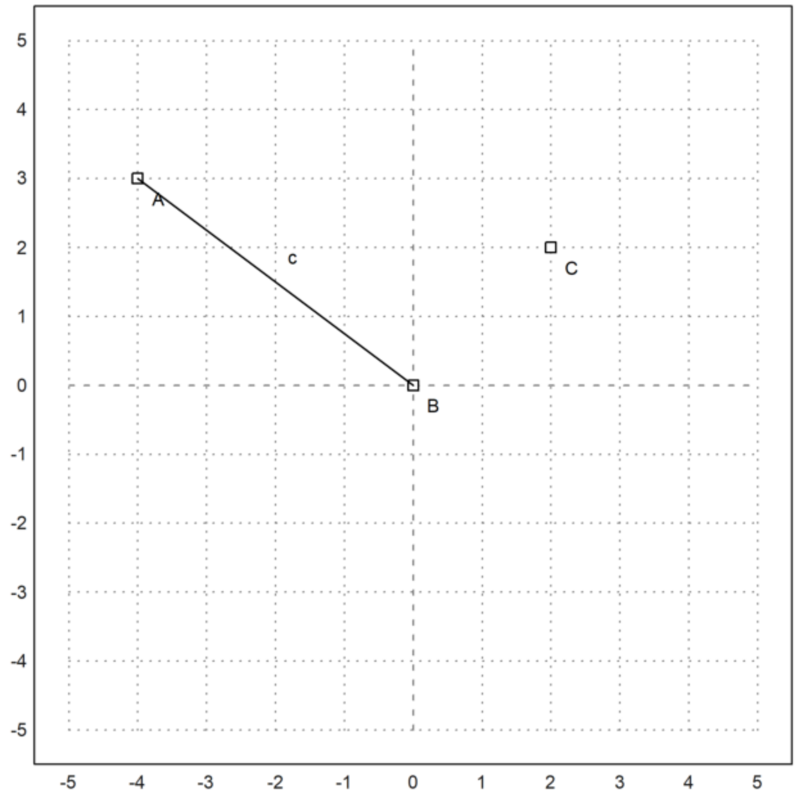

```
>B=[0,0]; plotPoint(B,"B"):
```



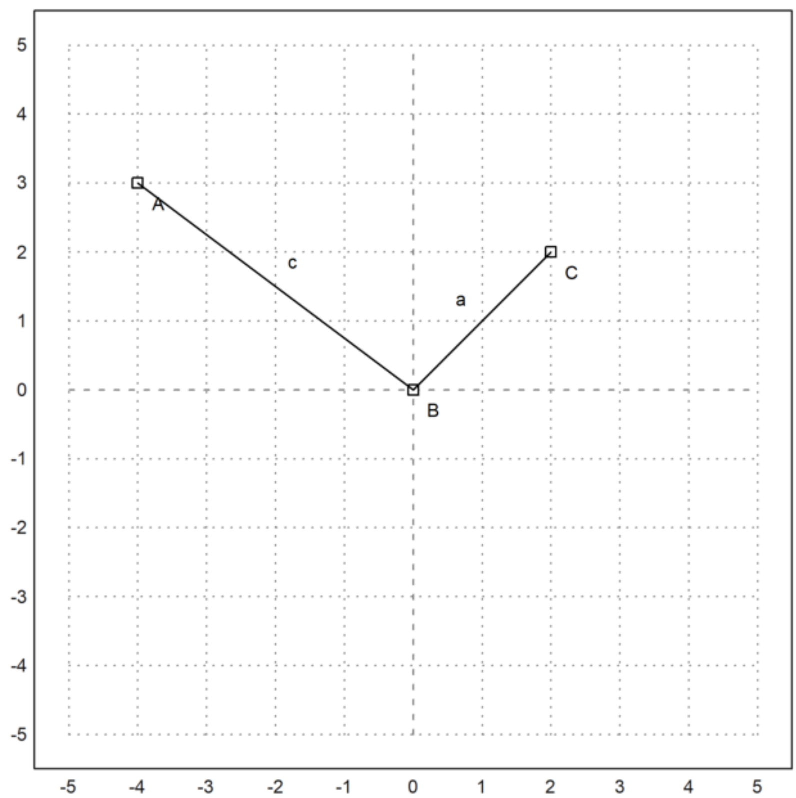
```
>C=[2,2]; plotPoint(C,"C"):
```



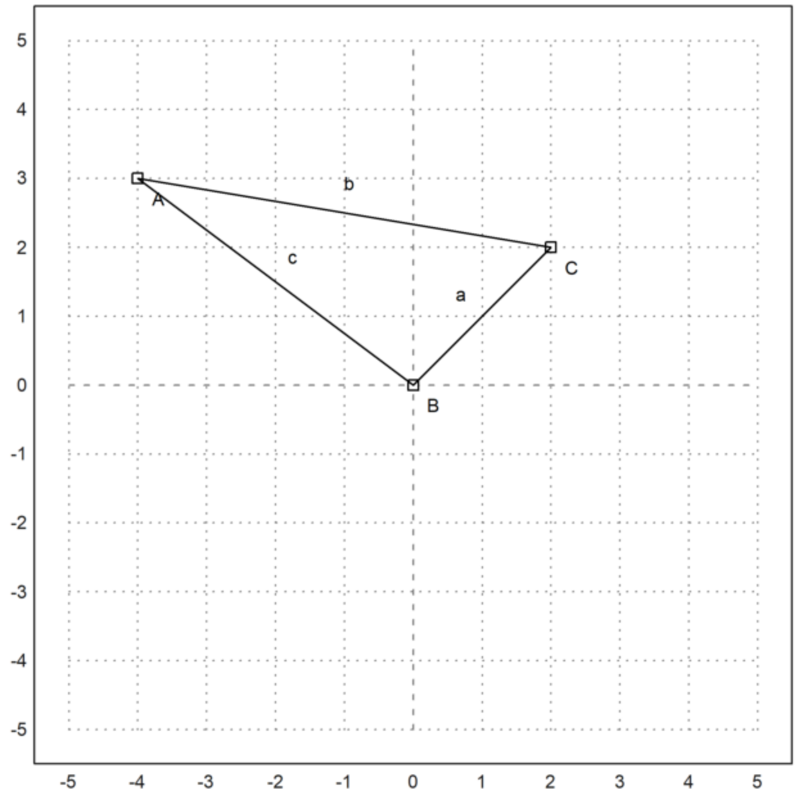

```
>plotSegment(A,B,"c"): // c=AB
```



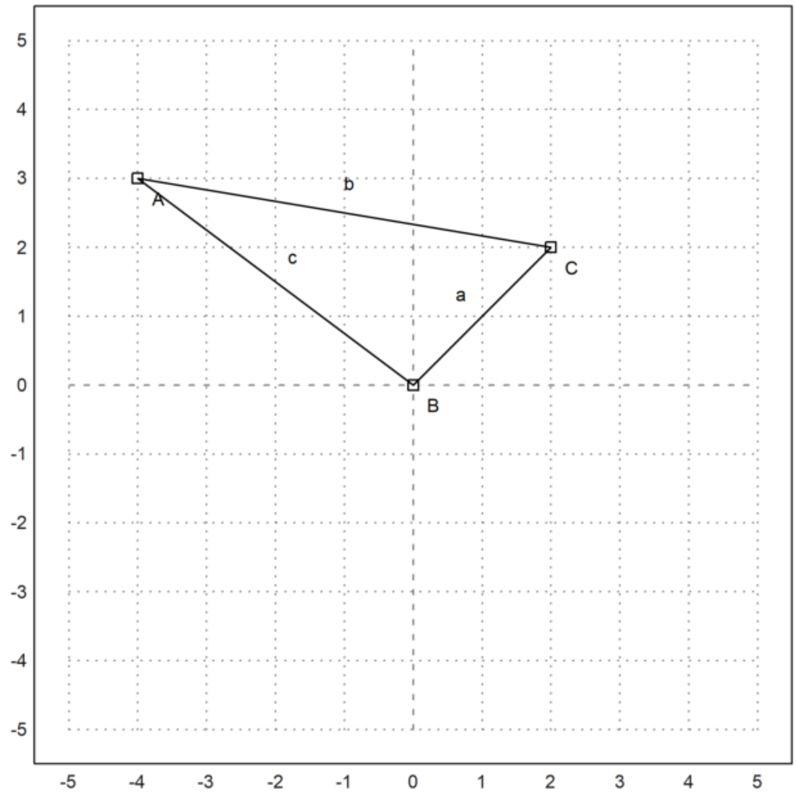

```
>plotSegment(B,C,"a"): // a=BC
```



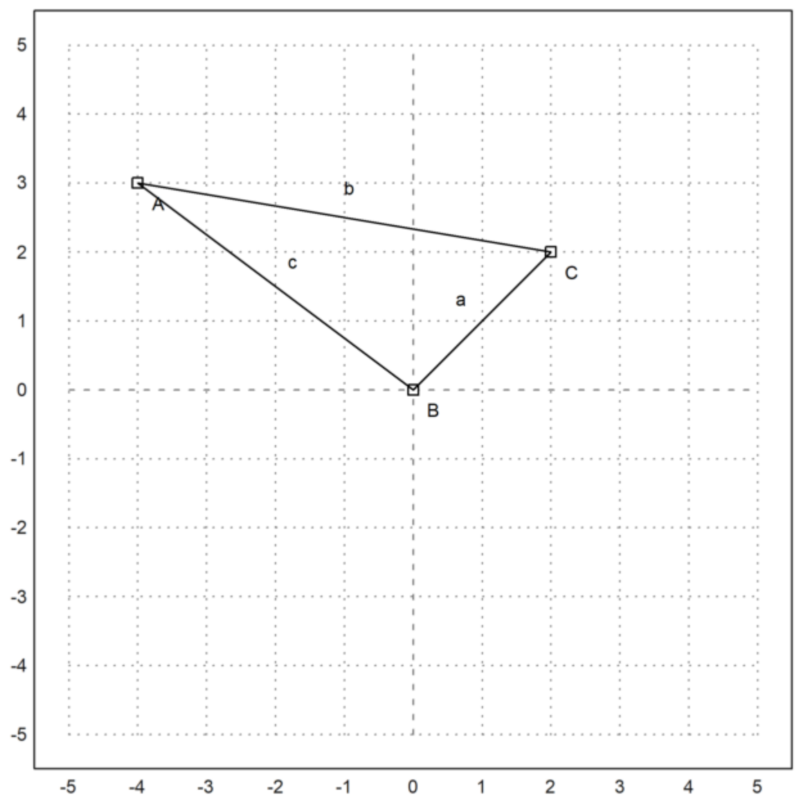
```
>plotSegment(A,C,"b"): // b=AC
```



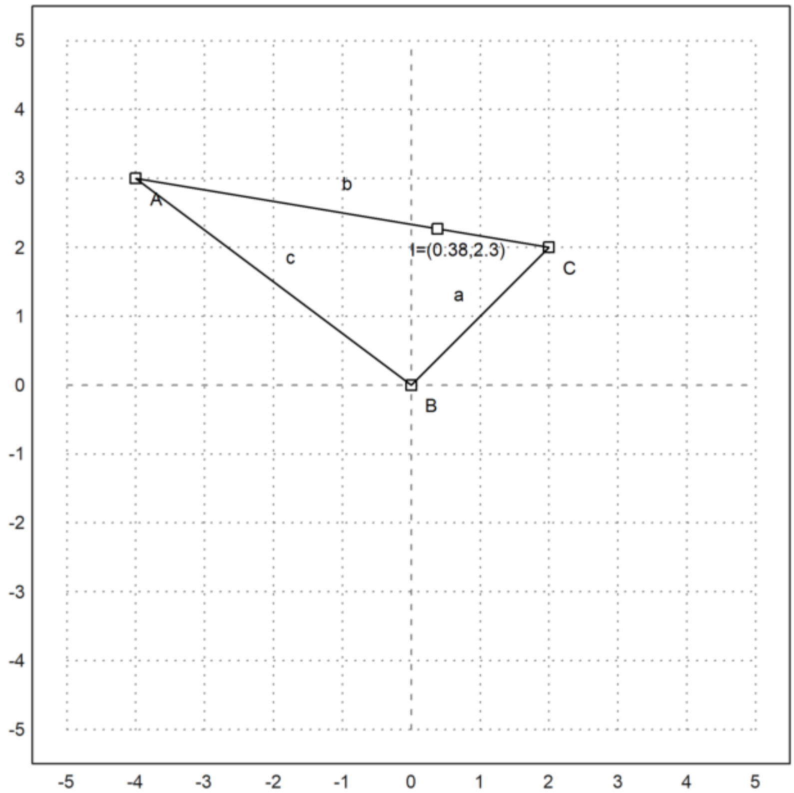

```
>g=lineThrough(A,C):
```



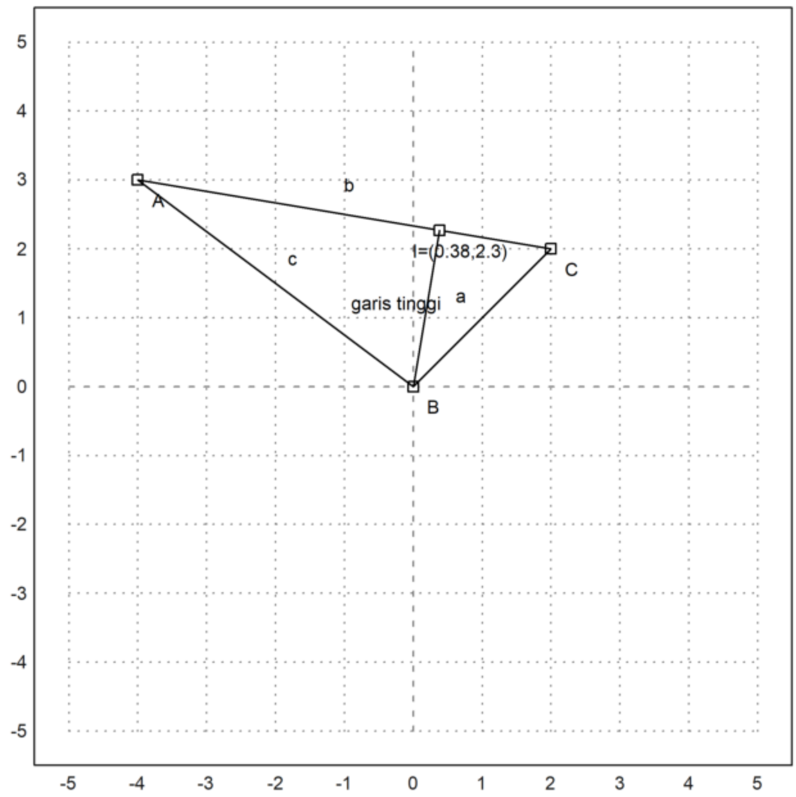

```
>h=perpendicular(B,g):
```



```
>I = lineIntersection(g,h); plotPoint(I,value=1):
```




```
>plotSegment(B,I,"garis tinggi"):
```




```
>A&=[-4,3]
```

[- 4, 3]

```
>B&=[0,0]
```

[0, 0]

```
>C&=[2,2]
```

[2, 2]

```
>g&=lineThrough(A,C)
```

[1, 6, 14]

```
>h&=perpendicular(B,g)
```

```
[6, - 1, 0]
```

```
>$getLineEquation(h,x,y)
```

$$6\,x-y=0$$

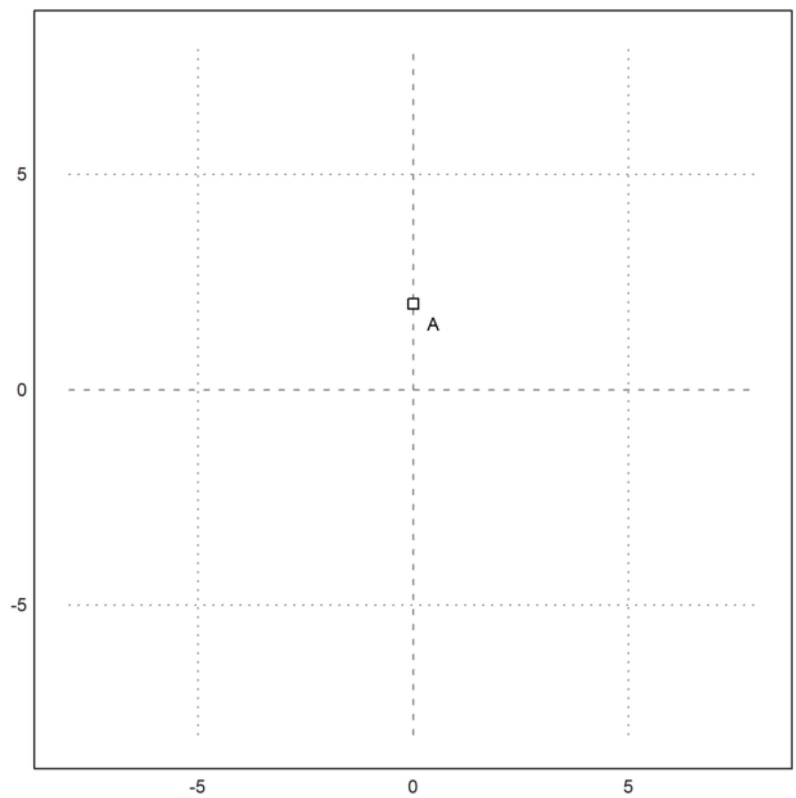
```
>$solve(%,y) | expand
```

$$[y=6\,x]$$

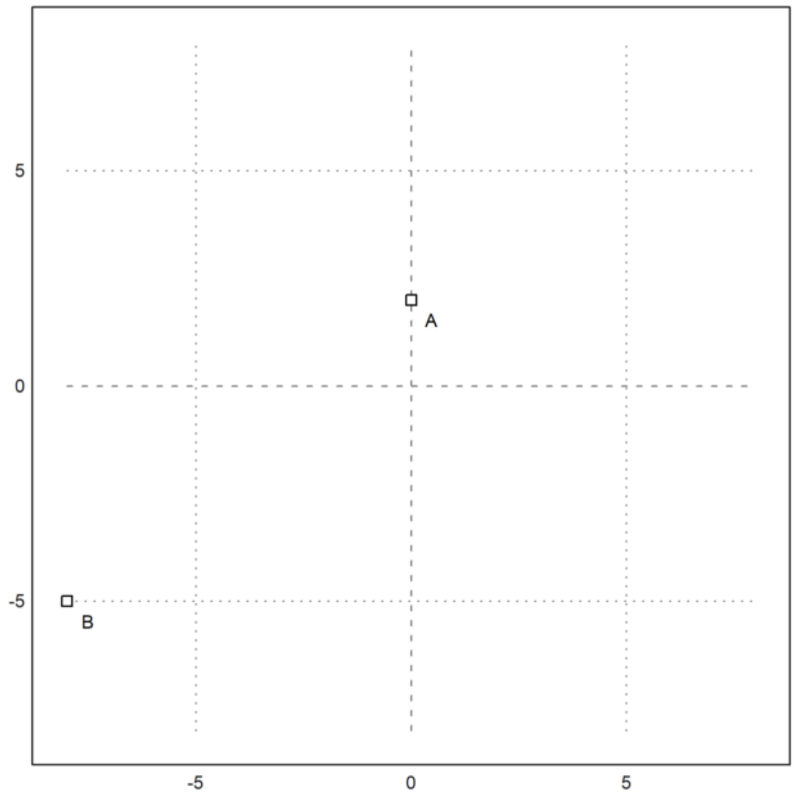
3. Tentukan tinggi segitiga dengan

$$A(0,2),B(-8,-5),C(6,-7)$$

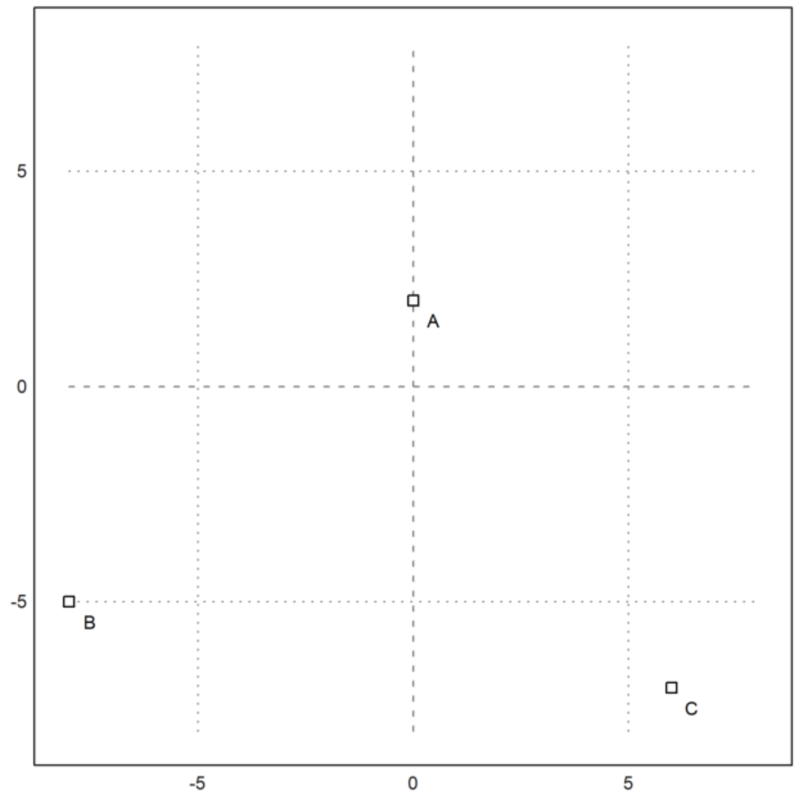
```
>setPlotRange(8);
>A=[0,2]; plotPoint(A,"A"):
```



