



Logo UNY.png

Disusun Oleh: Alya Putri Medilasari 23030630018 Matematika B 2023

Kata Pengantar

Puji syukur kami ucapkan kehadiran Allah SWT atas segala rahmat-Nya sehingga makalah ini dapat tersusun sampai dengan selesai. Tidak lupa kami mengucapkan terima kasih terhadap bantuan dari pihak yang telah berkontribusi dengan memberikan sumbangan baik pikiran maupun materinya

Penulis sangat berharap semoga makalah ini dapat menambah pengetahuan dan pengalaman bagi pembaca. Bahkan kami berharap lebih jauh lagi agar makalah ini bisa pembaca praktekan dalam kehidupan sehari-hari.

Bagi kami sebagai penyusun merasa bahwa masih banyak kekurangan dalam penyusunan makalah ini karena keterbatasan pengetahuan dan pengalaman Kami. Untuk itu kami sangat mengharapkan kritik dan saran yang membangun dari pembaca demi kesempurnaan makalah ini.

Yogyakarta, 29 November 2024

Penyusun Alya Putri Medilasari

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:

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

Contoh pertama

Menyederhanakan bentuk aljabar:

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

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

Menjabarkan:

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

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

Baris Perintah

Baris perintah Euler terdiri dari satu atau beberapa perintah Euler 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 long command 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.

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

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

A line starting with %% will be completely invisible.

```
81
```

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

```
>x=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; // tambahkan komentar disini seblum ...
> repeat xnew:=(x+2/x)/2; until xnew~=x; ...
> x := xnew; ...
> end; ...
> x,
```

```
1.41421356237
```

Struktur bersyarat juga berfungsi.

```
>if E<pi 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. Dan juga, perhatikan baris status. Setelah kurung buka fungsi sqrt(), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol kembali.

```
>sqrt(sin(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 juga dapat menyalin dan menempel di Euler. Gunakan Ctrl-C dan Ctrl-V untuk ini. Untuk menandai teks, seret mouse atau gunakan shift bersamaan dengan tombol kursor. Selain itu, Anda dapat menyalin tanda kurung yang disorot.

Sintaks Dasar

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

Untuk menyetel variabel, gunakan “=” atau “:=”. Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir/terbaru. Spasi tidak menjadi masalah. Tetapi spasi di antara perintah sangat 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. Untuk memasukkan

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

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

```
>E^2*(1/(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 diperlukan, hal tersebut harus berisi tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, memasang braket atau tanda kurung adalah ide yang bagus. Perhatikan bahwa EMT menunjukkan tanda kurung buka dan tutup saat mengedit baris perintah.

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

```
14.4978445072
```

Operator numerik Euler meliputi

1. ◦ unary or operator plus
2. ◦ unary or operator minus
3. ◦ operator perkalian
4. / operator pecahan
5. . produk matriks
6. a^b pangkat untuk a positif atau bilangan bulat b (a**b juga berfungsi)
7. n! operator faktorial
8. 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,floor,ceil,round,abs,sign
conj,re,im,arg,conj,real,complex
beta,betai,gamma,complexgamma,ellrf,ellf,ellrd,elle
bitand,bitor,bitxor,bitnot

Beberapa perintah memiliki alias, misalnya ln untuk log.

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

```
2  
0.5
```

```
>sin(30°)
```

```
0.5
```

Pastikan untuk menggunakan tanda kurung (kurung 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.

[illegible]

Strings

Fungsi `strtochr()` akan mengenali string Unicode, dan menerjemahkannya dengan benar.

```
>v=strtochar(u"Ä is a German letter")
```

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

Perintah ini menghasilkan array atau daftar angka berupa vektor angka yang mewakili karakter dalam string dalam bentuk kode Unicode.

Fungsi kebalikannya adalah `chartoutf()`.

```
>v[1]=strtochar(u"Ü")[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  $\alpha=\beta$ .
```

Memungkinkan juga untuk menggunakan entitas numerik.

```
>u"Ähnliches"
```

```
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
```

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

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

```
1
```

Operator Boolean mematuhi aturan bahasa matriks

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

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

Anda dapat menggunakan fungsi `nonzeros()` untuk mengekstrak elemen tertentu dari sebuah vektor. Pada contoh, kita menggunakan kondisional `isprime(n)`.

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

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

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

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

Output Formats

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

```
>defformat; pi
```

```
3.14159265359
```

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

```
>longest pi
```

```
3.141592653589793
```

Berikut ini adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhe(pi)
```

0

Expressions

Strings atau names 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 menamainya “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 tersebut. Parameter tambahan dapat ditambahkan dengan menggunakan parameter yang ditetapkan.

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

```
-0.919535764538
```

Perhatikan bahwa ekspresi akan selalu menggunakan variabel global, meskipun 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” selain nilai global, Anda perlu menambahkan “at=value”.

```
>at:=4; function f(expr,x,a) := expr(x,at=a); ...
```

```
> f("at*x^2",3,5)
```

```
45
```

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

```
>at:=4; function f(expr,x) := expr(x); ...
```

```
> f({{"at*x^2",at=5}},3)
```

```
45
```

Ekspresi dalam x sering digunakan seperti halnya fungsi.

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

```
>f &= 5*x;
```

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

```
>f(2)
```

```
12
```

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

```
>fx &= diff(x^x,x); Misplaced &^x (log x + 1)$Bentuk khusus dari sebuah ekspresi memungkinkan variabel apa pun sebagai parameter tanpa nama untuk evaluasi ekspresi, bukan hanya “x”, “y”, dll. Untuk ini, mulailah ekspresi dengan “@(variabel)...”.
```

```
>“@(a,b) a^2+b^2”, %(4,5)
```

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

```
41
```

Hal ini memungkinkan untuk memanipulasi ekspresi dalam variabel lain untuk fungsi EMT yang membutuhkan ekspresi dalam “x”.

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

Seperti yang Anda lihat pada 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.

Symbolic Mathematics

EMT melakukan matematika simbolik dengan bantuan Maxima. Untuk detailnya, mulailah dengan tutorial berikut ini, atau telusuri referensi untuk Maxima. Para ahli dalam Maxima harus mencatat bahwa ada perbedaan dalam sintaks antara sintaks asli Maxima dan sintaks default ekspresi simbolik dalam EMT.

Matematika simbolik diintegrasikan dengan mulus ke dalam Euler dengan &. Setiap ekspresi yang dimulai dengan & adalah ekspresi simbolik. Ekspresi ini dievaluasi dan dicetak oleh Maxima.

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

```
>Misplaced &2658271574788448768043625811014615890319638528000000000
```

Dengan caraini, kitadapatmenghitunghasilnyangbesardengantepat. Marikitahitung!

```
C(44, 10) =  $\frac{44!}{34!10!}$  & 44!/(34!*10!) // nilai C(44,10)
```

```
2481256778
```

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

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

2481256778\$Untuk mempelajari lebih lanjut tentang fungsi tertentu klik dua kali di atasnya. Misalnya, coba klik dua kali pada “&binomial” di baris perintah sebelumnya. Ini membuka dokumentasi Maxima seperti yang disediakan oleh penulis program itu.

Anda akan belajar bahwa yang berikut ini juga berfungsi.

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

```
>binomial(x, 3) // C(x, 3)  $\frac{(x-2)(x-1)x}{6}$  $Jika Anda ingin mengganti x dengan nilai tertentu, gunakan “with”.
```

>Misplaced &20\$Dengan begitu, Anda dapat menggunakan solusi dari sebuah persamaan dalam persamaan lain.

Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Alasannya adalah sebuah bendera simbolik khusus dalam string. Seperti yang telah Anda lihat pada contoh sebelumnya dan contoh berikut, jika Anda telah menginstal LaTeX, Anda dapat mencetak ekspresi simbolik dengan Latex. Jika tidak, perintah berikut ini akan mengeluarkan pesan kesalahan. Untuk mencetak ekspresi simbolik dengan LaTeX, gunakan \$ di depan & (atau Anda dapat menghilangkan &) sebelum perintah. Jangan jalankan perintah Maxima dengan \$, jika Anda tidak memiliki 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 mengapit ekspresi dalam “...”. Menggunakan lebih dari satu ekspresi sederhana dimungkinkan, tetapi sangat tidak disarankan.
```

```
>&“v := 5; v^2”
```

25

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

```
>$&expand((1+x)^4), Misplaced &(x + 1)^3 $ Sekali lagi, % mengacu pada hasil sebelumnya.
```

Untuk mempermudah, kita menyimpan solusi ke dalam sebuah variabel simbolik. Variabel simbolik didefinisikan dengan “&=”.

```
>fx &= (x+1)/(x^4+1); Misplaced & $\frac{x+1}{x^4+1}$  $Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.
```

```
>Misplaced & $\frac{3x^4-4x^3+1}{(x^4+1)^2}$  $Masukan langsung dari perintah Maxima juga tersedia. Mulai baris perintah dengan “:”. Sintaks Maxima disesuaikan dengan sintaks EMT (disebut “mode kompatibilitas”).
```

>&factor(20!)

2432902008176640000

>::: factor(10!)

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

>::: factor(20!)

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

Jika Anda adalah seorang ahli dalam Maxima, Anda mungkin ingin menggunakan sintaks asli Maxima. Anda dapat melakukan ini dengan “:::”.

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

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

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

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

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

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

$$\begin{matrix} 5 \\ 125 & E \end{matrix}$$
$$125 e^5$$

18551.64488782208

>fx(5)

18551.6448878

Untuk mengevaluasi ekspresi dengan nilai variabel tertentu, Anda dapat menggunakan operator “with”. Baris perintah berikut ini juga mendemonstrasikan bahwa Maxima dapat mengevaluasi sebuah ekspresi secara numerik dengan float().

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

$$\begin{matrix} 10 & 5 \\ 1000 & E & - & 125 & E \end{matrix}$$

2.20079141499189e+7

>factor(diff(fx, x, 2))x (x^2 + 6x + 6) e^x\$Untuk mendapatkan kode Latex untuk sebuah ekspresi, Anda dapat menggunakan perintah tex.

>tex(fx)

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

Ekspresi simbolik dapat dievaluasi seperti halnya ekspresi numerik.

>fx(0.5)

0.206090158838

Dalam ekspresi simbolik, hal ini tidak dapat dilakukan, karena Maxima tidak mendukungnya. Sebagai gantinya, gunakan sintaks “with” (bentuk yang lebih baik dari perintah at(...) pada Maxima).

>Misplaced & $\frac{\sqrt{e}}{8}$ \$Penugasan ini juga bisa bersifat simbolis.

>\$&fx with x=1+t

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

Perintah “solve” menyelesaikan ekspresi simbolik untuk sebuah variabel di Maxima. Hasilnya adalah sebuah vektor solusi.

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

\$\$\$

Bandingkan dengan perintah “solve” numerik 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 simbolik. Euler akan membaca penugasan $x = \text{dst}$. Jika Anda tidak membutuhkan hasil numerik untuk perhitungan lebih lanjut, Anda juga bisa membiarkan Maxima menemukan nilai numeriknya

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

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

```
$$$
```

Untuk mendapatkan solusi simbolik yang spesifik, seseorang dapat menggunakan “with” dan indeks.

```
>$solve(x^2+x=1,x), x2 &= x with %[2]; Misplaced &  $\frac{\sqrt{5}-1}{2}$ 
```

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

```
>sol &= solve([x+y=3,x^2+y^2=5],[x,y]); $sol, Misplaced & Ekspresi simbolik dapat memiliki flags, yang menunjukkan perlakuan khusus di Maxima. Beberapa flag dapat digunakan sebagai perintah juga, namun ada juga yang tidak. Flags ditambahkan dengan “|” (bentuk yang lebih baik dari “ev(...,flags)”)
```

```
>Misplaced &  $\frac{3x^2}{x+1} - \frac{x^3-1}{(x+1)^2}$  & diff((x^3-1)/(x+1),x) | ratsimp //menyederhanakan pecahan
```

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

```
>Misplaced &  $\frac{2x^3+3x^2+1}{(x+1)^2}$ 
```

Fungsi

Di EMT fungsi adalah program yang didefinisikan dengan perintah “function”. Fungsi dapat menjadi fungsi satu baris atau fungsi multi baris. Fungsi satu baris dapat berupa numerik atau simbolis. Fungsi satu baris didefinisikan oleh “:=”.

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

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

```
>f(2)
```

```
4.472135955
```

Fungsi ini dapat digunakan juga dalam vektor, dengan mengikuti aturan bahasa matrik Euler, karena ekspresi yang digunakan dalam fungsi divektorkan. Kita akan mencobanya menggunakan fungsi f di atas.

```
>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 juga dapat menjadi plot, hanya dengan 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 built-in, Anda harus menambahkan kata kunci “overwrite”. Menimpa fungsi bawaan berbahaya dan dapat menyebabkan masalah bagi fungsi lain yang bergantung pada fungsi tersebut. Anda masih dapat memanggil fungsi bawaan sebagai “_...”, jika fungsi tersebut merupakan fungsi dalam inti Euler.

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

```
>sin(45)
```

```
0.707106781187
```

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

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

```
0.707106781187
```

Default Parameters

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

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

16

Menetapkannya akan menimpa nilai default.

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

80

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

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

16

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

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

24

Tetapi parameter yang ditetapkan akan menggantikan nilai global. Jika argumen tidak ada dalam daftar parameter yang telah ditetapkan sebelumnya, argumen tersebut harus dideklarasikan dengan “:=”!

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

20

Fungsi simbolik didefinisikan dengan “&=”. Fungsi-fungsi ini didefinisikan dalam Euler dan Maxima, dan dapat digunakan di kedua bahasa tersebut. Ekspresi pendefinisian dijalankan melalui Maxima sebelum definisi.

```
>function g(x) &= x^3-x*exp(-x); Misplaced &3 - x e-x Fungsi simbolik dapat digunakan dalam ekspresi simbolik
```

```
>$&diff(g(x),x), Misplaced &-4/3 + 16/3$
```

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

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

178.635099908

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

```
>function G(x) &= factor(integrate(g(x),x)); Misplaced &e-c (c4 ec + 4 c + 4)/4  
$>solve(&g(x),0.5)
```

0.703467422498

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

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

0.703467422498

```
>function P(x,n) &= (2*x-1)^n; Misplaced &2 x - 1n >function Q(x,n) &= (x+2)^n;  
Misplaced &x + 2n &P(x,4), Misplaced &6 x4 - 32 x3 + 24 x2 - 8 x + 1
```

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

625

```
>$&P(x,4)+Q(x,3), Misplaced &6 x4 - 31 x3 + 30 x2 + 4 x + 9$
```

```
>$&P(x,4)-Q(x,3), $&expand(%), Misplaced &6 x4 - 33 x3 + 18 x2 - 20 x - 7$
```

```
>$&P(x,4)*Q(x,3), $&expand(%), Misplaced &x + 23 (2 x - 1)4 $>$&P(x,4)/Q(x,1),  
$&expand(%), Misplaced &(2 x - 1)4/x + 2 $>function f(x) &= x^3-x; Misplaced &3 - x $Dengan  
&= fungsinya simbolis, dan dapat digunakan dalam ekspresi simbolik lainnya.  
Contohnya dalam integral tak tentu sebagai berikut.
```

$$>\text{Misplaced } \frac{x^4}{4} - \frac{x^2}{2}$$

Dengan $\int_1^x t^t dt$ fungsi numerik. Contoh yang baik adalah integral tak tentu seperti $f(x) = \int_1^x t^t dt$, yang tidak dapat dievaluasi secara simbolik.

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

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

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

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

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

```
2
```

Sering kali, kita ingin menggunakan fungsi untuk vektor di satu tempat, dan untuk elemen individual di tempat lain. Ini tepat terjadi dengan vektor parameter.

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

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

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

```
17
```

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

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

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

```
>$&realpart((x+I*y)^4), Misplaced &0$
```

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

```
>function f(x,y) &= factor(lapl((x+y^2)^5,x,y)); Misplaced &0 (y^2 + x)^3 (9 y^2 + x + 2)
$Ringkasan: 1. &= mendefinisikan fungsi simbolis, 2. := mendefinisikan fungsi numerik,
3. &= mendefinisikan fungsi simbolis murni
```

Memecahkan Ekspresi

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

```
>solve(“x^2-2”,1)
```

```
1.41421356237
```

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

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

```
$$$
```

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

```
$$$
```

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

```
$$$
```

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

```
$] $$
```

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

```
$$$
```

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

```
>longest sol()
```

```
-0.6180339887498949      1.618033988749895
```

Untuk menggunakan solusi secara simbolis dalam ekspresi lain, cara termudah adalah “with”.

```
>$&x^2 with sol[1], Misplaced &
```

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

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

```
$, , ] $$
```

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

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

dengan a=3. >function f(x,a) := x^{a-a}x;

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

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

```
2.54116291558
```

Ini juga bekerja dengan ekspresi. Tapi daftar elemen yang ada harus digunakan. (Lebih lanjut tentang daftar dalam tutorial tentang sintaks EMT).

```
>solve({{"xa-ax",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)
```

```
D:/Euler x64/maxima/share/maxima/5.35.1/share/fourier_elim/fo\
urier_elim.lisp
```

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

```
$$$
```

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

```
$$$
```

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

```
$$$
```

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

```
$$$
```

>Misplaced &emptyset &fourier_elim([minf < x, x < inf],[x]) // solusinya R

universalset

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

\$\$\$

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

\$\$\$

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

\$\$\$

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

\$\$\$

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

\$\$\$

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

\$\$\$

Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler. Vektor dan matriks dimasukkan dengan tanda kurung siku, elemen dipisahkan dengan koma, baris dipisahkan dengan titik koma.

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

1	2
3	4

Hasil kali matriks dilambangkan dengan sebuah titik.

>b=[3;4]

3
4

>b' // transpose b

[3, 4]

>inv(A) //inverse A

-2	1
1.5	-0.5

>A.b //perkalian matriks

11
25

>A.inv(A)

1	0
0	1

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

>A.A

7	10
15	22

>A^2 //perpangkatan elemen2 A

1	4
9	16

>A.A.A

37	54
81	118

>power(A,3) //perpangkatan matriks

37	54
81	118

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

1	1
1	1

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

0.333333	0.666667
0.75	1

>A\b // hasilkali invers A dan b, $A^{-1}b$

-2
2.5

>inv(A).b

-2
2.5

>A\A // $A^{-1}A$

1	0
0	1

>inv(A).A

1	0
0	1

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

1	4
9	16

Ini bukan hasil kali 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, maka operan tersebut akan diperluas dengan cara alami.

>2*A

2	4
6	8

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

>[1,2]*A

1	4
3	8

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

>A*[2,3]

2	6
6	12

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

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

1	2
1	2

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

1	4
3	8

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

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

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

Sekali lagi, ingatlal 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]
```

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

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

```
5
```

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

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

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

Dari vektor seperti itu, “nonzeros” memilih elemen bukan nol. Dalam hal ini, kita mendapatkan indeks semua elemen yang lebih besar dari 50.

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

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

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

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

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

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

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

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

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

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

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

```
112
```

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

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

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

Ini mengembalikan indeks elemen, yang bukan nol.

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

1	4
2	1
2	2
3	2

Indeks ini dapat digunakan untuk menetapkan elemen ke suatu nilai.

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

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

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

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

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

Dan dimungkinkan untuk mendapatkan elemen-elemen dalam vektor.

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

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

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

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

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

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

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

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

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

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

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

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

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

1	1
2	4
3	1

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

Fungsi Matriks Lainnya

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 matriks lain secara berdampingan, jika keduanya memiliki jumlah baris yang sama.

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

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

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

```
>A|1
```

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

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

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

1	2	3
1	2	3

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

1	1
2	2
3	3

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

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

1	1
2	4
3	9

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

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

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

yaitu menggunakan perintah length().

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

```
9
```

Ada banyak fungsi lain yang menghasilkan matriks.

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

1	1
1	1

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

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

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

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

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

0.66566	0.831835
0.977	0.544258

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

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

1	2	3
4	5	6
7	8	9

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

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

Mari kita coba

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

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

Fungsi multdup() menduplikasi elemen-elemen sebuah vektor.

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

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

Fungsi flipx() dan flipy() membalik urutan baris atau kolom dari sebuah matriks. Misalnya, fungsi flipx() membalik secara horizontal.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1	2	3
4	5	6
7	8	9

Sekarang kita bisa mengekstrak diagonal.

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

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

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

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

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

Vectorization

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

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

Dengan ini dan operator titik dua a:delta:b, vektor nilai fungsi dapat dihasilkan dengan mudah. Pada contoh berikut, kita menghasilkan vektor nilai $t[i]$ dengan jarak 0,1 dari -1 hingga 1. Kemudian kita menghasilkan vektor nilai dari fungsi.

$$s = t^3 - t$$

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

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

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

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

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

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

Perhatikan, bahwa ini sangat berbeda 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 1×1 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 dikali vektor b .

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

```
30
70
```

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

```
>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 1×1 , 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.

1. Fungsi yang diterapkan ke vektor atau matriks diterapkan ke setiap elemen.
2. Operator yang beroperasi pada dua matriks dengan ukuran yang sama diterapkan berpasangan ke elemen matriks.
3. 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 menduplikasi.

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 mengembangkannya 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, terdapat operator "|" dan "_".

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

1	2	3	4	5
1	1	1	1	1

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

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

```
6
```

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

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

Fungsi yang tidak berfungsi untuk input vektor harus “vectorized”. Ini dapat dicapai dengan kata kunci “map” 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 “map” membuat vektor fungsi. Fungsinya sekarang akan bekerja untuk vektor bilangan.

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

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

Sub-Matrices dan Matrix-Elements

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.

1	2	3
4	5	6

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

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

Kita juga dapat menetapkan nilai pada deretan A.

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

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

Kami bahkan dapat menetapkan 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 dilakukan untuk menyusun ulang vektor lain dengan cara yang sama. Mari kita mengacak 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]
```

Hal ini juga berlaku untuk vektor string.

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

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

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

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

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

Fungsi `unique` mengembalikan daftar terurut dari elemen unik sebuah vektor.

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

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

Hal ini juga berlaku untuk vektor string.

```
>unique(s)
```

```
a  
aa  
d  
e
```

Aljabar Linier

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

Untuk sistem linier $Ax=b$, dengan A adalah matriks koefisien, x adalah vektor solusi yang ingin kita cari dan b adalah vektor hasil yang diberikan. Anda dapat menggunakan algoritma Gauss, matriks invers atau kecocokan linier.

Operator A menggunakan versi algoritma Gauss.

Operator backslash digunakan untuk menyelesaikan sistem persamaan linier ini. Ketika menulis A perangkat lunak akan menghitung solusi yang memenuhi persamaan $Ax=b$. Operator ini secara otomatis menggunakan algoritma eliminasi Gauss atau metode numerik serupa untuk menemukan solusi.

```
>A=[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), Misplaced &(a-1)^2 (a+2)$
```

 /s/EMT%20untuk%20Perhitungan%20Aljabar_Isni%20Azizah%20Utami_23030630016-089.png

```
>Misplaced &\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} $dengan
```

1. `det(A)` menghitung determinan matriks A.
2. `factor(%)` memberikan faktor-faktor dari determinan.
3. `invert(A)` memberikan invers dari matriks A.

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

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

```
$$$
```

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

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

```
$, ] $$
```

Untuk mengekstrak vektor eigen tertentu perlu pengindeksan yang cermat.

```
>$&eigenvectors([a,1;1,a]), &%[2][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 “with”.

> **Misplaced &** $\frac{1}{5} \frac{4}{2}$) \$Akses ke baris matriks simbolik bekerja seperti halnya dengan matriks numerik.

> \$&A[1]

\$\$\$

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

> &A[1,1]:=t+1; **Misplaced &** $\frac{t+1}{b} \frac{a}{2}$) \$Terdapat fungsi-fungsi simbolik dalam

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

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

\$\$\$

> B &:= [1,2,3,4]; \$B, **Misplaced &** $\frac{-2}{2} \frac{1}{2}$) \$

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

> \$&invert(B)()

$\frac{-2}{1.5} \frac{1}{-0.5}$

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)

$\frac{1}{0} \frac{0}{1}$

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?))

\$,] \$\$

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]

$\frac{1}{4} \frac{3.14159}{5}$

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

> **Misplaced &** $\frac{1}{4} \frac{\frac{1146408}{364913}}{5}$) \$Untuk menghindarinya, ada fungsi “mxmset(variable)”.

> mxmset(A); \$&A

$\left(\frac{1}{4} \frac{3.141592653589793}{5} \right)$

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

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

1.414213562373095

Ketepatan angka floating point besar dapat diubah.

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

```
3.14159265358979323846264338327950288419716939937510582097494\
4592307816406286208998628034825342117068b0
```

Variabel numerik dapat digunakan dalam ekspresi simbolis apa pun menggunakan “(var?)”. Perhatikan bahwa ini hanya diperlukan, jika variabel telah didefinisikan dengan “:=” atau “=” sebagai variabel numerik.

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

```
- 5.424777960769379
```

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 kedua hasil tersebut, dengan dan tanpa pembulatan.

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

```
1271.61
1271.6071
```

Sekarang tidak ada rumus sederhana untuk tahun ke-n, dan kita harus mengulang selama bertahun-tahun. Euler 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], VKr[-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("onipay",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 nonzeros(VKR<0) mengembalikan vektor indeks i, di mana VKR[i]<0, dan min menghitung indeks minimal.

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

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

```
>{x,n}=iterate("onipay",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 onipay() kita secara simbolis.

```
>function op(K) &= K*q+R; Misplaced &R + q K $Kita sekarang dapat mengulangi ini.
```

>\$\&op(op(op(K))), **Misplaced &** $q^3 R + q^2 R + q R + R + q^4 K$

Kami melihat sebuah pola. Setelah n periode yang kita miliki

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

Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

>&sum(q^k,k,0,n-1); **Misplaced &** $\sum_{k=0}^{n-1} q^k = \frac{q^n - 1}{q - 1}$ Ini agak rumit. Jumlahnya dievaluasi dengan bendera "simsup" untuk menguranginya menjadi hasil bagi. Mari kita membuat fungsi untuk ini.

Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

>function fs(K,R,P,n) &= (1+P/100)^n*K + ((1+P/100)^n-1)/(P/100)*R; **Misplaced &**
 $\frac{100((\frac{P}{100}+1)^n-1)R}{P} + K(\frac{P}{100}+1)^n$ 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.

fungsi untuk ini. Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

>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; **Misplaced &** $\frac{100((\frac{P}{100}+1)^n-1)R}{P} + K(\frac{P}{100}+1)^n = Kn$

Biasanyarumusinidiberikandalambentuk.i = $\frac{P}{100}$ >equ &= (equ with P=100*i);

Misplaced & $\frac{(i+1)^n-1)R}{i} + (i+1)^n K = Kn$ Kita dapat memecahkan tingkat R secara simbolis.

>\$&solve(equ,R)

\$\$\$

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

>**Misplaced &** $\lim_{x \rightarrow 0} R(5000, 0, x, 10)$ Jelas, tanpa bunga kita harus membayar kembali 10 tarif 500. Persamaan juga dapat diselesaikan untuk n. Kelihatannya lebih bagus, jika kita menerapkan beberapa penyederhanaan untuk itu.

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

\$\$\$

Latihan Soal

R.2 Exercise set

Soal No 49 Menyederhanakan:

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

>**Misplaced &** $\frac{b^{25}}{32a^{20}c^{10}}$ Soal No 50 Menyederhanakan : $\left(\frac{125p^{12}q^{-14}r^{22}}{25p^8q^6r^{-15}} \right)^{-4}$ &

$((125 * p^{(12)} * q^{(-14)} * r^{(22)}) / (25 * p^{(8)} * q^{(6)} * r^{(-15)}))^{-4}$

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

Soal No 90 calculate

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

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

2

Soal No 91 Calculate

$$\left(\frac{4(8-6)^2 - 4 \times 3 + 2 \times 8}{3^1 + 9^0} \right)$$

>Misplaced & \$SoalNo92Calculate($\frac{14(8-6)^2+4(3-2*8)}{2^2(2^5+5)}$) \$>((4*(8-6)^{2+4})*3-2*8)/(3^1+9^0)

11

R.3 Exercise Set

Perform the indicated operations. no 27

$$(x + 3)^2$$

>Misplaced & ^2 + 6 x + 9no29(y - 5)^2 & expand((y-5)^2)

$$y^2 - 10y + 25$$

no 33

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

>Misplaced & y^2 + 12 x y + 4 x^2no39(3y + 4)(3y - 4) & expand ((3*y+4)*(3*y-4))

$$9y^2 - 16$$

no 42

$$(3x + 5y)(3x - 5y)$$

>Misplaced & x^2 - 25 y^2\$ ### R.4 Exercise set

Faktor the trinomial no 23

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

>\$& solve (t^2+8*t+15)

\$\$\$

no 24.

$$y^2 + 12y + 27$$

>\$& solve(y^2+12*y+27)

\$\$\$

Factor the difference of squares no 47

$$z^2 - 81$$

>\$& solve(z^2-81)

\$\$\$

no 48

$$m^2 - 4$$

>\$& solve(m^2-4)

\$\$\$

no 49

$$16x^2 - 9$$

>\$& solve(16*x^2-9)

\$\$\$

R.5 Exercise set

no 31 Tentukan nilai x!

$$7(3x + 6) = 11 - (x + 2)$$

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

\$\$\$

no 37 Tentukan nilai x!

$$x^2 + 5x = 0$$

>\$& solve(x^2+5*x=0)

\$\$\$

no 42 Tentukan nilai y!

$$y^2 + 25 = 10y$$

>\$& solve(y^2+25=10*y)

\$\$\$

no 47 Tentukan nilai z!

$$12z^2 + z = 6$$

>\$& solve(12*z^(2)+z=6)

\$\$\$

no 60 Tentukan nilai x!

$$5x^2 - 75$$

>\$& solve(5*x^2-75)

\$\$\$

R.6 Exercise set

Sederhanakan!

no 9

>\$& ((x^2-4)/(x^2-4*x+4)), Misplaced & $\frac{x+2}{x-2}$ \$ no 10

>\$& ((x^2+2*x-3)/(x^2-9)), Misplaced & $\frac{x-1}{x-3}$ \$

no 11

>\$& ((x^3-6*x^2+9*x)/(x^3-3*x^2)), Misplaced & $\frac{x-3}{x}$ \$

no 12

>\$& ((y^5-5*y^4+4*y^3)/(y^3-6*y^2+8*y)), Misplaced & $\frac{(y-1)y^2}{y-2}$ \$

no13 >\$& ((6*y^2+12*y-48)/(3*y^2-9*y+6)), Misplaced & $\frac{2(y+4)}{y-1}$ \$ # Menggambar Grafik 2D dengan EMT

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

Plot Dasar

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

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

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

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

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

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

Untuk contoh lain, kami menggambar plot sebagai sisipan di plot lain. Ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu 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);
```

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

Aspek Plot

Plot default menggunakan jendela plot persegi. Anda dapat mengubah ini dengan fungsi `aspect()`. 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 diubah, dan jendela diatur sehingga label memiliki cukup ruang.

```
>aspect(2); // rasio panjang dan lebar 2:1

>plot2d(["sin(x)", "cos(x)"],0,2pi);

>aspect();

>reset;
```

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

Plot 2D di Euler

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

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

Euler dapat 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. " $4x^2$ ") 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");

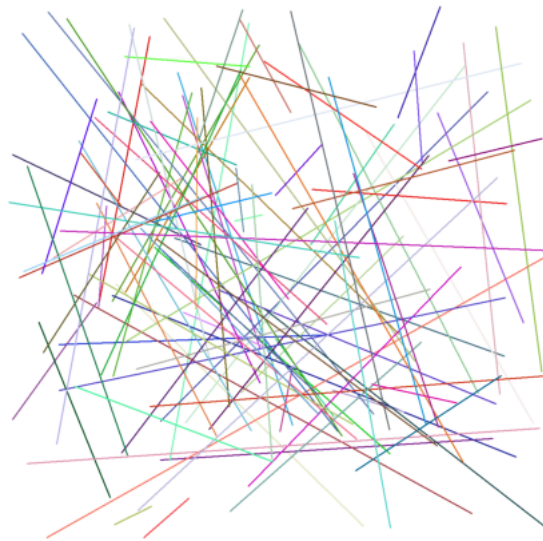
>aspect(1.5); plot2d("x^3-x");

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

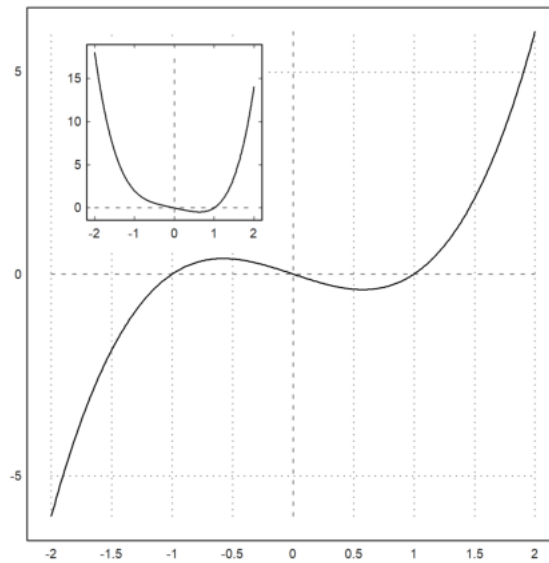
Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

Rentang plot diatur dengan parameter yang ditetapkan sebagai berikut * a,b: rentang x (default -2,2) * c,d: rentang y (default: skala dengan nilai) * r: alternatifnya radius di sekitar pusat plot * cx,cy: koordinat pusat plot (default 0,0)

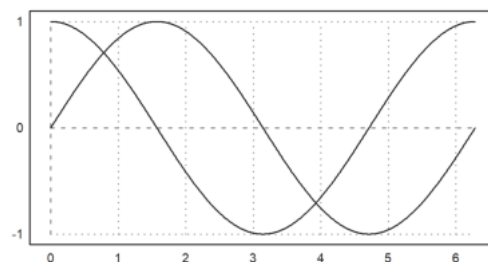
```
>plot2d("x^3-x",-1,2):
```



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



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



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-003.png

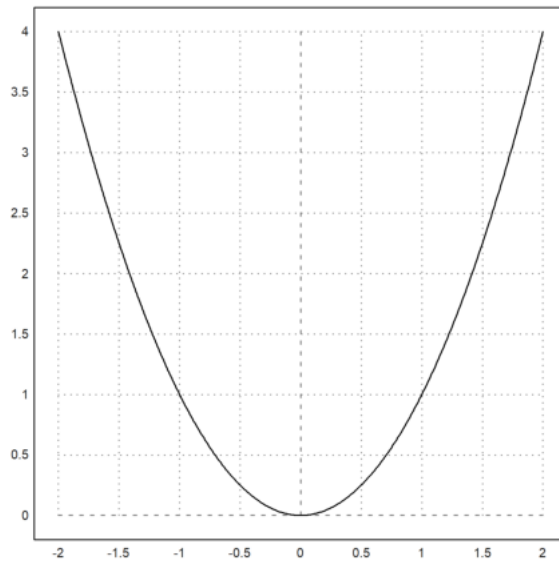
Alternatif untuk titik dua adalah perintah insimg(baris), yang menyisipkan plot yang menempati sejumlah baris teks tertentu.

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.

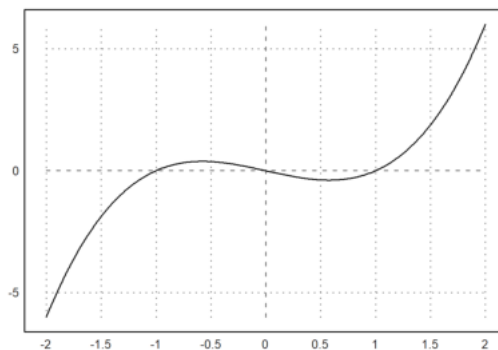
```
>reset; >figure(2,2); ...
> for n=1 to 4; figure(n); plot2d("x^"+n); end; ...
> figure(0):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-004.png

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):
```

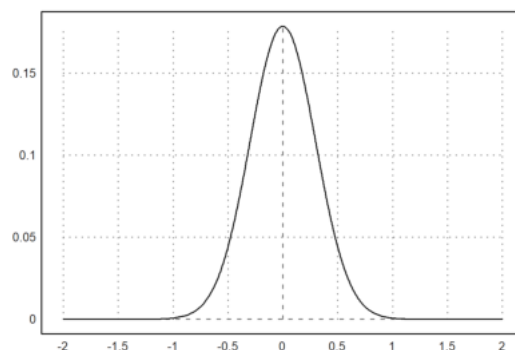


Jika argumen ke plot2d() adalah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot.

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

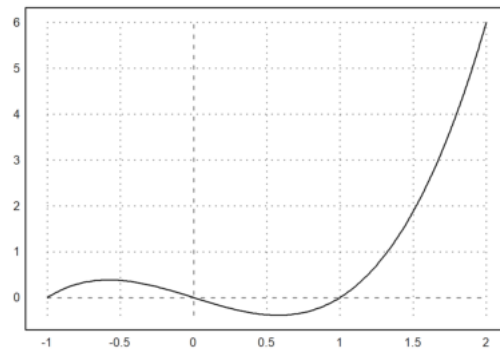
Dalam contoh berikut, kita mengubah gaya kisi, menambahkan label, dan menggunakan label vertikal untuk sumbu y.

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



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-006.png

```
>plot2d("sin(x)+cos(2*x)",0,4pi):
```



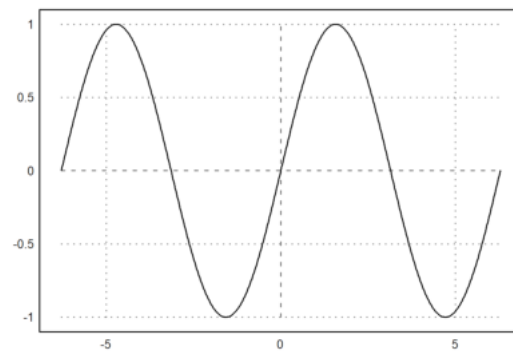
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-007.png

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.

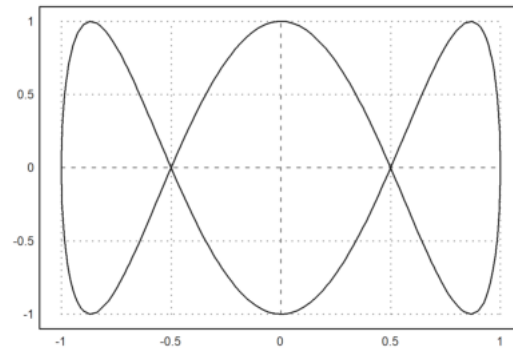
Anda cukup menandai gambar apa saja dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengekspor grafik saat ini dengan fungsi di menu File.

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):`

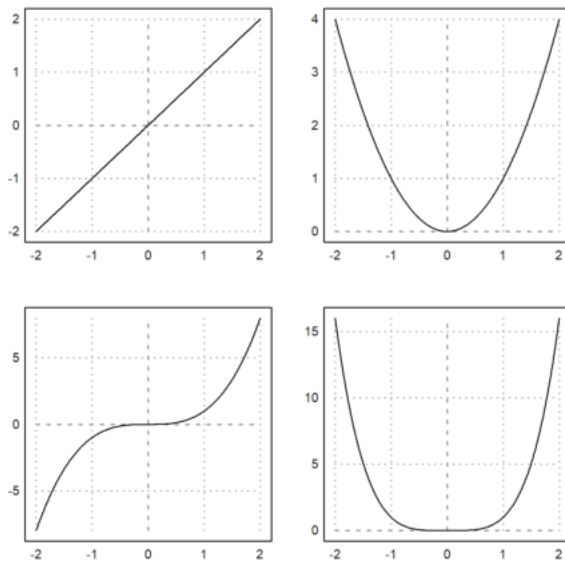


`>plot2d(“x^x”,r=1.2,cx=1,cy=1):`



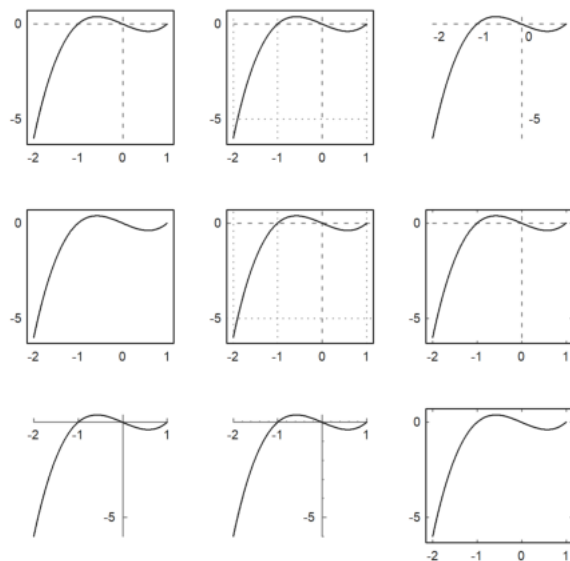
Perhatikan bahwa x^x tidak didefinisikan untuk $x \leq 0$. Fungsi plot2d menangkap kesalahan ini, dan mulai merencanakan segera setelah fungsi didefinisikan. Ini berfungsi untuk semua fungsi yang mengembalikan NAN keluar dari jangkauan definisinya.

`>plot2d(“log(x)”,-0.1,2):`



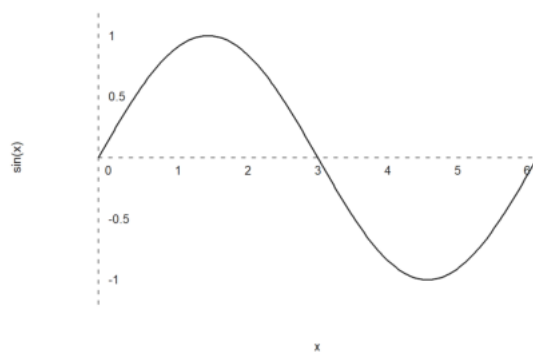
Parameter `square=true` (atau `>square`) memilih y-range secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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



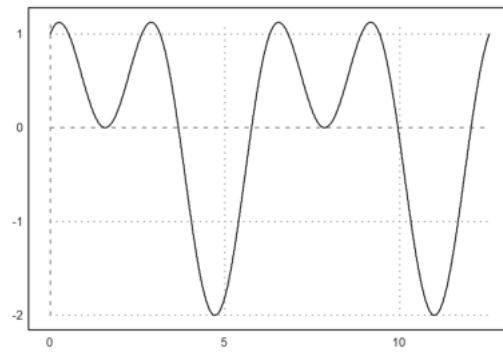
`images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-011.png`

`>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()`.

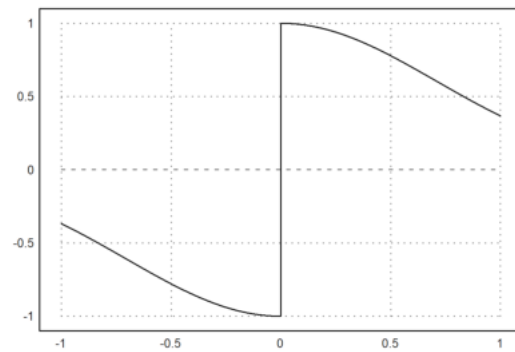
`>plot2d("gamma(x)",1,10,yl="y-values",smaller=6,<vertical):`



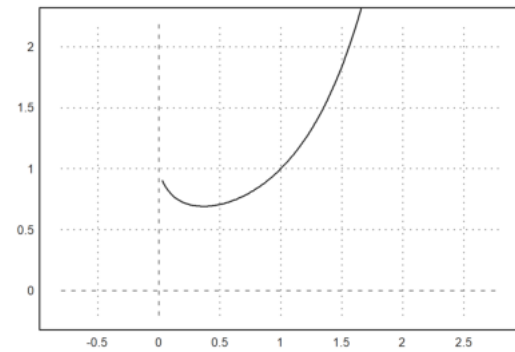
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-013.png

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

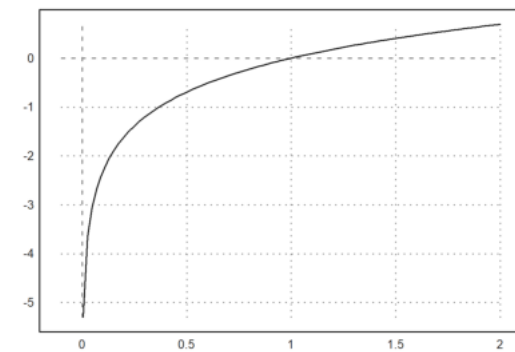
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



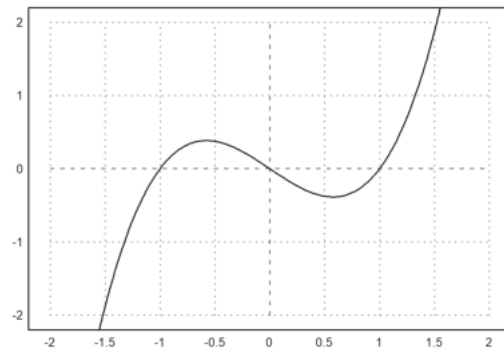
```
>a:=5.6; expr &=
exp(-a*x^2)/a; // define expression >plot2d(expr,-2,2): // plot from -2 to 2
```



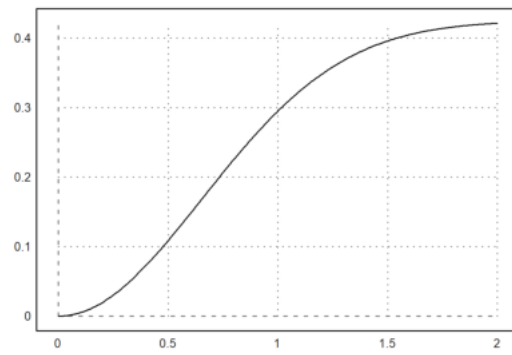
```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```



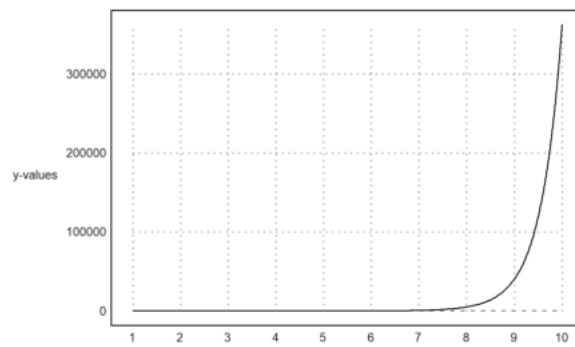
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```



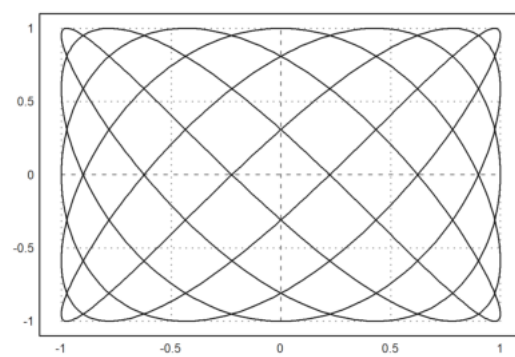
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle



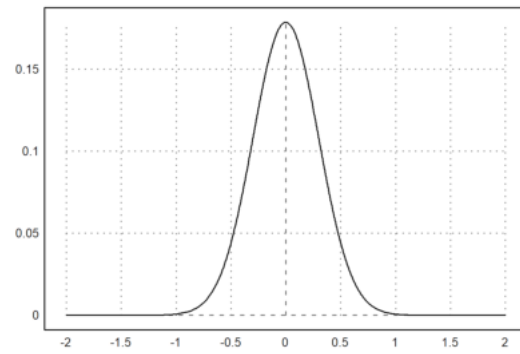
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square



>plot2d("x^2",0,1,steps=1,color=red,n=10):



>plot2d("x^2",>add,steps=2,color=blue,n=10):



Fungsi dalam satu

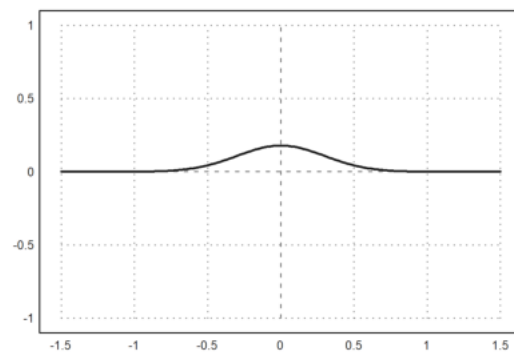
Parameter

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.

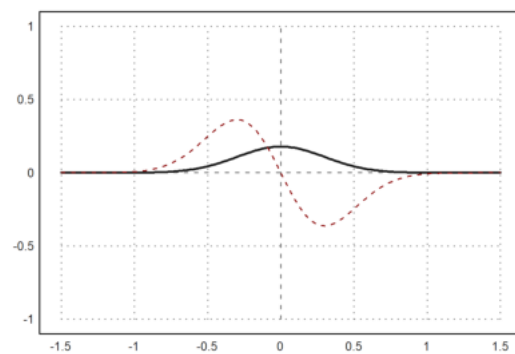
```
>function f(x,a) := x2/a+a*x2-x; // define a function
```

```
>a=0.3; plot2d(“f”,0,1;a): // plot with a=0.3
```

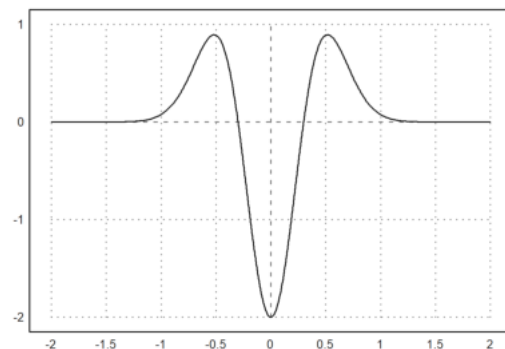


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-022.png

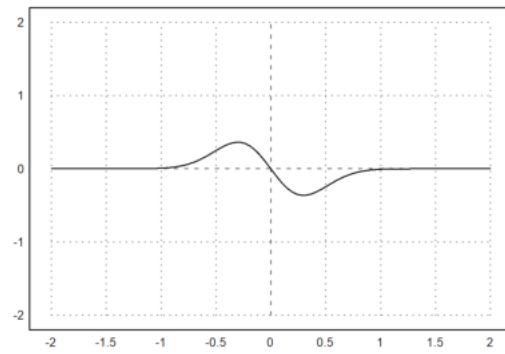
```
>plot2d(“f”,0,1;0.4): // plot with a=0.4
```



```
>plot2d({{“f”,0.2}},0,1): // plot with a=0.2
```

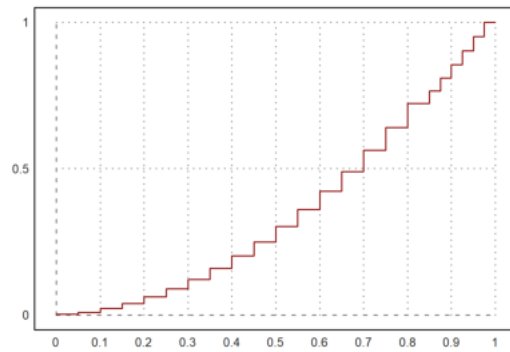


```
>plot2d({{“f(x,b)”,b=0.1}},0,1): // plot with 0.1
```



>function f(x) :=

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



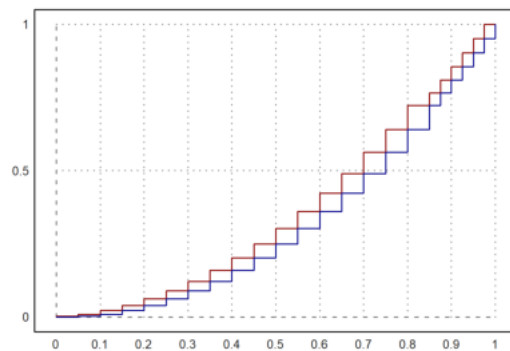
Berikut adalah ringkasan dari fungsi yang diterima * ekspresi atau ekspresi simbolik dalam x * fungsi atau fungsi simbolis dengan nama sebagai "f" * fungsi simbolis hanya dengan nama

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

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

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

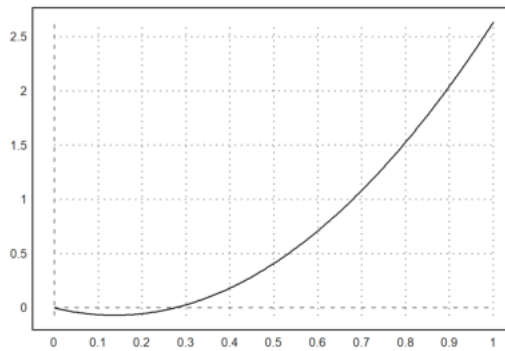
>plot2d(f,0,2):



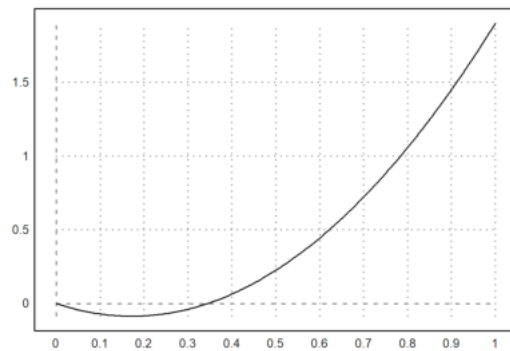
Tentu saja, untuk ekspresi atau ekspresi simbolik, nama variabel sudah cukup untuk memplotnya.

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

$$e^{-x} \sin(x)$$



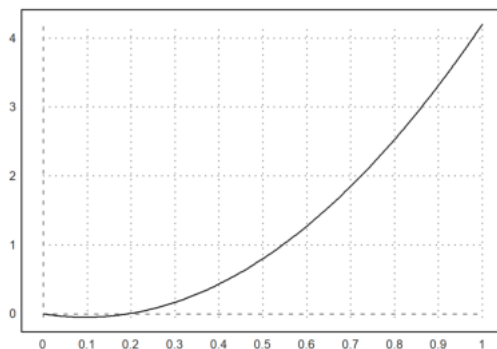
```
>plot2d(expr,0,3pi):
>function f(x) &= x^x;
>plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
>plot2d(&diff(f(x),x),>add,color=red,style="-.-"):
```



Untuk gaya garis ada berbagai pilihan. * gaya="...". Pilih dari "-", "_", ".", ",", "-.-", "-.-". * warna: Lihat di bawah untuk warna. * ketebalan: Default adalah 1.

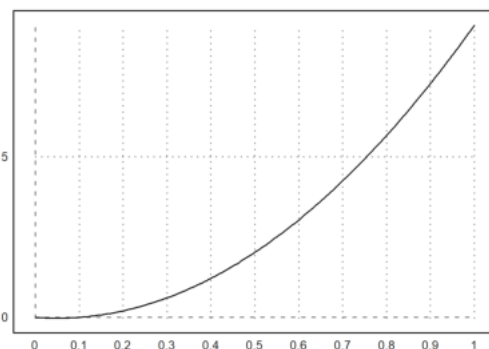
Warna dapat dipilih sebagai salah satu warna default, atau sebagai warna RGB. * 0.15: indeks warna default. * konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, * abu-abu muda, abu-abu, abu-abu tua, oranye, hijau muda, pirus, biru * muda, oranye terang, kuning * rgb(merah, hijau, biru): parameter adalah real dalam [0,1].

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



tampilan warna EMT yang telah ditentukan sebelumnya.

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

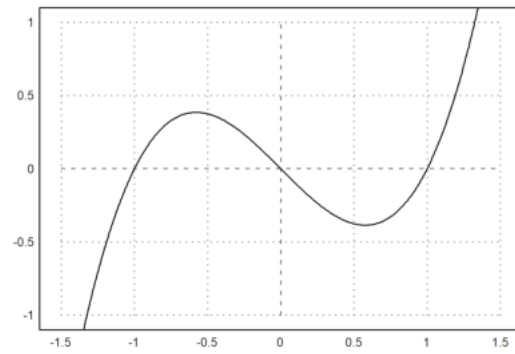


menggunakan warna apa saja.

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

Berikut adalah

Tapi Anda bisa

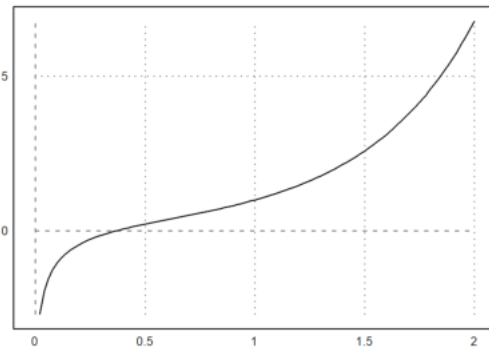


Menggambar

Beberapa Kurva pada bidang koordinat yang sama

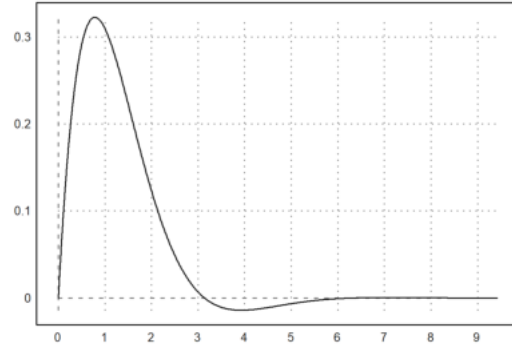
Plot lebih dari satu fungsi (multiple function) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu metode menggunakan `>add` untuk beberapa panggilan ke `plot2d` secara keseluruhan, tetapi panggilan pertama. Kami telah menggunakan fitur ini dalam contoh di atas.

```
>aspect(); plot2d("cos(x)",r=2,grid=6); plot2d("x",style=".",>add):
```



>aspect(1.5);

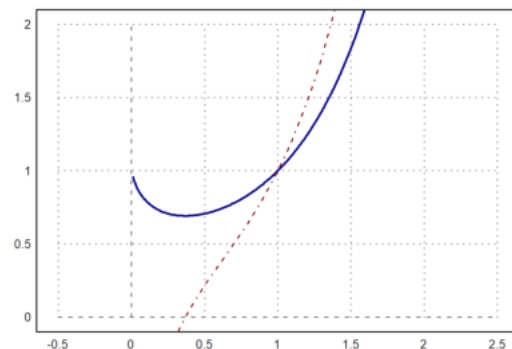
```
plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



Salah satu kegunaan

`>add` adalah untuk menambahkan titik pada kurva.

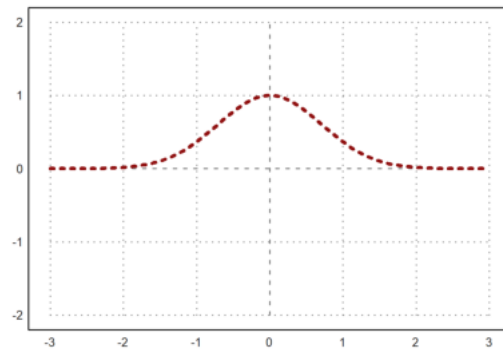
```
>plot2d("sin(x)",0,pi); plot2d(2,sin(2),>points,>add):
```



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); >x0=solve("cos(x)-x",1); ...
> plot2d(x0,x0,>points,>add,title="Intersection Demo"); ...
> label("cos(x) = x",x0,x0,pos="cl",offset=20):
```



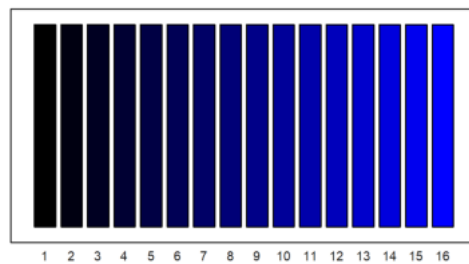
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-036.png

Dalam demo berikut, kami memplot fungsi $\text{sinc}(x) = \sin(x)/x$ dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi 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.

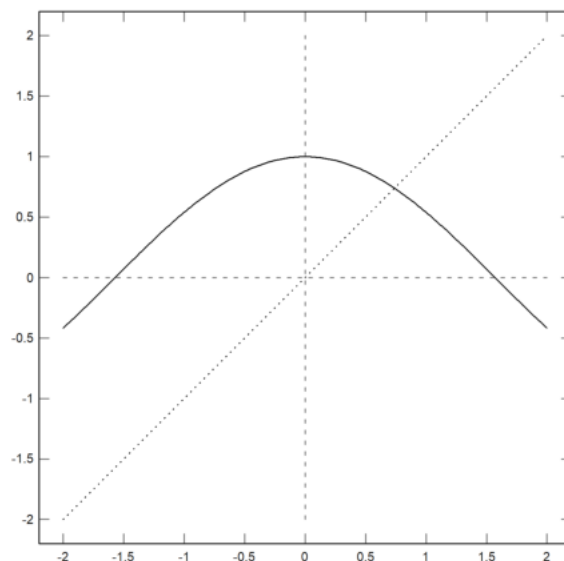
```
>taylor(sin(x)/x, x, 0, 4)  $\frac{x^4}{120} - \frac{x^2}{6} + 1$ 
$>plot2d("sinc(x)", 0, 4pi, color=green, thickness=2); ...
>plot2d(&taylor(sin(x)/x, x, 0, 8), >add, color=blue, style="--"); ...
>plot2d(&taylor(sin(x)/x, x, 0, 16), >add, color=red, style="-.-"); ...
>labelbox(["sinc", "T8", "T16"], styles=["-", "--", "-.-"], ...
>colors=[black, blue, red]):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-038.png

Dalam contoh berikut, kami menghasilkan Bernstein-Polynomial.

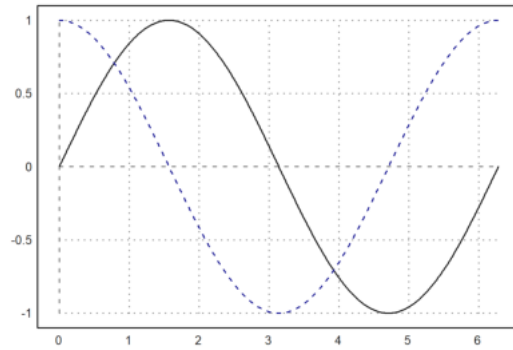
```
>plot2d("(1-x)^10", 0, 1); // plot first function
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)", >add); end;
>insimg;
```



Metode kedua menggunakan pasangan matriks nilai-x dan matriks nilai-y yang berukuran sama.

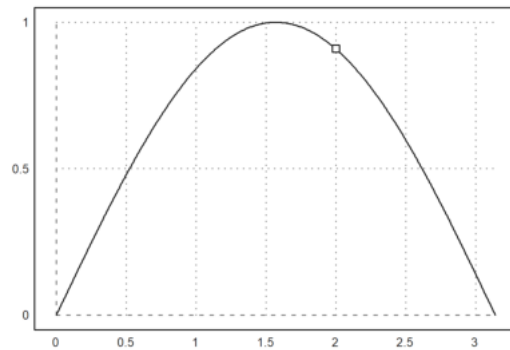
Kami menghasilkan matriks nilai dengan satu Polinomial Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihat pengantar tentang bahasa matriks untuk mempelajari lebih detail.

```
>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):
```



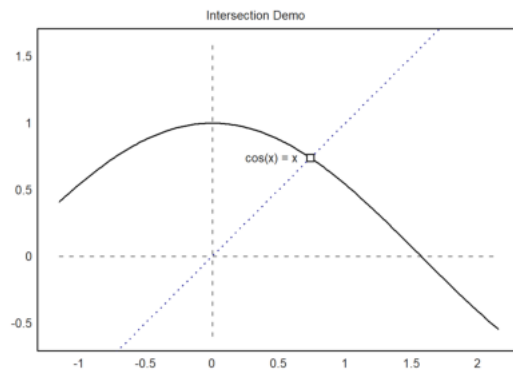
Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

```
>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
```

$\frac{x^4}{120} - \frac{x^2}{6} + 1$ 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
```

```

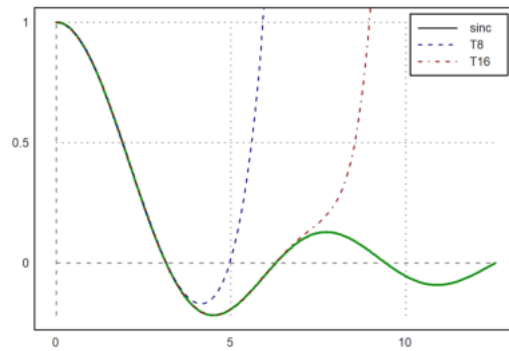
(1-x)^10
10*(1-x)^9*x
45*(1-x)^8*x^2
120*(1-x)^7*x^3
```

```

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
```



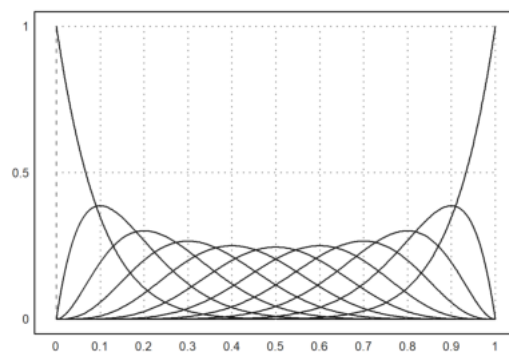
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-044.png

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):
```



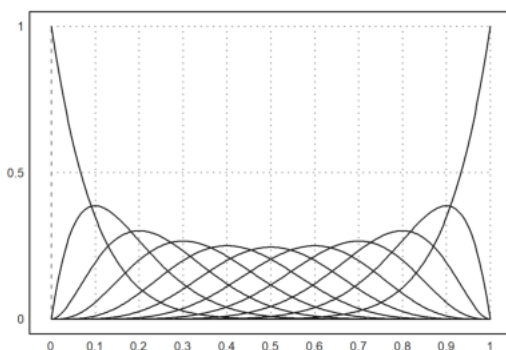
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-045.png

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"):
```

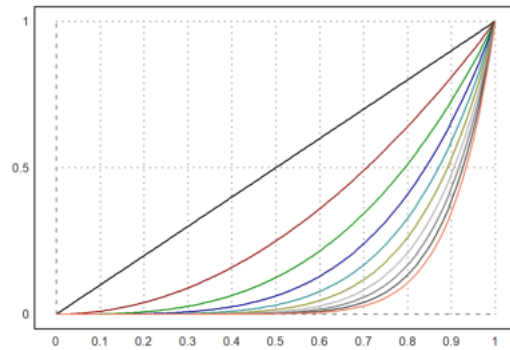


Atau, gunakan

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.

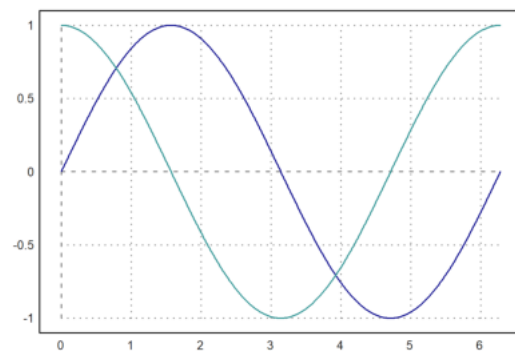
Dalam contoh berikut, kami menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loop).

