

# APLIKASI KOMPUTER



logo-uny.png

Siti Faltipah Hayani

23030630004

## FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS NEGERI YOGYAKARTA 2024

### Kata Pengantar

Assalamualaikum warahmatullahi wabarakatuh Segala puji bagi Allah SWT yang telah menolong kami menyelesaikan tugas ini dengan penuh kemudahan. Tanpa pertolongan - Nya mungkin penyusun dapat menyelesaikan tugas yang diberikan untuk memenuhi syarat. Shalawat serta salam mari kita curah limpahkan kepada suri tauladan kita baginda tercinta yakni Nabi Muhammad SAW.Semoga dengan adanya tugas ini mampu memberikan jembatan ilmu, walaupun hanya secilil semoga dapat berguna bagi siapa saja. Segala usaha telah kami lakukan untuk menyelesaikan tugas ini, Namun dalam usaha yang maksimal itu kami menyadari tentu masih terdapat kekurangan. Kesempurnaan hanya milik Allah SWT kami hanya dapat melakukan semaksimal mungkin ,untuk itu kami mengharapkan sebuah kritik dan saran dari semua pihak demi penyempurnaan tugas ini.

Yogyakarta, 29 November 2024

Penulis # EMT untuk Perhitungan Aljabar

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

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

### Contoh pertama

Menyederhanakan bentuk aljabar:

$> \$\&5*x^{(-4)}*y^3 - 8*x^5*y^{(-6)}$

$$-\frac{40x}{y^3}$$

Menyederhanakan fungsi :

$> \$\&5*y^{2+3*x_6} - 7*y^{2+2*x_2}$

$$-2y^2 + 3x^6 + 2x^2$$

Menjabarkan:

```
>\$&showev('expand((5*x^{-4}+y3)*(-8*x^5-y^{-6})))  
expand( ( y^3 + 5 x^{-4}) ( -y^{-6} - 8 x^5) ) = -y^{-6}+3 - 5 x^{-4} y^{-6} - 8 x^5 y^3 - 40 x^{-4}+5  
##
```

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

```
37.6991118431
```

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.
```

```
75.3982236862  
150.796447372
```

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

```
>x := 4;  
>x := cos(x) // nilai cosinus (x dalam radian)  
-0.653643620864  
>x := cos(x)  
0.793873449226
```

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

```
>x := 3.5; ...  
>x := (x+2/x)/2, x := (x+2/x)/2,  
2.03571428571  
1.50908521303  
1.41719571011
```

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 cursor 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
```

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

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=6; for i=1 to 10; x := (x+2/x)/2, end; // menghitung akar 2
```

```
3.16666666667  
1.89912280702  
1.47612029496  
1.4155117098  
1.41421415763  
1.41421356237  
1.41421356237  
1.41421356237  
1.41421356237
```

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

```
>x := 2.5; // comments go here before the ...
> repeat xnew:=(x+2/x)/2; until xnew~=x; ...
> x := xnew; ...
> end; ...
> x,
```

1.41421356237

Struktur bersyarat juga berfungsi.

```
>if Epi>piE; then "Halo Rakyatku!", endif;
```

Halo Rakyatku!

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(45°)/cos(60°))
```

1.189207115

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(5.7))
```

5.7

## 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 ruang antara perintah diharapkan untuk ada.

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

```
>g:=10.73; t:=4.2; 1/2*g*t^2
```

94.6386

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk mengetik

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/(9+5*log(0.7))+4/9)
```

4.30791848586

Untuk menghitung ekspresi rumit seperti

Anda harus memasukkannya dalam bentuk baris.

```
>((5/8 + 7/6 + 3) / (3/7 + 5/9))^2 * pi
```

74.4767625442

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.

```
>2/3+5/7, fraction %
```

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)+2)^3*(sin(pi/4)+5)2
```

```
646.172032434
```

Operator numerik Euler meliputi

- unary atau operator plus
- unary atau operator minus
- operator perkalian
  - / operator pecahan
  - . produk matriks
- a<sup>b</sup> daya untuk positif a atau bilangan bulat b (a\*\*b juga berfungsi)
- n! operator faktorial dan masih banyak lagi.

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

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

Beberapa perintah memiliki alias, mis. ln untuk log.

```
>ln(E^4), arctan(tan(0.75)), logbase(30,10)
```

```
4  
0.75  
1.47712125472
```

```
>sin(90°)
```

```
1
```

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(23/3)
```

```
7.6666666666666667
```

Representasi ganda internal membutuhkan 8 byte.

Representasi ganda adalah format penyimpanan untuk floating-point yang menggunakan 64 bit(8 byte)

```
>printdual(23/3)
```

```
1.110101010101010101010101010101010101010101010101011*2^2
```

```
>printhex(1/7)
```

```
2.4924924924924*16^-1
```

Perbedaan ‘printdual’ dan ‘printhex’ adalah ‘printdual’ yakni mencetak representasi internal dari sebuah bilangan floating-point dalam format presisi ganda (pendekatan yang sangat dekat dengan nilai aslinya tetapi tidak persis sama.) meskipun ia tergantung pada konteks bahasa pemrograman tertentu. sedangkan ‘printhex’ yakni representasi dari nilai floating-point dalam bentuk heksadesimal(basis 16), heksadesimal ini adalah cara yang lebih ringkas untuk menampilkan nilai biner karena setiap digit heksadesimal mempresentasikan empat digit biner.

## String

Sebuah string dalam Euler didefinisikan dengan “...”

>“A string can contain anything.”

A string can contain anything.

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

>“Terjadi Gempa Mag pada hari Senin 26 Agustus 2024 dengan pusat gempa berada di laut”+95+ ” km barat daya Gunungkidul.”

Terjadi Gempa Mag pada hari Senin 26 Agustus 2024 dengan pusat gempa berada di laut 95 km barat daya Gunungkidul.

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

>“Golden Ratio :” + print((1+sqrt(5))/2,5,0)

Golden Ratio : 1.61803

Terdapat spesial string ‘none’, yang tidak dicetak.

>none

Untuk mengonversi string menjadi angka, cukup mengevaluasinya. Ini bekerja untuk ekspresi juga (lihat dibawah).

>“1234.5567”()

1234.5567

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

>v:= [“Indonesia”,“Malaysia”,“Brunei Darussalam”]

```
Indonesia
Malaysia
Brunei Darussalam
```

Vektor pada string kosong dilambangkan dengan [none]. Dan vektor string dapat digabungkan dengan ‘|’.

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

```
Indonesia
Malaysia
Brunei Darussalam
Indonesia
Malaysia
Brunei Darussalam
```

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

>u”β = ” + 90 + u”° ” // pdfLaTeX mungkin gagal menampilkan secara benar

β = 90°

Dalam komentar, entitas yang sama seperti alpha; beta; dll dapat digunakan untuk lateks.

Ada beberapa fungsi untuk membuat atau menganalisis string unicode.

Fungsi strtochs() 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”Ä“)[1]; chartoutf(v)

Ä is a German letter

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

```
>a="We have  $\alpha=\beta$ "; utf(a)// PdfLaTeX mungkin gagal menampilkanya
```

```
We have  $\alpha=\beta$ .
```

Memungkinkan juga untuk menggunakan entitas numerik.

```
>u"Älphabet"
```

```
Älphabet
```

## Nilai Boolean

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

```
>"saya">"aku", 6==3
```

```
1  
0
```

```
>5>1, "mobil"=="motor"
```

```
1  
0
```

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

```
>2<E || E<3
```

```
1
```

```
>6>E && E<2
```

```
0
```

Operator Boolean mematuhi aturan bahasa matriks

```
>(2:9)>3, nonzeros (%)
```

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

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

```
>N=5 | 7:2:50 // N berisi elemen 5 dan bilangan bilangan ganjil dari 7:50
```

```
[5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33,  
35, 37, 39, 41, 43, 45, 47, 49]
```

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

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

## Output Formats

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

```
>defformat; pi
```

```
3.1415926535
```

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
```

```
>longestformat; pi
```

```
3.141592653589793
```

Berikut ini adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhex(pi)
```

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

Heksadesimal adalah sistem bilangan yang menggunakan basis 16. Di mana angka 0 hingga 9 (untuk mewakili nilai 0 hingga 9) dan huruf A hingga F (untuk mewakili nilai 10 hingga 15).

Format standarnya adalah 12.

```
>format(12); 1/7
```

```
0.142857142857
```

Format output dapat diubah secara permanen dengan perintah format.

```
>format(12,5); 1/9, pi, cos(1)
```

```
0.11111  
3.14159  
0.54030
```

Format standar untuk skalar adalah 12, tetapi ini dapat diubah.

```
>setscalarformat(7); pi
```

```
3.14159
```

Begitu juga dengan fungsi “longestformat” mengatur format skalar.

```
>longestformat; pi
```

```
3.141592653589793
```

Notes: beberapa format output yang penting.

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

Akurasi internal EMT adalah sekitar 16 digit desimal mengikuti standar dari IEEE. Angka disimpan dalam format internal. Namun, format output EMT dapat diatur secara fleksibel.

```
>fraction 2+3/10+7/14+21/7
```

```
29/5
```

```
>fraction 5*7/2/3*2/5
```

```
7/3
```

Digunakan untuk menampilkan ke bentuk pecahan sederhana.

```
>longest 0.2+0.25+0.3+0.3+0.25+0.2+0.5-2.6
```

```
-0.6000000000000001
```

Perintah ini menunjukkan presisi penuh dari operasi aritmetika yang melibatkan angka-angka kecil, dan bagaimana kesalahan akumulatif bisa muncul dalam perhitungan biner.

## Expressions

String atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang dapat dievaluasi oleh EMT. Gunakan tanda kurung setelah ekspresi.

```
>k:=5; fx:="pi*k"; fx()
```

```
15.70796326794897
```

Ekspresi akan selalu menggunakan global variable, bahkan jika ada variabel dalam fungsi dengan nama yang sama.

```
>fx:="a*cos(x)"; fx(10,a=0.8)
```

```
-0.671257223261162
```

Menggunakan parameter yang ditetapkan ke x, y, z, dll. Jika tidak, evaluasi ekspresi dalam fungsi dapat memberikan hasil yang membingungkan bagi pengguna yang memanggil fungsi tersebut.

```
>at:=6; function f(expr,x,at):=expr(x); ...  
> f("at*x^4",2,3) // computes 6*2^4 not 3*2^4
```

96

Menggunakan global variable pada fungsi, dimana “at” merupakan global variables. Jika ingin menggunakan nilai lain untuk “at” perlu menambahkan “at=value”.