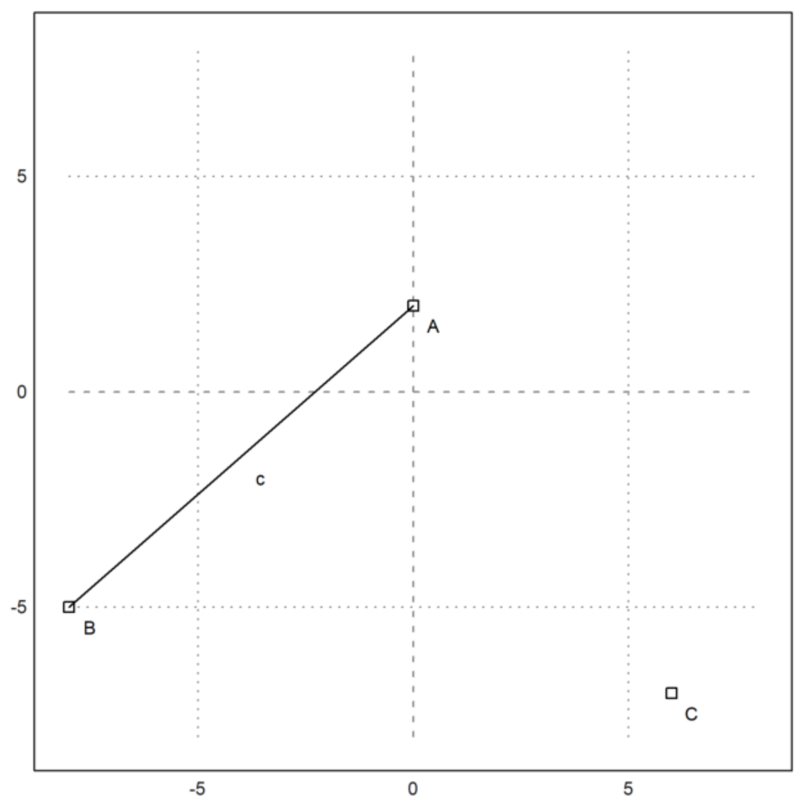
```
>B=[-8,-5]; plotPoint(B,"B"):
```



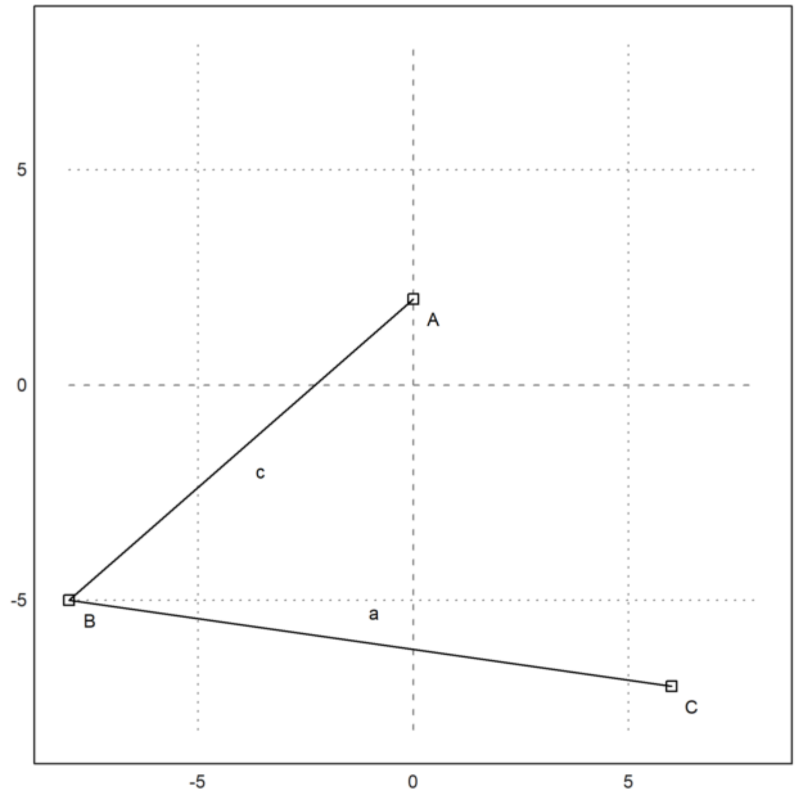

```
>C=[6,-7]; plotPoint(C,"C"):
```



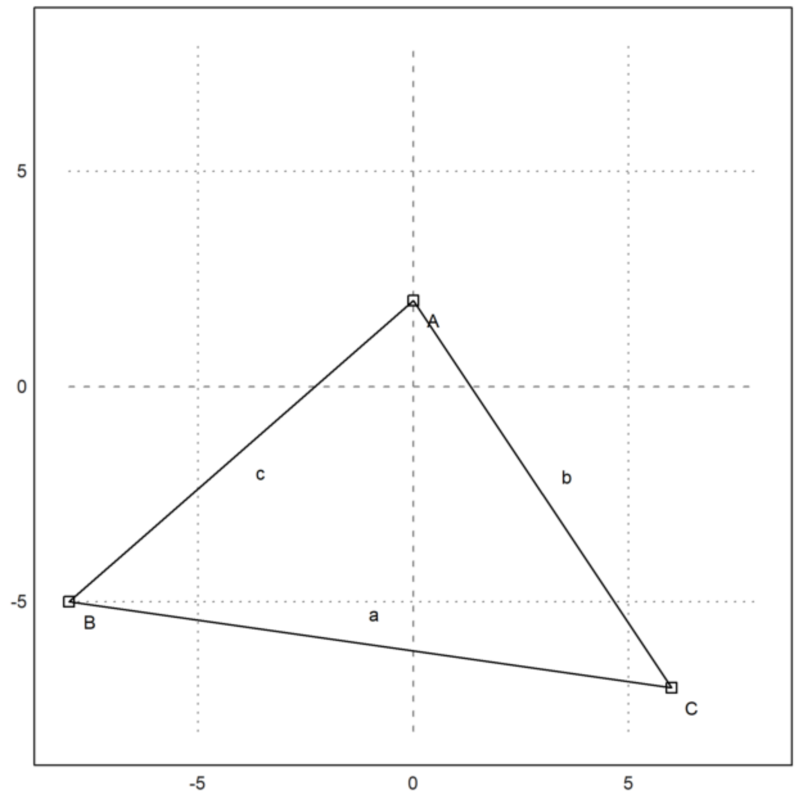

```
>plotSegment(A,B,"c"): // c=AB
```



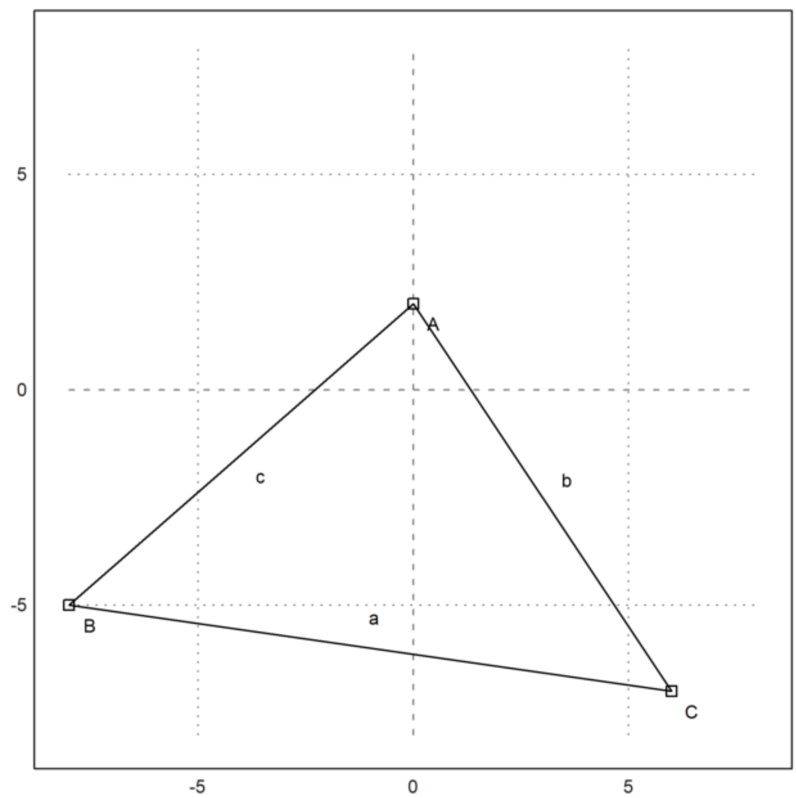
```
>plotSegment(B,C,"a"): // a=BC
```



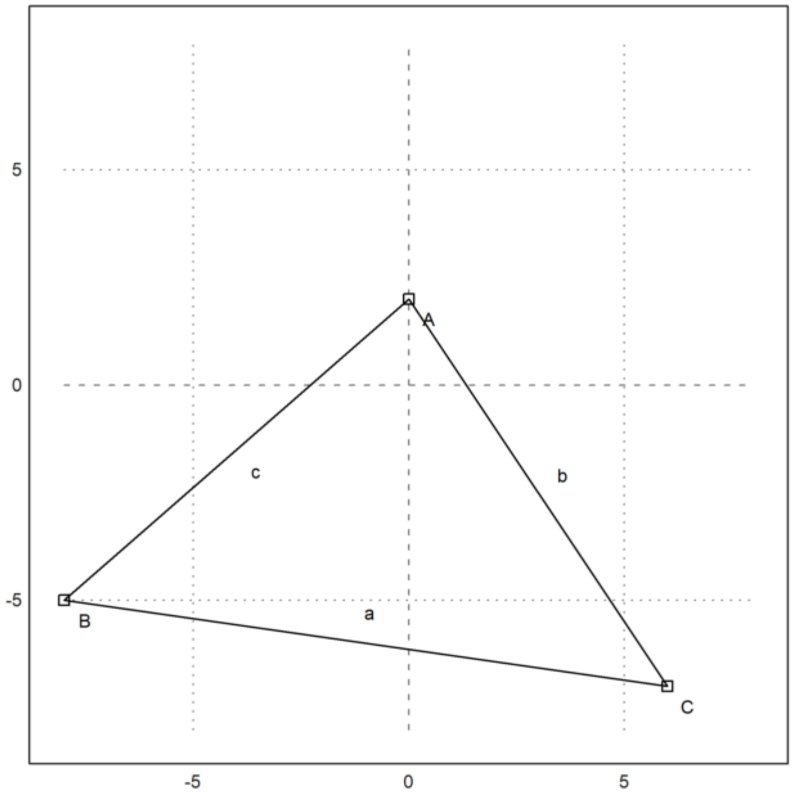

```
>plotSegment(A,C,"b"): // b=AC
```



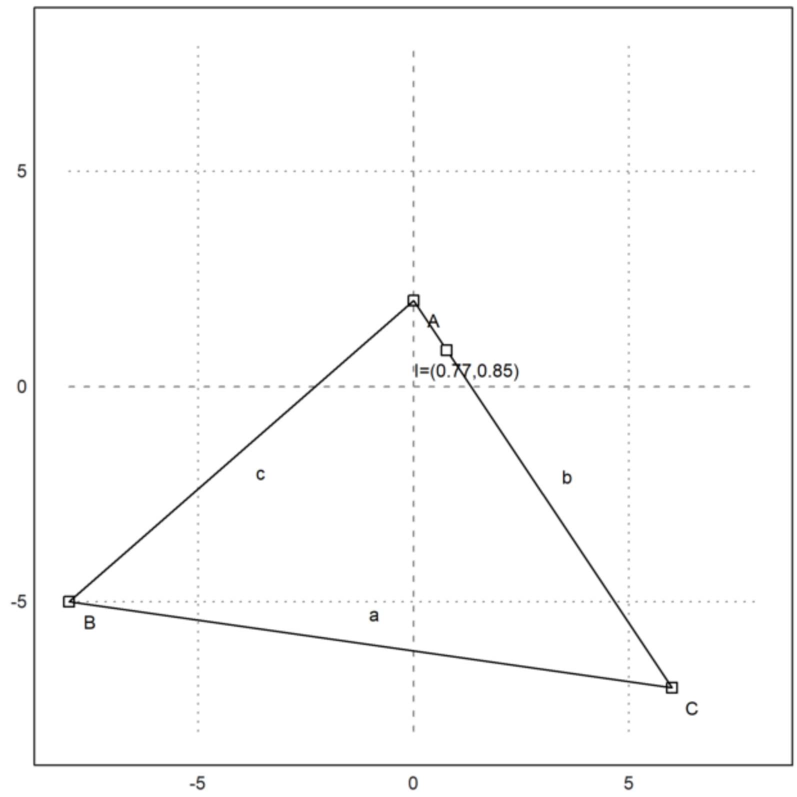

```
>g=lineThrough(A,C):
```



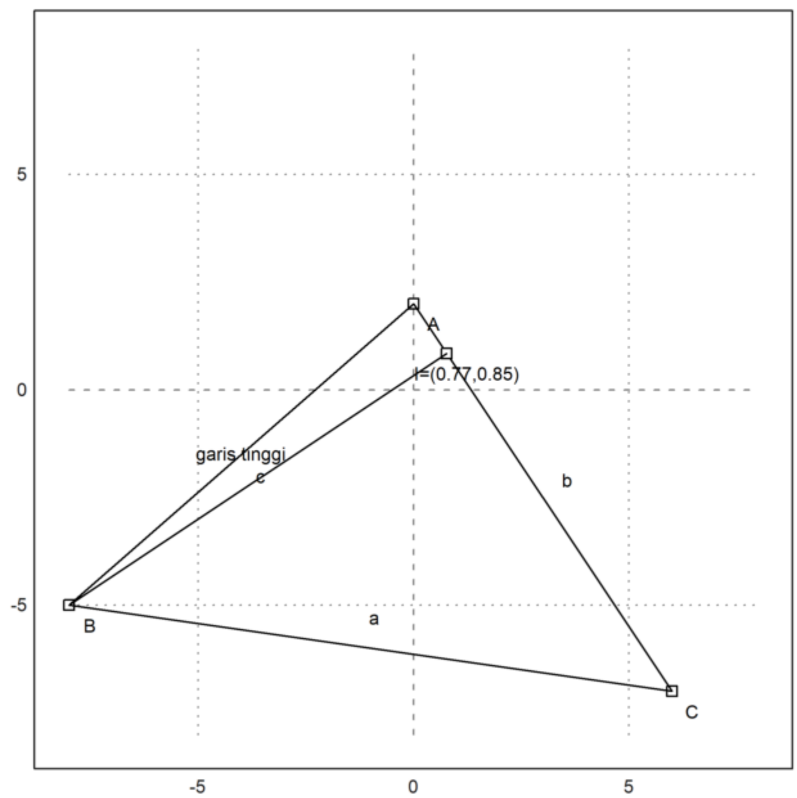
```
>h=perpendicular(B,g):
```




```
>I = lineIntersection(g,h); plotPoint(I,value=1):
```




```
>plotSegment(B,I,"garis tinggi"):
```



```
>A&=[0,2]
```

[0, 2]

```
>B&=[-8,-5]
```

[- 8, - 5]

```
>C&=[6,-7]
```

[6, - 7]

```
>g&=lineThrough(A,C)
```

[9, 6, 12]

```
>h&=perpendicular(B,g)
```

[6, - 9, - 3]

```
>$getLineEquation(h,x,y)
```

$$6\,x-9\,y=-3$$

```
>$solve(%,y) | expand
```

$$\left[y=\frac{2\,x}{3}+\frac{1}{3}\right]$$

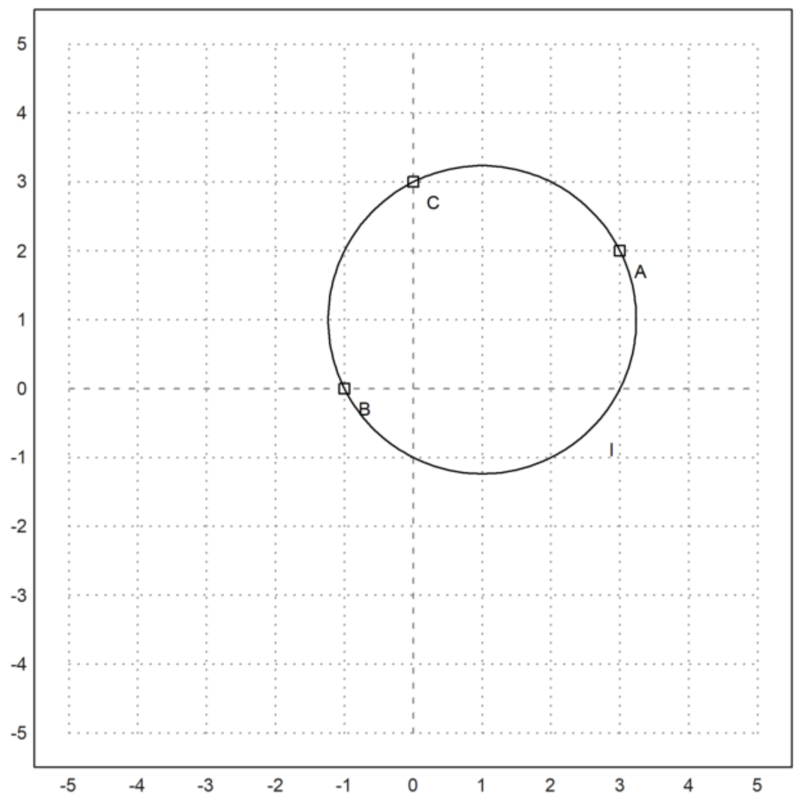
Materi 6 Menentukan titik pusat dan jari-jari suatu lingkaran

Lingkaran adalah himpunan titik-titik yang berjarak sama terhadap suatu titik tertentu. Jarak tersebut dinamakan jari-jari lingkaran, sedangkan titik tertentu tersebut dinamakan pusat lingkaran. Dengan kata lain:
a. Titik pusat lingkaran adalah titik yang berada di tengah lingkaran
b. Jari-jari lingkaran adalah ruas garis yang menghubungkan titik pusat lingkaran dan titik pada lingkaran

```
>load geometry
```

```
Numerical and symbolic geometry.
```

```
>setPlotRange(5);
>A=[3,2]; plotPoint(A,"A");
>B=[-1,0]; plotPoint(B,"B");
>C=[0,3]; plotPoint(C,"C");
>l=circleThrough(A,B,C); // membuat lingkaran yang melalui titik A,B,C
>plotCircle(l):
```



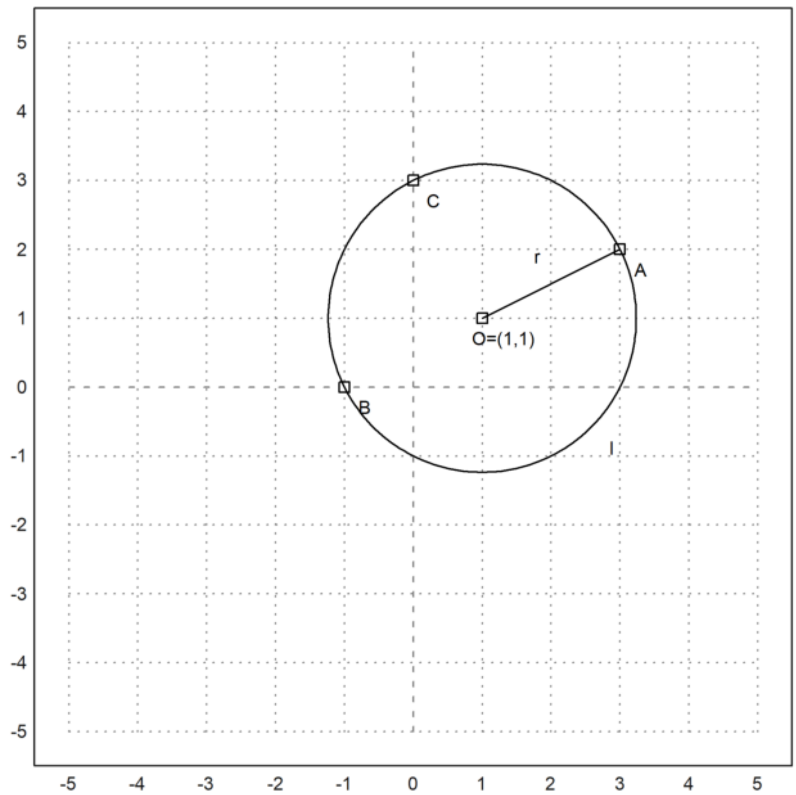

```
>O=getCircleCenter(l); O // menentukan titik pusat lingkaran
```

```
[1,  1]
```

```
>r=getCircleRadius(l); r // menentukan jari-jari lingkaran
```

```
2.2360679775
```

```
>plotPoint(O, value=1); plotSegment(O,A,"r"):
```



Untuk memuktikan hasil tersebut benar dapat kita buktikan sebagai berikut:
1. Menentukan persamaan lingkaran yang melalui titik tersebut.
Persamaan lingkaran adalah

$$x^2 + y^2 + Ax + By + C = 0$$

Substitusi titik ke persamaan tersebut
A(3,2):

$$3^2 + 2^2 + A(3) + B(2) + C = 0$$

$$9 + 4 + 3A + 2B + C = 0$$

$$3A + 2B + C = -13.....(1)$$

B(-1,0):

$$(-1)^2 + (0)^2 + A(-1) + B(0) + C = 0$$

$$1 + 0 - A + C = 0$$

$$-A + C = -1.....(2)$$

C(0,3):

$$(0)^2 + (3)^2 + A(0) + B(3) + C = 0$$

$$0 + 9 + 3B + C = 0$$

$$3B + C = -9.....(3)$$

Eliminasi C dari persamaan (1) dan (2)

$$3A + 2B + C = -13$$

$$-A + C = -1$$

diperoleh

$$4A + 2B = -12....(4)$$

Eliminasi C dari persamaan (1) dan (3)

$$3A + 2B + C = -13$$

$$3B + C = -9$$

$$3A - B = -4.....(5)$$

Eliminasi B dari persamaan (4) dan (5)

$$4A + 2B = -12$$

$$3A - B = -4$$

diperoleh

$$10A = -20$$

$$A = -2$$

Substiusi A=-2 ke persamaan (2)

$$-(-2) + C = -1$$

$$C = -3$$

Substitusi C=-3 ke persamaan (3)

$$3B + (-3) = -9$$

$$B = -2$$

Jadi diperoleh persamaan lingkaran yang melalui 3 titik tersebut yaitu

$$x^2 + y^2 - 2x - 2y - 3 = 0$$

2. Menghitung titik pusat dengan rumus:

$$(\frac{-A}{2}, \frac{-B}{2})$$

$$(\frac{-(-2)}{2}, \frac{-(-2)}{2})$$

$$(1, 1)$$

3. Menghitung jari-jari dengan rumus

$$r = \sqrt{\frac{1}{4}A^2 + \frac{1}{4}B^2 - C}$$

$$r = \sqrt{\frac{1}{4}(-2)^2 + \frac{1}{4}(-2)^2 - (-3)}$$

$$r = \sqrt{\frac{4}{4} + \frac{4}{4} + 3}$$

$$r = \sqrt{5}$$

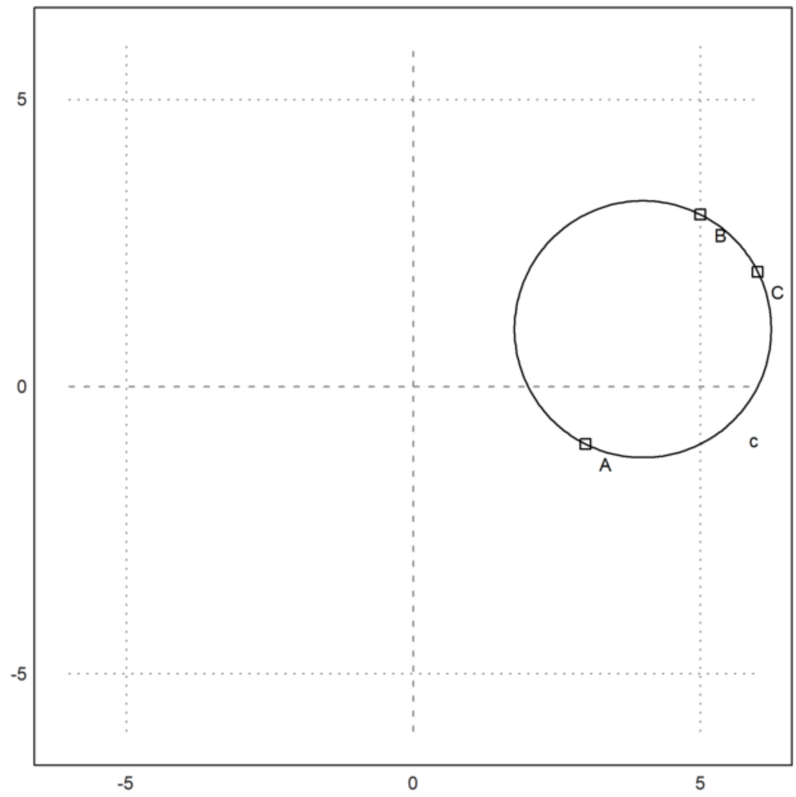
```
>sqrt(5)
```

```
2.2360679775
```

Jadi terbukti bahwa fungsi tersebut benar **Latihan**

Diketahui sebuah lingkaran melalui tiga titik dengan koordinat (3, –1), (5, 3), dan (6, 2). Tentukan pusat lingkaran, dan jari-jari lingkaran!

```
>A=[3,-1]; B=[5,3]; C=[6,2]; //mendefinsikan titik
>setPlotRange(6);
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
>c=circleThrough(A,B,C);
>plotCircle(c):
```




```
>O=getCircleCenter(c); O
```

    [4, 1]

```
>r=getCircleRadius(c); r
```

    2.2360679775

Jadi, titik pusat lingkaran tersebut terletak pada (4,1) dan panjang jari jari lingkaran tersebut adalah 2,236067 atau akar 5