```
>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)):
```

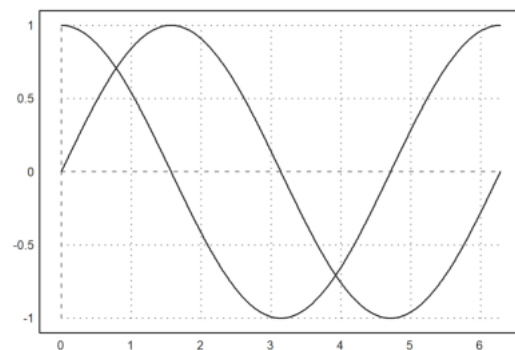


Label Teks

Dekorasi sederhana bisa * judul dengan `judul="..."` * x- dan y-label dengan `xl="..."`, `yl="..."` * label teks lain dengan `label("...",x,y)`

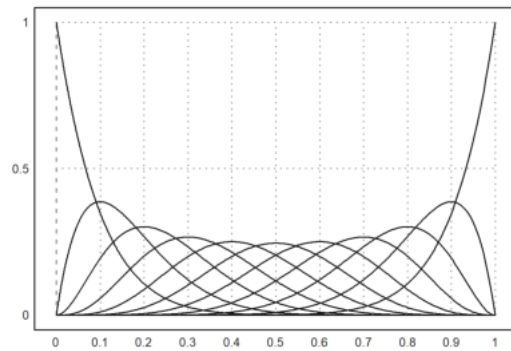
Perintah label akan memplot ke dalam plot saat ini pada koordinat `plot(x,y)`. Itu bisa mengambil argumen posisi.

```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x"):
```



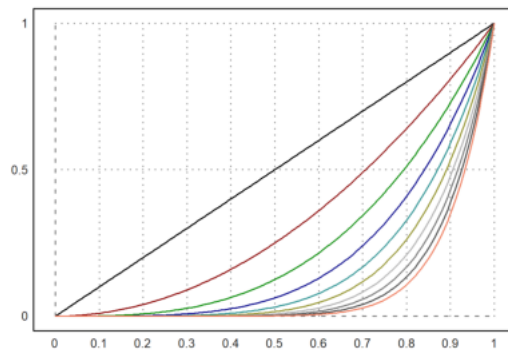
```
>expr := "log(x)/x";
```

```
...
> 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):
```



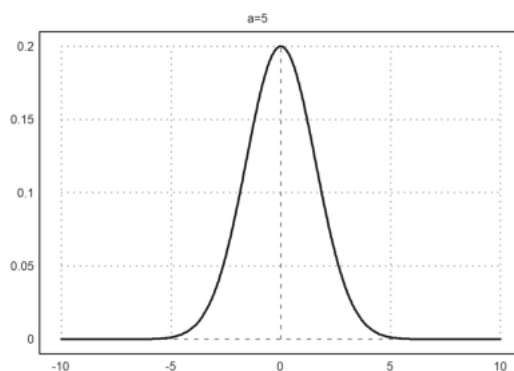
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-051.png

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. Lebarannya 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.

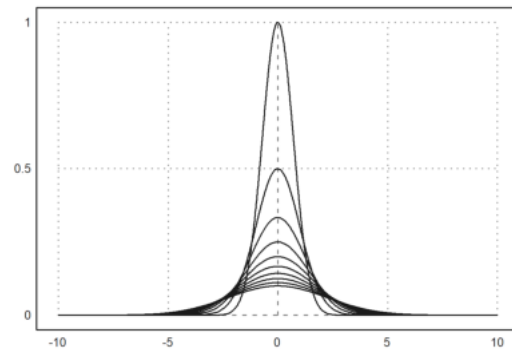
```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
>labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
>tcolor=black,>left):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-052.png

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]):
```

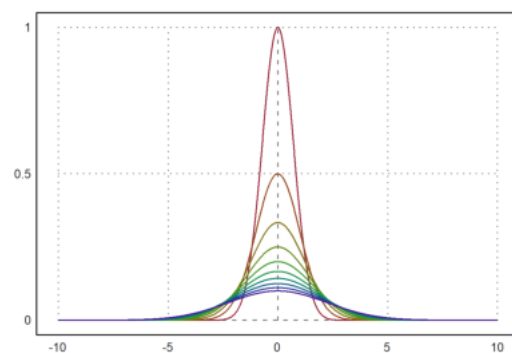


Fitur serupa adalah

fungsi `textbox()`.

Lebar secara default adalah lebar maksimal dari baris teks. Tapi itu bisa diatur oleh pengguna juga.

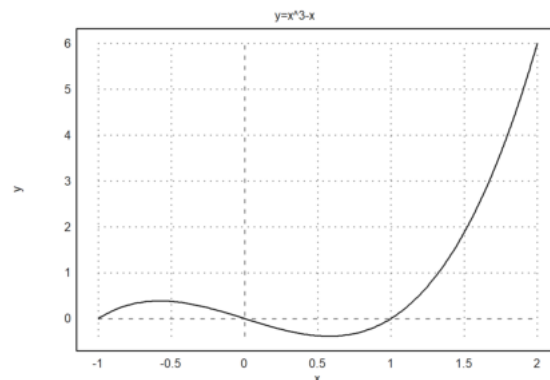
```
>function f(x) &= exp(-x)*sin(2*pi*x); ...
>plot2d("f(x)",0,2pi); ...
>textbox(latex("text{Example of a damped oscillation}\ f(x)=e^{-x}\sin(2\pi x)"),w=0.85);
```



Label teks, judul,

kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

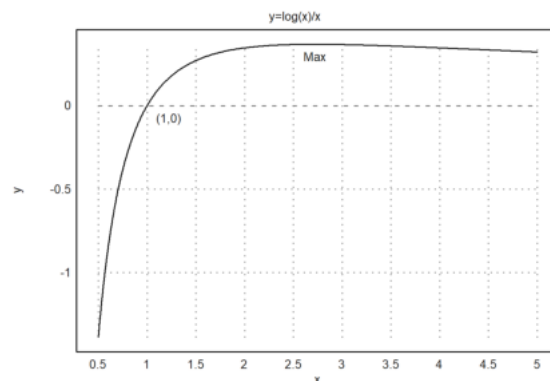
```
>plot2d("x^3-x",title="x → x³ - x");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-055.png

Label pada sumbu x dan y bisa vertikal, begitu juga sumbunya.

```
>plot2d("sinc(x)",0,2pi,xl="x",yl="u" → sinc(x)",>vertical);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-056.png

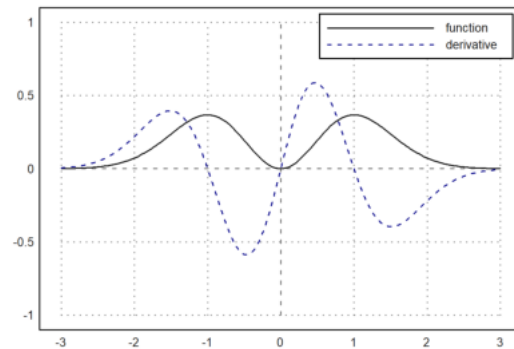
LaTeX

Anda juga dapat memplot rumus LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke biner “latex” dan “dvi2png” 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}"), ...
> x1=latex("\phi"),y1=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):
```

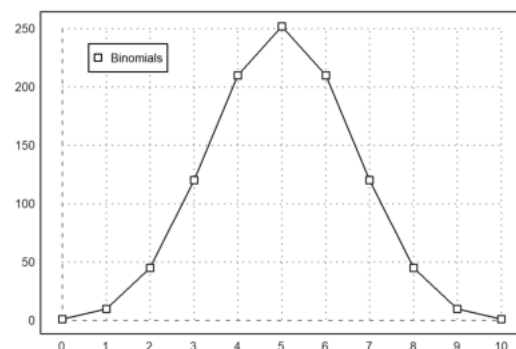


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-057.png

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","2pi"],>latex):
```



Tentu saja, fungsi

juga dapat digunakan. `>function map f(x) ...`

```
if x>0 then return x^4
else return x^2
endif
endfunction
```

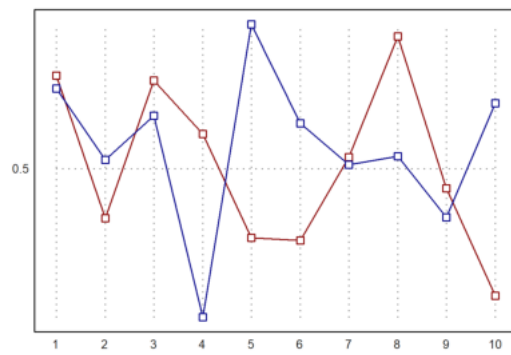
Parameter “peta” membantu menggunakan fungsi untuk vektor. Untuk plot, itu tidak perlu. Tetapi untuk mendemonstrasikan vektorisasi itu berguna, kami menambahkan beberapa poin kunci ke plot di $x=-1$, $x=0$ dan $x=1$.

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,x1="x",y1="f(x)",grid=6); ...
> plot2d([-1,0,1],f([-1,0,1]),>points,>add); ...
> label(latex("x^3"),0.72,f(0.72)); ...
> label(latex("x^2"),-0.52,f(-0.52),pos="ll"); ...
> textbox( ...
> latex("f(x)=
```

$$\begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0 \end{cases}$$

”), ...
> x=0.7,y=0.2):



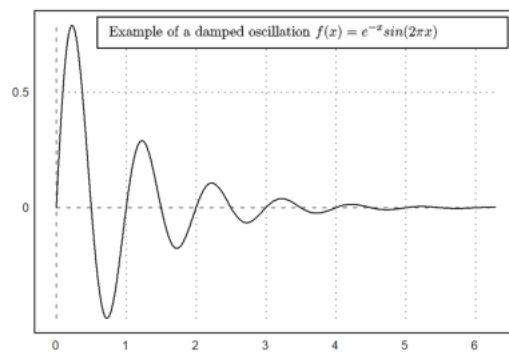
Interaksi

pengguna

Saat memplot 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

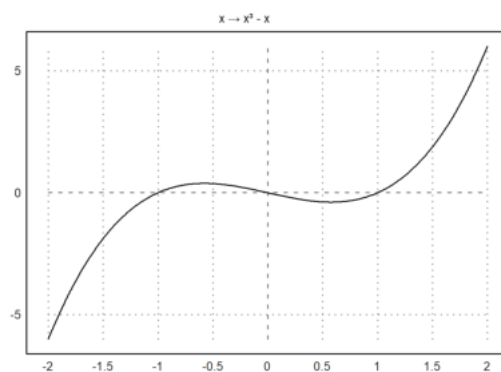
Tombol spasi akan mengatur ulang plot ke jendela plot asli. Saat memplot data, flag >user hanya akan menunggu penekanan tombol.

>plot2d({{"x^3-a*x",a=1}},>user,title="Press any key!"):



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-060.png

>plot2d("exp(x)*sin(x)",user=true, ...
> title="+/- or cursor keys (return to exit)");



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-061.png

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) ...

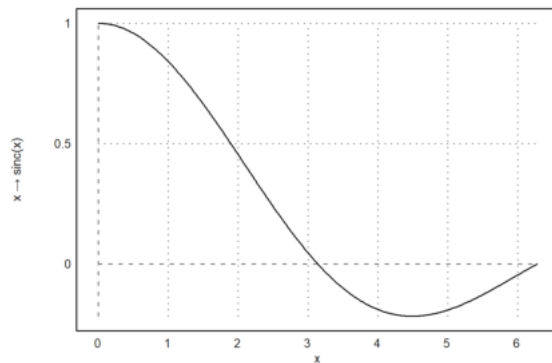
```

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.

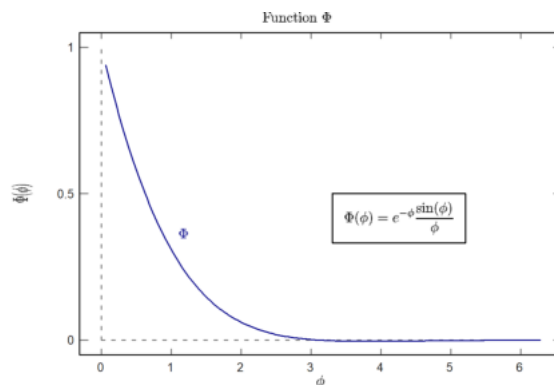
Pertama kita membutuhkan fungsi plot.

```
>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):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-063.png

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

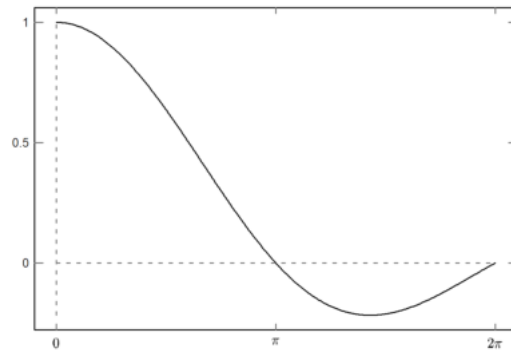
```

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

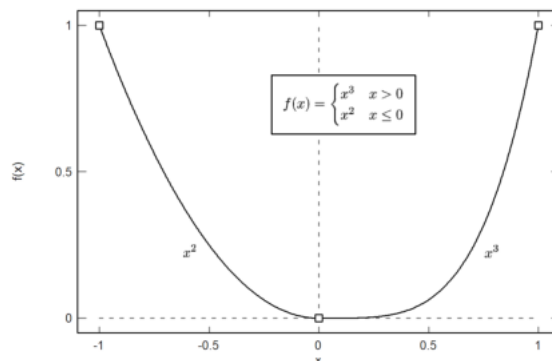
> figure(8); plot2d("x^3-x",grid=7); ... // axes only, ticks at axis

> 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):
```

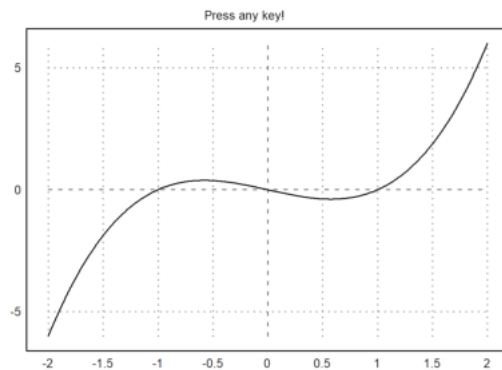


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-065.png

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.

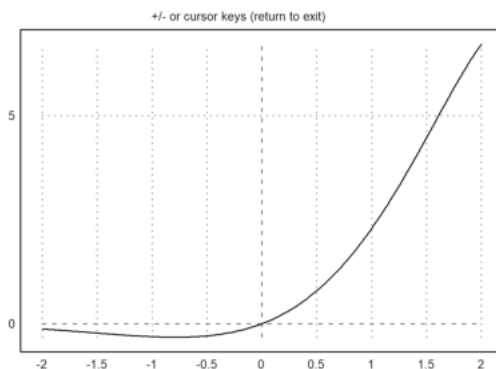
```
>aspect(1.5);
>plot2d("x^3-x",grid=0); // plot
>frame; xgrid([-1,0,1]); ygrid(0); // add frame and grid
```



Untuk judul plot dan

label sumbu, lihat contoh berikut.

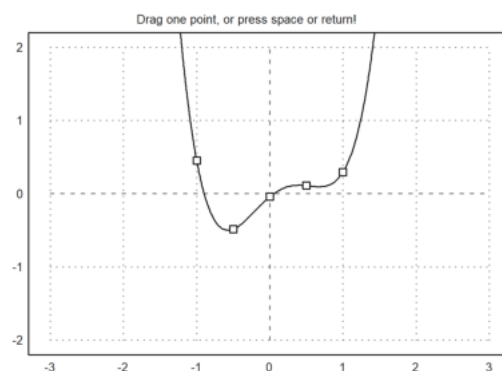
```
>plot2d("exp(x)",-1,1);
>textcolor(black); // set the text color to black
>title(latex("y=e^x")); // title above the plot
>xlabel(latex("x")); // "x" for x-axis
>ylabel(latex("y"),>vertical); // vertical "y" for y-axis
>label(latex("(0,1)"),0,1,color=blue); // label a point
```



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="->");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-068.png

Teks pada plot dapat diatur dengan `label()`. Dalam contoh berikut, "lc" berarti tengah bawah. Ini mengatur posisi label relatif terhadap koordinat plot.

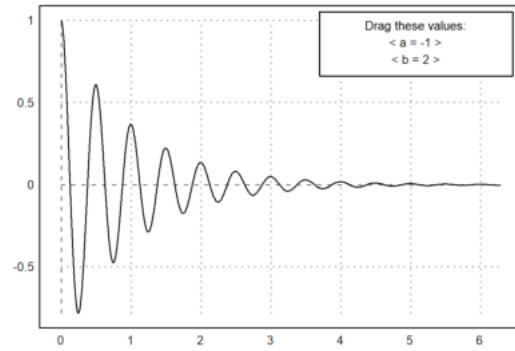
```
>function f(x) &= x^3-x
```

$$x^3 - x$$

```
>plot2d(f,-1,1,>square);
```

```
>x0=fmin(f,0,1); // compute point of minimum
```

```
>label("Rel. Min.",x0,f(x0),pos="lc"): // add a label there
```

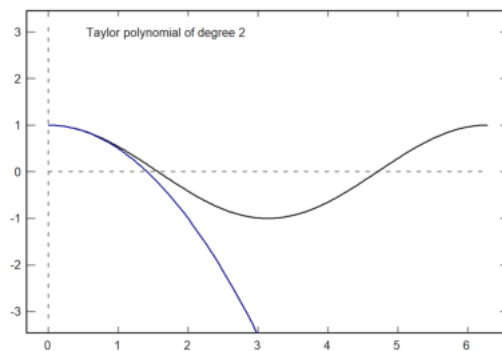


Ada juga kotak teks.

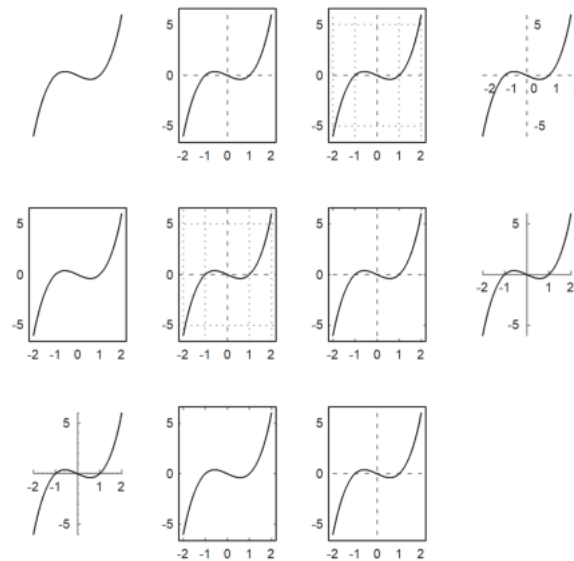
```
>plot2d(&f(x),-1,1,-2,2); // function
```

```
>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=["-", "-. "]):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-071.png

```
>gridstyle(">",color=gray,textcolor=gray,framecolor=gray); ...
```

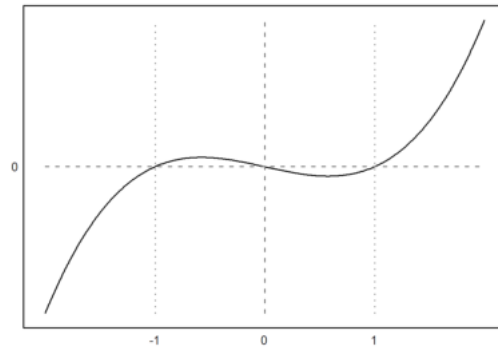
```
>plot2d("x^3-x",grid=1); ...
```

```
>settitle("y=x^3-x",color=black); ...
```

```
>label("x",2,0,pos="bc",color=gray); ...
```

```
>label("y",0,6,pos="cl",color=gray); ...
```

```
>reset():
```

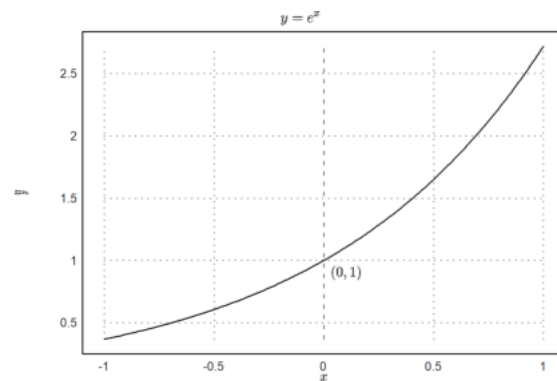


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-072.png

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.

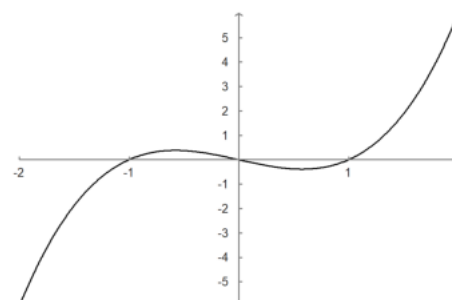
```
>fullwindow; ...
> gridstyle(color=darkgray,textcolor=darkgray); ...
> plot2d(["2^x","1","2(-x)"],a=-2,b=2,c=0,d=4,<grid,color=4:6,<frame); ...
> xaxis(0,-2:1,style="-->"); xaxis(0,2,"x",<axis); ...
> yaxis(0,4,"y",style="-->"); ...
> yaxis(-2,1:4,>left); ...
> yaxis(2,2^(-2:2),style="--,<left); ...
> labelbox(["2^x","1","2-x"],colors=4:6,x=0.8,y=0.2); ...
> reset:
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-073.png

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"2pi"],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"φ"); ylabel(u"f(φ)");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-074.png

Merencanakan Data 2D

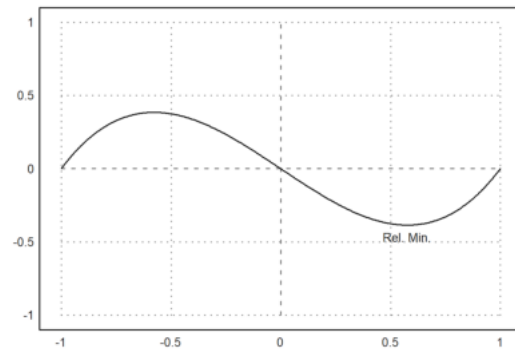
Jika x dan y adalah vektor data, data ini akan digunakan sebagai koordinat x dan y dari suatu kurva. Dalam hal ini, a , b , c , dan d , atau radius r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Atau, >persegi dapat diatur untuk menjaga rasio aspek persegi.

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".

Tapi Anda bisa memasukkan data secara langsung. * Gunakan vektor baris untuk x dan y untuk satu fungsi. * Matriks untuk x dan y diplot baris demi baris.

Berikut adalah contoh dengan satu baris untuk x dan y .

```
>x=-10:0.1:10; y=exp(-x^2)*x; plot2d(x,y):
```



Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plotnya bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya. * style="...": Pilih dari "[", "<>", "o", ".", "..", "+", ",", "[/]", "#", ">#", "o#", ".#", "#", ";", "|".

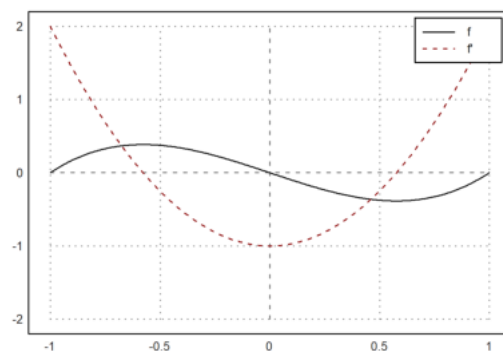
Untuk memplot set poin gunakan >points. Jika warna adalah vektor warna, setiap titik mendapat warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna berlaku untuk baris matriks.

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
```

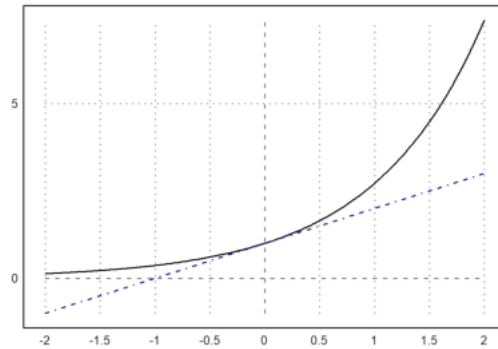
```
>plot2d(xdata,ydata,>points,>add,style="o"); // add points
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-076.png

```
>p=polyfit(xdata,ydata,1); // get regression line
```

```
>plot2d("polyval(p,x)",>add,color=red); // add plot of line
```



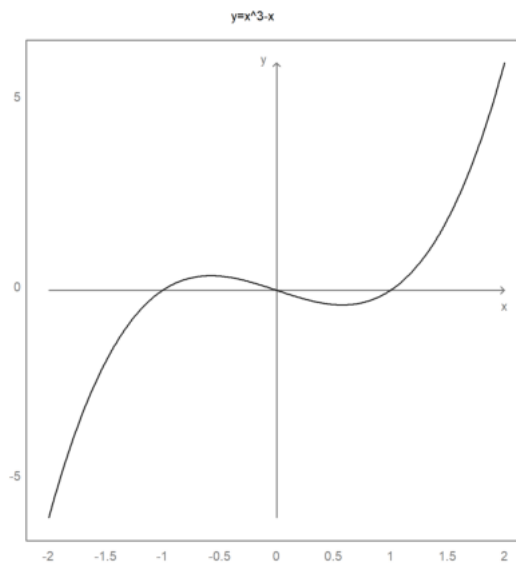
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-077.png

Menggambar Daerah Yang Dibatasi Kurva

Plot data benar-benar poligon. Kita juga dapat memplot kurva atau kurva terisi. * terisi=benar mengisi plot. * style="...": Pilih dari "#", "p", "u", "r". * 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.

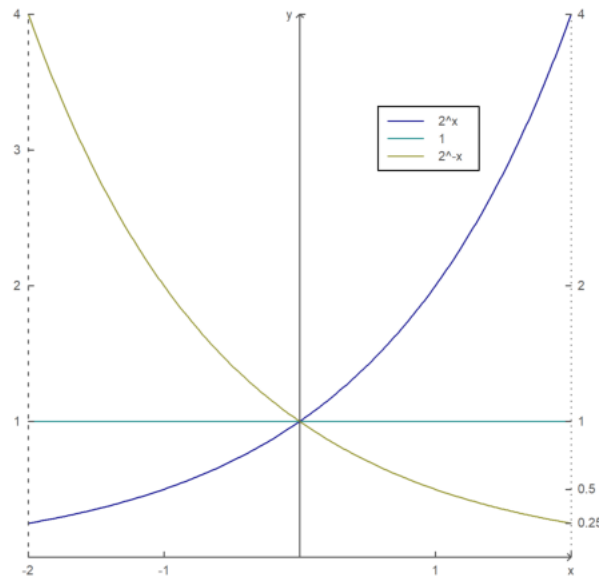
```
>t=linspace(0,2pi,1000); // parameter for curve
>x=sin(t)*exp(t/pi); y=cos(t)*exp(t/pi); // x(t) and y(t)
>figure(1,2); aspect(16/9)
>figure(1); plot2d(x,y,r=10); // plot curve
>figure(2); plot2d(x,y,r=10,>filled,style="r",fillcolor=red); // fill curve
>figure(0):
```



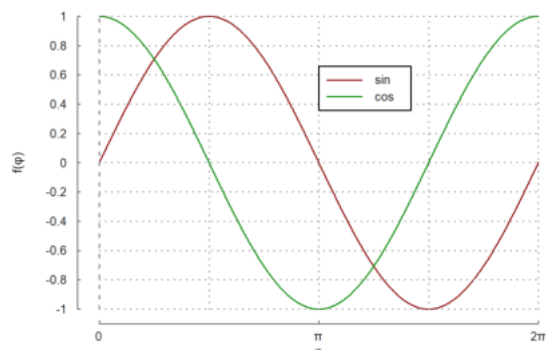
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-078.png

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="r");
```

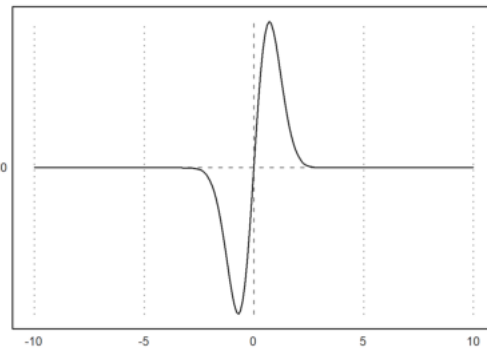


```
>t=linspace(0,2pi,6); ...
> plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.2):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-080.png

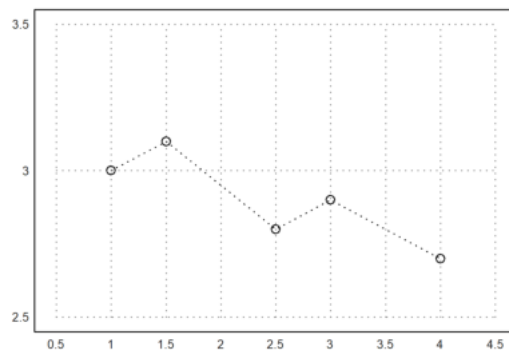
```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-081.png

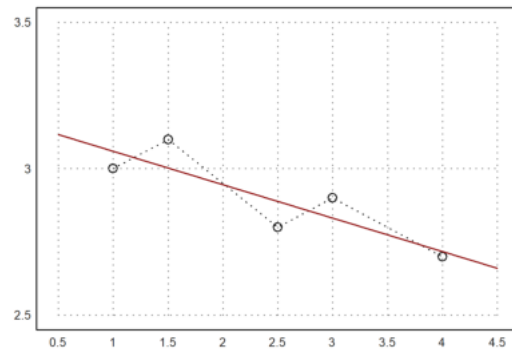
Contoh lainnya adalah segi empat, yang kita buat dengan 7 titik pada lingkaran satuan.

```
>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 $A[k], v \leq 3$ 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);
```

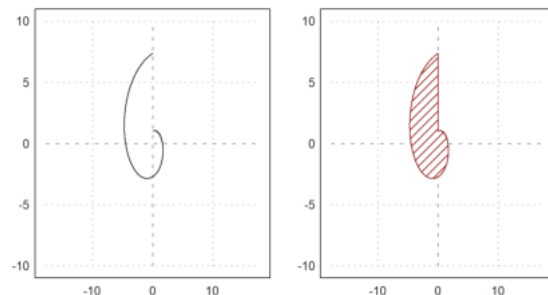


Poin utama dari bahasa matriks adalah memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

```
>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.

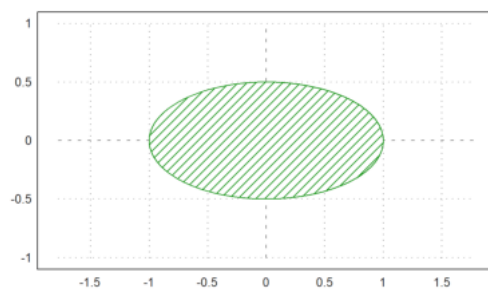
```
>plot2d(x,y,<grid,<frame,>filled):
```



Sebuah vektor interval diplot terhadap nilai x sebagai daerah terisi antara nilai interval bawah dan atas.

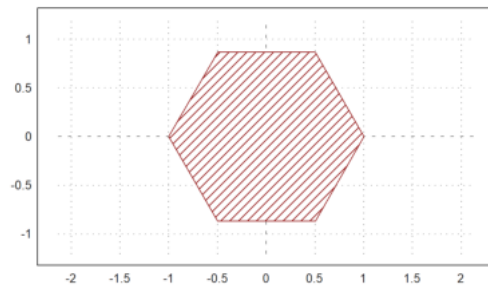
Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga digunakan untuk memplot kesalahan statistik.

```
>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.

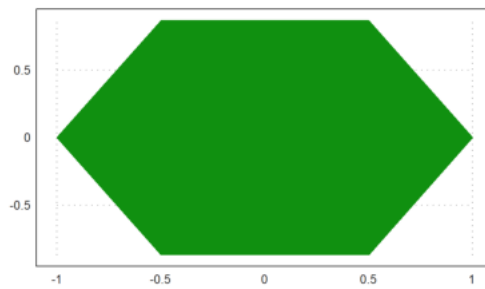
```
>t=-1:0.01:1; x=t-0.01,t+0.01; y=x^3-x;
>plot2d(t,y):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-086.png

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.

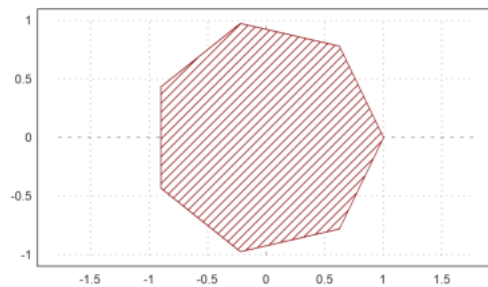
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```



Kami juga dapat

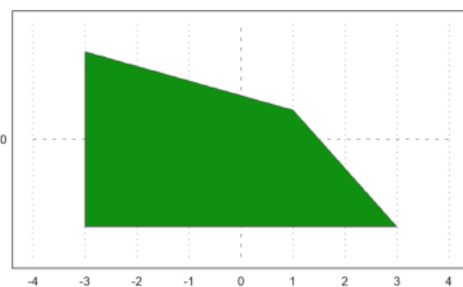
mengisi rentang nilai seperti

```
>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/"): 
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-088.png

```
>plot2d("cos(x)", "sin(x)^3",xmin=0,xmax=2pi,>filled,style="/"): 
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-089.png

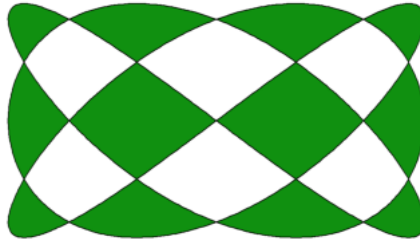
Grafik Fungsi Parametrik

Nilai-x tidak perlu diurutkan. (x,y) hanya menggambarkan kurva. Jika x diurutkan, kurva tersebut merupakan grafik fungsi.

Dalam contoh berikut, kami memplot spiral

Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptif() untuk mengevaluasi ekspresi (lihat fungsi adaptif()) untuk lebih jelasnya.

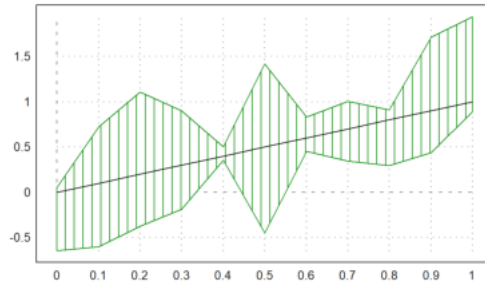
```
>t=linspace(0,1,1000); ...
> plot2d(t*cos(2*pi*t),t*sin(2*pi*t),r=1):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-090.png

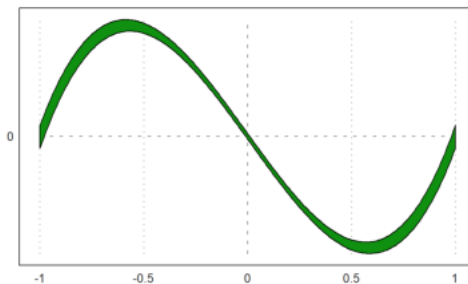
Atau, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

```
>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(2*pi*t); y=r*sin(2*pi*t);
```

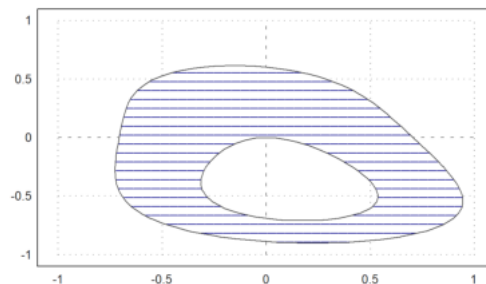
```
>plot2d(x,y,r=1):
```



Dalam contoh

berikutnya, kami memplot kurva dengan

```
>t=linspace(0,2*pi,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):
```



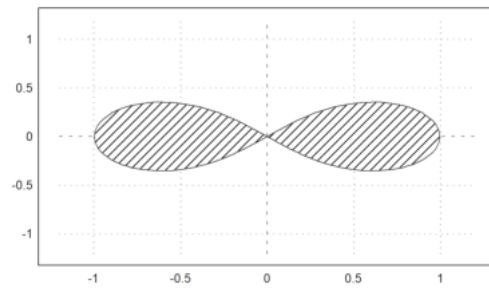
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-093.png

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.

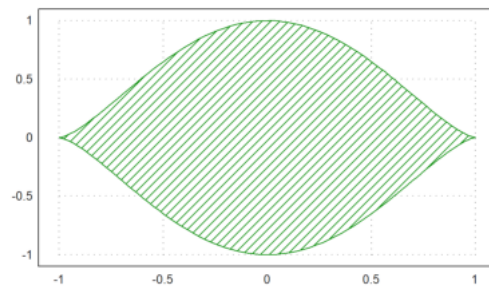
Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks. Dalam contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

```
>aspect(); r=linspace(0,1,50); a=linspace(0,2*pi,80)'; z=r*exp(1*a);...
> plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```



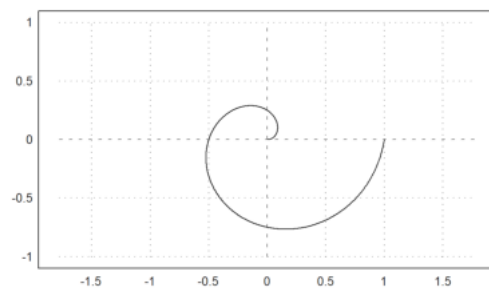
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-094.png

```
>aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=r*exp(I*a);
>plot2d(exp(z),cgrid=[40,10]):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-095.png

```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```

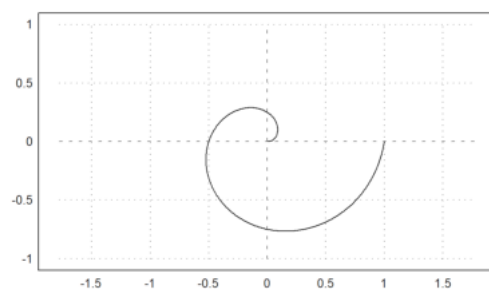


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-096.png

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

```
>t=linspace(0,2pi,1000); ...
> plot2d(exp(I*t)+exp(4*I*t),r=2):
```



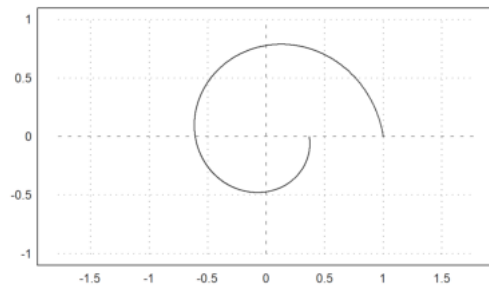
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-097.png

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.

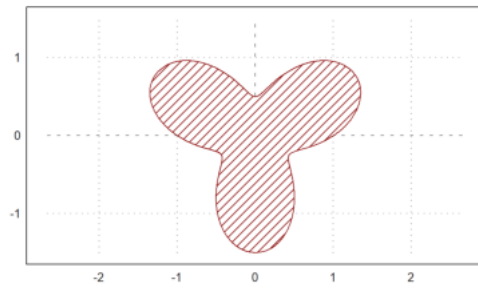
```
>plot2d(cumsum(randnormal(1,1000))):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-098.png

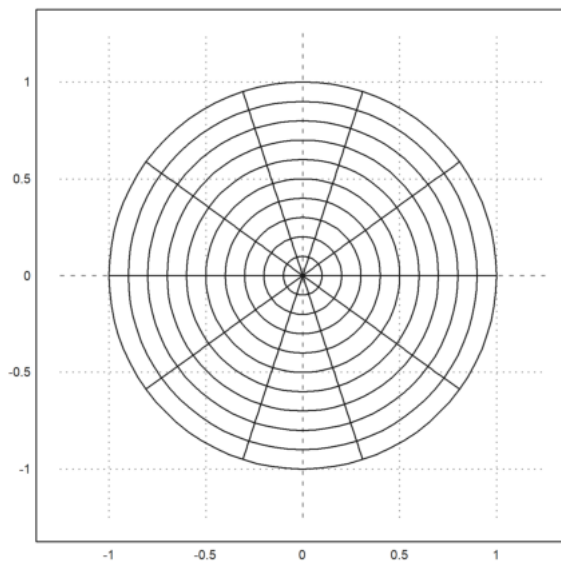
Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

```
>X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-099.png

```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-100.png

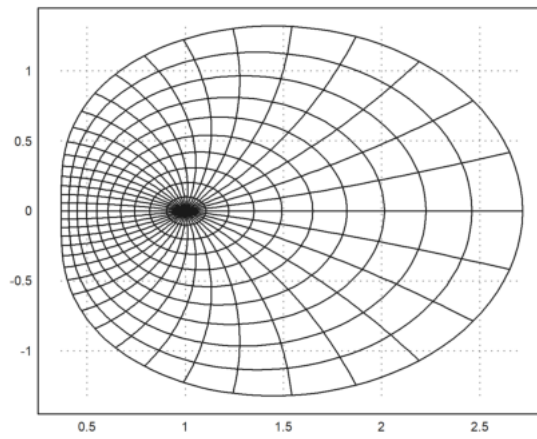
Itu juga dapat menampilkan string sebagai label.

```
>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="-");
```

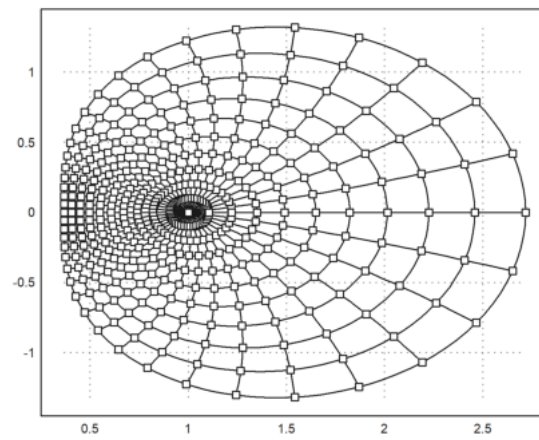
```
>title("Temperature");
```



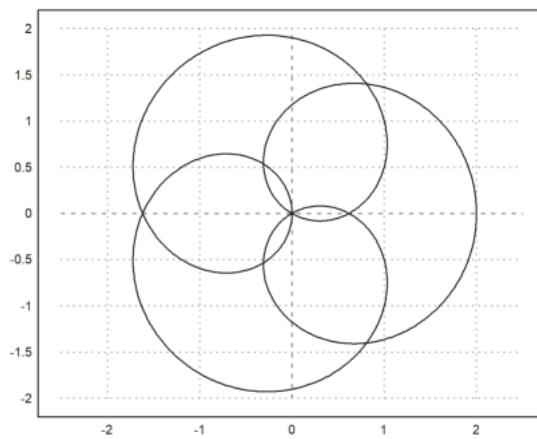
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-101.png

```
>k=0:10;
```

```
>plot2d(k,bin(10,k),>bar):
```



```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



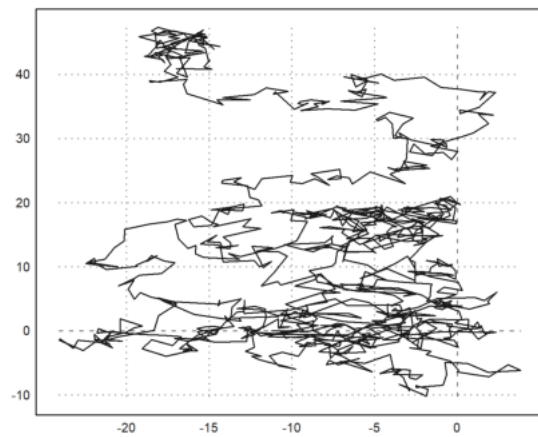
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-103.png

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```



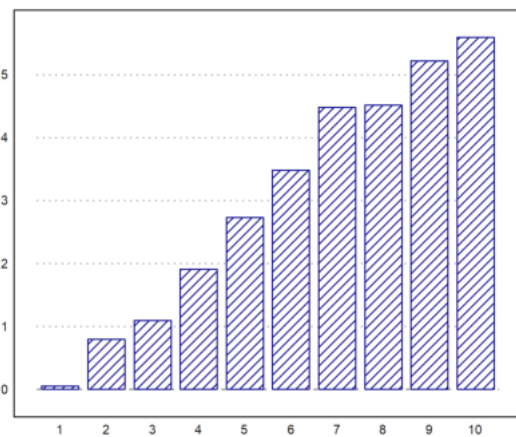
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-104.png

`>plot2d(normal(1,1000),>distribution,style="O"):`



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-105.png

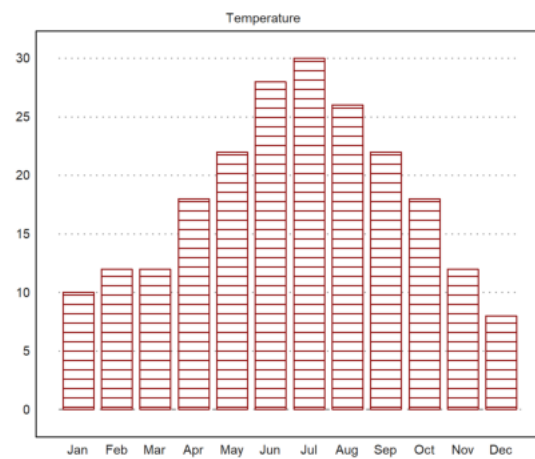
`>plot2d("qnormal",0,5;2.5,0.5,>filled):`



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-106.png

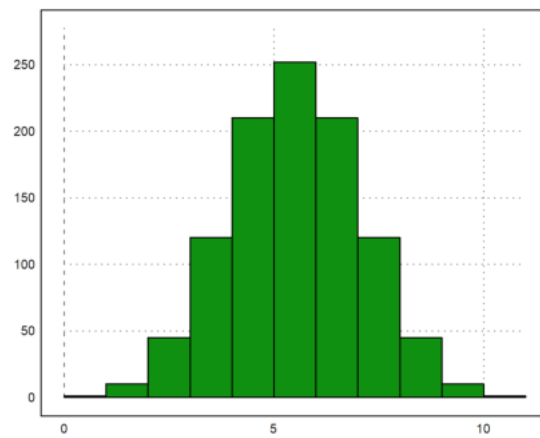
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan `distribution=n` dengan `plot2d`. `>w=randexponential(1,1000); // exponential distribution`

`>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`.

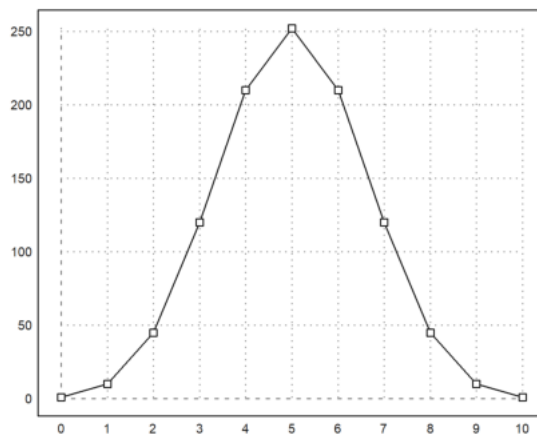
```
>w=normal(1000); // 0-1-normal distribution
>{x,y}=histo(w,10,v=[-6,-4,-2,-1,0,1,2,4,6]); // interval bounds v
>plot2d(x,y,>bar):
```



Fungsi `statplot()`

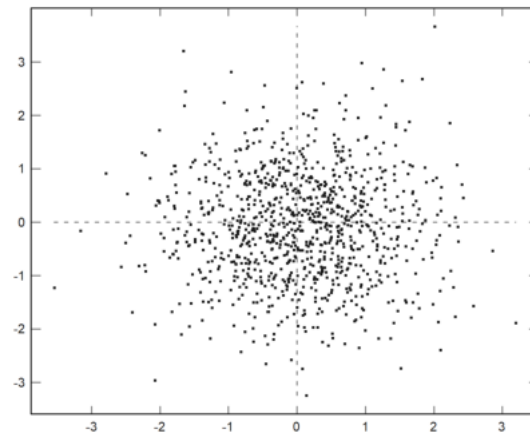
menyeting gaya dengan string sederhana.

```
>statplot(1:10,cumsum(random(10)),"b"):
```



`images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-109.png`

```
>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):
```

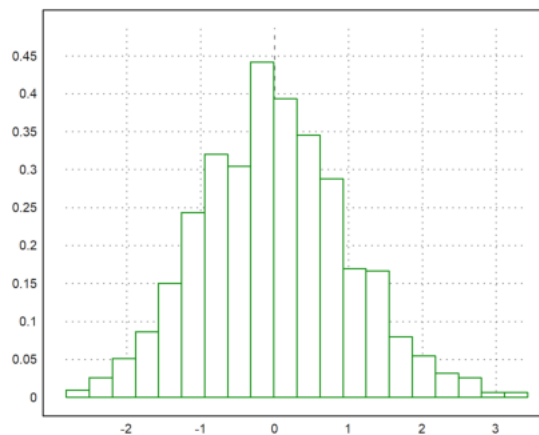


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-110.png

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.

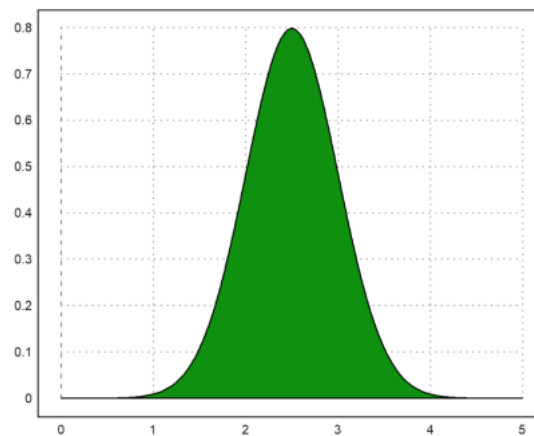
```
>n=10; k=bin(n,0:n); ...
>plot2d(-0.5:n+0.5,k,bar=true,fillcolor=lightgray):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-111.png

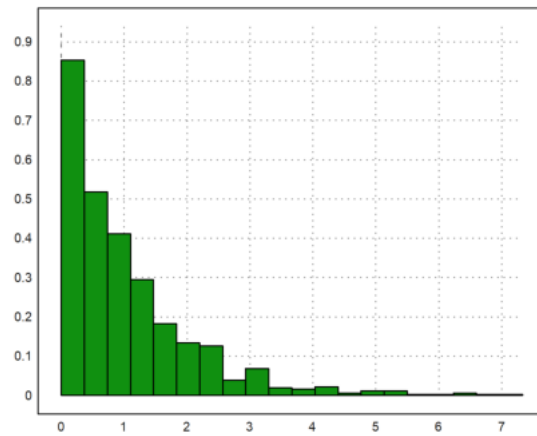
Data untuk plot batang ($\text{bar}=1$) dan histogram ($\text{histogram}=1$) dapat dinyatakan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan $>\text{distribusi}$ (atau $\text{distribusi}=n$). Histogram nilai xv akan dihitung secara otomatis dengan $>\text{histogram}$. Jika $>\text{genap}$ ditentukan, nilai xv akan dihitung dalam interval bilangan bulat.

```
>plot2d(normal(10000),distribution=50):
```



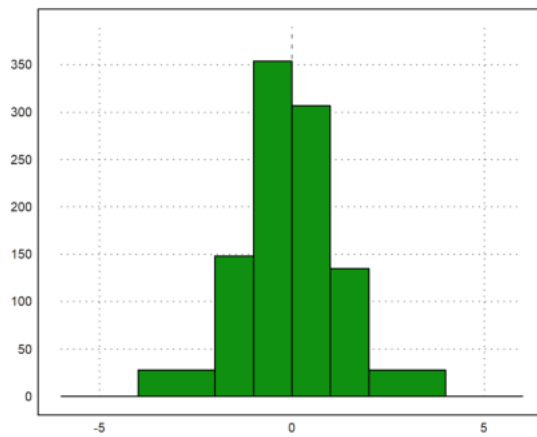
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-112.png

```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```



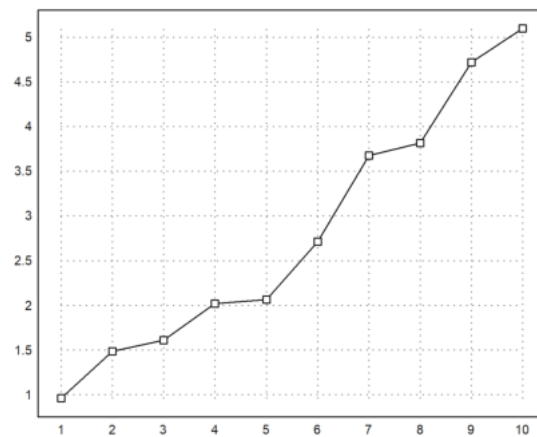
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-113.png

>columnsplot(m,k):



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-114.png

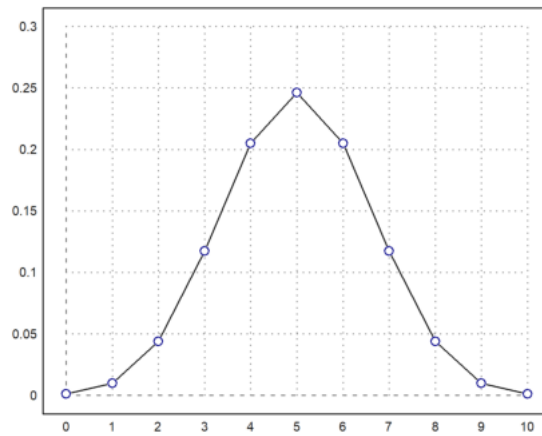
>plot2d(random(600)*6,histogram=6):



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-115.png

Untuk distribusi, ada parameter distribusi=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

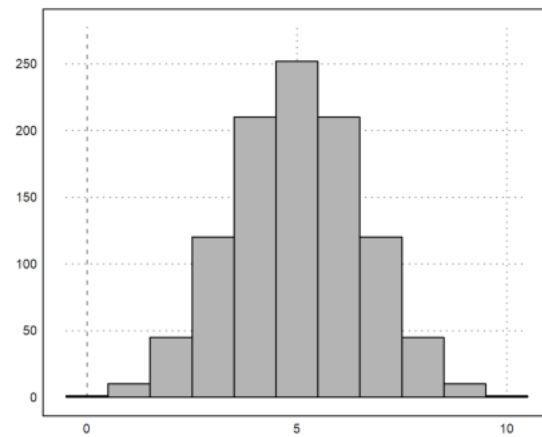
>plot2d(normal(1,1000),distribution=10,style="v"):



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-116.png

Dengan parameter even=true, ini akan menggunakan interval integer.

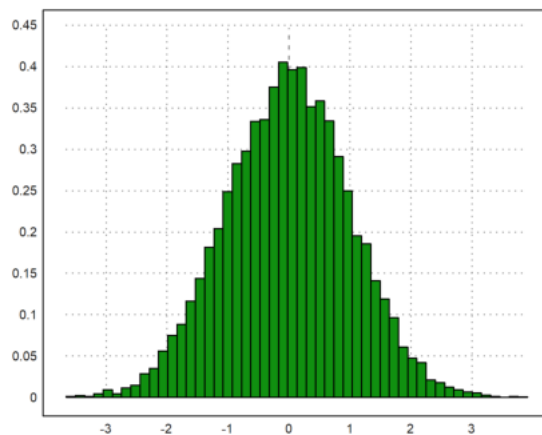
>plot2d(intrandom(1,1000,10),distribution=10,even=true):



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-117.png

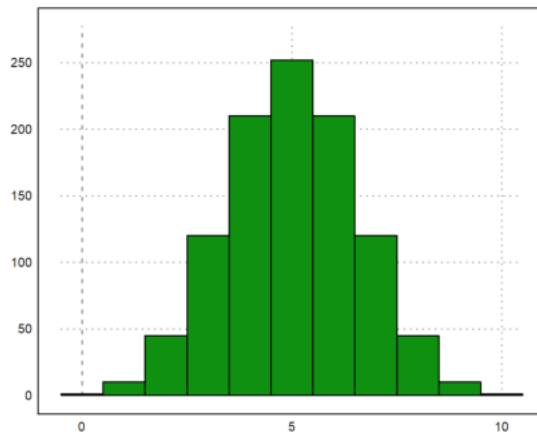
Perhatikan bahwa ada banyak plot statistik, yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

>columnsplot(getmultiplicities(1:6,intrandom(1,6000,6))):



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-118.png

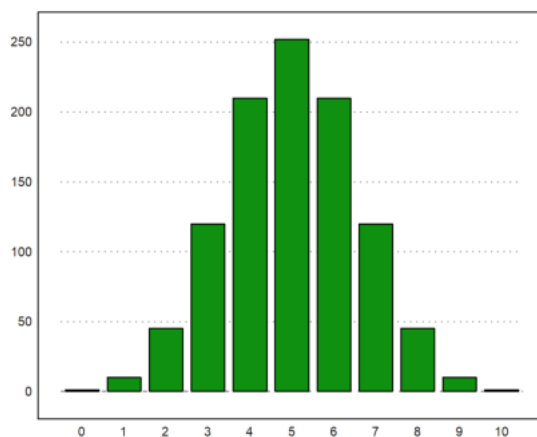
>plot2d(normal(1,1000),>distribution); ...
> plot2d("qnormal(x)",color=red,thickness=2,>add):



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-119.png

Ada juga banyak plot khusus untuk statistik. Sebuah boxplot menunjukkan kuartil dari distribusi ini dan banyak dari outlier. Menurut definisi, outlier dalam boxplot adalah data yang melebihi 1,5 kali kisaran 50% tengah plot.

`>M=normal(5,1000); boxplot(quartiles(M)):`



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-120.png

Fungsi Implisit

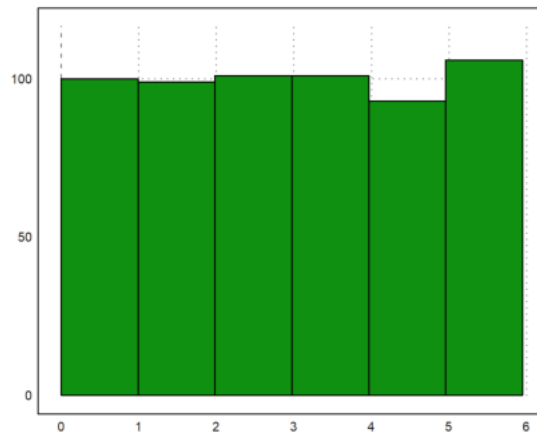
Plot implisit menunjukkan garis level yang menyelesaikan $f(x,y)=\text{level}$, di mana “level” dapat berupa nilai tunggal atau vektor nilai. Jika `level=“auto”`, akan ada garis level n_c , yang akan menyebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat 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 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.

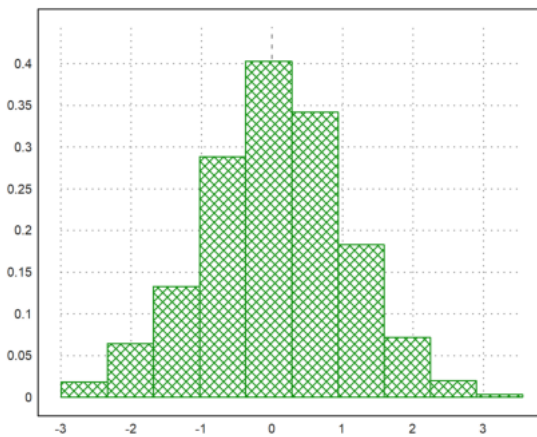
`>aspect(1.5);`

`>plot2d(“x2+y2-x*y-x”,r=1.5,level=0,contourcolor=red):`



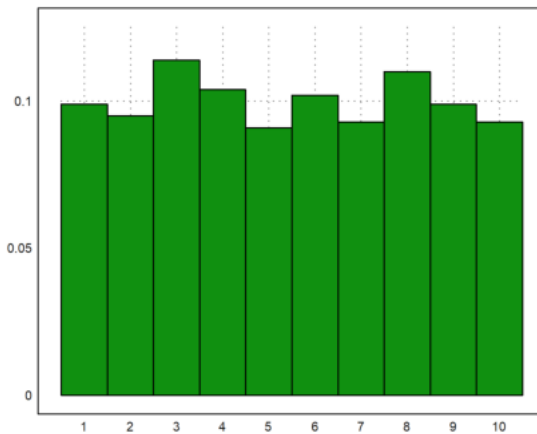
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-121.png

```
>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
```



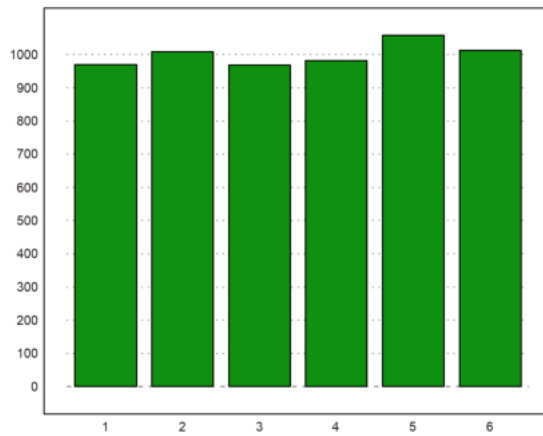
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-122.png

```
>plot2d(expr,level=0:0.5:20,>hue,contourcolor=white,n=200): // nice
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-123.png

```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

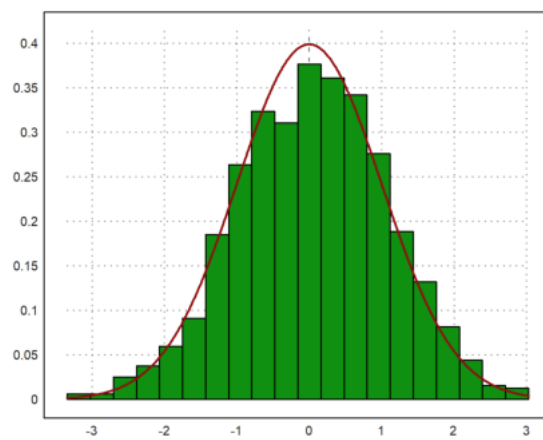


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-124.png

Ini berfungsi untuk plot data juga. Tetapi Anda harus menentukan rentangnya untuk label sumbu.

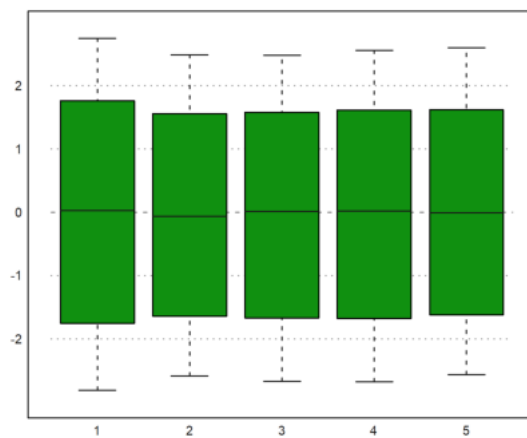
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
```

```
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



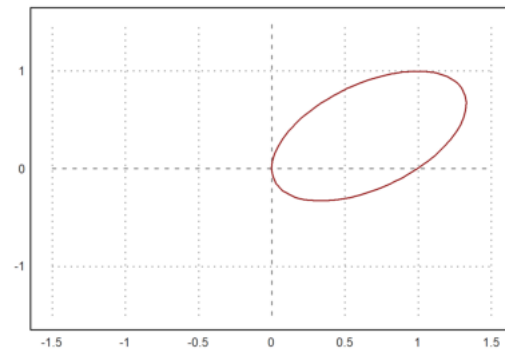
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-125.png

```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



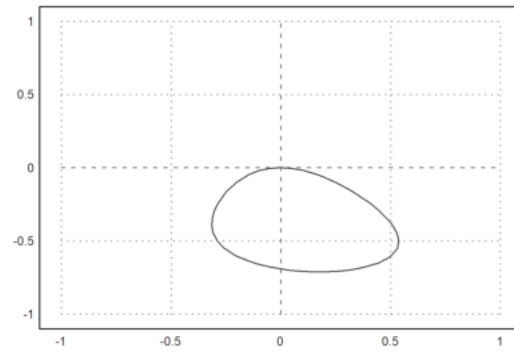
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-126.png

```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



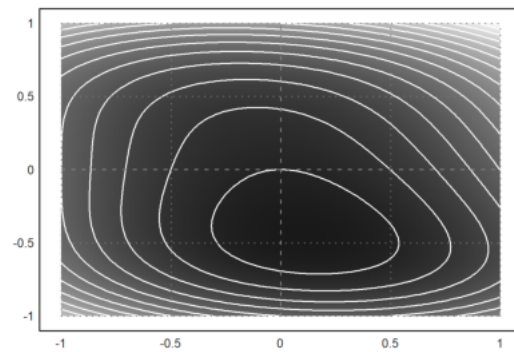
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-127.png

>z=z+normal(size(z))*0.2; >plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):



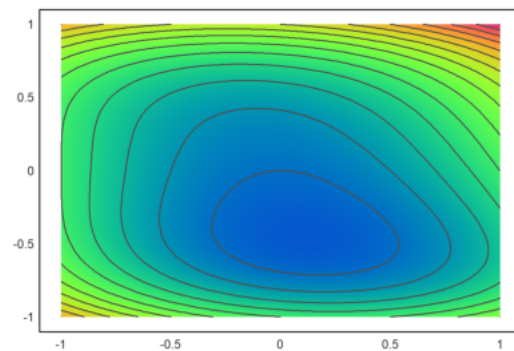
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-128.png

>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):



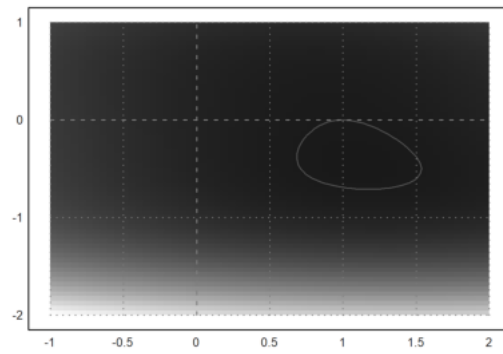
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-129.png

>plot2d("x²+y³+x*y",level=1,r=4,n=100):



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-130.png

>plot2d("x²+2*y²-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):

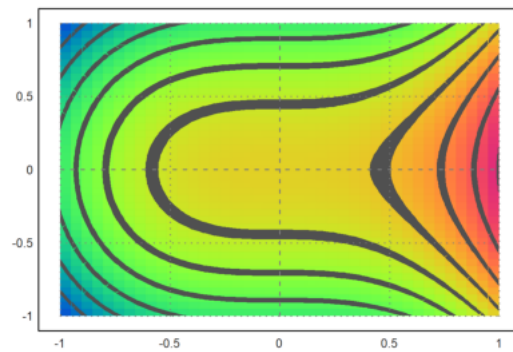


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-131.png

Juga dimungkinkan untuk mengisi set 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.