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

128

Walaupun “ut” sebagai global variable sudah didefinisikan, tetapi didefinisikan kembali pada ekspresi fungsinya dimana “ut=p” sehingga nilainya berganti dari yang awalnya ut=5 menjadi ut=p=2.

```

>f &= x^2
      2
     x

>function f(x) := x^4
>f(2)
16

Ekspresi dalam x sering digunakan seperti fungsi.

Mendefinisikan fungsi dengan nama yang sama seperti ekspresi simbolik global (&=) menghapus nilai variabel sebelumnya untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

> "@(a,b) a^3+b^2", %(2,4)
@(a,b) a^3+b^2
24

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

>fx &= 2*x-3*t; ...
>t=2.5; fx(0.8)
-5.9

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

>fx(1,t=1.5)
-2.5

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



## Symbolic Mathematics



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

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

>$&35!
10333147966386144929666651337523200000000
Dengan cara ini, kita dapat menghitung hasil yang besar dengan tepat.

Mari kita hitung!

lateks: C(35,15)=
> $& 35!/(20!*15!) // nilai C(35,15)
3247943160
Maxima memiliki fungsi yang lebih efisien untuk ini (seperti halnya bagian numerik dari EMT).

>$binomial(35,15) //menghitung C(35,15) menggunakan fungsi binomial()
3247943160
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.

>$&binomial(x,3) with x=5 // substitusi x=5 ke C(x,3)
10
Dengan begitu kita dapat menggunakan solusi persamaan dalam persamaan lain.

>sol &= solve(x^2+3*x=9,x); $&sol, sol(), $&float(sol)

[ x =  $\frac{-3\sqrt{5} - 3}{2}$ , x =  $\frac{3\sqrt{5} - 3}{2}$  ]
[-4.854101966249685, 1.854101966249685]

```

```
[x = - 4.854101966249685, x = 1.854101966249685]
```

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

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

```
>\$&solve(x^2+3*x=1,x), x2 &= x with %[2]; \$&x2
```

$$[x = \frac{-\sqrt{13} - 3}{2}, x = \frac{\sqrt{13} - 3}{2}]$$
$$\frac{\sqrt{13} - 3}{2}$$

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

```
>sol &= solve([x+y=5,x^2+y=15],[x,y]); \$&sol, \$&x*y with sol[1]
```

$$[[x = \frac{5 - \sqrt{5}}{2}, y = \frac{\sqrt{5} + 5}{2}], [x = \frac{\sqrt{5} + 5}{2}, y = \frac{5 - \sqrt{5}}{2}]]$$
$$\frac{(5 - \sqrt{5})(\sqrt{5} + 5)}{4}$$

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

## Functions

Dalam matematika, fungsi aljabar adalah fungsi yang bisa didefinisikan sebagai akar dari sebuah persamaan aljabar. Fungsi aljabar merupakan ekspresi aljabar menggunakan sejumlah suku terbatas, yang melibatkan operasi aljabar seperti penambahan, pengurangan, perkalian, pembagian, dan peningkatan menjadi pangkat pecahan.

contohnya:

```
>\$& f(x)=1/x
```

$$(x^2)(x) = \frac{1}{x}$$

```
>\$& f(x)=sqrt(x)
```

$$(x^2)(x) = \sqrt{x}$$

```
>\$& f(x)=(sqrt(1+x^3))/(x^(3/7)-(sqrt(7)*x^(1/3)))
```

$$(x^2)(x) = \frac{\sqrt{x^3 + 1}}{x^{3/7} - \sqrt{7}x^{1/3}}$$

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)
```

Lalu, semua fungsi pasti dapat didefinisikan oleh fungsi satu baris. Suatu fungsi dapat dievaluasi, contohnya kita akan mencari nilai f(5) dari fungsi f diatas.

```
>f(2)
```

```
4.47213595499958
```

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

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

```
[0, 0.1004987562112089, 0.2039607805437114, 0.3132091952673166,
0.4308131845707604, 0.5590169943749475, 0.699714227381436,
0.8544588931013591, 1.024499877989256, 1.210826164236634,
1.414213562373095]
```

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)
```

0.7861513777574233

Secara default, jika ingin menimpa fungsi bawaan bisa menambahkan kata kunci “overwrite”. Menimpa fungsi bawaan berbahaya dan menyebabkan masalah bagi fungsi lain yang bergantung pada fungsi tersebut.

Jika ingin memanggil fungsi bawaan bisa memakai “\_”, jika fungsi tersebut merupakan fungsi inti Euler.

```
>function overwrite sin(x) := _sin(x°) // tentukan kembali sinus dalam derajat
```

```
>sin(45)
```

0.7071067811865476

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

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

0.7071067811865476

## 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

Menimpa default value.

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

80

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

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

16

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

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

```
>a=6; f(2)
```

24

Tetapi parameter yang ditetapkan menimpa global value.

Jika argumen tidak ada dalam daftar parameter yang telah ditentukan sebelumnya, argumen tersebut harus dideklarasikan dengan “:=”!

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

20

Fungsi simbolis didefinisikan dengan “\$&=” . Fungsi simbolis didefinisikan dalam Euler dan maxima,dan bekerja keduaanya. Ekspresi yang mendefinisikan dijalankan melalui Maxima sebelum di definisi.

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

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

Fungsi simbolik dapat digunakan dalam ekspresi simbolik

```
>$&diff(g(x),x), $&% with x=4/3 //1. turunan pertama dari g(x), 2. memasukkan nilai x=4/3
```

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

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

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

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

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

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

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

>solve(&g(x),0.5)

$$0.7034674224983917$$

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)

$$0.7034674224983917$$

Dengan &= fungsinya simbolis, dan dapat digunakan dalam ekspresi simbolik lainnya. Contohnya dalam integral tak tentu sebagai berikut.

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

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

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

$$(x + 2)^n$$

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

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

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

>P(3,4)

625

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

>function f(x) &= x^3-2; \$&f(x)

$$x^3 - 2$$

Dengan &= maka fungsi adalah simbolik, dan dapat digunakan di ekspresi simbolik lainnya.

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

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

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

```
//latex: f(x) = _1^x t^t , dt,
yang tidak dapat dinilai secara simbolis.
```

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

```
>function map f(x) := integrate (“x^x”,1,x)
>f(0:0.5:2)
[-0.7834305107120823, -0.4108156482543905, 0, 0.6768632787990813,
2.050446234534731]
```

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), $&f(x,y)
b^2 - a b + b + a^2
y^2 - x y + y + x^2
```

Fungsi simbolik seperti itu dapat digunakan untuk variabel simbolik.

tetapi 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), $&lapl(% ,x,y)
y^4 - 6 x^2 y^2 + x^4
0
```

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

```
>function f(x,y) &= factor(lapl((x+y)^5,x,y)); $&f(x,y)
10 (y^2 + x)^3 (9 y^2 + x + 2)
```

Ringkasam:

- &= mendefinisikan fungsi simbolis,
- := mendefinisikan fungsi numerik,
- &&= mendefinisikan fungsi simbolis murni

## Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolis.

Untuk menyelesaikan ekspresi sederhana dari satu variabel, kita dapat menggunakan fungsi solve(). Perlu

nilai awal untuk memulai pencarian. Secara internal, solve() menggunakan metode secant.

```
>solve(“x^2-2”,1)
```

1.414213562373095

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

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

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

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

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

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

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

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

$$[[x = \frac{-\sqrt{a^2 e^2 + (4bc - 2ad)e + b^2 d^2} - ae + bd}{2b}, y = \frac{a \sqrt{a^2 e^2 - 2abd + 4bce + b^2 d^2} + a^2 e - abd + 2bc}{2b^2}], [$$

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.

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

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

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

0.966715594850973  
2.00000000000001

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

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

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

$$\frac{(\sqrt{5} - 1)^2}{4}$$
  
0

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

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

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

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

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 lainnya adalah parameter titik koma).

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

-0.7102421508578388

Ini juga bekerja dengan ekspresi. Tapi daftar elemen yang ada harus digunakan.

```
>solve({{"x^2+a*x",a=3}},2,y=0.1)  
0.03297097167558916
```

## 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.

Eliminasi Fourier adalah analog dari eliminasi Gauss untuk linear (persamaan atau pertidaksamaan). Panggilan fungsi `fourier_elim([eq1, eq2, ...], [var1, var2, ...])'` melakukan eliminasi Fourier eliminasi pada pertidaksamaan linear`[eq1, eq2, ...]'` dengan berkenaan dengan variabel `[var1, var2, ...]'`; sebagai contoh

```
>&load(fourier_elim)  
  
C:/Program Files/Euler x64/maxima/share/maxima/5.35.1/share/f\  
ourier_elim/fourier_elim.lisp  
  
>$&fourier_elim([y-x < 5, x - y < 7, 10 < y],[x,y])  
[y - 5 < x, x < y + 7, 10 < y]  
  
>$&fourier_elim([x^2 - 1>0],[x])  
[1 < x] ∨ [x < -1]  
  
>$&fourier_elim([x^2 - 4<0],[x])  
[-2 < x, x < 2]  
  
>$&fourier_elim([x^2 - 9# 0],[x])  
[-3 < x, x < 3] ∨ [3 < x] ∨ [x < -3]  
  
>$&fourier_elim([x # 10],[x])  
[x < 10] ∨ [10 < x]
```

Ketika himpunan penyelesaiannya adalah kosong maka ‘emptyset’, dan ketika himpunan penyelesaiannya adalah semua bilangan real, maka ‘universalset’; sebagai contoh

```
>$&fourier_elim([minf < x, x < inf],[x])  
universalset  
  
>$&fourier_elim([x < 1, x > 1],[x])  
emptyset
```

Untuk persamaan nonlinier, ‘`fourier_elim`’ mengembalikan sebuah daftar persamaan yang disederhanakan:

```
>$&fourier_elim([x^3 - 8 > 0],[x])  
[2 < x, x^2 + 2 x + 4 > 0] ∨ [x < 2, -x^2 - 2 x - 4 > 0]  
  
>$&fourier_elim([cos(x) < 1/2],[x])  
[1 - 2 cos x > 0]
```

Alih-alih sebuah daftar pertidaksamaan, ‘`fourier_elim`’ juga dapat berupa disjungsi atau konjungsi logika:

```
>$&fourier_elim((x + y < 5) and (x - y > 8),[x,y])  
[y + 8 < x, x < 5 - y, y < - $\frac{3}{2}$ ]  
  
>$&fourier_elim([y-x < 5, x - y < 7, 10 < y],[x,y])  
[y - 5 < x, x < y + 7, 10 < y]  
  
>$&fourier_elim(((x + y < 5) and x < 1) or (x - y > 8),[x,y])  
[y + 8 < x] ∨ [x < min(1, 5 - y)]
```

Fungsi `fourier_elim` mendukung operator pertidaksamaan `<`, `<=`, `>`, `>=`, `#`, dan `=`.