Tentukan titik pusat dan jari-jari lingkaran yang melalui tiga titik berikut ini O(0, 0), P(-2, 4), dan Q(-1, 7)

```
>O&=[0,0]; P&=[-2,4]; Q&=[-1,7];
>c&=circleThrough(O,P,Q);
>O&=getCircleCenter(c); $O
```

$$[3, 4]$$

```
>r&=getCircleRadius(c); $r
```

$$5$$

Jadi, titik pusat lingkaran tersebut terletak pada titik (3,4) dan panjang jari-jari lingkara tersebut adalah 5

subtopik 7

Menentuk

Persamaan lingkaran yang melalui tiga titik

Untuk menentukan persamaan lingkaran melalui 3 titik, dapat digunakan persamaan lingkaran

$$(x-a)^2+(y-b)^2=r^2$$

dimana (a,b) adalah pusat lingkaran dan r jari-jari lingkaran
atau bentuk umum persamaan lingkaran

$$x^2+y^2+Ax+By+C=0.$$

```
> load geometry
```

```
Numerical and symbolic geometry.
```

```
>A &= [3,-1]; B &= [5,3]; C &= [6,2]; // menentukan tiga titik A, B, C
>c &= circleThrough(A,B,C);
>$getCircleEquation(c,x,y), //mencari persamaan lingkaran
```

$$(y-1)^2+(x-4)^2=5$$

```
>$expand(%)
```

$$y^2-2\,y+x^2-8\,x+17=5$$

Jadi, didaptkan persamaan lingkaran yang melalui 3 titik tersebut adalah

$$x^2+y^2-8x-2y+12=0$$

Untuk membuktikan hasil tersebut persamaan lingkaran melalui 3 titik dapat dicari dengan cara:
1. Memisalkan bentuk umum persamaan lingkaran, yaitu

$$x^2+y^2+Ax+By+C=0$$

2. Substitusi ketiga titik koordinat pada pemisalan bentuk umum persamaan lingkaran pada langkah pertama
3. Akan diperoleh tiga persamaan dengan tiga variabel
4. Tentukan nilai ketiga variabel (A, B daan C)
5. Substitusikan nilia variabel yang sudah diperoleh ke bentuk umum persamaan lingkaran
6. Diperoleh bentuk umum persamaan lingkaran

Sehingga contoh tersebut dapat diselesaikan sebagai berikut:
1. Substitusi titik (3,-1) pada bentuk umum persamaan lingkaran.

$$x^2 + y^2 + Ax + By + C = 0$$

$$3^2 + (-1)^2 + A(3) + B(-1) + C = 0$$

$$3A - B + C = -10.....(1)$$

2. Substitusi titik (5,3) pada bentuk umum persamaan lingkaran.

$$x^2 + y^2 + Ax + By + c = 0$$

$$5^2 + (3)^2 + A(5) + B(3) + C = 0$$

$$5A + 3B + C = -34.....(2)$$

3. Substitusi titik (6,2) pada bentuk umum persamaan lingkaran.

$$x^2 + y^2 + Ax + By + C = 0$$

$$6^2 + (2)^2 + A(6) + B(2) + C = 0$$

$$6A + 2B + C = -40.....(3)$$

4. Eliminasi C dari persamaan (1) dan (2)

$$3A - B + C = -10$$

$$5A + 3B + C = -34$$

diperoleh

$$-2A - 4B = 24$$

$$A + 2B = -12.....(4)$$

5. Eliminasi C dari pesamaan (2) dan (3)

$$5A + 3B + C = -34$$

$$6A + 2B + C = -40$$

diperoleh

$$-A + B = 6$$

$$A - B = -6.....(5)$$

6. Cari nilai A dan B dengan mengeliminasi persamaan (4) dan (5)

$$A + 2B = -12$$

$$A - B = -6$$

diperoleh

$$3B = -6$$

$$B = -2$$

sehingga,

$$A - (-2) = -6$$

$$A = -8$$

7. Substitusi niali a dan b ke persamaan 1

$$3(-8) - (-2) + C = -10$$

$$C = 12$$

Diperoleh persamaan lingkaran:

$$x^2 + y^2 + Ax + By + c = 0$$

$$x^2 + y^2 - 8x - 2y + 12 = 0$$

Karena hasilnya sama, maka terbukti
Mencari persamaan lingkaran dengan mencari titik pusat dan jari-jarinya terlebih dahulu

```
>c &= circleThrough(A,B,C); $c  // (titik pusat dan jari-jari) lingkaran yang melalui A, B
```

$$\left[4, 1, \sqrt{5}\right]$$

Persaman umum lingkaran adalah

$$(x - a)^2 + (y - b)^2 = r^2$$

dimana (a,b) adalah pusat lingkaran dan r jari-jari lingkaran
Maka diperoleh persmaan lingkaran yang melalui 3 titik tersebut adalah:

$$(x - 4)^2 + (y - 1)^2 = (\sqrt{5})^2$$

$$(x - 4)^2 + (y - 1)^2 = 5$$

latihan

Carilah persamaan lingkaran yang melalui titik A (0,2), B (3,3) dan C (6,2) !

```
>A &= [0,2]; B &= [3,3]; C &= [6,2]; // menentukan tiga titik A, B, C
>c &= circleThrough(A,B,C);
>$getCircleEquation(c,x,y), //mencari persamaan lingkaran
```

$$(y+2)^2+(x-3)^2=25$$

Tentukan persamaan lingkaran yang melalui titik (1,1), (2,-1), dan (3,2)!

```
>A &= [1,1]; B &= [2,-1]; C &= [3,2]; // menentukan tiga titik A, B, C
>c &= circleThrough(A,B,C);
>$getCircleEquation(c,x,y), //mencari persamaan lingkaran
```

$$\left(y-\frac{1}{2}\right)^2+\left(x-\frac{5}{2}\right)^2=\frac{5}{2}$$

Materi 9

Menentukan persamaan dan menggambar parabola yang ditentukan titik fokus dan garis arahnya

Parabola adalah kedudukan titik-titik yang jaraknya ke suatu ttik tertentu(titik api/titik focus)dan jaraknya ke suatu garis tertentu (direktriks/garis arah) adalah sama.
Untuk menemukan persamaan parabola yang ditentukan titik fokus dan garis arahnya akan diilustrasikan gambar dari kurva parabola.Misal titik fokus(p,0), titik puncak O(0,0), garis direktriks(garis arah) yaitu garis g kemudian pilih titik R(-p,y) pada garis y, kemudian pilih sembarang titik P(x,y) yang terdapat dalam parabola.
image: Parabola1.JPG
sesuai definisi parabola,jarak titik P ke titik fokus(|PF|) sama dengan jarak titik P ke titik R(|PR|)

$$|PF|=|PR|$$

$$\sqrt{(x-p)^2+(y-0)^2}=\sqrt{(x-(-p))^2+(y-y)^2}$$

$$\sqrt{(x-p)^2+y^2}=\sqrt{(x-p)^2+(0)^2}$$

$$\sqrt{(x-p)^2+y^2}=\sqrt{(x-p)^2}$$

$$(x-p)^2+y^2=(x-p)^2$$

$$x^2-2px+p^2+y^2=x^2+2px+p^2$$

$$y^2=2px+2px$$

$$y^2=4px$$

sehingga persamaan parabola adalah y^2 = 4px dengan titik puncak O(0,0) dengan titik fokus F(p,0) akan mempresentasikan parabola terbuka ke kanan (arah sumbu x positif).

```
>load geometry
```

```
Numerical and symbolic geometry.
```

```
>A&=[-1,-1]; B&=[2,0]; C&=[1,2];
>A=[-1,1]; B=[2,0]; C=[1,2];
>r&=(lineThrough(A,B)); $r // garis yang melalui A dan B
```

$$[-1, 3, -2]$$

```
>$getLineEquation(r,x,y); $solve(%,y) // persamaan garis r yang dinyatakan dalam x dan y
```

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

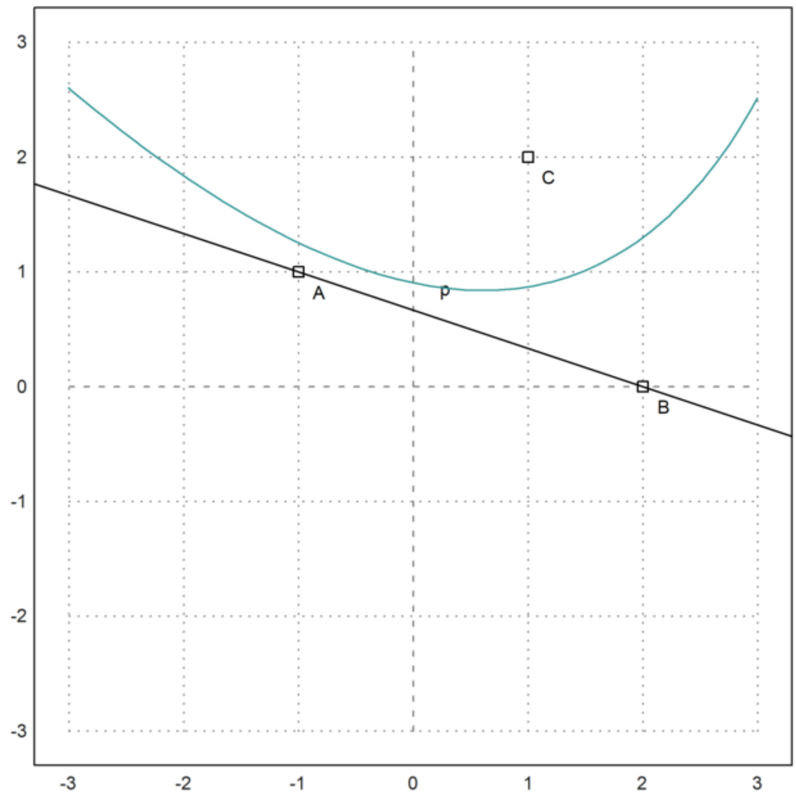
Menentukan persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3\,y-x+2}{\sqrt{10}} - \sqrt{(2-y)^2 + (1-x)^2} = 0$$

Selanjutnya kita akan menggambar persamaan tersebut

```
>setPlotRange(3); // mendefinisikan bidang koordinat
>A=[-1,1]; B=[2,0]; C=[1,2];
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); // Memplot titik-titik koordinat
>plotLine(lineThrough(A,B));
>plot2d(p,level=0,add=1,contourcolor=5): // membuat plot grafik dua dimensi dari data yang
```

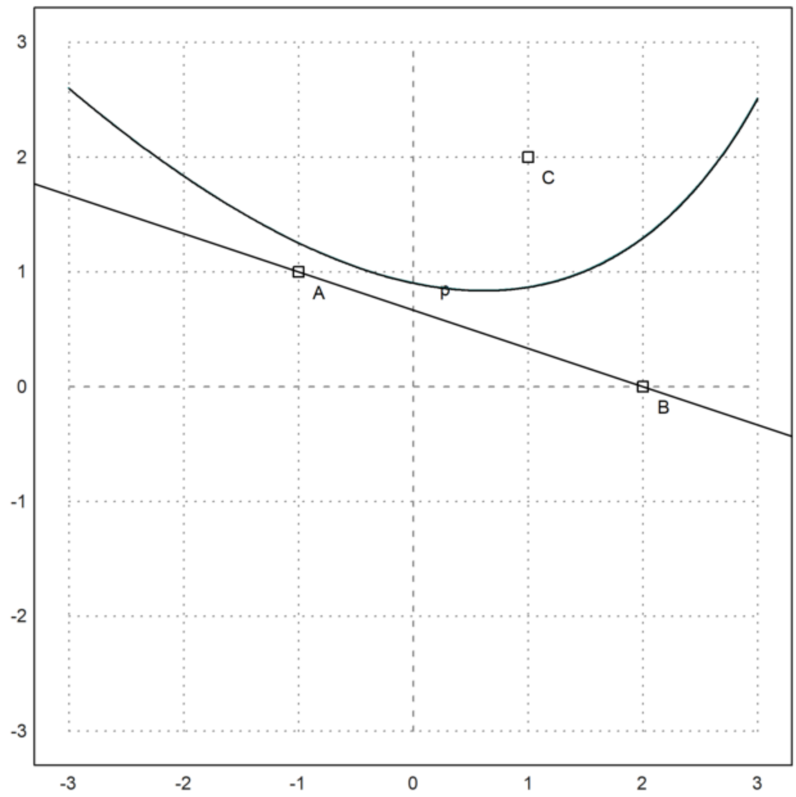



```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

```
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26,
                      y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 26]
```

Solusinya adalah
maxima: akar[1]
maxima: akar[2]

```
>plot2d(&rhs(akar[1]),add=1):
```




```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan kurva di atas
```

$$g\left(x\right)=-3\,x-\sqrt{70}\,\sqrt{9-2\,x}+26$$

Untuk membuktikan bahwa persamaan parabola benar, kita dapat mengambil sebarang titik pada kurva tersebut, misal titik T.

```
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
```

$$\sqrt{1503-54\,\sqrt{11}\,\sqrt{70}}$$

$$2.135605779339061$$

```
>U &= projectToLine(T,lineThrough(A,B)); $U // proyeksi T pada garis AB
```

$$\left[\frac{80-3\,\sqrt{11}\,\sqrt{70}}{10},\frac{20-\sqrt{11}\,\sqrt{70}}{10}\right]$$

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jatak T ke AB
```

$$\sqrt{1503-54\,\sqrt{11}\,\sqrt{70}}$$

$$2.135605779339061$$

Dari pembuktian diatas TERBUKTI jarak T ke C sama dengan jarak T ke AB

Latihan 1

Jika terdapat garis melalui titik A(4,3) dan B(2,-2) dengan titik fokus C(5,2), tentukan persamaan parabola dan parabola yang terbentuk

```
>A&=[4,3]; B&=[2,-2]; C&=[5,2];
>r&=(lineThrough(A,B)); $r
```

$$[5,-2,14]$$

```
>$getLineEquation(r,x,y); $solve(%,y) // persamaan garis r dinyatakan dalam x dan y
```

$$\left[y=\frac{5\,x-14}{2}\right]$$

Menentukan 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
```

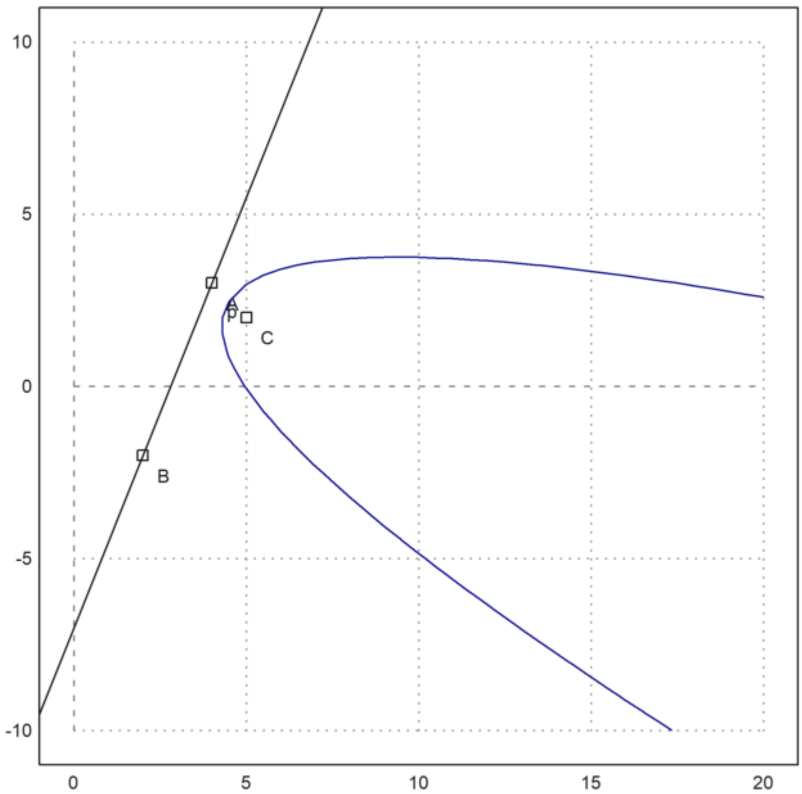
$$\frac{-2\,y+5\,x-14}{\sqrt{29}}-\sqrt{\left(2-y\right)^2+\left(5-x\right)^2}=0$$

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

```
          - sqrt(203) sqrt(10 x - 43) - 10 x + 86
     [y = ---------------------------------------,
                            25
                    sqrt(203) sqrt(10 x - 43) - 10 x + 86
                y = -------------------------------------]
                                     25
```

solusinya adalah
maxima: akar[1]
maxima: akar[2]
Selanjutnya kita akan menggambar persamaan tersebut

```
>setPlotRange(0,20,-10,10); // mendefinisikan bidang koordinat
>A=[4,3]; B=[2,-2]; C=[5,2];
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); // memplot titik-titik koordinat
>plotLine(lineThrough(A,B));
>plot2d(p,level=0,add=1,contourcolor=4):
```



Latihan 2

Jika terdapat garis melalui titik O(7,0) dan P(-5,3) dengan titik fokus Q(1,-2),tentukan persamaan parabola dan parabola yang terbentuk

```
>O&=[7,0]; P&=[-5,3]; Q&=[1,-2];
>r&=(lineThrough(O,P)); $r
```

$$[-3,-12,-21]$$

```
>$getLineEquation(r,x,y); $solve(%,y) // Persamaan garis r dinyatakan dalam x dan y
```

$$\left[y=\frac{7-x}{4}\right]$$

Menentukan persamaan tempat kedudukan titik-titik yang berjarak sama ke titik Q dan ke garis OP

```
>p &= getHesseForm(lineThrough(O,P),x,y,Q)-distance([x,y],Q); $p='0
```

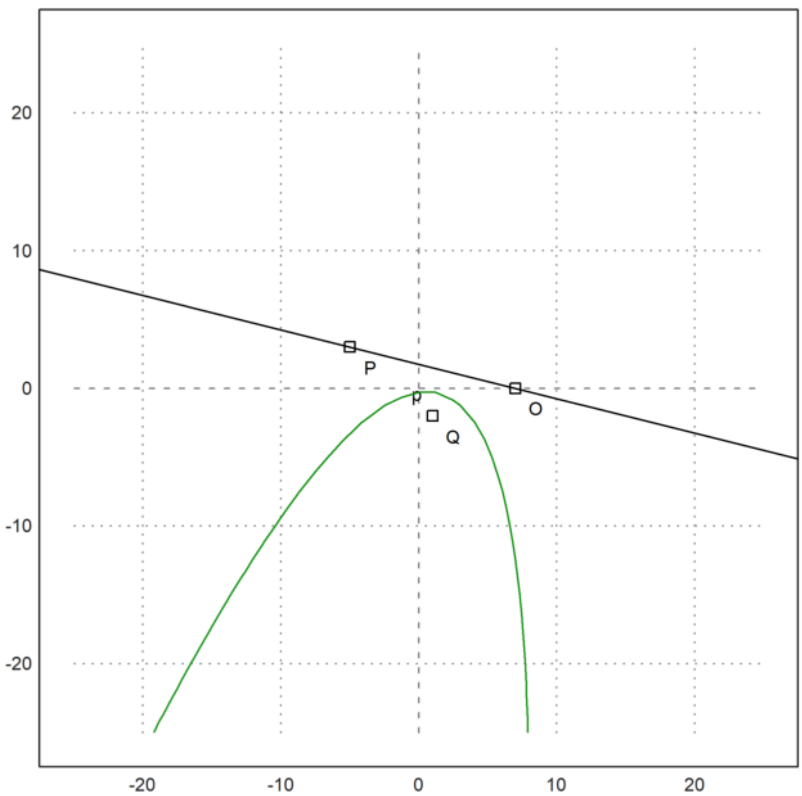
$$\frac{-12\,y-3\,x+21}{3\,\sqrt{17}}-\sqrt{\left(-y-2\right)^2+\left(1-x\right)^2}=0$$

```
>akar &= solve(getHesseForm(lineThrough(O,P),x,y,Q)^2-distance([x,y],Q)^2,y)
```

```
          [y = 4 x - 2 sqrt(119) sqrt(8 - x) - 62,
                                   y = 4 x + 2 sqrt(119) sqrt(8 - x) - 62]
```

Solusinya adalah
maxima: akar[1]
maxima: akar[2]
Selanjutnya kita akan menggambar persamaan tersebut

```
>setPlotRange(25); // mendefinisikan bidang koordinat
>O=[7,0]; P=[-5,3]; Q=[1,-2];
>plotPoint(O,"O"); plotPoint(P,"P"); plotPoint(Q,"Q"); // memplot titik-titik koordinat
>plotLine(lineThrough(O,P));
>plot2d(p,level=0,add=1,contourcolor=3):
```



Materi 10 Menentukan Persamaan dan Menggambar Ellips yang Diketahui Titik Fokusnya

Elips merupakan salah satu bagian dari kerucut yang dihasilkan ketika sebuah bidang memotong kerucut secara miring, tapi tidak sampai memotong alasnya

```
>load geometry
```

```
Numerical and symbolic geometry.
```

Mencari persamaan ellips, dengan cara menghitung luas segitiga dengan panjang sisi a, b, dan c, pertama kita meletakkan titik-titik pada (0,0), (a,0), dan (x,y).

$$x^2 + y^2 = b^2, \quad (x-a)^2 + y^2 = c$$

untuk x dan y

```
>Hasil &= solve([x^2+y^2=b^2,(x-a)^2+y^2=c^2],[x,y])
```

```
[]
```

2.Menjabarkan hasil y

```
>Hasily &= y with Hasil[2][2]
```

```
Maxima said:
part: invalid index of list or matrix.
 -- an error. To debug this try: debugmode(true);

Error in:
Hasily &= y with Hasil[2][2] ...
                            ^
```

3.Mendapatkan Rumus Heron
Rumus Heron pertama kali ditemukan oleh Heron dari Alexandria. Rumus Heron digunakan untuk menentukan luas segitiga jika panjang semua sisinya atau untuk mencari luas segi empat . Rumus mencari luas ini tidak bergantung pada sudut suatu segitiga tetapi bergantung pada panjang semua sisi segitiga.

```
>function F(a,b,c) &= sqrt(factor((Hasily*a/2)^2))
```

```
32 mabs(Hasily)
```

```
>$solve(diff(F(a,b,c)^2,c)=0,c)
```

```
Maxima said:
diff: second argument must be a variable; found
    [5/2,1/2,sqrt(5)/sqrt(2)]
 -- an error. To debug this try: debugmode(true);

Error in:
$solve(diff(F(a,b,c)^2,c)=0,c) ...
                              ^
```

4.Mencari himpunan semua titik di mana b+c=d untuk d adalah konstanta.