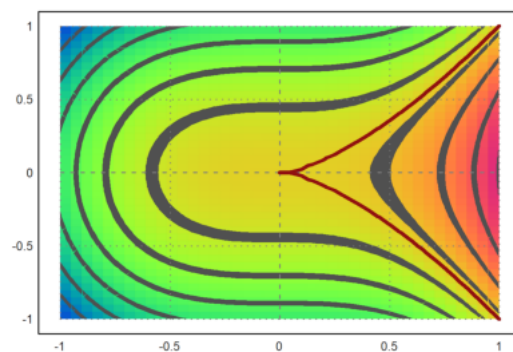
`>plot2d(expr,level=[0;1],style="--",color=blue): //  $0 \leq f(x,y) \leq 1$`



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-132.png

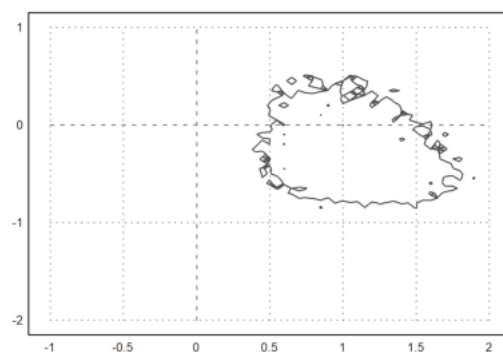
Plot implisit juga dapat menunjukkan rentang level. Kemudian level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Atau, vektor baris sederhana dapat digunakan untuk level, dan parameter `dl` memperluas nilai level ke interval.

`>plot2d("x4+y4",r=1.5,level=[0;1],color=blue,style="/"):`



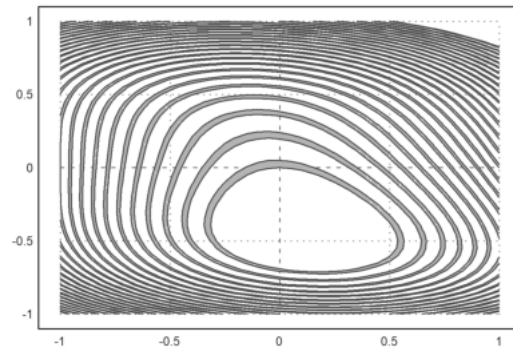
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-133.png

`>plot2d("x2+y3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):`



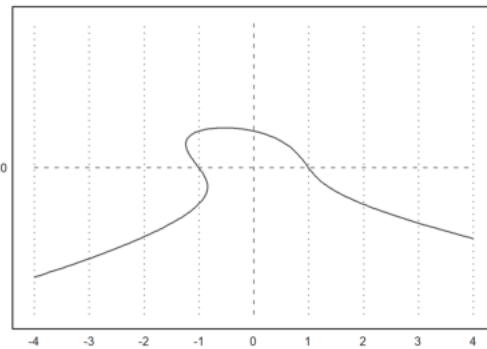
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-134.png

```
>plot2d("x^2+y^3+x*y",level=-10:20,r=2,style="-",dl=0.1,n=100):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-135.png

```
>plot2d("sin(x)*cos(y)",r=pi,>hue,>levels,n=100):
```

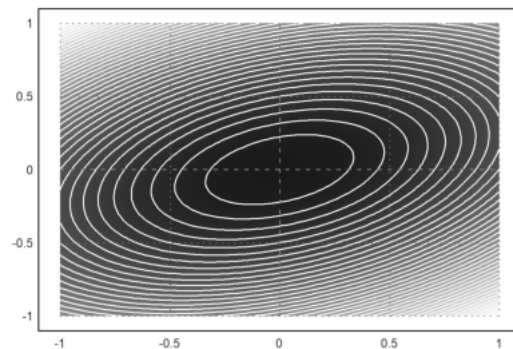


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-136.png

Dimungkinkan juga untuk menandai suatu wilayah

Ini dilakukan dengan menambahkan level dengan dua baris.

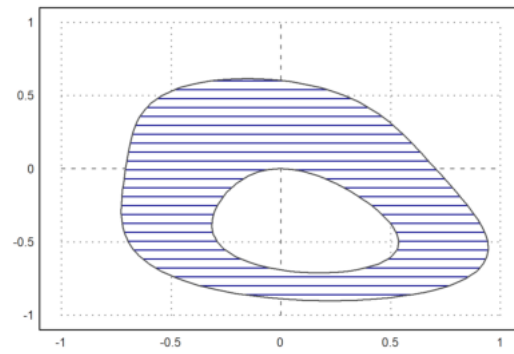
```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
> style="#",color=red,<outline, ...
> level=[-2;0],n=100):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-137.png

Dimungkinkan untuk menentukan level tertentu. Misalnya, kita dapat memplot solusi persamaan seperti

```
>plot2d("x^3-x*y+x^2*y^2",r=6,level=1,n=100):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-138.png

>function starplot1 (v, style="/", color=green, lab=none) ...

```

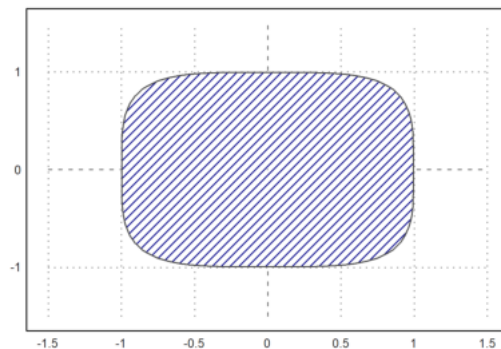
if !holding() then clg; endif;
w=window(); window(0,0,1024,1024);
h=holding(1);
r=max(abs(v))*1.2;
setplot(-r,r,-r,r);
n=cols(v); t=linspace(0,2pi,n);
v=v|v[1]; c=v*cos(t); s=v*sin(t);
cl=barcolor(color); st=barstyle(style);
loop 1 to n
  polygon([0,c[#],c[#+1]], [0,s[#],s[#+1]],1);
  if lab!=none then
    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):



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-139.png

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) ...

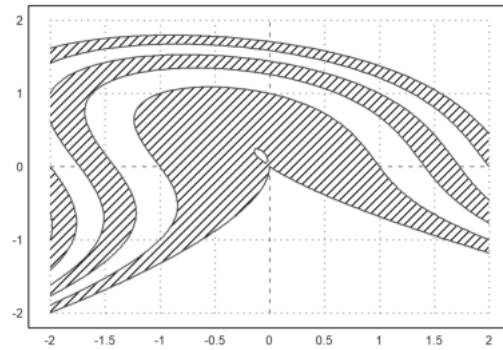
```

{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):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-140.png

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(l);  
endfunction
```

Tekan sembarang tombol untuk menghentikan animasi ini.

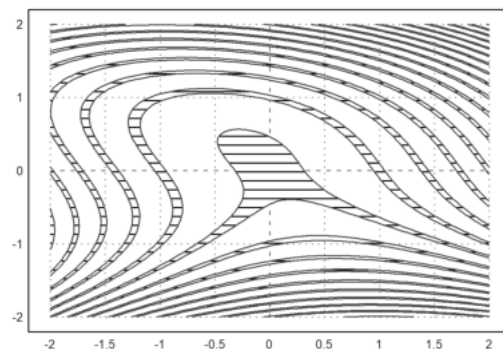
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Plot Logaritmik

EMT menggunakan parameter “logplot” untuk skala logaritmik.

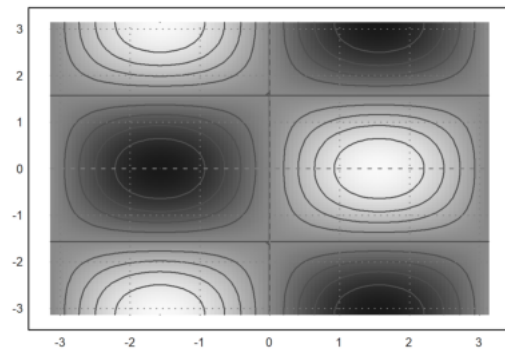
Plot logaritma dapat diplot baik menggunakan skala logaritma dalam y dengan logplot=1, atau menggunakan skala logaritma dalam x dan y dengan logplot=2, atau dalam x dengan logplot=3. * logplot=1: y-logaritma * logplot=2: x-y-logaritma * logplot=3: x-logaritma

```
>plot2d(“exp(x3-x)*x2”,1,5,logplot=1):
```



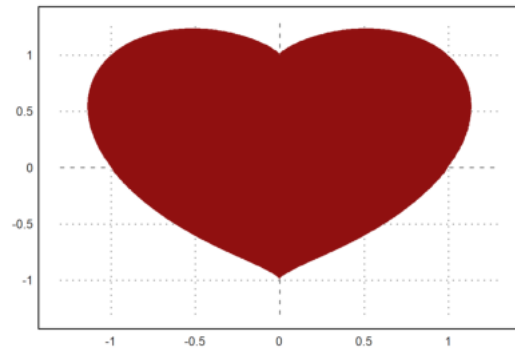
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-141.png

```
>plot2d(“exp(x+sin(x))”,0,100,logplot=1):
```



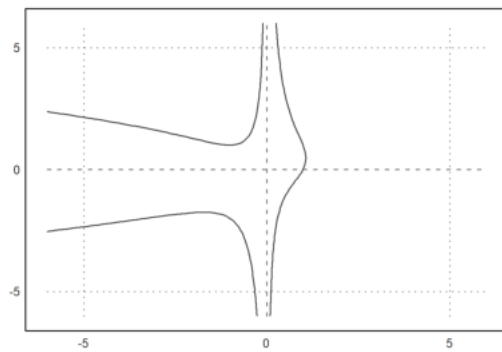
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-142.png

>plot2d("exp(x+sin(x))",10,100,logplot=2):



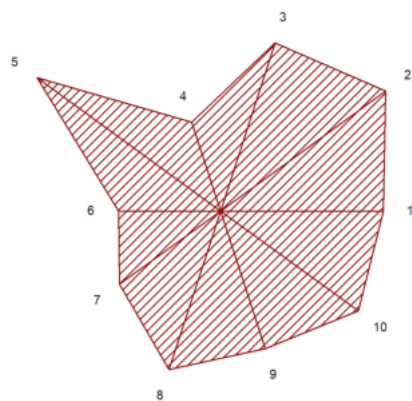
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-143.png

>plot2d("gamma(x)",1,10,logplot=1):



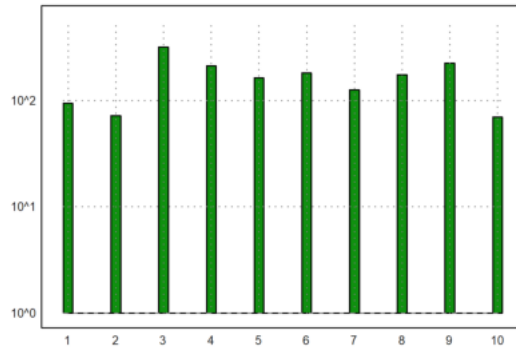
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-144.png

>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-145.png

Ini juga berfungsi dengan plot data. $x=10^{(1:20)}$; $y=x^2-x$; `>plot2d(x,y,logplot=2)`:



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-146.png

Rujukan Lengkap Fungsi `plot2d()` function `plot2d (xv, yv, btest, a,`

`b, c, d, xmin, xmax, r, n, ..`

`logplot`, kisi, bingkai, warna bingkai, kotak, warna, ketebalan, gaya, ..

otomatis, tambahkan, pengguna, delta, poin, titik tambahan, gaya titik, bilah, histogram, ..

distribusi, genap, langkah, sendiri, adaptif, rona, level, kontur, ..

`nc`, terisi, `fillcolor`, `outline`, `title`, `xl`, `yl`, `maps`, `contourcolor`, ..

`contourwidth`, `ticks`, `margin`, `clipping`, `cx`, `cy`, `insimg`, `spectral`, ..

`cgrid`, vertikal, lebih kecil, `dl`, `niveau`, `level`)

Fungsi plot serbaguna untuk plot di pesawat (plot 2D). Fungsi ini dapat melakukan plot fungsi satu variabel, plot data, kurva bidang, plot batang, kisi-kisi bilangan kompleks, dan plot implisit fungsi dua variabel.

Parameter

`x,y` : persamaan, fungsi atau vektor data

`a,b,c,d` : Area plot (default `a=-2,b=2`)

`r` : jika `r` diset, maka `a=cx-r`, `b=cx+r`, `c=cy-r`, `d=cy+r`

`r` dapat berupa vektor `[rx,ry]` atau vektor

`[rx1,rx2,ry1,ry2]`.

`xmin,xmax` : rentang parameter untuk kurva

`auto` : Menentukan `y-range` secara otomatis (default)

`kuadrat` : jika benar, coba pertahankan rentang `x-y` persegi

`n` : jumlah interval (default adaptif)

`kisi` : 0 = tidak ada kisi dan label

1 = sumbu saja,
2 = grid normal (lihat di bawah untuk jumlah garis grid)
3 = sumbu dalam
4 = tidak ada kisi-kisi
5 = kisi penuh termasuk margin
6 = kutu di bingkai
7 = sumbu saja
8 = sumbu saja, sub-centang

`bingkai` : 0 = tanpa bingkai

`framecolor` : warna bingkai dan kisi

`margin` : angka antara 0 dan 0,4 untuk margin di sekitar plot

`warna` : Warna kurva. Jika ini adalah vektor warna,

itu akan digunakan untuk setiap baris matriks plot. Dalam

kasus plot titik, itu harus berupa vektor kolom. Jika vektor baris atau matriks penuh warna digunakan untuk plot titik, itu akan digunakan untuk setiap titik data. ketebalan: ketebalan garis untuk kurva

Nilai ini bisa lebih kecil dari 1 untuk garis yang sangat tipis.

style : Plot style untuk garis, spidol, dan isian.

```
Untuk poin gunakan  
"[]", "<>", ".", "..", "...",  
"*", "+", "|", "-", "o"  
"[]#", "<>#", "o#" (bentuk terisi)  
"[]w", "<>w", "ow" (tidak transparan)
```

Untuk penggunaan garis

```
"-", "--", "-.", ".", "-.", "-.", "->"
```

Untuk poligon terisi atau plot batang gunakan

```
"#", "#0", "0", "/", "\", "\/",  
"+", "|", "-", "t"
```

poin : plot titik tunggal alih-alih segmen garis

addpoints : jika benar, plot segmen garis dan titik

add : menambahkan plot ke plot yang ada

pengguna : aktifkan interaksi pengguna untuk fungsi

delta : ukuran langkah untuk interaksi pengguna

bar : plot batang (x adalah batas interval, y nilai interval)

histogram : memplot frekuensi x dalam n subinterval

distribusi=n : memplot distribusi x dengan n subinterval

even : gunakan nilai antar untuk histogram otomatis.

langkah : memplot fungsi sebagai fungsi langkah (langkah=1,2)

adaptif : gunakan plot adaptif (n adalah jumlah langkah minimal)

level : plot garis level dari fungsi implisit dua variabel

outline : menggambar batas rentang level.

Jika nilai level adalah matriks 2xn, rentang level akan ditarik dalam warna menggunakan gaya isian yang diberikan. Jika garis besar benar, itu akan digambar dalam warna kontur. Dengan menggunakan fitur ini, wilayah

$f(x,y)$ antara batas dapat ditandai.

hue : tambahkan warna hue ke plot level untuk menunjukkan fungsinya

nilai

kontur : Gunakan plot level dengan level otomatis

nc : jumlah garis level otomatis

judul : judul plot (default "")

xl, yl : label untuk sumbu x dan y

lebih kecil : jika >0, akan ada lebih banyak ruang di sebelah kiri untuk label.

vertikal : Mengaktifkan atau menonaktifkan label vertikal. Ini mengubah variabel global

verticallabels secara lokal untuk satu plot. Nilai 1 hanya set vertikal

teks, nilai 2 menggunakan label numerik vertikal pada sumbu y.

terisi : mengisi plot kurva

fillcolor : mengisi warna untuk bar dan kurva yang terisi

outline : batas poligon yang terisi

logplot : mengatur plot logaritma

```
1 = logplot di y,  
2 = plot log di xy,  
3 = logplot dalam x
```

memiliki : Sebuah string, yang menunjuk ke rutinitas plot sendiri. Dengan > pengguna, Anda mendapatkan interaksi pengguna yang sama seperti di plot2d. Rentang akan diatur sebelum setiap panggilan ke fungsi Anda. peta : ekspresi peta (0 lebih cepat), fungsi selalu dipetakan.

contourcolor : warna garis kontur

contourwidth : lebar garis kontur

clipping : mengaktifkan clipping (default adalah true)

judul : Ini dapat digunakan untuk menggambarkan plot. Judul akan muncul di atas jalan cerita. Selain itu, label untuk sumbu x dan y dapat ditambahkan dengan

xl="string" atau yl="string". Label lain dapat ditambahkan dengan

fungsi label() atau labelbox(). Judulnya bisa unicode

string atau gambar rumus LATEX.

jaringan : Menentukan jumlah garis grid untuk plot grid yang kompleks.

Harus merupakan pembagi dari ukuran matriks dikurangi 1 (jumlah subinterval). cgrid dapat berupa vektor [cx,cy].

Ringkasan

Fungsi dapat merencanakan

* ekspresi, koleksi panggilan atau fungsi dari satu variabel, * kurva parametrik, * data x terhadap data y, * fungsi implisit, * petak batang, * jaringan kompleks, * poligon.

Jika fungsi atau ekspresi untuk xv diberikan, plot2d() akan menghitung nilai dalam rentang yang diberikan menggunakan fungsi atau ekspresi. Itu ekspresi harus berupa ekspresi dalam variabel x. Rentang harus didefinisikan dalam parameter a dan b kecuali rentang default [-2,2] harus digunakan. Rentang y akan dihitung secara otomatis, kecuali c dan d ditentukan, atau radius r, yang menghasilkan kisaran [-r,r] untuk x dan y. Untuk plot fungsi, plot2d akan menggunakan evaluasi adaptif fungsi secara default. Untuk mempercepat plot untuk fungsi yang rumit, matikan ini dengan <adaptif, dan opsional mengurangi jumlah interval n. Selain itu, plot2d() akan secara default menggunakan pemetaan. Yaitu, itu akan menghitung elemen plot untuk elemen. Jika ekspresi atau fungsi Anda dapat menangani a vector x, Anda dapat menonaktifkannya dengan <maps untuk evaluasi yang lebih cepat.

Perhatikan bahwa plot adaptif selalu dihitung elemen untuk elemen.

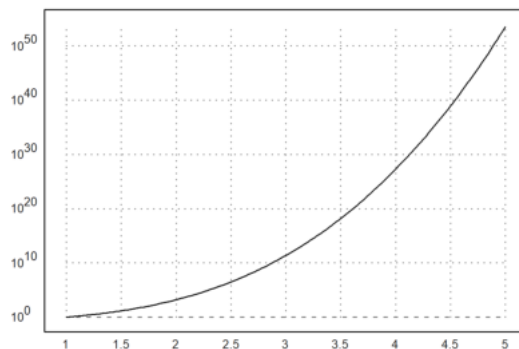
Jika fungsi atau ekspresi untuk xv dan untuk yv ditentukan, plot2d() akan menghitung kurva dengan nilai xv sebagai koordinat x dan nilai yv sebagai koordinat y. Dalam hal ini, rentang harus didefinisikan untuk parameter menggunakan xmin, xmax. Ekspresi yang terkandung dalam string harus selalu ekspresi dalam variabel parameter x.

Soal

1. Buatlah plot 2 dimensi dari fungsi

dengan $a=4$ dan interval dari $x=-1$ sampai $x=1$. Lalu, buatlah plot tersebut dalam grid 3 dan ketebalan 3!

`>a:=4; expr &=exp(a*(x^2)/a); >plot2d(expr,-1,1,grid=3,thickness=3):`

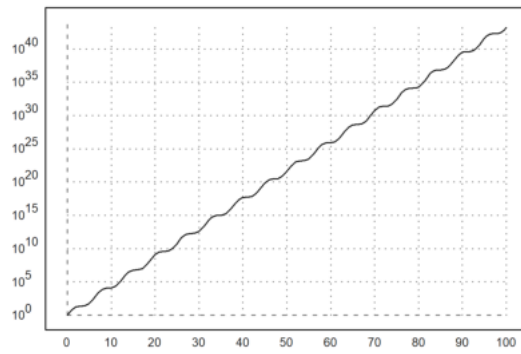


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-147.png

2. Buatlah plot 2 dimensi dari fungsi

dengan nilai parameternya adalah 5, interval x nya dari 1 sampai 2

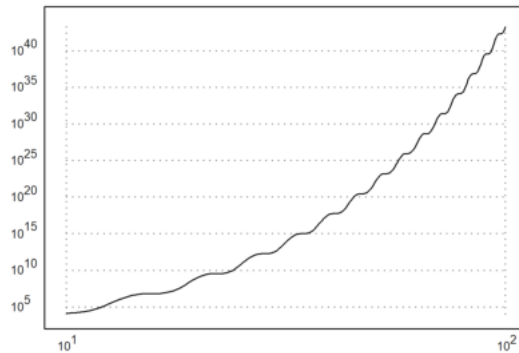
`>a=5; expr &=(x^2-a*x); >plot2d(expr,1,2):`



3. Selidi dimanakah

fungsi $f(x)$ berikut naik dan turun

`>expr &= ((1/3)*x^3-x^2-3*x+4); >plot2d(expr,-4,5):`



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot2D-149.png

Nama : Alya Putri Medilasari

NIM : 23030630018

Kelas: Matematika B 2023

Menggambar Plot 3D dengan EMT

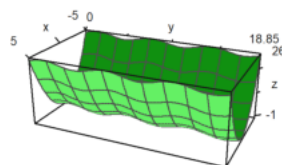
Ini adalah pengenalan plot 3D di Euler. Kami membutuhkan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

Euler menggambar fungsi seperti itu menggunakan algoritma pengurutan untuk menyembunyikan bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Defaultnya adalah dari kuadran x-y positif ke arah asal $x = y = z = 0$, tetapi sudut = 0 ? melihat dari arah sumbu y. Sudut pandang dan ketinggian dapat diubah.

Euler bisa merencanakan * permukaan dengan bayangan dan garis level atau rentang level, * awan poin, * kurva parametrik, * permukaan implisit.

Plot 3D dari suatu fungsi menggunakan plot3d. Cara termudah adalah dengan memplot ekspresi dalam x dan y. Parameter r mengatur kisaran plot sekitar (0,0).

`>aspect(1.5); plot3d("x^2+sin(y)",-5,5,0,6*pi):`



Fungsi Dua

Variabel

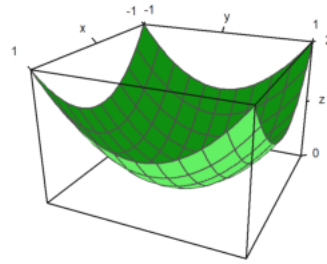
Untuk grafik suatu fungsi, gunakan * ekspresi sederhana dalam x dan y, * nama fungsi dari dua variabel l * atau matriks data.

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
- $\text{grid} = 10$, $\text{grid} = [10,10]$: jumlah garis kisi di setiap arah.

Kami menggunakan default $n = 40$ dan $\text{grid} = 10$.

`>plot3d("x2+y2"):`



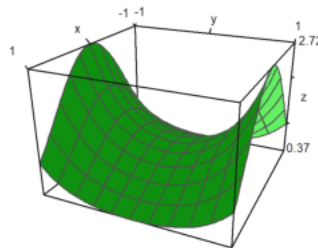
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-002.png

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(-x2+y2)",>user, ...`

`> title="Turn with the vector keys (press return to finish)":`
Turn with the vector keys (press return to finish)



Rentang plot untuk fungsi dapat ditentukan dengan * a, b: rentang-x * c, d: rentang y * r: persegi simetris di sekitar (0,0). * n: jumlah subinterval untuk plot.

Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

`fscale`: menskalakan ke nilai fungsi (defaultnya adalah `<fscale>`).

`scale`: angka atau vektor 1x2 untuk skala ke arah x dan y.

`frame`: jenis bingkai (default 1)

`>plot3d("exp(-(x2+y2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):`



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-004.png

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.

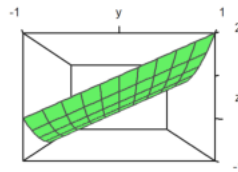
```
>view
```

```
[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.

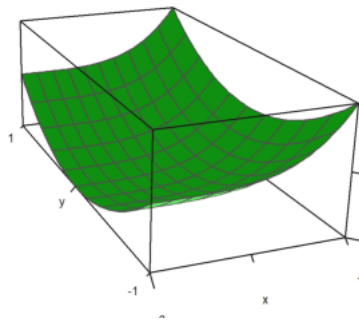
```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-005.png

Plot selalu terlihat ke tengah plot kubus. Anda dapat memindahkan pusat dengan parameter tengah.

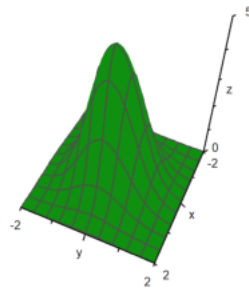
```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...  
> center=[0.4,0,0],zoom=5):
```



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.

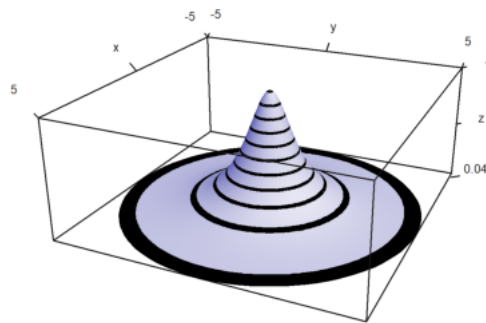
Jika Anda mematakannya dengan scale = false, Anda harus berhati-hati, bahwa plot masih pas dengan jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan pusat.

```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...  
> center=[0,0,-2],frame=3):
```

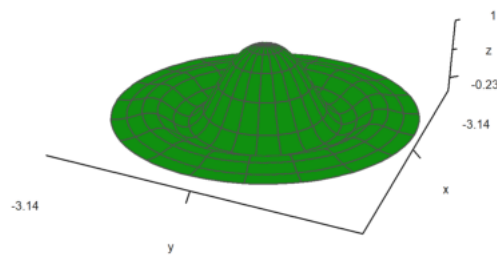


Plot kutub juga tersedia. The parameter `polar= true` menggambarkan plot kutub. Fungsi tersebut harus tetap merupakan fungsi dari x dan y . Parameter “`fscale`” menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi diskalakan agar sesuai dengan kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
> fscale=2,>hue,n=100,zoom=4,>contour,color=blue);
```

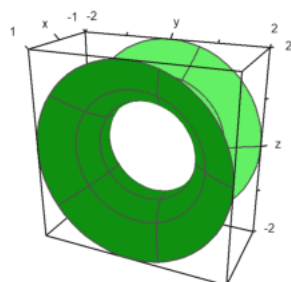


```
>function f(r) := exp(-r/2)*cos(r); ...
> plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4);
```



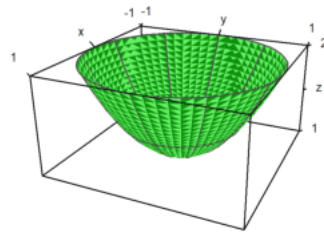
Parameter memutar memutar fungsi dalam x di sekitar sumbu x . * `rotate = 1`: Menggunakan sumbu x * `rotate = 2`: Menggunakan sumbu z

```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



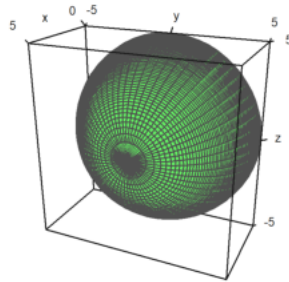
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-010.png

```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```



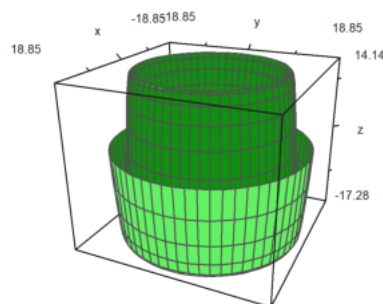
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-011.png

```
>plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-012.png

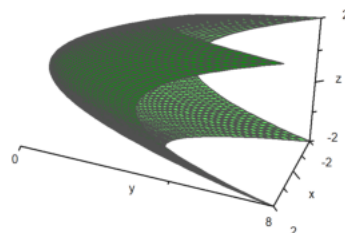
```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```



Ini adalah plot

dengan tiga fungsi

```
>plot3d("x","x^2+y^2","y",r=2,zoom=3.5,frame=3):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-014.png

Plot Kontur

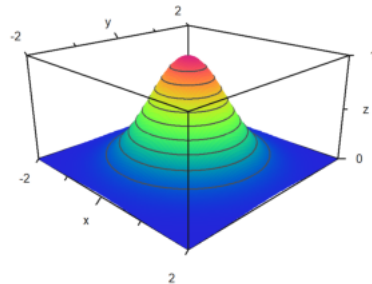
Untuk plot, Euler menambahkan garis kisi. Alih-alih, dimungkinkan untuk menggunakan garis level dan corak satu warna atau corak warna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan shading. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah / cyan.

-> hue: Mengaktifkan bayangan terang, bukan kabel. -> contour: Membuat plot garis kontur otomatis pada plot. - level = ... (atau level): Vektor nilai untuk garis kontur.

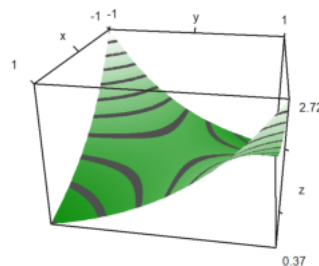
Standarnya adalah level = "auto", yang menghitung beberapa baris level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut, kami menggunakan kisi yang lebih halus untuk 100x100 titik, menskalakan fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...
>>contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):
```



```
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```

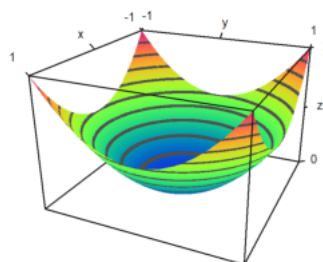


Bayangan default menggunakan warna abu-abu. Tetapi berbagai spektrum warna juga tersedia.

-> spectral: Menggunakan skema spektral default - color = ...: Menggunakan warna khusus atau skema spektral

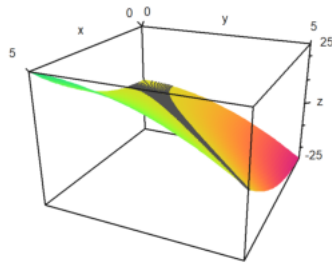
Untuk plot berikut, kami menggunakan skema spektral default dan menambah jumlah poin untuk mendapatkan tampilan yang sangat mulus.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```



Alih-alih garis level otomatis, kami juga dapat mengatur nilai garis level. Ini akan menghasilkan garis level tipis, bukan rentang level.

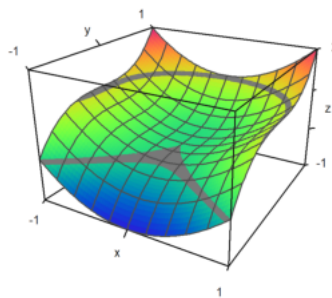
```
>plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=redgreen):
```



Dalam plot berikut, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas level sebagai kolom.

Selain itu, kami melapisi kisi dengan 10 interval di setiap arah.

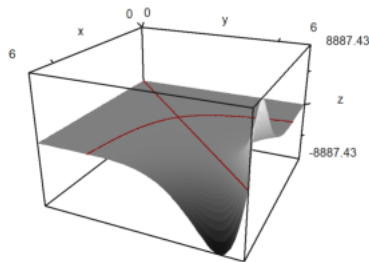
```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...
>>spectral,angle=30°,grid=10,contourcolor=gray):
```



Dalam contoh berikut, kami memplot set, di mana

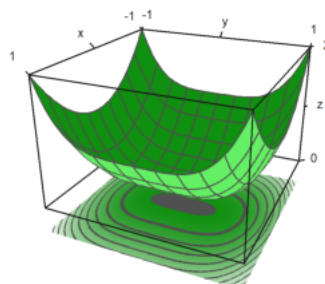
Kami menggunakan satu garis tipis untuk garis level.

```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```



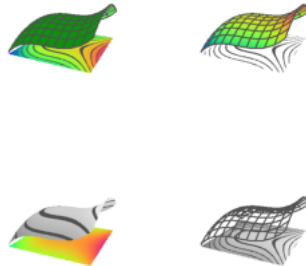
Dimungkinkan untuk menunjukkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

```
>plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2):
```



Berikut beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi.

```
>figure(2,2); ...
> expr="y3-x2"; ...
> 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):
```

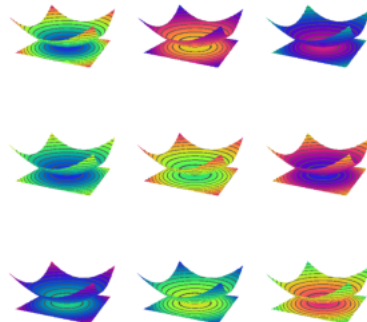


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-022.png

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("x2+y2",spectral=i,>contour,>cp,<frame,zoom=4); ...
> end; ...
> figure(0):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-023.png

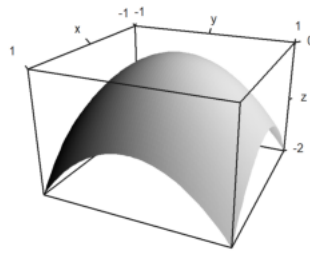
Sumber cahaya dapat diubah dengan l dan tombol kursor selama interaksi pengguna. Itu juga dapat diatur dengan parameter.

- 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("-x2-y2", ...
> hue=true,light=[0,1,1],amb=0,user=true, ...
> title="Press l and cursor keys (return to exit)");
```

Press I and cursor keys (return to exit)



Parameter warna

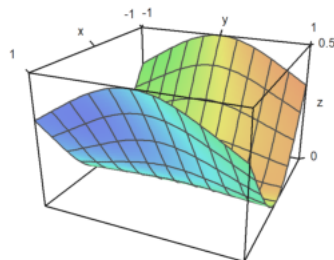
mengubah warna permukaan. Warna garis level juga dapat diubah.

```
>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):
```



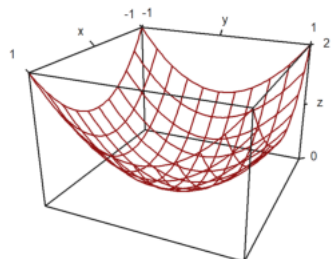
Warna 0 memberikan efek pelangi khusus.

```
>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

$$f(x, y, z) = 0$$

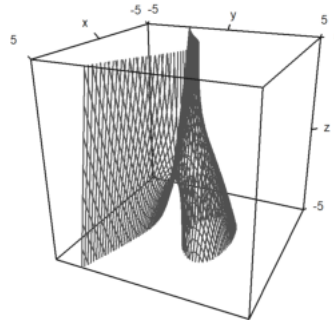
dapat divisualisasikan dalam potongan sejajar dengan bidang x - y -, x - z -, dan y - z .

- 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

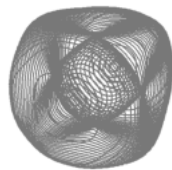
$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

`>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):`

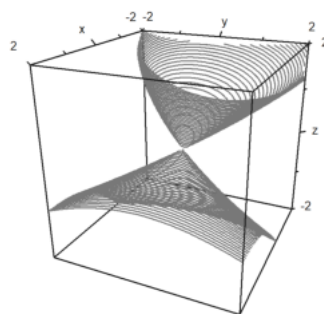


`>c=1; d=1;`

`>plot3d("((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)-d",r=2,
<frame,>implicit,>user):`



`>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):`



Plotting 3D Data

Sama seperti `plot2d`, `plot3d` menerima data. Untuk objek 3D, Anda perlu memberikan matriks nilai x , y dan z , atau tiga fungsi atau ekspresi $f_x(x, y)$, $f_y(x, y)$, $f_z(x, y)$.

$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

Karena x , y , z adalah matriks, kita asumsikan bahwa (t, s) berjalan melalui kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di luar angkasa.

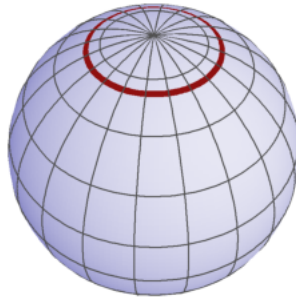
Anda dapat menggunakan bahasa matriks Euler untuk menghasilkankoordinat secara efektif.

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°):

```



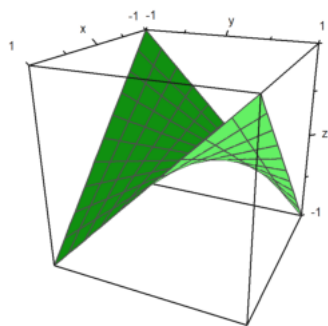
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-034.png

Berikut adalah contoh grafik dari suatu fungsi.

```

>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):

```

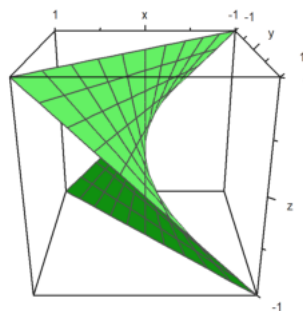


Namun, kita bisa membuat semua jenis permukaan. Ini adalah permukaan yang sama sebagai suatu fungsi

```

>plot3d(t*s,t,s,angle=180°,grid=10):

```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-036.png

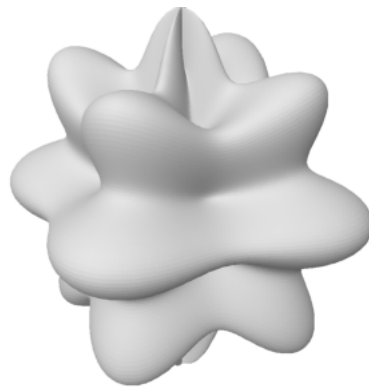
Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

Dalam contoh berikut, kami membuat tampilan berbayang dari bola yang terdistorsi. Koordinat biasa untuk bola adalah dengan Kami menyimpangkan ini dengan sebuah faktor

```

>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
> d=1+0.2*(cos(4*t)+cos(8*s)); ...
> plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
> light=[1,0,1],frame=0,zoom=5):

```

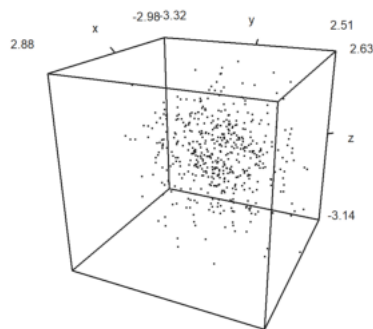


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-037.png

Tentu saja, point cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor sebagai koordinat titik.

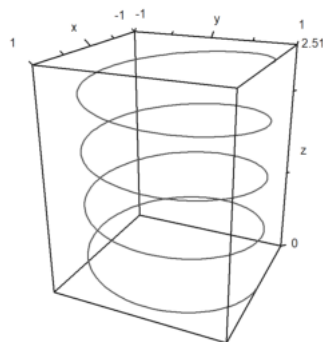
Gayanya sama seperti di plot2d dengan points = true;

```
>n=500; ...
> plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

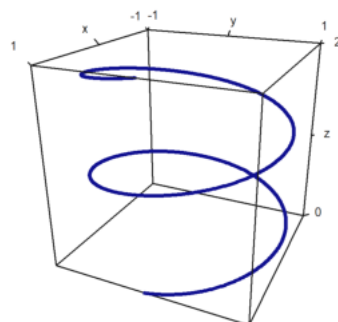


Juga dimungkinkan untuk memplot kurva dalam 3D. Dalam kasus ini, lebih mudah untuk menghitung sebelumnya titik-titik kurva. Untuk kurva di bidang kita menggunakan urutan koordinat dan parameter wire = true.

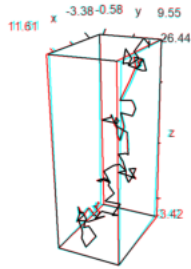
```
>t=linspace(0,8pi,500); ...
> plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
> linewidth=3,wirecolor=blue):
```

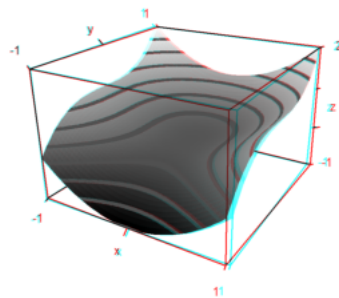


```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



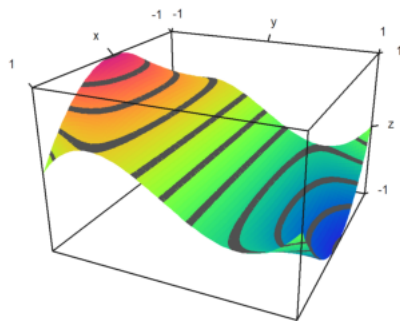
EMT juga dapat memplot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda membutuhkan red/cyan glasses.

```
> 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):
```

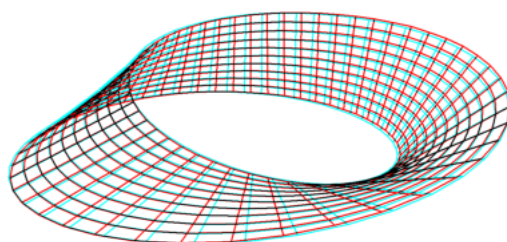


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-043.png

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.

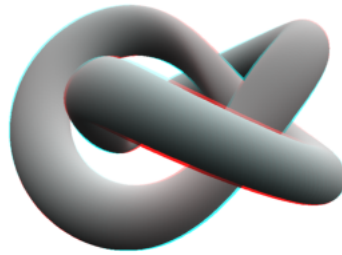
```
> u=linspace(-1,1,10); v=linspace(0,2*pi,50); ...
> X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
> plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-044.png

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

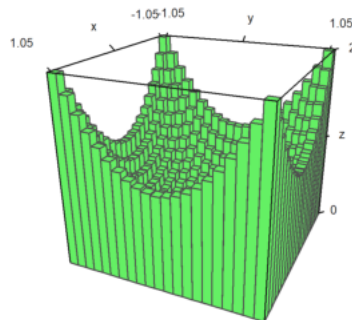
Plot batang juga dimungkinkan. Untuk ini, kami harus menyediakan

- 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.

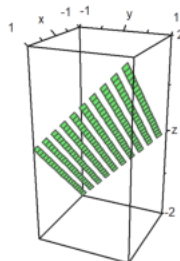
```
>x=-1:0.1:1; y=x'; z=x2+y2; ...  
> xa=(x|1.1)-0.05; ya=(y|1.1)-0.05; ...  
> plot3d(xa,ya,z,bar=true):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-046.png

Dimungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

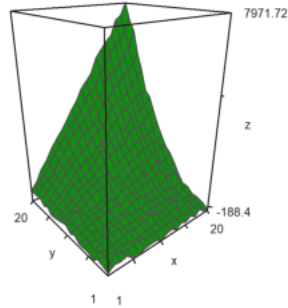
```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...  
> plot3d(x,y,z,disconnect=2:2:20):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-047.png

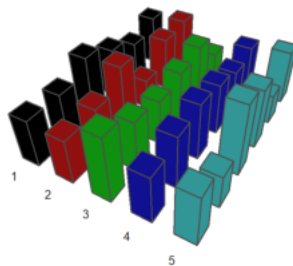
Jika memuat atau membuat matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke $[-1,1]$ dengan skala (M), atau menskalakan matriks dengan `zscale`. Ini dapat dikombinasikan dengan faktor penskalaan individu yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...
> plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-048.png

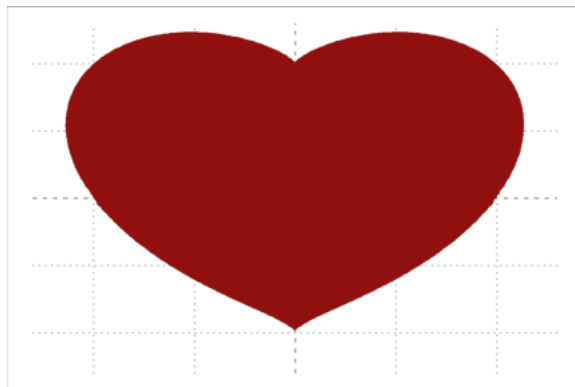
```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
> loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
> columnsplot3d(v',scols=1:5,ccols=[1:5]):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-049.png

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):
```



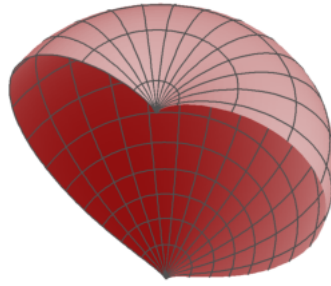
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-050.png

>ekspresi $(x^2+y^2-1)^3-x^2*y^3$; ekspresi $(y^2+x^2-1)^3-x^2*y^3$ Kami ingin memutar kurva jantung di sekitar sumbu y. Inilah ungkapan yang mendefinisikan hati:

Selanjutnya kita atur

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; f r (r, a)
 $(r^2-1)^3 + \frac{(\sin(5a)-\sin(3a)-2\sin a)r^5}{16}$  Hal ini memungkinkan untuk menentukan fungsi numerik, yang menyelesaikan r, jika a diberikan. Dengan fungsi itu kita dapat memplot jantung yang berubah sebagai permukaan parametrik.
```

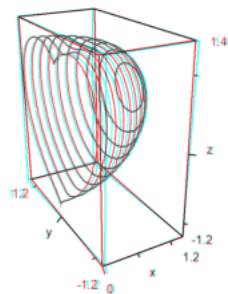
```
>function map f(a) := bisect("fr",0,2;a); ...
> t=linspace(-pi/2,pi/2,100); r=f(t); ...
> s=linspace(pi,2pi,100); ...
> plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
> >hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```



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

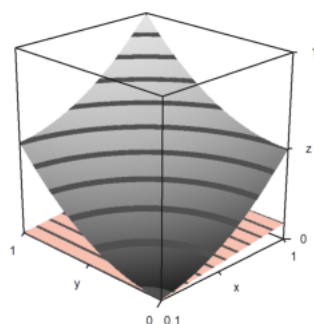
Fungsi plot3d bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda bisa mendapatkan plot berbingkai dari objek apa pun yang Anda suka.

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.

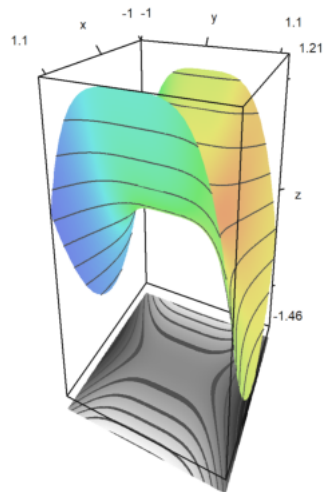
```
>x=-1:0.02:1.1; y=x'; z=x2-y4;
```

```
>function myplot(x,y,z) ...
```

```
>
```

```
    zoom(2);  
    wi=fullwindow();  
    plotcontourplane(x,y,z,level="auto",hue,add,color=whit  
    window(wi);  
    reset();  
endfunction
```

```
>myplot(x,y,z):
```



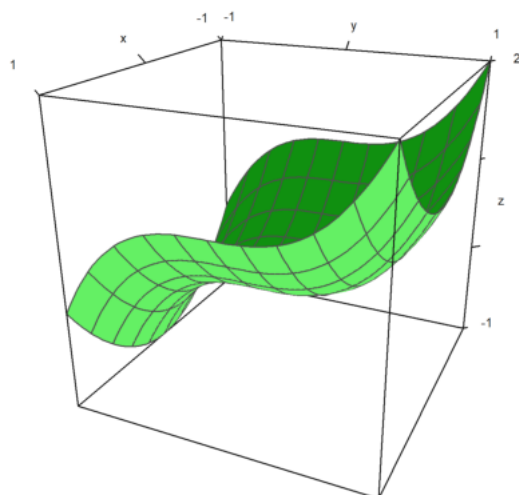
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-056.png

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("x2+y3"); ...  
> rotate("testplot"); testplot();
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-057.png

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 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.

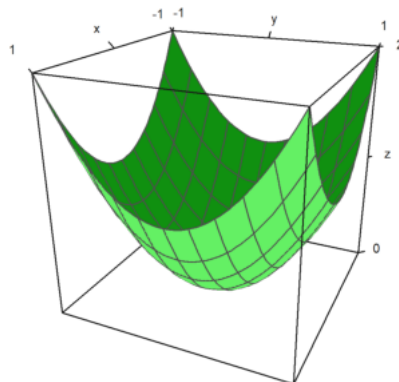
```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

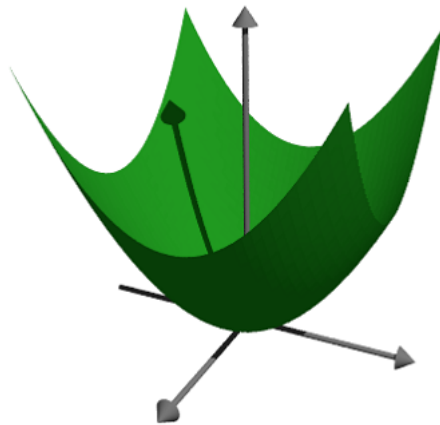
Untuk pertamakali, kami memplot fungsi sederhana. Perintah berikut menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk menelusuri file ini.

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.

```
>plot3d("x2+y2",zoom=2):
```



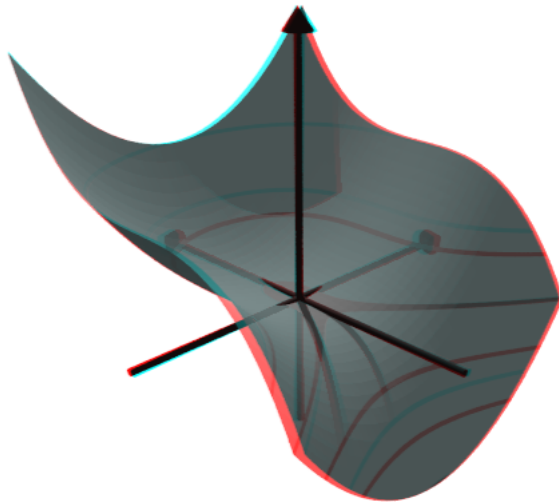
```
>pov3d("x^2+y^2",zoom=3);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-059.png

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=-45°,>anaglyph, ...  
> look=povlook(cyan,0.2),level=-1:0.5:1,zoom=3.8);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-060.png

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);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-061.png

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]);
```

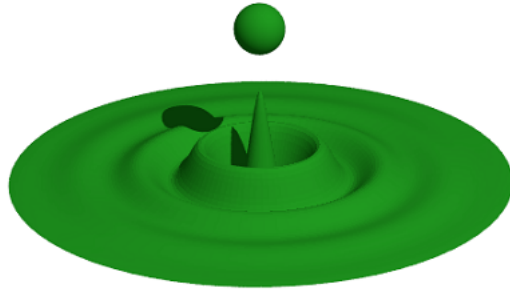


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-062.png

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);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-063.png

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 &= x2*y3
```

$$\begin{matrix} 2 & 3 \\ x & y \end{matrix}$$

Persamaan permukaannya adalah $[x, y, Z]$. Kami menghitung dua turunan menjadi x dan y dari ini dan mengambil produk silang sebagai normal.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

Kami mendefinisikan normal sebagai produk silang dari turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

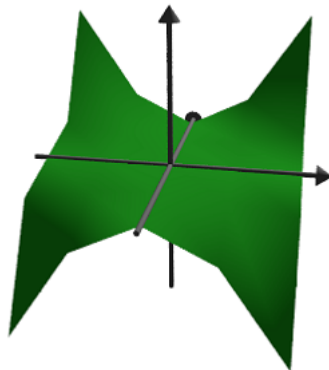
$$\begin{bmatrix} -2xy^3, -3x^2y^2, 1 \end{bmatrix}$$

We use only 25 points.

```
>x=-1:0.5:1; y=x';
```

```
>pov3d(x,y,Z(x,y),angle=10°, ...
```

```
> xv=NX(x,y),yv=NY(x,y),zv=NZ(x,y),<shadow);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-064.png

Berikut ini adalah simpul Trefoil yang dilakukan oleh A. Busser di Povray. Ada versi perbaikannya dalam contoh.

Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normal bagi kami. Pertama, tiga fungsi koordinat sebagai ekspresi simbolik.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
```

```
> Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
```

```
> Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunannya menjadi x dan y .

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang normal, yang merupakan produk persilangan dari dua turunannya.

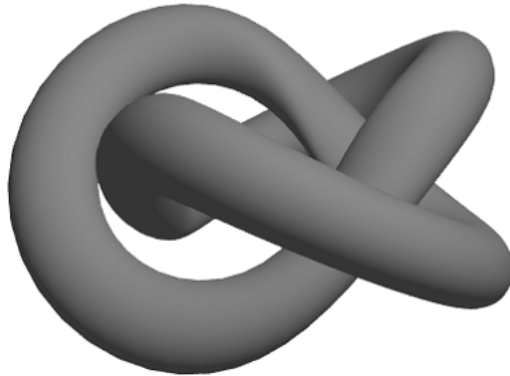
```
>dn &= crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100);
```

Vektor normal adalah evaluasi dari ekspresi simbolik $dn[i]$ untuk $i=1,2,3$. Sintaks untuk ini adalah `&"expresi"` (parameter). Ini adalah alternatif metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX , NY , NZ terlebih dahulu.

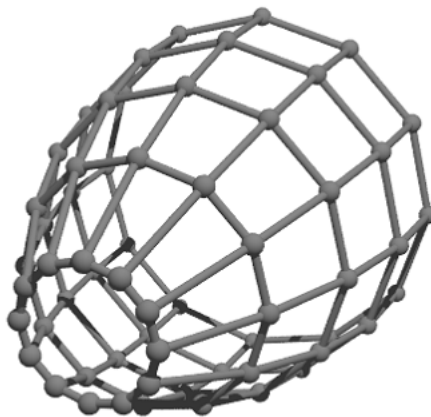
```
>pov3d(X(x,y),Y(x,y),Z(x,y),axis=0,zoom=5,w=450,h=350, ...  
> <shadow,look=povlook(gray), ...  
> xv=&"dn[1]"(x,y), yv=&"dn[2]"(x,y), zv=&"dn[3]"(x,y));
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-065.png

Kami juga dapat membuat grid dalam 3D.

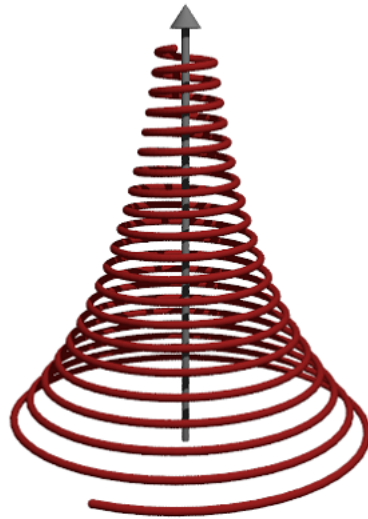
```
>povstart(zoom=4); ...  
> x=-1:0.5:1; r=1-(x+1)^2/6; ...  
> t=(0°:30°:360°); y=r*cos(t); z=r*sin(t); ...  
> writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...  
> povend();
```



kurva dimungkinkan.

Dengan `povgrid()`,

```
>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();
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-067.png

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.

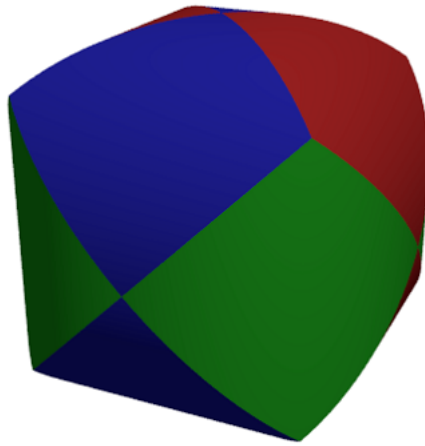
```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

```
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;
```

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) ...
```

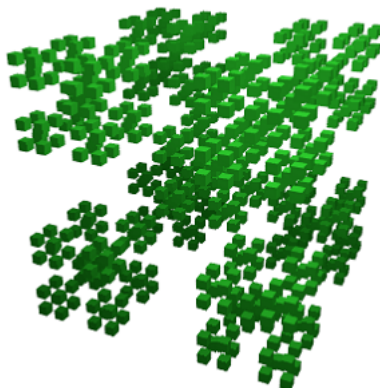
```
>
```

```
    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();
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-069.png

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.

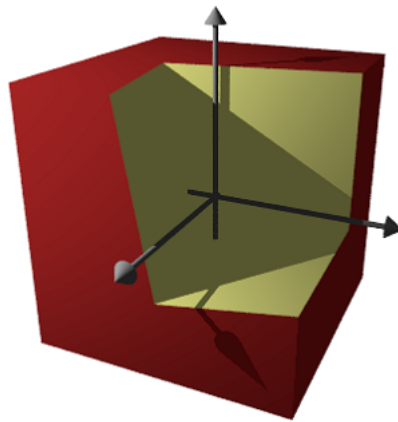
```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...  
> rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

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();
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-070.png

Fungsi Implisit

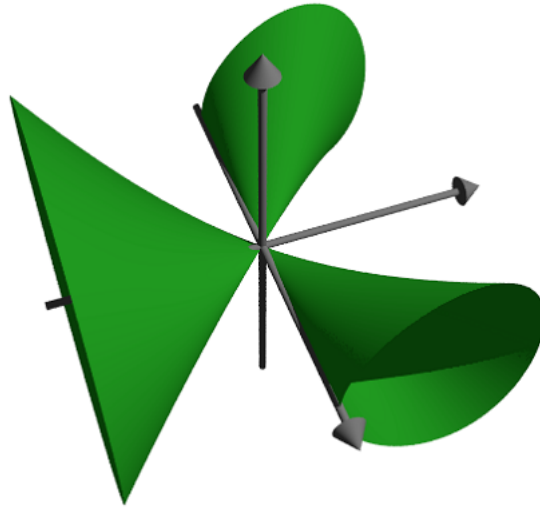
Povray dapat memplot himpunan di mana $f(x, y, z) = 0$, seperti parameter implisit di plot3d. Namun, hasilnya terlihat jauh lebih baik.

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();
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-071.png

Objek Jaring

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarinya dengan informasi tambahan.

Kami ingin memaksimalkan xy di bawah kondisi $x + y = 1$ dan mendemonstrasikan sentuhan tangensial dari garis level.

```
>povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kami tidak dapat menyimpan objek dalam string seperti sebelumnya, karena terlalu besar. Jadi kami mendefinisikan objek dalam file Povray menggunakan #declare. Fungsi povtriangle () melakukan ini secara otomatis. Ia dapat menerima vektor normal seperti pov3d ().

Yang berikut ini mendefinisikan objek mesh, dan langsung menulisnya ke dalam file.

```
>x=0:0.02:1; y=x'; z=x*y; vx=-y; vy=-x; vz=1;
```

```
>mesh=povtriangles(x,y,z,“”,vx,vy,vz);
```

Sekarang kami mendefinisikan dua disk, yang akan berpotongan dengan permukaan.

```
>cl=povdisc([0.5,0.5,0],[1,1,0],2); ...
```

```
>ll=povdisc([0,0,1/4],[0,0,1],2);
```

Tulis permukaan dikurangi dua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll],povlook(green)));
```

Tuliskan dua persimpangan tersebut.

```
>writeln(povintersection([mesh,cl],povlook(red))); ...
```

```
> writeln(povintersection([mesh,ll],povlook(gray)));
```

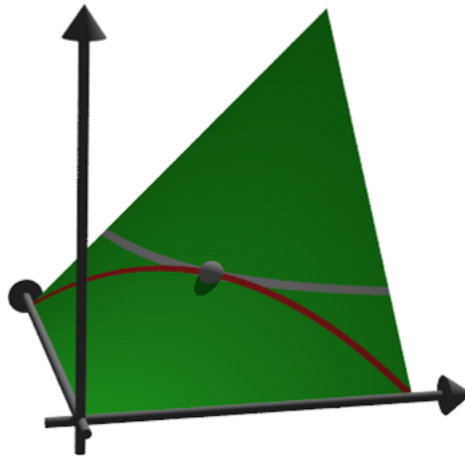
Tulis titik maksimal.

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));
```

Tambahkan sumbu dan selesai.

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...
```

```
> povend();
```

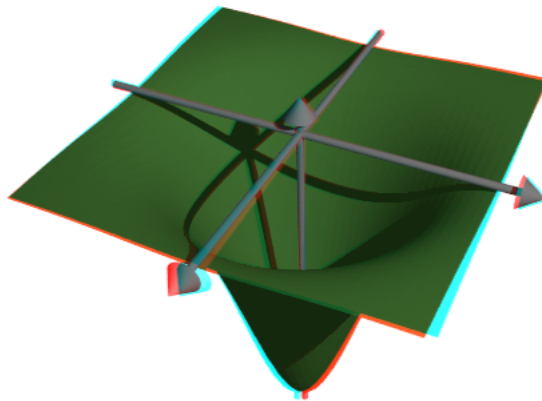


Anaglyph di
Povray Untuk menghasilkan anaglyph untuk kacamata merah / cyan, Povray harus dijalankan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph ().

Tentu saja, Anda memerlukan kaca mata merah / cyan untuk melihat contoh berikut dengan benar.

Fungsi pov3d () memiliki tombol sederhana untuk menghasilkan anaglyph.

```
>pov3d("-exp(-x^2-y^2)/2",r=2,height=45°,>anaglyph, ...
> center=[0,0,0.5],zoom=3.5);
```



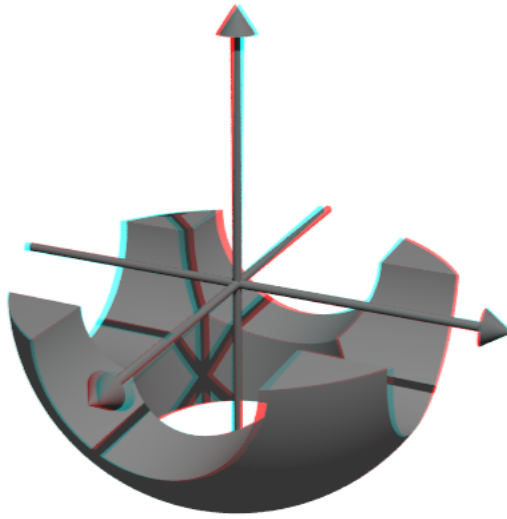
Jika Anda membuat adegan dengan objek, Anda perlu memasukkan pembuatan adegan ke dalam fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

```
>function myscene ...
```

```
    s=povsphere(povc,1);
    cl=povcylinder(-povz,povz,0.5);
    clx=povobject(cl,rotate=xrotate(90°));
    cly=povobject(cl,rotate=yrotate(90°));
    c=povbox([-1,-1,0],1);
    un=povunion([cl,clx,cly,c]);
    obj=povdifference(s,un,povlook(red));
    writeln(obj);
    writeAxes();
endfunction
```

Fungsi povanaglyph () melakukan semua ini. Parameternya seperti di povstart () dan povend () digabungkan.

```
>povanaglyph("myscene",zoom=4.5);
```



Mendefinisikan

Objek Sendiri

Antarmuka povray Euler berisi banyak objek. Tetapi Anda tidak dibatasi untuk ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau merupakan objek yang sama sekali baru.

Kami mendemonstrasikan torus. Perintah Povray untuk ini adalah "torus". Jadi kami mengembalikan string dengan perintah ini dan 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 &lt;0.8,0,0&gt;
}
```

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 dan menunjukkan poin yang layak dan optimal dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
```

```
>b=[10,10,10,10]';
```

```
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini memiliki solusi.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0, 1, 0.5]
```

Yes, it has.