Kode eliminasi Fourier memiliki sebuah preprocessor yang mengubah beberapa persamaan nonlinier yang melibatkan nilai absolut, minimum, dan fungsi maksimum menjadi linear dalam persamaan. Selain itu, preprocessor menangani beberapa ekspresi yang merupakan hasil kali atau hasil bagi dari suku-suku linier:

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

```
[6 < x, x < 8, y < -11] ∨ [8 < x, y < -11] ∨ [x < 8, 13 < y] ∨ [x = y, 13 < y] ∨ [8 < x, x < y, 13 < y] ∨ [y < x, 13 < y]
>&fourier_elim([(x+2)/(x-4) <= 2],[x])
```

```
[x = 10] ∨ [10 < x] ∨ [x < 4]
```

# Bahasa Matriks

Dalam matematika, matriks adalah susunan[1] bilangan, simbol, atau ekspresi yang disusun dalam baris dan kolom sehingga membentuk suatu bangun persegi

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

Matriks 1x2

```
>a=[1;2]
```

```
1  
2
```

```
>b=[3,4;5,6]
```

```
3                  4  
5                  6
```

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

Real 3 x 3 matrix

```
1        ...  
4        ...  
7        ...
```

Transpose matriks adalah matriks baru yang diperoleh dengan cara menukar elemen-elemen baris menjadi elemen kolom atau sebaliknya.

```
>a'
```

```
[1, 2]
```

```
>b'
```

```
3                  5  
4                  6
```

```
>c'
```

Real 3 x 3 matrix

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

Invers matriks adalah matriks baru yang merupakan kebalikan dari matriks asal

```
>inv(b)
```

```
-2.99999999999997      1.99999999999999  
2.49999999999998      -1.49999999999999
```

Perkalian matriks sendiri adalah proses mengalikan setiap elemen baris pada matriks pertama dengan elemen kolom pada matriks kedua.

```
>b.a
```

```
11  
17
```

Perkalian dari matriks dengan invers matriks itu sendiri akan menghasilkan matriks identitas

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

```
0.999999999999982      1.77635683940025e-15  
0                              1
```

Perkalian matriks dan perpangkatan matriks

```
>b.b
```

```
29                  36  
45                  56
```

```
>b^2
```

```
9                  16  
25                  36
```

```
>b.b.b
```

```
267      332  
415      516
```

>power(b,3)

```
267      332  
415      516
```

Pembagian matriks

>a/a

```
1  
1
```

>a/b

```
0.3333333333333333 0.25  
0.4      0.3333333333333333
```

Perkalian invers matriks dengan matriks lainnya

>b\|a

```
0.9999999999999993  
-0.4999999999999994
```

>inv(b).a

```
1  
-0.4999999999999996
```

Perkalian skalar

>b\*2

```
6      8  
10     12
```

>2\*b

```
6      8  
10     12
```

>[1,2]\*2

```
[2, 4]  
## 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

Real 2 x 3 matrix

```
1      ...  
1      ...
```

>A=random(3,4)

Real 3 x 4 matrix

```
0.6554163483485531 ...  
0.5250003829714993 ...  
0.2826898577756425 ...
```

>A|1

Real 3 x 5 matrix

```
0.6554163483485531 ...  
0.5250003829714993 ...  
0.2826898577756425 ...
```

>[v,v]

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

>[v;v]

Real 2 x 3 matrix

```
1      ...  
1      ...
```

>[v',v']

```
1      1  
2      2
```

```

3           3
>“[x,x^2]”(v')
1           1
2           4
3           9

>length(2:10)
9

>ones(2,2)
1           1
1           1

>zeros(2,2)
0           0
0           0

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

>random(1,2)
[0.217693385437102, 0.4453627564273003]

```

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

```

>redim(1:9,3,3)
Real 3 x 3 matrix

1     ...
4     ...
7     ...

>function rep(v,n) := redim(dup(v,n),1,n*cols(v))
>rep(1:3,5)
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
>multdup(1:3,3)
[1, 1, 1, 2, 2, 3, 3, 3]

```

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

Keduanya tidak memiliki jumlah kolom yang sama, kolom yang lebih pendek akan diisi dengan 0.

```

>flipx(1:5)
[5, 4, 3, 2, 1]
>rotleft(1:5)
[2, 3, 4, 5, 1]
>rotright(1:5)
[5, 1, 2, 3, 4]

```

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

```

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

```

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

```

>A=id(3)
Real 3 x 3 matrix

1     ...
0     ...
0     ...

```

#3 Vektorisasi

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

```
>sqrt(1:4)
[1, 1.414213562373095, 1.732050807568877, 2]
```

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

```
>x=3:0.05:6; y=log(x)
[1.09861228866811, 1.11514159061932, 1.1314021114911,
1.147402452837541, 1.163150809805681, 1.178654996341646,
1.193922468472434, 1.208960345836975, 1.223775431622115,
1.238374231043268, 1.252762968495367, 1.266947603487324,
1.280933845462064, 1.294727167594399, 1.308332819650178,
1.321755839982319, 1.335001066732339, 1.348073148299692,
1.3609765531356, 1.37371557891303, 1.38629436111989,
1.39871688118447, 1.410986973710261, 1.423108334242606,
1.435084525289322, 1.446918982936324, 1.458615022699516,
1.470175845100592, 1.481604540924214, 1.492904096178148,
1.504077396776273, 1.515127232962858, 1.526056303495048,
1.536867219599264, 1.547562508716012, 1.558144618046549,
1.568615917913844, 1.57897870494939, 1.589235205116579,
1.599387576580598, 1.609437912434099, 1.619388243287267,
1.629240539730279, 1.638996714675643, 1.64865862558738,
1.658228076603531, 1.667706820558075, 1.677096560907914,
1.686398953570227, 1.695615608675151, 1.704748092238424,
1.713797927758341, 1.722766597741102, 1.731655545158348,
1.740466174840503, 1.749199854809257, 1.757857917552372,
1.766441661243763, 1.774952350911672, 1.783391219557537,
... ]
```

Dengan ini dan operator titik dua `a:delta:b`, vektor nilai fungsi dapat dihasilkan dengan mudah. Pada contoh berikut, kita membangkitkan vektor nilai `t[i]` dengan spasi 0,1 dari -1 hingga 3. Kemudian kita membangkitkan vektor nilai fungsi.

lateks:`s=t^3-t`

```
>t=-1:0.1:3; s=t^3-t
[0, 0.1709999999999999, 0.2879999999999999, 0.357, 0.384, 0.375,
0.3360000000000001, 0.2730000000000001, 0.1920000000000001,
0.0990000000000014, 1.387778780781446e-16, -0.0989999999999987,
-0.1919999999999999, -0.2729999999999999, -0.336, -0.375, -0.384,
-0.3570000000000001, -0.2880000000000001, -0.1710000000000003,
-4.440892098500626e-16, 0.2309999999999997, 0.527999999999998,
0.897, 1.3440000000000001, 1.8750000000000001, 2.496000000000002,
3.213000000000004, 4.032000000000004, 4.959000000000006,
6.000000000000005, 7.161000000000006, 8.448000000000008,
9.867000000000001, 11.42400000000001, 13.12500000000001,
14.976000000000002, 16.98300000000003, 19.15200000000003,
21.48900000000003, 24.00000000000004]
```

```
>shortest (1:7)*(1:7)'
```

|   |    |    |    |    |    |    |
|---|----|----|----|----|----|----|
| 1 | 2  | 3  | 4  | 5  | 6  | 7  |
| 2 | 4  | 6  | 8  | 10 | 12 | 14 |
| 3 | 6  | 9  | 12 | 15 | 18 | 21 |
| 4 | 8  | 12 | 16 | 20 | 24 | 28 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 |

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

```
>(1:7).(1:7)'
```

140

Secara default, vektor baris dicetak dalam format yang ringkas.

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

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

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

```
>v:=[2,3]; v.v', %^2
```

13  
169

Untuk mentranspos matriks kita menggunakan apostrof

```
>v=3:6; v'
```

```
3  
4  
5  
6
```

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

```
50
```

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

```
>v'.v
```

```
Real 4 x 4 matrix
```

```
9      ...  
12     ...  
15     ...  
18     ...
```

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

```
>v.v'
```

```
86
```

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

```
>norm(v)^3
```

```
797.5311906126306
```

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ringkasan aturannya.

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

Misalnya, nilai skalar kali vektor mengalikan nilai dengan setiap elemen vektor. Atau matriks kali vektor (dengan \*, bukan.) memperluas vektor ke ukuran matriks dengan menduplikasikan.

Berikut ini adalah kasus sederhana dengan operator^.

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

```
[4, 9, 36]
```

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

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

```
Real 3 x 3 matrix
```

```
4      ...  
6      ...  
12     ...
```

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

```
>v.v'
```

```
49
```

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)
```

```
[4, 9, 36]
```

```
49
```

```
[4, 13, 49]
```

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

```

>3:9, 0:2:8
[3, 4, 5, 6, 7, 8, 9]
[0, 2, 4, 6, 8]
>[1,2][[3,4,5], [1,2]_5
[1, 2, 3, 4, 5]
      1           2
      5           5

```

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

```

>A:=[4,5,6,7;8,9,10,11]; A[2,2]
9

```

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

```

>v:=[1,3,5,7]; v[1], A[1]
1
[4, 5, 6, 7]

```

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

```

>v[2:4], A[:,2]
[3, 5, 7]
      5
      9

```

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

```

>A[2:4]
Real 2 x 3 matrix

```

```

      5     ...
      9     ...

```

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

```

>A{5}
8

```

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

```

>redim(A,1,prod(size(A))), flatten(A)
[4, 5, 6, 7, 8, 9, 10, 11]
[4, 5, 6, 7, 8, 9, 10, 11]

```

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.

```

>deformat; 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 sekali  $\square$  gus.

Dalam contoh berikut, kita menghitung  $t[j]^i$  untuk  $i$  dari 1 hingga  $n$ . Kami mendapatkan matriks, di mana

setiap baris adalah tabel  $t^i$  untuk satu  $i$ . Yaitu, matriks memiliki elemen lateks:  $a_{\{i,j\}} = t_{j}^{i}$ ,

,  $i \in$

Fungsi yang tidak berfungsi untuk input vektor harus “divektorkan”. Ini dapat dicapai dengan kata kunci

“peta” dalam definisi fungsi. Kemudian fungsi tersebut akan dievaluasi untuk setiap elemen dari parameter vektor.

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

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

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

```
>f([1:3])
```

```
[0, 2.05045, 13.7251]
```

## Sub-Matriks dan Matriks-Elemen

Untuk mengakses elemen matriks, gunakan notasi braket.

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

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 6 |   |   |

Kita dapat mengakses satu baris matriks yang lengkap.

```
>A[1]
```

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

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

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

```
2
```

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

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

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

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

|   |   |   |
|---|---|---|
| 4 | 5 | 6 |
| 7 | 8 | 9 |

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

|   |   |
|---|---|
| 2 | 3 |
| 5 | 6 |

Untuk singkatan “:” menunjukkan semua indeks baris atau kolom.

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

|   |
|---|
| 2 |
| 5 |
| 8 |

Atau, biarkan indeks pertama kosong.

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

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Kita juga bisa mendapatkan baris terakhir dari A.

>A[3]

[7, 8, 9]

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

>A[2,3]=9

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 9 |
| 7 | 8 | 9 |

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

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

|   |   |   |
|---|---|---|
| 4 | 5 | 3 |
| 6 | 7 | 9 |
| 7 | 8 | 9 |

Selain itu, beberapa jalan diperbolehkan.

>A[1:2,1:2]=-1

|    |    |   |
|----|----|---|
| -1 | -1 | 3 |
| -1 | -1 | 9 |
| 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[5]

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

## Menyortir dan Mengacak

Fungsi sort() mengurutkan vektor baris.

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

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

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

Mari kita mengacak vektor.

>v=shuffle(1:8)

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

Indeks berisi urutan yang tepat dari v.

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

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

>s=[“d”,“f”,“c”,“b”,“aa”,“g”]

d  
f  
c  
b  
aa  
g

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

aa  
b  
c  
d  
f  
g

endfunction

## 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 Aenggunaan 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=[5,6;7,8]; b= [4;3]; A\b
```

```
-7  
6.5
```

Untuk contoh lain, kami membuat matriks 100x100 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(100,100); b=sum(A); longest totalmax(abs(inv(A).b-1))
```

```
4.585221091701897e-14
```

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

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

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

Determinan matriks ini adalah ()

```
> det(A)
```

```
-34
```

## 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 menggunakan dalam ekspresi simbolis. Bentuk [...] biasa untuk mendefinisikan matriks dapat digunakan di Euler untuk mendefinisikan matriks simbolik.

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

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

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

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

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

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

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

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

dengan

`det(A)` menghitung determinan matriks A.

`factor(% )` memberikan faktor-faktor dari determinan.  
`invert(A)` memberikan invers dari matriks A.

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

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

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

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

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

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

Untuk mengekstrak vektor eigen tertentu perlu pengindeksan yang cermat.

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

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

$$[1, - 1]$$

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

`> A(a=6,b=7)`

$$\begin{matrix} 1 & 6 \\ 7 & 2 \end{matrix}$$

Dalam ekspresi simbolik, gunakan dengan.

`> $&A with [a=6,b=7]`

$$\begin{pmatrix} 1 & 6 \\ 7 & 2 \end{pmatrix}$$

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

`> $&A[1]`

$$[1, a]$$

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

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

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

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

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

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

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

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

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

`> $&invert(B)()`

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

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

Perhatikan, bahwa dengan `&:=` matriks B telah didefinisikan sebagai simbolik dalam ekspresi simbolik dan sebagai numerik dalam ekspresi numerik. Jadi kita bisa menggunakanya di sini.

`> longest B.xinv(B)`

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

Misalnya. nilai eigen dari A dapat dihitung secara numerik.