```
>p1 &= subst(d-c,b,Hasil[2])// subtitusi kan b=d-c di hasil 2
```

```
Maxima said:
part: invalid index of list or matrix.
 -- an error. To debug this try: debugmode(true);

Error in:
... (d-c,b,Hasil[2])// subtitusi kan b=d-c di hasil 2 ...
                                                      ^
```

5.Membuat fungsi

```
>function fx(a,c,d) &= rhs(p1[1]); $fx(a,c,d)// rhs = ambil hasil di ruas kanan
```

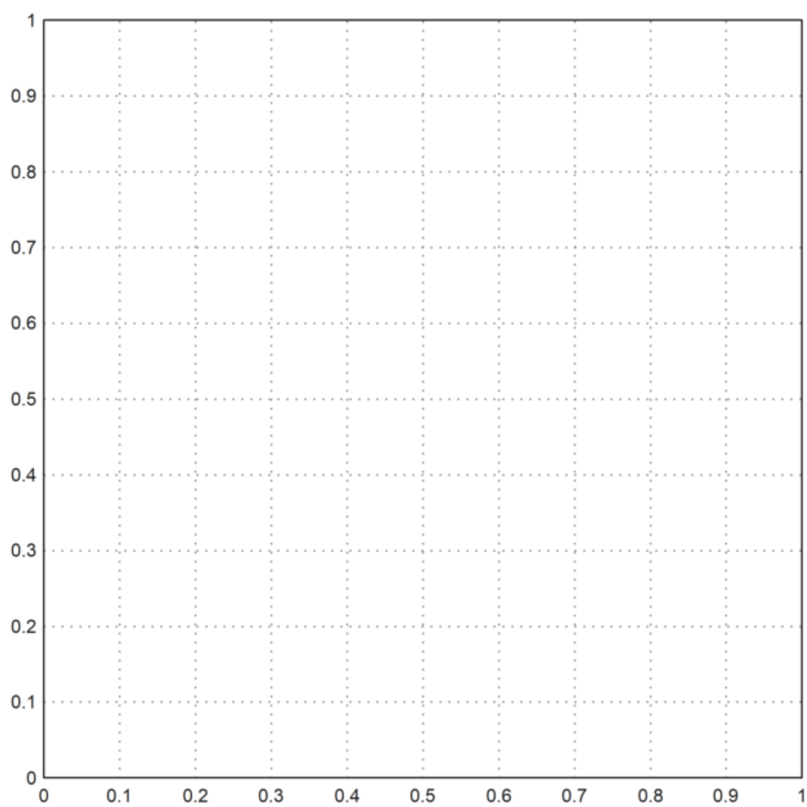
$$0$$

```
>function fy(a,c,d) &= rhs(p1[2]);$fy(a,c,d)// rhs = ambil hasil di ruas kanan
```

$$0$$

6.Menggambar himpunan tersebut. (mendapatkan sebuah ellips)

```
>plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```



```
>$((fx(a,c,d)-a/2)^2/a^2+fy(a,c,d)^2/b^2 with [a=d/2,b=sqrt(d^2-a^2)/2])
```

$$\frac{1}{4}$$

```
>$ratsimp(%)//menyederhanakan
```

$$\frac{1}{4}$$

7.Mendapatkan persamaan umum untuk ellips, yaitu

$$\frac{(x-x_p)^2}{a^2}+\frac{(y-y_p)^2}{b^2}=1,$$

Di mana (xp, yp) adalah titik pusat, dan a dan b adalah setengah dari sumbu-sumbunya.

```
>reset();
```

Cara mendapatkan rumus ellips dengan pembutian secara matematis

Misal kedua titik tetap tersebut F1(-c,0) dan F2(c,0) dan jumlah jarak yang tetap tersebut 2a. Maka untuk sembarang titik P(x,y) pada tepat kedudukan memenuhi:

$$|F_1P|+|F_2P|=2a$$

$$\sqrt{(x-(-c))^2+(y-0)^2}+\sqrt{(x-c)^2+(y-0)^2}=2a$$

$$\sqrt{(x+c)^2+y^2}+\sqrt{(x-c)^2+y^2}=2a$$

$$2a-\sqrt{(x-c)^2+y^2}=\sqrt{(x+c)^2+y^2}$$

$$(2a-\sqrt{(x-c)^2+y^2})^2=(\sqrt{(x+c)^2+y^2})^2$$

$$4a^2-4a\sqrt{(x-c)^2+y^2}+((x-c)^2+y^2)=(x+c)^2+y^2$$

$$4a^2-4a\sqrt{(x-c)^2+y^2}+x^2-2cx+c^2+y^2=x^2+2cx+c^2+y^2$$

$$4a^2-4a\sqrt{(x-c)^2+y^2}-2cx=2cx$$

$$4a^2-4cx=4a\sqrt{(x-c)^2+y^2}$$

$$a^2-cx=a\sqrt{(x-c)^2+y^2}$$

$$a\sqrt{(x-c)^2+y^2}=a^2-cx$$

$$(a\sqrt{(x-c)^2+y^2})^2=(a^2-cx)^2$$

$$a^2((x-c)^2+y^2)=a^4-2ca^2x+c^2x^2$$

$$a^2(x^2-2cx+c^2+y^2)=a^4-2ca^2x+c^2x^2$$

$$a^2x^2-2ca^2x+c^2a^2+a^2y^2=a^4-2ca^2x+c^2x^2$$

$$a^2x^2 - c^2x^2 + a^2y^2 = a^4 + c^2x^2$$

$$a^2x^2 - c^2x^2 + a^2y^2 = a^4 - c^2a^2$$

$$(a^2 - c^2)x^2 + a^2y^2 = a^2(a^2 - c^2)$$

Dengan memisalkan

$$b^2 = a^2 - c^2$$

Diperoleh:

$$b^2x^2 + a^2y^2 = a^2b^2$$

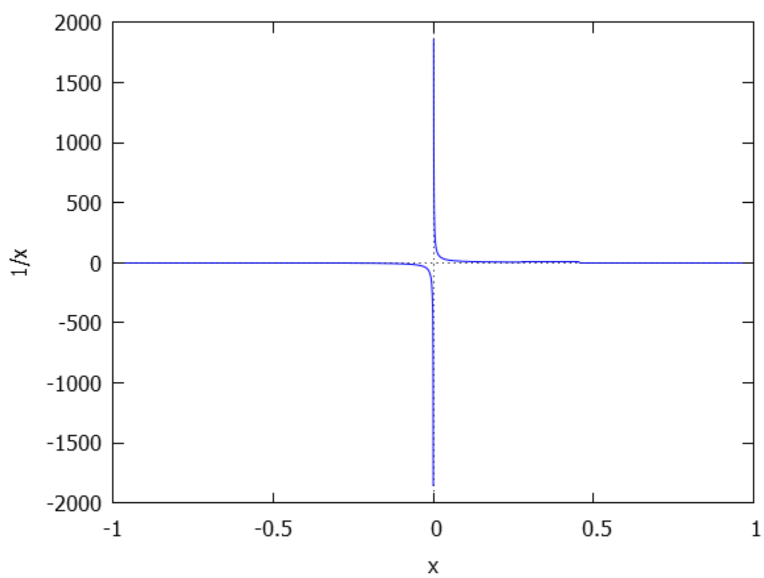
$$\frac{b^2x^2}{a^2b^2} + \frac{a^2y^2}{a^2b^2} = \frac{a^2b^2}{a^2b^2}$$

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

Menggambar Ellips Jika Diketahui Dua Titik Fokus

Untuk menggambar ellips , langkah pertama yaitu dengan menggunakan "load(draw)"

```
>$load(draw):
```



"Draw" adalah sistem antarmuka dari Maxima-Gnuplot.
Ada tiga fungsi utama yang digunakan pada tingkat Maxima, yaitu:
"draw2d", "draw3d", dan "draw".
Perintah "ellipse (<xp>, <yp>, <a>, <b>, <ang1>, <ang2>)" menggambar
sebuah elips yang berpusat di "[<xp>, <yp>]" dengan sumbu semi
horizontal dan vertikal sebesar <a> dan <b>, berturut-turut, dimulai
dari sudut <ang1> sejauh sudut <ang2>.
Soal 1
Sebuah elips memiliki fokus di (0,-2) dan (0,2) serta melalui titik (0, 6). Tentukan persamaan elips tersebut dan gambarkan.
Penyelesaian
Titik Fokus
F1 = (-c, 0) = (2, 0)
F2 = (c, 0) = (2, 0)
Melalui titik (0, 6)

$$d_1 = \sqrt{(x+c)^2 + y^2}$$

```
>d1 &= sqrt((0+(2))^2+(6)^2); $d1 // d1 = jarak F1 ke titik (0,6)
```

$$2\sqrt{10}$$

$$d_2 = \sqrt{(x-c)^2 + y^2}$$

```
>d2 &= sqrt((0-2)^2+(6)^2); $d2 // d2 = jarak F2 ke titik (0,6)
```

$$2\sqrt{10}$$

Diketahui

$$d_1 + d_2 = 2a$$

```
>a &= (d1+d2)/2; $a // menentukan nilai a
```

$$2\sqrt{10}$$

```
>xp &= (2+2)/2 // titik pusat ellips xp
```

2

```
>yp &= (0-0)/2 // titik pusat ellips yp
```

0

Titik pusat ellips (xp, yp) = (2, 0)

```
>c &= (2+2)/2 // jarak fokus ke pusat
```

2

```
>b &= sqrt(a^2-c^2) // menentukan nilai b
```

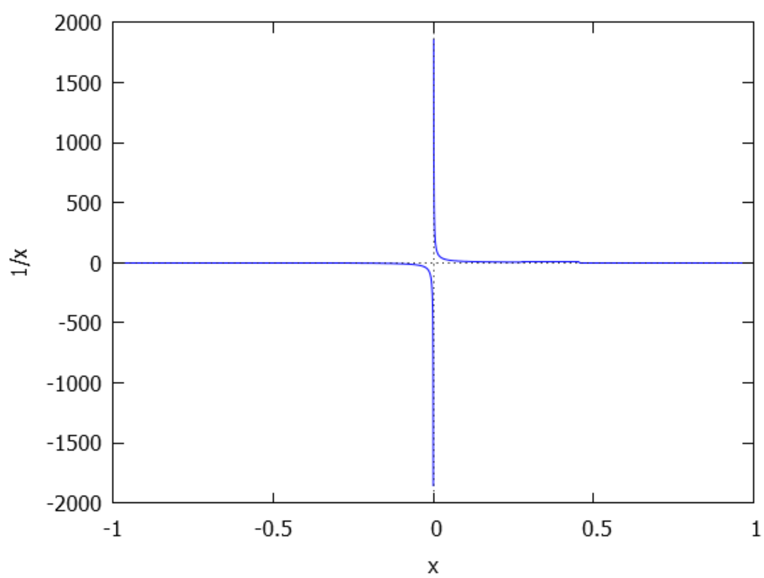
6

```
>$(x-xp)^2/a^2+(y-yp)^2/b^2=1 // persamaan ellips
```

$$\frac{y^2}{36}+\frac{(x-2)^2}{40}=1$$

Selanjutnya, kita akan menggambar ellips menggunakan fungsi "draw2d"

```
>$draw2d(ellipse(2, 0, 6, 6, 360, -360)): // menggambar ellips
```



```
>reset();
```

Materi 11 Melakuka

Berbagai Perhitungan Geometris, seperti Jarak Dua Titik

(Panjang Ruas Garis), Luas Segitiga, Keliling Segitiga,

Luas Lingkaran, Keliling Lingkaran, dan sebagainya.

Pada Sub bab kali ini akan dibahas:
1. jarak dua titik (panjang ruas garis)
2. luas segitiga dan keliling segitiga
3. luas lingkaran dan keliling lingkaran
4. perhitungan geometri lainnya

```
>load geometry
```

```
Numerical and symbolic geometry.
```

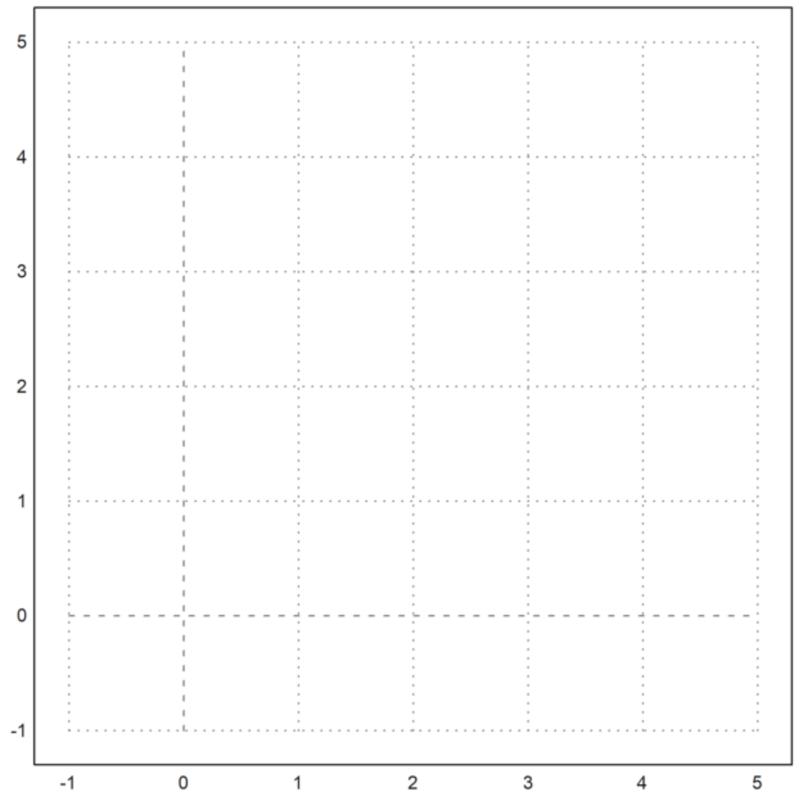
Jarak dua titik

Jarak antara dua titik adalah panjang ruas garis yang menghubungkan kedua titik tersebut.
Jika diketahui dua titik pada koordinat kartesius, misal A(x1,y1) dan B(x2,y2), maka jarak antara titik A dan B adalah

$$AB = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

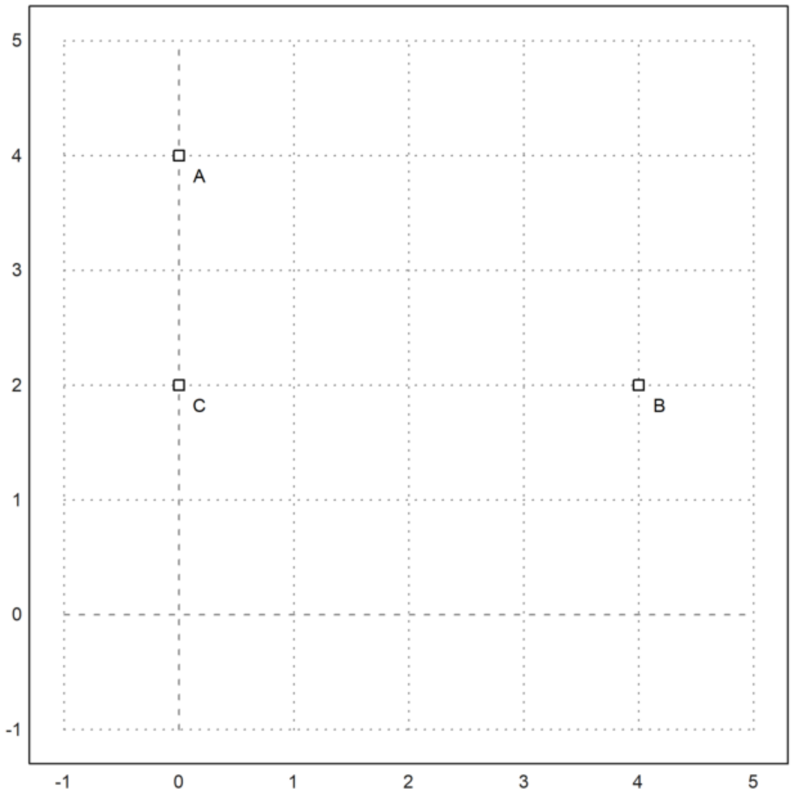
sebagai contoh digambar suatu bidang koordinat

```
>setPlotRange(-1,5,-1,5):
```



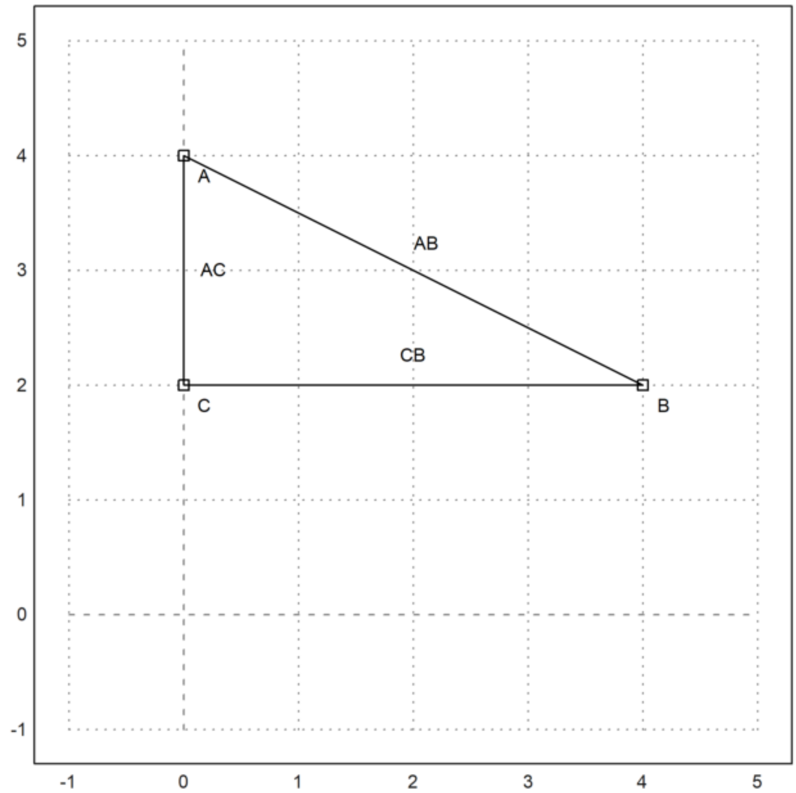
kemudian tentukan dua titik yang akan dihitung jaraknya

```
>A=[0,4]; plotPoint(A,"A");
>B=[4,2]; plotPoint(B,"B");
>C=[0,2]; plotPoint (C,"C"):
```



hitung jarak antara titik A dan titik B

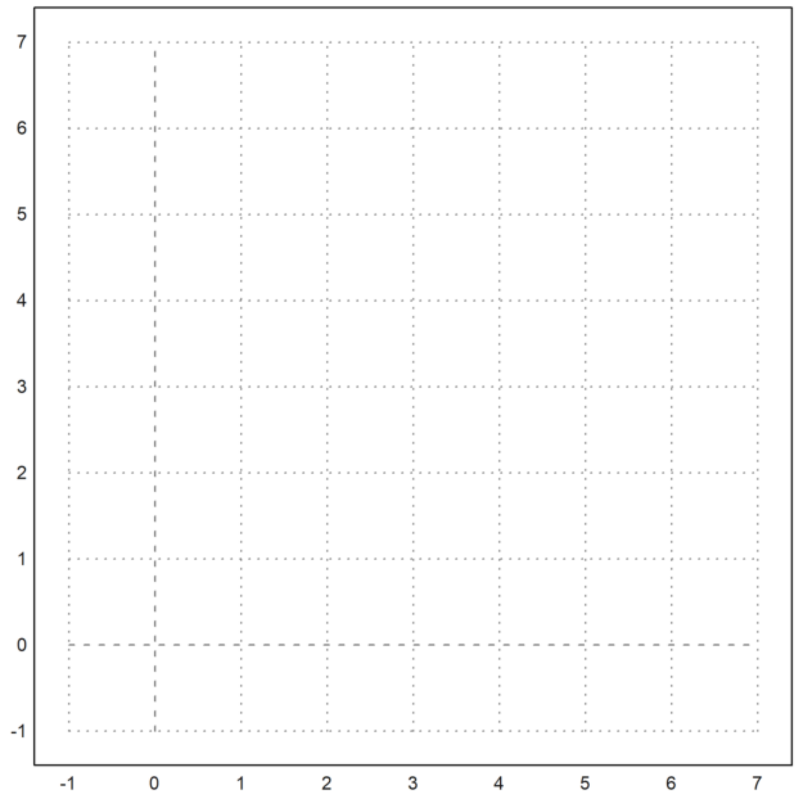
```
>plotSegment(A,B);
>plotSegment(A,C);
>plotSegment(C,B):
```



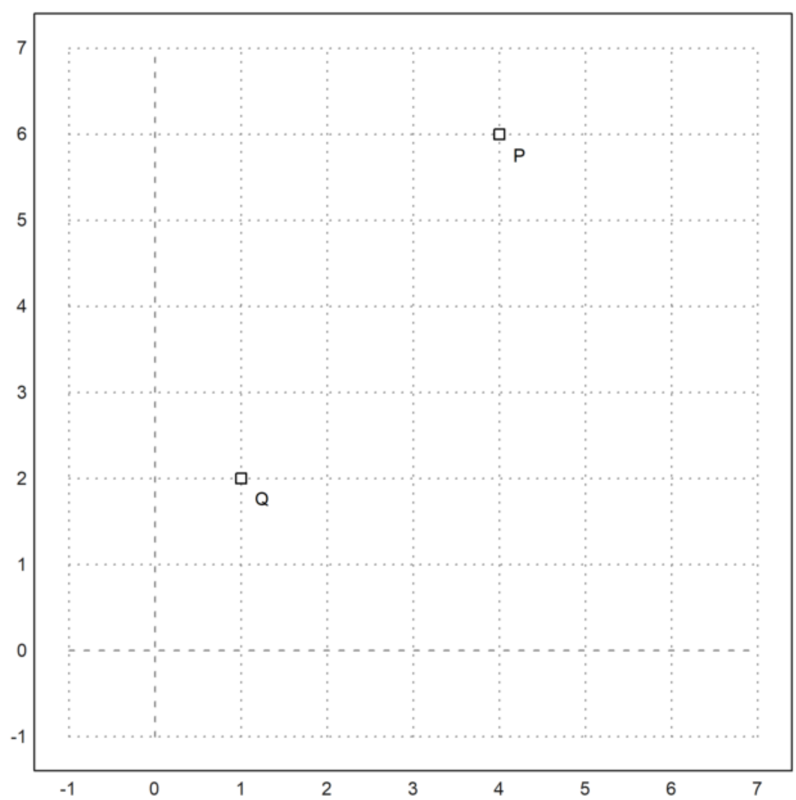
Latihan

1. Diketahui dua buah titik P(2,4) dan Q(1,3). Tentukanlah panjang garis PQ.

```
>setPlotRange(-1,7,-1,7):
```



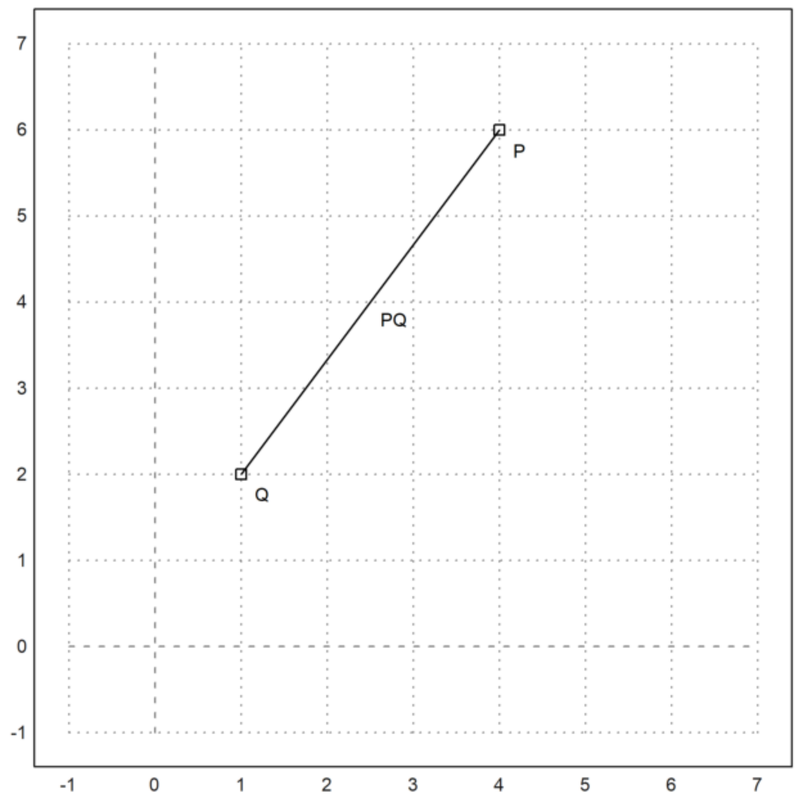

```
>P=[4,6]; plotPoint(P,"P");
>Q=[1,2]; plotPoint(Q,"Q"):
```



```
>distance(P, Q)
```

5

```
>plotSegment(P,Q):
```

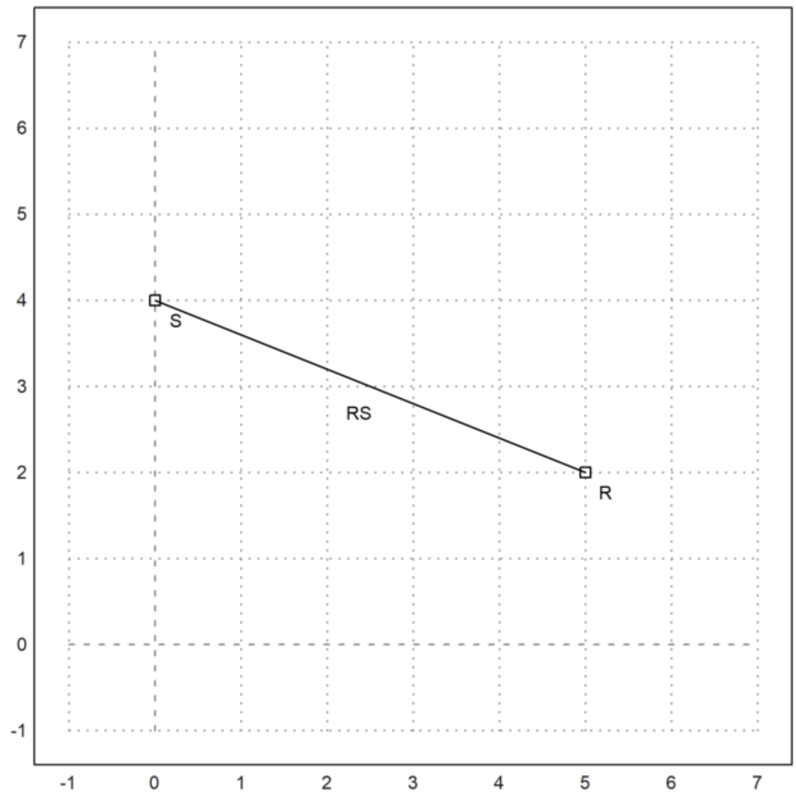


2. Diketahui dua buah titik R(5,2) dan S(0,4). Tentukanlah panjang garis RS.

```
>setPlotRange(-1,7,-1,7);
>R=[5,2]; plotPoint(R,"R");
>S=[0,4]; plotPoint(S,"S");
>distance(R, S)
```