Next we define two objects. The first is the plane

```
>function oneplane (a,b,look=“”) ...
```

```
    return povplane(a,b,look)
endfunction
```

Kemudian kami mendefinisikan perpotongan dari semua setengah spasi dan sebuah kubus.

```
>function adm (A, b, r, look=“”) ...
```

```
    ol=[];
    loop 1 to rows(A); ol=ol|oneplane(A[#],b[#]); 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)));
```

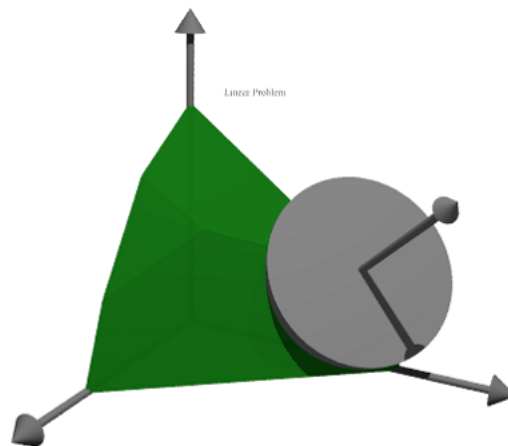
Dan ada kesalahan di arah optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah objek 3D. Kita perlu menempatkan dan memutarinya sesuai dengan pandangan kita.

```
>writeln(povtext(“Linear Problem”,[0,0.2,1.3],size=0.05,rotate=125°)); ...
```

```
> povend();
```



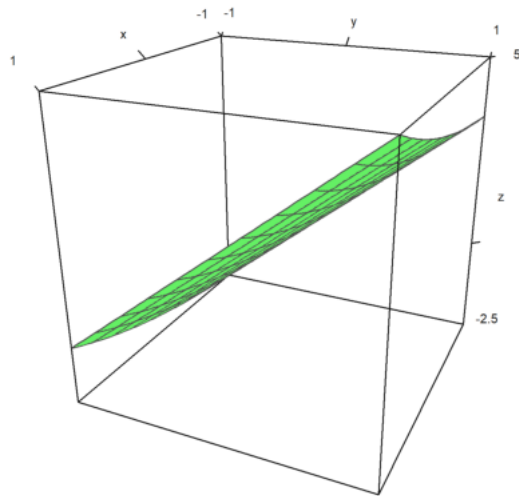
Latihan Soal

1. Buatlah grafik

$$f(x,y) = 2^x + 3y$$

```
>function f(x,y):= 2^x+3*y
```

```
>plot3d(“f”):
```



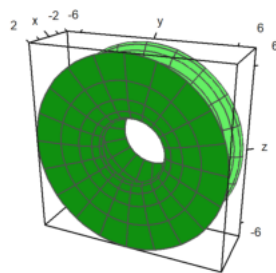
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-077.png

2. Buatlah grafik

$$f(x) = x^3 - 3x + 4$$

dengan $a=-2$, $b=2$, dan $\text{grid}=18$

`>plot3d("x^3-3*x+4",a=-2,b=2,rotate=true,grid=18):`

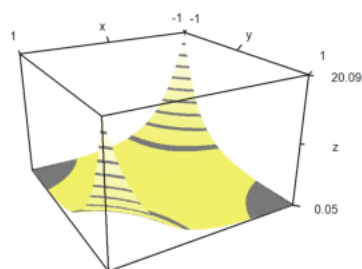


images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-079.png

3. Buatlah grafik plot kontur

$$f(x) = e^{3xy}$$

`>plot3d("exp(3*x*y)",angle=150,>contour,color=yellow):`

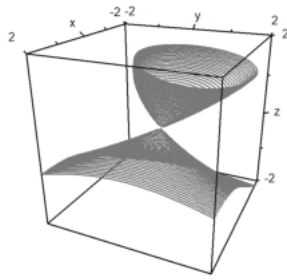


implisit

$$x^2 + y^2 + 5xz + 3z^3$$

`>plot3d("x^2+y^2+5*x*z+3*z^3",>implicit,r=2,zoom=2.2):`

4. Buat grafik plot



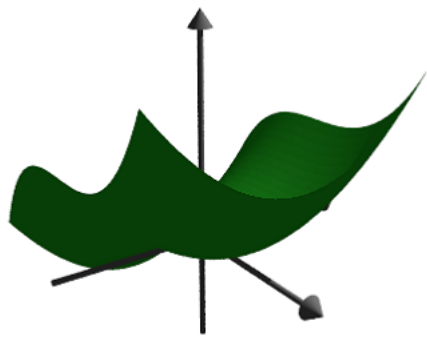
images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-083.png

5. Buatlah plot 3D dari fungsi

$$f(x, y) = x^3 + y^2$$

dengan zoom 3 dan angle 55 derajat menggunakan povray

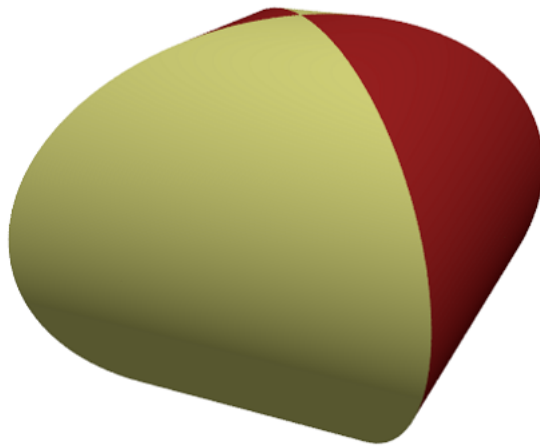
```
>pov3d("x^3+y^2",zoom=3,angle=55°);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-085.png

6. Buatlah gabungan 2 silinder dengan fungsi povx() berwarna merah dan povz() berwarna kuning dan zoom 4

```
>povstart(zoom=4)
>c1 = povcylinder(-povx,povx,1,povlook(red));
>c2 = povcylinder(-povz,povz,1,povlook(yellow));
>writeln(povintersection([c1,c2]));
>povend();
```



images/Alya%20Putri%20Medilasari_23030630018_EMTPlot3D-086.png

Kalkulus dengan EMT

Materi Kalkulus mencakup di antaranya 1. Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi) Limit Fungsi, 2. Turunan Fungsi, 3. Integral Tak Tentu, 4. Integral Tentu dan Aplikasinya, 5. 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: 1. Menggunakan format `nama_fungsi := rumus fungsi` (untuk fungsi numerik), 2. Menggunakan format `nama_fungsi &= rumus fungsi` (untuk fungsi simbolik, namun dapat dihitung secara numerik), 3. Menggunakan format `nama_fungsi &&= rumus fungsi` (untuk fungsi simbolik murni, tidak dapat dihitung langsung), 4. Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah `function` (bukan sebagai ekspresi).

Berikut adalah beberapa contoh cara mendefinisikan fungsi:

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
```

```
>f(0), f(1), f(pi)
```

```
1
4.31977682472
20.7392088022
```

```
>f(a) // tidak dapat dihitung nilainya
```

```
Variable or function a not found.
```

```
Error in:
```

```
f(a) // tidak dapat dihitung nilainya ...
^
```

Silakan Anda plot kurva fungsi di atas!

Berikutnya kita definisikan fungsi:

```
>function g(x) := sqrt(x^2-3*x)/(x+1)
```

```
>g(3)
```

```
0
```

```
>g(0)
```

```
0
```

```
>g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik
```

```
Floating point error!
```

```
Error in sqrt
```

```
Try "trace errors" to inspect local variables after errors.
```

```
g:
```

```
useglobal; return sqrt(x^2-3*x)/(x+1)
```

```
Error in:
```

```
g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik ...
```

Silakan Anda plot kurva fungsi di atas!

```
>f(g(5)) // komposisi fungsi
```

```
2.20920171961
```

```
>g(f(5))
```

```
0.950898070639
```

```
>function h(x) := f(g(x)) // definisi komposisi fungsi
```

```
>h(5) // sama dengan f(g(5))
```

```
2.20920171961
```

Silakan Anda plot kurva fungsi komposisi fungsi f dan g:

dan

bersama-sama kurva fungsi f dan g dalam satu bidang koordinat.

```
>f(0:10) // nilai-nilai f(0), f(1), f(2), ..., f(10)
```

```
[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,  
99.929, 130.69, 163.51, 200.58]
```

```
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
```

```
[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,  
99.929, 130.69, 163.51, 200.58]
```

```
>gmap(200:210)
```

```
[0.987534, 0.987596, 0.987657, 0.987718, 0.987778, 0.987837,  
0.987896, 0.987954, 0.988012, 0.988069, 0.988126]
```

Misalkan kita akan mendefinisikan fungsi

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):
```

$$f(x) = 2x^2 + e^{\sin(x)}.$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-001.png

```
>function f(x) &= 2*E^x // fungsi simbolik
```

$$2 E^x$$

```
>f(a) // nilai f fungsi secara simbolik 2 e^a $>f(E) // nilai fungsi berupa bilangan desimal
```

```
30.308524483
```

```
>$f(E), float(2 e^E) 30.30852448295852
```

```
>function g(x) &= 3*x+1
```

$$3 x + 1$$

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

$$2 E^{3 x + 1}$$

```
>plot2d("h(x)",-1,1):
```

$1 + a + a^2$ ## Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan fungsi-fungsi tersebut dan komposisinya di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik fungsi-fungsi tersebut dan komposisi-komposisi 2 fungsi.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

Soal 1

Diketahui fungsi

tentukan nilai saat $a(-4)$, $a(4)$ kemudian gambarkan grafiknya

```
>function a(x):= x^2+3
```

```
>a(-4), a(4)
```

19

19

```
>plot2d("a",-4,4):
```

$1 + e + e^2$ Soal 2

Diberikan fungsi

tentukan nilai fungsi $z(-4)$, $z(6)$, dan gambarkan grafiknya

```
>function z(x) := (x^2+16)/(x-4)
```

```
>z(-4), z(6)
```

-4

26

```
>plot2d("z",-4,6):
```

11.1073379273897 3. Untuk fungsi

tentukan nilai $r(4)$, $r(-6)$, $r(8)$

```
>function r(x) := x^3-3*x^2+2*x-4
```

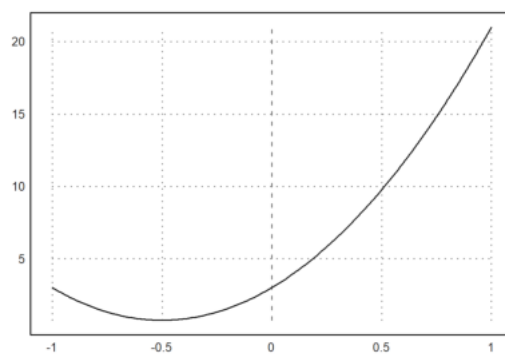
```
>r(4), r(-6), r(8)
```

20

-340

332

```
>plot2d("r",-2,9):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-008.png

4. Tentukan nilai $f(200)$ dari fungsi berikut

```
>function b(x) := sqrt(x-64)
```

```
>b(100), b(80)
```

6

4

```
>plot2d("sqrt(x-64)",0,200):
```

$a(x) = x^2 + 3$ Soal 5

Diberikan fungsi seperti berikut terdapat fungsi

```
>function a(x):= 2*x+3
```

terdapat fungsi

```
>function b(x):= x^2
```

terdapat fungsi

```
>function c(x):= 1/2^x
```

terdapat fungsi

```
>function d(x):= sqrt(4x)
```

terdapat fungsi

```
>function e(x):= sin(3*x)
```

a. komposisi fungsi $n(x) = c(b(x))$ terhadap nilai dan vektor juga gambarkan sketsanya

```
>function n(x):= c(b(x)) //mendefinisikan komposisi fungsi
```

```
>n(1) //fungsi komposisi terhadap suatu nilai
```

0.5

```
>n(1:5) //fungsi komposisi terhadap vektor
```

[0.5, 0.0625, 0.00195313, 1.52588e-05, 2.98023e-08]

b. komposisi fungsi $m(x) = d(a(x))$ terhadap nilai dan vektor juga gambarkan sketsanya

```
>function m(x):= d(a(x)) //mendefinisikan komposisi fungsi
```

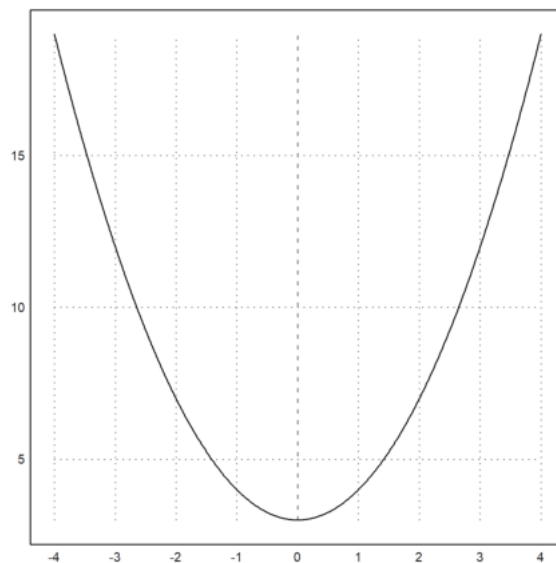
```
>m(1) // fungsi komposisi terhadap suatu nilai
```

4.472135955

```
>m(1:5) // fungsi komposisi terhadap vektor
```

[4.47214, 5.2915, 6, 6.63325, 7.2111]

```
>plot2d ("m(x)": //visualisasi dari fungsi komposisi
```



images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-010.png

c. komposisi fungsi $o(x) = e(b(x))$ terhadap nilai dan vektor juga gambarkan sketsanya

```
>function o(x):= e(b(x)) //mendefinisikan komposisi fungsi
```

```
>o(1) //fungsi komposisi terhadap suatu nilai
```

0.14112000806

```
>o(1:5) //fungsi komposisi terhadap vektor
```

[0.14112, -0.536573, 0.956376, -0.768255, -0.387782]

```
>plot2d ("o(x)": //visualisasi dari fungsi komposisi
```

$$z(x) = \frac{x^2 + 16}{x - 4}$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-011.png

Menghitung Limit

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni “limit”. Fungsi “limit” dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi “plus” atau “minus”. Hasil limit dapat berupa nilai, “und” (tak definisi), “ind” (tak tentu namun terbatas), “infinity” (kompleks tak hingga).

Perhatikan beberapa contoh berikut. Perhatikan cara menampilkan perhitungan secara lengkap, tidak hanya menampilkan hasilnya saja.

```
>showev('limit(sqrt(x^2 - 3)*x)/(x + 1), x, inf))lim_{x \to \infty} \frac{\sqrt{x^2-3}x}{x+1} = 1 limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

$$-\frac{4}{5}$$

```
'limit((x^3 - 13 * x^2 + 51 * x - 63)/(x^3 - 4 * x^2 - 3 * x + 18), x, 3)
```

```
= limit((x^3 - 13 * x^2 + 51 * x - 63)/(x^3 - 4 * x^2 - 3 * x + 18), x, 3)
```

Fungsi tersebut diskontinu di titik x=3. Berikut adalah grafik fungsinya.

```
>aspect(1.5); plot2d("(x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18)",0,4);
plot2d(3,-4/5,>points,style="ow",>add):
```

```
f(x) = \sqrt{x-64} >$limit(2*x*sin(x)/(1-cos(x)),x,0)
```

$$4$$

```
'limit(2 * x * sin(x)/(1 - cos(x)), x, 0) = limit(2 * x * sin(x)/(1 - cos(x)), x, 0)
```

Fungsi tersebut diskontinu di titik x=0. Berikut adalah grafik fungsinya.

```
>plot2d("(2*x*sin(x)/(1-cos(x))",-pi,pi); plot2d(0,4,>points,style="ow",>add):
```

```
a(x) = 2x + 3 >limit(cot(7*h)/cot(5*h),h,0)\frac{5}{7}
```

```
$$$showev('limit(cot(7h)/cot(5h),h,0))
```

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di x=0. Berikut adalah grafiknya.

```
>plot2d("cot(7*x)/cot(5*x)",-0.001,0.001); plot2d(0,5/7,>points,style="ow",>add):
```

```
c(x) = \frac{1}{2^x} >$showev('limit(((x/8)^(1/3)-1)/(x-8),x,8))
```

$$\lim_{x \rightarrow 8} \frac{\frac{x^{\frac{1}{3}}}{2} - 1}{x - 8} = \frac{1}{24}$$

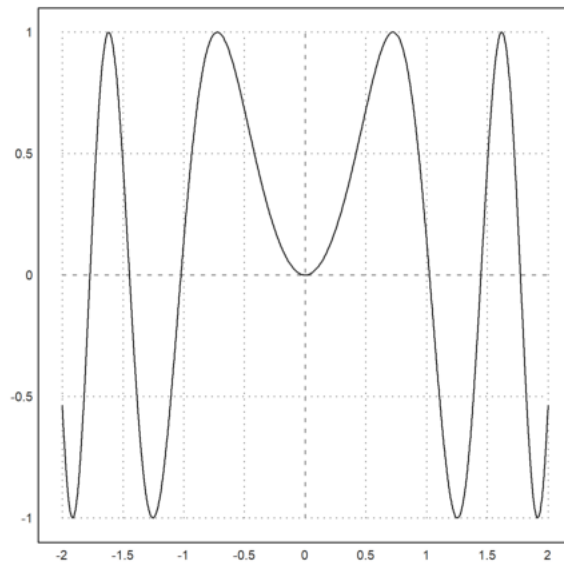
Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x/8)^(1/3)-1)/(x-8)",0,8); plot2d(8,1/24,>points,style="ow",>add):
```

```
e(x) = sin(3x) >showev('limit(1/(2*x-1), x, 0))lim_{x \to 0} \frac{1}{2x-1} = -1
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1/(2*x-1))",0,8); plot2d(0,-1,>points,style="ow",>add):
```



$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$ Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

`>plot2d("(x^2-3*x-10)/(x-5)",0,7); plot2d(5,7,>points,style="ow",>add):`

$$-\frac{4}{5}$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-024.png

$\lim_{x \rightarrow \infty} \sqrt{x^2 + x} - x = \frac{1}{2}$ Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

`>plot2d("(sqrt(x^2+x)-x)",0,1); plot2d(inf,1/2,>points,style="ow",>add):`

Variable or function inf not found.

Error in:

`plot2d("(sqrt(x^2+x)-x)",0,1); plot2d(inf,1/2,>points,style="o ...`

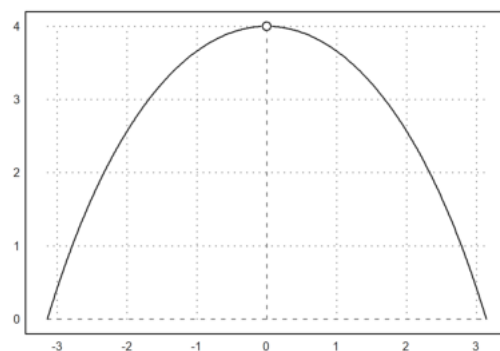
$\lim_{x \uparrow 1} \frac{|x-1|}{x-1} = -1$ Hitung limit di atas untuk x menuju 1 dari kanan.

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

$\lim_{x \uparrow 1} \frac{|x-1|}{x-1} = -1$
`showev('limit(sin(x)/x,x,0))`

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

`>plot2d("(sin(x)/x",-pi,pi); plot2d(0,1,>points,style="ow",>add):`



images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-029.png

$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$ Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

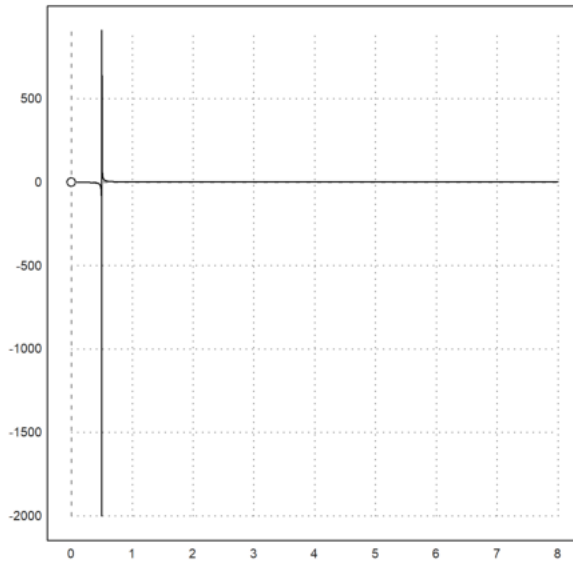
$\lim_{x \rightarrow -\infty} \log x = \text{infinity}$ `showev('limit((-2)^x,x,inf))`

$$\lim_{x \rightarrow \infty} (-2)^x = \text{infinity}$$

```
>showev('limit(t - sqrt(2 - t), t, 2, minus))lim_{t \uparrow 2} t - \sqrt{2 - t} = 2 showev('limit(t - sqrt(2-t), t, 2, plus))
```

$$\lim_{t \uparrow 2} t - \sqrt{2 - t} = 2$$

```
>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)",0,2):
```

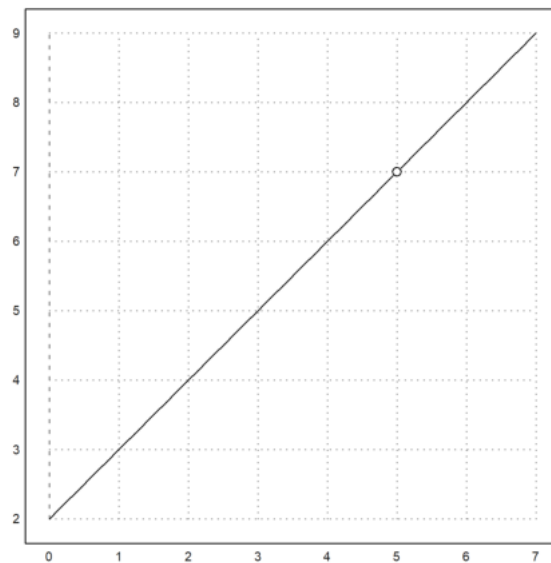


images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-036.png

```
>showev('limit((x^2 - 9)/(2 * x^2 - 5 * x - 3), x, 3))lim_{x \rightarrow 3} \frac{x^2 - 9}{2x^2 - 5x - 3} = \frac{6}{7}
```

\$Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(x^2-9)/(2*x^2-5*x-3)",0,4); plot2d(3,6/7,>points,style="ow",>add):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-038.png

```
>showev('limit(abs(x - 1)/(x - 1), x, 1, minus))lim_{x \uparrow 1} \frac{|x-1|}{x-1} = -1
showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1-cos(x))/x",0,2); plot2d(0,0,>points,style="ow",>add):
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-041.png

```
>$showev('limit((x^{2+abs(x)})/(x^2-abs(x)),x,0))
```

$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x^2+abs(x))/(x^2-abs(x)))",0,1); plot2d(0,-1,>points,style="ow",>add):
```

$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-043.png

```
>showev('limit((1 + 1/x)^x, x, inf))lim_{x \rightarrow \infty} (\frac{1}{x} + 1)^x = e
```

```
$>plot2d("(1+1/x)^x",0,1000):
```

```
lim_{x \rightarrow \infty} (-2)^x = infinity >showev('limit((1 + k/x)^x, x, inf))lim_{x \rightarrow \infty} (\frac{k}{x} + 1)^x = e^k
```

```
showev('limit((1+x)^(1/x),x,0))
```

$$\lim_{x \rightarrow 0} (x + 1)^{\frac{1}{x}} = e$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1+x)^(1/x)",0,2); plot2d(0,1000,>points,style="ow",>add):
```

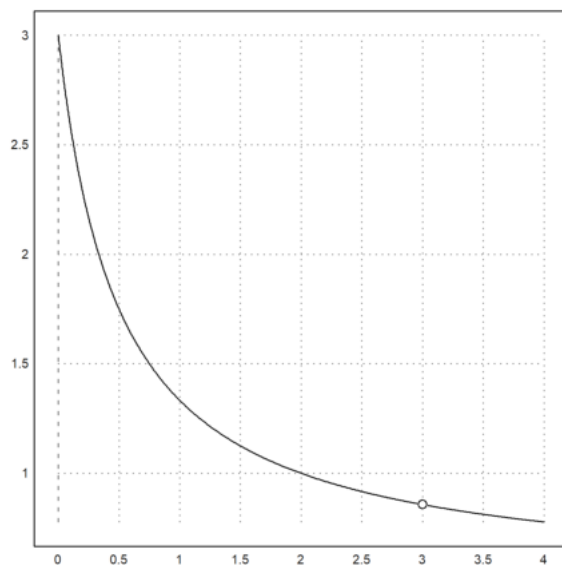
```
lim_{t \downarrow 5} -\sqrt{2-t} + t = 5 - \sqrt{3} i >$showev('limit((x/(x+k))^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left(\frac{x}{x+k} \right)^x = e^{-k}$$

```
>showev('limit((E^x - E^2)/(x - 2), x, 2))lim_{x \rightarrow 2} \frac{e^x - e^2}{x - 2} = e^2
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(E^x-E^2)/(x-2)",0,100); plot2d(2,1000^2,>points,style="ow",>add):
```



```
showev('limit(sin(1/x), x, 0))lim_{x \rightarrow 0} \sin(\frac{1}{x}) = ind showev('limit(sin(1/x),x,inf))
```

$$\lim_{x \rightarrow \infty} \sin\left(\frac{1}{x}\right) = 0$$

```
>plot2d("sin(1/x)",-0.001,0.001):
```

$$\lim_{x \rightarrow 0} \frac{x^2 + |x|}{x^2 - |x|} = -1$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-054.png

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.

Soal 1

Terdapat limit sebagai berikut, gambar grafik fungsi tersebut

>showev('limit((2*x-4),x,0))lim_{x→0} 2x-4 = -4 showev('limit((2*x-4),x,3))

$$\lim_{x \rightarrow 3} 2x - 4 = 2$$

>\$showev('limit((2*x-4),x,inf))

$$\lim_{x \rightarrow \infty} 2x - 4 = \infty$$

>plot2d("2*x-4",0,5); plot2d(0,-4,>points,style="ow",>add);
plot2d(3,2,>points,style="ow",>add):

$$\lim_{x \rightarrow \infty} \left(1 + \frac{1}{x}\right)^x = e$$

Soal 2

Terdapat limit sebagai berikut, gambar grafik fungsi tersebut

$$\lim_{x \rightarrow a} \frac{x^2 - 1}{x - 1}$$

untuk a bilangan real >showev('limit(sin(x)/x,x,0))// untuka = 0lim_{x→0} $\frac{\sin x}{x} = 1$
showev('limit(sin(x)/x,x,pi)) // untuk a=pi

$$\lim_{x \rightarrow \pi} \frac{\sin x}{x} = 0$$

>showev('limit(sin(x)/x,x,inf))// untuka = infinitylim_{x→∞} $\frac{\sin x}{x} = 0$

\$>plot2d("sin(x)/x",0,5); plot2d(0,1,>points,style="ow",>add);
plot2d(pi,0,>points,style="ow",>add):

$$\lim_{x \rightarrow 2} \frac{-e^2 + e^x}{-2 + x} = e^2$$

Soal 3 Terdapat limit fungsi sebagai berikut, gambar grafik fungsi limitnya!

$$\lim_{x \rightarrow a} \frac{\sin(x)}{\cos(x)}$$

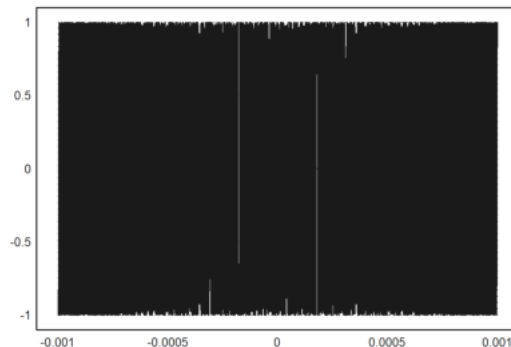
untuk a bilangan real

>showev('limit(((sin(x))/(cos(x))),x,0))// untuka = 0lim_{x→0} $\frac{\sin x}{\cos x} = 0$
showev('limit(((sin(x))/(cos(x))),x,pi)) //untuk a=pi

$$\lim_{x \rightarrow \pi} \frac{\sin x}{\cos x} = 0$$

>showev('limit(((sin(x))/(cos(x))),x,inf))// untuka = tak hingga

lim_{x→∞} $\frac{\sin x}{\cos x} = \text{und}$ >plot2d("sin(x)/(cos(x))",0,5);
plot2d(0,0,>points,style="ow",>add); plot2d(pi,0,>points,style="ow",>add):



images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-066.png

Soal 4

terdapat limit fungsi sebagai berikut

$$\lim_{x \rightarrow a} |x|$$

untuk a bilangan real

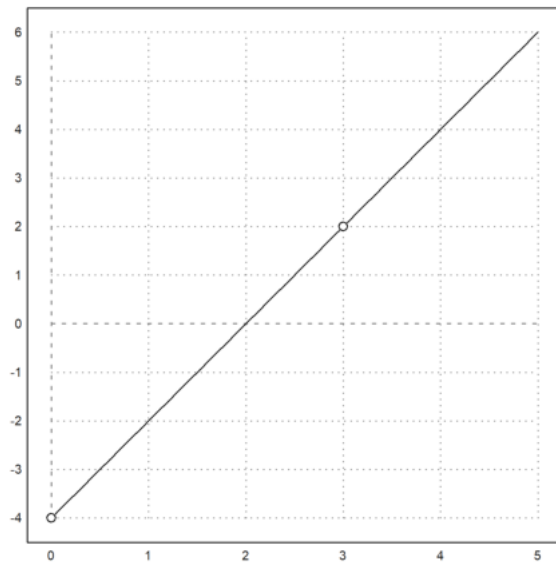
>showev('limit(((abs(x))),x,0))// untuka = 0lim_{x→0} |x| = 0
showev('limit(((abs(x))),x,2)) //untuk a=2

$$\lim_{x \rightarrow 2} |x| = 2$$

>\$showev('limit(((abs(x))),x,inf)) //untuk a= tak hingga

$$\lim_{x \rightarrow \infty} |x| = \infty$$

>plot2d("abs(x)",0,3); plot2d(0,0,>points,style="ow",>add);
plot2d(2,2,>points,style="ow",>add):



images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-070.png

Soal 5

Diberikan limit fungsi seperti berikut

$$\lim_{x \rightarrow a} \sqrt{x}$$

>showev('limit(((sqrt(x))), x, 0)) // untuka = 0
 lim_{x→0} √x = 0
 showev('limit(((sqrt(x))), x, 4)) // untuk a=4

$$\lim_{x \rightarrow 4} \sqrt{x} = 2$$

>\$showev('limit(((sqrt(x))), x, inf)) // untuk a=tak hingga

$$\lim_{x \rightarrow \infty} \sqrt{x} = \infty$$

>plot2d("((sqrt(x)))", 0, 5); plot2d(0, 0, >points, style="ow", >add);
 plot2d(4, 2, >points, style="ow", >add):

$$\lim_{x \rightarrow \infty} \frac{\sin x}{x} = 0$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-074.png

Turunan Fungsi

Definisi turunan:

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

>showev('limit(((x + h)² - x²)/h, h, 0)) // turunan x²
 lim_{h→0} $\frac{(x+h)^2 - x^2}{h} = 2x$

&= expand((x+h)²-x²)|simplify; p / pembilang dijabarkan dan disederhanakan
 2hx + h² / h &= ratsimp(p/h);

q / ekspresi yang akan dihitung limitnya disederhanakan 2x + h
 limit(q, h, 0) // nilai limit sebagai turunan

$$2x$$

>\$showev('limit(((x+h)ⁿ-xⁿ)/h, h, 0)) // turunan xⁿ

$$\lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} = nx^{n-1}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan (x+h)ⁿ dengan menggunakan teorema binomial.

Bukti

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Untuk

$$f(x) = x^n$$

$$\frac{d}{dx} \sin(x) = \lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h}$$

Dengan

$$(a + b)^n = \sum_{k=0}^n a^k b^{n-k}$$

maka

$$\begin{aligned} \lim_{h \rightarrow 0} \frac{(x^n + \frac{n}{1!} x^{n-1} h + \frac{n(n-1)}{2!} x^{n-2} h^2 + \frac{n(n-1)(n-2)}{3!} x^{n-3} h^3 + \dots) - x^n}{h} \\ \lim_{h \rightarrow 0} \frac{n \cdot x^{n-1} h + \frac{n(n-1)}{2!} x^{n-2} h^2 + \frac{n(n-1)(n-2)}{3!} x^{n-3} h^3 + \dots}{h} \\ \lim_{h \rightarrow 0} n \cdot x^{n-1} + \frac{n(n-1)}{2!} \cdot x^{n-2} h + \frac{n(n-1)(n-2)}{3!} \cdot x^{n-3} h^2 + \dots \\ = n \cdot x^{n-1} + 0 + 0 + \dots + 0 \\ = n \cdot x^{n-1} \end{aligned}$$

Jadi, terbukti benar bahwa

$$f'(x^n) = n \cdot x^{n-1}$$

>\$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)

$$\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h} = \cos x$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan sin(x+h) dengan menggunakan rumus jumlah dua sudut.
Bukti

$$\begin{aligned} f'(x) &= \lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin(x)}{h} \\ \sin(a+b) &= \sin(a)\cos(b) + \cos(a)\sin(b) \\ &= \lim_{h \rightarrow 0} \frac{\sin(x)\cos(h) + \cos(x)\sin(h) - \sin(x)}{h} \\ &= \lim_{h \rightarrow 0} \sin x \cdot \frac{\cos(h) - 1}{h} + \lim_{h \rightarrow 0} \cos(x) \cdot \frac{\sin(h)}{h} \\ &= \sin(x) \cdot 0 + \cos(x) \cdot 1 \\ &= \cos(x) \end{aligned}$$

Jadi, terbukti benar bahwa

$$: f'(\sin(x)) = \cos(x)$$

>\$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)

$$\lim_{h \rightarrow 0} \frac{\log(x+h) - \log x}{h} = \frac{1}{x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, gunakan sifat-sifat logaritma dan hasil limit pada bagian sebelumnya di atas.

Bukti

$$\begin{aligned} f'(x) &= \lim_{h \rightarrow 0} \frac{\log(x+h) - \log x}{h} \\ &= \lim_{h \rightarrow 0} \frac{\frac{d}{dh}(\log(x+h) - \log x)}{\frac{d}{dh}(h)} \\ &= \lim_{h \rightarrow 0} \frac{\frac{1}{x+h}}{1} \end{aligned}$$

$$= \lim_{h \rightarrow 0} \frac{1}{x + h}$$

$$= \frac{1}{x}$$

Jadi, terbukti benar bahwa

$$f'(x) = \lim_{h \rightarrow 0} \frac{\log(x + h) - \log x}{h} = \frac{1}{x}$$

```
>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:

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 showev('factor(E^{(x+h)}-E^x))
```

$$\text{factor}(e^{x+h} - e^x) = (e^h - 1) e^x$$

```
>showev('limit(factor((E^{(x+h)} - E^x)/h),h,0))// turunan f(x) = e^x
```

```
(lim_{h \to 0} \frac{e^h - 1}{h}) e^x = e^x $>function f(x) &= x^x
```

$$\frac{x^x}{x}$$

```
>showev('limit(f(x),x,0))lim_{x \to 0} x^x = 1$Silakan Anda gambar kurva
```

```
>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} = infinity$Di sini Maxima juga bermasalah terkait limit:
```

Dalam hal ini diperlukan asumsi nilai x.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x + h)^{x+h} - x^x}{h} = x^x (\log x + 1)$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

$$[x \geq 0]$$

```
>&forget(x<0)
```

$$[x \leq 0]$$

```
>&facts()
```

$$[]$$

```
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
```

$$\lim_{h \rightarrow 0} \frac{\arcsin(x + h) - \arcsin x}{h} = \frac{1}{\sqrt{1 - x^2}}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

$$\lim_{h \rightarrow 0} \frac{\tan(x + h) - \tan x}{h} = \frac{1}{\cos^2 x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar,

sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

$$\sinh(x)$$

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); df(x) // df(x) = f'(x)  $\frac{e^{-x}(e^{2x}+1)}{2}$ 
```

\$Hasilnya adalah cosh(x), karena

```
>plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]):
```

$$2x$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-092.png

```
>function f(x) &= sin(3*x5+7)2
```

$$\sin(3x^5 + 7)$$

```
>diff(f,3), diffc(f,3)
```

1198.32948904

1198.72863721

Apakah perbedaan diff dan diffc?

```
>showev('diff(f(x), x))  $\frac{d}{dx} \sin^2(3x^5 + 7) = 30x^4 \cos(3x^5 + 7) \sin(3x^5 + 7)$  %  
with x=3
```

$$\%at\left(\frac{d}{dx} \sin^2(3x^5 + 7), x = 3\right) = 2430 \cos 736 \sin 736$$

```
>$float(%)
```

$$\%at\left(\frac{d^{1.0}}{dx^{1.0}} \sin^2(3.0x^5 + 7.0), x = 3.0\right) = 1198.728637211748$$

```
>plot2d(f,0,3.1):
```

$$\frac{d}{dx} \sin(x) = \lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h}$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-096.png

```
>function f(x) &=5*cos(2*x)-2*x*sin(2*x) // mendefinisikan fungsi f
```

$$5 \cos(2x) - 2x \sin(2x)$$

```
>function df(x) &=diff(f(x),x) // fd(x) = f'(x)
```

$$-12 \sin(2x) - 4x \cos(2x)$$

```
>$'f(1)=f(1), $float(f(1)), $'f(2)=f(2), float(f(2)) // nilai f(1) dan f(2)
```

f(1) = 5 cos 2 - 2 sin 2\$

$$-3.899329036387075$$

$$f(2) = 5 \cos 4 - 4 \sin 4$$

$$-0.2410081230863468$$

```
>xp=solve("df(x)",1,2,0) // solusi f'(x)=0 pada interval [1, 2]
```

1.35822987384

```
>df(xp), f(xp) // cek bahwa f'(xp)=0 dan nilai ekstrim di titik tersebut
```

0

-5.67530133759

```
>plot2d(["f(x)", "df(x)"], 0, 2*pi, color=[blue, red]): //grafik fungsi dan turunannya
```

$$= n \cdot x^{n-1} + 0 + 0 + \dots + 0$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-101.png

Perhatikan titik-titik “puncak” grafik $y=f(x)$ dan nilai turunan pada saat grafik fungsinya mencapai titik “puncak” tersebut.

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), menggunakan perintah diff, dan secara manual (langkah demi langkah yang dihitung dengan Maxima) seperti contoh-contoh di atas. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

1. Tentukan nilai turunan berikut dan sketsakan grafiknya.

```
>function f(x) &= 4*x^2+8; f(x) 4 x^2 + 8$>function df(x) &= limit((f(x+h)-f(x))/h,h,0);
&df(x)/df(x)=f'(x)
```

8 x

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[red,green]):
```

$$f'(x^n) = n \cdot x^{n-1}$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-103.png

2. Carilah turunan dari fungsi berikut

```
>function f(x) &= (x-1)/(x-2); f(x) x-1 x-2$>function df(x) &= limit((f(x+h)-f(x))/h,h,0);
df(x) // df(x) = f'(x) - 1 x^2-4 x+4$>plot2d(["f(x)","df(x)"],-10,10,color=[blue,red]):
```

$$\sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b)$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-106.png

3. Carilah turunan dari fungsi berikut

```
>function f(x) &= 3/sqrt(x-2); f(x) 3 sqrt(x-2)$>function df(x) &= limit((f(x+h)-f(x))/h,h,0);
$df(x) // df(x) = f'(x)function f(x) &= 3/sqrt(x-2); f(x) - 3 2 (x-2) 1/2$>plot2d(["f(x)","df(x)"],-10,10,color=[yellow,red]):
```

$$= \sin(x).0 + \cos(x).1$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-109.png

4. Carilah turunan fungsi berikut.

```
>function f(x) &= (4*sin(x)+6*cos(x)); f(x) 4 sin x + 6 cos x$>function df(x) &=
limit((f(x+h)-f(x))/h,h,0); &df(x)
```

$$- 2 (3 \sin(x) - 2 \cos(x))$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,yellow]):
```

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

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-111.png

5. Tentukan turunan dan grafik fungsi berikut.

```
>function f(x) &= (sin(x)+cos(x))/(cos(x)); f(x) sin x+cos x cos x$>function df(x) &=
limit((f(x+h)-f(x))/h,h,0); df(x) // df(x) = f'(x) sin^2 x+cos^2 x cos^2 x$>plot2d(["f(x)","df(x)"],-
pi,pi,color=[blue,yellow]):
```

$$= \lim_{h \rightarrow 0} \frac{\frac{d}{dh}(\log(x+h) - \log x)}{\frac{d}{dh}(h)}$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-114.png

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:

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 1 x+1 dx = log(x+1)
```

```
showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2+1} dx = \arctan x$$

>showev('integrate(1/sqrt(1-x^2),x))\int \frac{1}{\sqrt{1-x^2}} dx = \arcsin x
 showev('integrate(sin(x),x,0,pi))

$$\int_0^{\pi} \sin x \, dx = 2$$

>plot2d("sin(x)",0,2*pi):

$$\lim_{h \rightarrow 0} \frac{e^{x+h} - e^x}{h}.$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-120.png

>showev('integrate(sin(x),x,a,b))\int_a^b \sin x \, dx = \cos a - \cos b
 showev('integrate(x^n,x,a,b))

Answering "Is n positive, negative or zero?" with "positive"

$$\int_a^b x^n \, dx = \frac{b^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}$$

>\$showev('integrate(x^2*sqrt(2*x+1),x))

$$\int x^2 \sqrt{2x+1} \, dx = \frac{(2x+1)^{\frac{7}{2}}}{28} - \frac{(2x+1)^{\frac{5}{2}}}{10} + \frac{(2x+1)^{\frac{3}{2}}}{12}$$

>\$showev('integrate(x^2*sqrt(2*x+1),x,0,2))

$$\int_0^2 x^2 \sqrt{2x+1} \, dx = \frac{25^{\frac{5}{2}}}{21} - \frac{2}{105}$$

>\$ratsimp(%)

$$\int_0^2 x^2 \sqrt{2x+1} \, dx = \frac{25^{\frac{7}{2}} - 2}{105}$$

>\$showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi2))

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x}+a) e^{\sqrt{x}}}{\sqrt{x}} \, dx = (-e^{\pi} - 1) \sin a + (e^{\pi} + 1) \cos a$$

>\$factor(%)

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x}+a) e^{\sqrt{x}}}{\sqrt{x}} \, dx = (-e^{\pi} - 1) (\sin a - \cos a)$$

>function map f(x) &= E^(-x2)

$$\frac{2}{-x} E$$

>\$showev('integrate(f(x),x))

$$\int e^{-x^2} \, dx = \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

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:

'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):

$$\lim_{h \rightarrow 0} \frac{-\tan x + \tan(h+x)}{h} = \frac{1}{\cos^2 x} \text{ ## Integral tentu}$$

'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!

$$\frac{1}{\int \sin(3x^5 + 7) dx} = \frac{\pi}{10} \frac{\Gamma(-\frac{1}{5}) \sin(14)}{\Gamma(\frac{1}{5}) \sin(\frac{\pi}{10})}$$

```

                    5
4/5      1      pi
- 6 I gamma_incomplete(-, - 6 I)) cos(14)) sin(-- - 60)/120
                    5                    10

>&float(%)

1.0
/
[      2      5
I      sin (3.0 x  + 7.0) dx =
]
/
0.0
0.09820784258795788 - 0.008333333333333333
(0.3090169943749474 (0.1367372182078336
(4.192962712629476 I gamma_incomplete(0.2, 6.0 I)
- 4.192962712629476 I gamma_incomplete(0.2, - 6.0 I))
+ 0.9906073556948704 (4.192962712629476 gamma_incomplete(0.2, 6.0 I)
+ 4.192962712629476 gamma_incomplete(0.2, - 6.0 I))) - 60.0)

>$showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)

$\int_0^1 x e^{-x} dx = 1 - 2 e^{-1}

```

Aplikasi Integral Tentu

```

>plot2d("x^3-x",-0.1,1.1); plot2d("-x^2",>add); ...
>b=solve("x^3-x+x^2",0.5); x=linspace(0,b,200); xi=flipx(x); ...
>plot2d(x|xi,x^3-x|xi^2,>filled,style="|",fillcolor=1,>add): // Plot daerah antara 2 kurva

```

$$\%at\left(\frac{d}{dx}\sin^2(7+3x^5),x=3\right)=2430\cos736\sin736$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-134.png

```

>a=solve("x^3-x+x^2",0),b=solve("x^3-x+x^2",1) // absis titik-titik potong kedua kurva

```

```

0
0.61803398875

```

```

>integrate("(-x^2)-(x3-x)",a,b) // luas daerah yang diarsir

```

```

0.0758191713542

```

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```

>a &= solve((-x^2)-(x3-x),x); $a // menentukan absis titik potong kedua kurva secara eksak

```

$$\left[x = \frac{-\sqrt{5}-1}{2}, x = \frac{\sqrt{5}-1}{2}, x = 0 \right]$$

```

>$showev('integrate(-x^2-x3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara eksak

```

$$\int_0^{\frac{\sqrt{5}-1}{2}} -x^3 - x^2 + x dx = \frac{13-5^{\frac{3}{2}}}{24}$$

```

>$float(%)

```

```

$\int_{0,0}^{0,6180339887498949} -1.0x^3 - 1.0x^2 + x dx = 0.0758191713542103

```

Panjang Kurva

Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

dengan

```

>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",r=1.5): // Kita gambar kurvanya terlebih dahulu

```

```

-3.899329036387075>function r(t) &= 1+sin(3*t)/2; $r(t)=r(t)

```

```

[ 0 , 0.01 , 0.02 , 0.03 , 0.04 , 0.05 , 0.06 , 0.07 , 0.08 , 0.09 , 0.1 , 0.11 , 0.12 , 0.13 , 0.14
, 0.15 , 0.16 , 0.17 , 0.18 , 0.19 , 0.2 , 0.21 , 0.2200000000000001 , 0.2300000000000001 ,
0.2400000000000001 , 0.2500000000000001 , 0.2600000000000001 ,
0.2700000000000001 , 0.2800000000000001 , 0.2900000000000001 ,
0.3000000000000001 , 0.3100000000000001 , 0.3200000000000001 ,
0.3300000000000001 , 0.3400000000000001 , 0.3500000000000001 ,
0.3600000000000002 , 0.3700000000000002 , 0.3800000000000002 ,
0.3900000000000002 , 0.4000000000000002 , 0.4100000000000002 ,
0.4200000000000002 , 0.4300000000000002 , 0.4400000000000002 ,
0.4500000000000002 , 0.4600000000000002 , 0.4700000000000003 ,

```

```
0.48000000000000003 , 0.49000000000000003 , 0.50000000000000002 ,
0.51000000000000002 , 0.52000000000000002 , 0.53000000000000002 ,
0.54000000000000003 , 0.55000000000000003 , 0.56000000000000003 ,
0.57000000000000003 , 0.58000000000000003 , 0.59000000000000003 ,
0.60000000000000003 , 0.61000000000000003 , 0.62000000000000003 ,
0.63000000000000003 , 0.64000000000000003 , 0.65000000000000004 ,
0.66000000000000004 , 0.67000000000000004 , 0.68000000000000004 ,
0.69000000000000004 , 0.70000000000000004 , 0.71000000000000004 ,
0.72000000000000004 , 0.73000000000000004 , 0.74000000000000004 ,
0.75000000000000004 , 0.76000000000000005 , 0.77000000000000005 ,
0.78000000000000005 , 0.79000000000000005 , 0.80000000000000005 ,
0.81000000000000005 , 0.82000000000000005 , 0.83000000000000005 ,
0.84000000000000005 , 0.85000000000000005 , 0.86000000000000005 ,
0.87000000000000006 , 0.88000000000000006 , 0.89000000000000006 ,
0.90000000000000006 , 0.91000000000000006 , 0.92000000000000006 ,
0.93000000000000006 , 0.94000000000000006 , 0.95000000000000006 ,
0.96000000000000006 , 0.97000000000000006 , 0.98000000000000006 ,
0.99000000000000007 ]$$
```

```
)=$$
```

```
>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
```

```
fx ([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.22000
```

```
>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
```

```
$$ {} ()=
```

```
$$>function ds(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t)
```

```
Maxima said:
```

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

```
Error in:
```

```
... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t ...
^
```

```
>integrate(ds(x),x,0,2\*pi)// panjang(keliling)kurva  $\int_0^{2\pi} ds(x) dx$ Maxima
gagal melakukan perhitungan eksak integral tersebut.
```

Berikut kita hitung integralnya secara umerik dengan perintah EMT.

```
>integrate("ds(x)",0,2*pi)
```

```
Function ds not found.
```

```
Try list ... to find functions!
```

```
Error in expression: ds(x)
```

```
%mapexpression1:
```

```
return expr(x,args());
```

```
Error in map.
```

```
%evalexpression:
```

```
if maps then return %mapexpression1(x,f$,args());
```

```
gauss:
```

```
if maps then y=%evalexpression(f$,a+h-(h*xn)',maps,args());
```

```
adaptivegauss:
```

```
t1=gauss(f$,c,c+h,args(),=maps);
```

```
Try "trace errors" to inspect local variables after errors.
```

```
integrate:
```

```
return adaptivegauss(f$,a,b,eps*1000,args(),=maps);
```

Spiral Logaritmik

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi):
```

$4x^2 + 8$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-143.png

```
>&kill(a) // hapus expresi a
```

done

```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
```

```
fx ([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.22000
```

```
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
```

```
fy ([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.22000
```

```
>function df(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t)
```

```
Maxima said:
```

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

```
Error in:
```

```
... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t) ...
```

```
>S &=integrate(df(t),t,0,2*pi); $S // panjang kurva (spiral)
```

Maxima said:

```
defint: variable of integration cannot be a constant; found errexpl
-- an error. To debug this try: debugmode(true);
```

Error in:

```
S &=integrate(df(t),t,0,2*pi); $S // panjang kurva (spiral) ...
```

```
>S(a=0.1) // Panjang kurva untuk a=0.1
```

Function S not found.

Try list ... to find functions!

Error in:

```
S(a=0.1) // Panjang kurva untuk a=0.1 ...
```

Soal:

Tunjukkan bahwa keliling lingkaran dengan jari-jari r adalah $K=2\pi r$.

Berikut adalah contoh menghitung panjang parabola.

```
>plot2d("x^2",xmin=-1,xmax=1):
```

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

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-146.png

```
>$showev('integrate(sqrt(1+diff(x^2,x),-1,1))
```

$$\int_{-1}^1 \sqrt{4x^2 + 1} \, dx = \frac{\operatorname{asinh} 2 + 2\sqrt{5}}{2}$$

```
>$float(%)
```

$$\int_{-1.0}^{1.0} \sqrt{4.0x^2 + 1.0} \, dx = 2.957885715089195$$

```
>x=-1:0.2:1; y=x^2; plot2d(x,y); ...
```

```
> plot2d(x,y,points=1,style="o#",add=1):
```

$$f(x) = \frac{3}{\sqrt{x-2}}$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-149.png

Panjang tersebut dapat dihamperi dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```
>i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])^2+(y[i+1]-y[i])^2))
```

```
2.95191957027
```

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

Koordinat Kartesius

Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

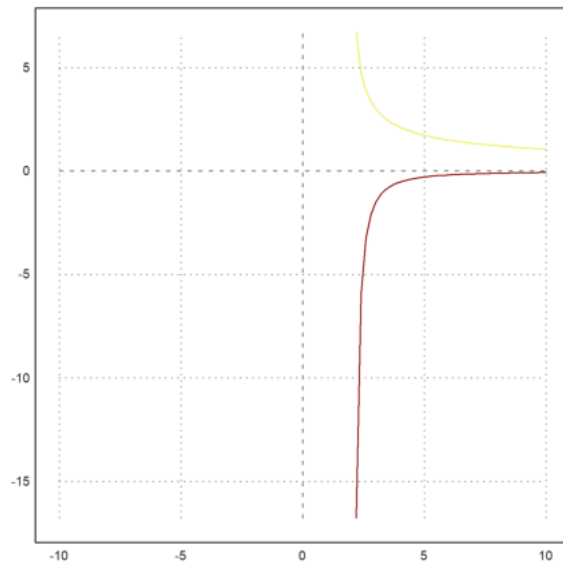
```
>z &= x^3+y^3-3*x*y; zy^3 - 3 x y + x^3$>plot2d(z,r=2,level=0,n=100):
```

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

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-151.png

Kita tertarik pada kurva di kuadran pertama.

```
>plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="r"):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-152.png

Kita selesaikan persamaannya untuk x.

```
>$z with y=l*x, sol &= solve(%x); sol3 x3 + x3 - 3 l x2$
```

$$\left[x = \frac{3l}{l^3 + 1}, x = 0 \right]$$

Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); 'f(l) = f(l)f(l) = \frac{3l}{l^3+1}$
```

Fungsi tersebut juga dapat digunakan untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai y=lx, yakni $x=f(l)$ dan $y=lf(l)$.

```
>plot2d(&f(x),&x*f(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):
```

$$f(x) = \frac{\sin(x) + \cos(x)}{\cos(x)}$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-156.png

Elemen panjang kurva adalah:

```
>function ds(l) &= ratsimp(sqrt(diff(f(l),l)2+diff(l*f(l),l)2)); $'ds(l)=ds(l)
```

$$ds(l) = \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}}$$

```
>fs$integrate(ds(l),l,0,1)
```

Variable or function l not found.

Error in:

```
fs$integrate(ds(l),l,0,1) ...
```

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral tersebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
>2*integrate("ds(x)",0,1)
```

4.91748872168

```
>2*romberg(&ds(x),0,1)// perintah Euler lain untuk menghitung nilai hampiran integral yang sama
```

4.91748872168

Perhitungan di atas dapat dilakukan untuk sebarang fungsi x dan y dengan mendefinisikan fungsi EMT, misalnya kita beri nama panjangkurva. Fungsi ini selalu memanggil Maxima untuk menurunkan fungsi yang diberikan.

```
>function panjangkurva(fx,fy,a,b) ...
```

```
ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");
return romberg(ds,a,b);
endfunction
```

```
>panjangkurva("x","x^2",-1,1) // cek untuk menghitung panjang kurva parabola sebelumnya
```

2.95788571509

Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)",mxm("x*f(x)",0,1)// cek contoh terakhir, bandingkan hasilnya!
```

4.91748872168

Kita hitung panjang spiral Archimides berikut ini dengan fungsi tersebut.

```
>plot2d("x*cos(x)","x*sin(x)",xmin=0,xmax=2*pi,square=1):
```

$$\frac{\sin^2 x + \cos^2 x}{\cos^2 x}$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-158.png

```
>panjangkurva("x*cos(x)","x*sin(x)",0,2*pi)
```

21.2562941482

Berikut kita definisikan fungsi yang sama namun dengan Maxima, untuk perhitungan eksak.

```
>&kill(ds,x,fx,fy)
```

done

```
>function ds(fx,fy) &&= sqrt(diff(fx,x)2+diff(fy,x)2)
```

$$\sqrt{\text{diff}(fy, x)^2 + \text{diff}(fx, x)^2}$$

```
>sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk menghitung panjang kurva terakhir di atas
```

$$\sqrt{(\cos x - x \sin x)^2 + (\sin x + x \cos x)^2}$$

```
>$sol | trigreduce | expand, integrate(√ x2 + 1$
```

$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$

21.2562941482

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.

Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
>plot2d("3*x2-1","3*x3-1",xmin=-1/sqrt(3),xmax=1/sqrt(3),square=1):
```

$$\int \frac{1}{1+x^2} dx = \arctan x$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-162.png

```
>sol &= radcan(ds(3*x2-1,3*x3-1)); sol3 √ 9 x2 + 4
```

```
showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%) // panjang kurva di atas
```

$$3 \int_0^{\frac{1}{\sqrt{3}}} x \sqrt{9x^2 + 4} dx = 3 \left(\frac{7^{\frac{3}{2}}}{27} - \frac{8}{27} \right)$$

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0x^2 + 4.0} dx = 2.337835372767141$$

Sikloid

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar kekanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah r. Posisi titik pusat lingkaran pada saat t adalah:

Misalkan posisi titik pada lingkaran tersebut mula-mula (0,0) dan posisinya pada saat t adalah:

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika t=0, t=pi/2, t=pi.

```
> x &= r(t-sin(t))
```

```
[0, 1.66665833335744e-7 r, 1.33330666692022e-6 r, 4.499797504338432e-6 r,
1.066581336583994e-5 r, 2.083072932167196e-5 r, 3.599352055540239e-5 r,
5.71526624672386e-5 r, 8.530603082730626e-5 r, 1.214508019889565e-4 r,
1.665833531718508e-4 r, 2.216991628251896e-4 r, 2.877927110806339e-4 r,
3.658573803051457e-4 r, 4.568853557635201e-4 r, 5.618675264007778e-4 r,
6.817933857540259e-4 r, 8.176509330039827e-4 r, 9.704265741758145e-4 r,
0.001141105023499428 r, 0.001330669204938795 r, 0.001540100153900437 r,
0.001770376919130678 r, 0.002022476464811601 r, 0.002297373572865413 r,
0.002596040745477063 r, 0.002919448107844891 r, 0.003268563311168871 r,
0.003644351435886262 r, 0.004047774895164447 r, 0.004479793338660443 r,
0.0049413635565565 r, 0.005433439383882244 r, 0.005956971605131645 r,
```

0.006512907859185624 r, 0.007102192544548636 r, 0.007725766724910044 r,
0.00838456803503801 r, 0.009079530587017326 r, 0.009811584876838586 r,
0.0105816576913495 r, 0.01139067201557714 r, 0.01223954694042984 r,
0.01312919757078923 r, 0.01406053493400045 r, 0.01503446588876983 r,
0.01605189303448024 r, 0.01711371462093175 r, 0.01822082445851714 r,
0.01937411182884202 r, 0.02057446139579705 r, 0.02182275311709253 r,
0.02311986215626333 r, 0.02446665879515308 r, 0.02586400834688696 r,
0.02731277106934082 r, 0.02881380207911666 r, 0.03036795126603076 r,
0.03197606320812652 r, 0.0336389770872163 r, 0.03535752660496472 r,
0.03713253989951881 r, 0.03896483946269502 r, 0.0408552420577305 r,
0.04280455863760801 r, 0.04481359426396048 r, 0.04688314802656623 r,
0.04901401296344043 r, 0.05120697598153157 r, 0.05346281777803219 r,
0.05578231276230905 r, 0.05816622897846346 r, 0.06061532802852698 r,
0.0631303649963022 r, 0.06571208837185505 r, 0.06836123997666599 r,
0.07107855488944881 r, 0.07386476137264342 r, 0.07672058079958999 r,
0.07964672758239233 r, 0.08264390910047736 r, 0.0857128256298576 r,
0.08885417027310427 r, 0.09206862889003742 r, 0.09535688002914089 r,
0.0987195948597075 r, 0.1021574371047232 r, 0.1056710629744951 r,
0.1092611211010309 r, 0.1129282524731764 r, 0.1166730903725168 r,
0.1204962603100498 r, 0.1243983799636342 r, 0.1283800591162231 r,
0.1324418995948859 r, 0.1365844952106265 r, 0.140808431699002 r,
0.1451142866615502 r, 0.1495026295080298 r, 0.1539740213994798 r]`

| y &= r*(1-cos(t))

[0, 4.999958333473664e-5 r, 1.999933334222437e-4 r, 4.499662510124569e-4 r,
7.998933390220841e-4 r, 0.001249739605033717 r, 0.00179946006479581 r,
0.002448999746720415 r, 0.003198293697380561 r, 0.004047266988005727 r,
0.004995834721974179 r, 0.006043902043303184 r, 0.00719136414613375 r,
0.00843810628521191 r, 0.009784003787362772 r, 0.01122892206395776 r,
0.01277271662437307 r, 0.01441523309043924 r, 0.01615630721187855 r,
0.01799576488272969 r, 0.01993342215875837 r, 0.02196908527585173 r,
0.02410255066939448 r, 0.02633360499462523 r, 0.02866202514797045 r,
0.03108757828935527 r, 0.03361002186548678 r, 0.03622910363410947 r,
0.03894456168922911 r, 0.04175612448730281 r, 0.04466351087439402 r,
0.04766643011428662 r, 0.05076458191755917 r, 0.0539576564716131 r,
0.05724533447165381 r, 0.06062728715262111 r, 0.06410317632206519 r,
0.06767265439396564 r, 0.07133536442348987 r, 0.07509094014268702 r,
0.07893900599711501 r, 0.08287917718339499 r, 0.08691105968769186 r,
0.09103425032511492 r, 0.09524833678003664 r, 0.09955289764732322 r,
0.1039475024744748 r, 0.1084317118046711 r, 0.113005077220716 r,
0.1176671413898787 r, 0.1224174381096274 r, 0.1272554923542488 r,
0.1321808203223502 r, 0.1371929294852391 r, 0.1422913186361759 r,
0.1474754779404944 r, 0.152744888986584 r, 0.1580990248377314 r,
0.1635373500848132 r, 0.1690593208998367 r, 0.1746643850903219 r,
0.1803519821545206 r, 0.1861215433374662 r, 0.1919724916878484 r,
0.1979042421157076 r, 0.2039162014509444 r, 0.2100077685026351 r,
0.216178334119151 r, 0.2224272812490723 r, 0.2287539850028937 r,
0.2351578127155118 r, 0.2416381240094921 r, 0.2481942708591053 r,
0.2548255976551299 r, 0.2615314412704124 r, 0.2683111311261794 r,
0.2751639892590951 r, 0.2820893303890569 r, 0.2890864619877229 r,
0.2961546843477643 r, 0.3032932906528349 r, 0.3105015670482534 r,
0.3177787927123868 r, 0.3251242399287333 r, 0.3325371741586922 r,
0.3400168541150183 r, 0.3475625318359485 r, 0.3551734527599992 r,
0.3628488558014202 r, 0.3705879734263036 r, 0.3783900317293359 r,
0.3862542505111889 r, 0.3941798433565377 r, 0.4021660177127022 r,
0.4102119749689023 r, 0.418316910536117 r, 0.4264800139275439 r,
0.4347004688396462 r, 0.4429774532337832 r, > % Berikut kita gambar sikloid untuk
r=1. > >ex &= x-sin(x); ey &= 1-cos(x); aspect(1);`
>plot2d(ex,ey,xmin=0,xmax=4pi,square=1); ... >
plot2d("2+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ... > plot2d([2,ex(2)],
[1,ey(2)],color=red,>add); ... > plot2d(ex(2),ey(2),>points,>add,color=red); ... >
plot2d("2pi+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ... >
plot2d([2pi,ex(2pi)],[1,ey(2pi)],color=red,>add); ... >
plot2d(ex(2pi),ey(2pi),>points,>add,color=red):

[0,1.6666583335744e-7r-sin(1.6666583335744e-7r),1.33330666692022e-6r-
sin(1.33330666692022e-6r),4.499797504338432e-6r-sin(4.499797504338432e-
6r),1.066581336583994e-5r-sin(1.066581336583994e-5r),2.083072932167196e-5r-
sin(2.083072932167196e-5r),3.599352055540239e-5r-sin(3.599352055540239e-
5r),5.71526624672386e-5r-sin(5.71526624672386e-5r),8.530603082730626e-5r-
sin(8.530603082730626e-5r),1.214508019889565e-4r-sin(1.214508019889565e-
4r),1.665833531718508e-4r-sin(1.665833531718508e-4r),2.216991628251896e-4r-
sin(2.216991628251896e-4r),2.877927110806339e-4r-sin(2.877927110806339e-
4r),3.658573803051457e-4r-sin(3.658573803051457e-4r),4.5688535576352e-4r-
sin(4.5688535576352e-4r),5.618675264007778e-4r-sin(5.618675264007778e-
4r),6.817933857540259e-4r-sin(6.817933857540259e-4r),8.176509330039827e-4r-
sin(8.176509330039827e-4r),9.704265741758145e-4r-sin(9.704265741758145e-
4r),0.001141105023499428r-sin(0.001141105023499428r),0.001330669204938795r-
sin(0.001330669204938795r),0.001540100153900437r-
sin(0.001540100153900437r),0.001770376919130678r-
sin(0.001770376919130678r),0.002022476464811601r-
sin(0.002022476464811601r),0.002297373572865413r-
sin(0.002297373572865413r),0.002596040745477063r-
sin(0.002596040745477063r),0.002919448107844891r-
sin(0.002919448107844891r),0.003268563311168871r-
sin(0.003268563311168871r),0.0036443351435886262r-

```

sin(0.003644351435886262r),0.004047774895164447r-
sin(0.004047774895164447r),0.004479793338660443r-
sin(0.004479793338660443r),0.0049413635565565r-
sin(0.0049413635565565r),0.005433439383882244r-
sin(0.005433439383882244r),0.005956971605131645r-
sin(0.005956971605131645r),0.006512907859185624r-
sin(0.006512907859185624r),0.007102192544548636r-
sin(0.007102192544548636r),0.007725766724910044r-
sin(0.007725766724910044r),0.00838456803503801r-
sin(0.00838456803503801r),0.009079530587017326r-
sin(0.009079530587017326r),0.009811584876838586r-
sin(0.009811584876838586r),0.0105816576913495r-
sin(0.0105816576913495r),0.01139067201557714r-
sin(0.01139067201557714r),0.01223954694042984r-
sin(0.01223954694042984r),0.01312919757078923r-
sin(0.01312919757078923r),0.01406053493400045r-
sin(0.01406053493400045r),0.01503446588876983r-
sin(0.01503446588876983r),0.01605189303448024r-
sin(0.01605189303448024r),0.01711371462093175r-
sin(0.01711371462093175r),0.01822082445851714r-
sin(0.01822082445851714r),0.01937411182884202r-
sin(0.01937411182884202r),0.02057446139579705r-
sin(0.02057446139579705r),0.02182275311709253r-
sin(0.02182275311709253r),0.02311986215626333r-
sin(0.02311986215626333r),0.02446665879515308r-
sin(0.02446665879515308r),0.02586400834688696r-
sin(0.02586400834688696r),0.02731277106934082r-
sin(0.02731277106934082r),0.02881380207911666r-
sin(0.02881380207911666r),0.03036795126603076r-
sin(0.03036795126603076r),0.03197606320812652r-
sin(0.03197606320812652r),0.0336389770872163r-
sin(0.0336389770872163r),0.03535752660496472r-
sin(0.03535752660496472r),0.03713253989951881r-
sin(0.03713253989951881r),0.03896483946269502r-
sin(0.03896483946269502r),0.0408552420577305r-
sin(0.0408552420577305r),0.04280455863760801r-
sin(0.04280455863760801r),0.04481359426396048r-
sin(0.04481359426396048r),0.04688314802656623r-
sin(0.04688314802656623r),0.04901401296344043r-
sin(0.04901401296344043r),0.05120697598153157r-
sin(0.05120697598153157r),0.05346281777803219r-
sin(0.05346281777803219r),0.05578231276230905r-
sin(0.05578231276230905r),0.05816622897846346r-
sin(0.05816622897846346r),0.06061532802852698r-
sin(0.06061532802852698r),0.0631303649963022r-
sin(0.0631303649963022r),0.06571208837185505r-
sin(0.06571208837185505r),0.06836123997666599r-
sin(0.06836123997666599r),0.07107855488944881r-
sin(0.07107855488944881r),0.07386476137264342r-
sin(0.07386476137264342r),0.07672058079958999r-
sin(0.07672058079958999r),0.07964672758239233r-
sin(0.07964672758239233r),0.08264390910047736r-
sin(0.08264390910047736r),0.0857128256298576r-
sin(0.0857128256298576r),0.08885417027310427r-
sin(0.08885417027310427r),0.09206862889003742r-
sin(0.09206862889003742r),0.09535688002914089r-
sin(0.09535688002914089r),0.0987195948597075r-
sin(0.0987195948597075r),0.1021574371047232r-
sin(0.1021574371047232r),0.1056710629744951r-
sin(0.1056710629744951r),0.1092611211010309r-
sin(0.1092611211010309r),0.1129282524731764r-
sin(0.1129282524731764r),0.1166730903725168r-
sin(0.1166730903725168r),0.1204962603100498r-
sin(0.1204962603100498r),0.1243983799636342r-
sin(0.1243983799636342r),0.1283800591162231r-
sin(0.1283800591162231r),0.1324418995948859r-
sin(0.1324418995948859r),0.1365844952106265r-
sin(0.1365844952106265r),0.140808431699002r-
sin(0.140808431699002r),0.1451142866615502r-
sin(0.1451142866615502r),0.1495026295080298r-
sin(0.1495026295080298r),0.1539740213994798r-sin(0.1539740213994798r)]

```

```

>ds &= radcan(sqrt(diff(ex,x)2+diff(ey,x)2)); $ds=trigsimp(ds) // elemen panjang kurva
sikloid

```

^

```

>ds &= trigsimp(ds); ds >showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu
putaran penuh

```

```

>showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid sat ...^
>integrate(mxm("ds"),0,2*pi) // hitung secara numerik

```

```

>romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik

```

Kurvatur (Kelengkungan) Kurva

Aslinya, kelengkungan kurva diferensiabel (yakni, kurva mulus yang tidak lancip) di titik P didefinisikan melalui lingkaran oskulasi (yaitu, lingkaran yang melalui titik P dan terbaik memperkirakan, paling banyak menyinggung kurva di sekitar P). Pusat dan radius kelengkungan kurva di P adalah pusat dan radius lingkaran oskulasi. Kelengkungan adalah kebalikan dari radius kelengkungan:

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar 2π sejauh $2\pi R$.)

Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekuivalen.

Kurvatur (Kelengkungan) Kurva

Aslinya, kelengkungan kurva diferensiabel (yakni, kurva mulus yang tidak lancip) di titik P didefinisikan melalui lingkaran oskulasi (yaitu, lingkaran yang melalui titik P dan terbaik memperkirakan, paling banyak menyinggung kurva di sekitar P). Pusat dan radius kelengkungan kurva di P adalah pusat dan radius lingkaran oskulasi. Kelengkungan adalah kebalikan dari radius kelengkungan:

$$\kappa = \frac{1}{R}$$

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar 2π sejauh $2\pi R$.) Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekuivalen.

Definisi Kurvatur dengan Fungsi Parametrik Panjang Kurva

Setiap kurva diferensiabel dapat dinyatakan dengan persamaan parametrik terhadap panjang kurva s:

$$\gamma(s) = (x(s), y(s)),$$

dengan x dan y adalah fungsi riil yang diferensiabel, yang memenuhi:

$$\|\gamma'(s)\| = \sqrt{x'(s)^2 + y'(s)^2} = 1.$$

Ini berarti bahwa vektor singgung

$$T(s) = (x'(s), y'(s))$$

memiliki norm 1 dan merupakan vektor singgung satuan.

Apabila kurvanya memiliki turunan kedua, artinya turunan kedua x dan y ada, maka $T'(s)$ ada. Vektor ini merupakan normal kurva yang arahnya menuju pusat kurvatur, norm-nya merupakan nilai kurvatur (kelengkungan):

$$T(s) = \gamma'(s),$$

$$T^2(s) = 1$$

$$\text{text(konstanta)} \Rightarrow T'(s) \cdot T(s) = 0$$

$$\kappa(s) = \|T'(s)\| = \|\gamma''(s)\| = \sqrt{x''(s)^2 + y''(s)^2}.$$

Nilai

$$R(s) = \frac{1}{\kappa(s)}$$

disebut jari-jari (radius) kelengkungan kurva.

Bilangan riil

$$k(s) = \pm \kappa(s)$$

disebut nilai kelengkungan bertanda.

Contoh: Akan ditentukan kurvatur lingkaran

$$x = r \cos t, y = r \sin t.$$

Akan ditentukan kurvatur lingkaran

$$x = r \cos(t); y = r \sin(t);$$

> assume(t>0, r>0); s := integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2), t, 0, t); s // elemen panjang kurva, panjang busur lingkaran (s)

Maxima said:

diff: second argument must be a variable; found errexpl

```
-- an error. To debug this try: debugmode(true);

Error in:
... =integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s // elemen ...