```
>A=[1,2,3;4,5,6;7,8,9]; real(eigenvalues(A))
[16.1168, -1.11684, 0]
```

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

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

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

# 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]
```

$$\begin{array}{cc} 1 & 3.14159 \\ 4 & 5 \end{array}$$

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

```
>$&A
```

$$\left(\begin{array}{cc} 1 & \frac{1146408}{364913} \\ 4 & 5 \end{array}\right)$$

Untuk menghindarinya, ada fungsi “mxmset(variable)”.

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

$$\left(\begin{array}{cc} 1 & 3.141592653589793 \\ 4 & 5 \end{array}\right)$$

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

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

$$1.4142135623730950488016887242097_B \times 10^0$$

$$1.414213562373095$$

Ketepatan angka floating point besar dapat diubah

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

$$3.14159265358979323846264338327950288419716939937510582097494\backslash$$

$$4592307816406286208998628034825342117068b0$$

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 4000 (katakanlah dalam dolar).

```
>M=4000
```

$$4000$$

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

```
>M*1.03
```

$$4120$$

Euler akan memahami sintaks berikut juga.

```
>M+M*3%
```

4120

Tetapi lebih mudah menggunakan faktornya.

>q=1+3%, M\*q

1.03

4120

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

>M\*q^10

5375.66551738

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

>format(12,2); M\*q^10

5375.67

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

>“Starting from” + M + “\$ you get” + round(M\*q^10,2) + “.”

Starting from 4000\$ you get 5375.67\$.

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

>M\*q^(0:10)

Real 1 x 11 matrix

4000.00 4120.00 4243.60 4370.91 ...

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.

>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 M, dan kami mendapatkan vektor nilai.

>VM=M\*q^(0:10)

Real 1 x 11 matrix

4000.00 4120.00 4243.60 4370.91 ...

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

>function oneyear(M) := round(M \* q, 2)

Mari kita bandingkan dua hasil, dengan dan tanpa pembulatan.

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

1271.61  
1271.6071

Sekarang tidak ada rumus sederhana untuk tahun ke-n, dan kita harus mengulang selama bertahun-tahun.

Euler memberikan banyak solusi untuk ini.

Cara termudah adalah iterasi fungsi, yang mengulangi fungsi tertentu beberapa kali.

>VMr=iterate(“oneyear”,4000,10)

Real 1 x 11 matrix

4000.00 4120.00 4243.60 4370.91 ...

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

>VMr'

```

4000.00
4120.00
4243.60
4370.91
4502.04
4637.10
4776.21
4919.50
5067.09
5219.10
5375.67

```

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

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

```

4120.00
4000.00    4120.00    4243.60

```

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.

```
>VMr[-2], VM[-2]
```

```

5219.10
5219.09

```

Perbedaannya sangat kecil.

## Memecahkan Persamaan

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

```
>function onepay (M) := M*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",4000,10)
```

```
Real 1 x 11 matrix
```

```

4000.00    4320.00    4649.60    4989.09    ...

```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

```
>R=-200; iterate("onepay",4000,10)
```

```
Real 1 x 11 matrix
```

```

4000.00    3920.00    3837.60    3752.73    ...

```

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.

```
>VMR=iterate("onepay",4000,50)
```

```
Real 1 x 51 matrix
```

```

4000.00    3920.00    3837.60    3752.73    ...

```

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

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

```
32.00
```

Alasan untuk ini adalah bahwa bukan nol(VKR<0) mengembalikan vektor indeks i, di mana VKR[i]<0, dan min menghitung indeks minimal. Karena vektor selalu dimulai dengan indeks 1, jawabannya adalah 31 tahun.

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

```
>{x,n}=iterate("onepay",4000,till="x<0"); x, n
```

```

-0.21
31.00

```

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

Iterasinya sama seperti di atas.

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(4000,-200,3,31)
```

-0.21

Sekarang kita bisa menyelesaikan masalah kita.

```
>solve("f(4000,-200,x,50)",3)
```

4.43

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(4000,x,3,20)",-200)
```

-268.86

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

## Solusi Simbolik untuk Masalah Suku Bunga

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

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

R + qM

Kita sekarang dapat mengulangi ini.

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

$$q (q (q (R + qM) + R) + R) + R$$

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

Kami melihat sebuah pola. Setelah n periode yang kita miliki

Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

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

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

Ini agak rumit. Jumlahnya dievaluasi dengan bantuan "simpsum" untuk menguranginya menjadi hasil bagi.

Mari kita membuat fungsi untuk ini.

Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

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

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

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

>longest f(4000,-200,3,31), longest fs(4000,-200,3,31)

```
-0.2142542009293322
-0.2142542009369208
```

Kita sekarang dapat menggunakan 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(4000,-330,3,x)",30)

15.29

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(M,R,P,n)=Mn; \$&equ

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

Biasanya rumus ini diberikan dalam bentuk

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

$$\frac{((i + 1)^n - 1) R}{i} + (i + 1)^n M = Mn$$

Kita dapat memecahkan tingkat R secara simbolis.

>\$&solve(equ,R)

$$[ R = \frac{i Mn - i (i + 1)^n M}{(i + 1)^n - 1} ]$$

Seperti yang Anda lihat dari rumus, fungsi ini mengembalikan kesalahan titik mengambang untuk i=0.

Euler tetap merencanakannya.

Tentu saja, kami memiliki batas berikut.

>\$&limit(R(4000,0,x,10),x,0)

$$\lim_{x \rightarrow 0} R(4000, 0, x, 10)$$

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

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

# Menggambar Grafik 2D dengan EMT

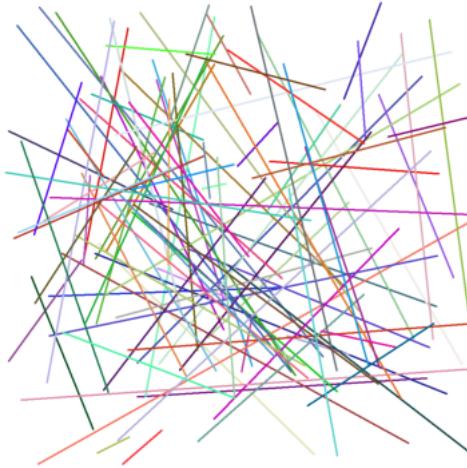
Notebook ini menjelaskan tentang cara menggambar berbagai kurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi plot2d() untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

## Basic Plots

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

Pada contoh, kita hanya menggambar beberapa garis acak dalam berbagai warna. Untuk detail mengenai fungsi-fungsi ini, pelajari fungsi-fungsi inti dari EMT.

```
>clc; // 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
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-001.png

```
>reset;
```

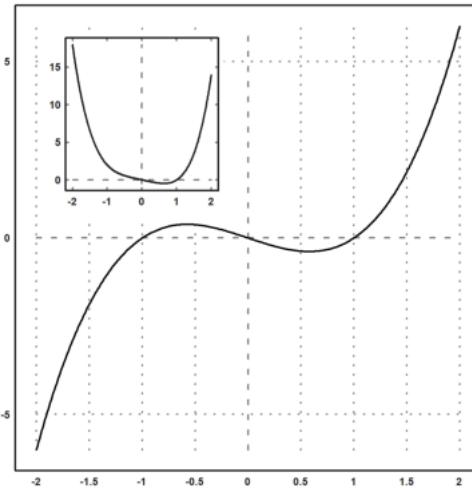
Anda harus menahan grafik, karena perintah plot() akan menghapus jendela plot.

Untuk menghapus semua yang telah kita lakukan, kita gunakan 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 gunakan perintah insimg() untuk menampilkan gambar hasil plot.

Sebagai contoh lain, kita menggambar sebuah plot sebagai inset pada plot yang lain. Hal 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 hal ini sesuai kebutuhan. Perhatikan bahwa kita menyimpan dan mengembalikan jendela penuh, dan menahan plot saat ini ketika kita membuat 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);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-002.png

```
>hold off;
```

```
>window(ow);
```

Plot dengan beberapa angka dicapai dengan cara yang sama. Ada fungsi utility figure() untuk ini.

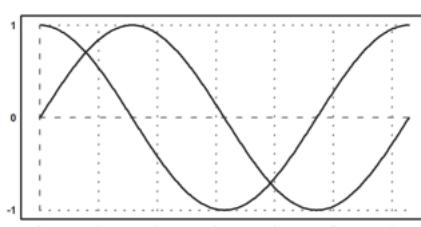
## Plot Aspect

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

Tetapi Anda juga dapat mengubahnya untuk satu plot. Untuk ini, ukuran area plot saat ini diubah, dan jendela diatur sedemikian rupa sehingga label memiliki ruang yang cukup.

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

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-003.png