5.38516480713

```
>plotSegment(R,S):
```



Luas dan Keliling Segitiga

```
Terdapat cukup banyak bangun datar yang bisa dihitung luasnya namun
```

pada kesempatan kali ini kita akan mempelajari luas dari bangun datar sedderhana yaitu segitiga. Segitiga dipilih menjadi topik kali ini bukan tanpa alasan namun karena bangun datar dengan sisi-n teratur paling tidak terbentuk dari segitiga sama sisi.
Selanjutnya sebelum kita ke rumus dari kedua bangun datar tersebut kita akan mencari tahu dulu apa itu luas. Luas adalah sebuah ukuran yang digunakan untuk mengukur seberapa besar atau seberapa banyak ruang atau wilayah yang ditempati oleh suatu objek atau bentuk dalam ruang dua atau tiga dimensi dan kali ini akan berfokus pada ruang dua dimensi terlebih dahulu. Maka dari itu ketika disediakan sebuah persegi atau segiempat kita bisa mengukur luasnya dengan:
Segitiga memiliki beberapa jenis, ada segitiga siku-siku, segitiga sama sisi, segitiga sama kaki, dan segitiga sembarang.
Untuk menghitung luas dari segitiga siku-siku, segitiga sama sisi dan segitiga sama kaki menggunakan rumus :

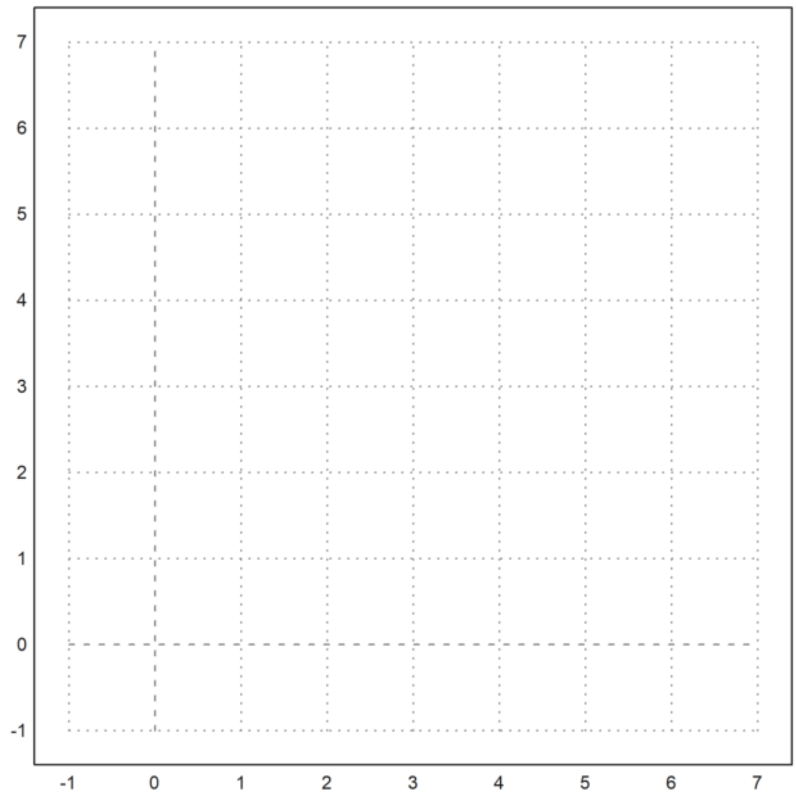
$$L_{\triangle} = \frac{1}{2} alas.tinggi$$

Sedangkan untuk mencari luas dari segitiga sembarang dapat menggunakan rumus :

$$L = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{dengan } s = \frac{(a+b+c)}{2}$$

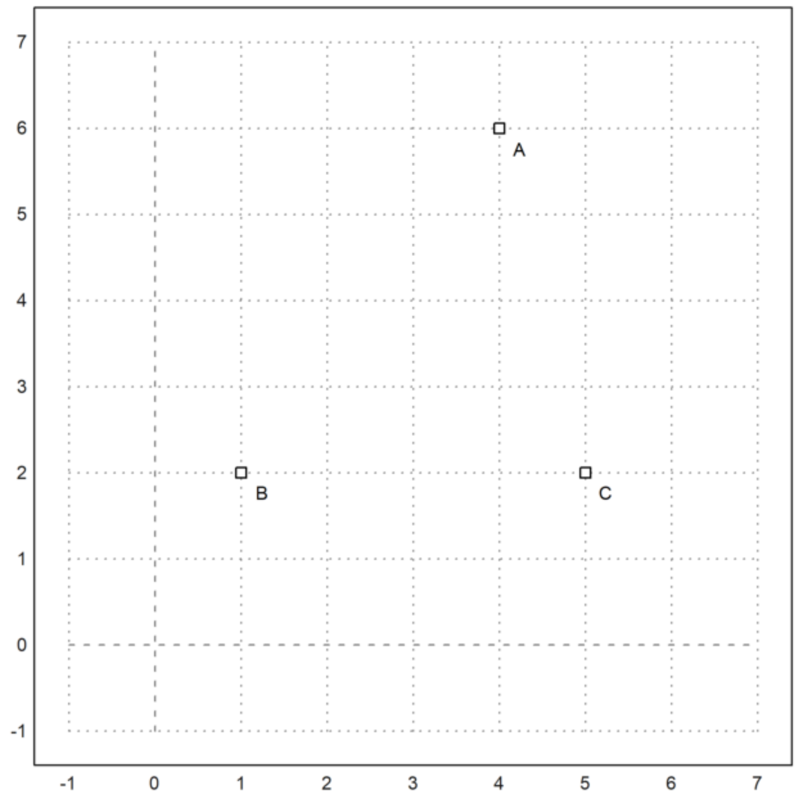
Sebagai contoh digambar sebuah segitiga sembarang.

```
>setPlotRange(-1,7,-1,7):
```



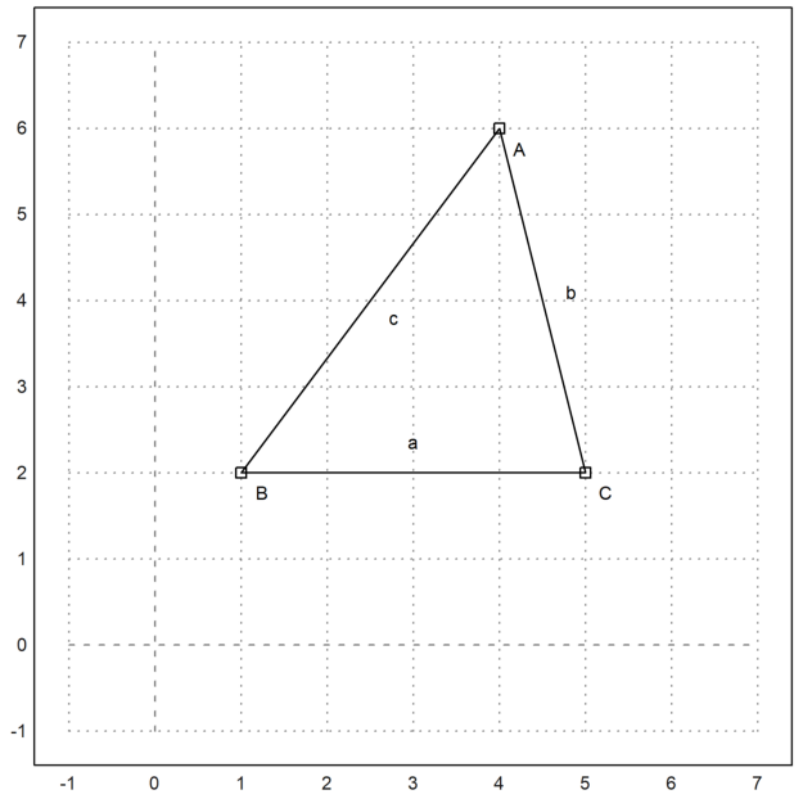
dipilih tiga titik sembarang.

```
>A=[4,6]; plotPoint(A,"A");
>B=[1,2]; plotPoint(B,"B");
>C=[5,2]; plotPoint(C,"C"):
```



kemudian menentukan tiga segmen garis.

```
>plotSegment(A,B,"c"); // c=AB
>plotSegment(B,C,"a"); // a=BC
>plotSegment(A,C,"b"): // b=AC
```



menentukan masing-masing panjang dari a, b, dan c.

```
>distance (B, C)
```

```
4
```

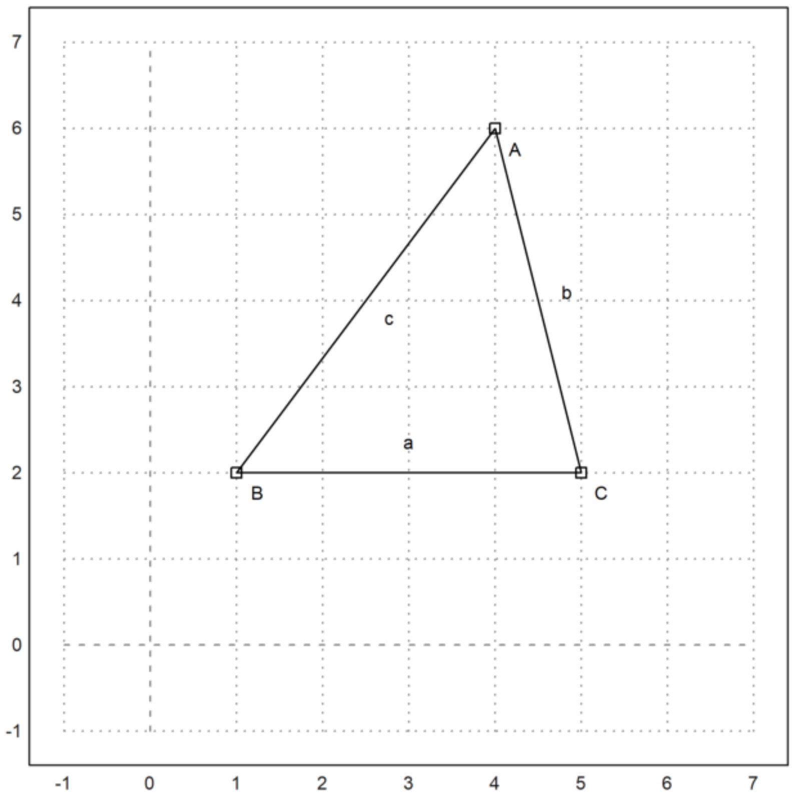
```
>distance (A, C)
```

```
4.12310562562
```

```
>distance (A, B)
```

```
5
```

```
>areaTriangle(A,B,C):
```




```
>areaTriangle(A,B,C)
```

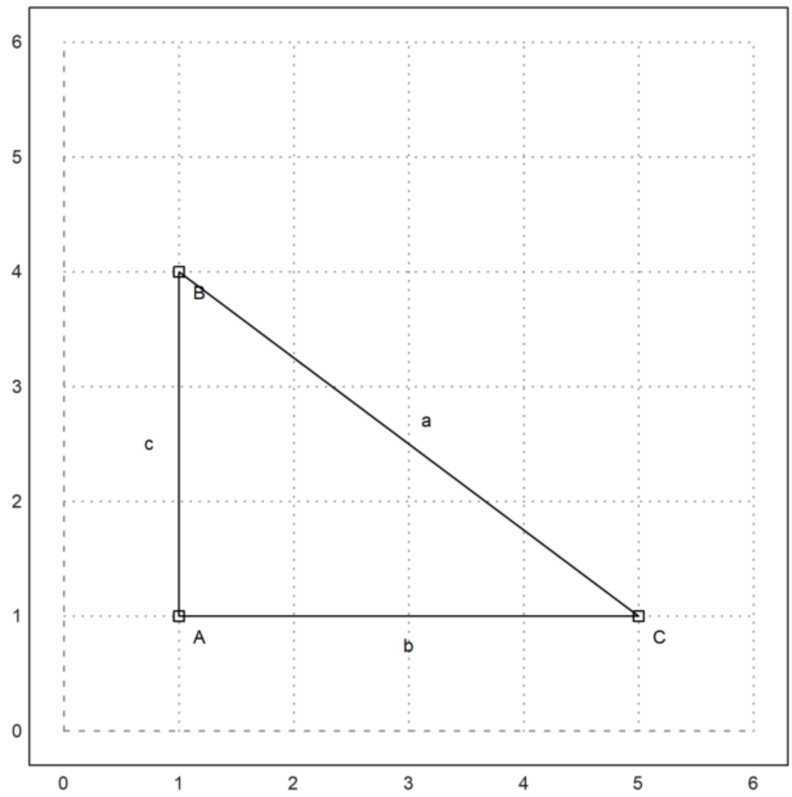
8

```
>setPlotRange(0,6,0,6) // mendefinisikan bidang koordinat
```

[0,   6,   0,   6]

Segitiga Siku-Siku

```
>A=[1,1]; plotPoint (A,"A");
>B=[1,4]; plotPoint (B,"B");
>C=[5,1]; plotPoint (C,"C");
>plotSegment (A,B,"c"); // c=AB
>plotSegment (B,C,"a"); // a=BC
>plotSegment (C,A,"b"): // b=CD
```




```
>b=distance (A, C)
```

4

```
>c=distance (A, B)
```

3

```
>a=distance (B,C)
```

5

```
>Luas:=(distance (A, C)*distance (A, B))/2
```

6

```
>areaTriangle (A,B,C)
```

6

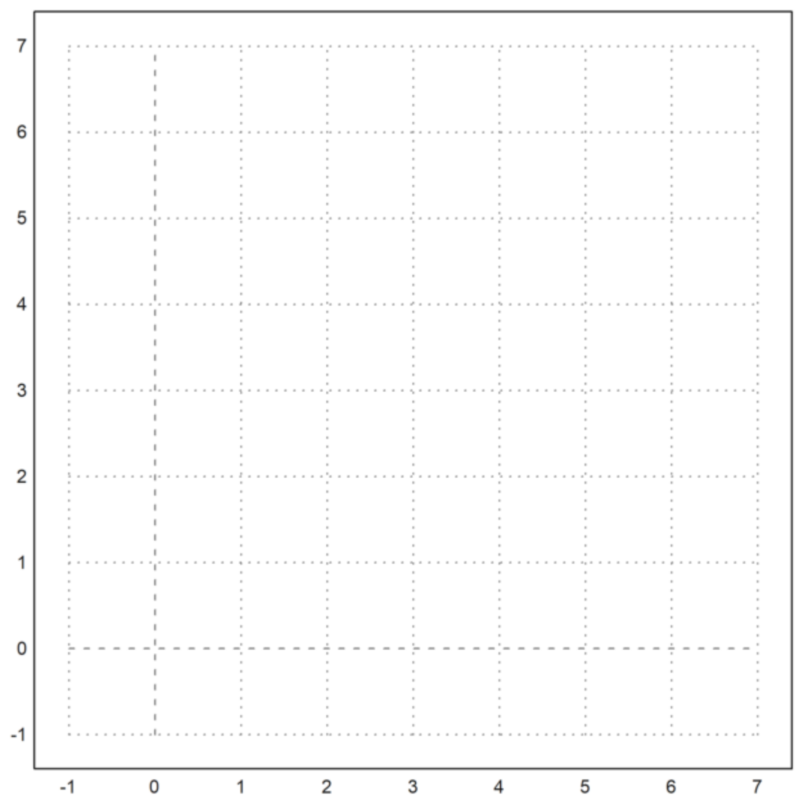
```
>KelilingSegitiga:= b+c+a
```

```
12
```

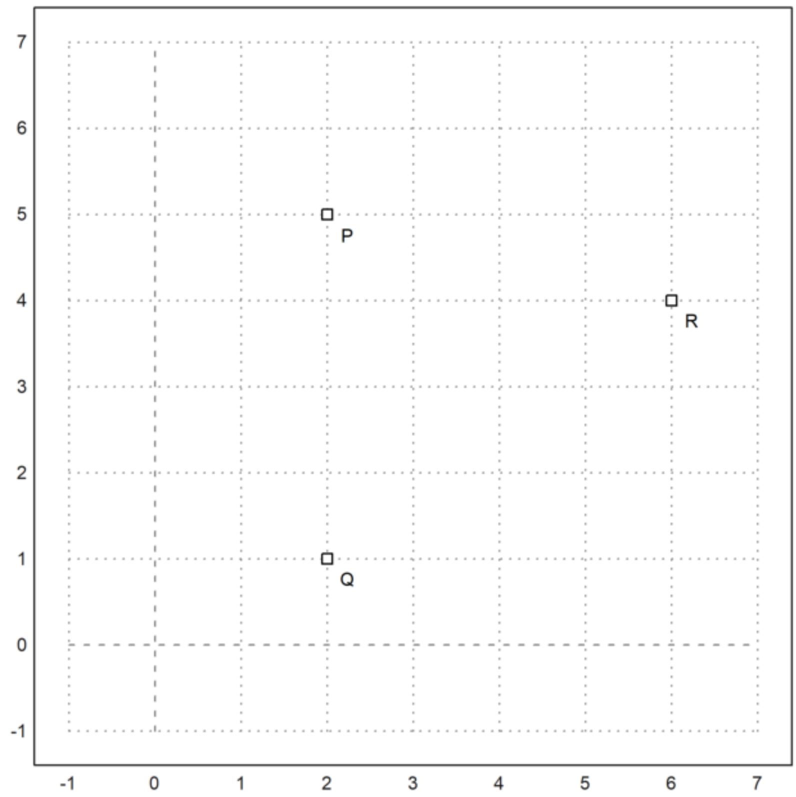
Latihan

1. Diketahui sebuah segitiga PQR dengan titik P(2,5), Q(2,1), dan R(6,4)
Hitunglah luas dari segitiga PQR tersebut.

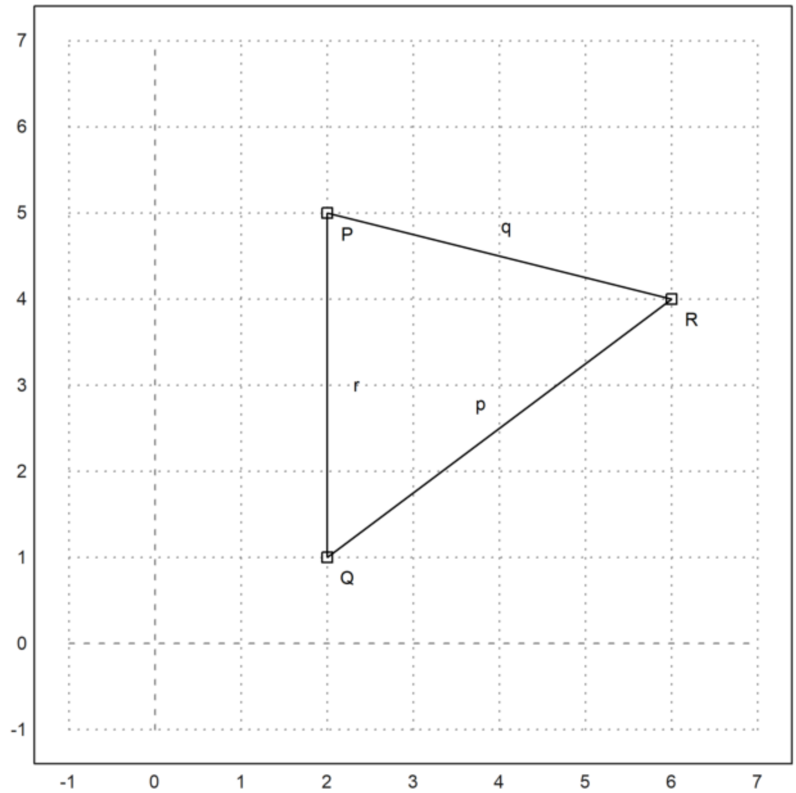
```
>setPlotRange(-1,7,-1,7):
```




```
>P=[2,5]; plotPoint(P,"P");
>Q=[2,1]; plotPoint(Q,"Q");
>R=[6,4]; plotPoint(R,"R"):
```




```
>plotSegment(P,Q,"r"); // r=PQ
>plotSegment(Q,R,"p"); // p=QR
>plotSegment(P,R,"q"): // q=PR
```




```
>distance (Q, R)
```

    5

```
>distance (P, Q)
```

    4

```
>distance (R, P)
```

    4.12310562562

```
>areaTriangle(R,Q,P)
```

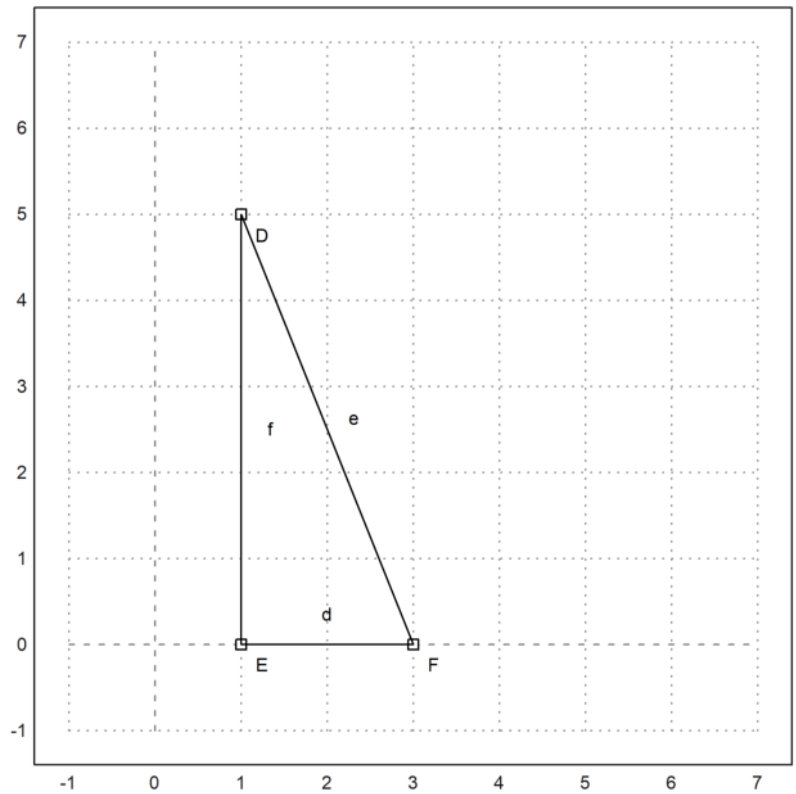
    8

```
>Luas:=(distance (R, P)*distance(Q, R))/2
```

    10.307764064

2. Diketahui sebuah segitiga DEF dengan titik D(1,5), E(1,0), dan F(3,0)
Hitunglah luas dari segitiga DEF tersebut.

```
>setPlotRange(-1,7,-1,7);
>D=[1,5]; plotPoint(D,"D");
>E=[1,0]; plotPoint(E,"E");
>F=[3,0]; plotPoint(F,"F");
>plotSegment(D,E,"f"); // f=DE
>plotSegment(E,F,"d"); // d=EF
>plotSegment(D,F,"e"): // e=DF
```




```
>distance (D, E)
```