>&kill(s); fx &= r*cos(s/r); fy &= r*sin(s/r); // definisi ulang persamaan parametrik
terhadap s dengan substitusi t=s/r

>k &= trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur lingkaran dengan
menggunakan definisi di atas
```

$$\frac{1}{r}$$

Untuk representasi parametrik umum, misalkan

$$x = x(t), y = y(t)$$

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

$$\begin{aligned}\kappa &= \frac{d\phi}{ds} = \frac{\frac{d\phi}{dt}}{\frac{ds}{dt}} \quad (\phi \text{ adalah sudut kemiringan garis singgung dan } s \text{ adalah panjang kurva}) \\ &= \frac{\frac{d\phi}{dt}}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}} = \frac{\frac{d\phi}{dt}}{\sqrt{x'(t)^2 + y'(t)^2}}.\end{aligned}$$

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

$$\begin{aligned}\sec^2 \phi \frac{d\phi}{dt} &= \frac{d}{dt}(\tan \phi) = \frac{d}{dt}\left(\frac{dy}{dx}\right) = \frac{d}{dt}\left(\frac{dy/dt}{dx/dt}\right) = \frac{d}{dt}\left(\frac{y'(t)}{x'(t)}\right) = \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ \frac{d\phi}{dt} &= \frac{1}{\sec^2 \phi} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1 + \tan^2 \phi} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1 + \left(\frac{y'(t)}{x'(t)}\right)^2} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2 + y'(t)^2}.\end{aligned}$$

Jadi, rumus kurvatur untuk kurva parametrik

$$x = x(t), y = y(t)$$

adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}.$$

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kutub

$$x = r(\theta) \cos \theta, y = r(\theta) \sin \theta,$$

maka rumus kurvturnya adalah

$$\kappa(\theta) = \frac{r(\theta)^2 + 2r'(\theta)^2 - r(\theta)r''(\theta)}{(r'(\theta)^2 + r(\theta)^2)^{3/2}}.$$

Contoh: Lingkaran dengan pusat (0,0) dan jari-jari r dapat dinyatakan dengan persamaan parametrik

$$x = r \cos t, y = r \sin t.$$

Nilai kelengkungan lingkaran tersebut adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}} = \frac{r^2}{r^3} = \frac{1}{r}.$$

Hasil cocok dengan definisi kurvatur suatu kelengkungan.

Kurva

$$y = f(x)$$

dapat dinyatakan ke dalam persamaan parametrik

$$x = t, y = f(t), \text{ dengan } x'(t) = 1, x''(t) = 0,$$

sehingga kurvturnya adalah

$$\kappa(t) = \frac{y''(t)}{(1 + y'(t)^2)^{3/2}}.$$

Contoh: Akan ditentukan kurvatur parabola

$$y = ax^2 + bx + c.$$

```
>function f(x) &= a*x^2+b*x+c; $y=f(x)
```

```
[0, 4.999958333473664 × 10-5 r, 1.999933334222437 × 10-4 r, 4.499662510124569 × 10-4 r, 7.998933390220841 × 10-4 r,
```

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) // kelengkungan parabola
```

Maxima said:

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

Error in:

```
... (x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) ...
```

```
>function f(x) &= x^2+x+1; $y=f(x) // akan kita plot kelengkungan parabola untuk a=b=c=1
```

```
[0, 4.999958333473664 × 10-5 r, 1.999933334222437 × 10-4 r, 4.499662510124569 × 10-4 r, 7.998933390220841 × 10-4 r,
```

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) // kelengkungan parabola
```

Maxima said:

```
diff: second argument must be a variable; found errexp1
-- an error. To debug this try: debugmode(true);
```

Error in:

```
... (x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) ...
```

Berikut kita gambar parabola tersebut beserta kurva kelengkungan, kurva jari-jari kelengkungan dan salah satu lingkaran oskulasi di titik puncak parabola. Perhatikan, puncak parabola dan jari-jari lingkaran oskulasi di puncak parabola adalah

sehingga pusat lingkaran oskulasi adalah $(-1/2, 5/4)$.

```
>plot2d(["f(x)", "k(x)"],-2,1, color=[blue,red]);
plot2d("1/k(x)",-1.5,1,color=green,>add); ...
>
plot2d("-1/2+1/k(-1/2)*cos(x)","5/4+1/k(-1/2)*sin(x)",xmin=0,xmax=2pi,>add,color=blue);
```

Error : f(x) does not produce a real or column vector

Error generated by error() command

%ploteval:

```
error(f$|" does not produce a real or column vector");
```

adaptiveevalone:

```
s=%ploteval(g$,t;args());
```

Try "trace errors" to inspect local variables after errors.

plot2d:

```
dw/n,dw/n^2,dw/n,auto;args());
```

Untuk kurva yang dinyatakan dengan fungsi implisit

$$F(x,y)=0$$

dengan turunan-turunan parsial

$$F_x = \frac{\partial F}{\partial x}, F_y = \frac{\partial F}{\partial y}, F_{xy} = \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial x} \right), F_{xx} = \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial x} \right), F_{yy} = \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial y} \right),$$

berlaku

$$F_x dx + F_y dy = 0 \text{ atau } \frac{dy}{dx} = - \frac{F_x}{F_y},$$

sehingga kurvturnya adalah

$$\kappa = \frac{F_y^2 F_{xx} - 2 F_x F_y F_{xy} + F_x^2 F_{yy}}{(F_x^2 + F_y^2)^{3/2}}.$$

(Silakan Anda turunkan sendiri!)

Contoh 1: Parabola

$$y = ax^2 + bx + c$$

dapat dinyatakan ke dalam persamaan implisit

```

$$ax^2+bx+c-y=0. >function F(x,y) &=a*x^2+b*x+c-y; $F(x,y)

[c, 2.7777500001498 × 10-14 a r2 + 1.66665833335744 × 10-7 br - 4.999958333473664 × 10-5 r + c, 1.777706668053906

>Fx &= diff(F(x,y),x), Fxx &=diff(F(x,y),x,2), Fy &=diff(F(x,y),y), Fxy
&=diff(diff(F(x,y),x),y), Fyy &=diff(F(x,y),y,2)

Maxima said:
diff: second argument must be a variable; found errexpl
-- an error. To debug this try: debugmode(true);

Error in:
Fx &= diff(F(x,y),x), Fxx &=diff(F(x,y),x,2), Fy &=diff(F(x,y)
...
^

>function k(x) &= (Fy2*Fxx-2*Fx*Fy*Fxy+Fx2*Fyy)/(Fx2+Fy2)^(3/2); $k(x)=k(x) //
kurvatur parabola tersebut

k([0, 1.66665833335744 × 10-7 r, 1.33330666692022 × 10-6 r, 4.499797504338432 × 10-6 r, 1.066581336583994 × 10-5 r

```

Hasilnya sama dengan sebelumnya yang menggunakan persamaan parabola biasa.

Latihan

- Bukalah buku Kalkulus.
- Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi).
- Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah integral tentu dengan batas-batas yang menarik (Anda tentukan sendiri), seperti contoh-contoh tersebut.
- Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat diintegrasikan (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

(Pilih fungsinya dan gambar kurva dan benda putar yang dihasilkan. Anda dapat mencari contoh-contoh bagaimana cara menggambar benda hasilperputaran suatu kurva.)

- Gunakan integral tentu untuk menghitung panjang kurva $y=f(x)$ dari $x=a$ sampai $x=b$ dengan menggunakan rumus:

(Pilih fungsi dan gambar kurvanya.)

- Apabila fungsi dinyatakan dalam koordinat kutub $x=f(r,t)$, $y=g(r,t)$, $r=h(t)$, $x=a$ bersesuaian dengan $t=t_0$ dan $x=b$ bersesuaian dengan $t=t_1$, maka rumus di atas akan menjadi:
- Pilih beberapa kurva menarik (selain lingkaran dan parabola) dari buku kalkulus. Nyatakan setiap kurva tersebut dalam bentuk:
 - a. koordinat Kartesius (persamaan $y=f(x)$)
 - b. koordinat kutub ($r=r(\theta)$)
 - c. persamaan parametrik $x=x(t)$, $y=y(t)$
 - d. persamaan implit $F(x,y)=0$
- Tentukan kurvatur masing-masing kurva dengan menggunakan keempat representasi tersebut (hasilnya harus sama).
- Gambarkan kurva asli, kurva kurvatur, kurva jari-jari lingkaran oskulasi, dan salah satu lingkaran oskulasinya.

1. Tentukan panjang kurva dan volume benda putar kurva $y=f(x)$ yang diputar mengelilingi sumbu x dari $x=0$ sampai $x=4$

```

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

[0, 4.629560185733296 × 10-21 r3, 2.370228152344803 × 10-18 r3, 9.111269894211209 × 10-17 r3, 1.213338392913399 × 1

>$showev('integrate(pi*(f(x))^2,x,0,4))

Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexpl
#0: showev(f='integrate([0,2.143282751332691e-41*pi*r^6,5.61798149416786e-
36*pi*r^6,8.301523908515954e-33*pi*r^6,...)
-- an error. To debug this try: debugmode(true);

Error in:
$showev('integrate(pi*(f(x))^2,x,0,4)) ...
^

turunan fungsi f(x)

```

```
Error in:
$showev('integrate(f(x), x, -1, 1)) ...
```

[illegible]

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan(menggunakan titik dua “:”);
- menggunakan perintah “sequence” dan rumus barisan (suku ke -n);
- menggunakan perintah “iterate” atau “niterate”;
- menggunakan fungsi Maxima “create_list” atau “makelist” untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan:

- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni: * sum: menghitung jumlah semua elemen suatu barisan * cumsum: jumlah kumulatif suatu barisan * differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana
```

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

```
>1:2:30
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
```

Iterasi dan Barisan

EMT menyediakan fungsi iterate("g(x)", x0, n) untuk melakukan iterasi

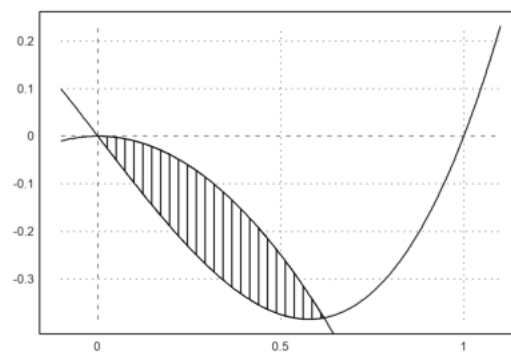
Berikut ini disajikan contoh-contoh penggunaan iterasi dan rekursi dengan EMT. Contoh pertama menunjukkan pertumbuhan dari nilai awal 1000 dengan laju pertambahan 5%, selama 10 periode.

```
>q=1.05; iterate("x*q",1000,n=10)'
```

```
1000
1050
1102.5
1157.63
1215.51
1276.28
1340.1
1407.1
1477.46
1551.33
1628.89
```

Contoh berikutnya memperlihatkan bahaya menabung di bank pada masa sekarang! Dengan bunga tabungan sebesar 6% per tahun atau 0.5% per bulan dipotong pajak 20%, dan biaya administrasi 10000 per bulan, tabungan sebesar 1 juta tanpa diambil selama sekitar 10 tahunan akan habis diambil oleh bank!

```
>r=0.005; plot2d(iterate("(1+0.8*r)*x-10000",1000000,n=130)):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-180.png

Silakan Anda coba-coba, dengan tabungan minimal berapa agar tidak akan habis diambil oleh bank dengan ketentuan bunga dan biaya administrasi seperti di atas.

Berikut adalah perhitungan minimal tabungan agar aman di bank dengan bunga sebesar r dan biaya administrasi a, pajak bunga 20%.

```
>$solve(0.8*r*A-a,A), $% with [r=0.005, a=10]
```

```
[ A=2500.0 ]
```

Berikut didefinisikan fungsi untuk menghitung saldo tabungan, kemudian dilakukan iterasi.

```
>function saldo(x,r,a) := round((1+0.8*r)*x-a,2);
```

```
>iterate({{"saldo",0.005,10}},1000,n=6)
```

```
[1000, 994, 987.98, 981.93, 975.86, 969.76, 963.64]
```

```
>iterate({{"saldo",0.005,10}},2000,n=6)
```

```
[2000, 1998, 1995.99, 1993.97, 1991.95, 1989.92, 1987.88]
```

```
>iterate({{"saldo",0.005,10}},2500,n=6)
```

```
[2500, 2500, 2500, 2500, 2500, 2500, 2500]
```

Tabungan senilai 2,5 juta akan aman dan tidak akan berubah nilai (jika tidak ada penarikan), sedangkan jika tabungan awal kurang dari 2,5 juta, lama kelamaan akan berkurang meskipun tidak pernah dilakukan penarikan uang tabungan.

```
>iterate({{"saldo",0.005,10}},3000,n=6)
```

```
[3000, 3002, 3004.01, 3006.03, 3008.05, 3010.08, 3012.12]
```

Tabungan yang lebih dari 2,5 juta baru akan bertambah jika tidak ada penarikan.

Untuk barisan yang lebih kompleks dapat digunakan fungsi “sequence()”. Fungsi ini menghitung nilai-nilai $x[n]$ dari semua nilai sebelumnya, $x[1], \dots, x[n-1]$ yang diketahui.

Berikut adalah contoh barisan Fibonacci.

```
>sequence("x[n-1]+x[n-2]",[1,1],15)
```

```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]
```

Barisan Fibonacci memiliki banyak sifat menarik, salah satunya adalah akar pangkat ke- n suku ke- n akan konvergen ke pecahan emas:

```
>$^(1+sqrt(5))/2=float((1+sqrt(5))/2)
```

$$\frac{\sqrt{5} + 1}{2} = 1.618033988749895$$

```
>plot2d(sequence("x[n-1]+x[n-2]",[1,1],250)^(1/(1:250))):
```

$$\gamma(t) = (r(t) \cos(t), r(t) \sin(t))$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-184.png

Barisan yang sama juga dapat dihasilkan dengan menggunakan loop.

```
>x=ones(500); for k=3 to 500; x[k]=x[k-1]+x[k-2]; end;
```

Rekursi dapat dilakukan dengan menggunakan rumus yang tergantung pada semua elemen sebelumnya. Pada contoh berikut, elemen ke- n merupakan jumlah $(n-1)$ elemen sebelumnya, dimulai dengan 1 (elemen ke-1). Jelas, nilai elemen ke- n adalah $2^{(n-2)}$, untuk $n=2, 4, 5, \dots$

```
>sequence("sum(x)",1,10)
```

```
[1, 1, 2, 4, 8, 16, 32, 64, 128, 256]
```

Selain menggunakan ekspresi dalam x dan n , kita juga dapat menggunakan fungsi.

Pada contoh berikut, digunakan iterasi

dengan A suatu matriks 2×2 , dan setiap $x[n]$ merupakan matriks/vektor 2×1 .

```
>A=[1,1;1,2]; function suku(x,n) := A.x[n-1]
```

```
>sequence("suku",[1;1],6)
```

Real 2 x 6 matrix

1	2	5	13	...
1	3	8	21	...

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks “matrixpower()”. Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak $\log_2(n)$.

```
>sequence("matrixpower(A,n).[1;1]",1,6)
```

Real 2 x 6 matrix

1	5	13	34	...
1	8	21	55	...

Spiral Theodorus

image: Spiral_of_Theodorus.png

Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

yang menghasilkan barisan bilangan kompleks.

```
>function g(n) := 1+I/sqrt(n)
```

Rekursinya dapat dijalankan sebanyak 17 untuk menghasilkan barisan 17 bilangan kompleks, kemudian digambar bilangan-bilangan kompleksnya.

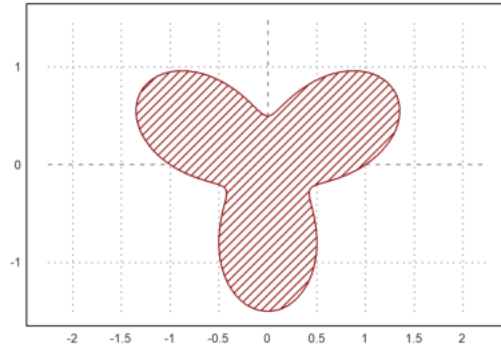
```
>x=sequence("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox($"Spiral\
Theodorus"),0.4):
```

$$r(t) = 1 + \frac{\sin(3t)}{2}, \quad 0 \leq t \leq 2\pi.$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-185.png

Selanjutnya dihubungkan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end:
```



images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-186.png

>

Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memerlukan indeks dan bilangan kompleks. Dalam hal ini digunakan vektor kolom pada bidang.

```
>function gstep (v) ...
```

```
w=[-v[2];v[1]];
return v+w/norm(w);
endfunction
```

Jika dilakukan iterasi 16 kali dimulai dari [1;0] akan didapatkan matriks yang memuat vektor-vektor dari setiap iterasi.

```
>x=iterate("gstep",[1;0],16); plot2d(x[1],x[2],r=3.5,>points):
```

$$r(t) = \frac{\sin(3t)}{2} + 1$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-187.png

Kekonvergenan

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```
>iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1.
```

```
0.739085133216
```

Iterasi tersebut konvergen ke penyelesaian persamaan

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.

```
>hasil := iterate("cos(x)",1,2) //iterasi x(n+1)=cos(x(n)), dengan interval awal (1, 2)
```

```
~0.739085133211,0.7390851332133~
```

Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan $x=\cos(x)$.

```
>h=expand(hasil,100), cos(h) << h
```

```
~0.73908513309,0.73908513333~
1
```

Iterasi juga dapat digunakan pada fungsi yang didefinisikan.

```
>function f(x) := (x+2/x)/2
```

Iterasi $x(n+1)=f(x(n))$ akan konvergen ke akar kuadrat 2.

```
>iterate("f",2), sqrt(2)
```

```
1.41421356237
1.41421356237
```

Jika pada perintah `iterate` diberikan tambahan parameter `n`, maka hasil iterasinya akan ditampilkan mulai dari iterasi pertama sampai ke-`n`.

```
>iterate("f",2,5)

[2, 1.5, 1.41667, 1.41422, 1.41421, 1.41421]
```

Untuk iterasi ini tidak dapat dilakukan terhadap interval.

```
>niterate("f",1,2,5)

[ ~1,2~, ~1,2~, ~1,2~, ~1,2~, ~1,2~, ~1,2~ ]
```

Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya adalah perhitungannya dengan interval bersifat terlalu longgar. Untuk meningkatkan perhitungan pada ekspresi dapat digunakan pembagian intervalnya, menggunakan fungsi `ieval()`.

```
>function s(x) := ieval("(x+2/x)/2",x,10)
```

Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal, dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang memuat akar persamaan:

Satu-satunya solusi adalah

```
>iterate("s",1,2)

~1.41421356236,1.41421356239~
```

Fungsi `"iterate()"` juga dapat bekerja pada vektor. Berikut adalah contoh fungsi vektor, yang menghasilkan rata-rata aritmetika dan rata-rata geometri.

Iterasi ke-`n` disimpan pada vektor kolom `x[n]`.

```
>function g(x) := [(x[1]+x[2])/2;sqrt(x[1]*x[2])]
```

Iterasi dengan menggunakan fungsi tersebut akan konvergen ke rata-rata aritmetika dan geometri dari nilai-nilai awal.

```
>iterate("g",[1;5])

2.60401
2.60401
```

Hasil tersebut konvergen agak cepat, seperti kita cek sebagai berikut.

```
>iterate("g",[1;5],4)

1      3      2.61803      2.60403      2.60401
5      2.23607      2.59002      2.60399      2.60401
```

Iterasi pada interval dapat dilakukan dan stabil, namun tidak menunjukkan bahwa limitnya pada batas-batas yang dihitung.

```
>iterate("g",[1;5],4)

Interval 2 x 5 matrix

~0.999999999999999778,1.00000000000000022~ ...
~4.99999999999999911,5.00000000000000089~ ...
```

Iterasi berikut konvergen sangat lambat.

```
>iterate("sqrt(x)",2,10)

[2, 1.41421, 1.18921, 1.09051, 1.04427, 1.0219, 1.01089,
1.00543, 1.00271, 1.00135, 1.00068]
```

Kekonvergenan iterasi tersebut dapat dipercepat dengan percepatan Steffenson:

```
>steffenson("sqrt(x)",2,10)

[1.04888, 1.00028, 1, 1]
```

Iterasi menggunakan Loop yang ditulis Langsung

Berikut adalah beberapa contoh penggunaan loop untuk melakukan iterasi yang ditulis langsung pada baris perintah.

```
>x=2; repeat x=(x+2/x)/2; until x^2~2; end; x,

1.41421356237
```

Penggabungan matriks menggunakan tanda `"|"` dapat digunakan untuk menyimpan semua hasil iterasi.

```
>v=[1]; for i=2 to 8; v=v/(v[i-1]*i); end; v,
```

```
[1, 2, 6, 24, 120, 720, 5040, 40320]
```

hasil iterasi juga dapat disimpan pada vektor yang sudah ada.

```
>v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]*i; end; ...
> plot2d(v,logplot=1); textbox($(&log(n)),x=0.5):
```

$$f_x(t) = \cos t \left(\frac{\sin(3t)}{2} + 1 \right)$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-188.png

```
>A=[0.5,0.2;0.7,0.1]; b=[2;2]; ...
> x=[1;1]; repeat xnew=A.x-b; until all(xnew~=x); x=xnew; end; ...
> x,

-7.09677
-7.74194
```

Iterasi di dalam Fungsi

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.

Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

```
>function map hiter(f$,x0) ...
```

```
x=x0;
maxiter=0;
repeat
  xnew=f$(x);
  maxiter=maxiter+1;
  until xnew~=x;
  x=xnew;
end;
return maxiter;
endfunction
```

Misalnya, berikut adalah iterasi untuk mendapatkan hampiran akar kuadrat 2, cukup cepat, konvergen pada iterasi ke-5, jika dimulai dari hampiran awal 2.

```
>hiter("(x+2/x)/2",2)
```

```
5
```

Karena fungsinya didefinisikan menggunakan "map". maka nilai awalnya dapat berupa vektor.

```
>x=1.5:0.1:10; hasil=hiter("(x+2/x)/2",x); ...
> plot2d(x,hasil):
```

$$f_y(t) = \sin t \left(\frac{\sin(3t)}{2} + 1 \right)$$

images/Alya%20Putri%20Medilasari_23030630018_EMTKalkulus-189.png

Dari gambar di atas terlihat bahwa kekonvergenan iterasinya semakin lambat, untuk nilai awal semakin besar, namun penambahannya tidak kontinu. Kita dapat menemukan kapan maksimum iterasinya bertambah.

```
>hasil[1:10]
```

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

```
>x[nonzeros(differences(hasil))]
```

```
[1.5, 2, 3.4, 6.6]
```

maksimum iterasi sampai konvergen meningkat pada saat nilai awalnya 1.5, 2, 3.4, dan 6.6.

Contoh berikutnya adalah metode Newton pada polinomial kompleks berderajat 3.

```
>p &= x^3-1; newton &= x-p/diff(p,x); $newton
```

```
Maxima said:
```

```
diff: second argument must be a variable; found errexpl
-- an error. To debug this try: debugmode(true);
```

```
Error in:
```

```
p &= x^3-1; newton &= x-p/diff(p,x); $newton ...
^
```

Selanjutnya didefinisikan fungsi untuk melakukan iterasi (aslanya 10 kali).

```
>function iterasi(f$,x,n=10) ...
```

```

loop 1 to n; x:=f$(x); end;
return x;
endfunction

```

Kita mulai dengan menentukan titik-titik grid pada bidang kompleksnya.

```
>r=1.5; x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z);
```

```

Function newton needs at least 3 arguments!
Use: newton (f$: call, df$: call, x: scalar complex {, y: number, eps: none})
Error in:
... x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z); ...

```

Berikut adalah akar-akar polinomial di atas.

```
>z=&solve(p)()
```

```

Maxima said:
solve: more equations than unknowns.
Unknowns given :
[r]
Equations given:
errexpl
-- an error. To debug this try: debugmode(true);

```

```

Error in:
z=&solve(p)() ...

```

Untuk menggambar hasil iterasinya, dihitung jarak dari hasil iterasi ke-10 ke masing-masing akar, kemudian digunakan untuk menghitung warna yang akan digambar, yang menunjukkan limit untuk masing-masing nilai awal.

Fungsi `plotrgb()` menggunakan jendela gambar terkini untuk menggambar warna RGB sebagai matriks.

```

>C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1)); ...
> plot2d(none,-r,r,-r,r); plotrgb(C):

```

```

Variable W not found!
Error in:
C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1)) ...

```

Iterasi Simbolik

Seperti sudah dibahas sebelumnya, untuk menghasilkan barisan ekspresi simbolik dengan Maxima dapat digunakan fungsi `makelist()`.

```
>&powerdisp:true // untuk menampilkan deret pangkat mulai dari suku berpangkat terkecil
```

```
true
```

```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Taylor untuk e^x
```

```

Maxima said:
taylor: 0.1539740213994798*r cannot be a variable.
-- an error. To debug this try: debugmode(true);

```

```

Error in:
deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // baris ...

```

Untuk mengubah barisan deret tersebut menjadi vektor string di EMT digunakan fungsi `mxm2str()`. Selanjutnya, vektor string/ekspresi hasilnya dapat digambar seperti menggambar vektor ekspresi pada EMT.

```
>plot2d("exp(x)",0,3); // plot fungsi aslinya, e^x
```

```
>plot2d(mxm2str("deret"),>add,color=4:6): // plot ketiga deret taylor hampiran fungsi tersebut
```

```

Maxima said:
length: argument cannot be a symbol; found deret
-- an error. To debug this try: debugmode(true);

```

```

mxmeval:
return evaluate(mxm(s));
Try "trace errors" to inspect local variables after errors.
mxm2str:
n=mxmeval("length(VV)");

```

Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

```
>$deret[3]
```

```

deret3
>plot2d(["exp(x)",&deret[1],&deret[2],&deret[3]],0,3,color=1:4):

deret is not a variable!
Error in expression: deret[1]
%ploteval:
  y0=f$(x[1],args());
Try "trace errors" to inspect local variables after errors.
plot2d:
  u=u_(%ploteval(xx[#],t,args()));

>$sum(sin(k*x)/k,k,1,5)

```

$$\left[0, \sin \left(1.6666583335744 \times 10^{-7} r \right) + \frac{\sin \left(3.333316666714881 \times 10^{-7} r \right)}{2} + \frac{\sin \left(4.999975000072321 \times 10^{-7} r \right)}{3} + \frac{\sin \left(6.666641666692022 \times 10^{-7} r \right)}{4} + \frac{\sin \left(8.333308333307778 \times 10^{-7} r \right)}{5} \right]$$

Berikut adalah cara menggambar kurva

```

>plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi):

Maxima output too long!
Error in:
plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi): ...
^

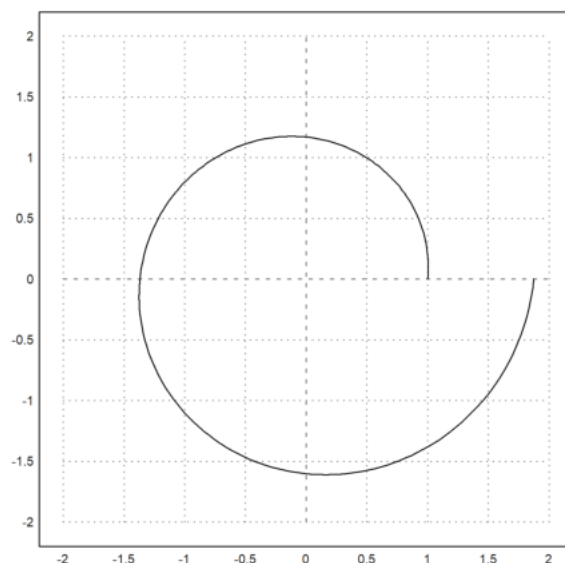
```

Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan kita akan menggambar kurva

```

>x=linspace(0,2pi,1000); k=1:100; y=sum(sin(k*x)/k)'; plot2d(x,y):

```



Tabel Fungsi

Terdapat cara menarik untuk menghasilkan barisan dengan ekspresi Maxima. Perintah `mxmtable()` berguna untuk menampilkan dan menggambar barisan dan menghasilkan barisan sebagai vektor kolom.

Sebagai contoh berikut adalah barisan turunan ke- n x^x di $x=1$.

```

>mxmtable("diffat(x^x,x=1,n)","n",1,8,frac=1);

Maxima said:
diff: second argument must be a variable; found errexp1
#0: diffat(expr=[0,1.6666583335744e-7*r,1.33330666692022e-6*r,4.499797504338432e-6*r,1.066581336583994e-5*r,2.08307...],x=
[[0,1.6666583335744e-7*r,1.33330666692022e-6*r,4.499797504338432e-6*r,1.066581336583994e-5*r,2.0830...])
-- an error. To debug this try: debugmode(true);

%mxmevtable:
  return mxm("@expr,@var=@value");
Try "trace errors" to inspect local variables after errors.
mxmtable:
  y[#,1]=%mxmevtable(expr,var,x[#]);

>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:menghitung
deret secara simbolik

```

$$\sum_{k=1}^n k = \frac{n(1+n)}{2}$$

>\$`sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))

$$\sum_{k=0}^{\infty} \frac{1}{k+3^k} = \sum_{k=0}^{\infty} \frac{1}{k+3^k}$$

Di sini masih gagal, hasilnya tidak dihitung. >\$`sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai ekspresi

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

>\$`sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))

$$\sum_{k=1}^{\infty} \frac{(-1)^{-1+k}}{k} = - \sum_{x=1}^{\infty} \frac{(-1)^x}{x}$$

Di sini masih gagal, hasilnya tidak dihitung.

>\$`sum((-1)^k/(2*k-1), k, 1, inf) = factor(ev(sum((-1)^k/(2*k-1), k, 1, inf),simpsum=true))

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{-1+2k} = \sum_{k=1}^{\infty} \frac{(-1)^k}{-1+2k}$$

>\$ev(sum(1/n!, n, 0, inf),simpsum=true)

$$\sum_{n=0}^{\infty} \frac{1}{n!}$$

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

>&assume(abs(x)<1); \$`sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forget(abs(x)<1);

Answering "Is -94914474571+15819*r positive, negative or zero?" with "positive"

Maxima said:

sum: sum is divergent.

-- an error. To debug this try: debugmode(true);

Error in:

... k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forget(abs ...

Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1.

>\$`sum(x^k/k!,k,0,inf)=ev(sum(x_k/k!,k,0,inf),simpsum=true)

$$\left[0, \sum_{k=0}^{\infty} \frac{(1.66665833335744 \times 10^{-7})^k r^k}{k!}, \sum_{k=0}^{\infty} \frac{(1.33330666692022 \times 10^{-6})^k r^k}{k!}, \sum_{k=0}^{\infty} \frac{(4.499797504338432 \times 10^{-6})^k r^k}{k!}, \sum_{k=0}^{\infty} \right]$$

>\$limit(sum(x^k/k!,k,0,n),n,inf)

$$\left[0, \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{(1.66665833335744 \times 10^{-7})^k r^k}{k!}, \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{(1.33330666692022 \times 10^{-6})^k r^k}{k!}, \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{(4.499797504338432 \times 10^{-6})^k r^k}{k!}, \right]$$

>function d(n) &= sum(1/(k^2-k),k,2,n); \$`d(n)=d(n)

$$d(n) = \sum_{k=2}^n \frac{1}{-k + k^2}$$

>\$d(10)=ev(d(10),simpsum=true)

$$\sum_{k=2}^{10} \frac{1}{-k + k^2} = \frac{9}{10}$$

>\$d(100)=ev(d(100),simpsum=true)

$$\sum_{k=2}^{100} \frac{1}{-k + k^2} = \frac{99}{100}$$

Deret Taylor

Deret Taylor suatu fungsi f yang diferensiabel sampai tak hingga di sekitar x=a adalah:

>\$`e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai suku ke-11

Maxima said:

taylor: 0.1539740213994798*r cannot be a variable.

```
-- an error. To debug this try: debugmode(true);

Error in:
$'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x= ...
^

>$'log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1
```

Visualisasi dan Perhitungan Geometri dengan EMT

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan geometri tersebut disimpan di dalam file program “geometry.e”, sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

```
>load geometry
```

```
Numerical and symbolic geometry.
```

Fungsi-fungsi Geometri

Fungsi-fungsi untuk Menggambar Objek Geometri:

```
defaultd:=textheight()*1.5: nilai asli untuk parameter d
setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang koordinat
setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas sumbu-x dan y adalah -r sd
r
plotPoint (P, “P”): menggambar titik P dan diberi label “P”
plotSegment (A,B, “AB”, d): menggambar ruas garis AB, diberi label “AB” sejauh d
plotLine (g, “g”, d): menggambar garis g diberi label “g” sejauh d
plotCircle (c, “c”, v,d): Menggambar lingkaran c dan diberi label “c”
plotLabel (label, P, V, d): menuliskan label pada posisi P
```

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

```
turn(v, phi): memutar vektor v sejauh phi
turnLeft(v): memutar vektor v ke kiri
turnRight(v): memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang vektor v dan w.
lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh. ax+by=c.
lineWithDirection(A,v): garis melalui A searah vektor v
getLineDirection(g): vektor arah (gradien) garis g
getNormal(g): vektor normal (tegak lurus) garis g
getPointOnLine(g): titik pada garis g
perpendicular(A, g): garis melalui A tegak lurus garis g
parallel (A, g): garis melalui A sejajar garis g
lineIntersection(g, h): titik potong garis g dan h
projectToLine(A, g): proyeksi titik A pada garis g
distance(A, B): jarak titik A dan B
distanceSquared(A, B): kuadrat jarak A dan B
quadrance(A, B): kuadrat jarak A dan B
areaTriangle(A, B, C): luas segitiga ABC
computeAngle(A, B, C): besar sudut <ABC
angleBisector(A, B, C): garis bagi sudut <ABC
circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r
getCircleCenter(c): pusat lingkaran c
getCircleRadius(c): jari-jari lingkaran c
circleThrough(A,B,C): lingkaran melalui A, B, C
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan lingkaran c
circleCircleIntersections (c1, c2): titik potong lingkaran c1 dan c2
planeThrough(A, B, C): bidang melalui titik A, B, C
```

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

```
getLineEquation (g,x,y): persamaan garis g dinyatakan dalam x dan y
getHesseForm (g,x,y,A): bentuk Hesse garis g dinyatakan dalam x dan y dengan titik A
pada
sisi positif (kanan/atas) garis
quad(A,B): kuadrat jarak AB
spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni  $\sin(\alpha)^2$  dengan
alpha sudut yang menghadap sisi a.
crosslaw(a,b,c,sa): persamaan 3 quads dan 1 spread pada segitiga dengan panjang sisi a,
b, c.
triplespread(sa,sb,sc): persamaan 3 spread sa,sb,sc yang membentuk suatu segitiga
doublespread(sa): Spread sudut rangkap  $2\phi$ , dengan  $sa=\sin(\phi)^2$  spread a.
```

Contoh 1: Luas, Lingkaran Luar, Lingkaran Dalam Segitiga

Untuk menggambar objek-objek geometri, langkah pertama adalah menentukan rentang sumbu-sumbu koordinat. Semua objek geometri akan digambar pada satu bidang koordinat, sampai didefinisikan bidang koordinat yang baru.

```
>setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
```

Sekarang tetapkan tiga titik dan plotkan.

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
```

```
>B=[0,1]; plotPoint(B,"B");
```

```
>C=[2,2]; plotPoint(C,"C");
```

Kemudian tiga segmen.

```
>plotSegment(A,B,"c"); // c=AB
```

```
>plotSegment(B,C,"a"); // a=BC
```

```
>plotSegment(A,C,"b"); // b=AC
```

Fungsi geometri mencakup fungsi untuk membuat garis dan lingkaran. Format untuk garis adalah [a,b,c], yang merepresentasikan garis dengan persamaan $ax+by=c$.

```
>lineThrough(B,C) // garis yang melalui B dan C
```

```
[-1, 2, 2]
```

Hitung garis tegak lurus yang melalui A pada BC.

```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

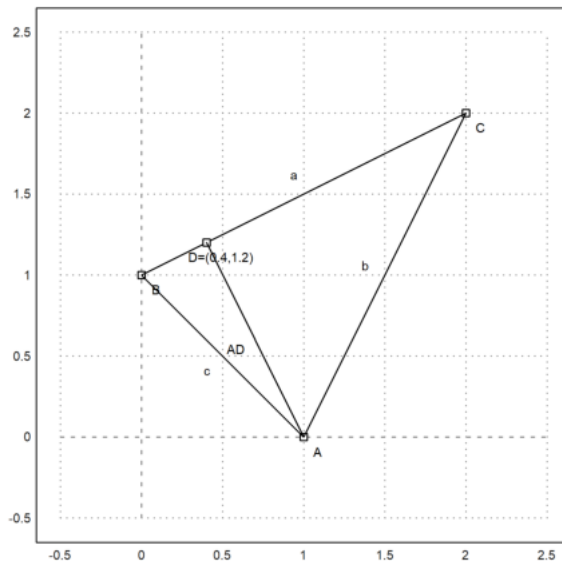
Dan persinggungannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Plotkan itu.

```
>plotPoint(D,value=1); // koordinat D ditampilkan
```

```
>aspect(1); plotSegment(A,D); // tampilkan semua gambar hasil plot...()
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-001.png

Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2} AD \cdot BC.$$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
```

1.5

Bandingkan dengan rumus determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langsung dengan fungsi
```

1.5

Cara lain menghitung luas segitiga ABC:

```
>distance(A,D)*distance(B,C)/2
```

1.5

Sudut pada C.

```
>degprint(computeAngle(B,C,A))
```

36°52'11.63''

Sekarang, lingkarilah segitiga tersebut.

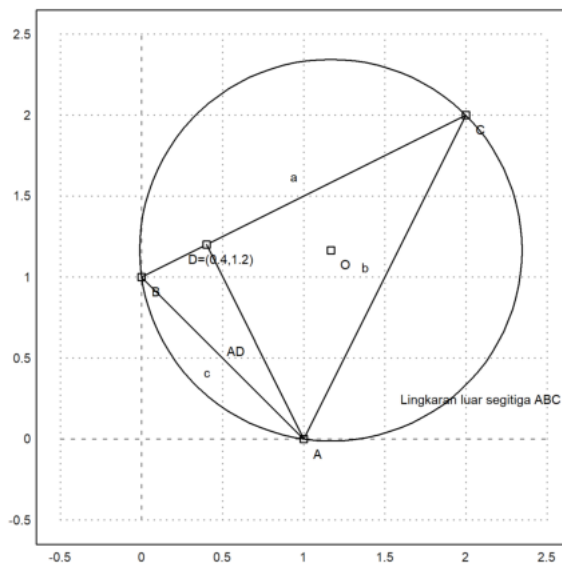
```
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
```

```
>R=getCircleRadius(c); // jari2 lingkaran luar
```

```
>O=getCircleCenter(c); // titik pusat lingkaran c
```

```
>plotPoint(O,"O"); // gambar titik "O"
```

```
>plotCircle(c,"Lingkaran luar segitiga ABC");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-002.png

Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>O, R
```

```
[1.16667, 1.16667]
```

```
1.17851130198
```

Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

```
>l=angleBisector(A,C,B); // garis bagi <ACB
```

```
>g=angleBisector(C,A,B); // garis bagi <CAB
```

```
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

```
[0.86038, 0.86038]
```

Tambahkan semuanya ke plot.

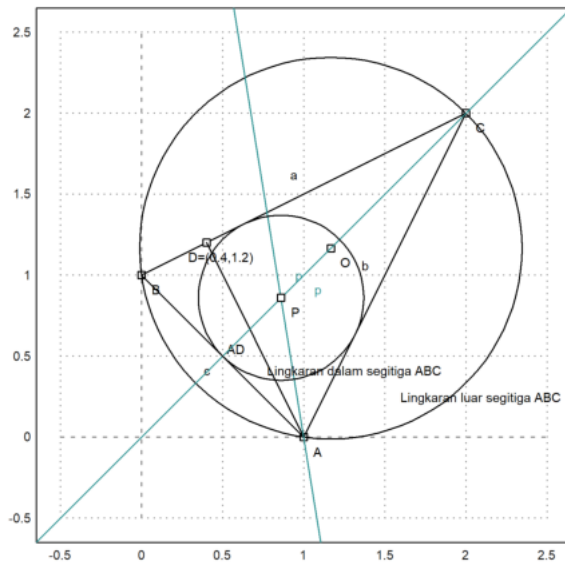
```
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut
```

```
>plotPoint(P,"P"); // gambar titik potongnya
```

```
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

```
0.509653732104
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"); // gambar lingkaran dalam
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-003.png

Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.
2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. Merupakan segitiga apakah itu?
3. Hitung luas segitiga tersebut.
4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.
5. Gambar jari-jari lingkaran dalam.
6. Hitung luas lingkaran luar dan luas lingkaran dalam segitiga ABC. Adakah hubungan antara luas kedua lingkaran tersebut dengan luas segitiga ABC?

Jawaban

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

>reset;

>load geometry

Numerical and symbolic geometry.

>setPlotRange(-3,6,-3,6);

>A=[-2,1]; plotPoint(A,"A");

>B=[1,-2]; plotPoint(B,"B");

>C=[4,4]; plotPoint(C,"C");

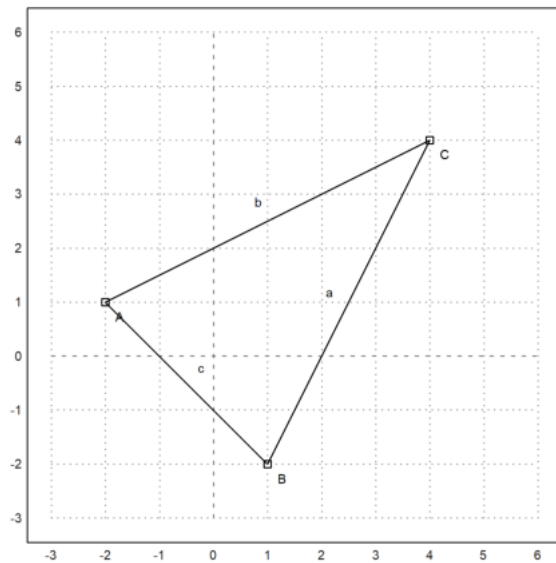
2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. Merupakan segitiga apakah itu?

>plotSegment(A,B,"c");

>plotSegment(B,C,"a");

>plotSegment(A,C,"b");

>aspect(1);



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-004.png

3. Hitung luas segitiga tersebut.

```
>areaTriangle(A,B,C)
```

13.5

4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

```
>l=angleBisector(A,C,B);
```

```
>g=angleBisector(C,A,B);
```

```
>P=lineIntersection(l,g)
```

```
[0.581139, 0.581139]
```

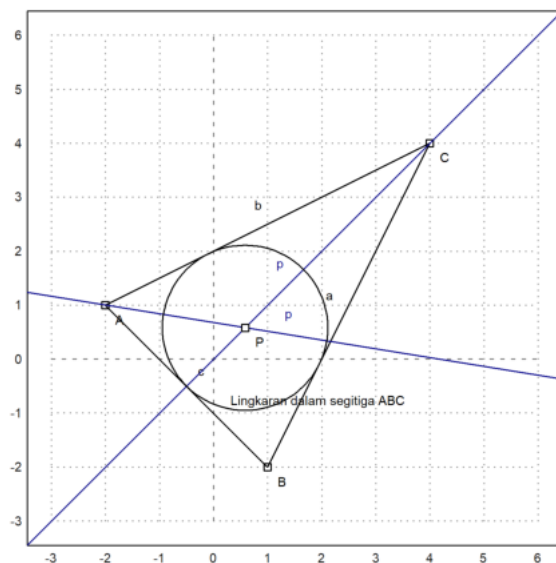
```
>color(4); plotLine(l); plotLine(g); color(1);
```

```
>plotPoint(P,"P");
```

```
>r=norm(P-projectToLine(P,lineThrough(A,B)))
```

```
1.52896119631
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC");
```

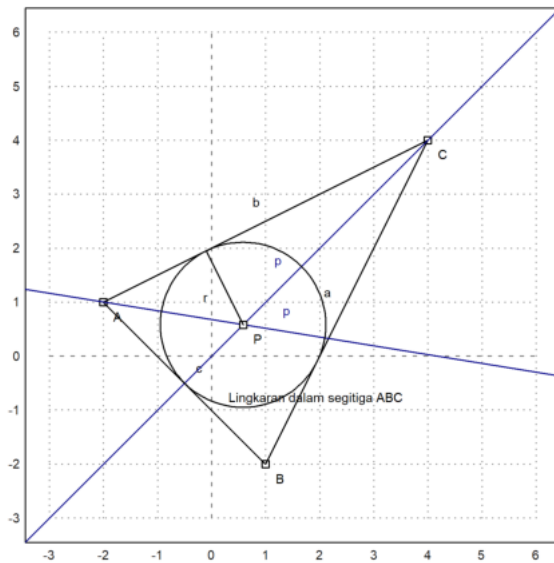


images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-005.png

Terbukti bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

5. Gambar jari-jari lingkaran dalam.

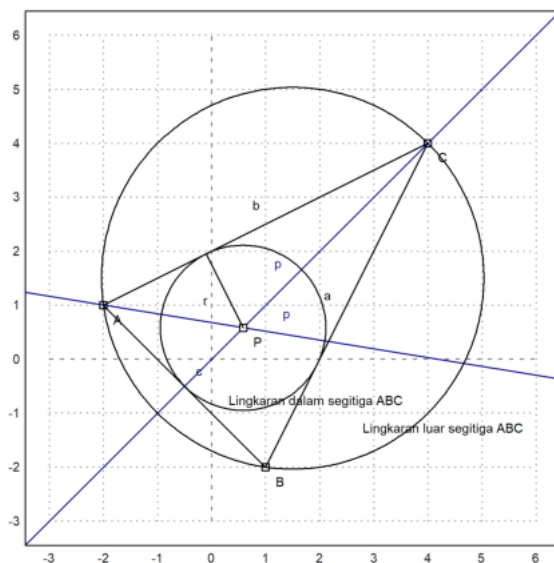
```
>plotSegment(P,projectToLine(P, lineThrough(A, C)),"r");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-006.png

6. Hitung luas lingkaran luar dan luas lingkaran dalam segitiga ABC. Adakah hubungan antara luas kedua lingkaran tersebut dengan luas segitiga ABC?

```
>c=circleThrough(A,B,C);
>plotCircle(c, "Lingkaran luar segitiga ABC");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-007.png

Contoh 2: Geometri Smbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometri.e menyediakan fungsi-fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, kita dapat menggunakan komputasi simbolik sekarang.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi menyediakan komputasi simbolis.

```
>c &= lineThrough(B,C) // c=BC
```

```
[- 1, 2, 2]
```

Kita bisa mendapatkan persamaan untuk sebuah garis dengan mudah.

```
>$getLineEquation(c,x,y), $solve(%y) | expand // persamaan garis c
```

$$[y = \frac{x}{2} + 1]$$

$$\left[y = \frac{x}{2} + 1 \right]$$

images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-009.png

```
>$getLineEquation(lineThrough([x1,y1],[x2,y2]),x,y), $solve(%y) // persamaan garis
melalui(x1, y1) dan (x2, y2)
```

$$[y = \frac{-(x_1 - x)y_2 - (x - x_2)y_1}{x_2 - x_1}]$$

$$\left[y = \frac{-(x_1 - x)y_2 - (x - x_2)y_1}{x_2 - x_1} \right]$$

```
>
```

```
getLineEquation(lineThrough(A, [x1, y1]), x, y) // persamaangaris melalui Adan(x1, y1)
(x1 - 1) y - x y1 = -y1 $>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak
lurus BC
```

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

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

$$\begin{bmatrix} 2 & 6 \\ - & - \\ 5 & 5 \end{bmatrix}$$

```
>projectToLine(A, lineThrough(B, C)) // proyeksi Apada BC [ 2/5, 6/5 ] distance(A,Q)
// jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C);
```

```
cc // (titikpusatdan jari - jari)lingkaran melalui A, B, C [ 7/6, 7/6, -5/3√2 ]
$r &= getCircleRadius(cc); $r, $float(r) // tampilkan nilai jari-jari
```

```
1.178511301977579
```

```
1.178511301977579
```

```
>computeAngle(A, C, B) // nilai < ACBarccos ( 4/5)
solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A));
```

```
P // titikpotong 2 garis bagisudut [ √2√5+2/6, √2√5+2/6 ] $>P() // hasilnya sama dengan
perhitungan sebelumnya
```

```
[0.86038, 0.86038]
```

Intersecting Lines and Circles

Tentu saja, kita juga bisa memotong garis dengan lingkaran, dan lingkaran dengan lingkaran.

```
>A &:= [1,0]; c:=circleWithCenter(A,4);
```

```
>B &:= [1,2]; C &:= [2,1]; l:=lineThrough(B,C);
```

```
>setPlotRange(5); plotCircle(c); plotLine(l);
```

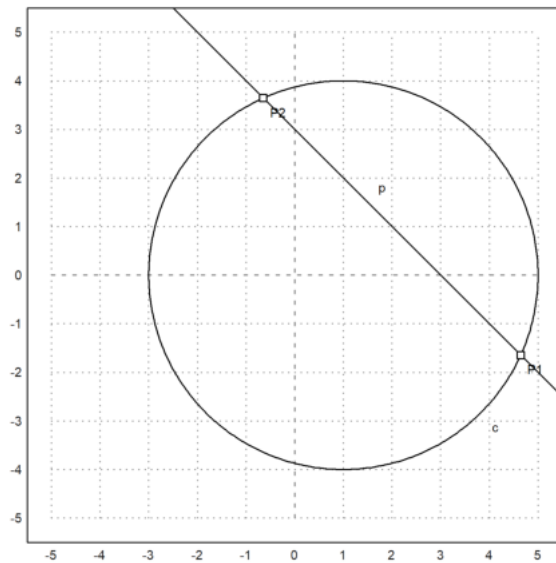
Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik perpotongan.

```
>{P1,P2,f}=lineCircleIntersections(l,c);
```

```
>P1, P2, f
```

```
[4.64575, -1.64575]
[-0.645751, 3.64575]
2
```

```
>plotPoint(P1); plotPoint(P2):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-021.png

Hal yang sama pada Maxima.

```
>c:= circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

```
[1, 0, 4]
```

```
>l:= lineThrough(B,C) // garis l melalui B dan C
```

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

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan garis l
```

```
[[ $\sqrt{7} + 2, 1 - \sqrt{7}$ ], [ $2 - \sqrt{7}, \sqrt{7} + 1$ ]]$
```

Akan ditunjukkan bahwa sudut-sudut yang menghadap busur yang sama adalah sama besar.

```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
```

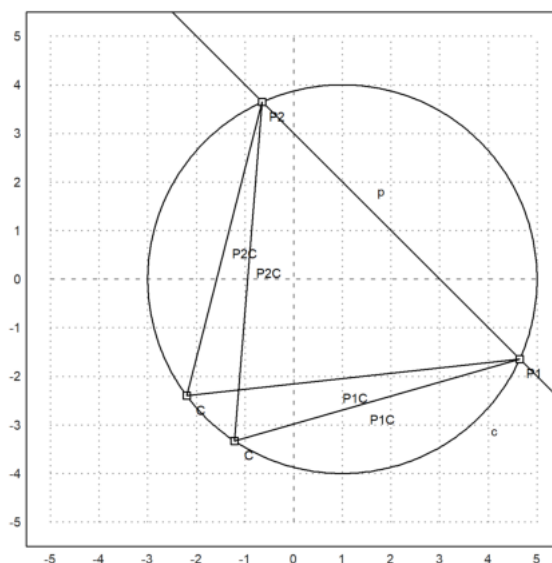
```
>degprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
```

```
>degprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```



```
>insimg;
```

Garis Sumbu

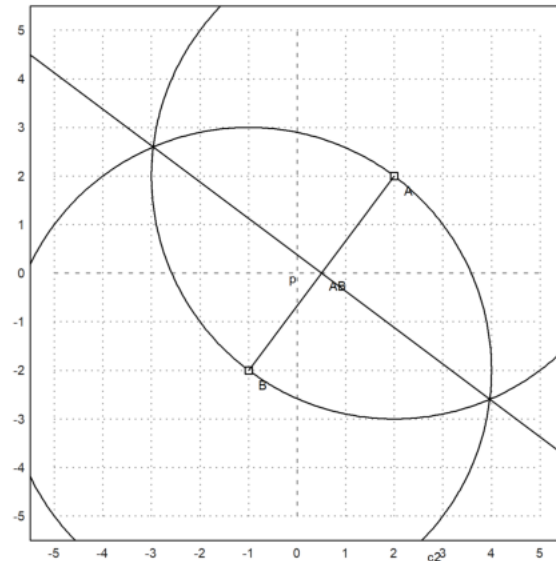
Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B.
2. Gambar lingkaran dengan pusat B melalui A.
3. Tarik garis melalui kedua titik potong kedua lingkaran tersebut. Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

```

>A=[2,2]; B=[-1,-2];
>c1=circleWithCenter(A,distance(A,B));
>c2=circleWithCenter(B,distance(A,B));
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(5); plotCircle(c1); plotCircle(c2);
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l);

```



Selanjutnya, kami melakukan hal yang sama di Maxima dengan koordinat umum.

```

>A &= [a1,a2]; B &= [b1,b2];
>c1 &= circleWithCenter(A,distance(A,B));
>c2 &= circleWithCenter(B,distance(A,B));
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];

```

Persamaan untuk persimpangan cukup rumit. Tetapi kita dapat menyederhanakannya, jika kita menyelesaikan untuk y.

```

>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)

```

$$[y = \frac{-(2b_1 - 2a_1)x + b_1^2 + b_2^2 - a_2^2 - a_1^2}{2b_2 - 2a_2}]$$$

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sama sekali berbeda.

```

>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)

```

$$[y = \frac{-(2b_1 - 2a_1)x + b_1^2 + b_2^2 - a_2^2 - a_1^2}{2b_2 - 2a_2}]$$$

```

>h &= getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)

```

$$[y = \frac{(b_2 - a_2)x - a_1b_2 + a_2b_1}{b_1 - a_1}]$$$

Perhatikan hasil kali gradien garis g dan h adalah:

$$\frac{-(b_1 - a_1)}{(b_2 - a_2)} \times \frac{(b_2 - a_2)}{(b_1 - a_1)} = -1.$$

Artinya kedua garis tegak lurus.

Contoh 3: Rumus Heron

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a , b dan c adalah:

$$L = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{dengan } s = (a+b+c)/2,$$

atau bisa ditulis dalam bentuk lain:

$$L = \frac{1}{4} \sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}$$

Untuk membuktikan hal ini kita misalkan $C(0,0)$, $B(a,0)$ dan $A(x,y)$, $b=AC$, $c=AB$. Luas segitiga ABC adalah

$$L_{\triangle ABC} = \frac{1}{2} a \times y.$$

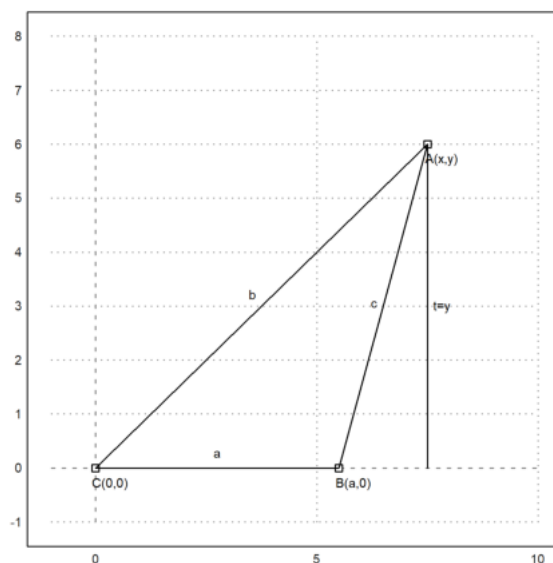
Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2 + y^2 = b^2, \quad (x-a)^2 + y^2 = c^2.$$

```
>setPlotRange(-1,10,-1,8); plotPoint([0,0], "C(0,0)"); plotPoint([5.5,0], "B(a,0)"); ...
>plotPoint([7.5,6], "A(x,y)");

>plotSegment([0,0],[5.5,0], "a",25); plotSegment([5.5,0],[7.5,6], "c",15); ...
>plotSegment([0,0],[7.5,6], "b",25);

>plotSegment([7.5,6],[7.5,0], "t=y",25);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-029.png

```
>&assume(a>0); sol &= solve([x^2+y^2=b^2,(x-a)^2+y^2=c^2],[x,y])
```

[]

ekstrak solusi y .

```
>ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2))
```

Maxima said:

part: invalid index of list or matrix.

-- an error. To debug this try: debugmode(true);

Error in:

```
ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2)) ...
^
```

Kami mendapatkan rumus Heron.

```
>function H(a,b,c) &= sqrt(factor(ysol*a/2)^2)); 'H(a,b,c) = H(a,b,c)
```

```
H(a,b,[1,0,4]) =  $\frac{a|ysol|}{2}$  'Luas=H(2,5,6) // luas segitiga dengan panjang sisi-sisi 2, 5, 6
```

$$\text{Luas} = |ysol|$$

Tentu saja, setiap segitiga persegi panjang adalah kasus yang terkenal.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

Variable or function $ysol$ not found.

Try "trace errors" to inspect local variables after errors.

H:

```

useglobal; return a*abs(ysol)/2
Error in:
H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...
^

```

Dan juga jelas, bahwa ini adalah segitiga dengan luas maksimal dan kedua sisi 3 dan 4.

```

>aspect(1.5); plot2d(&H(3,4,x),1,7): // Kurva luas segitiga dengan panjang sisi 3, 4, x
(1<= x <=7)

```

```

Variable or function ysol not found.
Error in expression: 3*abs(ysol)/2
%ploteval:
y0=f$(x[1],args());
adaptiveevalone:
s=%ploteval(g$,t,args());
Try "trace errors" to inspect local variables after errors.
plot2d:
dw/n,dw/n^2,dw/n,auto,args());

```

Kasus umum juga bisa digunakan.

```

>$solve(diff(H(a,b,c)^2,c)=0,c)

```

```

Maxima said:
diff: second argument must be a variable; found [1,0,4]
-- an error. To debug this try: debugmode(true);

```

```

Error in:
$solve(diff(H(a,b,c)^2,c)=0,c) ...
^

```

Sekarang mari kita cari himpunan semua titik di mana $b+c=d$ untuk suatu konstanta d . Sudah diketahui bahwa ini adalah sebuah elips.

```

>s1 &= subst(d-c,b,sol[2]); $s1

```

```

Maxima said:
part: invalid index of list or matrix.
-- an error. To debug this try: debugmode(true);

```

```

Error in:
s1 &= subst(d-c,b,sol[2]); $s1 ...
^

```

Dan membuat fungsi-fungsi ini.

```

>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); f y(a, c, d)
0$0

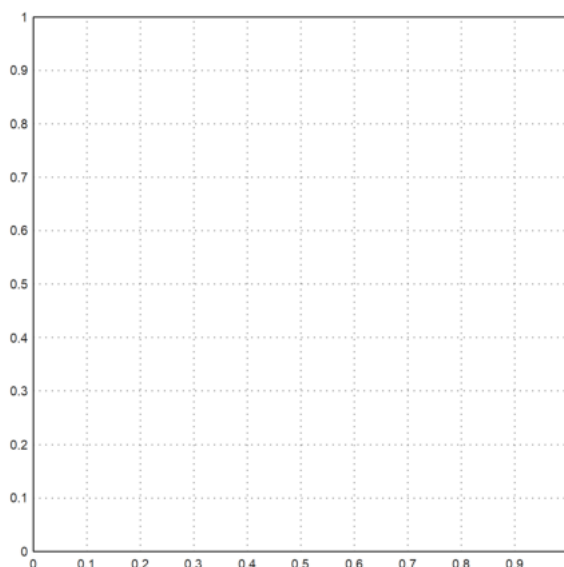
```

Sekarang kita dapat menggambar himpunan tersebut. Sisi b bervariasi dari 1 hingga 4. Sudah diketahui bahwa kita mendapatkan sebuah elips.

```

>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):

```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-034.png

We can check the general equation for this ellipse, i.e.

where (x_m, y_m) is the center, and u and v are the half axes.

```
>
ratsimp((fx(a, c, d) - a/2)^2/u^2 + fy(a, c, d)^2/v^2 with [u = d/2, v = sqrt(d^2 - a^2)/2])

$$\frac{a^2}{d^2}$$

Kita melihat bahwa tinggi dan luas segitiga adalah maksimal untuk  $x=0$ . Dengan demikian, luas segitiga dengan  $a+b+c=d$  adalah maksimal, jika segitiga tersebut sama sisi. Kita ingin membuktikannya secara analitis.
```

```
>eqns &= [diff(H(a,b,d-(a+b))^2,a)=0,diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

$$\left[\frac{a}{2} \frac{y_{sol}^2}{2} = 0, 0 = 0 \right]$$

Kita mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusi $a = b = c = d/3$.

```
>solve(eqns, [a, b][[a = 0, b = %r1]])$Ada juga metode Lagrange, yang memaksimalkan H(a,b,c)^2 sehubungan dengan a+b+c=d.
```

```
>&solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la, ...
> diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la])
```

```
Maxima said:
diff: second argument must be a variable; found [1,0,4]
-- an error. To debug this try: debugmode(true);
```

```
Error in:
... la,      diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la]) ...
^
```

ita bisa membuat plot situasi

Pertama-tama, tetapkan titik-titik di Maxima.

```
>A &= at([x,y],sol[2]); $A
```

```
Maxima said:
part: invalid index of list or matrix.
-- an error. To debug this try: debugmode(true);
```

```
Error in:
A &= at([x,y],sol[2]); $A ...
^
```

```
>B &= [0,0]; $B, C &= [a,0]; $C
```

[a, 0]

[a, 0]

Kemudian, tetapkan kisaran plot, dan plot titik-titiknya.

```
>setPlotRange(0,5,-2,3); ...
>a=4; b=3; c=2; ...
>plotPoint(mxmeval("B"),"B"); plotPoint(mxmeval("C"),"C"); ...
>plotPoint(mxmeval("A"),"A");
```

```
Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
  return evaluate(mxm(s));
Error in:
... otPoint(mxmeval("C"),"C"); plotPoint(mxmeval("A"),"A"): ...
^
```

Plot the segments.

```
>plotSegment(mxmeval("A"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("A"));
```

```
Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
  return evaluate(mxm(s));
Error in:
plotSegment(mxmeval("A"),mxmeval("C")); plotSegment(mxmeval("B ...
^
```

Hitung garis tegak lurus tengah dalam Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan bagian tengah lingkaran.