```
>aspect();
```

```
>reset;
```

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

## Plot 2D di Euler

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

Diumungkinkan untuk memplot di Maxima menggunakan Gnuplot atau di Python menggunakan Math Plot Lib.

Euler dapat memplot plot 2D dari

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

Fungsi yang kompleks

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

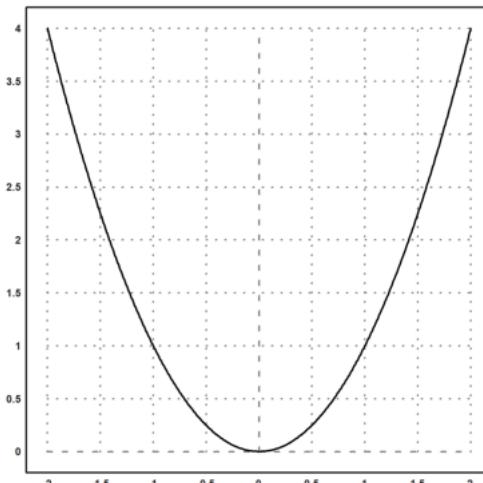
## Plot Ekspresi atau Variabel

Ekspresi tunggal dalam “x” (misalnya “ $4*x^2$ ”) atau nama fungsi (misalnya “f”) menghasilkan grafik fungsi.

Berikut ini adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsi.

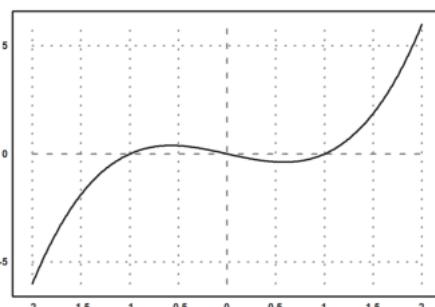
Catatan: Jika Anda mengakhiri baris perintah dengan tanda titik dua “:”, plot akan disisipkan ke dalam jendela teks. Jika tidak, tekan TAB untuk melihat plot jika jendela plot tertutup.

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



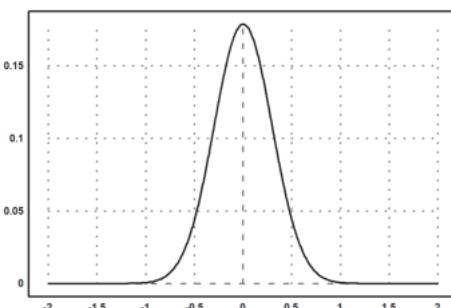
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-004.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-005.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-006.png

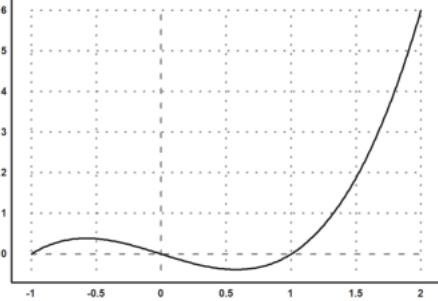
Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

Rentang plot ditetapkan dengan parameter yang ditetapkan berikut ini

- a,b: rentang x (default -2,2)

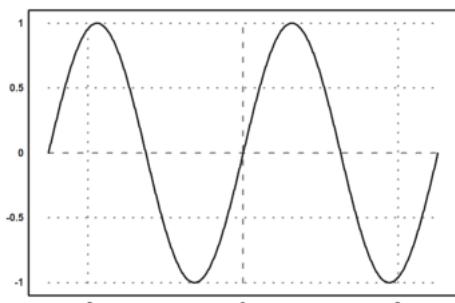
- c, d: rentang y (default: skala dengan nilai)
- r: sebagai alternatif, radius di sekitar pusat plot
- cx, cy: koordinat pusat plot (default 0,0)

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



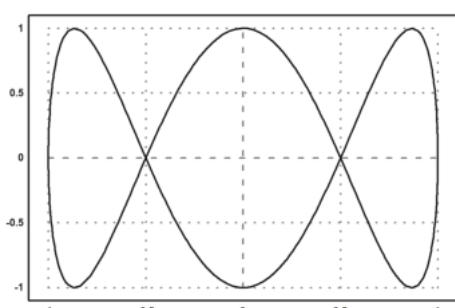
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-007.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-008.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-009.png

Alternatif untuk tanda titik dua adalah perintah insimg(lines), 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.

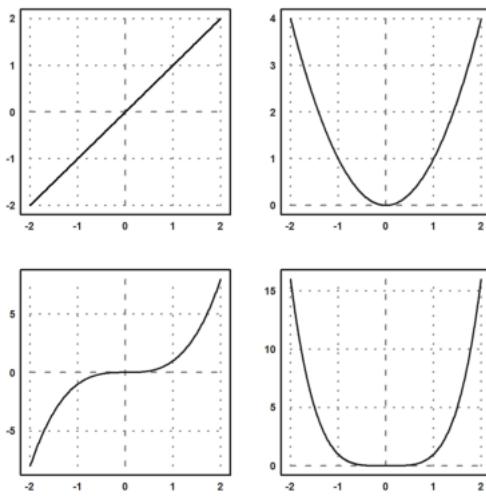
Lebih banyak gaya yang dapat dicapai dengan perintah plot tertentu.

Dalam hal apa pun, tekan tombol tabulator untuk melihat plot, jika disembunyikan.

Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Pada contoh, kita memplot  $x^1$  sampai  $x^4$  ke dalam 4 bagian jendela. figure(0) akan mereset jendela default.

>reset;

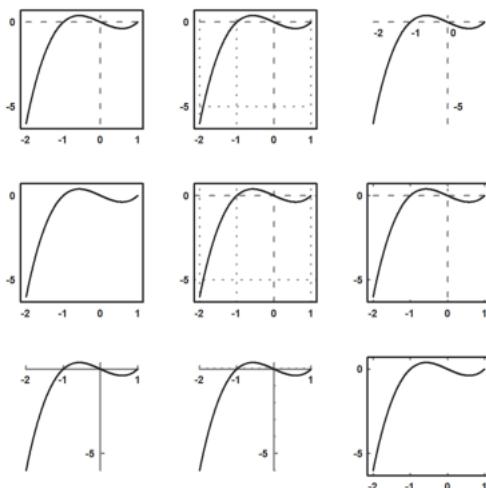
```
>figure(2,2); ...
>for n=1 to 4; figure(n); plot2d("x^"+n); end; ...
>figure(0);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-010.png

Pada plot2d(), terdapat beberapa gaya alternatif yang tersedia dengan grid=x. Sebagai gambaran umum, kami menampilkan berbagai gaya grid dalam satu gambar (lihat di bawah ini untuk perintah figure()). Gaya grid=0 tidak disertakan. Gaya ini tidak menampilkan grid dan frame.

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



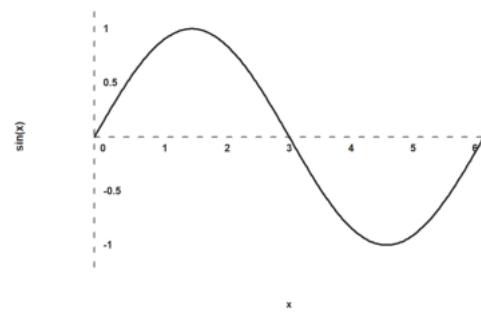
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-011.png

Jika argumen untuk plot2d() adalah sebuah 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=... dst.

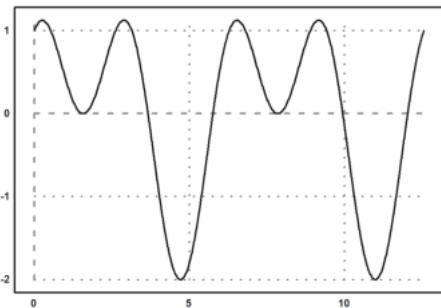
Pada contoh berikut, kita mengubah gaya grid, menambahkan label, dan menggunakan label vertikal untuk sumbu y.e y-axis.

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-012.png

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



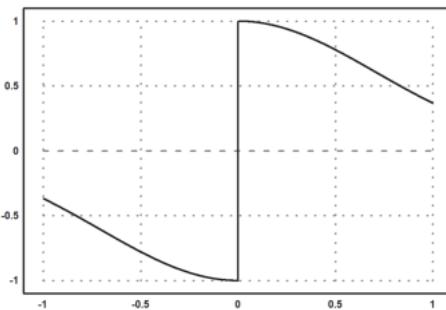
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-013.png

Gambar yang dihasilkan dengan menyisipkan plot ke dalam jendela teks disimpan dalam direktori yang sama dengan notebook, secara default dalam subdirektori bernama “images”. Gambar-gambar tersebut juga digunakan oleh ekspor HTML.

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

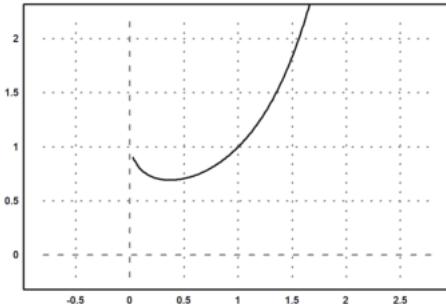
Fungsi atau ekspresi dalam plot2d dievaluasi secara adaptif. Untuk kecepatan yang lebih tinggi, matikan plot adaptif dengan <adaptive> dan tentukan jumlah subinterval dengan n=... Hal ini hanya diperlukan pada kasus-kasus yang jarang terjadi.

> plot2d("sign(x)\*exp(-x^2)",-1,1,<adaptive,n=10000):



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-014.png

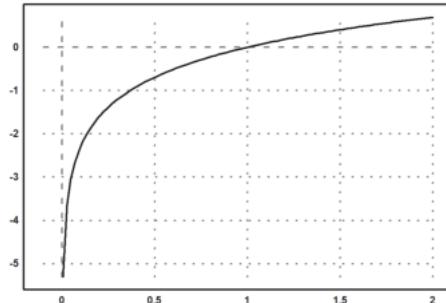
>plot2d("x^x",r=1.2,cx=1,cy=1):



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-015.png

Perhatikan bahwa x^x tidak didefinisikan untuk x<=0. Fungsi plot2d menangkap kesalahan ini, dan mulai memplot segera setelah fungsi didefinisikan. Hal ini berlaku untuk semua fungsi yang mengembalikan NAN di luar jangkauan definisinya.

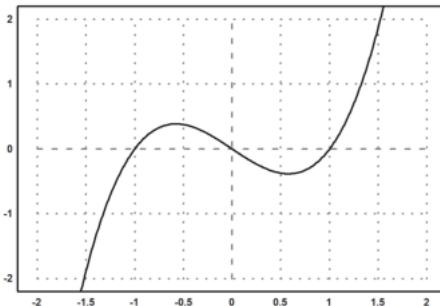
>plot2d("log(x)",-0.1,2):



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-016.png

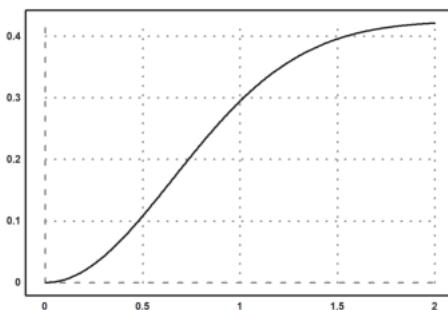
Parameter square=true (atau >square) memilih rentang y secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-017.png

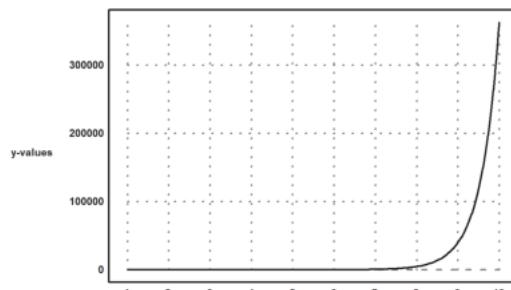
```
>plot2d("integrate("sin(x)*exp(-x^2)",0,x)",0,2); // plot integral
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-018.png

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

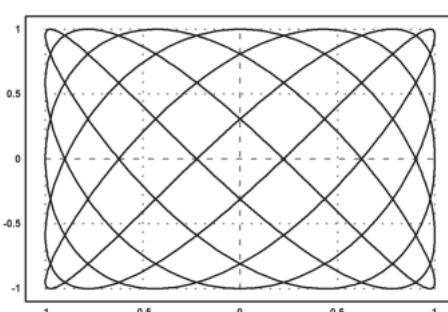
```
>plot2d("gamma(x)",1,10,yl="y-values",smaller=6,<vertical):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-019.png

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

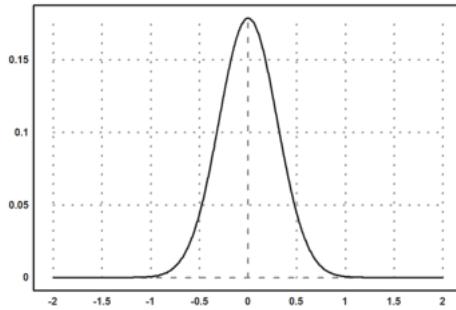
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x));
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-020.png