5

```
>distance (E, F)
```

2

```
>areaTriangle(D,E,F)
```

5

Luas dan keliling lingkaran

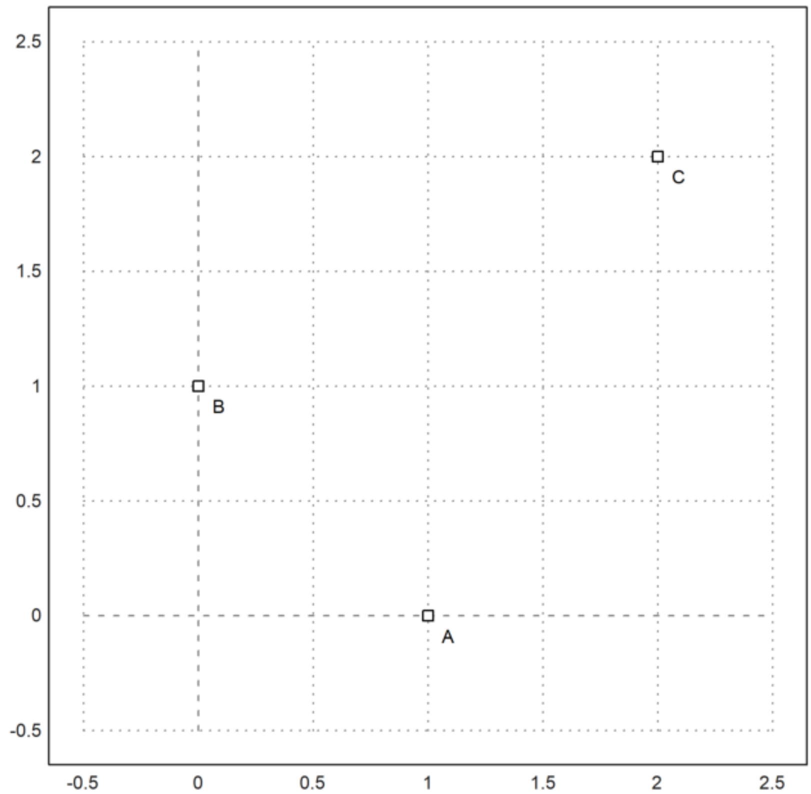
Luas lingkaran adalah luasan daerah pada lingkaran tersebut.
Luas pada sebuah lingkaran memiliki rumus :

$$L = \pi . r^2$$

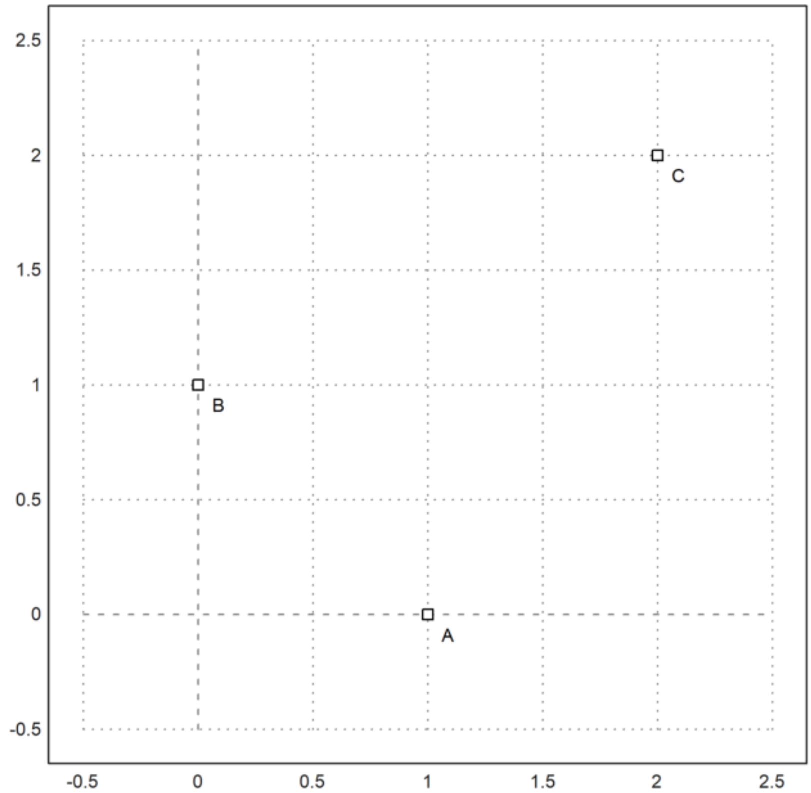
sedangkan untuk keliling lingkaran memiliki rumus :

$$K = 2.\pi . r$$

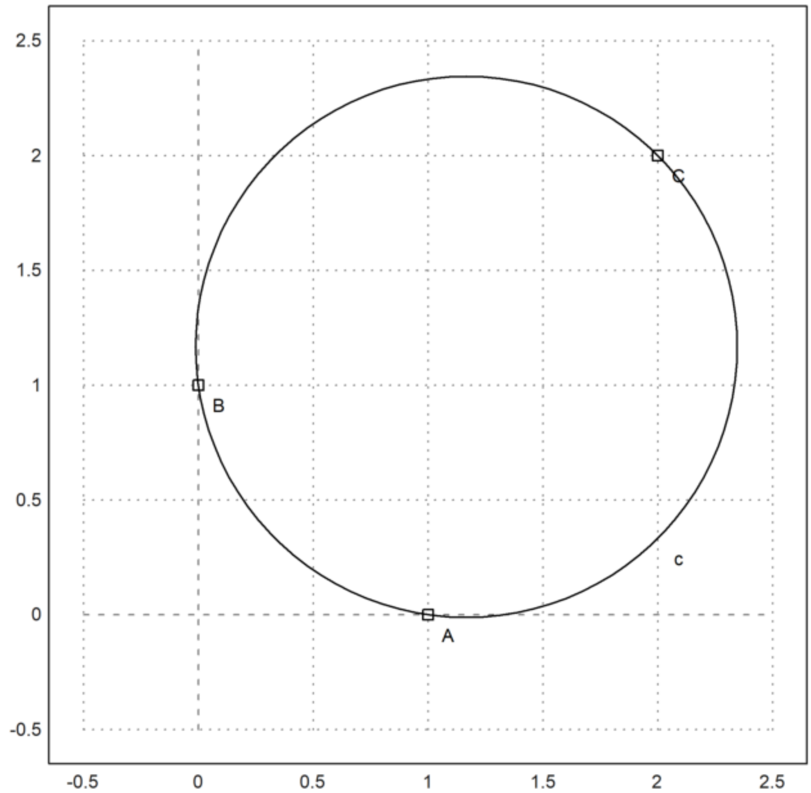
```
>setPlotRange(-0.5,2.5,-0.5,2.5);
>A=[1,0]; plotPoint(A,"A");
>B=[0,1]; plotPoint(B,"B");
>C=[2,2]; plotPoint(C,"C"):
```




```
>c=circleThrough(A,B,C):
```




```
>plotCircle(c):
```



```
>r=getCircleRadius(c)
```

```
1.17851130198
```

```
>LuasLingkaran:=pi*r^2
```

```
4.36332312999
```

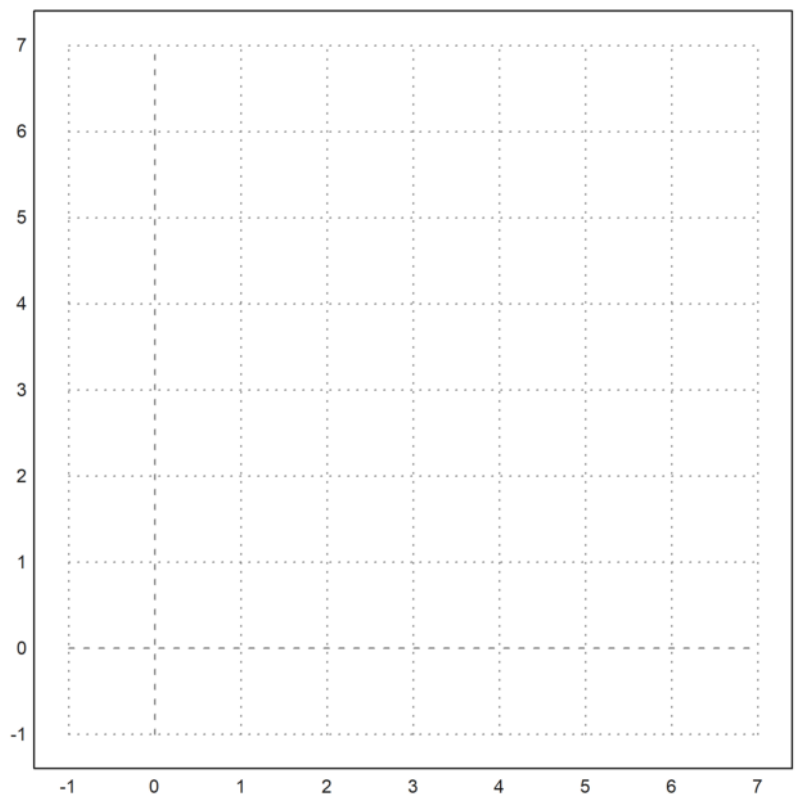
```
>KelilingLingkaran:=2*pi*r
```

```
7.40480489693
```

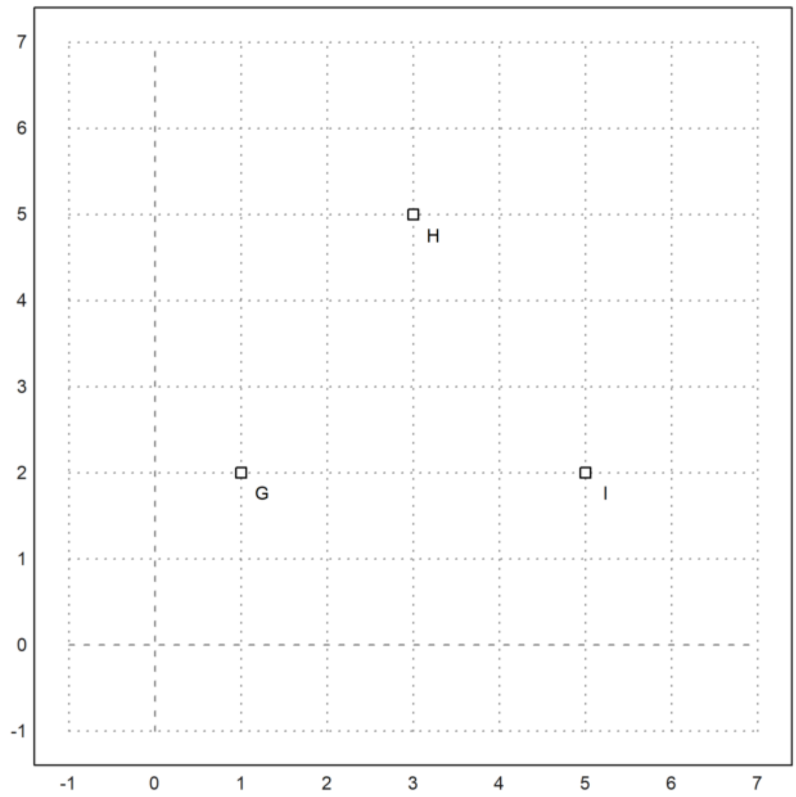
Latihan

Diketahui sebuah lingkaran yang melalui titik G(1,2), H(3,5), dan I(5,2). Hitunglah luas dan keliling lingkaran tersebut.

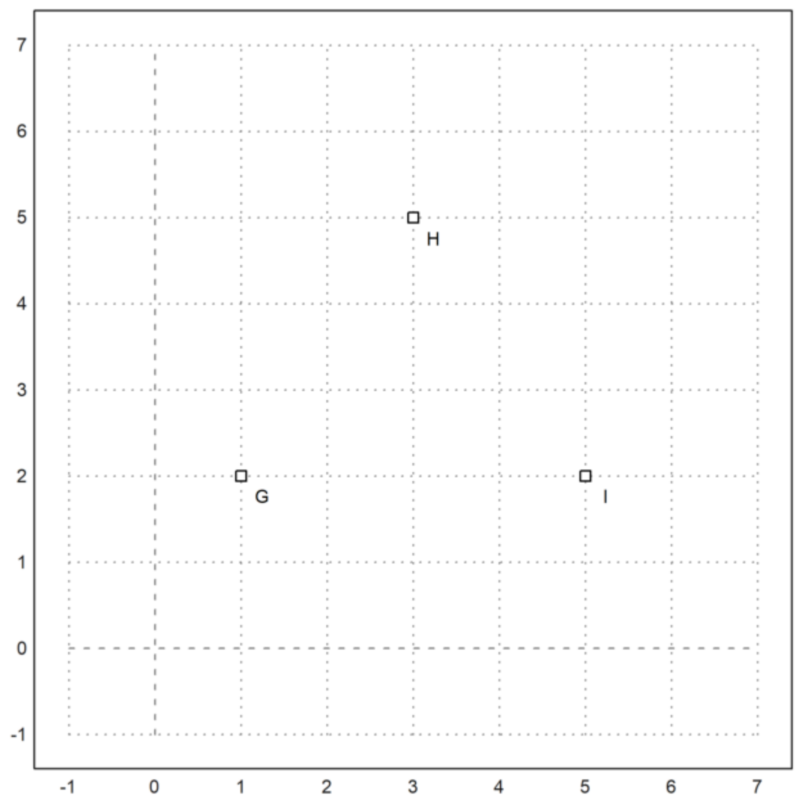
```
>setPlotRange(-1,7,-1,7):
```



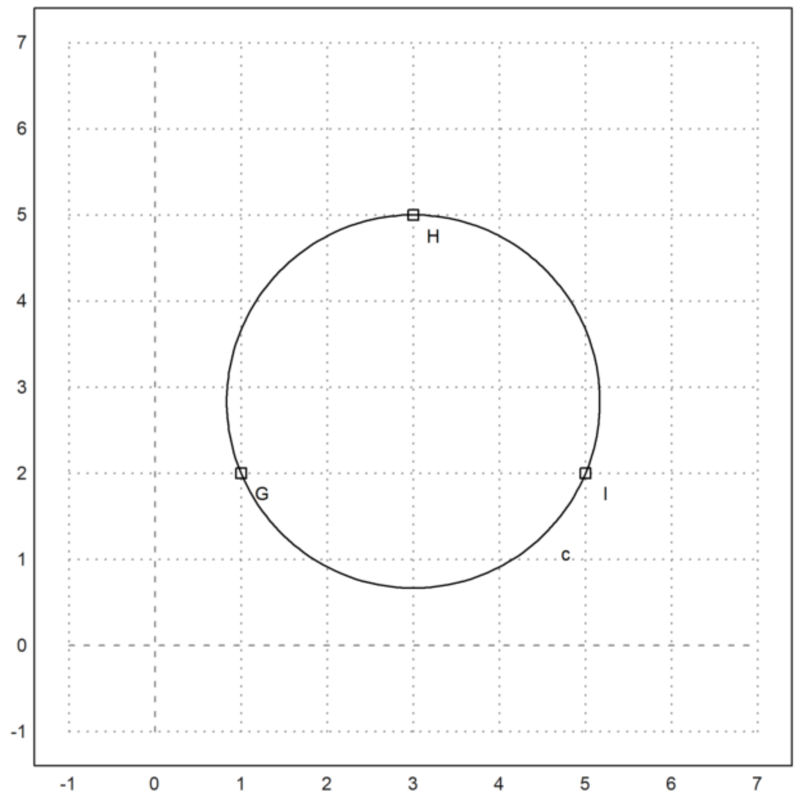
```
>G=[1,2]; plotPoint(G,"G");
>H=[3,5]; plotPoint(H,"H");
>I=[5,2]; plotPoint(I,"I"):
```




```
>c=circleThrough(G,H,I):
```



```
>plotCircle(c):
```




```
>r=getCircleRadius(c)
```

```
2.16666666667
```

```
>LuasLingkaran:=pi*r^2
```

```
14.7480321794
```

```
>KelilingLingkaran:=2*pi*r
```

```
13.6135681656
```

Perhitungan Geometri lainnya

Perhitungan sudut pada bangun datar

```
Perhitungan sudut dalam konteks bangun datar dan antara dua garis
```

melibatkan pengukuran besarnya sudut antara dua garis atau sudut dalam suatu bangun datar. Sudut diukur dalam derajat (°) atau radian (rad) tergantung pada preferensi dan konteks pengukuran yang digunakan. Dan dalam kesempatan kali ini kita akan menggunakan sudut derajat (°). Contoh:

$$S = ((n-2).180) - (k)$$

dimana S adalah sudut yang akan ditentukan dalam derajat (°)

```
n adalah jumlah sudut yang ada pada bangun dtar tersebut
dan k adalah jumlah sududt segitiga yang diketahui
```

contoh:
Andi ingin membuat segitiga menggunakan kawat dan dia ingin membuat segitiga siku-siku dan dia telah menentukan bahwa salah satu sudut selain siku-sikunya adalah 30° maka berapakah sudut yang belkum diketahuinya? **Geometri Bumi**

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan sferis. Fungsi-fungsi tersebut terdapat dalam file "spherical.e" di folder contoh. Kita perlu memuat file itu terlebih dahulu.

```
>load "spherical.e";
```

Untuk memasukkan posisi geografis, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut koordinat Kampus UNY pusat.

```
>UNY=[rad(-7,-46.17),rad(110,22.39)]
```

```
[-0.135603,  1.92638]
```

Anda dapat mencetak posisi ini dengan sposprint (cetak posisi spherical).

```
>sposprint(UNY) // posisi garis lintang dan garis bujur UNY
```

```
S 7°46.170' E 110°22.390'
```

Mari kita tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,24.533)];
>sposprint(Solo), sposprint(Semarang),
```

```
S 7°34.333' E 110°49.683'
S 6°59.050' E 110°24.533'
```

Pertama kita menghitung vektor dari satu ke yang lain pada bola ideal. Vektor ini [pos,jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang 7°.

```
>br=svector(UNY,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan jarak UNY-Solo
```

```
66°23'43.70''
54.7764836559
```

Ini adalah perkiraan yang baik. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak yang begitu pendek hasilnya hampir sama.

```
>esdist(UNY,Semarang)->" km", // perkiraan jarak UNY-Semarang
```

```
87.5069716842 km
```

Ada fungsi untuk heading, dengan mempertimbangkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(UNY,Solo))
```

```
      66.40°
```

Sudut segitiga melebihi 180° pada bola.

```
>asum=sangle(Solo,UNY,Semarang)+sangle(UNY,Solo,Semarang)+sangle(UNY,Semarang,Solo); degpr
```

```
180°0'10.91''
```

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, ini tidak akurat karena kesalahan pengurangan dalam asum-pi.

```
>(asum-pi)*rearth(48°)^2->" km^2", // perkiraan luas segitiga UNY-Solo-Semarang,
```

```
2142.91458761 km^2
```

Ada fungsi untuk ini, yang menggunakan garis lintang rata-rata segitiga untuk menghitung jari-jari bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

```
>esarea(Solo,UNY,Semarang)->" km^2", //perkiraan yang sama dengan fungsi esarea()
```

```
2150.62504635 km^2
```

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi heading dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kami menggunakan vektor. Untuk menambahkan vektor ke posisi, kami menggunakan vektor sadd.

```
>v=svector(UNY,Solo); sposprint(saddvector(UNY,v)), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'
S 7°34.333' E 110°49.683'
```

Fungsi-fungsi ini mengasumsikan bola yang ideal. Hal yang sama di bumi.

```
>sposprint(esadd(UNY,esdir(UNY,Solo),esdist(UNY,Solo))), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'
S 7°34.333' E 110°49.683'
```

```
>load "spherical.e";
```

Mari kita beralih ke contoh yang lebih besar, Merapi Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
>Merapi=[rad(-7,-22.33),rad(110,15.00)]; Monas=[rad(-6,17.50),rad(106,81.1944)];
>sposprint(Merapi), sposprint(Monas)
```

```
S 7°22.330' E 110°15.000'
S 5°42.500' E 107°21.194'
```

Menurut Google Earth, jaraknya adalah 429,66 km. Kami mendapatkan pendekatan yang baik.

```
>esdist(Merapi,Monas)->" km", // perkiraan jarak Merapi Jogja - Monas Jakarta
```

```
370.027986205 km
```

Judulnya sama dengan judul yang dihitung di Google Earth.

```
>degprint(esdir(Merapi,Monas))
```

```
299°51'45.84''
```

Namun, kita tidak lagi mendapatkan posisi target yang tepat, jika kita menambahkan heading dan jarak ke posisi semula. Hal ini terjadi, karena kita tidak menghitung fungsi invers secara tepat, tetapi mengambil perkiraan jari-jari bumi di sepanjang jalan.

```
>sposprint(esadd(Merapi,esdir(Merapi,Monas),esdist(Merapi,Monas)))
```

```
S 5°42.500' E 107°21.195'
```

Namun, kesalahannya tidak besar.

```
>sposprint(Monas),
```

```
S 5°42.500' E 107°21.194'
```

Tentu kita tidak bisa berlayar dengan tujuan yang sama dari satu tujuan ke tujuan lainnya, jika kita ingin menempuh jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti heading yang konstan!
Perhitungan berikut menunjukkan bahwa kami jauh dari tujuan yang benar, jika kami menggunakan pos yang sama selama perjalanan kami.

```
>dist=esdist(Merapi,Monas); hd=esdir(Merapi,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh dari jarak, menggunakan pos ke Monas, kita sampai di Merapi.

```
>p=Merapi; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya jauh.

```
>sposprint(p), skmprint(esdist(p,Monas))
```

```
  S 5°42.969' E 107°20.921'
       1.006km
```

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran garis lintang 30°, melainkan jalur terpendek yang dimulai 10° lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

```
       79.69°
```

Tapi, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kami menyesuaikannya pada 1/10 dari total jarak.

```
>p=P1;  dist=esdist(P1,P2); ...
>  loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
       79.69°
       81.67°
       83.71°
       85.78°
       87.89°
       90.00°
       92.12°
       94.22°
       96.29°
       98.33°
```

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti heading yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

```
       0.203km
```

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan pos setelah setiap 1/100 dari total jarak dari Merapi ke Monas.