```
>U &= lineIntersection(h,g);
```

Kita mendapatkan rumus untuk jari-jari lingkaran.

```
>&assume(a>0,b>0,c>0); distance(U,B)|radcan  $\frac{\sqrt{a_2^2+a_1^2}\sqrt{a_2^2+a_1^2-2aa_1+a^2}}{2|a_2|}$ 
Mari kita tambahkan ini ke dalam plot.
```

```
>plotPoint(U()); ...
>plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
```

```
Variable a2 not found!
Use global variables or parameters for string evaluation.
Error in ^
Error in expression: [a/2,(a2^2+a1^2-a*a1)/(2*a2)]
Error in:
plotPoint(U()); plotCircle(circleWithCenter(mxmeval("U"),mxmev ...
^
```

Dengan menggunakan geometri, kami memperoleh rumus sederhana

$$\frac{a}{\sin(\alpha)} = 2r$$

untuk jari-jari. Kita bisa mengeceknya dengan Maxima, apakah ini benar dengan Maxima. Maxima akan memfaktorkan ini hanya jika kita mengkuadratkannya.

```
>$c^2/sin(computeAngle(A,B,C))2 | factor
```

$$\left[\frac{a_2^2 + a_1^2}{a_2^2}, 0, \frac{16(a_2^2 + a_1^2)}{a_2^2} \right]$$

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari segitiga apa pun yang tidak sama sisi. Garis ini merupakan garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk ortosentrum, circumcentrum, centroid, titik Exeter, dan pusat lingkaran sembilan titik segitiga.

Sebagai demonstrasi, kami menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kita mendefinisikan sudut-sudut segitiga dalam Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolis.

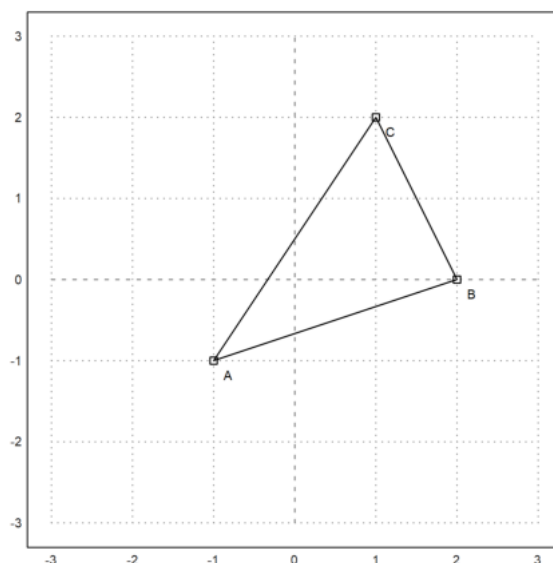
```
>A::=[-1,-1]; B::=[2,0]; C::=[1,2];
```

Untuk memplot objek geometris, kita menyiapkan area plot, dan menambahkan titik-titik. Semua plot objek geometris ditambahkan ke plot saat ini.

```
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```

Kita juga bisa menambahkan sisi-sisi segitiga.

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,"");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-043.png

Berikut ini adalah luas area segitiga, dengan menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

```
>areaTriangle(A,B,C)-  $\frac{7}{2}$ 
Kita dapat menghitung koefisien dari sisi c.
```

```
>c &= lineThrough(A,B)
```

[- 1, 3, - 2]

Dan juga mendapatkan formula untuk baris ini.

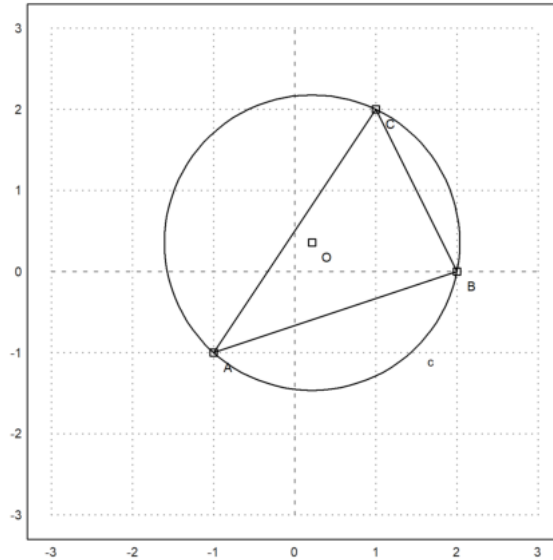
>getLineEquation(c, x, y) $3y - x = -2$ Untuk bentuk Hesse, kita perlu menentukan sebuah titik, sehingga titik tersebut berada di sisi positif dari bentuk Hesse. Memasukkan titik tersebut akan menghasilkan jarak positif ke garis.

>\$getHesseForm(c,x,y,C), at($\frac{7}{\sqrt{10}}$)

Sekarang kita menghitung keliling ABC.

>LL &= circleThrough(A,B,C); getCircleEquation(LL, x, y)
 $(y - \frac{5}{14})^2 + (x - \frac{3}{14})^2 = \frac{325}{98}$ >O &= getCircleCenter(LL); O[$\frac{3}{14}, \frac{5}{14}$] \$Plot lingkaran dan pusatnya. Cu dan U adalah simbolik. Kami mengevaluasi ekspresi ini untuk Euler.

>plotCircle(LL()); plotPoint(O, "O"):



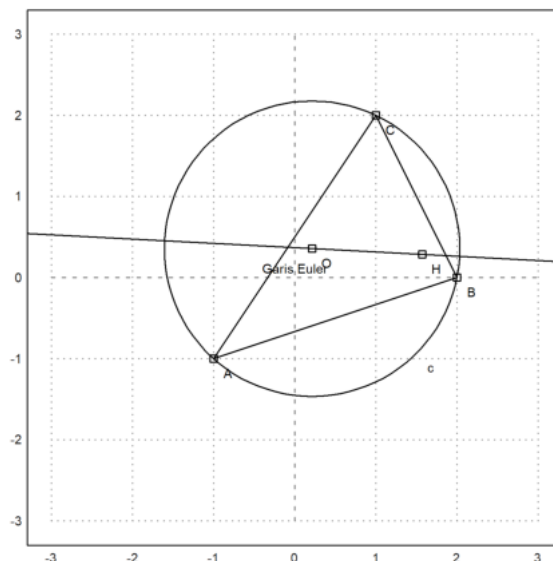
images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-050.png

Kita dapat menghitung perpotongan ketinggian di ABC (pusat ortosentrum) secara numerik dengan perintah berikut ini.

>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...
 > perpendicular(B,lineThrough(A,C))); H[$\frac{11}{7}, \frac{2}{7}$] \$Sekarang kita dapat menghitung garis Euler dari segitiga tersebut.

>el &= lineThrough(H,O); getLineEquation(el, x, y) $-\frac{19x}{14} - \frac{y}{14} = -\frac{1}{2}$ \$Tambahkan ke plot kami.

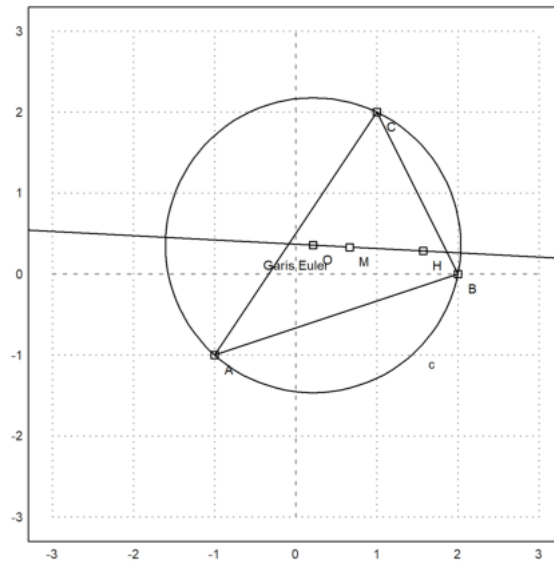
>plotPoint(H, "H"); plotLine(el, "Garis Euler"):



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-053.png

Pusat gravitasi harus berada pada garis ini.

```
>M &= (A+B+C)/3; getLineEquation(el, x, y)with[x = M[1], y = M[2]]- 1/2 = - 1/2
$>plotPoint(M(),"M"); // titik berat
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-055.png

Teori mengatakan bahwa $MH = 2 \cdot MO$. Kita perlu menyederhanakan dengan radcan untuk mencapai hal ini.

```
>distance(M, H) / distance(M, O)|radcan2$Fungsi-fungsi ini juga mencakup fungsi untuk sudut.
```

```
>$computeAngle(A,C,B), degprint(%)
```

$$\arccos\left(\frac{4}{\sqrt{5}\sqrt{13}}\right)$$

60°15'18.43''

Persamaan untuk bagian tengah lingkaran tidak terlalu bagus.

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; Q
```

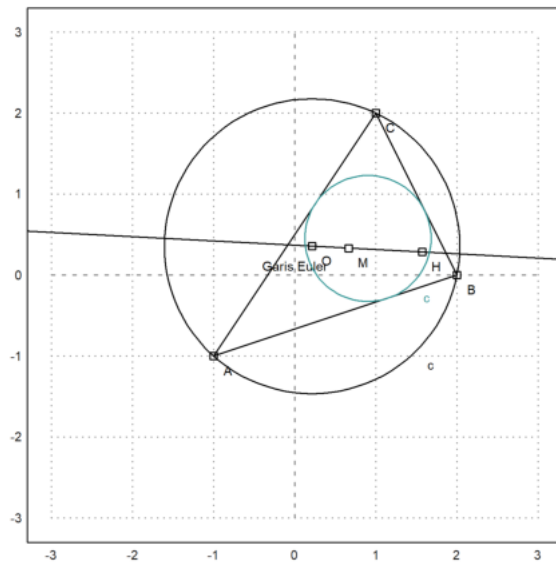
```
[ ( 2 3/2+1) sqrt(5) sqrt(13)-15 sqrt(2)+3, (sqrt(2)-3) sqrt(5) sqrt(13)+5 2 3/2+5 ]
[ 14, 14 ] $Mari kita hitung juga ekspresi untuk
jari-jari lingkaran yang tertulis.
```

```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; r
```

```
sqrt((-41 sqrt(2)-31) sqrt(5) sqrt(13)+115 sqrt(2)+614)
7 sqrt(2) $>LD &= circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke dalam plot.

```
>color(5); plotCircle(LD());
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-060.png

Parabola

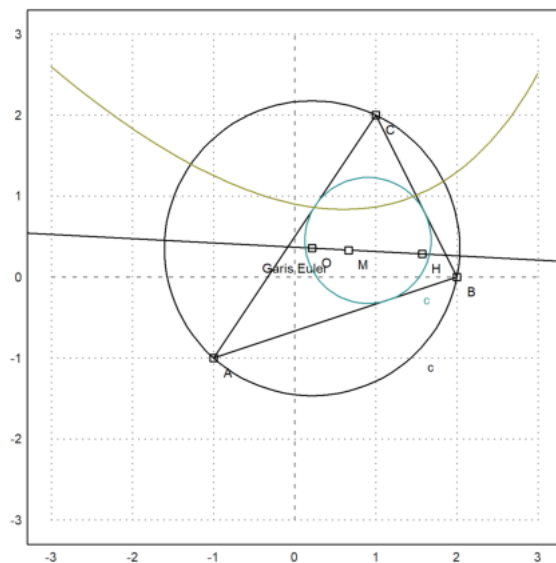
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3y - x + 2}{\sqrt{10}} - \sqrt{(2-y)^2 + (1-x)^2} = 0$$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-062.png

Ini seharusnya merupakan suatu fungsi, tetapi solver default Maxima hanya bisa menemukan solusinya jika kita mengkuadratkan persamaannya. Akibatnya, kita mendapatkan solusi palsu.

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

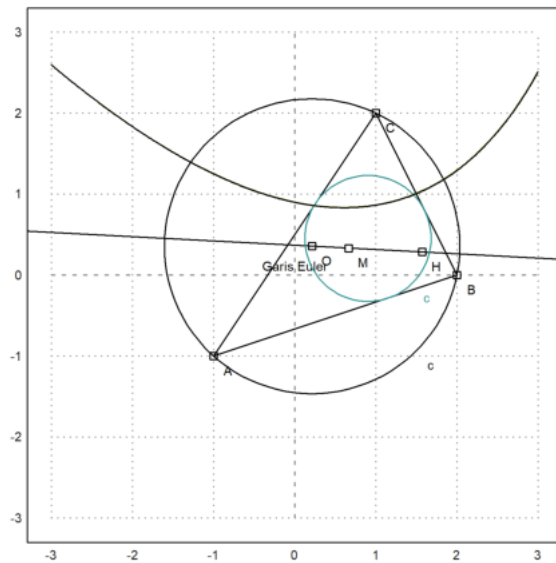
$$\begin{aligned} [y &= -3x - \sqrt{70} \sqrt{9 - 2x} + 26, \\ y &= -3x + \sqrt{70} \sqrt{9 - 2x} + 26] \end{aligned}$$

Solusi pertama adalah

```
maxima: akar[1]
```

Menambahkan solusi pertama ke dalam plot menunjukkan, bahwa ini memang jalur yang kita cari. Teori mengatakan bahwa ini adalah sebuah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-063.png

```
>function g(x) &= rhs(akar[1]);
'g(x) = g(x) // fungsi yang mendefinisikan kurva diatas
g(x) = -3x - sqrt(70) sqrt(9 - 2x) + 26$>T &= [-1, g(-1)]; // ambil sebarang titik pada kurva
tersebut

>dTC &= distance(T,C); $fullratsimp(dTC), float(2.135605779339061$
2.135605779339061

>U &= projectToLine(T,lineThrough(A,B)); U // proyeksi T pada garis AB
[ 80-3 sqrt(11) sqrt(70), 20-sqrt(11) sqrt(70) ]$>dU2AB &= distance(T,U); $fullratsimp(dU2AB), float(
2.135605779339061$2.135605779339061
```

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

Contoh 5: Trigonometri Rasional

Ini terinspirasi dari sebuah ceramah N.J. Wildberger. Dalam bukunya “Proporsi Ilahi”, Wildberger mengusulkan untuk mengganti gagasan klasik tentang jarak dan sudut dengan kuadransi dan penyebaran. Dengan menggunakan ini, memang memungkinkan untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap “rasional”.

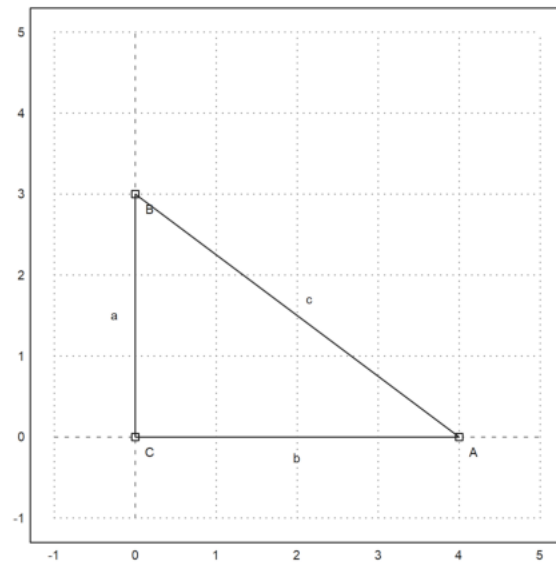
Berikut ini, saya akan memperkenalkan konsep-konsep tersebut, dan memecahkan beberapa masalah. Saya menggunakan komputasi simbolik Maxima di sini, yang menyembunyikan keuntungan utama dari trigonometri rasional yaitu komputasi dapat dilakukan dengan kertas dan pensil saja. Anda dipersilakan untuk memeriksa hasilnya tanpa komputer.

intinya adalah bahwa komputasi rasional simbolik sering kali memberikan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang dievaluasi dengan pendekatan numerik saja.

```
>load geometry;
```

Untuk pengenalan pertama, kita menggunakan segitiga persegi panjang dengan proporsi Mesir yang terkenal 3, 4, dan 5. Perintah berikut ini adalah perintah Euler untuk memplot geometri bidang yang terdapat pada file Euler “geometry.e”.

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,“A”); plotPoint(B,“B”); plotPoint(C,“C”); ...
>plotSegment(B,A,“c”); plotSegment(A,C,“b”); plotSegment(C,B,“a”); ...
>insimg(30);
```



Tentu saja,

di mana α adalah sudut di A. Cara biasa untuk menghitung sudut ini, adalah dengan mengambil kebalikan dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak kira-kira.

```
>wa := arcsin(3/5); degprint(wa)
```

$36^{\circ}52'11.63''$

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadrat, yang menggantikan jarak. Sebenarnya, ini hanyalah jarak yang dikuadratkan. Berikut ini, a , b , dan c menunjukkan kuadrat sisi-sisinya.

Teorema Pythagoras menjadi $a + b = c$.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

$25 = 25$

Gagasan kedua dari trigonometri rasional adalah penyebaran. Penyebaran mengukur bukaan di antara garis-garis. Ini adalah 0, jika garis-garisnya sejajar, dan 1, jika garis-garisnya persegi panjang. Ini adalah kuadrat dari sinus sudut antara dua garis.

Penyebaran garis AB dan AC pada gambar di atas didefinisikan sebagai

di mana a dan c adalah kuadrat dari segitiga persegi panjang dengan satu sudut di A.

```
>sa &= a/c; sa  $\frac{9}{25}$ 
```

Tentu saja, hal ini lebih mudah dihitung daripada sudut. Tetapi Anda kehilangan sifat bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita bisa mengonversi nilai perkiraan kita untuk sudut α ke sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

$9/25$

Hukum kosinus trigonometri klasik diterjemahkan ke dalam “hukum silang” berikut ini.

di sini a , b , dan c adalah kuadrat dari sisi-sisi segitiga, dan sa adalah penyebaran di sudut A. Sisi a , seperti biasa, berlawanan dengan sudut A.

Hukum-hukum ini diimplementasikan dalam file `geometri.e` yang kita masukkan ke dalam Euler.

```
>crosslaw(aa, bb, cc, saa)
```

$$\left[\left(bb - aa + \frac{7}{6} \right)^2, \left(bb - aa + \frac{7}{6} \right)^2, \left(bb - aa + \frac{5}{3\sqrt{2}} \right)^2 \right] = \left[\frac{14bb(1-saa)}{3}, \frac{14bb(1-saa)}{3}, \frac{52\frac{3}{2}bb(1-saa)}{3} \right]$$

\$Dalam kasus kami, kami mendapatkan

```
>crosslaw(a,b,c,sa)1024 = 1024
```

Mari kita gunakan `crosslaw` ini untuk mencari sebaran di A. Untuk melakukannya, kita buat `crosslaw` untuk kuadrat a , b , dan c , dan selesaikan untuk sebaran sa yang tidak diketahui.

Anda bisa melakukan ini dengan tangan dengan mudah, tapi saya menggunakan Maxima. Tentu saja, kita mendapatkan hasil yang sudah kita dapatkan.

```
>$crosslaw(a,b,c,x), solve([x =  $\frac{9}{25}$ ], [x =  $\frac{9}{25}$ ])
```

Kita sudah mengetahui hal ini. Definisi penyebaran adalah kasus khusus dari crosslaw.

Kita juga dapat menyelesaikannya untuk a, b, c secara umum. Hasilnya adalah sebuah rumus yang menghitung penyebaran sudut sebuah segitiga dengan kuadran ketiga sisinya.

```
>solve(crosslaw(aa, bb, cc, x), x)
```

$$\left[\left[\frac{168bbx + 36bb^2 + (-72aa - 84)bb + 36aa^2 - 84aa + 49}{36}, \frac{168bbx + 36bb^2 + (-72aa - 84)bb + 36aa^2 - 84aa + 49}{36}, \frac{152\sqrt{2}bbx + 18bb^2 + (-36aa - 152\sqrt{2})bb + 18aa^2 - 1}{18} \right] \right]$$

\$Kita dapat membuat sebuah fungsi dari hasil tersebut. Fungsi seperti itu sudah didefinisikan dalam file geometry.e dari Euler.

```
>spread(a, b, c)  $\frac{9}{25}$  Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi
```

Hasilnya adalah rasional, yang tidak mudah didapat jika kita menggunakan trigonometri klasik.

```
>spread(a, a, 4\ *a/7)  $\frac{6}{7}$  Ini adalah sudut dalam derajat.
```

```
>degprint(arcsin(sqrt(6/7)))
```

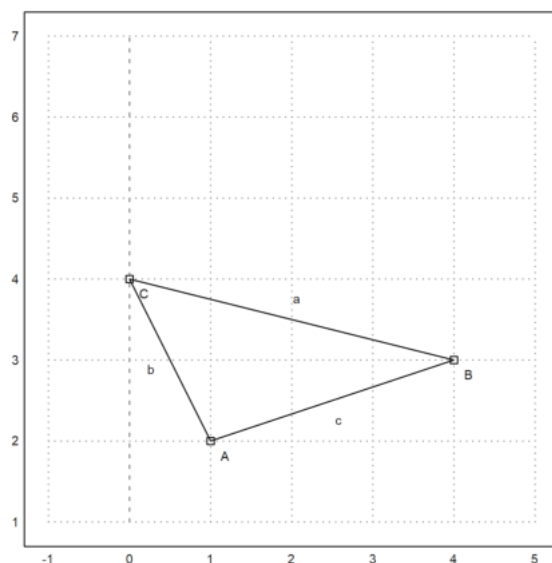
67°47'32.44''

Contoh Yang Lain

Sekarang, mari kita coba contoh yang lebih lanjut.

Kami menetapkan tiga sudut segitiga sebagai berikut.

```
>A&:=[1,2]; B&:=[4,3]; C&:=[0,4]; ...
> setPlotRange(-1,5,1,7); ...
> plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
> plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
> insimg;
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-079.png

Dengan menggunakan Pythagoras, mudah untuk menghitung jarak antara dua titik. Pertama-tama saya menggunakan jarak fungsi dari file Euler untuk geometri. Jarak fungsi menggunakan geometri klasik.

```
>$distance(A,B)
```

$$\sqrt{10}$$

Euler juga memiliki fungsi untuk kuadratan antara dua titik.

Pada contoh berikut, karena c+b bukan a, maka segitiga tersebut tidak berbentuk persegi panjang.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); a,17$5
```

17
images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-083.png

Pertama, mari kita menghitung sudut tradisional. Fungsi computeAngle menggunakan metode yang biasa berdasarkan hasil kali titik dari dua vektor. Hasilnya adalah beberapa perkiraan titik mengambang.

>wb &= computeAngle(A,B,C); \$wb, (wb/ pi\ *180)()arccos ($\frac{11}{\sqrt{10}\sqrt{17}}$) \$
32.4711922908

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kita masukkan kuadran a, b, dan c ke dalam hukum silang dan selesaikan untuk x.

>\$crosslaw(a,b,c,x), solve([x = $\frac{49}{50}$])\$ $x = \frac{49}{50}$

Itulah yang dilakukan oleh fungsi spread yang didefinisikan dalam “geometry.e”.

>sb &= spread(b,a,c); sb $\frac{49}{170}$ \$Maxima mendapatkan hasil yang sama dengan menggunakan trigonometri biasa, jika kita memaksakannya. Ia menyelesaikan suku sin(arccos(...)) menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

>sin(computeAngle(A, B, C)) $\frac{49}{170}$ \$Setelah kita memiliki penyebaran di B, kita dapat menghitung tinggi ha di sisi a. Ingatlah bahwa

menurut definisi.

>ha &= c*sb; ha $\frac{49}{17}$ \$Gambar berikut ini dibuat dengan program geometri C.a.R., yang dapat menggambar kuadran dan penyebaran.

image: (20) Rational_Geometry_CaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadrannya.

>sqr(ha) $\frac{7}{\sqrt{17}}$ \$Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita berurusan dengan kuadran!

>sqr(ha) \ *sqr(a)/ 2 $\frac{7}{2}$ \$Rumus penentu yang biasa menghasilkan hasil yang sama.

>areaTriangle(B, A, C) $\frac{7}{2}$ \$ ## The Heron Formula Sekarang, mari kita selesaikan masalah ini secara umum!

>&remvalue(a,b,c,sb,ha);

Pertama-tama kita menghitung penyebaran di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat (“quadrea”?), memfaktorkannya dengan Maxima, dan kita mendapatkan rumus Heron yang terkenal.

Memang, hal ini sulit dilakukan dengan pensil dan kertas.

>\$spread(b²,c²,a²), f actor ($\frac{(-c+b+a)(c-b+a)(c+b-a)(c+b+a)}{(-c+b+a)(c-b+a)(c+b-a)(c+b+a)^{16}}$) \$
 $\frac{16}{16}$

The Triple Spread Rule

Kerugian dari spread adalah bahwa mereka tidak lagi hanya menambahkan sudut seperti.

Namun, tiga spread dari sebuah segitiga memenuhi aturan “triple spread” berikut ini.

>&remvalue(sa,sb,sc); triplespread(sa, sb, sc)
(sc + sb + sa)² = 2 (sc² + sb² + sa²) + 4 sa sb sc\$Aturan ini berlaku untuk tiga sudut yang berjumlah 180°.

Karena spread dari adalah sama, aturan triple spread juga benar, jika

Karena penyebaran sudut negatif adalah sama, aturan triple spread juga berlaku, jika

Sebagai contoh, kita dapat menghitung penyebaran sudut 60°. Ini adalah 3/4. Namun, persamaan ini memiliki solusi kedua, di mana semua penyebarannya adalah 0.

>solve(triplespread(x, x, x), x)[x = $\frac{3}{4}$, x = 0]\$Penyebaran 90° jelas adalah 1. Jika dua sudut ditambahkan ke 90°, penyebarannya akan menyelesaikan persamaan penyebaran tiga dengan a, b, 1. Dengan perhitungan berikut, kita mendapatkan a + b = 1.

>\$triplespread(x,y,1), solve([x = 1 - y])\$ $x = 1 - y$

Karena penyebaran 180°-t sama dengan penyebaran t, rumus penyebaran tiga kali lipat juga berlaku, jika satu sudut adalah jumlah atau selisih dari dua sudut lainnya.

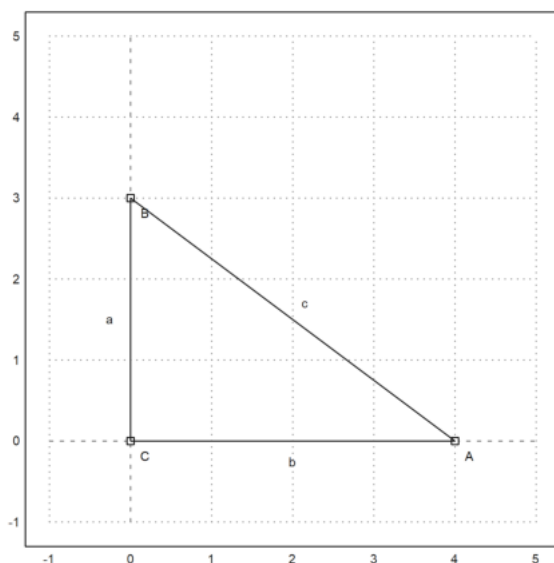
Jadi kita dapat menemukan penyebaran sudut dua kali lipat. Perhatikan bahwa ada dua solusi lagi. Kita jadikan ini sebuah fungsi.

>Misplaced &[x = 4a - 4a², x = 0]\$ - 4 (a - 1) a

Angle Bisectors

Ini adalah situasi yang sudah kita ketahui.

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
> setPlotRange(-1,5,-1,5); ...
> plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
> plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
> insimg;
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-100.png

Mari kita hitung panjang garis bagi sudut di A. Tetapi kita ingin menyelesaikannya untuk a, b, c secara umum.

```
>&remvalue(a,b,c);
```

Jadi, pertama-tama kita menghitung penyebaran sudut yang dibelah dua di A, menggunakan rumus penyebaran tiga.

Masalah dengan rumus ini muncul lagi. Rumus ini memiliki dua solusi. Kita harus memilih salah satu yang benar. Solusi lainnya mengacu pada sudut terbagi dua 180° -wa.

```
>$stripespread(x,x,a/(a+b)), $solve(%,x), sa2 &= rhs( %[1]); sa2  $-\frac{\sqrt{b}\sqrt{b+a+b+a}}{2b+2a}$ 
```

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

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

images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-103.png

Mari kita periksa persegi panjang Mesir.

```
>sa2with[a = 3^2, b = 4^2]  $\frac{1}{10}$  $Kita bisa mencetak sudut dalam Euler, setelah mentransfer penyebaran ke radian.
```

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

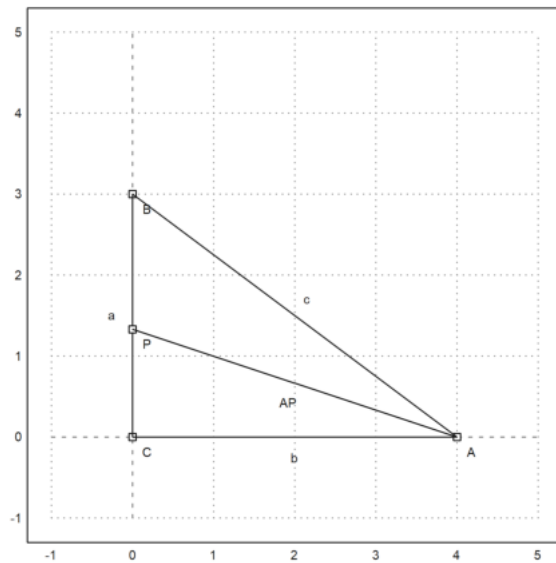
```
18°26'5.82''
```

Titik P adalah perpotongan garis bagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
```

```
[0, 1.33333]
```

```
>plotPoint(P,"P"); plotSegment(A,P):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-105.png

Mari kita periksa sudut-sudutnya dalam contoh spesifik kita.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

0.321750554397

0.321750554397

Sekarang kita hitung panjang garis bagi AP.

Kita menggunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

berlaku untuk semua segitiga. Kuadratkan, ini diterjemahkan ke dalam apa yang disebut “hukum penyebaran”

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

di mana a, b, c menunjukkan qudrance.

Karena spread CPA adalah $1-sa^2$, kita mendapatkan $bisa/1 = b/(1-sa^2)$ dan bisa menghitung bisa (kuadransi dari garis-bagi sudut).

>factor(ratsimp(b/(1-sa^2))); bisa &= %; bisa $\frac{2b(b+a)}{\sqrt{b}\sqrt{b+a+b+a}}$ \$Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])"), distance(A,P)
```

4.21637021356

4.21637021356

Kita juga dapat menghitung P dengan menggunakan rumus penyebaran.

>py&=factor(ratsimp(sa^2*bisa)); py- $\frac{b(\sqrt{b}\sqrt{b+a}-b-a)}{\sqrt{b}\sqrt{b+a+b+a}}$ \$Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

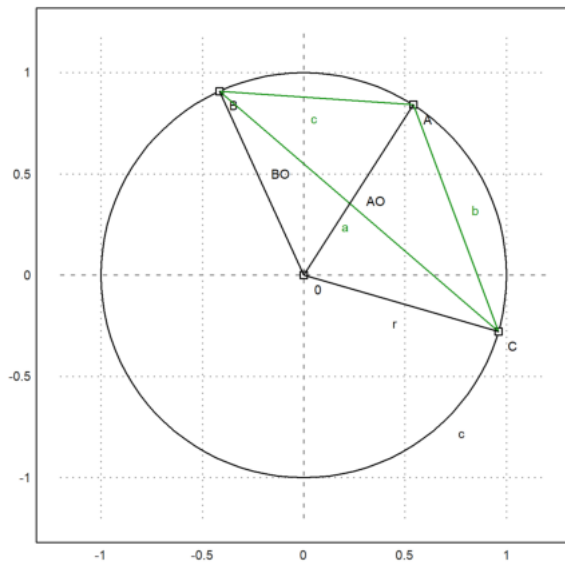
```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

1.33333333333

The Chord Angle

Lihatlah situasi berikut ini.

```
>setPlotRange(1.2); ...
>color(1); plotCircle(circleWithCenter([0,0],1)); ...
>A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>color(1); O:=[0,0]; plotPoint(O,"O"); ...
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
>insimg;
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-110.png

Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran tiga untuk sudut-sudut di pusat O untuk r. Dengan demikian kita mendapatkan rumus untuk jari-jari kuadrat dari pericircle dalam hal kuadran sisi-sisinya.

Kali ini, Maxima menghasilkan beberapa angka nol yang rumit, yang kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
```

```
>rabc := rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); rabc
- (a b c) / (c^2 - 2 b c + a (-2 c - 2 b) + b^2 + a^2)
```

\$Kita dapat menjadikannya sebuah fungsi Euler.

```
>function periradius(a,b,c) &:= rabc;
```

Mari kita periksa hasilnya untuk poin A, B, C.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Radiusnya memang 1.

```
>periradius(a,b,c)
```

1

Faktanya adalah, bahwa penyebaran CBA hanya bergantung pada b dan c. Ini adalah teorema sudut akar.

```
>spread(b, a, c) * rabc | ratsimp b/4
```

Faktanya, penyebarannya adalah $b/(4r)$, dan kita melihat bahwa sudut chord b adalah setengah dari sudut tengah.

```
>doublespread(b/(4 * r)) - spread(b, r, r) | ratsimp 0
```

\$Contoh 6: Jarak Minimal pada Bidang

Keterangan awal

Fungsi yang, pada sebuah titik M pada bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis-garis tingkat yang cukup sederhana: lingkaran yang berpusat di A.

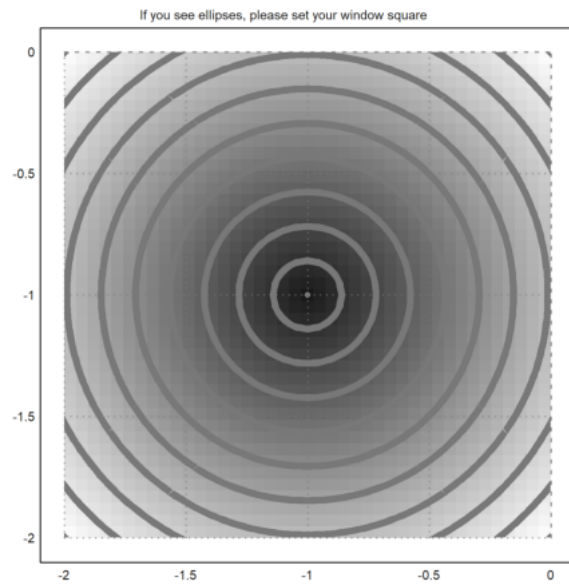
```
>&remvalue();
```

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

```
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)
```

```
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...
```

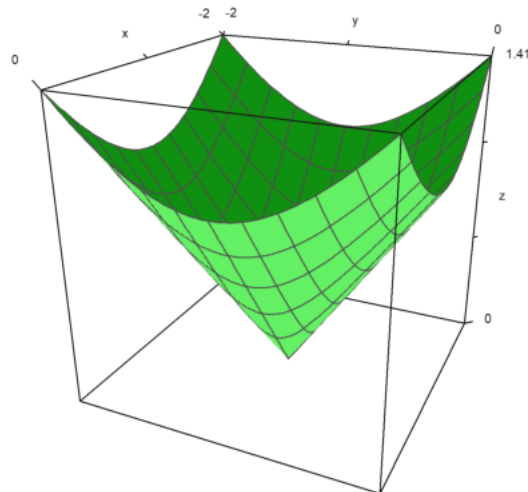
```
>title="If you see ellipses, please set your window square");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-114.png

dan grafiknya juga cukup sederhana: bagian atas kerucut:

>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):



Tentu saja nilai minimum 0 diperoleh dalam A.

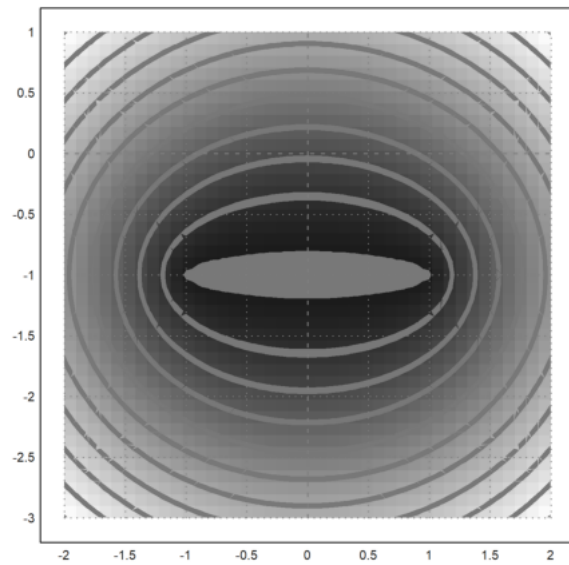
Dua titik

Sekarang kita lihat fungsi $MA+MB$ di mana A dan B adalah dua titik (tetap). Ini adalah "fakta yang terkenal" bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali AB minimum yang konstan pada segmen $[AB]$:

>B=[1,-1];

>function d2(x,y):=d1(x,y)+sqrt((x-B[1])²+(y-B[2])²)

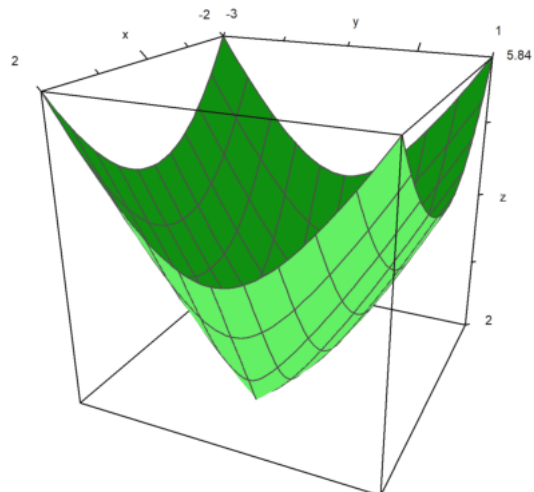
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-116.png

Grafiknya lebih menarik:

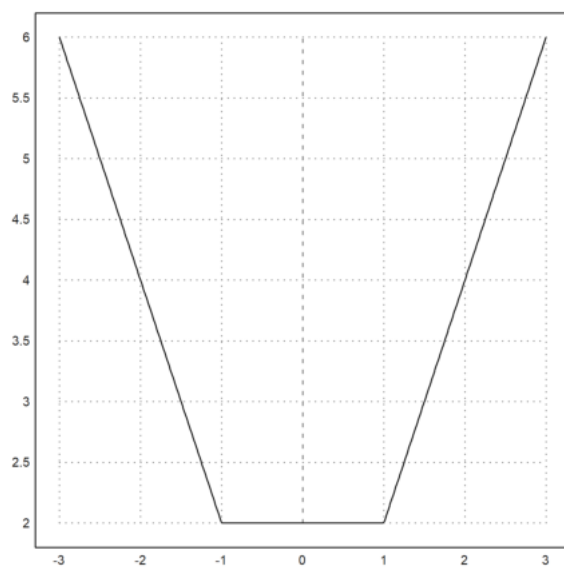
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-117.png

Pembatasan pada garis (AB) lebih terkenal:

>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):



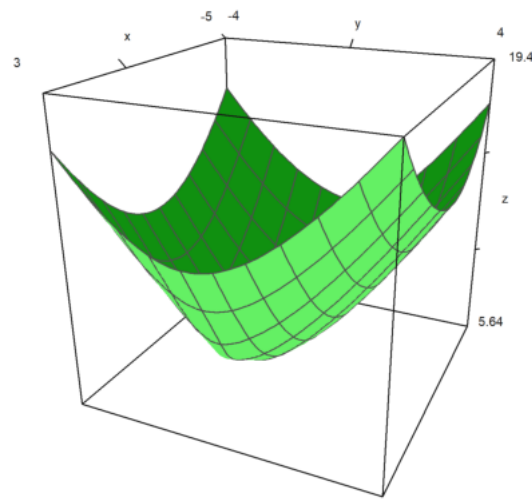
Tiga poin

Sekarang, hal-hal menjadi kurang sederhana: Hal ini sedikit kurang dikenal bahwa $MA+MB+MC$ mencapai minimumnya pada satu titik di bidang, tetapi untuk menentukannya tidak sesederhana itu:

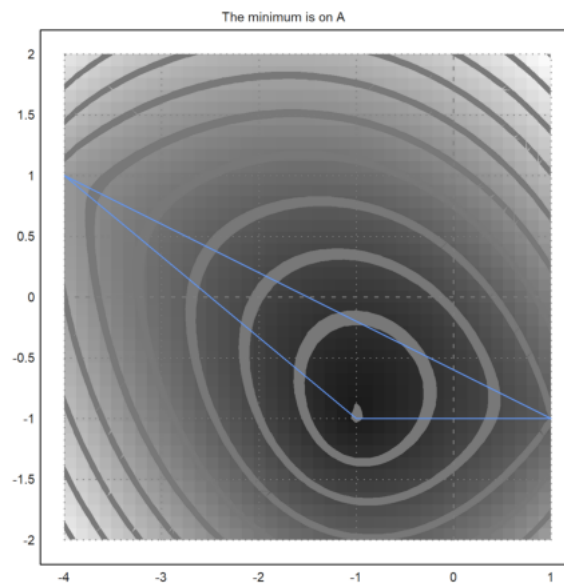
1. Jika salah satu sudut segitiga ABC lebih dari 120° (katakanlah di A), maka minimum dicapai pada titik ini (katakanlah AB+AC).

Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
```

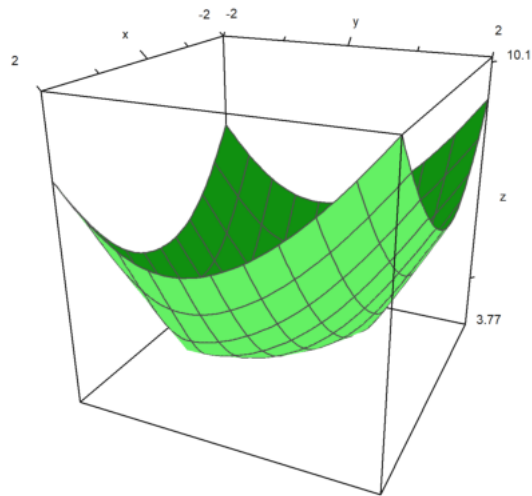


```
>insimg;
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```



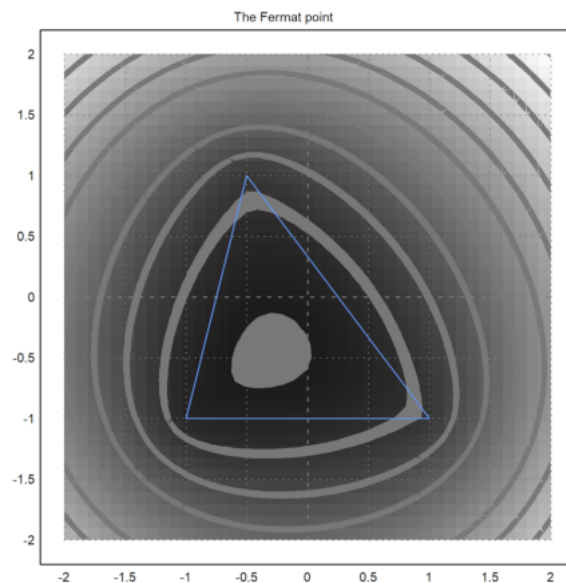
2. Tetapi jika semua sudut segitiga ABC kurang dari 120° , minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (masing-masing 120°):

```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-121.png

```
>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-122.png

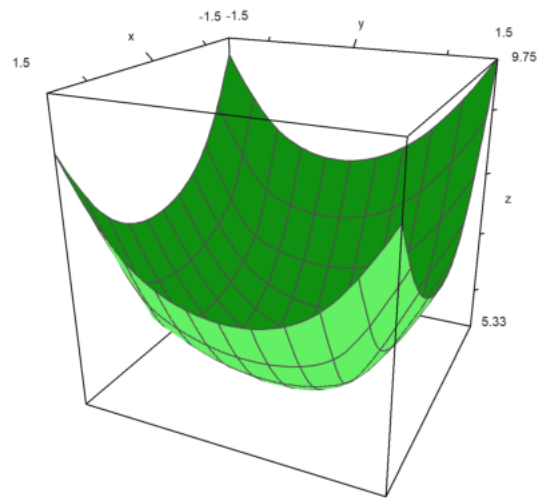
Merupakan kegiatan yang menarik untuk merealisasikan gambar di atas dengan perangkat lunak geometri; sebagai contoh, saya tahu sebuah perangkat lunak yang ditulis dalam bahasa Java yang memiliki instruksi “garis kontur”...

Semua hal di atas telah ditemukan oleh seorang hakim Perancis bernama Pierre de Fermat; dia menulis surat kepada para ahli lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di bagian pajak penghasilan. Jadi titik unik F sedemikian rupa sehingga $FA+FB+FC$ minimal, disebut titik Fermat dari segitiga. Namun tampaknya beberapa tahun sebelumnya, Torricelli dari Italia telah menemukan titik ini sebelum Fermat menemukannya! Bagaimanapun juga, tradisinya adalah mencatat titik F ini...

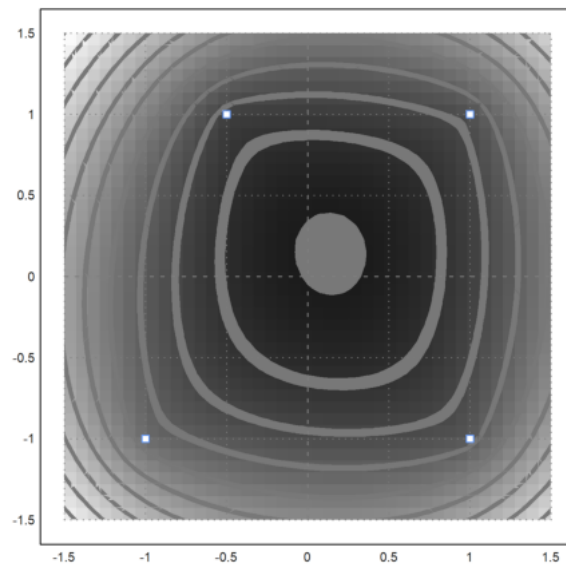
Empat poin

Langkah selanjutnya adalah menambahkan titik ke-4 D dan mencoba meminimumkan $MA+MB+MC+MD$; misalkan Anda adalah operator TV kabel dan ingin menemukan di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])2+(y-D[2])2);
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
```



```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
>insimg;
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-124.png

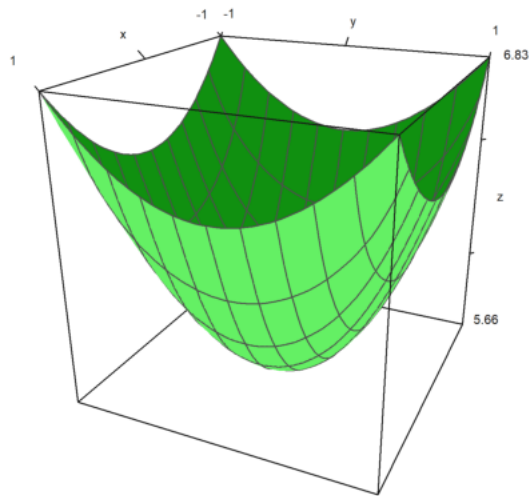
Masih ada nilai minimum dan tidak ada simpul A, B, C, maupun D:

```
>function f(x):=d4(x[1],x[2])
>neldermin("f",[0.2,0.2])
[0.142858, 0.142857]
```

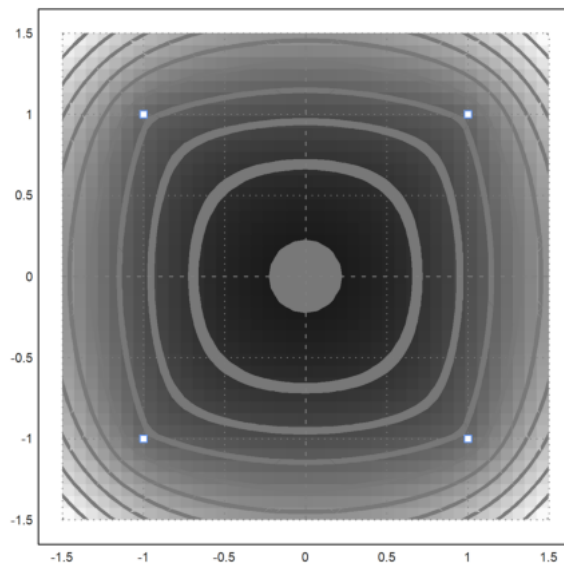
Tampaknya dalam kasus ini, koordinat titik optimal adalah rasional atau mendekati rasional...

Karena ABCD adalah sebuah bujur sangkar, maka kita berharap bahwa titik optimalnya adalah pusat dari ABCD:

```
>C=[-1,1];
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1)
```



```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);
>insimg;
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-126.png

Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan pvenigne.exe pada path program.

Pertama, kita menghitung jari-jari bola.

Jika Anda melihat gambar di bawah ini, Anda dapat melihat bahwa kita membutuhkan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kita menggunakan file geometri.e dari Euler untuk hal ini.

```
>load geometry;
```

Pertama, dua garis yang membentuk kerucut.

```
>g1 &= lineThrough([0,0],[1,a])
```

```
[- a, 1, 0]
```

```
>g2 &= lineThrough([0,0],[-1,a])
```

```
[- a, - 1, 0]
```

Kemudianm baris ketiga

```
>g &= lineThrough([-1,0],[1,1])
```

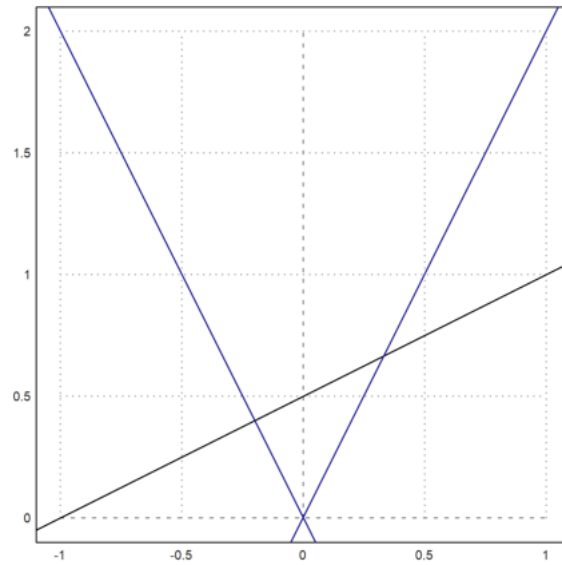
[- 1, 2, 1]

Kami memplot semuanya sejauh ini.

```
>setPlotRange(-1,1,0,2);
```

```
>color(black); plotLine(g(),"")
```

```
>a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```



Sekarang, kita

ambil titik umum pada sumbu y.

```
>P &= [0,u]
```

[0, u]

Hitung jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); d1*sqrt((a^2*u/(a^2+1) - u)^2 + (a^2*u^2/(a^2+1)^2))
```

Hitung jarak ke g.

```
>d &= distance(P,projectToLine(P,g)); d*sqrt((u/5 - u)^2 + ((2*u-1)^2/25))
```

Dan temukan pusat kedua lingkaran, yang jaraknya sama.

```
>sol &= solve(d1^2=d2,u); sol[u = (-sqrt(5)*sqrt(a^2+1)+2*a^2+2)/(4*a^2-1), u = (sqrt(5)*sqrt(a^2+1)+2*a^2+2)/(4*a^2-1)]
```

Ada dua solusi.

Kami mengevaluasi solusi simbolis, dan menemukan kedua pusat, dan kedua jarak.

```
>u := sol()
```

```
[0.333333, 1]
```

```
>dd := d()
```

```
[0.149071, 0.447214]
```

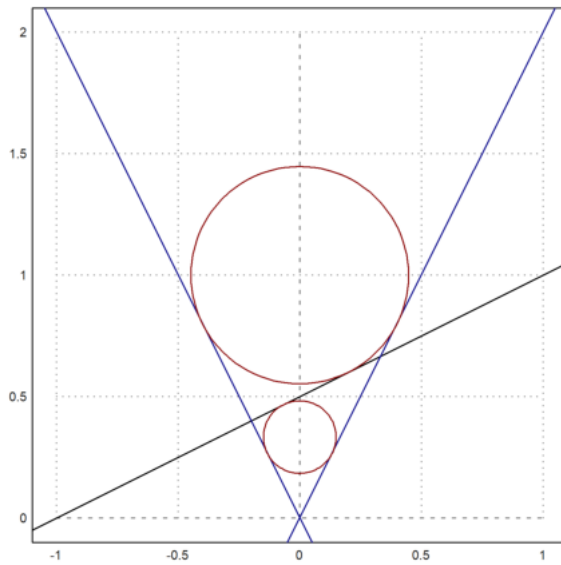
Plot lingkaran ke dalam gambar.

```
>color(red);
```

```
>plotCircle(circleWithCenter([0,u[1]],dd[1]),"");
```

```
>plotCircle(circleWithCenter([0,u[2]],dd[2]),"");
```

```
>insimg;
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-131.png

Plot dengan Povray

Selanjutnya kita plot semuanya dengan Povray. Perhatikan bahwa Anda mengubah perintah apapun pada urutan perintah Povray berikut ini, dan jalankan kembali semua perintah dengan Shift-Return.

Pertama kita memuat fungsi povray.

```
>load povray;
```

```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Kami menyiapkan pemandangan dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Selanjutnya kita tulis kedua bola tersebut ke file Povray.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
```

```
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

Dan kerucutnya, transparan

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

Kami menghasilkan bidang yang terbatas pada kerucut.

```
>gp=g();
```

```
>pc=povcone([0,0,0],0,[0,0,a],1,"");
```

```
>vp=[gp[1],0,gp[2]]; dp=gp[3];
```

```
>writeln(povplane(vp,dp,povlook(blue,0.5),pc));
```

Sekarang kita menghasilkan dua titik pada lingkaran, di mana bola menyentuh kerucut.

```
>function turnz(v) := return
```

```
>P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
```

```
>writeln(povpoint(P1,povlook(yellow)));
```

```
>P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
```

```
>writeln(povpoint(P2,povlook(yellow)));
```

Kemudian, kita menghasilkan dua titik di mana bola-bola tersebut menyentuh bidang. Ini adalah fokus elips.

```
>P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
```

```
>writeln(povpoint(P3,povlook(yellow)));
```

```
>P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
```

```
>writeln(povpoint(P4,povlook(yellow)));
```

Selanjutnya kita menghitung perpotongan P1P2 dengan bidang.

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
```

```
>writeln(povpoint(P5,povlook(yellow)));
```

Kami menghubungkan titik-titik dengan segmen garis.

```
>writeln(povsegment(P1,P2,povlook(yellow)));
```

```
>writeln(povsegment(P5,P3,povlook(yellow)));
```

```
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Sekarang, kita menghasilkan pita abu-abu, di mana bola-bola menyentuh kerucut.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);
```

```
>pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
```

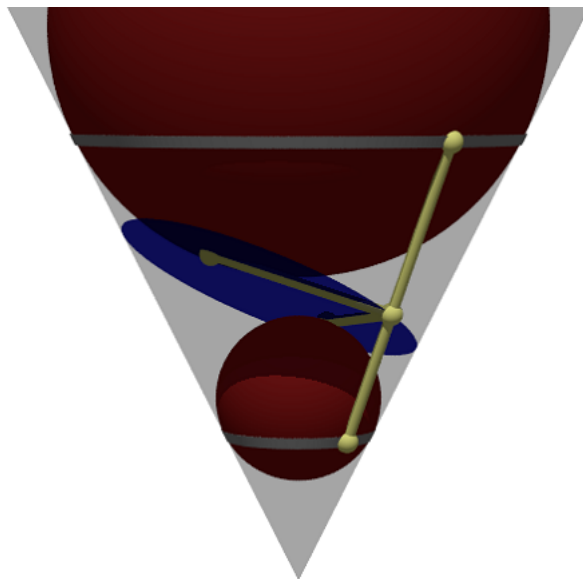
```
>writeln(povintersection([pcw,pc1],povlook(gray)));
```

```
>pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
```

```
>writeln(povintersection([pcw,pc2],povlook(gray)));
```

Mulai program Povray.

```
>povend();
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-132.png

Untuk mendapatkan Anaglyph ini, kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali nanti.

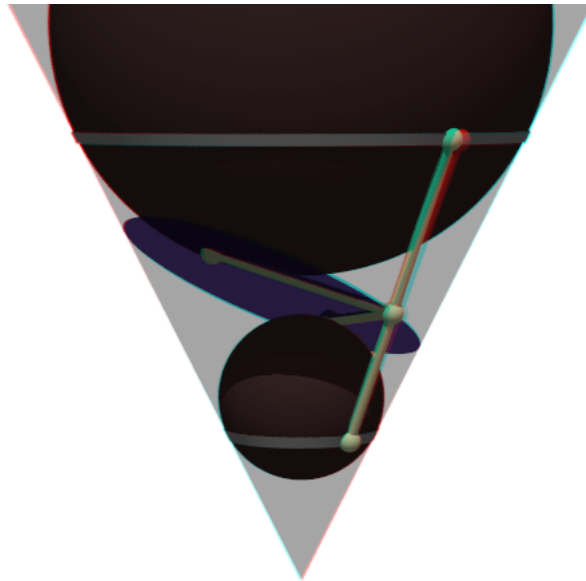
```
>function scene () ...
```

```
global a,u,dd,g,g1,defaultpointsize;
writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
gp=g();
pc=povcone([0,0,0],0,[0,0,a],1,"");
vp=[gp[1],0,gp[2]]; dp=gp[3];
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow)));
writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow)));
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],
[0,0,P1[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc1],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],
[0,0,P2[3]+defaultpointsize/2],1);
```

```
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction
```

Anda memerlukan kacamata merah/sian untuk mengapresiasi efek berikut ini.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-133.png

Contoh 8: Geometri Bumi

Pada buku catatan ini, kita ingin melakukan beberapa komputasi bola. Fungsi-fungsi tersebut terdapat pada file “spherical.e” pada folder contoh. Kita perlu memuat file tersebut terlebih dahulu.

```
>load “spherical.e”;
```

Untuk memasukkan posisi geografis, kita menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut ini adalah koordinat untuk Kampus FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569, 1.92657]
```

Anda dapat mencetak posisi ini dengan sposprint (cetak posisi bola).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
S 7°46.467' E 110°23.050'
```

Mari kita tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,24.533)];
```

```
>sposprint(Solo), sposprint(Semarang),
```

```
S 7°34.333' E 110°49.683'
```

```
S 6°59.050' E 110°24.533'
```

Pertama, kita menghitung vektor dari satu titik ke titik lainnya pada bola ideal. Vektor ini adalah [heading, jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang 7°.

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan jarak FMIPA-Solo
```

```
65°20'26.60''
```

```
53.8945384608
```

Ini adalah perkiraan yang baik. Rutinitas berikut ini menggunakan perkiraan yang lebih baik lagi. Pada jarak yang pendek, hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->” km” // perkiraan jarak FMIPA-Semarang
```

Commands must be separated by semicolon or comma!

Found: // perkiraan jarak FMIPA-Semarang (character 32)

You can disable this in the Options menu.

Error in:

```
esdist(FMIPA,Semarang)->” km” // perkiraan jarak FMIPA-Semarang ...
```

^

Ada fungsi untuk judul, dengan mempertimbangkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

65.34°

Sudut segitiga melebihi 180° pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,
Semarang,Solo); degprint(asum)
```

180°0'10.77''

Ini bisa digunakan untuk menghitung luas area segitiga. Catatan: Untuk segitiga kecil, cara ini tidak akurat karena kesalahan pengurangan dalam asum-pi.

```
>(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FMIPA-Solo-Semarang
```

Commands must be separated by semicolon or comma!

Found: // perkiraan luas segitiga FMIPA-Solo-Semarang (character 32)

You can disable this in the Options menu.

Error in:

```
(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FM ...
^
```

Ada sebuah fungsi untuk hal ini, yang menggunakan garis lintang rata-rata segitiga untuk menghitung radius bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi esarea()
```

2123.64310526 km^2

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi arah dan jarak, keduanya dalam radian. Untuk mendapatkan sebuah vektor, kita menggunakan svector. Untuk menambahkan sebuah vektor ke sebuah posisi, kita menggunakan saddvector.

```
>v=svector(FMIPA,Solo); sposprint(saddvector(FMIPA,v)), sposprint(Solo),
```

S 7°34.333' E 110°49.683'

S 7°34.333' E 110°49.683'

Fungsi-fungsi ini mengasumsikan bola yang ideal. Hal yang sama di bumi.

```
>sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(Solo),
```

S 7°34.333' E 110°49.683'

S 7°34.333' E 110°49.683'

Mari kita beralih ke contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk menemukan koordinatnya).

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];
```

```
>sposprint(Tugu), sposprint(Monas)
```

S 7°46.998' E 110°21.966'

S 6°10.500' E 106°48.717'

Menurut Google Earth, jaraknya adalah 429,66 km. Kami mendapatkan perkiraan yang bagus.

```
>esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Monas Jakarta
```

Commands must be separated by semicolon or comma!

Found: // perkiraan jarak Tugu Jogja - Monas Jakarta (character 32)

You can disable this in the Options menu.

Error in:

```
esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Mona ...
^
```

Judulnya sama dengan yang dihitung di Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

294°17'2.85''

Namun demikian, kita tidak lagi mendapatkan posisi target yang tepat, jika kita menambahkan arah dan jarak ke posisi semula. Hal ini terjadi, karena kita tidak menghitung fungsi inversi secara tepat, tetapi mengambil perkiraan radius bumi di sepanjang jalur.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

S 6°10.500' E 106°48.717'

Namun demikian, kesalahannya tidak besar.

```
>sposprint(Monas),
```

S 6°10.500' E 106°48.717'

Tentu saja, kita tidak bisa berlayar dengan arah yang sama dari satu tujuan ke tujuan lainnya, jika kita ingin mengambil jalur terpendek. Bayangkan, Anda terbang ke arah NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti arah yang konstan!

Perhitungan berikut ini menunjukkan bahwa kita akan melenceng dari tujuan yang benar, jika kita menggunakan arah yang sama selama perjalanan.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh dari jarak tersebut, dengan menggunakan arah menuju Monas, kita akan sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya jauh berbeda.

```
>sposprint(p), skmprint(esdist(p,Monas))
```

```
S 6°11.250' E 106°48.372'
1.529km
```

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran lintang 30°, tetapi jalur yang lebih pendek yang dimulai 10° lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

```
79.69°
```

Namun, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi, kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kita sesuaikan pada 1/10 dari jarak total.

```
>p=P1; dist=esdist(P1,P2); ...
> loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
79.69°
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 judul yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

```
0.203km
```

Kita akan mendapatkan perkiraan yang baik, jika kita menyesuaikan arah setiap 1/100 dari total jarak dari Tugu ke Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...
> loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;
```

```
>skmprint(esdist(p,Monas))
```

```
0.000km
```

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS di sepanjang Bundaran Hotel Indonesia menuju Monas dengan fungsi navigate.

```
>load spherical; v=navigate(Tugu,Monas,10); ...
> loop 1 to rows(v); sposprint(v[#]), end;
```

```
S 7°46.998' E 110°21.966'
S 7°37.422' E 110°0.573'
S 7°27.829' E 109°39.196'
S 7°18.219' E 109°17.834'
S 7°8.592' E 108°56.488'
S 6°58.948' E 108°35.157'
S 6°49.289' E 108°13.841'
S 6°39.614' E 107°52.539'
S 6°29.924' E 107°31.251'
S 6°20.219' E 107°9.977'
S 6°10.500' E 106°48.717'
```

Kami menulis sebuah fungsi, yang memplot bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
```

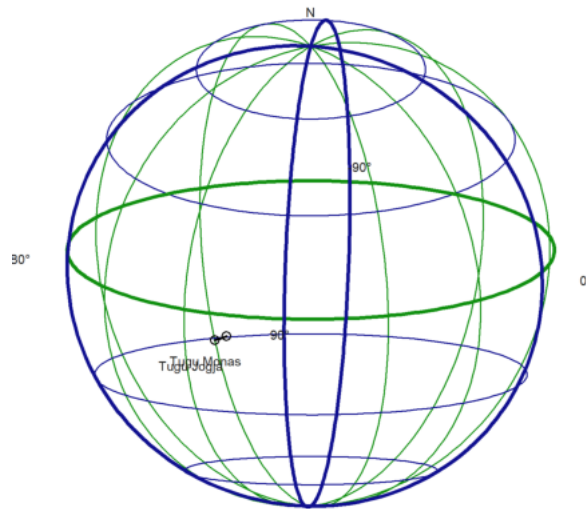
```

useglobal;
plotearth;
plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");
plotposline(v);
endfunction

```

Sekarang plot semuanya.