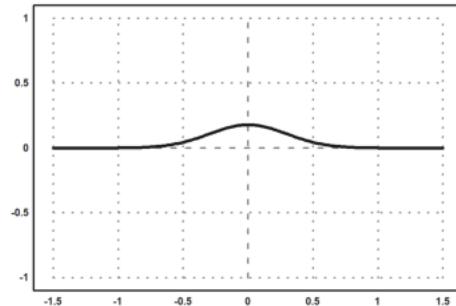
```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
```

```
>plot2d(expr,-2,2); // plot from -2 to 2
```



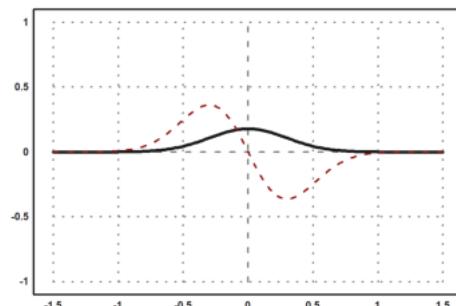
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-021.png

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



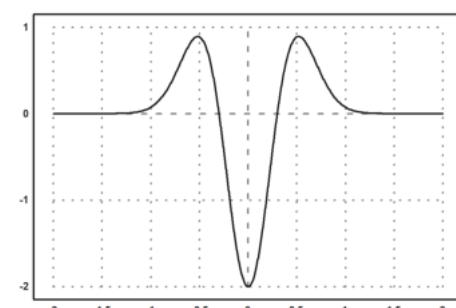
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-022.png

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



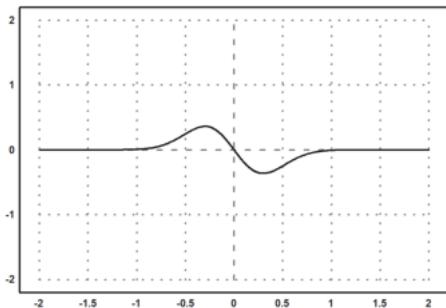
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-023.png

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



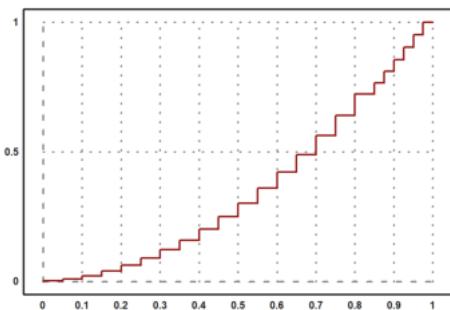
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-024.png

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



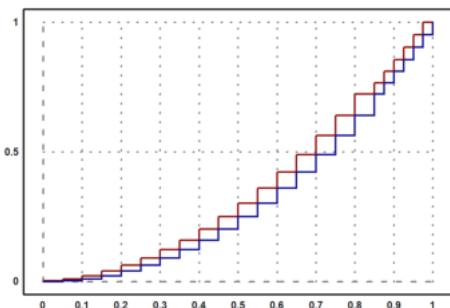
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-025.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-026.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-027.png

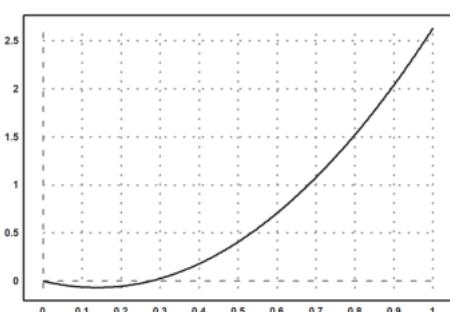
## Fungsi dalam satu Parameter

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

Berikut adalah beberapa contoh penggunaan fungsi. Seperti biasa dalam EMT, fungsi yang bekerja untuk fungsi atau ekspresi lain, Anda dapat mengoper parameter tambahan (selain  $x$ ) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan koleksi panggilan.

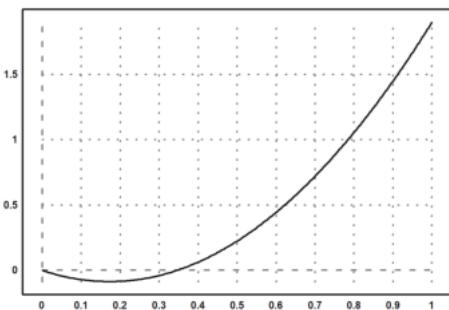
```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
```

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



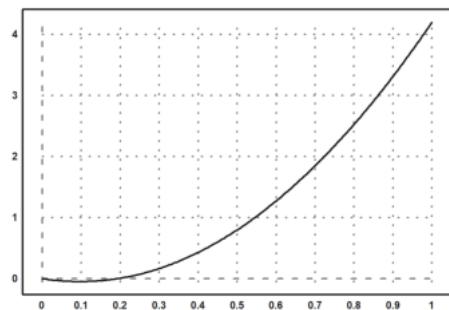
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-028.png

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



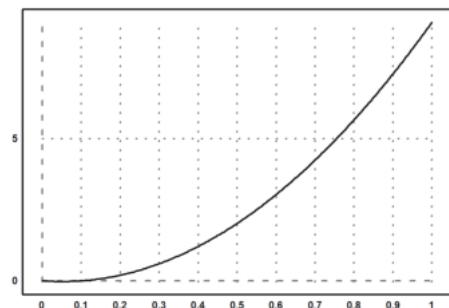
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-029.png

```
>plot2d({{"f",0.2}},0,1); // plot with a=0.2
```



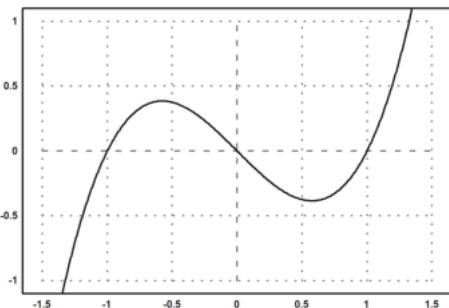
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-030.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-031.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-032.png

Berikut ini adalah ringkasan dari fungsi yang diterima

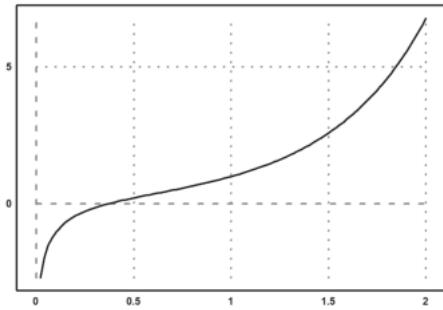
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolik dengan nama sebagai "f"
- fungsi-fungsi simbolik hanya dengan nama f

Fungsi plot2d() juga menerima fungsi simbolik. Untuk fungsi simbolik, nama saja sudah cukup.

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

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

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



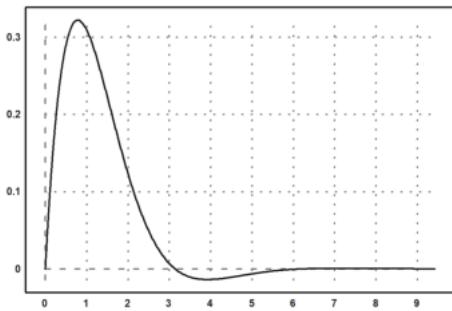
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-033.png

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

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

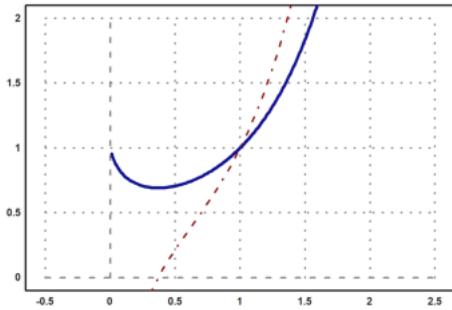
$$E^{-x} \sin(x)$$

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-034.png

```
>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="--."):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-035.png

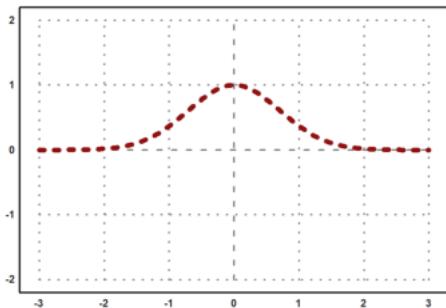
Untuk gaya garis, ada berbagai pilihan.

- style = "...". Pilih dari “-”, “-”, “-.”, “:”, “.-.”, “-.-”.
- color: 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 muda, kuning
- rgb (merah, hijau, biru): parameter adalah real dalam [0,1].

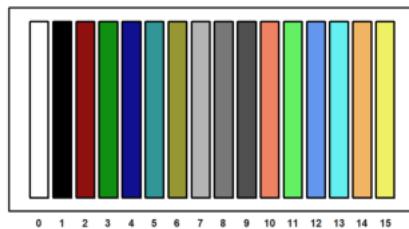
```
>plot2d("exp(-x^2)",r=2,color=red,thickness=3,style="--");
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-036.png

Berikut ini adalah pemandangan warna EMT yang sudah ditetapkan sebelumnya.

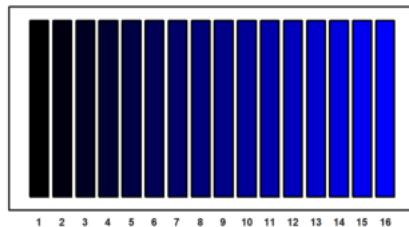
>aspect(2); columnsplot(ones(1,16),lab=0:15,grid=0,color=0:15):



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-037.png

Tetapi Anda bisa menggunakan warna apa pun.

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

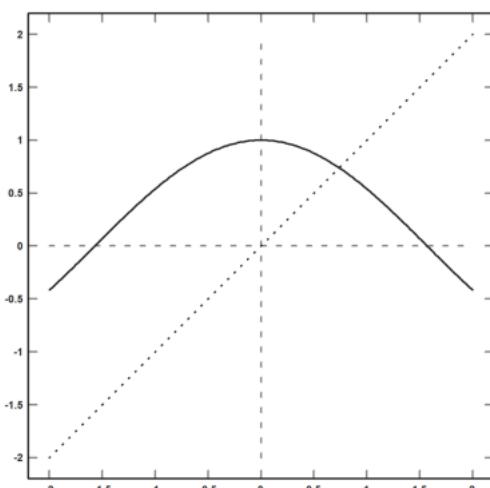


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-038.png

## Menggambar beberapa kurva pada bidang koordinat yang sama

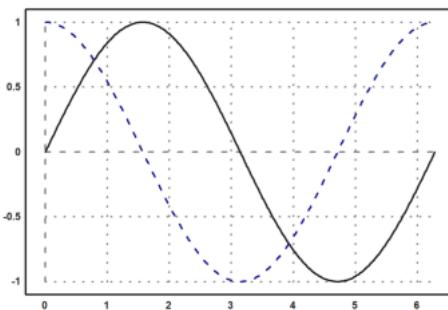
Memplot lebih dari satu fungsi (beberapa fungsi) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu caranya adalah dengan menggunakan >add untuk beberapa pemanggilan ke plot2d secara bersamaan, kecuali pemanggilan pertama. Kita telah menggunakan fitur ini pada contoh di atas.