```
>p=Merapi; dist=esdist(Merapi,Monas); ...
>  loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;
>skmprint(esdist(p,Monas))
```

```
        0.000km
```

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS di sepanjang lingkaran besar menuju Monas dengan fungsi navigasi.

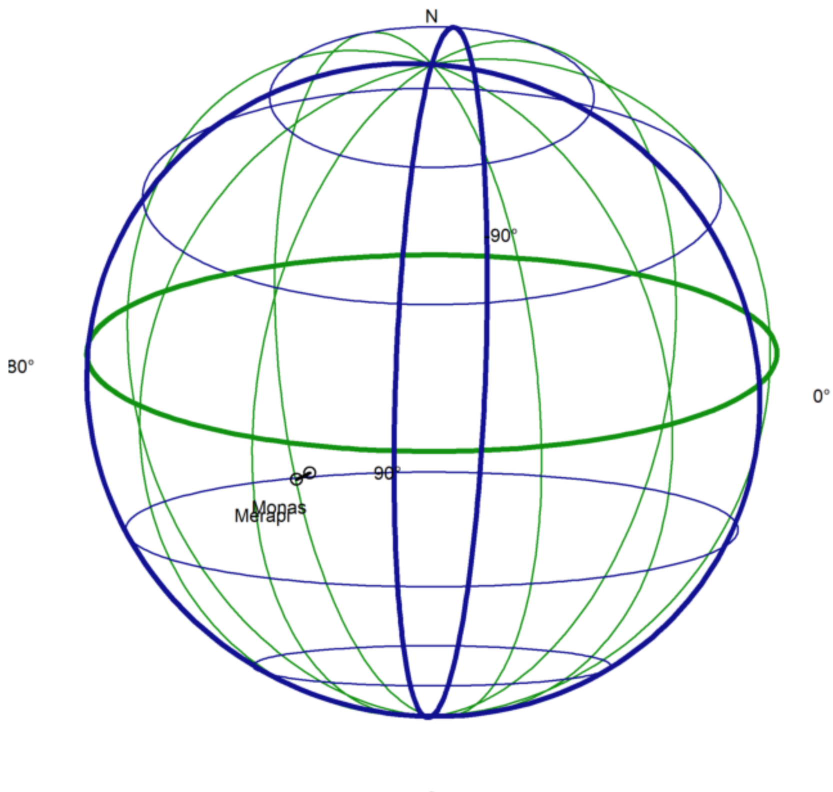
```
>load spherical; v=navigate(Merapi,Monas,10); ...
>loop 1 to rows(v); sposprint(v[#]), end;
```

```
  S 7°22.330' E 110°15.000'
  S 7°12.393' E 109°57.565'
  S 7°2.445' E 109°40.144'
  S 6°52.487' E 109°22.734'
  S 6°42.518' E 109°5.337'
  S 6°32.539' E 108°47.952'
  S 6°22.550' E 108°30.578'
  S 6°12.551' E 108°13.216'
  S 6°2.543' E 107°55.865'
  S 5°52.526' E 107°38.524'
  S 5°42.500' E 107°21.194'
```

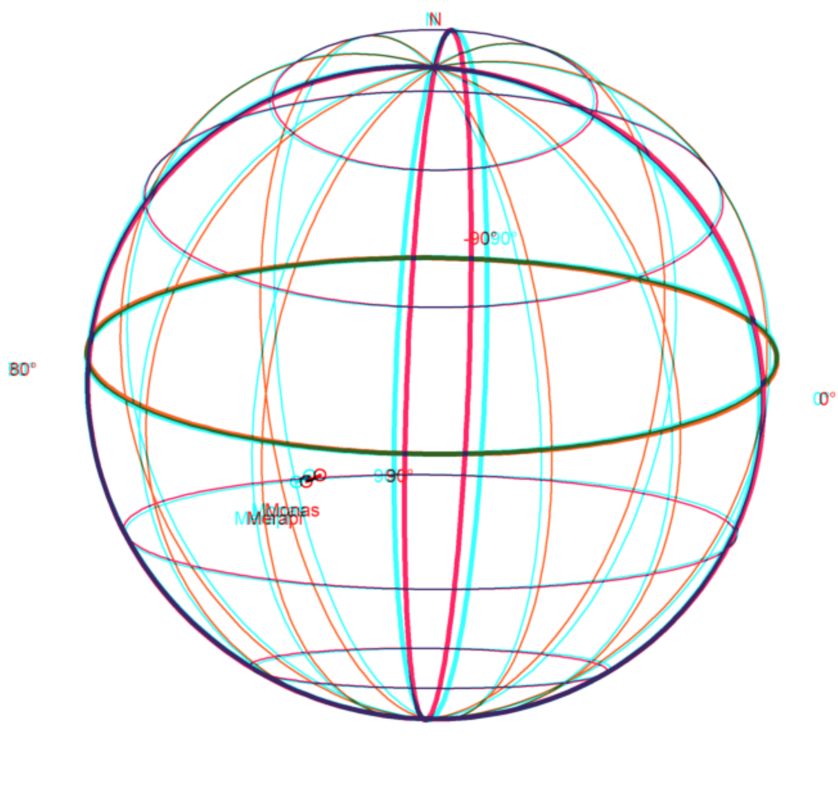
```
>function testplot ...
```

```
  useglobal;
  plotearth;
  plotpos(Merapi,"Merapi"); plotpos(Monas,"Monas");
  plotposline(v);
  endfunction
```

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```



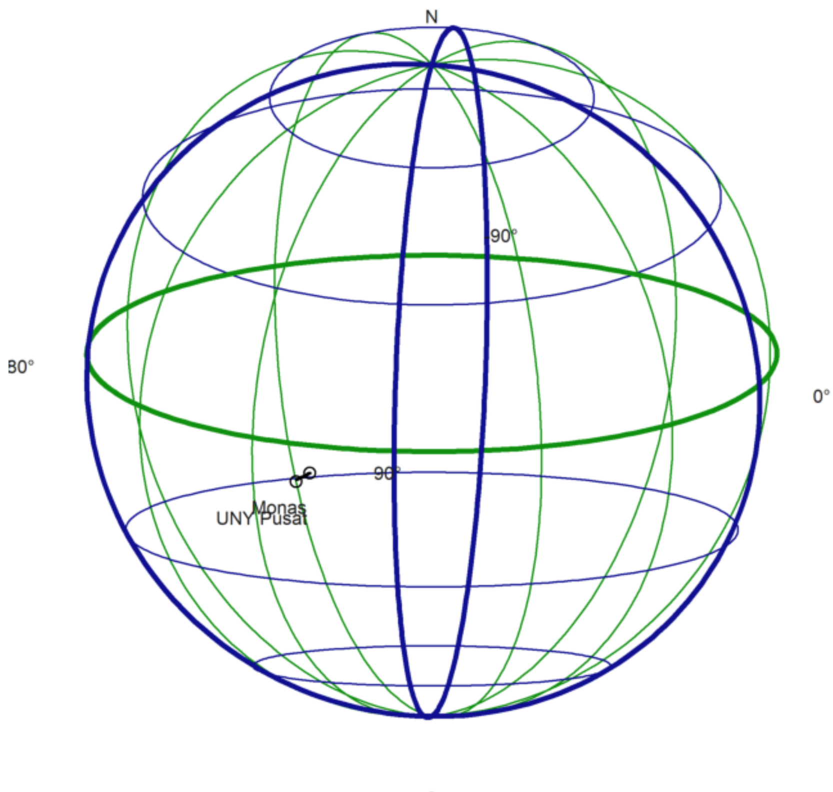

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```



```
>function testplot ...
```

```
useglobal;
plotearth;
plotpos(UNY,"UNY Pusat"); plotpos(Monas,"Monas");
plotposline(v);
endfunction
```

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```



BAB 6

MENGGUAKAN EMT UNTUK STATISTIKA

EMT untuk Statistika

In this notebook, we demonstrate the main statistical plots, tests and distributions in Euler.
Let us start with some descriptive statistics. This is not an introduction to statistics. So you might need some background to understand the details.
Assume the following measurements. We wish to compute the mean value and the measured standard deviation.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...
>median(M), mean(M), dev(M),
```

```
999
999.9
2.72641400622
```

We can plot the box-and-whiskers plot for the data. In our case there are no outliers.

```
>aspect(1.75); boxplot(M):
```

We compute the probability that a value is bigger than 1005, assuming the measured values from a normal distribution.
All functions for distributions in Euler end with ...dis and compute the cumulative probability distribution (CPF).

$$\text{normaldis(x,m,d)} = \int_{-\infty}^{x} \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} \, dt.$$

We print the result in % with 2 digits accuracy using the print function.

```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
```

```
3.07 %
```

For the next example, we assume the following numbers of men in given a size ranges.

```
>r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Here is a plot of the distribution.

```
>plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="\/"):
```

We can put such raw data into a table.
Tables are a method to store statistical data. Our table should contain three columns: Start of range, end of range, number of men in the range.
Tables can be printed with headers. We use a vector of strings to set the 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
```

If we need the mean value and other statistics of the sizes, we need to compute the midpoint of the ranges. We can use the first two columns of our table for this.
The sumbol "|" is used to separate column, the function "writetable" is used to write the table, with opsions "labc" is to specify column 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
```

But it is easier, to fold the ranges with the vector [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]
```

Now we can compute the mean and deviation of the sample with the given frequencies.

```
>{m,d}=meandev(M,v); m, d,
```

```
169.901234568
5.98912964449
```

Let us add the normal distribution of the values to the above bar plot. The formula for normal distribution with mean m and standard deviation d is:

$$y = \frac{1}{d\sqrt{2\pi}} e^{\frac{-(x-m)^2}{2d^2}}.$$

Because its values are between 0 and 1, to plot it on the bar plot it must be multiplied by 4 times the total number of data.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...
>  xmin=min(r),xmax=max(r),thickness=3,add=1):
```

Tables

In the directory of this notebook you find a file with a table. The data represent the results of a survey. Here are the first four lines of the file. The data are from an German online book "Einführung in die Statistik mit R" by 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
```

The table contains 7 columns of numbers or tokens (strings). We want read the table from the file. First, we use our own translation for the tokens.
Fo this, we define sets of tokens. The function strtokens() gets a string vector of tokens from a given string.

```
>mf:=["m","f"]; yn:=["y","n"]; ev:=strtokens("g vg m b vb");
```

Now we read the table with these translations.
The arguments tok2, tok4 etc. are the translations of the columns of the table. These arguments are not in the parameter list of readtable(), so you need to provide them with ":=".

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
>load over statistics;
```

For printing, we need to specify the same token sets. We print the first four lines only.

```
>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
```

The dots "." represent values, which are not available.
If we do not want to specify the tokens for the translation in advance, we only need to specify, which columns contain tokens and not numbers.

```
>ctok=[2,4,5,7]; {MT,hd,tok}=readtable("table.dat",ctok=ctok);
```

The function readtable() now returns a set of tokens.

```
>tok
```

```
  m
  n
  f
  y
  g
  vg
```

The table contains the entries from the file with tokens translated to numbers.
The special string NA="." is interpreted as "Not Available", and gets NAN (not a number) in the table. This translation can be changed with the parameters NA, and NAval.

```
>MT[1]
```

```
  [1,  1,  30,  2,  NAN,  1.8,  2]
```

Here is the content of the table with untranslated numbers.

```
>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
```

For convenience, you can put the output of readtable() into a list.

```
>Table={{readtable("table.dat",ctok=ctok)}};
```

Using the same token columns and the tokens read from the file, we can print the table. We can either specify ctok, tok, etc. or use the 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
```

The function tablecol() returns the values of columns of the table, skipping any rows with NAN values("." in the file), and the indices of the columns, which contain these values.

```
>{c,i}=tablecol(MT,[5,6]);
```

We can use this to extract columns from the table for a new table.

```
>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
```

Of course, we need to extract the table itself from the list Table in this case.

```
>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
```

The getstatistics() function returns the elements in a vector, and their counts. We apply it to the "m" and "f" values in the second column of our table.

```
>{xu,count}=getstatistics(tablecol(MT,2)); xu, count,
```

```
  [1,   3]
  [12,   13]
```

We can print the result in a new table.

```
>writetable(count',labr=tok[xu])
```

```
           m        12
           f        13
```

The function selecttable() returns a new table with the values in one column selected from a vector of indices. First we look up the indices of two of our values in the token table.

```
>v:=indexof(tok,["g","vg"])
```

```
  [5,  6]
```

Now we can select the rows of the table, which have any of the values in v in their 5-th row.

```
>MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5);
```

Now we can print the table, with extracted and sorted values in the 5-th column.

```
>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
```

For the next statistic, we want to relate two columns of the table. So we extract column 2 and 4 and sort the table.

```
>i=sortedrows(MT,[2,4]);  ...
>  writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)
```

```
          m          n
          m          n
          m          n
          m          n
          m          n
          m          n
          m          n
          m          y
          m          y
          m          y
          m          y
          m          y
          f          n
          f          y
```

```
          f          y
          f          y
          f          y
          f          y
          f          y
          f          y
          f          y
          f          y
          f          y
          f          y
          f          y
```

With getstatistics(), we can also relate the counts in two columns of the table to each other.

```
>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
```

A table can be written to a file.

```
>filename="test.dat"; ...
>writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
```

Then we can read the table from the file.

```
>{MT2,hd,tok2,hdr}=readtable(filename,>clabs,>rlabs); ...
>writetable(MT2,labr=hdr,labc=hd)
```

```
                     n          y
          m          7          5
          f          1         12
```

And delete the file.

```
>fileremove(filename);
```

Distributions

With plot2d, there is a very easy method to plot a distribution of experimental data.

```
>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
```

Please note the different between the bar plot (sample) and the normal curve (the real distribution) . Reenter the three commands to see another sampling result.

Here is a comparison of 10 simulations of 1000 normal distributed values using a so-called box plot. This plot shows the median, the 25% and 75% quartiles, the minimal and maximal values, and the outliers.

```
>p=normal(10,1000); boxplot(p):
```

To generate random integers, Euler has intrandom. Let us simulate dice throws and plot the distribution.
We use the getmultiplicities(v,x) function, which counts how often the elements of v appear in x. Then we plot the the result using columnsplot().

```
>k=intrandom(1,6000,6);  ...
>columnsplot(getmultiplicities(1:6,k));  ...
>ygrid(1000,color=red):
```

While intrandom(n,m,k) returns uniformly distributed integers from 1 to k, it is possible to use any other given distribution of integers with randpint().
In the following example, the probabilities for 1,2,3 are 0.4,0.1,0.5 respectively.

```
>randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,%)
```

```
 [402,  90,  508]
```

Euler can produce random values from more distributions. Have a look into the reference.
E.g., we try the exponential distribution. A continuous random variable X is said to have an exponential distribution, if its PDF is given by

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

with parameter

$$\lambda = \frac{1}{\mu}, \quad \mu \text{ is the mean, and denoted by } X \sim \text{Exponential}(\lambda).$$

```
>plot2d(randexponential(1,1000,2),>distribution):
```

For many distributions, Euler can compute the distribution function and the inverse.

```
>plot2d("normaldis",-4,4):
```

The following is one way to plot a quantile.

```
>plot2d("qnormal(x,1,1.5)",-4,6);  ...
>plot2d("qnormal(x,1,1.5)",a=2,b=5,>add,>filled):
```

$$\text{normaldis(x,m,d)} = \int_{-\infty}^{x} \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-m}{d}\right)^2}\, dt.$$

The probability to be in the green area is the following.

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

```
0.248662156979
```

This can be computed numerically with the following integral.

$$\int_2^5 \frac{1}{1.5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-1}{1.5})^2}\, dx.$$

```
>gauss("qnormal(x,1,1.5)",2,5)
```

```
0.248662156979
```

Let us compare the binomial distribution with the normal distribution of same mean and deviation. The function invbindis() solves a linear interpolation between integer values.

```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

```
525.516721219
526.007419394
```

The function qdis() is the density of the chi-square distribution. As usual, Euler maps vectors to this function. Thus we get a plot of all chi-square distributions with degrees 5 to 30 easily in the following way.

```
>plot2d("qchidis(x,(5:5:50)')",0,50):
```

Euler has accurate functions to evaluate distributions. Let us check chidis() with an integral.
The naming tries to be consistent. E.g.,
- the chi-square distribution is chidis(),
- the inverse function is invchidis(),
- the density is qchidis().
The complements of the distribution (upper tail) is chicdis().

```
>chidis(1.5,2), integrate("qchidis(x,2)",0,1.5)
```

```
0.527633447259
0.527633447259
```

Discrete Distributions

To define your own discrete distribution, you can use the following method.
First we set the distribution function.

```
>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]
```

The meaning is that with probability wd[i+1]-wd[i] we produce the random value i.
This is almost a uniform distribution. Let us define a random number generator for this. The find(v,x) function finds the value x in the vector v. It works for vectors x too.

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

The error is so subtle that we see it only with very many iterations.

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```

Here is a simple function to check for uniform distribution of the values 1...K in v. We accept the result, if for all frequencies

$$\left| f_i - \frac{1}{K} \right| < \frac{\delta}{\sqrt{n}}.$$

```
>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
```

Indeed the function rejects the uniform distribution.

```
>checkrandom(wrongdice(1,1000000))
```

```
0
```

And it accepts the built-in random generator.

```
>checkrandom(intrandom(1,1000000,6))
```

```
1
```

We can compute the binomial distribution. First there is binomialsum(), which returns the probability of i or less hits out of n trials.

```
>bindis(410,1000,0.4)
```

```
0.751401349654
```

The inverse Beta function is used to compute a Clopper-Pearson confidence interval for the parameter p. The default level is alpha.
The meaning of this interval is that if p is outside the interval, the observed result of 410 in 1000 is rare.

```
>clopperpearson(410,1000)
```

```
[0.37932,  0.441212]
```

The following commands are the direct way to get the above result. But for large n, the direct summation is not accurate and slow.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

```
0.751401349655
```

By the way, invbinsum() computes the inverse of binomialsum().

```
>invbindis(0.75,1000,0.4)
```

```
409.932733047
```

In Bridge, we assume 5 outstanding cards (out of 52) in two hands (26 cards). Let us compute the probability of a distribution worse than 3:2 (e.g. 0:5, 1:4, 4:1 or 5:0).

```
>2*hypergeomsum(1,5,13,26)
```

```
0.321739130435
```

There is also a simulation of multinomial distributions.

```
>randmultinomial(10,1000,[0.4,0.1,0.5])
```

```
            366           117           517
            400           109           491
            395            98           507
            395           103           502
            427           105           468
            414           114           472
            398           102           500
            387           102           511
            387            99           514
            423            98           479
```

Plotting Data

To plot data, we try the results of the German elections since 1990, measured in seats.

```
>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];
```

For the parties, we use a string of names.

```
>P:=["CDU/CSU","SPD","FDP","Gr","Li"];
```

Let us print the percentages nicely.
First we extract the necessary columns. Columns 3 to 7 are the seats of each party, and column 2 is the total number of seats. column is the year of the election.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Then we print the statistics in table form. We use the names as column headers, and the years as headers for the rows. The default width for the columns is wc=10, but we prefer a denser output. The columns will be expanded for the labels of the columns, if necessary.

```
>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
```

The following matrix multiplication extracts the sum of the percentages of the two big parties showing that the small parties have gained footage in the parliament until 2009.

```
>BT1:=(BT.[1;1;0;0;0])'*100
```

```
 [84.29,  81.25,  81.1659,  82.7529,  72.9642,  61.8971,  79.8732]
```

There is also a simple statistical plot. We use it to display lines and points simultaneously. The alternative is to call plot2d twice with >add.

```
>statplot(YT,BT1,"b"):
```

Define some colors for each party.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

Now we can plot the results of the 2009 election and the changes into one plot using figure. We can add a vector of columns to each 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):
```

Data plots combine rows of statistical data in one 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):
```

A 3D columns plot shows rows of statistical data in form of columns. We provide labels for the rows and the columns. angle is the viewing angle.

```
>columnsplot3d(BT,scols=P,srows=YT, ...
>  angle=30°,ccols=CP):
```

Another representation is the mosaic plot. Note that the columns of the plot represent the columns of the matrix here. Because of the length of the label CDU/CSU, we take a smaller window than usual.

```
>shrinkwindow(>smaller);  ...
>mosaicplot(BT',srows=YT,scols=P,color=CP,style="#"); ...
>shrinkwindow():
```

We can also do a pie chart. Since black and yellow form a coalition, we reorder the elements.

```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```

Here is another kind of plot.

```
>starplot(normal(1,10)+4,lab=1:10,>rays):
```

Some plots in plot2d are good for statics. Here is an impulse plot of random data, uniformly distributed in [0,1].

```
>plot2d(makeimpulse(1:10,random(1,10)),>bar):
```

But for exponentially distributed data, we may need a logarithmic plot.

```
>logimpulseplot(1:10,-log(random(1,10))*10):
```

The function columnsplot() is easier to use, since it needs just a vector of values. Moreover, it can set its labels to anything we want, we demonstrated this already in this tutorial.
Here is another application, where we count characters in a sentence and plot a statistics.

```
>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):
```

It is also possible to manually set axes.

```
>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"]):
```

The following is a way to plot the frequencies of numbers in a vector.
We create a vector of integer random numbers 1 to 6.

```
>v:=intrandom(1,10,10)
```

```
[2,   3,   6,   4,   8,   8,   10,   10,   5,   10]
```

Then extract the unique numbers in v.

```
>vu:=unique(v)
```

```
[2,   3,   4,   5,   6,   8,   10]
```

And plot the frequencies in a columns plot.

```
>columnsplot(getmultiplicities(vu,v),lab=vu,style="/"):
```

We want to demonstrate functions for the empirical distribution of values.

```
>x=normal(1,20);
```

The function empdist(x,vs) needs a sorted array of values. So we have to sort x before we can use it.

```
>xs=sort(x);
```

Then we plot the empirical distribution and some density bars into one plot. Instead of a bar plot for the distribution we use a sawtooth plot this time.

```
>figure(2,1); ...
>figure(1); plot2d("empdist",-4,4;xs); ...
>figure(2); plot2d(histo(x,v=-4:0.2:4,<bar));  ...
>figure(0):
```

A scatter plot is easy to do in Euler with the usual point plot. The following graph shows that the X and X+Y are clearly positively correlated.

```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style=".."):
```

Often, we wish to compare two samples of different distributions. This can be done with a quantile-quantile-plot.
For a test, we try the student-t distribution and exponential distribution.

```
>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):
```

The plot clearly shows that the normal distributed values tend to be smaller at the extreme ends.
If we have two distributions of different size, we can expand the smaller one or shrink the larger one. The following function is good for both. It takes the median values with percentages between 0 and 1.

```
>function medianexpand (x,n) := median(x,p=linspace(0,1,n-1));
```

Let us compare two equal distributions.

```
>x=random(1000); y=random(400); ...
>plot2d("x",0,1,style="--"); ...
>plot2d(sort(medianexpand(x,400)),sort(y),>points,color=red,style="x",>add):
```

Regression and Correlation

Linear regression can be done with the functions polyfit() or various fit functions.
For a start we find the regression line for univariate data with 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

We want to compare non-weighted and weighted fits. First the coefficients of the linear fit.

```
>p=polyfit(x,y,1)
```

```
[0.733333,  0.812121]
```

Now the coefficients with a weight that emphasizes the last values.

```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

```
 [4.71566,  0.38319]
```

We put everything into one plot for the points and the regression lines, and for the weights used.

```
>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):
```

For another example we read a survey of students, their ages, the ages of their parents and the number of siblings from a file.
This table contains "m" and "f" in the second column. We use the variable tok2 to set the proper translations instead of letting readtable() collect the translations.

```
>{MS,hd}:=readtable("table1.dat",tok2:=["m","f"]);  ...
>writetable(MS,labc=hd,tok2:=["m","f"]);
```

Person	Sex	Age	Mother	Father	Siblings
1	m	29	58	61	1
2	f	26	53	54	2
3	m	24	49	55	1
4	f	25	56	63	3
5	f	25	49	53	0
6	f	23	55	55	2
7	m	23	48	54	2
8	m	27	56	58	1
9	m	25	57	59	1
10	m	24	50	54	1
11	f	26	61	65	1
12	m	24	50	52	1
13	m	29	54	56	1
14	m	28	48	51	2
15	f	23	52	52	1
16	m	24	45	57	1
17	f	24	59	63	0
18	f	23	52	55	1
19	m	24	54	61	2
20	f	23	54	55	1

How do the ages depend on each other? A first impression comes from a pairwise scatterplot.

```
>scatterplots(tablecol(MS,3:5),hd[3:5]):
```

It is clear that the age of the father and mother depend on each other. Let us determine and plot the regression line.

```
>cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1)
```

```
[17.3789,  0.740964]
```

This is obviously the wrong model. The regression line would be s=17+0.74t, where t is the age of the mother and s the age of the father. The age difference may depend a little bit on the age, but not that much.
Rather, we suspect a function like s=a+t. Then a is the mean of the s-t. It is the average age difference between fathers and mothers.

```
>da:=mean(cs[2]-cs[1])
```

```
3.65
```

Let us plot this into one scatter plot.

```
>plot2d(cs[1],cs[2],>points);  ...
>plot2d("evalpoly(x,ps)",color=red,style=".",>add);  ...
>plot2d("x+da",color=blue,>add):
```

Here is a box plot of the two ages. This only shows, that the ages are different.

```
>boxplot(cs,["mothers","fathers"]):
```

It is interesting that the difference in medians is not as large as the difference in means.

```
>median(cs[2])-median(cs[1])
```

```
1.5
```

The correlation coefficient suggests a positive correlation.

```
>correl(cs[1],cs[2])
```

```
0.7588307236
```

The correlation of the ranks is a measure for the same order in both vectors. It is also quite positive.

```
>rankcorrel(cs[1],cs[2])
```

```
0.758925292358
```

Creating new Functions

Of course, the EMT language can be used to program new functions. E.g., we define the skewness function.

$$\text{sk}(x) = \frac{\sqrt{n}\sum_i (x_i - m)^3}{\left(\sum_i (x_i - m)^2\right)^{3/2}}$$

where m is the mean of x.

```
>function skew (x:vector) ...
```

```
  m=mean(x);
  return sqrt(cols(x))*sum((x-m)^3)/(sum((x-m)^2))^(3/2);
  endfunction
```

As you see, we can easily use the matrix language to get a very short and efficient implementation. Let us try this function.