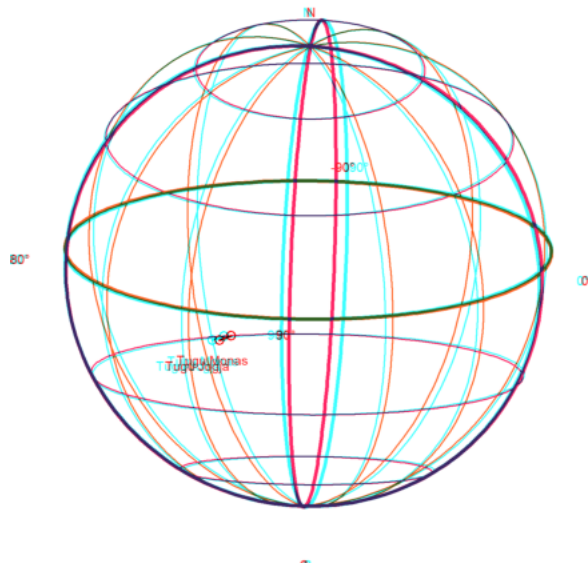
```
>plot3d("testplot",angle=25,height=6,>own,>user,zoom=4):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-134.png

Atau gunakan plot3d untuk mendapatkan tampilan anaglyph. Ini terlihat sangat bagus dengan kacamata merah/cyan.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```



Latihan

1. Gambarkan segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah $(360/n)$.
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan $(360/n)$.
- Untuk n ganjil, pilih salah satu titik sudut adalah di atas.
- Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat.
- Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

2. Gambarkan suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:

- Misalkan persamaan parabolanya $y = ax^2 + bx + c$.

- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
 - Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.
3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.
 - Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung (sisinya-sisinya merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).
 - Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya bertemu di satu titik.
 - Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya.
 - Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.
 4. Gambarlah suatu elips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat elips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).
 5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat elips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

Jawaban

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

Petunjuk: * Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah $(360/n)$.
 * Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan $(360/n)$. * Untuk n ganjil, pilih salah satu titik sudut adalah di atas. * Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat. * Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

```
>reset; >load geometry
```

```
Numerical and symbolic geometry.
```

```
>setPlotRange(-5,5,-5,5);
```

```
>A=[-2,-2]; plotPoint(A,"A");
```

```
>B=[2,-2]; plotPoint(B,"B");
```

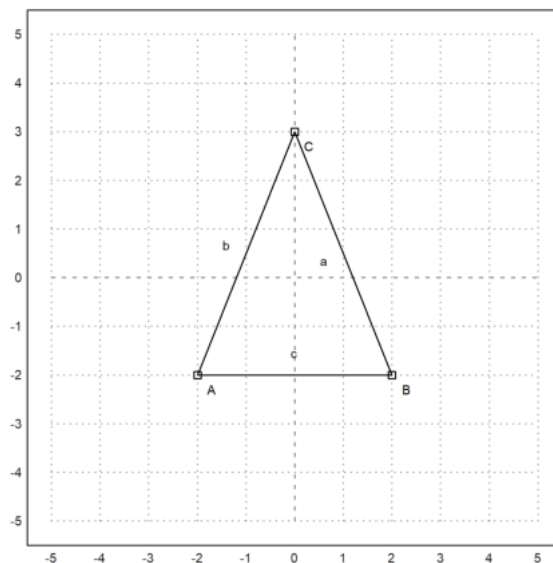
```
>C=[0,3]; plotPoint(C,"C");
```

```
>plotSegment(A,B,"c");
```

```
>plotSegment(B,C,"a");
```

```
>plotSegment(A,C,"b");
```

```
>aspect(1);
```



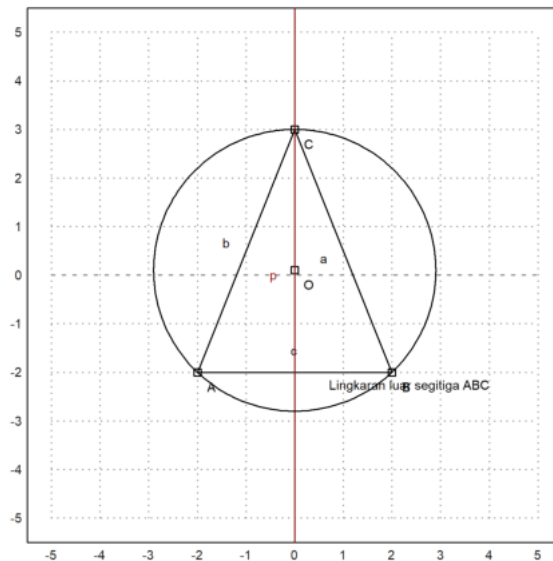
images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-136.png

```
>c=circleThrough(A,B,C);
```

```

>R=getCircleRadius(c);
>O=getCircleCenter(c);
>plotPoint(O,"O");
>l=angleBisector(A,C,B);
>color(2); plotLine(l); color(1);
>plotCircle(c,"Lingkaran luar segitiga ABC");

```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-137.png

2. Gambarkanlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:

- Misalkan persamaan parabolanya $y = ax^2 + bx + c$.
- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a , b , c .

```

>load geometry

```

Numerical and symbolic geometry.

```

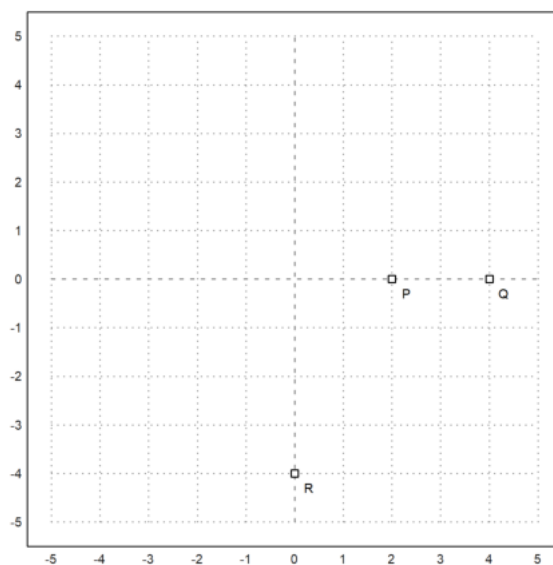
>setPlotRange(5); P=[2,0]; Q=[4,0]; R=[0,-4];

```

```

>plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R");

```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-138.png

```

>sol &= solve([a+b=-c, 16*a+4*b=-c,c=-4],[a,b,c])

```

```

[[a = - 1, b = 5, c = - 4]]

```

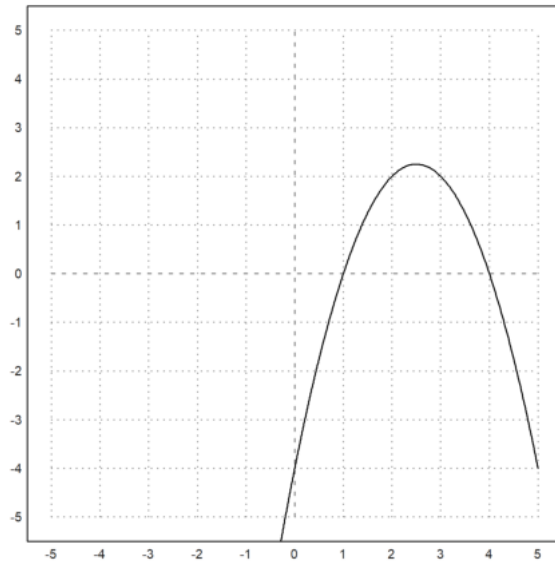
```

>function y&=-x^2+5*x-4

```

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

```
>plot2d("-x^2+5*x-4",-5,5,-5,5):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-139.png

3. Gambarkan suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.

- o Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung (sisinya-sisinya merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).
- o Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya bertemu di satu titik.
- o Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya.
- o Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

```
>load geometry
```

Numerical and symbolic geometry.

```
>setPlotRange(-4.5,4.5,-4.5,4.5);
```

```
>A=[-3,-3]; plotPoint(A,"A");
```

```
>B=[3,-3]; plotPoint(B,"B");
```

```
>C=[3,3]; plotPoint(C,"C");
```

```
>D=[-3,3]; plotPoint(D,"D");
```

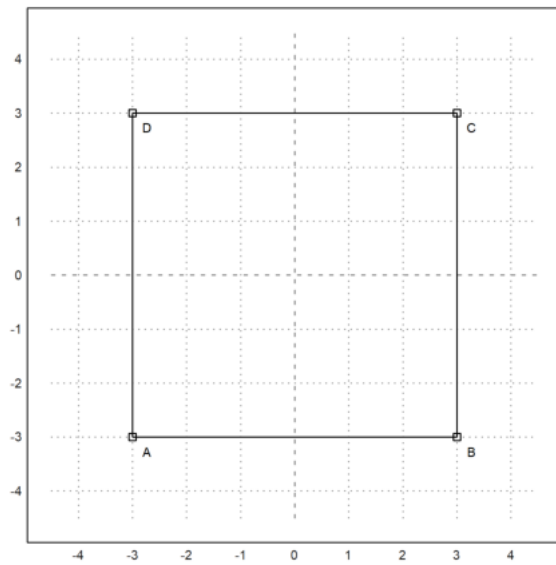
```
>plotSegment(A,B,"");
```

```
>plotSegment(B,C,"");
```

```
>plotSegment(C,D,"");
```

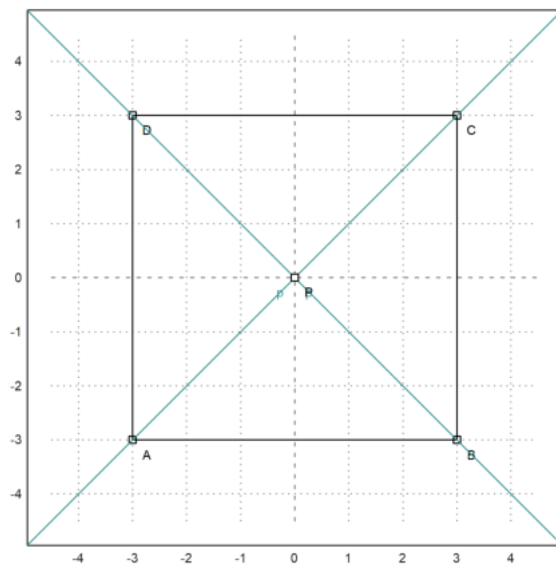
```
>plotSegment(A,D,"");
```

```
>aspect(1):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-140.png

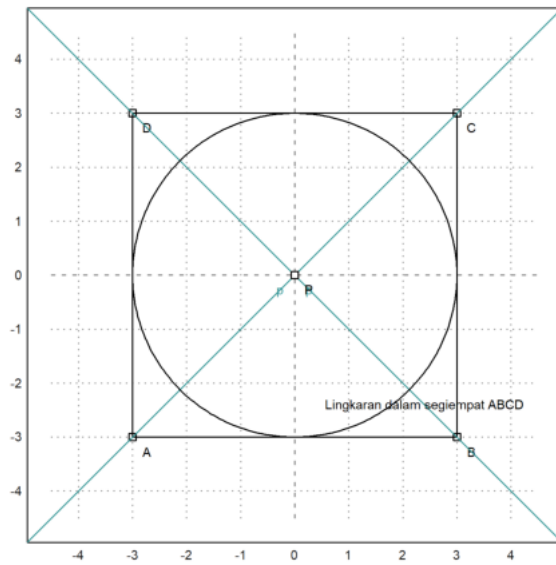
```
>l=angleBisector(A,B,C);
>m=angleBisector(B,C,D);
>P=lineIntersection(l,m);
>color(5); plotLine(l); plotLine(m); color(1);
>plotPoint(P,"P");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-141.png

Dari gambar diatas terlihat bahwa keempat garis bagi sudutnya bertemu di satu titik yaitu titik P.

```
>r=norm(P-projectToLine(P,lineThrough(A,B)));
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat ABCD");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-142.png

Dari gambar diatas, terlihat bahwa sisi-sisinya merupakan garis singgung lingkaran yang sama yaitu lingkaran

dalam segiempat.

Akan ditunjukkan bahwa hasil kali panjang sisi-sisi yang berhadapan sama.

>AB=norm(A-B) //panjang sisi AB

6

>CD=norm(C-D) //panjang sisi CD

6

>AD=norm(A-D) //panjang sisi AD

6

>BC=norm(B-C) //panjang sisi BC

6

>AB.CD

36

>AD.BC

36

Terbukti bahwa hasil kali panjang sisi-sisi yang berhadapan sama yaitu 36. Jadi dapat dipastikan bahwa

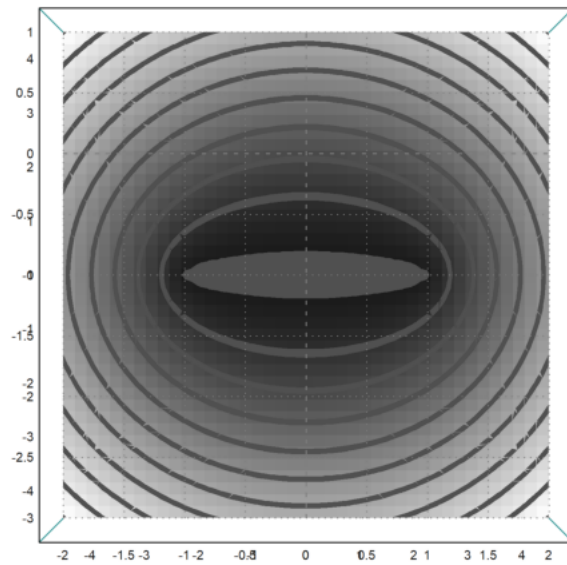
segiempat tersebut merupakan segiempat garis singgung.

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).

>P=[-1,-1]; Q=[1,-1];

>function d2(x,y):=sqrt((x-P[1])²+(y-P[2])²)+sqrt((x-Q[1])²+(y-Q[2])²)

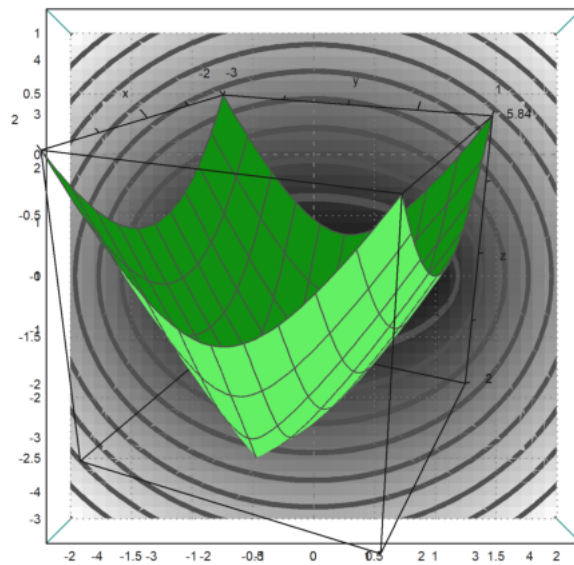
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):



Grafik yang lebih

menarik

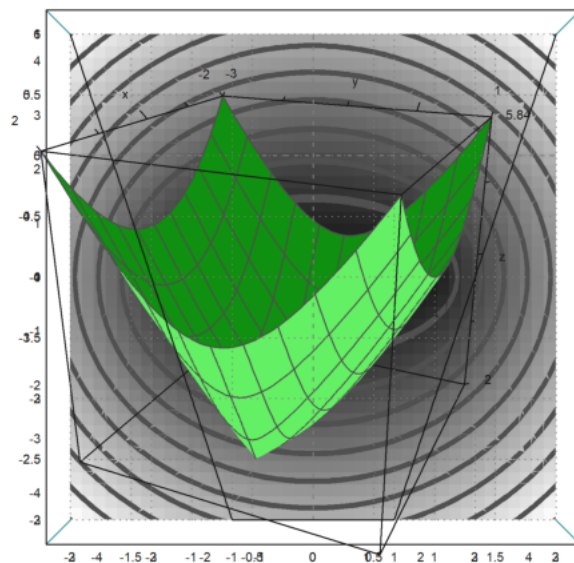
`>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):`



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-144.png

Batasan ke garis PQ

`>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):`



5. Gambarkan

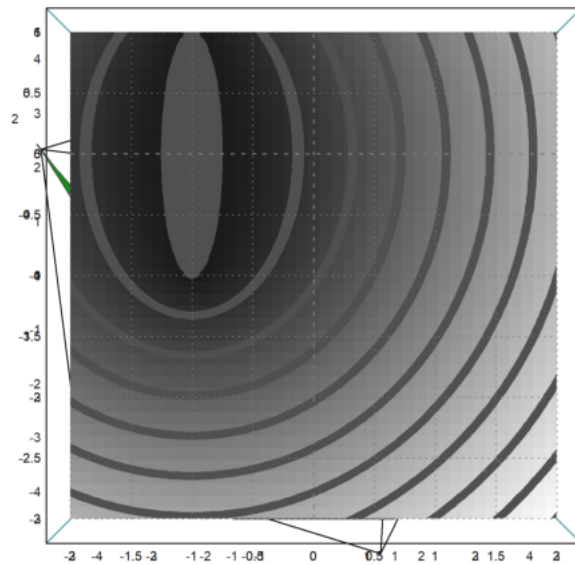
suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu

sama (konstan).

```
>P=[-1,-1]; Q=[1,-1];
```

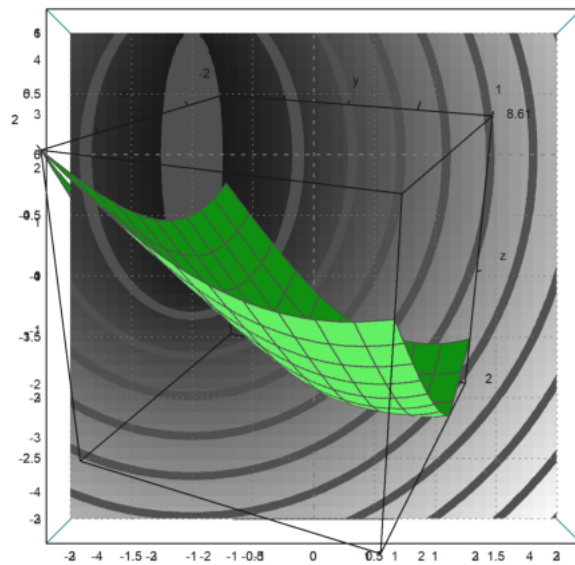
```
>function d2(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)+sqrt((x+Q[1])^2+(y+Q[2])^2)
```

```
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



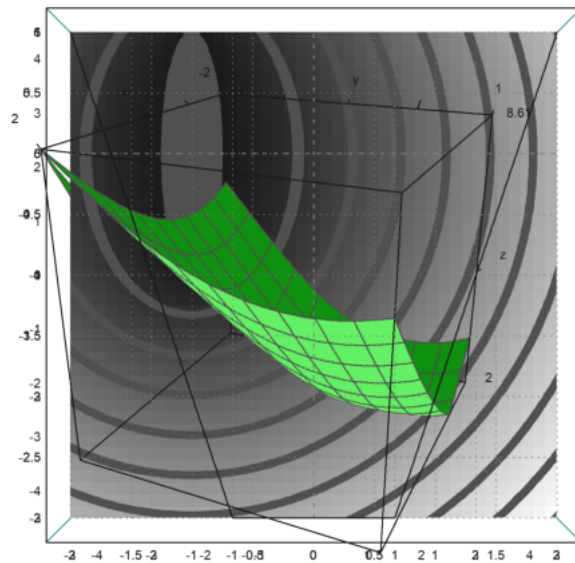
images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-146.png

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-147.png

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTGeometri-148.png

Nama : Alya Putri Medilasari Kelas : Matematika B 2023 NIM : 23030630018

EMT untuk Statistika

Dalam buku catatan ini, kami mendemonstrasikan plot statistik utama, tes dan distribusi dalam Euler.

Mari kita mulai dengan beberapa statistik deskriptif. Ini bukanlah sebuah pengantar statistik. Jadi, Anda mungkin memerlukan beberapa latar belakang untuk memahami detailnya.

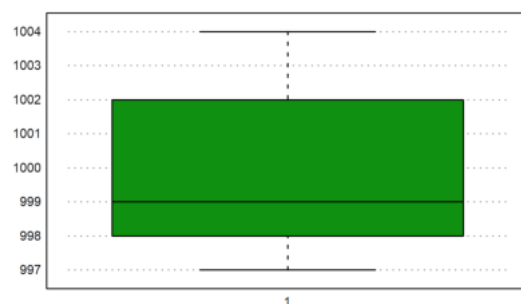
Asumsikan pengukuran berikut ini. Kita ingin menghitung nilai rata-rata dan deviasi standar yang diukur.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...
> median(M), mean(M), dev(M),
```

```
999
999.9
2.72641400622
```

We can plot the box-and-whiskers plot for the data. In our case there are no outliers.

```
>aspect(1.75); boxplot(M):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-001.png

Kami menghitung probabilitas bahwa suatu nilai lebih besar dari 1005, dengan mengasumsikan nilai yang diukur dari distribusi normal.

Semua fungsi untuk distribusi dalam Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

$$\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-m}{d}\right)^2} dt.$$

Kami mencetak hasilnya dalam % dengan akurasi 2 digit menggunakan fungsi cetak.

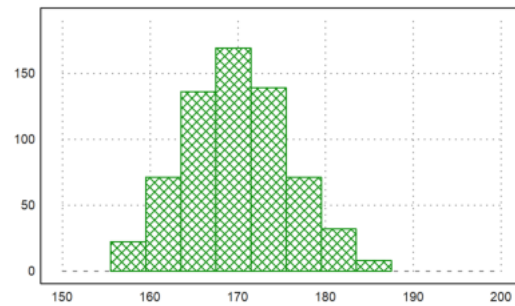
```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
0.00 %
```

Untuk contoh berikutnya, kami mengasumsikan jumlah pria berikut ini dalam rentang ukuran tertentu.

```
>r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Berikut ini adalah plot distribusinya.

```
>plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="v"):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-003.png

Kita dapat memasukkan data mentah tersebut ke dalam tabel.

Tabel adalah sebuah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal rentang, akhir rentang, jumlah orang dalam rentang.

Tabel dapat dicetak dengan header. Kami menggunakan vektor string untuk mengatur header.

```
>T:=r[1:8]' | r[2:9]' | v'; writetable(T,labc=["BB","BA","Frek"])
```

BB	BA	Frek
155.5	159.5	22
159.5	163.5	71
163.5	167.5	136
167.5	171.5	169
171.5	175.5	139
175.5	179.5	71
179.5	183.5	32
183.5	187.5	8

Jika kita membutuhkan nilai rata-rata dan statistik lain dari ukuran, kita perlu menghitung titik tengah rentang. Kita dapat menggunakan dua kolom pertama dari tabel kita untuk hal ini.

Sumbol “|” digunakan untuk memisahkan kolom, fungsi “writetable” digunakan untuk menulis tabel, dengan opsi “labc” untuk menentukan judul kolom.

```
>(T[,1]+T[,2])/2 // the midpoint of each interval
```

```
157.5
161.5
165.5
169.5
173.5
177.5
181.5
185.5
```

Tetapi akan lebih mudah, untuk melipat rentang dengan vektor [1/2,1/2].

```
>M=fold(r,[0.5,0.5])
```

```
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
```

Sekarang kita dapat menghitung rata-rata dan deviasi sampel dengan frekuensi yang diberikan.

```
>{m,d}=meandev(M,v); m, d,
```

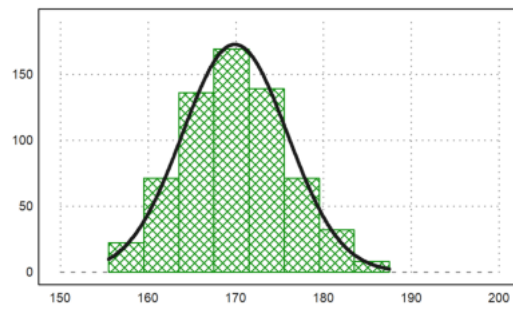
```
169.901234568
5.98912964449
```

Mari kita tambahkan distribusi normal dari nilai-nilai tersebut ke dalam diagram batang di atas. Rumus untuk distribusi normal dengan rata-rata m dan deviasi standar d adalah:

$$y = \frac{1}{d\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2d^2}}.$$

Karena nilainya antara 0 dan 1, untuk memplotnya pada diagram batang, nilai tersebut harus dikalikan dengan 4 kali jumlah data.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...
> xmin=min(r),xmax=max(r),thickness=3,add=1):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-005.png

Tabel

Dalam direktori buku catatan ini, Anda dapat menemukan file dengan tabel. Data tersebut mewakili hasil survei. Berikut adalah empat baris pertama dari file tersebut. Data berasal dari buku online berbahasa Jerman “Einführung in die Statistik mit R” oleh A. Handl.

```
>printfile("table.dat",4);
```

```
Person Sex Age Titanic Evaluation Tip Problem
1 m 30 n . 1.80 n
2 f 23 y g 1.80 n
3 f 26 y g 1.80 y
```

Tabel berisi 7 kolom angka atau token (string). Kita ingin membaca tabel tersebut dari file. Pertama, kita menggunakan terjemahan kita sendiri untuk token-token tersebut.

Untuk itu, kita mendefinisikan set token. Fungsi `strtokens()` mendapatkan vektor string token dari string yang diberikan.

```
>mf=["m","f"]; yn=["y","n"]; ev:=strtokens("g vg m b vb");
```

Sekarang kita membaca tabel dengan terjemahan ini.

Argumen `tok2`, `tok4`, dan lain-lain adalah terjemahan dari kolom-kolom tabel. Argumen-argumen ini tidak ada dalam daftar parameter `readtable()`, jadi Anda perlu memberikannya dengan “:=”.

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
>load over statistics;
```

Untuk mencetak, kita perlu menentukan set token yang sama. Kami mencetak empat baris pertama saja.

```
>writetable(MT[1:10],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n

Tanda titik “.” mewakili nilai yang tidak tersedia.

Jika kita tidak ingin menentukan token untuk terjemahan sebelumnya, kita hanya perlu menentukan kolom mana yang berisi token dan bukan angka.

```
>ctok=[2,4,5,7]; {MT,hd,tok}=readtable("table.dat",ctok=ctok);
```

fungsi `readtable()` sekarang mengembalikan satu set token.

```
>tok
```

```
m
n
f
y
g
vg
```

Tabel berisi entri dari file dengan token yang diterjemahkan ke angka.

String khusus `NA=""` ditafsirkan sebagai “Tidak Tersedia”, dan mendapatkan `NAN` (bukan angka) dalam tabel. Terjemahan ini dapat diubah dengan parameter `NA`, dan `NAval`.

```
>MT[1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Berikut ini adalah isi tabel dengan angka yang tidak diterjemahkan.

```
>writetable(MT,wc=5)
```

1	1	30	2	.	1.8	2
2	3	23	4	5	1.8	2
3	3	26	4	5	1.8	4
4	1	33	2	.	2.8	2
5	1	37	2	.	1.8	2
6	1	28	4	5	2.8	4
7	3	31	4	6	2.8	2
8	1	23	2	.	0.8	2
9	3	24	4	6	1.8	4
10	1	26	2	.	1.8	2
11	3	23	4	6	1.8	4
12	1	32	4	5	1.8	2
13	1	29	4	6	1.8	4
14	3	25	4	5	1.8	4
15	3	31	4	5	0.8	2
16	1	26	4	5	2.8	2
17	1	37	2	.	3.8	2
18	1	38	4	5	.	2
19	3	29	2	.	3.8	2
20	3	28	4	6	1.8	2
21	3	28	4	1	2.8	4
22	3	28	4	6	1.8	4
23	3	38	4	5	2.8	2
24	3	27	4	1	1.8	4
25	1	27	2	.	2.8	4

Untuk kenyamanan, Anda dapat menaruh output dari readtable() ke dalam sebuah daftar.

```
>Table={readtable("table.dat",ctok=ctok)};
```

Dengan menggunakan kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dll. atau menggunakan daftar Tabel.

```
>writetable(Table,ctok=ctok,wc=5);
```

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n
11	f	23	y	vg	1.8	y
12	m	32	y	g	1.8	n
13	m	29	y	vg	1.8	y
14	f	25	y	g	1.8	y
15	f	31	y	g	0.8	n
16	m	26	y	g	2.8	n
17	m	37	n	.	3.8	n
18	m	38	y	g	.	n
19	f	29	n	.	3.8	n
20	f	28	y	vg	1.8	n
21	f	28	y	m	2.8	y
22	f	28	y	vg	1.8	y
23	f	38	y	g	2.8	n
24	f	27	y	m	1.8	y
25	m	27	n	.	2.8	y

Fungsi tablecol() mengembalikan nilai kolom dari tabel, melewati setiap baris dengan nilai NAN ("." dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

```
>{c,i}=tablecol(MT,[5,6]);
```

Kita dapat menggunakan ini untuk mengekstrak kolom dari tabel untuk tabel baru.

```
>j=[1,5,6]; writetable(MT[i,j],lab=hd[j],ctok=[2],tok=tok)
```

Person	Evaluation	Tip
2	g	1.8
3	g	1.8
6	g	2.8
7	vg	2.8
9	vg	1.8
11	vg	1.8
12	g	1.8
13	vg	1.8
14	g	1.8

15	g	0.8
16	g	2.8
20	vg	1.8
21	m	2.8
22	vg	1.8
23	g	2.8
24	m	1.8

Tentu saja, kita perlu mengekstrak tabel itu sendiri dari daftar Tabel dalam kasus ini.

```
>MT=Table[1];
```

```
Table is not a variable!
Error in:
MT=Table[1]; ...
      ^
```

Tentu saja, kita juga dapat menggunakannya untuk menentukan nilai rata-rata kolom atau nilai statistik lainnya.

```
>mean(tablecol(MT,6))
```

```
Variable or function MT not found.
Error in:
mean(tablecol(MT,6)) ...
      ^
```

Fungsi `getstatistics()` mengembalikan elemen-elemen dalam sebuah vektor, dan jumlahnya. Kita menerapkannya pada nilai “m” dan “f” pada kolom kedua tabel kita.

```
>{xu,count}=getstatistics(tablecol(MT,2)); xu, count,
```

```
Variable or function MT not found.
Error in:
{xu,count}=getstatistics(tablecol(MT,2)); xu, count, ...
      ^
```

Kita bisa mencetak hasilnya dalam tabel baru.

```
>writetable(count',labr=tok[xu])
```

```
Variable count not found!
Error in:
writetable(count',labr=tok[xu]) ...
      ^
```

Fungsi `selecttable()` mengembalikan sebuah tabel baru dengan nilai dalam satu kolom yang dipilih dari vektor indeks. Pertama, kita mencari indeks dari dua nilai kita dalam tabel token.

```
>v:=indexof(tok,["g","vg"])
```

```
[0, 0]
```

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);
```

```
Variable or function MT not found.
Error in:
MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5); ...
      ^
```

Sekarang kita dapat mencetak tabel, dengan nilai yang diekstrak dan diurutkan di kolom ke-5.

```
>writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7);
```

```
MT1 is not a variable!
Error in:
writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7); ...
      ^
```

Untuk statistik berikutnya, kita ingin menghubungkan dua kolom tabel. Jadi kita mengekstrak kolom 2 dan 4 dan mengurutkan tabel.

```
>i:=sortedrows(MT,[2,4]); ...
> writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)
```

```
Variable or function MT not found.
Error in:
i:=sortedrows(MT,[2,4]); writetable(tablecol(MT[i],[2,4])',c ...
      ^
```

Dengan `getstatistics()`, kita juga dapat menghubungkan hitungan dalam dua kolom tabel satu sama lain.

```
>MT24=tablecol(MT,[2,4]); ...
> {xu1,xu2,count}=getstatistics(MT24[1],MT24[2]); ...
> writetable(count,labr=tok[xu1],labc=tok[xu2])

Variable or function MT not found.
Error in:
MT24=tablecol(MT,[2,4]); {xu1,xu2,count}=getstatistics(MT24[1] ...
^
```

Tabel dapat ditulis ke sebuah file.

```
>filename="test.dat"; ...
> writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);

Variable or function count not found.
Error in:
filename="test.dat"; writetable(count,labr=tok[xu1],labc=tok[x ...
^
```

Kemudian kita dapat membaca tabel dari file tersebut.

```
>{MT2,hd,tok2,hdr}=readtable(filename,>clabs,>rlabs); ...
> writetable(MT2,labr=hdr,labc=hd)

A BIndex 3 out of bounds!
Try "trace errors" to inspect local variables after errors.
writetable:
if separator==none then write(printstr(""+labc[j],wc{j}));
```

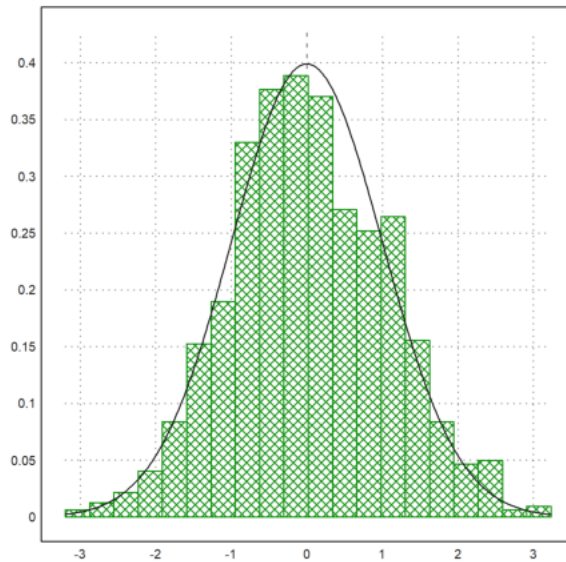
Dan hapus file nya

```
>fileremove(filename);
```

Distribusi

Dengan plot2d, ada metode yang sangat mudah untuk memplot distribusi data eksperimen.

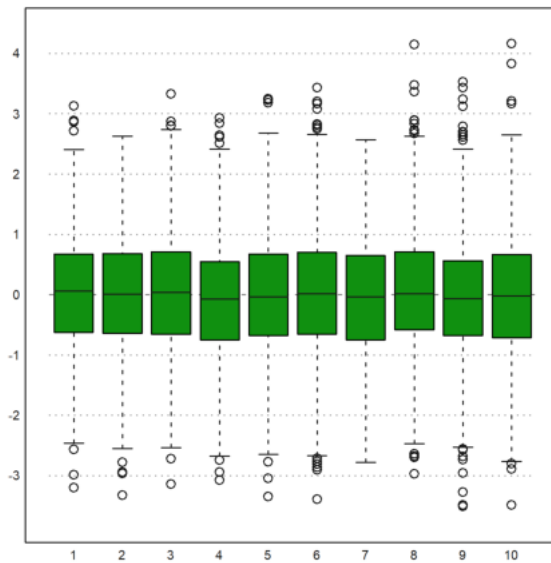
```
>p=normal(1,1000); //1000 random normal-distributed sample p
>plot2d(p,distribution=20,style="v"); // plot the random sample p
>plot2d("qnormal(x,0,1)",add=1); // add the standard normal distribution plot
```



Perhatikan perbedaan antara plot batang (sampel) dan kurva normal (distribusi sesungguhnya). Masukkan kembali ketiga perintah tersebut untuk melihat hasil pengambilan sampel yang lain.

Berikut ini adalah perbandingan 10 simulasi dari 1000 nilai terdistribusi normal dengan menggunakan apa yang disebut plot kotak. Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, serta pencilan.

```
>p=normal(10,1000); boxplot(p):
```



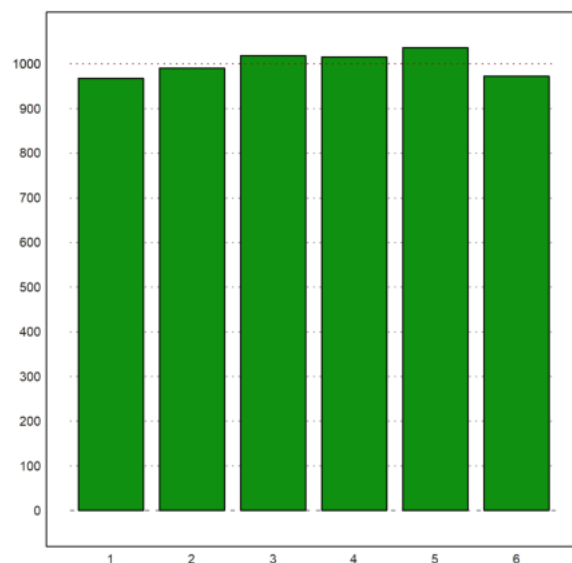
images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-007.png

Untuk menghasilkan bilangan bulat acak, Euler memiliki intrandom.

Mari kita simulasikan pelemparan dadu dan memplot distribusinya.

Kita menggunakan fungsi `getmultiplicities(v,x)`, yang menghitung seberapa sering elemen-elemen dari `v` muncul di dalam `x`. Kemudian kita memplot hasilnya menggunakan `columnplot()`.

```
>k=intrandom(1,6000,6); ...
> columnplot(getmultiplicities(1:6,k)); ...
> ygrid(1000,color=red):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-008.png

Meskipun `intrandom(n,m,k)` menghasilkan bilangan bulat yang terdistribusi secara seragam dari 1 sampai `k`, adalah mungkin untuk menggunakan distribusi bilangan bulat yang lain dengan `randpint()`.

Pada contoh berikut, probabilitas untuk 1,2,3 adalah 0.4, 0.1, 0.5 secara berurutan.

```
>randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,%)
```

```
[388, 109, 503]
```

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Lihatlah ke dalam referensi.

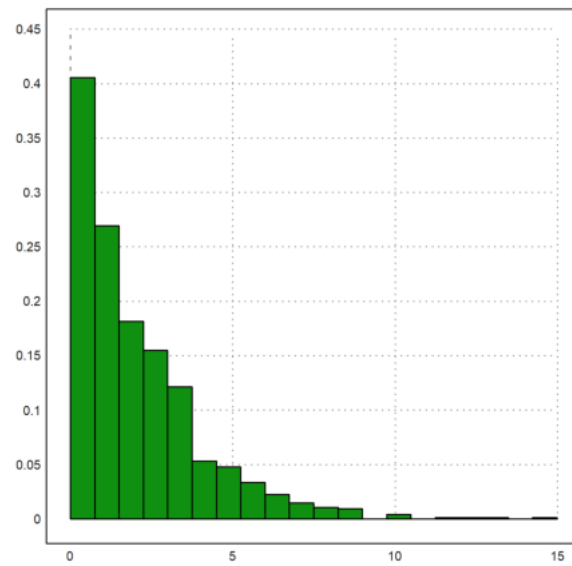
Misalnya, kita mencoba distribusi eksponensial. Sebuah variabel acak kontinu `X` dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

dengan parameter

$\lambda = \frac{1}{\mu}$, μ adalah rata-rata, dan dilambangkan dengan $X \sim \text{Eksponensial}(\lambda)$.

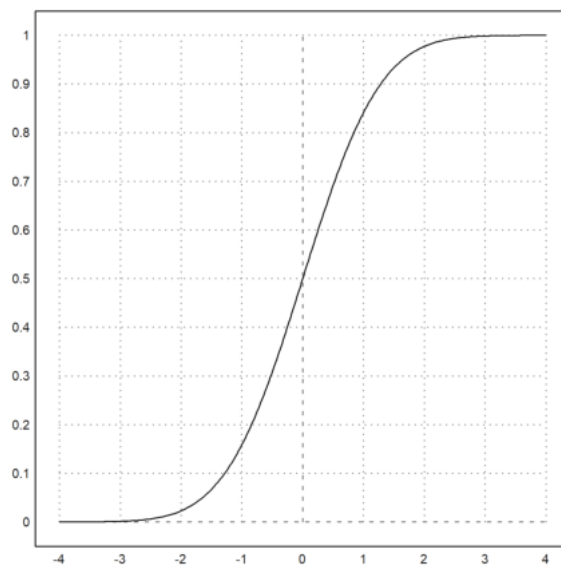
```
>plot2d(randexponential(1,1000,2),>distribution):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-011.png

Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan kebalikannya.

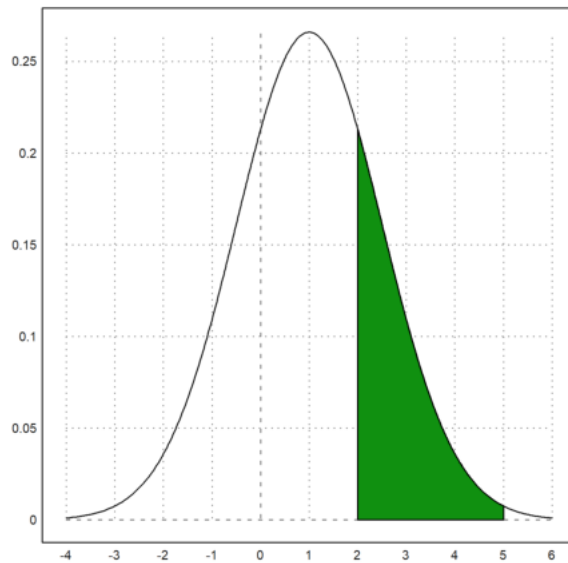
```
>plot2d("normaldis",-4,4):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-012.png

Berikut ini adalah salah satu cara untuk memplot kuantil.

```
>plot2d("qnormal(x,1,1.5)",-4,6); ...  
> plot2d("qnormal(x,1,1.5)",a=2,b=5,>add,>filled):
```



$$\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} dt.$$

The probability to be in the green area is the following.

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

```
0.248662156979
```

Hal ini dapat dihitung secara numerik dengan integral berikut ini.

$$\int_2^5 \frac{1}{1.5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-1}{1.5})^2} dx.$$

```
>gauss("qnormal(x,1,1.5)",2,5)
```

```
0.248662156979
```

Mari kita bandingkan distribusi binomial dengan distribusi normal dengan rata-rata dan deviasi yang sama. Fungsi `invbindis()` menyelesaikan interpolasi linier antara nilai bilangan bulat.

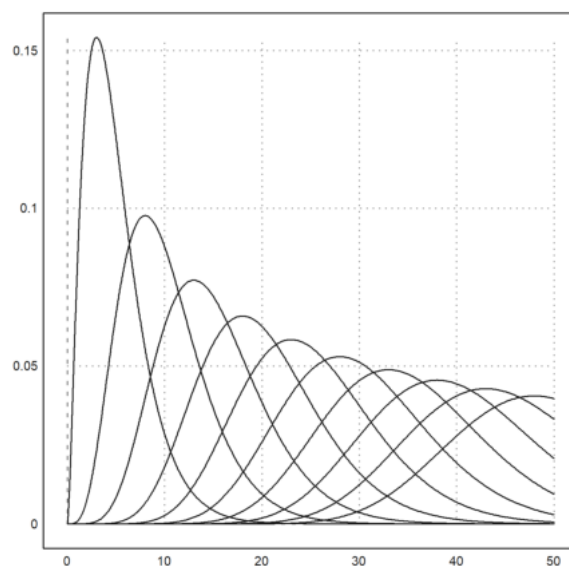
```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

```
525.516721219
```

```
526.007419394
```

Fungsi `qdis()` adalah densitas dari distribusi chi-square. Seperti biasa, Euler memetakan vektor ke fungsi ini. Dengan demikian kita mendapatkan plot semua distribusi chi-kuadrat dengan derajat 5 hingga 30 dengan mudah dengan cara berikut.

```
>plot2d("qchidis(x,(5:5:50)"),0,50):
```



Euler memiliki fungsi-fungsi yang akurat untuk mengevaluasi distribusi-distribusi. Mari kita periksa `chidis()` dengan sebuah integral.

Penamaannya diusahakan untuk konsisten. Sebagai contoh, * distribusi chi-kuadrat adalah `chidis()`, * fungsi kebalikannya adalah `invchidis()`, * densitasnya adalah `qchidis()`.

Pelengkap dari distribusi (ekor atas) adalah `chicdis()`.

```
>chidis(1.5,2), integrate("qchidis(x,2)",0,1.5)
```

```
0.527633447259
0.527633447259
```

Distribusi Diskrit

Untuk menentukan distribusi diskrit Anda sendiri, Anda dapat menggunakan metode berikut.

Pertama, kita tetapkan fungsi distribusinya.

```
>wd = 0/((1:6)+[-0.01,0.01,0,0,0,0])/6
```

```
[0, 0.165, 0.335, 0.5, 0.666667, 0.833333, 1]
```

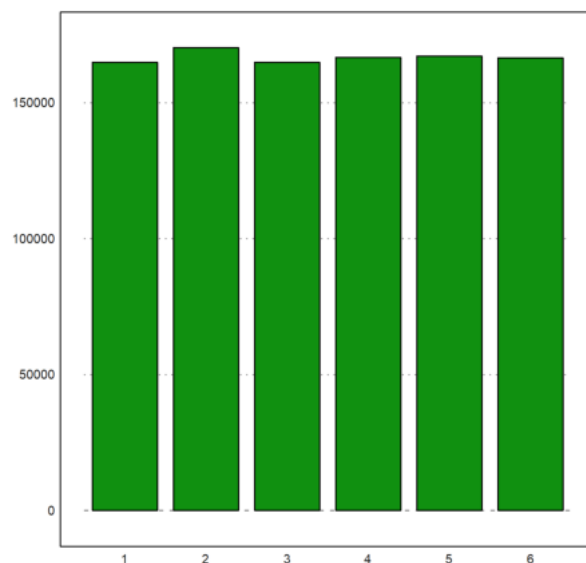
Artinya, dengan probabilitas `wd[i+1]-wd[i]` kita menghasilkan nilai acak `i`.

Ini hampir merupakan distribusi yang seragam. Mari kita definisikan sebuah generator bilangan acak untuk ini. Fungsi `find(v,x)` menemukan nilai `x` dalam vektor `v`. Fungsi ini juga dapat digunakan untuk vektor `x`.

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahan ini sangat halus sehingga kita hanya bisa melihatnya setelah melakukan iterasi yang sangat banyak.

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```



[images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-017.png](#)

Berikut ini adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai $1 \dots K$ dalam `v`. Kami menerima hasilnya, jika untuk semua frekuensi

$$f_i - \frac{1}{K} < \frac{\delta}{\sqrt{n}}.$$

```
>function checkrandom (v, delta=1) ...
```

```
  K=max(v); n=cols(v);
  fr=getfrequencies(v,1:K);
  return max(fr/n-1/K)<delta/sqrt(n);
endfunction
```

Memang fungsi ini menolak distribusi seragam.

```
>checkrandom(wrongdice(1,1000000))
```

```
0
```

Dan ini menerima generator acak bawaan.

```
>checkrandom(intrandom(1,1000000,6))
```

Kita dapat menghitung distribusi binomial. Pertama, ada `binomialsom()`, yang mengembalikan probabilitas i atau kurang dari n percobaan

```
>bindis(410,1000,0.4)
```

```
0.751401349654
```

Fungsi Beta invers digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p . Tingkat defaultnya adalah α .

Arti dari interval ini adalah jika p berada di luar interval, hasil yang diamati sebesar 410 dalam 1000 jarang terjadi.

```
>clopperpearson(410,1000)
```

```
[0.37932, 0.441212]
```

Perintah berikut ini adalah cara langsung untuk mendapatkan hasil di atas. Tetapi untuk n yang besar, penjumlahan langsung tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*pi(1-p)(n-i))
```

```
0.751401349655
```

Omong-omong, `invbinsum()` menghitung kebalikan dari `binomialsom()`.

```
>invbindis(0.75,1000,0.4)
```

```
409.932733047
```

Dalam Bridge, kita mengasumsikan 5 kartu yang terbuka (dari 52 kartu) di dua tangan (26 kartu). Mari kita hitung probabilitas distribusi yang lebih buruk dari 3:2 (misalnya 0:5, 1:4, 4:1, atau 5:0).

```
>2*hypergeomsum(1,5,13,26)
```

```
0.321739130435
```

Ada juga simulasi distribusi multinomial.

```
>randmultinomial(10,1000,[0.4,0.1,0.5])
```

386	90	524
422	82	496
438	95	467
411	116	473
408	92	500
387	105	508
405	96	499
426	90	484
394	100	506
377	103	520

Memplot Data

Untuk memplot data, kami mencoba hasil pemilihan umum Jerman sejak tahun 1990, yang diukur dalam kursi.

```
>BW := [ ...
> 1990,662,319,239,79,8,17; ...
> 1994,672,294,252,47,49,30; ...
> 1998,669,245,298,43,47,36; ...
> 2002,603,248,251,47,55,2; ...
> 2005,614,226,222,61,51,54; ...
> 2009,622,239,146,93,68,76; ...
> 2013,631,311,193,0,63,64];
```

Untuk pesta, kami menggunakan serangkaian nama.

```
>P:=[“CDU/CSU”, “SPD”, “FDP”, “Gr”, “Li”];
```

Mari kita cetak persentasenya dengan baik.

Pertama kita ekstrak kolom-kolom yang diperlukan. Kolom 3 sampai 7 adalah kursi masing-masing partai, dan kolom 2 adalah jumlah total kursi. kolom adalah tahun pemilihan.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]’;
```

Kemudian kita mencetak statistik dalam bentuk tabel. Kita menggunakan nama sebagai judul kolom, dan tahun sebagai judul baris. Lebar default untuk kolom adalah $wc = 10$, tetapi kami lebih suka output yang lebih padat. Kolom-kolom akan diperluas untuk label-kolom, jika perlu.

```
>writetable(BT*100,wc=6,dc=0,>fixed,labc=P,labr=YT)
```

	CDU/CSU	SPD	FDP	Gr	Li
1990	48	36	12	1	3
1994	44	38	7	7	4
1998	37	45	6	7	5
2002	41	42	8	9	0
2005	37	36	10	8	9
2009	38	23	15	11	12
2013	49	31	0	10	10

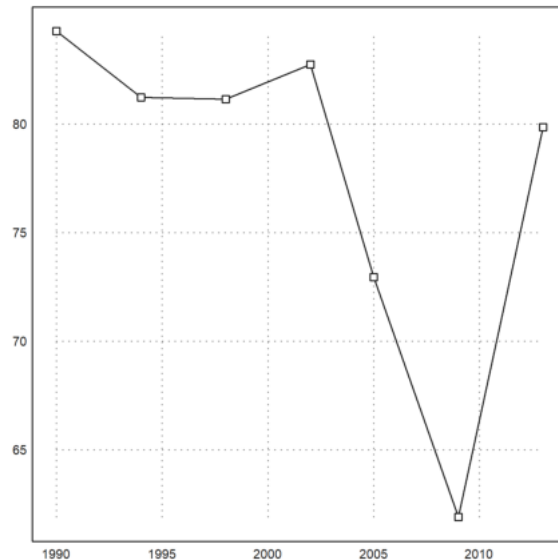
Perkalian matriks berikut ini mengekstrak jumlah persentase dua partai besar yang menunjukkan bahwa partai-partai kecil telah memperoleh suara di parlemen hingga tahun 2009.

```
>BT1:=(BT.[1;1;0;0;0])*100
```

```
[84.29, 81.25, 81.1659, 82.7529, 72.9642, 61.8971, 79.8732]
```

Ada juga plot statistik sederhana. Kita menggunakannya untuk menampilkan garis dan titik secara bersamaan. Alternatif lainnya adalah memanggil plot2d dua kali dengan >add.

```
>statplot(YT,BT1,"b"):
```



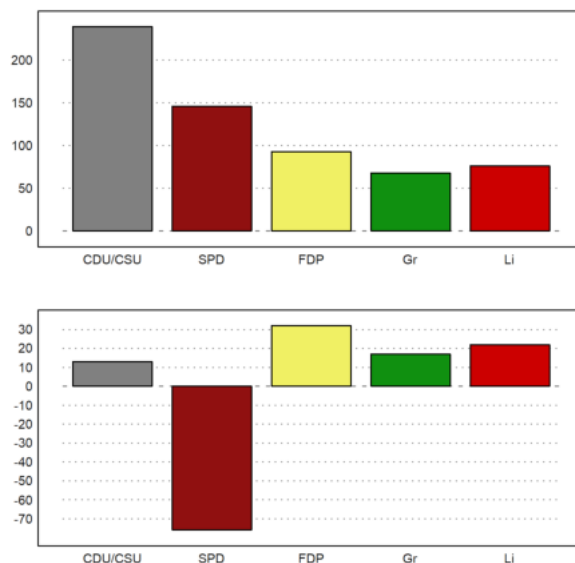
images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-019.png

Tentukan beberapa warna untuk masing-masing pihak.

```
>CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

Sekarang kita dapat memplot hasil pemilu 2009 dan perubahannya ke dalam satu plot menggunakan figure. Kita dapat menambahkan vektor kolom pada setiap plot.

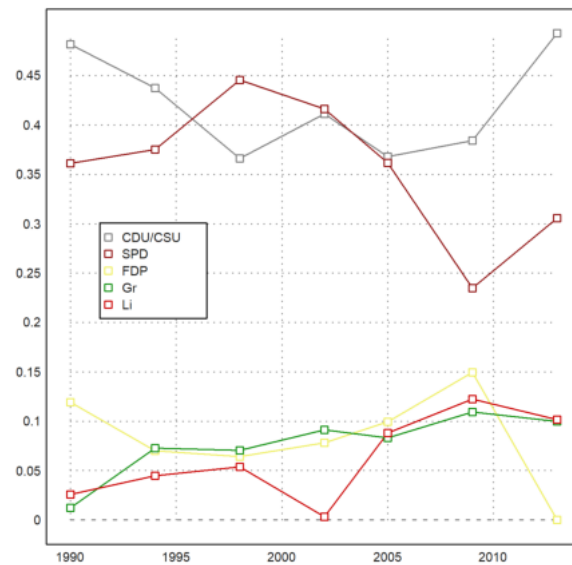
```
>figure(2,1); ...
> figure(1); columnsplot(BW[6,3:7],P,color=CP); ...
> figure(2); columnsplot(BW[6,3:7]-BW[5,3:7],P,color=CP); ...
> figure(0):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-020.png

Plot data menggabungkan baris data statistik dalam satu plot.

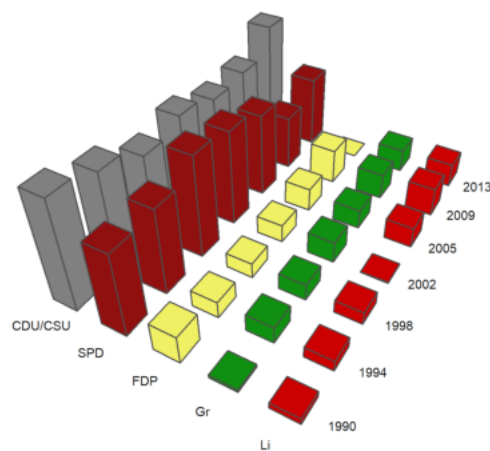
```
>J:=BW[,1]'; DP:=BW[,3:7]'; ...
> dataplot(YT,BT',color=CP); ...
> labelbox(P,color=CP,styles="[]",>points,w=0.2,x=0.3,y=0.4):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-021.png

Plot kolom 3D menunjukkan deretan data statistik dalam bentuk kolom. Kami menyediakan label untuk baris dan kolom. angle adalah sudut pandang.

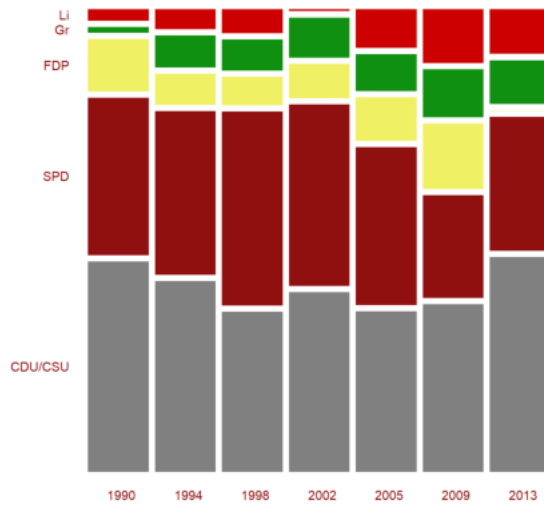
```
>columnsplot3d(BT,scols=P,srows=YT, ...
> angle=30°,ccols=CP):
```



Representasi

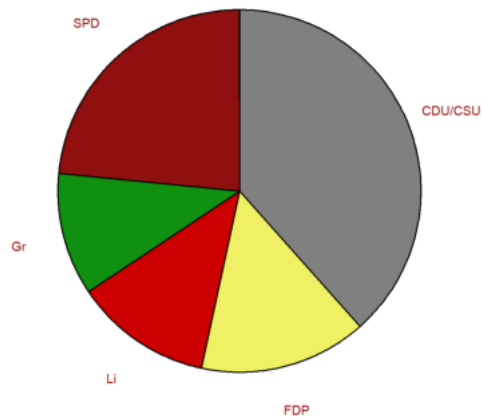
lainnya adalah plot mosaik. Perhatikan bahwa kolom-kolom pada plot mewakili kolom-kolom pada matriks di sini. Karena panjangnya label CDU/CSU, kita mengambil jendela yang lebih kecil dari biasanya.

```
>shrinkwindow(>smaller); ...
> mosaicplot(BT',srows=YT,scols=P,color=CP,style="#"); ...
> shrinkwindow():
```



Kita juga bisa membuat diagram lingkaran. Karena warna hitam dan kuning membentuk koalisi, kita menyusun ulang elemen-elemennya.

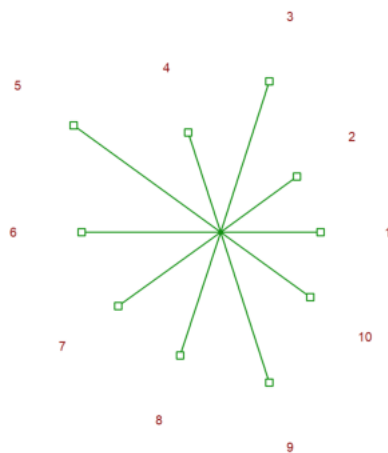
```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-024.png

Berikut ini jenis plot yang lain.

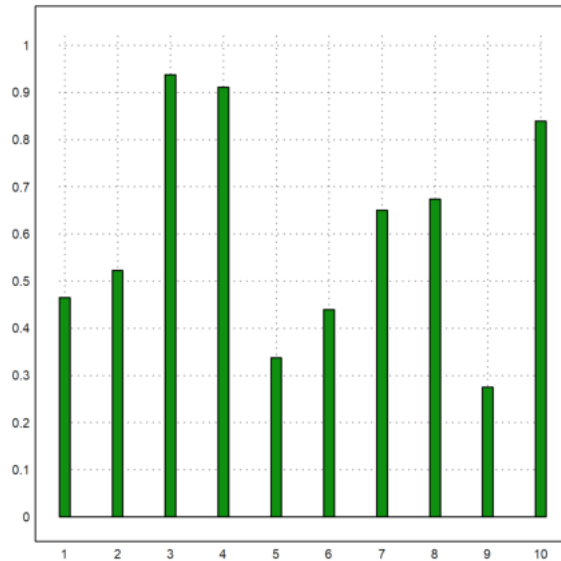
```
>starplot(normal(1,10)+4,lab=1:10,>rays):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-025.png

Beberapa plot di plot2d bagus untuk statika. Berikut ini adalah plot impuls dari data acak, yang terdistribusi secara seragam dalam $[0,1]$.

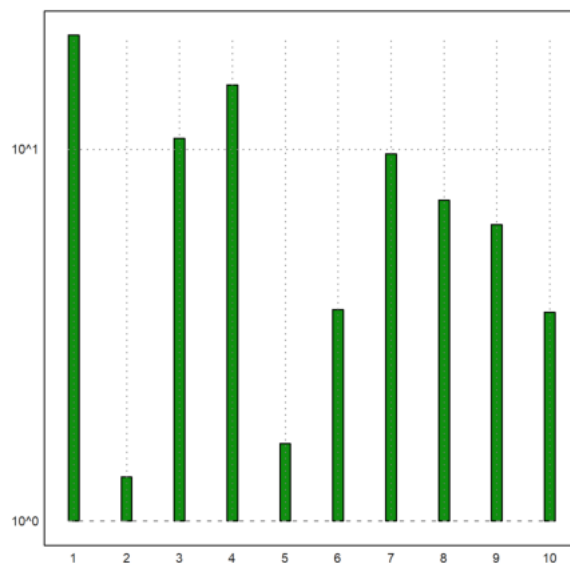
```
>plot2d(makeimpulse(1:10,random(1,10)),>bar):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-026.png

Tetapi untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

```
>logimpulseplot(1:10,-log(random(1,10))*10):
```

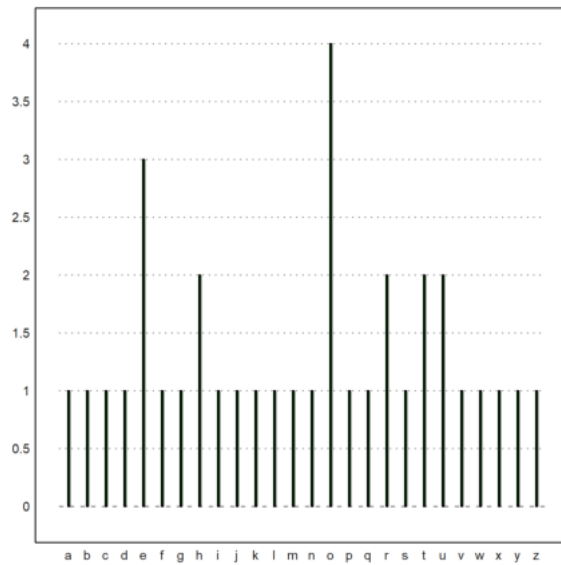


images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-027.png

Fungsi `columnspot()` lebih mudah digunakan, karena hanya membutuhkan sebuah vektor nilai. Selain itu, fungsi ini dapat mengatur labelnya menjadi apa pun yang kita inginkan, kita telah mendemonstrasikan hal ini dalam tutorial ini.

Berikut ini adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan memplot statistik.

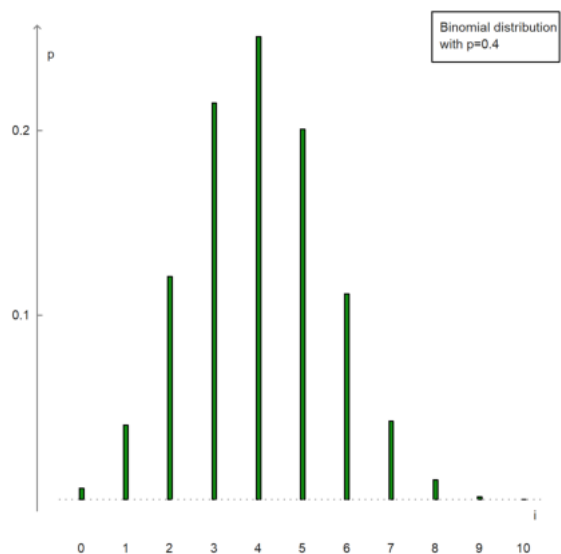
```
>v=strtochar("the quick brown fox jumps over the lazy dog"); ...
>w=ascii("a"):ascii("z"); x=getmultiplicities(w,v); ...
>cw=[]; for k=w; cw=cw|char(k); end; ...
>columnspot(x,lab=cw,width=0.05):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-028.png

Anda juga dapat menetapkan sumbu secara manual.

```
>n=10; p=0.4; i=0:n; x=bin(n,i)*pi*(1-p)(n-i); ...
> columnplot(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"]):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-029.png

Berikut ini adalah cara untuk memplot frekuensi angka dalam vektor.

Kami membuat vektor angka acak bilangan bulat 1 hingga 6.

```
>v:=inrandom(1,10,10)
```

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

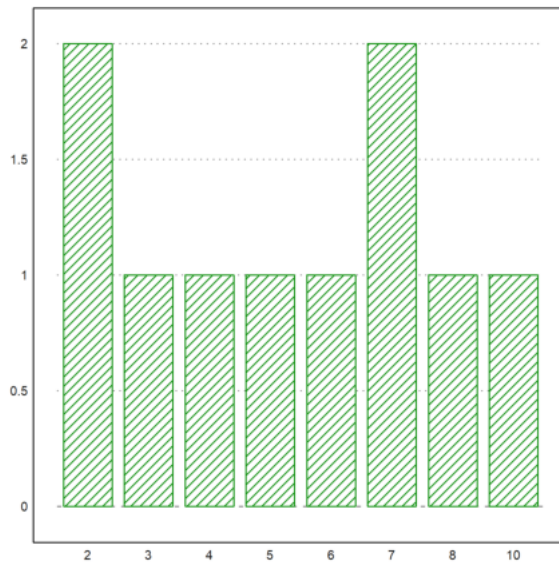
Kemudian ekstrak nomor unik dalam v.

```
>vu:=unique(v)
```

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

Dan memplot frekuensi dalam plot kolom.

```
>columnplot(getmultiplicities(vu,v),lab=vu,style="/"):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-030.png

Kami ingin mendemonstrasikan fungsi untuk distribusi nilai empiris.

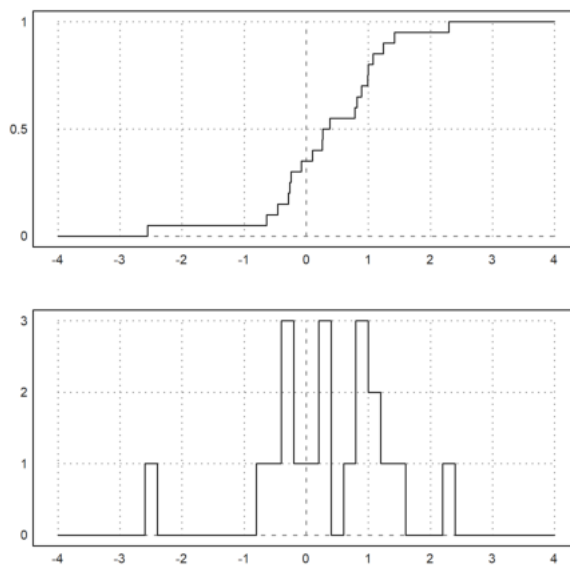
```
>x=normal(1,20);
```

Fungsi `empdist(x,vs)` membutuhkan larik nilai yang telah diurutkan. Jadi kita harus mengurutkan `x` sebelum dapat menggunakannya.

```
>xs=sort(x);
```

Kemudian kita memplot distribusi empiris dan beberapa batang kepadatan ke dalam satu plot. Alih-alih plot batang untuk distribusi, kali ini kami menggunakan plot gigi gergaji.

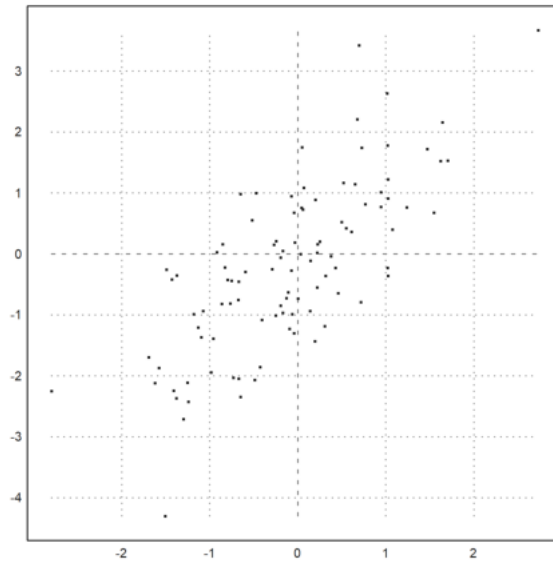
```
>figure(2,1); ...
> figure(1); plot2d("empdist",-4,4;xs); ...
> figure(2); plot2d(histo(x,v=-4:0.2:4,<bar)); ...
> figure(0):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-031.png

Plot sebaran mudah dilakukan di Euler dengan plot titik biasa. Grafik berikut ini menunjukkan bahwa X dan $X+Y$ berkorelasi positif secara jelas.

```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style=".."):
```



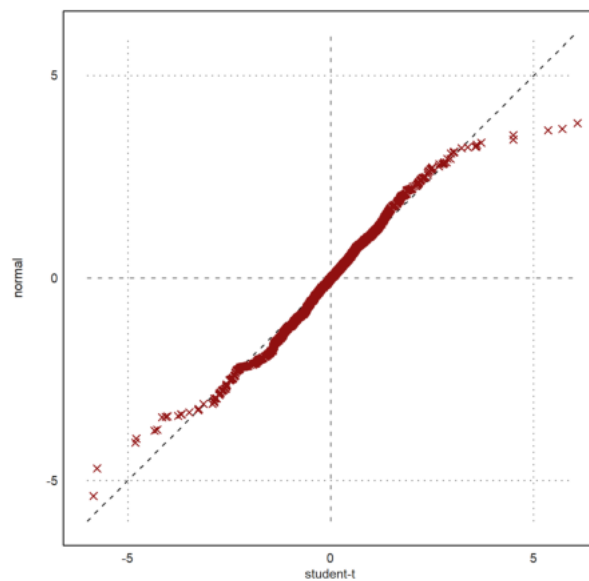
images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-032.png

Sering kali, kita ingin membandingkan dua sampel dari distribusi yang berbeda.