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-039.png

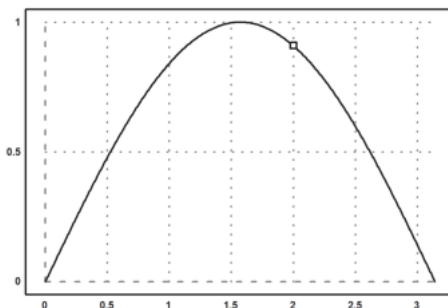
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-040.png

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

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

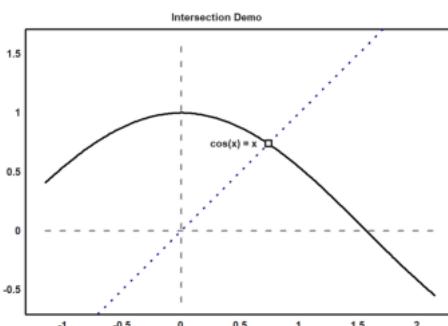


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-041.png

Kami menambahkan titik perpotongan dengan label (pada posisi “cl” untuk kiri tengah), dan menyisipkan hasilnya ke dalam buku catatan. 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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-042.png

Dalam demo berikut ini, 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 simbolik.

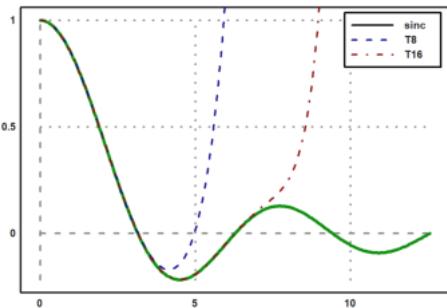
Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga pemanggilan `plot2d()`. Perintah kedua dan ketiga memiliki set flag `>add`, yang membuat plot menggunakan rentang sebelumnya.

Kami menambahkan sebuah kotak label yang menjelaskan fungsi-fungsi tersebut.

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

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

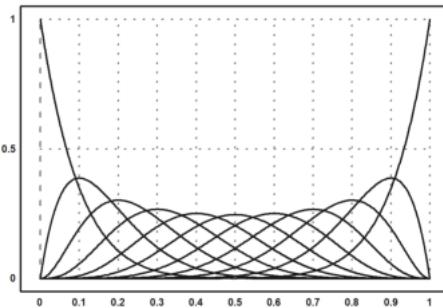
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
> plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style=""); ...
> plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="--"); ...
> labelbox(["sinc","T8","T16"],styles=[“-”,“-”,“--”], ...
> colors=[black,blue,red]);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-044.png

Pada contoh berikut, kami menghasilkan Polinomial Bernstein.

```
>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;
```

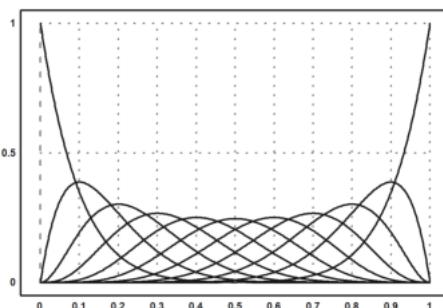


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-045.png

Metode kedua menggunakan sepasang matriks nilai x dan matriks nilai y dengan ukuran yang sama.

Kita membuat sebuah matriks nilai dengan satu Polinomial Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihatlah pengantar tentang bahasa matriks untuk mempelajari lebih lanjut.

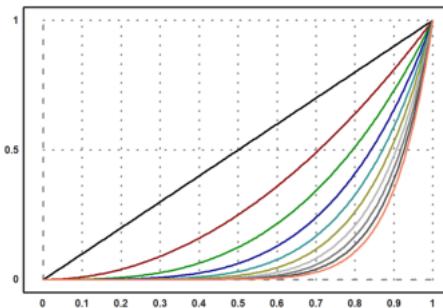
```
>x=linspace(0,1,500);
>n=10; k=(0:n)'; // n is row vector, k is column vector
>y=bin(n,k)*x.^k*(1-x).^(n-k); // y is a matrix then
>plot2d(x,y);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-046.png

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

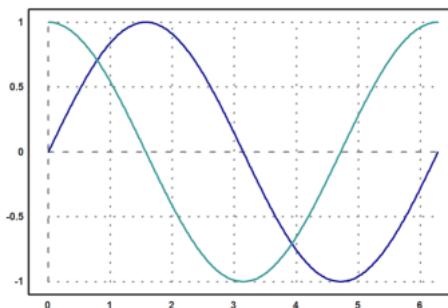
```
>x=linspace(0,1,200); y=x^(1:10)'; plot2d(x,y,color=1:10);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-047.png

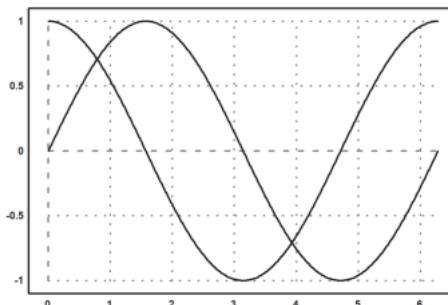
Metode lainnya adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan larik warna, larik gaya, dan larik ketebalan dengan panjang yang sama.

>plot2d([“sin(x)”,“cos(x)”),0,2pi,color=4:5]:



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-048.png

>plot2d([“sin(x)”,“cos(x)”),0,2pi]: // plot vector of expressions



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-049.png

Kita bisa mendapatkan vektor seperti itu dari Maxima dengan menggunakan makelist() dan mxm2str().

>v &= makelist(binomial(10,i)\*x^(1-x)(10-i),i,0,10) // make list

```


$$\begin{aligned} & [ \frac{1}{6} x^{10}, \frac{10}{4} x^9, \frac{45}{5} x^8, \frac{120}{4} x^7, \frac{210}{3} x^6, \\ & \quad \frac{252}{10} x^5, \frac{210}{9} x^4, \frac{120}{8} x^3, \frac{45}{7} x^2, \\ & \quad 10 x, x^1 ] \end{aligned}$$


```

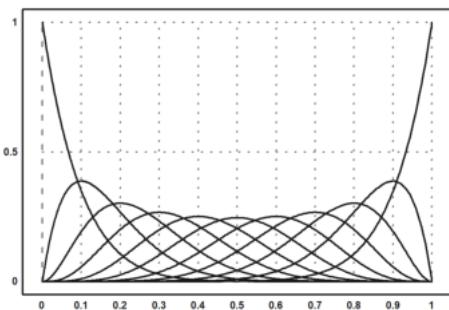
>mxm2str(v) // get a vector of strings from the symbolic vector

```

(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



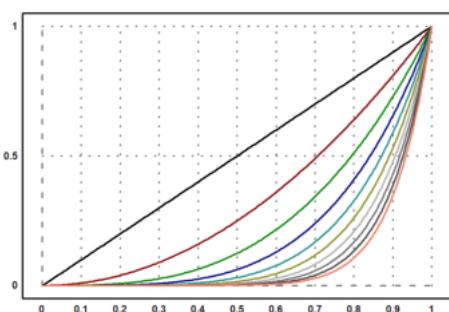
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-050.png

Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

Jika sebuah ekspresi menghasilkan sebuah 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 sebuah laris warna ditambahkan, maka akan digunakan untuk setiap baris plot.

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



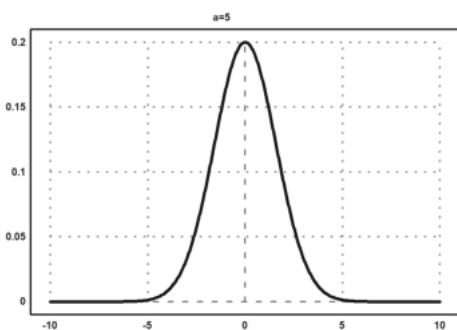
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-051.png

Ekspresi dan fungsi satu baris dapat melihat variabel global.

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

Berhati-hatilah untuk meletakkan semua parameter yang diberikan di akhir perintah plot2d. Pada contoh ini kita mengoper a=5 ke fungsi f, yang kita plot dari -10 ke 10.

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

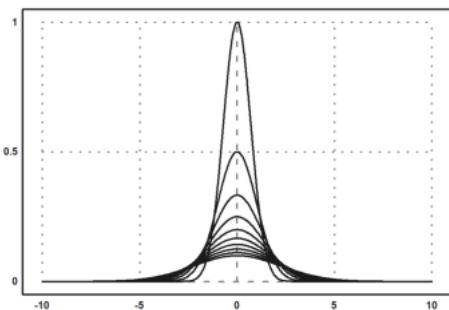


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-052.png

Atau, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut koleksi panggilan, dan itu adalah cara yang lebih disukai untuk mengoper argumen ke fungsi yang dengan sendirinya dioper sebagai argumen ke fungsi lain.

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

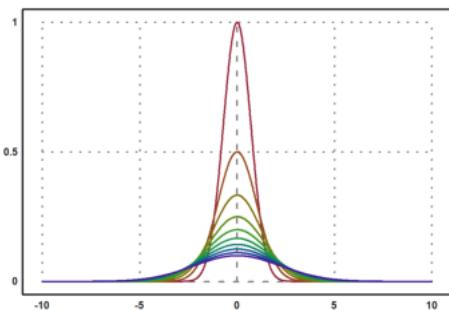
>plot2d({{"f",1}},-10,10); ...  
> for a=2:10; plot2d({{"f",a}},>add); end;



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-053.png

Kita dapat mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris dari 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));
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-054.png

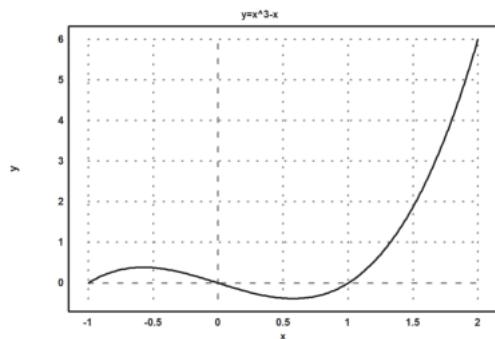
## Label Teks

Dekorasi sederhana dapat berupa

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

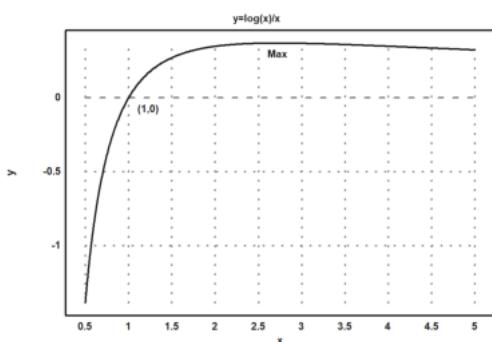
Perintah label akan memplotkan ke dalam plot saat ini pada koordinat plot (x,y). Perintah ini dapat menerima sebuah argumen posisi.

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-055.png

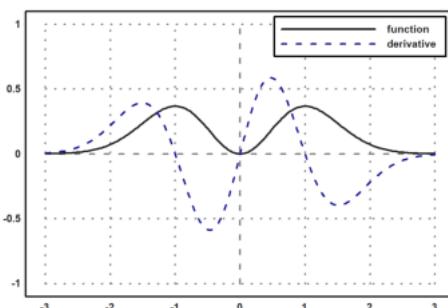
```
>expr := "log(x)/x"; ...
> plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
> label("(1,0)",1,0); label("Max",E,expr(E),pos="lc");
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-056.png

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

```
>function f(x) &= x^2*exp(-x); ...
>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);
```



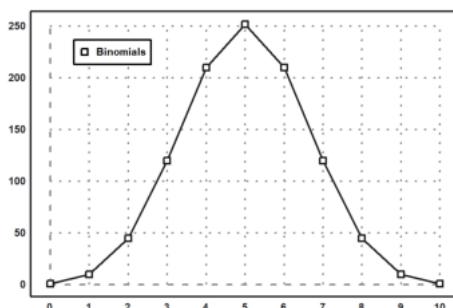
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-057.png

Kotak tersebut berlabuh di kanan atas secara default, tetapi >kiri menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat mana pun yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafik. Lebarnya otomatis.