```
>data=normal(20); skew(normal(10))
```

```
  -0.442260891308
```

Here is another function, called the Pearson skewness coefficient.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)
>skew1(data)
```

```
  -0.0512737797453
```

Monte Carlo Simulation

Euler can be used to simulate random events. We have already seen simple examples above. Here is another one, which simulates 1000 times 3 dice throws, and asks for the distribution of the sums.

```
>ds:=sum(intrandom(1000,3,6))';  fs=getmultiplicities(3:18,ds)
```

```
  [2,   13,   33,   44,   61,   109,   126,   125,   118,   114,   92,   76,   41,   36,   9,   1]
```

We can plot this now.

```
>columnsplot(fs,lab=3:18):
```

To determine the expected distribution is not so easy. We use an advanced recursion for this.
The following function counts the number of ways the number k can be represented as the sum of n numbers in the range of 1 to m. It works recursively in an obvious way.

```
>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
```

Here is the result for three throws of dices.

```
>countways(5:25,5,5)
```

```
[1,  5,  15,  35,  70,  121,  185,  255,  320,  365,  381,  365,  320,  255,  185,  121,
```

```
>cw=countways(3:18,3,6)
```

```
[1,  3,  6,  10,  15,  21,  25,  27,  27,  25,  21,  15,  10,  6,  3,  1]
```

We add the expected values to the plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```

For another simulation, the deviation of the mean value of n 0-1-normal distributed random variables is 1/sqrt(n).

```
>longformat; 1/sqrt(10)
```

```
0.316227766017
```

Let us check this with a simulation. We produce 10000 times 10 random vectors.

```
>M=normal(10000,10); dev(mean(M)')
```

```
0.312808356044
```

```
>plot2d(mean(M)',>distribution):
```

The median of 10 0-1-normal distributed random numbers has a larger deviation.

```
>dev(median(M)')
```

```
0.365189017877
```

Since we can easily generate random walks, we can simulate the Wiener process. We take 1000 steps of 1000 processes. We then plot the standard deviation and the mean of the n-th step of these processes together with the expected values in red.

```
>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):
```

Tests

Tests are an important tool in statistics. In Euler, many tests are implemented. All of these tests returns the error that we accept if we reject the zero hypothesis.
For an example, we test dice throws for uniform distribution. At 600 throws, we got the following values, which we plug into the chi-square test.

```
>chitest([90,103,114,101,103,89],dup(100,6)')
```

```
0.498830517952
```

The chi-square test also has a mode, which uses a Monte Carlo simulation to test the statistics. The result should be almost the same. The parameter >p interprets the y-vector as a vector of probabilities.

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

```
0.529
```

This error is much too large. So we cannot reject uniform distribution. This does not prove that our dice was fair. But we cannot reject our hypothesis.
Next we generate 1000 dice throws using the random number generator, and do the same test.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

```
0.471366391089
```

Let us test for the mean value 100 with the t-test.

```
>s=200+normal([1,100])*10; ...
>ttest(mean(s),dev(s),100,200)
```

```
0.226642848601
```

The function ttest() needs the mean value, the deviation, the number of data, and the mean value to test for.
Now let us check two measurements for the same mean. We reject the hypothesis that they have the same mean, if the result is <0.05.

```
>tcomparedata(normal(1,10),normal(1,10))
```

```
0.122700802664
```

If we add a bias to one distribution, we get more rejections. Repeat this simulation several times to see the effect.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

```
6.05989394731e-06
```

In the next example, we generate 20 random dice throws 100 times and count the ones in it. There must be 20/6=3.3 ones on average.

```
>R=random(100,20); R=sum(R*6<=1)'; mean(R)
```

```
3.39
```

We now compare the number of ones with the binomial distribution. First we plot the distribution of ones.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\/"):
>t=count(R,21);
```

Then we compute the expected values.

```
>n=0:20; b=bin(20,n)*(1/6)^n*(5/6)^(20-n)*100;
```

We have to collect several numbers to get categories, which are big enough.

```
>t1=sum(t[1:2])|t[3:7]|sum(t[8:21]); ...
>b1=sum(b[1:2])|b[3:7]|sum(b[8:21]);
```

The chi-square test rejects the hypothesis that our distribution is a binomial distribution, if its result is <0.05.

```
>chitest(t1,b1)
```

```
0.216232784449
```

The following example contains results of two groups of persons (male and female, say) voting for one out of six parties.

```
>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
```

We wish to test for independence of the votes from the sex. The chi^2 table test does this. The result is way to large to reject independence. So we cannot say, if the voting depends on the sex from these data.

```
>tabletest(A)
```

```
0.990701632326
```

The following is the expected table, if we assume the observed frequencies of voting.

```
>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
```

We can compute the corrected contingency coefficient. Since it very close to 0, we conclude that the voting does not depend on the sex.

```
>contingency(A)
```

```
0.0427225484717
```

Some More Tests

Next we use a variance analysis (F-test) to test three samples of normally distributed data for same mean value. The method is called ANOVA (analysis of variance). In Euler, the function varanalysis() is used.

```
>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
```

This means, we reject the hypothesis of same mean value. We do this with an error probability of 1.3%.
There is also the median test, which rejects data samples with different mean distribution testing the median of the united sample.

```
>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
```

Another test on equality is the rank test. It is much sharper than the median test.

```
>ranktest(a,b)
```

```
0.00199969612469
```

In the following example, both distributions have the same mean.

```
>ranktest(random(1,100),random(1,50)*3-1)
```

    0.153721132329

Let us now try to simulate two treatments a and b applied to different persons.

```
>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];
```

The signum test decides, if a is better than b.

```
>signtest(a,b)
```

    0.0546875

This is too much of an error. We cannot reject that a is as good as b.
The Wilcoxon test is sharper than this test, but relies on the quantitative value of the differences.

```
>wilcoxon(a,b)
```

    0.0296680599405

Let us try two more tests using generated series.

```
>wilcoxon(normal(1,20),normal(1,20)-1)
```

    0.000445901348481

```
>wilcoxon(normal(1,20),normal(1,20))
```

    0.814869364774

Random Numbers

The following is a test for the random number generator. Euler uses a very good generator, so we need not expect any problems.
First we generate ten millions of random numbers in [0,1].

```
>n:=10000000; r:=random(1,n);
```

Next we count the distances between two numbers less than 0.05.

```
>a:=0.05; d:=differences(nonzeros(r<a));
```

Finally, we plot the number of times, each distance occurred, and compare with the expected value.

```
>m=getmultiplicities(1:100,d); plot2d(m); ...
>  plot2d("n*(1-a)^(x-1)*a^2",color=red,>add):
```

Clear the data.

```
>remvalue n;
```

Introduction for Users of the R Project

Clearly, EMT is not competing with R as a statistical package. However, there are many statistical procedures and functions available in EMT too. So EMT may satisfy the basic needs. After all, EMT comes with numerical packages and a computer algebra system.
This notebook is for you if you are familiar with R, but need to know the differences of the syntax of EMT and R. We try to give an overview of obvious and less obvious things you need to know.
Moreover, we look at ways to exchange data between the two systems.
Note that this is a work in progress.

Basic Syntax

The first thing you learn in R is to make a vector. In EMT, the main difference is that the : operator can take a step size. Moreover it has a low binding power.

```
>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
```

The c() function does not exist. It is possible to use vectors to concatenate things.
The following example is, like many others, from the "Interoduction to R" that comes with the R project. If you read this PDF, you will find that I follow its path in this tutorial.

```
>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]
```

The colon operator with step size of EMT is replaced by the function seq() in R. We can write this function in 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]
```

The function rep() of R is not present in EMT. For vector input, it could be written as follows.

```
>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]
```

Note that "=" or ":=" is used for assignments. The "->" operator is used for units in EMT.

```
>125km -> " miles"
```

```
77.6713990297 miles
```

The "<-" operator for assignment is misleading anyway, and not a good idea of R. The following will compare a and -4 in EMT.

```
>a=2; a<-4
```

```
0
```

In R, "a<-4<3" works, but "a<-4<-3" does not. I had similar ambiguities in EMT too, but tried to eliminate them by and by.

EMT and R have vectors of boolean type. But in EMT, the numbers 0 and 1 are used to represent false and true. In R, the values true and false can nevertheless used in ordinary arithmetic just like in EMT.

```
>x<5, %*x
```

```
[0,  0,  1,  0,  0]
[0,  0,  3.1,  0,  0]
```

EMT throws errors or yields NAN depending on the flag "errors".

```
>errors off; 0/0, isNAN(sqrt(-1)), errors on;
```

```
NAN
1
```

Strings are the same in R and EMT. Both are in the current locale, not in Unicode.

In R there are packages for Unicode. In EMT, a string can be Unicode string. A unicode string can be translated to the local encoding and vice versa. Moreover, u"..." can contain HTML entities.

```
>u"&#169; Ren&eacut; Grothmann"
```

```
© René Grothmann
```

The following may or may not display correctly on your system as A with dot and dash above it. It depends on the font you are using.

```
>chartoutf([480])
```

The string concatenation is done with "+" or "|". It can include numbers, which will print in the current format.

```
>"pi = "+pi
```

```
pi = 3.14159265359
```

Indexing

Most of the time, this will work as in R.
But EMT will interpret negative indices from the back of the vector, while R interprets x[n] as x without the n-th elements.

```
>x, x[1:3], x[-2]
```

```
[10.4,   5.6,   3.1,   6.4,   21.7]
[10.4,   5.6,   3.1]
6.4
```

The behavior of R can be achieved in EMT with drop().

```
>drop(x,2)
```

```
[10.4,   3.1,   6.4,   21.7]
```

Logical vectors are not treated differently as index in EMT, in contrast to R. You need to extract the non-zero elements first in 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]
```

Just as in R, the index vector can contain repetitions.

```
>x[[1,2,2,1]]
```

```
[10.4,   5.6,   5.6,   10.4]
```

But names for indices are not possible in EMT. For a statistical package, this may often be necessary to ease access to elements of vectors.
To mimic this behavior, we can define a function as the following.

```
>function sel (v,i,s) := v[indexof(s,i)]; ...
>s=["first","second","third","fourth"]; sel(x,["first","third"],s)
```

```
[10.4,   3.1]
```

Data Types

EMT has more fixed data types than R. Obviously, in R there exist growing vectors. You can set an empty numerical vector v and assign a value to the element v[17]. This is not possible in EMT.
The following is a bit inefficient.

```
>v=[]; for i=1 to 10000; v=v|i; end;
```

EMT will now construct a vector with v and i appended on the stack and copy that vector back to the global variable v.
The more efficient pre-defines the vector.

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

To change date types in EMT, you can use functions like complex().

```
>complex(1:4)
```

```
[ 1+0i ,   2+0i ,   3+0i ,   4+0i   ]
```

Conversions to strings is possible for elementary data types only. The current format is used for simple string concatenation. But there are functions like print() or frac().
For vectors, you can easily write your own function.

```
>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]
```

For communication with Maxima, there exists a function convertmxm(), which can also be used to format a vector for output.

```
>convertmxm(1:10)
```

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

For Latex the tex command can be used to obtain the Latex command.

```
>tex(&[1,2,3])
```

```
\left[ 1 , 2 , 3 \right]
```

Factors and Tables

In the introduction to R there is an example with so called factors.
The following is a list of the territories of 30 states.

```
>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"];
```

Assume, we have corresponding incomes in each state.

```
>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];
```

Now, we want to compute the mean of incomes in the territories. Being a statistical program, R has factor() and tappy() for this.
EMT can make this by finding the index of territories in the unique list of territories.

```
>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,
5,  1,  2,  7,  7,  1]
```

At that point, we can write our own loop function to do things for one factor only.
Or we can mimic the tapply() function in the following way.

```
>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
```

It is a bit inefficient, since it computes The unique territories for each i, but it works.

```
>tappl(auterr,"mean",austates,incomes)
```

```
[44.5,  57.3333333333,  55.5,  53.6,  55,  60.5,  56,  52.25]
```

Note that it works for each vector of territories.

```
>tappl(["act","nsw"],"mean",austates,incomes)
```

```
[44.5,  57.3333333333]
```

Now, the statistical package of EMT defines tables just as in R. The functions readtable() and writetable() can be used for input and output.
So we can print the average state income in the territories in a friendly way.

```
>writetable(tappl(auterr,"mean",austates,incomes),labc=auterr,wc=7)
```

```
    act    nsw     nt    qld     sa    tas    vic     wa
   44.5  57.33   55.5   53.6     55   60.5     56  52.25
```

We can also try to mimic the behavior of R completely.
The factors should clearly be kept in a collection with the types and the categories (states and territories in our example). For EMT, we add the pre-computed indices.

```
>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);
```

Now the third element of the collection will contain the indices.

```
>statef[3]
```

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

Now we can mimic tapply() in the following way. It will return a table as a collection of table data and column headings.

```
>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
```

We did not add much type checking here. The only precaution concerns categories (factors) with no data. But one should check for the correct length of t and for the correctness of the collection tf.
This table can be printed as a table with 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
```

Arrays

EMT has only two dimensions for arrays. The data type is called a matrix. It would be easy to write functions for higher dimensions or a C library for this, however.
R has more than two dimensions. In R the array is a vector with a dimension field.
In EMT, a vector is a matrix with one row. It can be made into a matrix with 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
```

Extraction of rows and columns, or sub-matrices, is much like in R.

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

```
         2          3
         7          8
        12         13
        17         18
```

However, in R it is possible to set a list of specific indices of the vector to a value. The same is possible in EMT only with a loop.

```
>function setmatrixvalue (M, i, j, v) ...
```

```
  loop 1 to max(length(i),length(j),length(v))
     M[i{#},j{#}] = v{#};
  end;
  endfunction
```

We demonstrate this to show that matrices are passed by reference in EMT. If you do not want to change the original matrix M, you need to copy it in the function.

```
>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
```

The outer product in EMT can only be done between vectors. It is automatic due to the matrix language. One vector needs to be a column vector and the other a row vector.

```
>(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
```

In the introduction PDF for R there is an example, which computes the distribution of ab-cd for a,b,c,d chosen from 0 to n randomly. The solution in R is form a 4-dimensional matrix and run table() over it.
Of course, this can be achieved with a loop. But loops are not effective in EMT or R. In EMT, we could write the loop in C and that would be the quickest solution.
But we want to mimic the behavior of R. For this, we need to flatten the multiplications ab and make a matrix of 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"):
```

Besides the exact multiplicities, EMT can compute frequencies in vectors.

```
>getfrequencies(q,-50:10:50)
```

```
[0,   23,   132,   316,   602,   801,   333,   141,   53,   0]
```

The most easy way to plot this as a distribution is the following.

```
>plot2d(q,distribution=11):
```

But it is also possible to pre-compute the count in chosen intervals beforehand. Of course, the following uses getfrequencies() internally.
Since the histo() function returns frequencies, we need to scale these so that the integral under the bar graph is 1.

```
>{x,y}=histo(q,v=-55:10:55); y=y/sum(y)/differences(x); ...
>plot2d(x,y,>bar,style="/"):
```

Lists

EMT has two sorts of lists. One is a global list which is mutable, and the other is a list type which is immutable. We do not care about global lists here.
The immutable list type is called a collection in EMT. It behaves like a structure in C, but the elements are just numbered and not named.

```
>L={{"Fred","Flintstone",40,[1990,1992]}}
```

```
Fred
Flintstone
40
[1990,  1992]
```

Currently the elements do not have names, though names can be set for special purposes. They are accessed by numbers.

```
>(L[4])[2]
```

```
1992
```

File Input and Output (Reading and Writing Data)

You will often want to import a matrix of data from other sources to EMT. This tutorial tells you about the many ways to achieve this. Simple functions are writematrix() and readmatrix().
Let us demonstrate how to read and write a vector of reals to a file.

```
>a=random(1,100); mean(a), dev(a),
```

```
0.55269
0.29344
```

To write the data to a file, we use the function writematrix().
Since this introduction is most likely in a directory, where the user has no write access, we write the data to the user home directory. For own notebooks, this is not necessary, since the data file will be written into the same directory.

```
>filename="test.dat";
```

Now we write the column vector a' to the file. This yields one number in each line of the file.

```
>writematrix(a',filename);
```

To read the data, we use readmatrix().

```
>a=readmatrix(filename)';
```

And remove the file.

```
>fileremove(filename);
>mean(a), dev(a),
```

```
0.55269
0.29344
```

The functions writematrix() or writetable() can be configured for other languages.
E.g., if you have an Indonesian system (decimal point with comma), your Excel needs values with decimal commas separated by semicolons in a csv (the default is comma separated values) file. The following file "test.csv" should appear on your cuurent folder.

```
>filename="test.csv"; ...
>writematrix(random(5,3),file=filename,separator=",");
```

You can now open this file with an Indonesian Excel directly.

```
>fileremove(filename);
```

Sometimes we have strings with tokens like the following.

```
>s1:="f m m f m m m f f f m m f";  ...
>s2:="f f f m m f f";
```

To tokenize this, we define a vector of tokens.

```
>tok:=["f","m"]
```

```
f
m
```

Then we can count the number of times each token appears in the string, and put the result into a table.

```
>M:=getmultiplicities(tok,strtokens(s1))_ ...
>  getmultiplicities(tok,strtokens(s2));
```

Write the table with the token headers.

```
>writetable(M,labc=tok,labr=1:2,wc=8)
```

```
                f       m
       1        6       7
       2        5       2
```

For statics, EMT can read and write tables.

```
>file="test.dat"; open(file,"w"); ...
>writeln("A,B,C"); writematrix(random(3,3)); ...
>close();
```

The file looks like this.

```
>printfile(file)
```

```
A,B,C
0.870270022178512,0.02481087162574944,0.466040156601655
0.3026325231063014,0.6220889584522515,0.9367082489748491
0.7466896392724567,0.7368844620115959,0.3419781863075429
```

The function readtable() in its simplest form can read this and return a collection of values and heading lines.

```
>L=readtable(file,>list);
```

This collection can be printed with writetable() to the notebook, or to a file.

```
>writetable(L,wc=10,dc=5)
```

```
         A         B         C
   0.87027   0.02481   0.46604
   0.30263   0.62209   0.93671
   0.74669   0.73688   0.34198
```

The matrix of values is the first element of L. Note that mean() in EMT computes the mean values of the rows of a matrix.

```
>mean(L[1])
```

```
0.45371
0.62048
0.60852
```

CSV Files

First, let us write a matrix into a file. For the output, we generate a file in the current working directory.

```
>file="test.csv";  ...
>M=random(3,3); writematrix(M,file);
```

Here is the content of this file.

```
>printfile(file)
```

```
0.8644485191805866,0.823403481389068,0.790241662026635
0.5906186300341895,0.1676904904906134,0.6263113968244143
0.2579517401409792,0.8455384857350826,0.721718522415531
```

This CVS can be opened on English systems into Excel by a double click. If you get such a file on a German system, you need to import the data into Excel taking care of the decimal dot.
But the decimal dot is the default format for EMT too. You can read a matrix from a file with readmatrix().

```
>readmatrix(file)
```

```
0.86445    0.8234   0.79024
0.59062   0.16769   0.62631
0.25795   0.84554   0.72172
```

It is possible to write several matrices to one file. The open() command can open a file for writing with the "w" parameter. The default is "r" for reading.

```
>open(file,"w"); writematrix(M); writematrix(M'); close();
```

The matrices are separated by a blank line. To read the matrices, open the file and call readmatrix() several times.

```
>open(file); A=readmatrix(); B=readmatrix(); A==B, close();
```

```
1          0          0
0          1          0
0          0          1
```

In Excel or similar spreadsheets, you can export a matrix as CSV (comma separated values). In Excel 2007, use "save as" and "other formats", then select "CSV". Make sure, the current table contains only data you wish to export.
Here is an example.

```
>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
```

As you can see, my German system has used a semicolon as separator and a decimal comma. You can can change this in the system settings or in Excel, but it is not necessary for reading the matrix into EMT.
The easiest way to read this into Euler is readmatrix(). All commas are replaced by dots with the parameter >comma. For English CSV, simply omit this parameter.

```
>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
```

Let us plot this.

```
>plot2d(M'[1],M'[2:3],>points,color=[red,green]'):
```

There are more elementary ways to read data from a file. You can open the file and read the numbers line by line. The function getvectorline() will read numbers from a line of data. By default, it expects a decimal dot. But it can also use a decimal comma, if you call setdecimaldot(",") before you use this function.
The following function is an example for this. It will stop at the end of the file or an empty line.

```
>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)
```

```
    0.86445    0.8234   0.79024
    0.59062   0.16769   0.62631
    0.25795   0.84554   0.72172
```

It would also be possible to read all numbers in that file with getvector().

```
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
```

```
    0.86445    0.8234   0.79024
    0.59062   0.16769   0.62631
    0.25795   0.84554   0.72172
```

Thus it is very easy to save a vector of values, one value in each line and read back this vector.

```
>v=random(1000); mean(v)
```

```
  0.51267
```

```
>writematrix(v',file); mean(readmatrix(file)')
```

```
  0.51267
```

Using Tables

Tables can be used to read or write numerical data. For an example, we write a table with row and column headers to a file.

```
>file="test.tab"; M=random(3,3);  ...
>open(file,"w");  ...
>writetable(M,separator=",",labc=["one","two","three"]);  ...
>close(); ...
>printfile(file)
```

```
  one,two,three
        0.91,       0.19,       0.01
         0.9,       0.05,       0.94
        0.16,       0.81,       0.37
```

This can be imported into Excel.
To read the file in EMT, we use readtable().

```
>{M,headings}=readtable(file,>clabs); ...
>writetable(M,labc=headings)
```

```
       one       two     three
      0.91      0.19      0.01
       0.9      0.05      0.94
      0.16      0.81      0.37
```

Analyzing a Line

You could even evaluate each line by hand. Suppose, we have a line of the following format.

```
>line="2020-11-03,Tue,1'114.05"
```

```
  2020-11-03,Tue,1'114.05
```

First we can tokenize the line.

```
>vt=strtokens(line)
```

```
  2020-11-03
  Tue
  1'114.05
```

Then we can evaluate each element of the line using appropriate evaluations.

```
>day(vt[1]),  ...
>indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])),  ...
>strrepl(vt[3],"'","")()
```

```
  7.3816e+05
  2
  1114
```

Using regular expressions, it is possible to extract almost any information from a line of data. Assume we have the following line an HTML document.

```
>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>
```

To extract this, we use a regular expression, which searches for

```
 - a closing bracket >,
 - any string not containing brackets with a sub-match "(...)",
 - an opening and a closing bracket using the shortest solution,
 - again any string not containing brackets,
 - and an opening bracket <.
```

Regular expressions are somewhat difficult to learn but very powerful.

```
>{pos,s,vt}=strxfind(line,">([^<>]+)<.+?>([^<>]+)<");
```

The result is the position of the match, the matched string, and a vector of strings for sub-matches.

```
>for k=1:length(vt); vt[k](), end;
```

```
1145.5
5.6
```

Here is a function, which reads all numerical items between <td> and </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
```

Reading from the Web

A web site or a file with an URL can be opened in EMT and can be read line by line.
In the example, we read the current version from the EMT site. We use regular expression to scan for "Version ..." in a 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 2021-04-30
```

Input and Output of Variables

You can write a variable in the form of an Euler definition to a file or to the command line.

```
>writevar(pi,"mypi");
```

```
mypi = 3.141592653589793;
```

For a test, we generate an Euler file in the work directory of EMT.

```
>file="test.e"; ...
>writevar(random(2,2),"M",file); ...
>printfile(file,3)
```

```
M = [ ..
0.106576066957305, 0.3792882958611032;
0.7685505432667192, 0.8279275951785248];
```

We can now load the file. It will define the matrix M.

```
>load(file); show M,
```

```
M =
  0.10658   0.37929
  0.76855   0.82793
```

By the way, if writevar() is used on a variable, it will print the variable definition with the name of this variable.

```
>writevar(M); writevar(inch$)
```

```
M = [ ..
0.106576066957305, 0.3792882958611032;
0.7685505432667192, 0.8279275951785248];
inch$ = 0.0254;
```

We can also open a new file or append to an existing file. In the example we append to the previously generated file.

```
>open(file,"a"); ...
>writevar(random(2,2),"M1"); ...
>writevar(random(3,1),"M2"); ...
>close();
>load(file); show M1; show M2;
```

```
M1 =
  0.87558   0.40272
 0.010018   0.99217
M2 =
  0.66841
  0.91819
   0.3323
```

To remove any files use fileremove().

```
>fileremove(file);
```

A row vector in a file does not need commas, if each number is in a new line. Let us generate such a file, writing every line one by one with writeln().

```
>open(file,"w"); writeln("M = ["); ...
>for i=1 to 5; writeln(""+random()); end; ...
>writeln("];"); close(); ...
>printfile(file)
```

```
M = [
0.699685893708
0.56495656726
0.566012751828
0.913964584324
0.75449923013
];
```

```
>load(file); M
```

```
[0.69969,  0.56496,  0.56601,  0.91396,  0.7545]
```