Hal ini dapat dilakukan dengan plot kuantil-kuantil.

Untuk pengujian, kami mencoba distribusi student-t dan distribusi eksponensial.

```
>x=randt(1,1000,5); y=randnormal(1,1000,mean(x),dev(x)); ...
> plot2d("x",r=6,style="--",yl="normal",xl="student-t",>vertical); ...
> plot2d(sort(x),sort(y),>points,color=red,style="x",>add):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-033.png

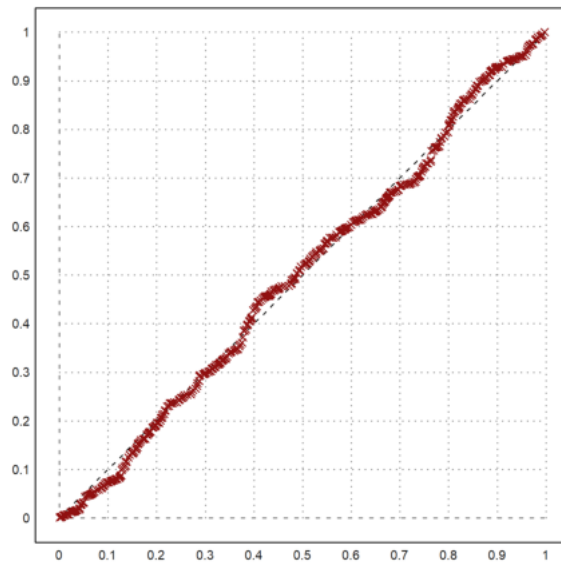
Plot ini dengan jelas menunjukkan bahwa nilai yang terdistribusi normal cenderung lebih kecil pada ujung yang ekstrim.

Jika kita memiliki dua distribusi dengan ukuran yang berbeda, kita dapat memperluas distribusi yang lebih kecil atau memperkecil distribusi yang lebih besar. Fungsi berikut ini bagus untuk keduanya. Fungsi ini mengambil nilai median dengan persentase antara 0 dan 1.

```
>function medianexpand (x,n) := median(x,p=linspace(0,1,n-1));
```

Mari kita bandingkan dua distribusi yang sama.

```
>x=random(1000); y=random(400); ...
> plot2d("x",0,1,style="--"); ...
> plot2d(sort(medianexpand(x,400)),sort(y),>points,color=red,style="x",>add):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-034.png

Regresi dan Korelasi

Regresi linier dapat dilakukan dengan fungsi `polyfit()` atau berbagai fungsi fit.

Sebagai permulaan, kita mencari garis regresi untuk data univariat dengan `polyfit(x,y,1)`.

```
>x=1:10; y=[2,3,1,5,6,3,7,8,9,8]; writetable(x'|y',labc=["x","y"])
```

x	y
1	2
2	3
3	1
4	5
5	6
6	3
7	7
8	8
9	9
10	8

Kami ingin membandingkan kecocokan tanpa bobot dan dengan bobot. Pertama, koefisien dari kecocokan linier.

```
>p=polyfit(x,y,1)
```

```
[0.733333, 0.812121]
```

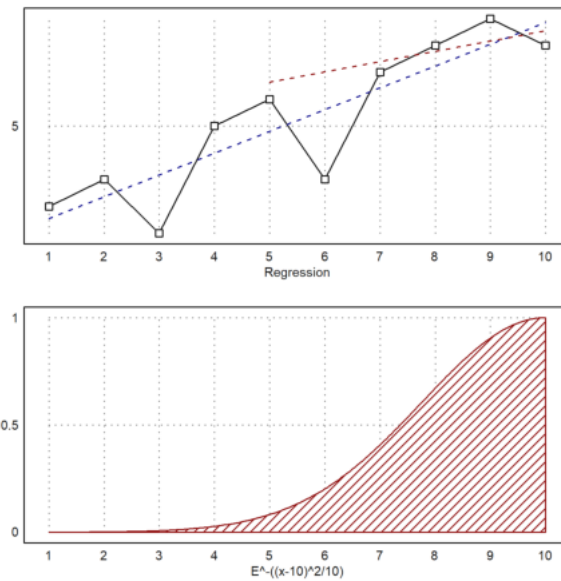
Sekarang, koefisien dengan bobot yang menekankan nilai terakhir.

```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

```
[4.71566, 0.38319]
```

Kami menempatkan semuanya ke dalam satu plot untuk titik-titik dan garis regresi, dan untuk bobot yang digunakan.

```
>figure(2,1); ...
> figure(1); statplot(x,y,"b",xl="Regression"); ...
> plot2d("evalpoly(x,p)",>add,color=blue,style="-"); ...
> plot2d("evalpoly(x,pw)",5,10,>add,color=red,style="-"); ...
> figure(2); plot2d(w,1,10,>filled,style="/",fillcolor=red,xl=w); ...
> figure(0);
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-035.png

Untuk contoh lain, kita membaca survei tentang siswa, usia mereka, usia orang tua mereka, dan jumlah saudara kandung dari sebuah file.

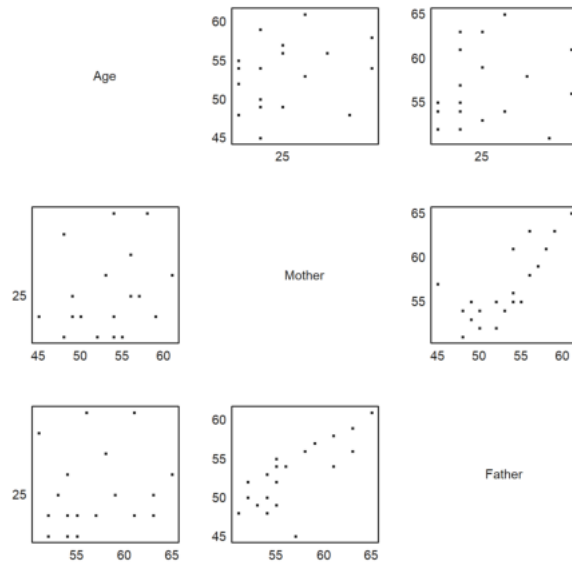
Tabel ini berisi “m” dan “f” pada kolom kedua. Kita menggunakan variabel tok2 untuk mengatur terjemahan yang tepat dan bukannya membiarkan readtable() mengumpulkan terjemahan.

```
> {MS,hd}:=readtable("table1.dat",tok2=["m","f"]); ...
> writetable(MS,labc=hd,tok2=["m","f"]);
```

Person	Sex	Age	Mother	Father	Siblings
1	m	29	58	61	1
2	f	26	53	54	2
3	m	24	49	55	1
4	f	25	56	63	3
5	f	25	49	53	0
6	f	23	55	55	2
7	m	23	48	54	2
8	m	27	56	58	1
9	m	25	57	59	1
10	m	24	50	54	1
11	f	26	61	65	1
12	m	24	50	52	1
13	m	29	54	56	1
14	m	28	48	51	2
15	f	23	52	52	1
16	m	24	45	57	1
17	f	24	59	63	0
18	f	23	52	55	1
19	m	24	54	61	2
20	f	23	54	55	1

Bagaimana usia saling bergantung satu sama lain? Kesan pertama datang dari scatterplot berpasangan.

```
> scatterplots(tablecol(MS,3:5),hd[3:5]):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-036.png

Jelas bahwa usia ayah dan ibu saling bergantung satu sama lain. Mari kita tentukan dan plot garis regresinya.

```
>cs:=MS[4:5]'; ps:=polyfit(cs[1],cs[2],1)
```

```
[17.3789, 0.740964]
```

Ini jelas merupakan model yang salah. Garis regresinya adalah $s = 17 + 0,74t$, di mana t adalah usia ibu dan s adalah usia ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak terlalu banyak.

Sebaliknya, kami menduga fungsi seperti $s = a + t$. Kemudian a adalah rata-rata dari $s-t$. Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

```
>da:=mean(cs[2]-cs[1])
```

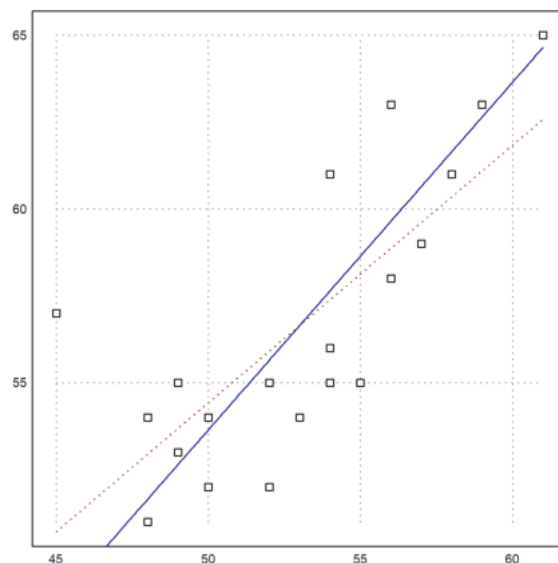
```
3.65
```

Mari kita plotkan ini ke dalam satu scatter plot.

ya adalah $s = 17 + 0,74t$, di mana t adalah usia ibu dan s adalah usia ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak terlalu banyak.

Sebaliknya, kami menduga fungsi seperti $s = a + t$. Kemudian a adalah rata-rata dari $s-t$. Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

```
>plot2d(cs[1],cs[2],>points); ...
>plot2d("evalpoly(x,ps)",color=red,style=".",>add); ...
>plot2d("x+da",color=blue,>add):
```



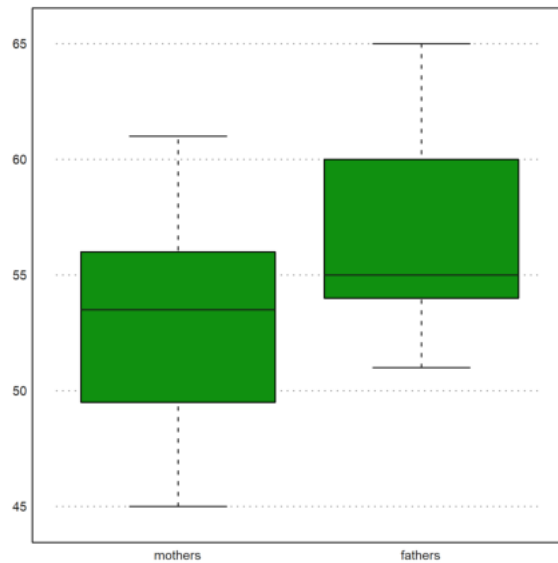
images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-037.png

Berikut ini adalah plot kotak dari kedua usia tersebut. Ini hanya menunjukkan, bahwa usia keduanya berbeda.

adalah usia ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak terlalu banyak.

Sebaliknya, kami menduga fungsi seperti $s = a + t$. Kemudian a adalah rata-rata dari $s-t$. Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

```
>boxplot(cs,["mothers","fathers"]):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-038.png

Sangat menarik bahwa perbedaan dalam median tidak sebesar perbedaan dalam mean.

```
>median(cs[2])-median(cs[1])
```

1.5

Koefisien korelasi menunjukkan korelasi positif.

```
>correl(cs[1],cs[2])
```

0.7588307236

Korelasi peringkat adalah ukuran untuk urutan yang sama dalam kedua vektor. Korelasi ini juga cukup positif.

```
>rankcorrel(cs[1],cs[2])
```

0.758925292358

Membuat Fungsi baru

Tentu saja, bahasa EMT dapat digunakan untuk memprogram fungsi baru. Misalnya, kita mendefinisikan fungsi kemiringan.

Di mana m adalah rata-rata dari x .

$$sk(x) = \frac{\sqrt{n} \sum_i (x_i - m)^3}{(\sum_i (x_i - m)^2)^{3/2}}$$

dimana m adalah rata-rata dari x

```
>function skew (x:vector) ...
```

```
m=mean(x);
return sqrt(cols(x))*sum((x-m)^3)/(sum((x-m)^2))^(3/2);
endfunction
```

Seperti yang Anda lihat, kita dapat dengan mudah menggunakan bahasa matriks untuk mendapatkan implementasi yang sangat singkat dan efisien. Mari kita coba fungsi ini.

```
>data=normal(20); skew(normal(10))
```

0.261583655199

Berikut ini adalah fungsi lain, yang disebut koefisien kemencengan Pearson.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)
```

```
>skew1(data)
```

0.0173032894764

Simulasi Monte Carlo

Euler dapat digunakan untuk mensimulasikan kejadian acak. Kita telah melihat contoh sederhana di atas. Berikut ini adalah contoh lainnya, yang mensimulasikan 1000 kali pelemparan 3 dadu, dan menanyakan distribusi dari jumlah tersebut.

```
right)^{3/2}}
```

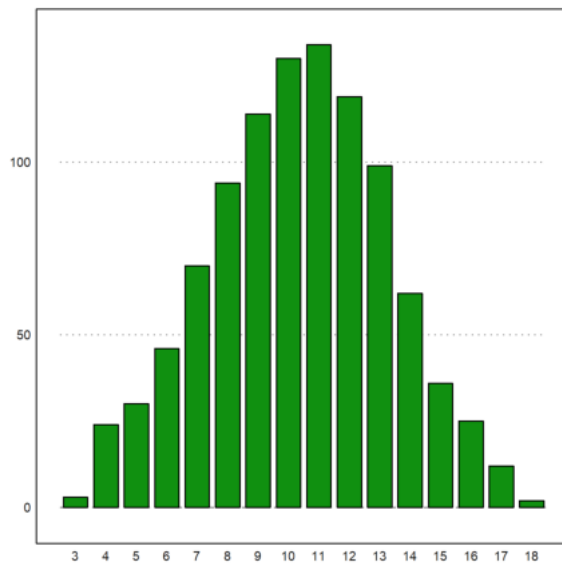
dimana m adalah rata-rata dari x

```
>ds:=sum(intrandom(1000,3,6))*'; fs=getmultiplicities(3:18,ds)
```

```
[3, 24, 30, 46, 70, 94, 114, 130, 134, 119, 99, 62, 36,  
25, 12, 2]
```

Kita bisa merencanakan ini sekarang.

```
>columnsplot(fs,lab=3:18):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-040.png

Untuk menentukan distribusi yang diharapkan tidaklah mudah. Kami menggunakan rekursi tingkat lanjut untuk hal ini.

Fungsi berikut ini menghitung jumlah cara angka k dapat direpresentasikan sebagai jumlah n angka dalam rentang 1 hingga m . Fungsi ini bekerja secara rekursif dengan cara yang jelas.

Data dari x

```
>function map countways (k; n, m) ...
```

```
    if n==1 then return k>=1 && k<=m  
    else  
        sum=0;  
        loop 1 to m; sum=sum+countways(k-#,n-1,m); end;  
        return sum;  
    end;  
endfunction
```

Berikut ini adalah hasil dari tiga lemparan dadu.

```
>countways(5:25,5,5)
```

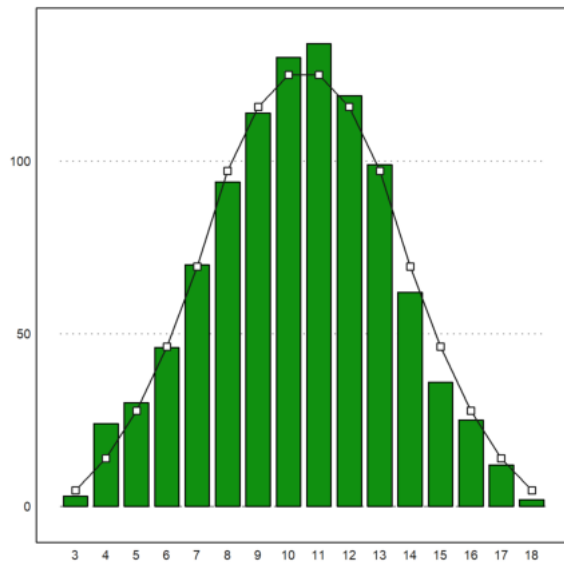
```
[1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320,  
255, 185, 121, 70, 35, 15, 5, 1]
```

```
>cw=countways(3:18,3,6)
```

```
[1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3,  
1]
```

Kami menambahkan nilai yang diharapkan ke plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-041.png

Untuk simulasi lainnya, deviasi nilai rata-rata dari n variabel acak berdistribusi normal 0-1 adalah $1/\sqrt{n}$.

```
>longformat; 1/sqrt(10)
```

```
0.316227766017
```

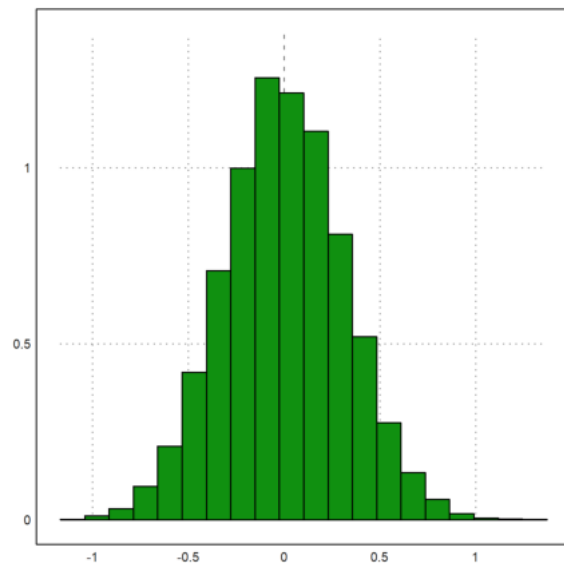
Mari kita periksa dengan sebuah simulasi. Kami menghasilkan 10.000 kali 10 vektor acak.

0-1 adalah $1/\sqrt{n}$.

```
>M=normal(10000,10); dev(mean(M)')
```

```
0.314940039504
```

```
>plot2d(mean(M)',>distribution):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-042.png

Median dari 10 bilangan acak berdistribusi normal 0-1 memiliki deviasi yang lebih besar.

0-1 adalah $1/\sqrt{n}$.

```
>dev(median(M)')
```

```
0.370754309885
```

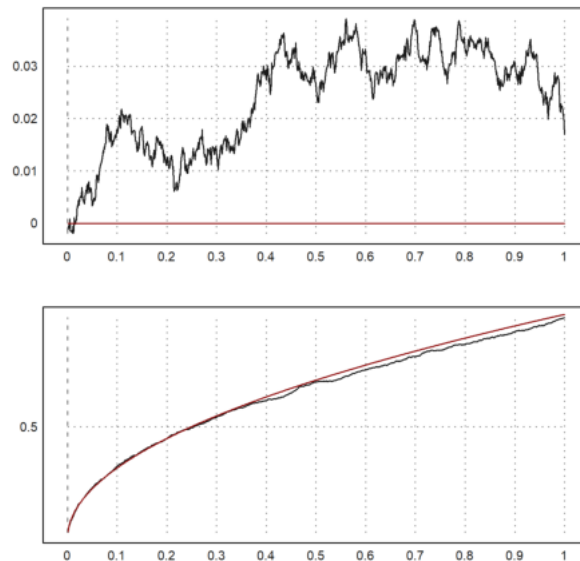
Karena kita dapat dengan mudah menghasilkan jalan acak, kita dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami kemudian memplot deviasi standar dan rata-rata dari langkah ke- n dari proses-proses ini bersama dengan nilai yang diharapkan dalam warna merah.

```
>n=1000; m=1000; M=cumsum(normal(n,m)/sqrt(m)); ...
```

```
> t=(1:n)/n; figure(2,1); ...
```

```
> figure(1); plot2d(t,mean(M')'); plot2d(t,0,color=red,>add); ...
```

```
> figure(2); plot2d(t,dev(M'))'; plot2d(t,sqrt(t),color=red,>add); ...
> figure(0):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-043.png

Tes

Tes adalah alat yang penting dalam statistik. Dalam Euler, banyak tes yang diterapkan. Semua tes ini mengembalikan kesalahan yang kita terima jika kita menolak hipotesis nol.

Sebagai contoh, kita menguji lemparan dadu untuk distribusi yang seragam. Pada 600 lemparan, kita mendapatkan nilai berikut, yang kita masukkan ke dalam uji chi-kuadrat.

```
>chitest([90,103,114,101,103,89],dup(100,6)')
```

```
0.498830517952
```

Uji chi-square juga memiliki mode, yang menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya seharusnya hampir sama. Parameter >p menginterpretasikan vektor y sebagai vektor probabilitas.

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

```
0.52
```

Kesalahan ini terlalu besar. Jadi kita tidak bisa menolak distribusi seragam. Ini tidak membuktikan bahwa dadu kita adil. Tetapi kita tidak dapat menolak hipotesis kita.

Selanjutnya kita buat 1000 lemparan dadu dengan menggunakan generator bilangan acak, dan lakukan pengujian yang sama.

tkan nilai berikut, yang kita masukkan ke dalam uji chi-kuadrat.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

```
0.489845011165
```

```
Mari kita uji nilai rata-rata 100 dengan uji-t. >s=200+normal([1,100])*10; ...
> ttest(mean(s),dev(s),100,200)
```

```
0.179168415311
```

Fungsi ttest() membutuhkan nilai rata-rata, deviasi, jumlah data, dan nilai rata-rata untuk diuji.

Sekarang mari kita periksa dua pengukuran untuk mean yang sama. Kita tolak hipotesis bahwa kedua pengukuran tersebut memiliki nilai rata-rata yang sama, jika hasilnya < 0,05.

```
0.14922232036
```

Jika kita menambahkan bias pada satu distribusi, kita akan mendapatkan lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

```
1.78688642497e-05
```

Pada contoh berikut, kita membuat 20 lemparan dadu secara acak sebanyak 100 kali dan menghitung jumlah dadu yang muncul. Rata-rata harus ada $20/6 = 3,3$ mata dadu.

Kita tolak hipotesis bahwa kedua pengukuran tersebut memiliki nilai rata-rata yang sama, jika hasilnya < 0,05.

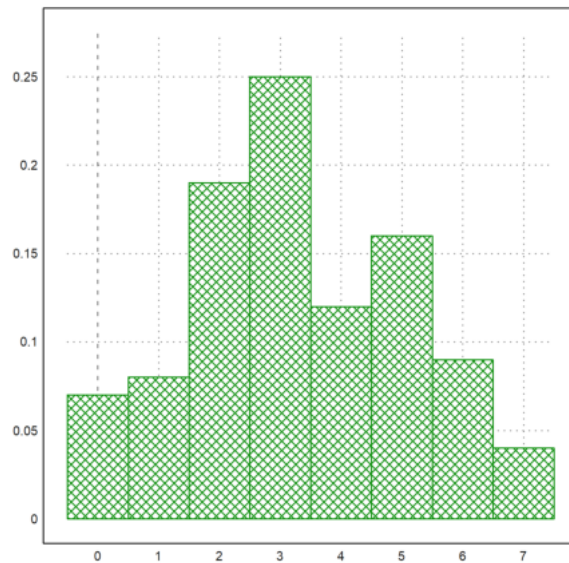
tkan nilai berikut, yang kita masukkan ke dalam uji chi-kuadrat.

```
>R=random(100,20); R=sum(R*6<=1)'; mean(R)
```

3.31

Sekarang kita bandingkan jumlah satu dengan distribusi binomial. Pertama, kita memplot distribusi angka satu.

```
>plot2d(R,distribution=max(R)+1,even=1,style="v"):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-044.png

```
>t=count(R,21);
```

Kemudian kami menghitung nilai yang diharapkan.

```
>n=0:20; b=bin(20,n)*(1/6)^n*(5/6)^(20-n)*100;
```

Kami harus mengumpulkan beberapa angka untuk mendapatkan kategori yang cukup besar.

t distribusi angka satu.

```
>t1=sum(t[1:2])|t[3:7]|sum(t[8:21]); ...  
>b1=sum(b[1:2])|b[3:7]|sum(b[8:21]);
```

Uji chi-square menolak hipotesis bahwa distribusi kita adalah distribusi binomial, jika hasilnya $< 0,05$.

t is < 0.05 .

```
>chitest(t1,b1)
```

0.485603081154

Contoh berikut ini berisi hasil dari dua kelompok orang (laki-laki dan perempuan, katakanlah) yang memberikan suara untuk satu dari enam partai.

3,3 mata dadu.

Kita tolak hipotesis bahwa kedua pengukuran tersebut memiliki nilai rata-rata yang sama, jika hasilnya $< 0,05$.

```
>A=[23,37,43,52,64,74;27,39,41,49,63,76]; ...  
>writetable(A,wc=6,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	23	37	43	52	64	74
f	27	39	41	49	63	76

Kami ingin menguji independensi suara dari jenis kelamin. Uji tabel χ^2 melakukan hal ini. Hasilnya terlalu besar untuk menolak independensi. Jadi kita tidak dapat mengatakan, jika pemungutan suara tergantung pada jenis kelamin dari data ini.

```
>tabletest(A)
```

0.990701632326

Berikut ini adalah tabel yang diharapkan, jika kita mengasumsikan frekuensi pemungutan suara yang diamati.

```
>writetable(expectedtable(A),wc=6,dc=1,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	24.9	37.9	41.9	50.3	63.3	74.7
f	25.1	38.1	42.1	50.7	63.7	75.3

Kita dapat menghitung koefisien kontingensi yang telah dikoreksi. Karena koefisien ini sangat dekat dengan 0, kami menyimpulkan bahwa pemungutan suara tidak bergantung pada jenis kelamin.

```
>contingency(A)
```

```
0.0427225484717
```

Beberapa Tes Lainnya

Selanjutnya kita menggunakan analisis varians (uji F) untuk menguji tiga sampel data yang terdistribusi secara normal dengan nilai rata-rata yang sama. Metode ini disebut ANOVA (analisis varians). Dalam Euler, fungsi `varanalysis()` digunakan.

```
>x1=[109,111,98,119,91,118,109,99,115,109,94]; mean(x1),
```

```
106.54545454
```

```
>x2=[120,124,115,139,114,110,113,120,117]; mean(x2),
```

```
119.111111111
```

```
>x3=[120,112,115,110,105,134,105,130,121,111]; mean(x3)
```

```
116.3
```

```
>varanalysis(x1,x2,x3)
```

```
0.0138048221371
```

Ini berarti, kami menolak hipotesis nilai rata-rata yang sama. Kami melakukan ini dengan probabilitas kesalahan sebesar 1,3%.

Ada juga uji median, yang menolak sampel data dengan distribusi rata-rata yang berbeda dengan menguji median dari sampel gabungan.

```
>a=[56,66,68,49,61,53,45,58,54];
```

```
>b=[72,81,51,73,69,78,59,67,65,71,68,71];
```

```
>mediantest(a,b)
```

```
0.0241724220052
```

Uji lain tentang kesetaraan adalah uji peringkat. Uji ini jauh lebih tajam daripada uji median.

```
>ranktest(a,b)
```

```
0.00199969612469
```

Dalam contoh berikut ini, kedua distribusi memiliki rata-rata yang sama.

```
>ranktest(random(1,100),random(1,50)*3-1)
```

```
0.00776138660569
```

Sekarang mari kita coba mensimulasikan dua perawatan a dan b yang diterapkan pada orang yang berbeda.

```
>a=[8.0,7.4,5.9,9.4,8.6,8.2,7.6,8.1,6.2,8.9];
```

```
>b=[6.8,7.1,6.8,8.3,7.9,7.2,7.4,6.8,6.8,8.1];
```

Uji signum memutuskan, apakah a lebih baik daripada b.

```
>signtest(a,b)
```

```
0.0546875
```

Ini adalah kesalahan yang terlalu besar. Kita tidak dapat menolak bahwa a sama baiknya dengan b.

Uji Wilcoxon lebih tajam daripada uji ini, tetapi bergantung pada nilai kuantitatif dari perbedaan.

```
>wilcoxon(a,b)
```

```
0.0296680599405
```

Mari kita coba dua pengujian lagi dengan menggunakan rangkaian yang dihasilkan.

```
>wilcoxon(normal(1,20),normal(1,20)-1)
```

```
0.00202259852485
```

```
>wilcoxon(normal(1,20),normal(1,20))
```

```
0.645547345262
```

Bilangan Acak

Berikut ini adalah tes untuk generator bilangan acak. Euler menggunakan generator yang sangat bagus, jadi kita tidak perlu mengharapkan adanya masalah.

Pertama, kita akan membangkitkan sepuluh juta bilangan acak dalam $[0,1]$.

```
>n:=10000000; r:=random(1,n);
```

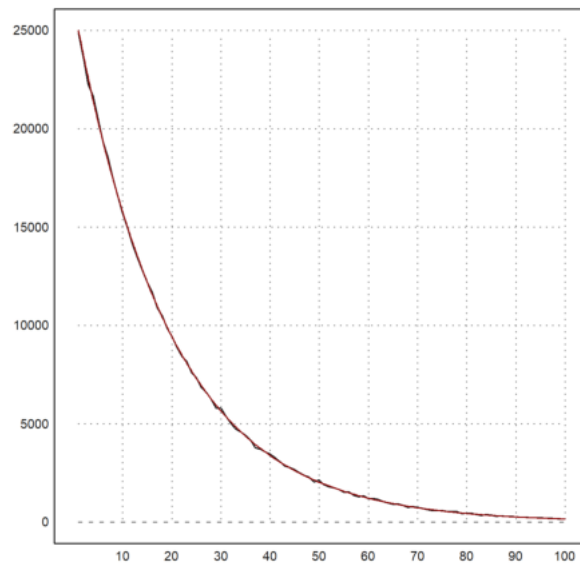
Selanjutnya, kami menghitung jarak antara dua angka yang kurang dari 0,05.

```
>a:=0.05; d:=differences(nonzeros(r<a));
```

Terakhir, kami memplot berapa kali, setiap jarak yang terjadi, dan membandingkannya dengan nilai yang diharapkan.

```
>m=getmultiplicities(1:100,d); plot2d(m); ...
```

```
> plot2d("n*(1-a)^((x-1)*a)",color=red,>add):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-045.png

Bersihkan data.

```
>remvalue n;
```

Pengantar untuk Pengguna Proyek R

Jelas, EMT tidak bersaing dengan R sebagai sebuah paket statistik. Namun, ada banyak prosedur dan fungsi statistik yang tersedia di EMT juga. Jadi EMT dapat memenuhi kebutuhan dasar. Bagaimanapun, EMT hadir dengan paket numerik dan sistem aljabar komputer.

Buku ini diperuntukkan bagi Anda yang sudah terbiasa dengan R, tetapi perlu mengetahui perbedaan sintaks EMT dan R. Kami mencoba memberikan gambaran umum mengenai hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami juga membahas cara-cara untuk bertukar data di antara kedua sistem tersebut.

Perhatikan bahwa ini adalah pekerjaan yang sedang berlangsung.

Sintaks Dasar

Hal pertama yang Anda pelajari dalam R adalah membuat sebuah vektor. Dalam EMT, perbedaan utamanya adalah bahwa operator `:` dapat mengambil ukuran langkah. Selain itu, operator ini memiliki daya ikat yang rendah.

```
>n=10; 0:n/20:n-1
```

```
[0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9]
```

Fungsi `c()` tidak ada. Anda dapat menggunakan vektor untuk menggabungkan beberapa hal.

Contoh berikut ini, seperti banyak contoh lainnya, berasal dari “Interduksi ke R” yang disertakan dengan proyek R. Jika Anda membaca PDF ini, Anda akan menemukan bahwa saya mengikuti alurnya dalam tutorial ini.

```
>x=[10.4, 5.6, 3.1, 6.4, 21.7]; [x,0,x]
```

```
[10.4, 5.6, 3.1, 6.4, 21.7, 0, 10.4, 5.6, 3.1, 6.4, 21.7]
```

Operator titik dua dengan ukuran langkah EMT digantikan oleh fungsi seq() dalam R. Kita dapat menulis fungsi ini dalam EMT.

```
>function seq(a,b,c) := a:b:c; ...  
> seq(0,-0.1,-1)
```

```
[0, -0.1, -0.2, -0.3, -0.4, -0.5, -0.6, -0.7, -0.8, -0.9, -1]
```

Fungsi rep() dari R tidak ada dalam EMT. Untuk input vektor, dapat dituliskan sebagai berikut.

```
>function rep(x:vector,n:index) := flatten(dup(x,n)); ...  
> rep(x,2)
```

```
[10.4, 5.6, 3.1, 6.4, 21.7, 10.4, 5.6, 3.1, 6.4, 21.7]
```

Perhatikan bahwa “=” atau “:=” digunakan untuk penugasan. Operator “->” digunakan untuk unit dalam EMT.

```
>125km -> ” miles”
```

```
77.6713990297 miles
```

Operator “<-” untuk penugasan menyatukan, dan bukan ide yang baik untuk R. Berikut ini akan membandingkan a dan -4 dalam EMT.

```
>a=2; a<-4
```

```
0
```

Dalam R, “a<-4<3” bisa digunakan, tetapi “a<-4<-3” tidak. Saya juga mengalami ambiguitas yang sama di EMT, tetapi saya mencoba untuk menghilangkannya.

EMT dan R memiliki vektor dengan tipe boolean. Tetapi dalam EMT, angka 0 dan 1 digunakan untuk merepresentasikan salah dan benar. Dalam R, nilai benar dan salah tetap dapat digunakan dalam aritmatika biasa seperti dalam EMT

```
>x<5, %*x
```

```
[0, 0, 1, 0, 0]  
[0, 0, 3.1, 0, 0]
```

EMT melempar kesalahan atau menghasilkan NAN tergantung pada flag “kesalahan”.

```
>errors off; 0/0, isNAN(sqrt(-1)), errors on;
```

```
NAN  
1
```

String sama saja dalam R dan EMT. Keduanya berada di lokal saat ini, bukan di Unicode.

Dalam R ada paket-paket untuk Unicode. Dalam EMT, sebuah string dapat berupa string Unicode. Sebuah string Unicode dapat diterjemahkan ke pengkodean lokal dan sebaliknya. Selain itu, “...” dapat berisi entitas HTML.

```
>u”© Ren&eacute; Grothmann”
```

```
© René Grothmann
```

Berikut ini mungkin atau mungkin tidak ditampilkan dengan benar pada sistem Anda sebagai A dengan titik dan tanda hubung di atasnya. Hal ini tergantung pada jenis huruf yang Anda gunakan.

```
>chartoutf([480])
```

```
Ã
```

Penggabungan string dilakukan dengan “+” atau “|”. Ini dapat menyertakan angka, yang akan dicetak dalam format saat ini.

```
>“pi =”+pi
```

```
pi = 3.14159265359
```

Pengindeksan

Sebagian besar waktu, ini akan bekerja seperti pada R.

Tetapi EMT akan menginterpretasikan indeks negatif dari bagian belakang vektor, sementara R menginterpretasikan x[n] sebagai x tanpa elemen ke-n.

```
>x, x[1:3], x[-2]
```

```
[10.4, 5.6, 3.1, 6.4, 21.7]
[10.4, 5.6, 3.1]
6.4
```

Perilaku R dapat dicapai dalam EMT dengan drop().

```
>drop(x,2)
```

```
[10.4, 3.1, 6.4, 21.7]
```

Vektor logika tidak diperlakukan secara berbeda dengan indeks di EMT, berbeda dengan R. Anda harus mengekstrak elemen-elemen yang bukan nol terlebih dahulu di EMT.

```
>x, x>5, x[nonzeros(x>5)]
```

```
[10.4, 5.6, 3.1, 6.4, 21.7]
[1, 1, 0, 1, 1]
[10.4, 5.6, 6.4, 21.7]
```

Sama seperti di R, vektor indeks dapat berisi pengulangan.

```
[10.4, 5.6, 5.6, 10.4]
```

Namun pemberian nama untuk indeks tidak dimungkinkan dalam EMT. Untuk paket statistik, hal ini mungkin sering diperlukan untuk memudahkan akses ke elemen-elemen vektor.

Untuk meniru perilaku ini, kita dapat mendefinisikan sebuah fungsi sebagai berikut.

```
>function sel (v,i,s) := v[indexof(s,i)]; ...
> s=["first","second","third","fourth"]; sel(x,["first","third"],s)
```

```
Trying to overwrite protected function sel!
Error in:
function sel (v,i,s) := v[indexof(s,i)]; ...
      ^
[10.4, 3.1]
```

Tipe Data

EMT memiliki lebih banyak tipe data yang tetap dibandingkan R. Jelas, dalam R terdapat vektor yang berkembang. Anda bisa mengatur sebuah vektor numerik kosong v dan memberikan sebuah nilai pada elemen v[17]. Hal ini tidak mungkin dilakukan dalam EMT.

Hal berikut ini sedikit tidak efisien.

```
>v=[]; for i=1 to 10000; v=v[i]; end;
```

EMT sekarang akan membuat vektor dengan v dan i yang ditambahkan pada tumpukan dan menyalin vektor tersebut kembali ke variabel global v.

Semakin efisien mendefinisikan vektor terlebih dahulu.

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Untuk mengubah jenis tanggal di EMT, Anda dapat menggunakan fungsi seperti complex().

```
>complex(1:4)
```

```
[ 1+0i , 2+0i , 3+0i , 4+0i ]
```

Konversi ke string hanya dapat dilakukan untuk tipe data dasar. Format saat ini digunakan untuk penggabungan string sederhana.

Tetapi ada fungsi-fungsi seperti print() atau frac().

Untuk vektor, Anda dapat dengan mudah menulis fungsi Anda sendiri.

```
>function tostr (v) ...
```

```
s="[";
loop 1 to length(v);
  s=s+print(v[#],2,0);
  if #<length(v) then s=s+","; endif;
end;
return s+"]";
endfunction
```

```
>tostr(linspace(0,1,10))
```

```
[0.00,0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1.00]
```

Untuk komunikasi dengan Maxima, ada sebuah fungsi convertmxm(), yang juga dapat digunakan untuk memformat vektor untuk output.

```
>convertmxm(1:10)
```

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

Untuk Latex, perintah tex dapat digunakan untuk mendapatkan perintah Latex.

```
>tex(&[1,2,3])
```

```
\left[ 1 , 2 , 3 \right]
```

Faktor dan Tabel

Pada pengantar R terdapat sebuah contoh dengan apa yang disebut faktor.

Berikut ini adalah daftar wilayah dari 30 negara bagian.

```
>austates = ["tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa", ...
> "qld", "vic", "nsw", "vic", "qld", "qld", "sa", "tas", ...
> "sa", "nt", "wa", "vic", "qld", "nsw", "nsw", "wa", ...
> "sa", "act", "nsw", "vic", "vic", "act"];
```

Asumsikan, kita memiliki pendapatan yang sesuai di setiap negara bagian.

```
>incomes = [60, 49, 40, 61, 64, 60, 59, 54, 62, 69, 70, 42, 56, ...
> 61, 61, 61, 58, 51, 48, 65, 49, 49, 41, 48, 52, 46, ...
> 59, 46, 58, 43];
```

Sekarang, kita ingin menghitung rata-rata pendapatan di wilayah tersebut. Sebagai sebuah program statistik, R memiliki fungsi factor() dan tapply() untuk hal ini.

EMT dapat melakukan hal ini dengan mencari indeks dari wilayah-wilayah di dalam daftar unik dari wilayah-wilayah tersebut.

```
>auterr=sort(unique(austates)); f=indexofsorted(auterr,austates)
```

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

Pada titik ini, kita dapat menulis fungsi perulangan kita sendiri untuk melakukan sesuatu untuk satu faktor saja.

Atau kita dapat meniru fungsi tapply() dengan cara berikut.

```
>function map tappl (i; f$:call, cat, x) ...

u=sort(unique(cat));
f=indexof(u,cat);
return f$(x[nonzeros(f==indexof(u,i))]);
endfunction
```

Ini sedikit tidak efisien, karena menghitung wilayah unik untuk setiap i, tetapi berfungsi.

```
>tappl(auterr,"mean",austates,incomes)
```

```
[44.5, 57.333333333, 55.5, 53.6, 55, 60.5, 56, 52.25]
```

Perhatikan bahwa ini bekerja untuk setiap vektor wilayah.

```
>tappl(["act","nsw"],"mean",austates,incomes)
```

```
[44.5, 57.333333333]
```

Sekarang, paket statistik EMT mendefinisikan tabel seperti halnya di R. Fungsi readtable() dan writetable() dapat digunakan untuk input dan output.

Jadi kita dapat mencetak rata-rata pendapatan negara di wilayah dengan cara yang ramah.

```
>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

Kita juga dapat mencoba meniru perilaku R sepenuhnya.

Faktor-faktor tersebut harus disimpan dengan jelas dalam sebuah koleksi dengan jenis dan kategorinya (negara bagian dan wilayah dalam contoh kita). Untuk EMT, kita menambahkan indeks yang telah dihitung sebelumnya.

```
>function makef(t) ...
```

```
## Factor data
## Returns a collection with data t, unique data, indices.
## See: tapply
u=sort(unique(t));
return {{t,u,indexofsorted(u,t)}};
endfunction
```

```
>statef=makef(austates);
```

Sekarang elemen ketiga dari koleksi ini akan berisi indeks.

```
>statef[3]
```

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

Sekarang kita dapat meniru `tapply()` dengan cara berikut. Ini akan mengembalikan sebuah tabel sebagai kumpulan data tabel dan judul kolom.

```
>4function tapply (t:vector,tf,f
: call)... Variable function not found! Error in : 4function tapply(t : vector,tf,f
: call) ... .. ^

Variable function not found!
Error in:
4function 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]);
end;
return ({x,uf});
endfunction
```

Kami tidak menambahkan banyak pemeriksaan tipe di sini. Satu-satunya tindakan pencegahan adalah kategori (faktor) yang tidak memiliki data. Tetapi kita harus memeriksa panjang `t` yang benar dan kebenaran koleksi `tf`.

Tabel ini bisa dicetak sebagai sebuah tabel dengan `writetable()`.

```
>writetable(tapply(incomes,statef,"mean"),wc=7)
```

```
Function tapply not found.
Try list ... to find functions!
Error in:
writetable(tapply(incomes,statef,"mean"),wc=7) ...
^
```

Larik

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Akan lebih mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau sebuah pustaka C untuk ini.

R memiliki lebih dari dua dimensi. Dalam R, larik adalah sebuah vektor dengan sebuah bidang dimensi.

Dalam EMT, sebuah vektor adalah sebuah matriks dengan satu baris. Ini bisa dibuat menjadi sebuah matriks dengan `redim()`.

```
>shortformat; X=redim(1:20,4,5)
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Ekstraksi baris dan kolom, atau sub-matriks, sama seperti di R.

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

2	3
7	8
12	13
17	18

Namun, dalam R dimungkinkan untuk mengatur daftar indeks tertentu dari vektor ke suatu nilai. Hal yang sama juga dapat dilakukan dalam EMT hanya dengan sebuah perulangan.

```
>function setmatrixvalue (M, i, j, v) ...

loop 1 to max(length(i),length(j),length(v))
    M[i{#},j{#}] = v{#};
end;
endfunction
```

Kami mendemonstrasikan hal ini untuk menunjukkan bahwa matriks diteruskan dengan referensi dalam EMT. Jika Anda tidak ingin mengubah matriks asli `M`, Anda perlu menyalinnya dalam fungsi.

```
>setmatrixvalue(X,1:3,3:-1:1,0); X,
```

1	2	0	4
5	0	7	8
0	10	11	12
13	14	15	16
17	18	19	20

Hasil kali luar dalam EMT hanya dapat dilakukan di antara vektor. Hal ini otomatis karena bahasa matriks. Satu vektor harus berupa vektor kolom dan vektor baris.

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

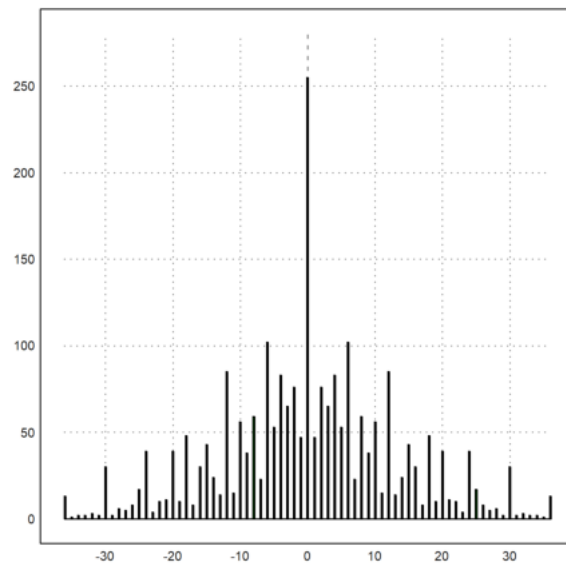
1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Dalam pengantar PDF untuk R ada sebuah contoh, yang menghitung distribusi ab-cd untuk a, b, c, d yang dipilih dari 0 sampai n secara acak. Solusinya dalam R adalah membentuk sebuah matriks 4 dimensi dan menjalankan table() di atasnya.

Tentu saja, ini bisa dicapai dengan sebuah perulangan. Tetapi perulangan tidak efektif dalam EMT atau R. Dalam EMT, kita bisa menulis perulangan dalam C dan itu adalah solusi tercepat.

Tetapi kita ingin meniru perilaku R. Untuk ini, kita perlu meratakan perkalian ab dan membuat sebuah matriks ab-cd.

```
> a=0:6; b=a'; p=flatten(a*b); q=flatten(p-p'); ...
> u=sort(unique(q)); f=getmultiplicities(u,q); ...
> statplot(u,f,"h"):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-046.png

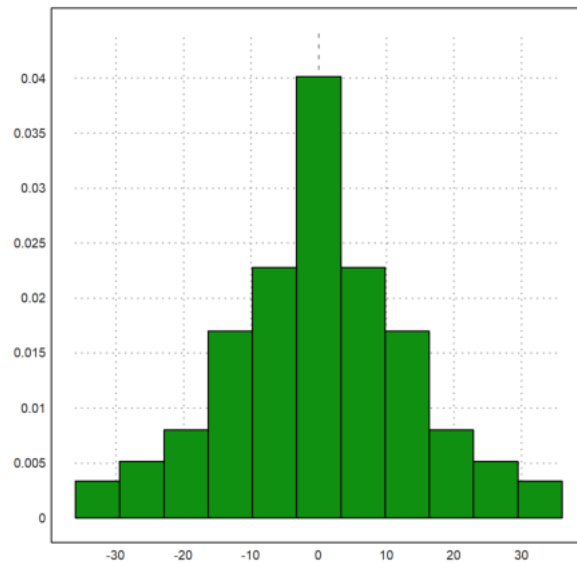
Selain kelipatan yang tepat, EMT dapat menghitung frekuensi dalam vektor.

```
>getfrequencies(q,-50:10:50)
```

```
[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]
```

Cara yang paling mudah untuk memplot ini sebagai distribusi adalah sebagai berikut.

```
>plot2d(q,distribution=11):
```

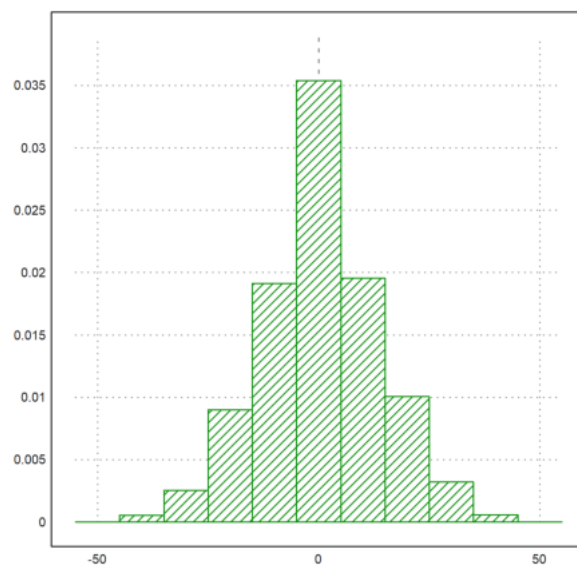


images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-047.png

Namun, Anda juga dapat menghitung terlebih dahulu jumlah dalam interval yang dipilih. Tentu saja, berikut ini menggunakan `getfrequencies()` secara internal.

Karena fungsi `histo()` mengembalikan frekuensi, kita perlu menskalakannya sehingga integral di bawah grafik batang adalah 1.

```
> {x,y}=histo(q,v=-55:10:55); y=y/sum(y)/differences(x); ...
> plot2d(x,y,>bar,style="/");
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-048.png

Daftar

EMT memiliki dua jenis daftar. Satu adalah daftar global yang dapat diubah, dan yang lainnya adalah jenis daftar yang tidak dapat diubah. Kami tidak peduli dengan daftar global di sini.

Jenis daftar yang tidak dapat diubah disebut koleksi dalam EMT. Ia berperilaku seperti struktur dalam C, tetapi elemennya hanya diberi nomor dan tidak diberi nama.

```
> L = { {"Fred", "Flintstone", 40, [1990, 1992]} }
```

```
Fred
Flintstone
40
[1990, 1992]
```

Saat ini unsur-unsur tersebut tidak memiliki nama, meskipun nama dapat ditetapkan untuk tujuan khusus. Unsur-unsur tersebut diakses dengan angka.

```
> (L[4])[2]
```

```
1992
```

Input dan Output File (Membaca dan Menulis Data)

Anda sering kali ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberi tahu Anda tentang berbagai cara untuk mencapainya. Fungsi sederhana adalah `writematrix()` dan `readmatrix()`.

Mari kita tunjukkan cara membaca dan menulis vektor bilangan real ke dalam file.

```
>a=random(1,100); mean(a), dev(a),
```

```
0.49931441472  
0.28488143099
```

Untuk menulis data ke dalam berkas, kami menggunakan fungsi `writematrix()`.

Karena pengantar ini kemungkinan besar berada di dalam direktori, tempat pengguna tidak memiliki akses tulis, kami menulis data ke direktori beranda pengguna. Untuk buku catatan sendiri, ini tidak diperlukan, karena berkas data akan ditulis ke dalam direktori yang sama.

```
>filename="test.dat";
```

Sekarang kita tulis vektor kolom `a` ke dalam berkas. Ini menghasilkan satu angka di setiap baris berkas.

```
>writematrix(a',filename);
```

Untuk membaca data, kita menggunakan `readmatrix()`.

```
>a=readmatrix(filename)';
```

Dan hapus file nya.

```
>fileremove(filename);
```

```
>mean(a), dev(a),
```

```
0.49931441472  
0.28488143099
```

Fungsi `writematrix()` atau `writetable()` dapat dikonfigurasi untuk bahasa lain.

Misalnya, jika Anda memiliki sistem bahasa Indonesia (titik desimal dengan koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (nilai default dipisahkan dengan koma). File berikut "test.csv" akan muncul di folder Anda saat ini.

```
>filename="test.csv"; ...  
> writematrix(random(5,3),file=filename,separator=",");
```

Anda sekarang dapat membuka berkas ini langsung dengan Excel Indonesia.

```
>fileremove(filename);
```

Terkadang kita memiliki string dengan token seperti berikut.

```
>s1:="f m m f m m f f f m f"; ...  
> s2:="f f f m m f f";
```

Untuk menokenisasi ini, kami mendefinisikan vektor token.

```
>tok:=["f","m"]
```

```
f  
m
```

Lalu kita dapat menghitung berapa kali setiap token muncul dalam string, dan memasukkan hasilnya ke dalam tabel.

```
>M:=getmultiplicities(tok,strtokens(s1))_ ...  
> getmultiplicities(tok,strtokens(s2));
```

Tulis tabel dengan tajuk token.

```
>writetable(M,labc=tok,labr=1:2,wc=8)
```

	f	m
1	6	7
2	5	2

Untuk statika, EMT dapat membaca dan menulis tabel.

```
>file="test.dat"; open(file,"w"); ...  
> writeln("A,B,C"); writematrix(random(3,3)); ...  
> close();
```

Berkas akan terlihat seperti ini.

```
>printfile(file)
```

```
A,B,C
0.6831595441645762,0.009067854737289088,0.0009599831335239887
0.1033880029749033,0.4360586470295605,0.4692395068171239
0.7954167121138519,0.124209456405511,0.5834210885645786
```

Fungsi `readtable()` dalam bentuk yang paling sederhana dapat membacanya dan mengembalikan kumpulan nilai dan baris judul.

```
>L=readtable(file,>list);
```

Koleksi ini dapat dicetak dengan `writetable()` ke buku catatan, atau ke berkas.

```
>writetable(L,wc=10,dc=5)
```

```
      A      B      C
0.68316 0.00907 0.00096
0.10339 0.43606 0.46924
0.79542 0.12421 0.58342
```

Matriks nilai adalah elemen pertama L. Perhatikan bahwa `mean()` dalam EMT menghitung nilai rata-rata baris matriks.

```
>mean(L[1])

0.231062460678
0.336228718941
0.501015752361
```

Berkas CSV

Pertama, mari kita tulis matriks ke dalam berkas. Untuk output, kita buat berkas di direktori kerja saat ini.

```
>file="test.csv"; ...
>M=random(3,3); writematrix(M,file);
```

Ini adalah isi dari berkas.

```
>printfile(file)

0.1561768801300948,0.3378085511334831,0.9254100833366891
0.6025983454756715,0.3143385582225292,0.1413753638122724
0.9219935524135139,0.4418918106522429,0.8444689241144148
```

CSV ini dapat dibuka pada sistem bahasa Inggris ke Excel dengan mengklik dua kali. Jika Anda mendapatkan berkas tersebut pada sistem bahasa Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal.

Namun, titik desimal juga merupakan format default untuk EMT. Anda dapat membaca matriks dari berkas dengan `readmatrix()`.

```
>readmatrix(file)

Real 3 x 5 matrix

      0.15617688013      0      ...
      0.602598345476      0      ...
      0.921993552414      0      ...
```

Dimungkinkan untuk menulis beberapa matriks ke dalam satu berkas. Perintah `open()` dapat membuka berkas untuk ditulis dengan parameter “w”. Nilai default untuk membaca adalah “r”.

```
>open(file,"w"); writematrix(M); writematrix(M'); close();
```

Matriks dipisahkan oleh baris kosong. Untuk membaca matriks, buka berkas dan panggil `readmatrix()` beberapa kali.

```
>open(file); A=readmatrix(); B=readmatrix(); A==B, close();
```

```
Real 3 x 5 matrix

      1      1      ...
      0      1      ...
      0      1      ...
```

Di Excel atau lembar kerja serupa, Anda dapat mengeksport matriks sebagai CSV (nilai yang dipisahkan koma). Di Excel 2007, gunakan “simpan sebagai” dan “format lain”, lalu pilih “CSV”. Pastikan, tabel saat ini hanya berisi data yang ingin Anda ekspor.

Berikut ini contohnya.

```
>printfile("excel-data.csv")

0;1000;1000
1;1051,271096;1072,508181
2;1105,170918;1150,273799
3;1161,834243;1233,67806
```

```

4;1221,402758;1323,129812
5;1284,025417;1419,067549
6;1349,858808;1521,961556
7;1419,067549;1632,31622
8;1491,824698;1750,6725
9;1568,312185;1877,610579
10;1648,721271;2013,752707

```

Di Excel atau lembar kerja serupa, Anda dapat mengekspor matriks sebagai CSV (nilai yang dipisahkan koma). Di Excel 2007, gunakan “simpan sebagai” dan “format lain”, lalu pilih “CSV”. Pastikan, tabel saat ini hanya berisi data yang ingin Anda ekspor.

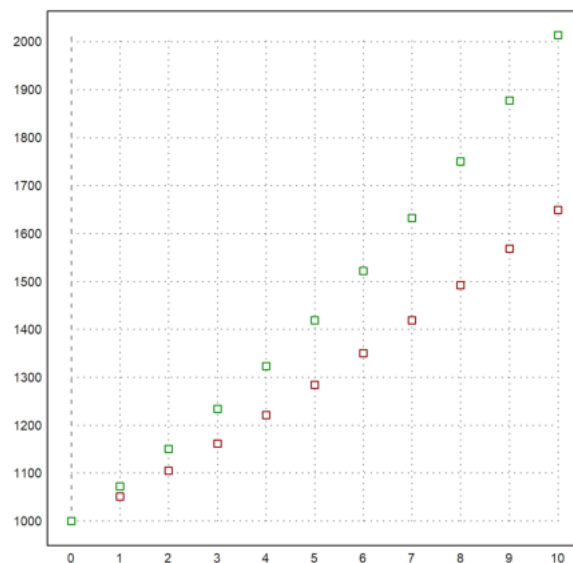
Berikut ini contohnya.

```
>M=readmatrix("excel-data.csv",>comma)
```

	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	

Mari plot.

```
>plot2d(M'[1],M'[2:3],>points,color=[red,green]):
```



images/Alya%20Putri%20Medilasari_23030630018_EMTStatistika-049.png

Ada cara yang lebih mendasar untuk membaca data dari sebuah berkas. Anda dapat membuka berkas dan membaca angka baris demi baris. Fungsi `getvectorline()` akan membaca angka dari sebaris data. Secara default, fungsi ini mengharapkan titik desimal. Namun, fungsi ini juga dapat menggunakan koma desimal, jika Anda memanggil `setdecimaldot(",")` sebelum menggunakan fungsi ini.

Fungsi berikut adalah contohnya. Fungsi ini akan berhenti di akhir berkas atau baris kosong.

```

>function myload (file) ...

open(file);
M=[];
repeat
    until eof();
    v=getvectorline(3);
    if length(v)>0 then M=M_v; else break; endif;
end;
return M;
close(file);
endfunction

```

```
>myload(file)
```

Real 3 x 5 matrix

0.15617688013	0	...
0.602598345476	0	...
0.921993552414	0	...

Semua angka dalam berkas itu juga dapat dibaca dengan `getvector()`.

```
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)

0.15617688013      0      0.337808551133
0      0.925410083337      0.602598345476
0      0.314338558223      0
```

Jadi sangat mudah untuk menyimpan vektor nilai, satu nilai di setiap baris dan membaca kembali vektor ini.

```
>v=random(1000); mean(v)

0.511171163266

>writematrix(v',file); mean(readmatrix(file)')

0.511171163266
```

Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Misalnya, kita menulis tabel dengan tajuk baris dan kolom ke dalam sebuah berkas.

```
>file="test.tab"; M=random(3,3); ...
> open(file,"w"); ...
> writetable(M,separator="," ,labc=["one","two","three"]); ...
> close(); ...
> printfile(file)
```

```
one, two, three
0.08,      0.64,      0.03
0.7,      0.32,      0.27
0.79,      0.64,      0.59
```

Ini dapat diimpor ke Excel.

Untuk membaca berkas di EMT, kami menggunakan `readtable()`.

```
>{M,headings}=readtable(file,>clabs); ...
> writetable(M,labc=headings)
```

```
one      two      three
0.08      0.64      0.03
0.7      0.32      0.27
0.79      0.64      0.59
```

Menganalisis Garis

Anda bahkan dapat mengevaluasi setiap garis secara manual. Misalkan, kita memiliki garis dengan format berikut.

```
>line="2020-11-03,Tue,1'114.05"
```

```
2020-11-03, Tue, 1 ' 114.05
```

Pertama, kita dapat membuat token pada baris tersebut.

```
>vt=strtokens(line)
```

```
2020-11-03
Tue
1'114.05
```

Kemudian kita dapat mengevaluasi setiap elemen garis menggunakan evaluasi yang tepat.

```
>3day(vt[1]), ...
> indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])), ...
> strrepl(vt[3],"","")()
```

```
2214486
2
1114.05
```

Dengan menggunakan ekspresi reguler, dimungkinkan untuk mengekstrak hampir semua informasi dari sebaris data.

Asumsikan kita memiliki baris berikut sebagai dokumen HTML.

```
>line="<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>"
```

```
&lt;tr&gt;&lt;td&gt;1145.45&lt;/td&gt;&lt;td&gt;5.6&lt;/td&gt;&lt;td&gt;-4.5&lt;/td&gt;&lt;/tr&gt;
```



Untuk mengekstraknya, kami menggunakan ekspresi reguler, yang mencari

- tanda kurung tutup >,
- string apa pun yang tidak mengandung tanda kurung dengan sub-kecocokan “(...)”,
- tanda kurung buka dan tutup menggunakan solusi terpendek,
- lagi-lagi string apa pun yang tidak mengandung tanda kurung, dan tanda kurung buka <.

Ekspresi reguler agak sulit dipelajari tetapi sangat ampuh.

```
>{pos,s,vt}=strxfind(line,">([^\>]+)<.+?>([^\>]+)<");
```

Hasilnya adalah posisi kecocokan, string yang cocok, dan vektor string untuk sub-kecocokan.

```
>for k=1:length(vt); vtk, end;
```

```
1145.45
5.6
```

Berikut adalah fungsi yang membaca semua item numerik antara <td> dan </td>.

```
>function readtd (line) ...
```

```
v=[]; cp=0;
repeat
    {pos,s,vt}=strxfind(line,"<td.*?>(.+?)</td>",cp);
    until pos==0;
    if length(vt)>0 then v=v|vt[1]; endif;
    cp=pos+strlen(s);
end;
return v;
endfunction
```

```
>readtd(line+"<td>non-numerical</td>")
```

```
1145.45
5.6
-4.5
non-numerical
```

Membaca dari Web

Situs web atau berkas dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris.

Dalam contoh ini, kami membaca versi terkini dari situs EMT. Kami menggunakan ekspresi reguler untuk memindai “Versi ...” dalam judul.

```
>function readversion () ...
```

```
urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html");
repeat
    until urleof();
    s=urlgetline();
    k=strfind(s,"Version ",1);
    if k>0 then substring(s,k,strfind(s,"<",k)-1), break; endif;
end;
urlclose();
endfunction
```

```
>readversion
```

```
Version 2024-01-12
```

Input dan Output Variabel

Anda dapat menulis variabel dalam bentuk definisi Euler ke dalam file atau ke baris perintah.

```
>writevar(pi,"mypi");
```

```
mypi = 3.141592653589793;
```

Untuk pengujian, kami membuat file Euler di direktori kerja EMT.

```
>file="test.e"; ...
> writevar(random(2,2),"M",file); ...
> printfile(file,3)
```

```
M = [ . .
0.8309086172968301, 0.8503923543185694;
0.05518309524774696, 0.1410652436271988];
```

Sekarang kita dapat memuat berkas tersebut. Berkas tersebut akan mendefinisikan matriks M.

```
>load(file); show M,
```

```
M =
    0.830908617297    0.850392354319
    0.0551830952477    0.141065243627
```

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 = [ ..
    0.8309086172968301, 0.8503923543185694;
    0.05518309524774696, 0.1410652436271988];
inch$ = 0.0254;
```

Kita juga dapat membuka berkas baru atau menambahkannya ke berkas yang sudah ada. Dalam contoh ini, kita menambahkannya ke berkas yang dibuat sebelumnya.

```
>open(file,"a"); ...
> writevar(random(2,2),"M1"); ...
> writevar(random(3,1),"M2"); ...
> close();
```

```
>load(file); show M1; show M2;
```

```
M1 =
    0.735301703448    0.682324166409
    0.368871161879    0.908904536281
M2 =
    0.109140841419
    0.920513676127
    0.492645530438
```

Untuk menghapus berbagai berkas gunakan `fileremove()`.

```
>fileremove(file);
```

Vektor baris dalam sebuah berkas tidak memerlukan koma, jika setiap angka berada di baris baru. Mari kita buat berkas seperti itu, tulis setiap baris satu per satu dengan `writeln()`.

```
>open(file,"w"); writeln("M = ["); ...
> for i=1 to 5; writeln(""+random()); end; ...
> writeln("]"); close(); ...
> printfile(file)
```

```
M = [
0.620171862056
0.396952370992
0.358527250809
0.621862720632
0.544862819981
];
```

```
>load(file); M
```

```
[0.620171862056, 0.396952370992, 0.358527250809, 0.621862720632,
0.544862819981]
```

Latihan

1. Misalkan anda memiliki vektor $x=[2,4,6,8,10]$

a. Buatlah vektor yang menggabungkan vektor x , angka 0 dan vektor x lagi

b. Tentukan apakah setiap elemen vektor x lebih besar dari 5 (hasil logika 1 untuk benar dan 0 untuk salah)

```
>x:=[2,4,6,8,10]; [x,0,x]
```

```
[2, 4, 6, 8, 10, 0, 2, 4, 6, 8, 10]
```

```
>x>5, %*x
```

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

2. Tentukan matriks X dengan elemen-elemen yang berurutan dari 1 hingga 20 dan susunlah elemen tersebut menjadi matriks berukuran 5×4 . Berdasarkan matriks X yang telah dibuat, ganti elemen-elemen dalam baris 1 hingga 3 dan kolom 1 baris 3, kolom 2 baris 2, dan kolom 3 baris 1 dengan angka 0 menggunakan fungsi `sematrixvalue`

```
>shortformat; X=redim (1:20,5,4)
```

```
1      2      3      4
5      6      7      8
9     10     11     12
```

13	14	15	16
17	18	19	20

```
>setmatrixvalue(X,1:3,3:-1:1,0); X,
```

1	2	0	4
5	0	7	8
0	10	11	12
13	14	15	16
17	18	19	20

3. Seorang analis memiliki data penjualan harian selama 5 hari(150,200,250,300,350) yang disimpan dalam bentuk vektor sebagai berikut:

a. mean(rata-rata)

b. deviasi standar

```
>Penjualan=[150,200,250,300,350]; ...
```

```
> mean(Penjualan), dev(Penjualan),
```

```
250
```

```
79.057
```

4. Buat fungsi yang membuka URL “https://en.wikipedia.org/wiki/Euler_(software)” dan mencari kata “Versi” di dalam URL tersebut, dan tampilkan hasilnya

```
>function readversionwebsite()...
```

```
urlopen("https://en.wikipedia.org/wiki/Euler_(software)");
```

```
repeat
```

```
    until urfeof();          s=urlgetline();
```

```
    k=strfind(s,"Version",1);
```

```
    if k>0 then substring(s,k,strfind(s,"<",k)-1), break; endif
```

```
end;
```

```
urlclose();
```

```
endfunction
```

```
>readversionwebsite
```

```
VersionList","mobile","model","platform","platformVersion"],"GEHomepageSugges
{"ext.globalCssJs.user.styles":"ready","site.styles":"ready","user.styles":"r
["ext.cite.ux-
enhancements","site","mediawiki.page.ready","jquery.makeCollapsible","mediawiki
```