Untuk plot titik, kotak label juga dapat digunakan. Tambahkan parameter >titik, atau vektor bendera, satu untuk setiap label.

Pada contoh berikut, hanya ada satu fungsi. Jadi kita dapat menggunakan string dan bukan vektor dari string. Kita mengatur warna teks menjadi hitam untuk contoh ini.

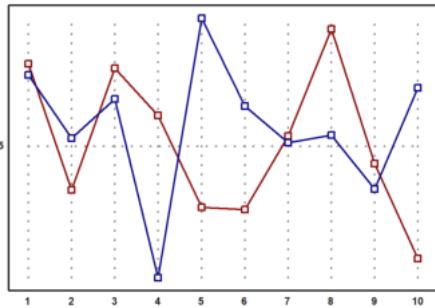
```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
>labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
>tcolor=black,>left);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-058.png

Gaya plot ini juga tersedia di statplot(). Seperti pada plot2d() warna dapat diatur untuk setiap baris plot. Terdapat lebih banyak plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

```
>statplot(1:10,random(2,10),color=[red,blue]);
```

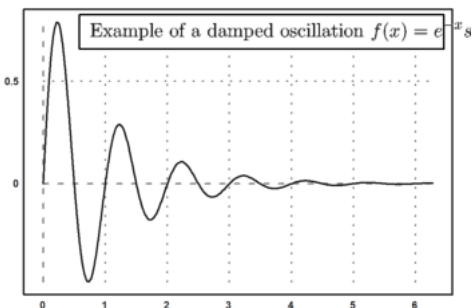


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-059.png

Fitur yang serupa adalah fungsi `textbox()`.

Lebarnya secara default adalah lebar maksimal baris teks. Tetapi bisa juga diatur oleh pengguna.

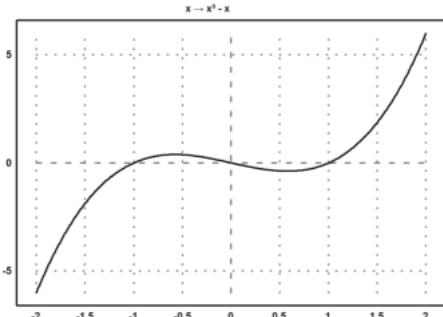
```
>function f(x) &= exp(-x)*sin(2*pi*x); ...
> plot2d("f(x)",0,2pi);
> textbox(latex("Example of a damped oscillation")\ f(x)=e^{-x}sin(2\pi x)),w=0.85;
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-060.png

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

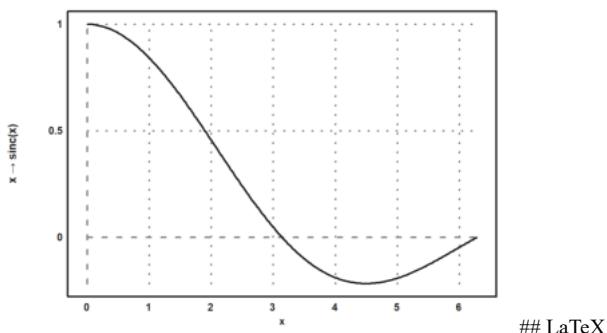
```
>plot2d("x^3-x",title=u"x → x³ - x");
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-061.png

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

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



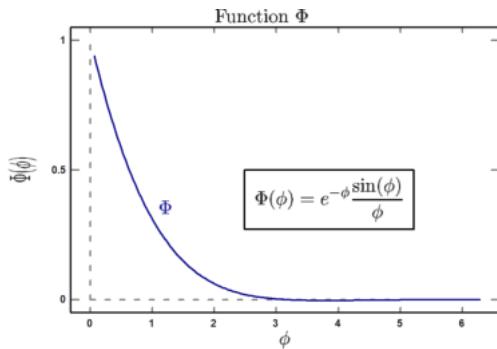
## LaTeX

Anda juga dapat memplot formula LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke binari "lateks" dan "dvipng" harus berada di jalur sistem, atau Anda harus mengatur LaTeX di menu opsi.

Perhatikan, bahwa penguraian LaTeX berjalan lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil `latex()` sebelum perulangan satu kali dan menggunakan hasilnya (gambar dalam matriks RGB).

Pada plot berikut ini, kita menggunakan LaTeX untuk label x dan y, sebuah label, kotak label dan judul plot.

```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
> title=latex("text{Function Phi}"), ...
> xl=latex("\phi"),yl=latex("\Phi(\phi)"); ...
> textbox( ...
> latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
> label(latex("\Phi"),color=blue),1,0.4):
```

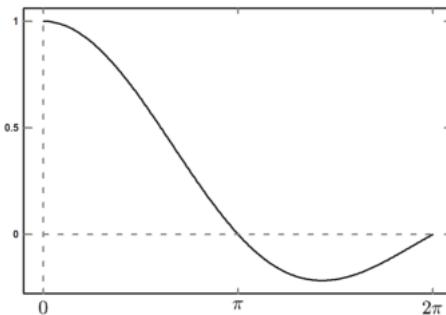


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-063.png

Seringkali, kita menginginkan spasi dan label teks yang tidak sesuai pada sumbu x. Kita dapat menggunakan `xaxis()` dan `yaxis()` seperti yang akan kita tunjukkan nanti.

Cara termudah adalah dengan membuat plot kosong dengan sebuah frame menggunakan `grid=4`, dan kemudian menambahkan grid dengan `ygrid()` dan `xgrid()`. Pada contoh berikut, kita menggunakan tiga buah string LaTeX untuk label pada sumbu x dengan `xtick()`.

```
>plot2d("sinc(x)",0,2pi,grid=4,<ticks); ...
> ygrid(-2:0.5:2,grid=6); ...
> xgrid([0:2]*pi,<ticks,grid=6); ...
> xtick([0,pi,2pi],[“0”,“\pi”,“2\pi”],>latex):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-064.png

Tentu saja, fungsi juga dapat digunakan.

```
>function map f(x) ...
if x>0 then return x^4
else return x^2
endif
endfunction
```

Parameter “map” membantu menggunakan fungsi untuk vektor. Untuk plot, hal ini tidak diperlukan. Tetapi untuk mendemonstrasikan bahwa vektorisasi berguna, kami menambahkan beberapa titik kunci pada plot pada  $x=-1$ ,  $x=0$  dan  $x=1$ .

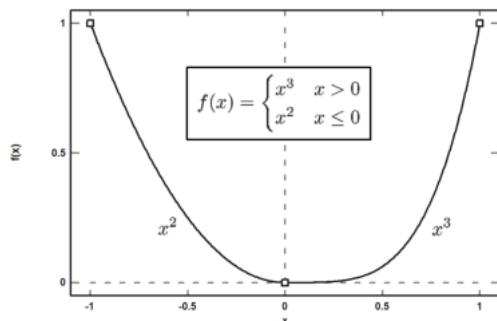
Pada plot berikut, kita juga memasukkan beberapa kode LaTeX. Kita menggunakanannya untuk

dua label dan sebuah kotak teks. Tentu saja, Anda hanya dapat menggunakan

LaTeX jika Anda telah menginstal LaTeX dengan benar.

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

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

"), ...  
 > x=0.7,y=0.2):


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-065.png

## Interaksi Pengguna

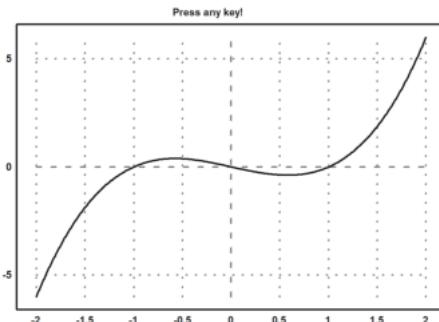
Ketika memplot fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat

- memperbesar dengan + atau -
- memindahkan plot dengan tombol kursor
- memilih jendela plot dengan mouse
- mengatur ulang tampilan dengan spasi
- keluar dengan return

Tombol spasi akan mengatur ulang plot ke jendela plot awal.

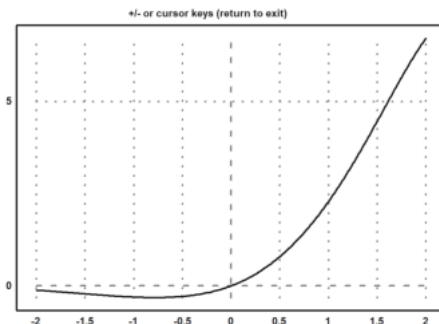
Ketika memplot data, bendera >user hanya akan menunggu penekanan tombol.

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-066.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-067.png

Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi bawaan mousedrag() menunggu peristiwa mouse atau keyboard. Fungsi ini melaporkan mouse ke bawah, mouse bergerak atau mouse ke atas, dan penekanan tombol. Fungsi dragpoints() memanfaatkan hal ini, dan mengizinkan pengguna untuk menyeret titik manapun di dalam plot.

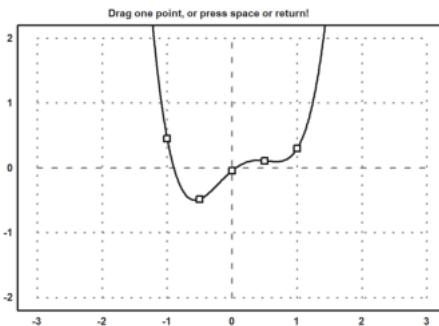
Kita membutuhkan fungsi plot terlebih dahulu. Sebagai contoh, kita melakukan interpolasi pada 5 titik dengan sebuah polinomial. Fungsi ini harus memplot ke dalam area plot yang tetap.

```
>function plotf(xp,yp,select) ...
d=interp(xp,yp);
plot2d("interval(xp,d,x)",d,xp,r=2);
plot2d(xp,yp,>points,>add);
if select>0 then
  plot2d(xp[select],yp[select],color=red,>points,>add);
endif;
title("Drag one point, or press space or return!");
endfunction
```

Perhatikan parameter titik koma pada plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, untuk mengakses nilai secara global.

Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret titik-titiknya.

```
>t=-1:0.5:1; dragpoints("plotf",t,random(size(t))-0.5);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-068.png

Ada juga fungsi yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

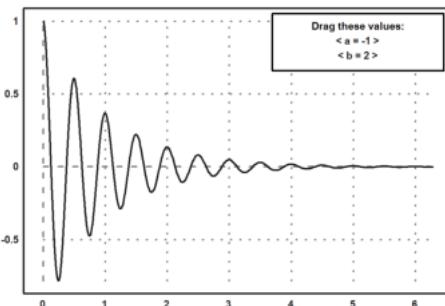
Pertama, kita memerlukan fungsi plot.

```
>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, dan secara opsional, sebuah garis judul.

Terdapat 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);
```



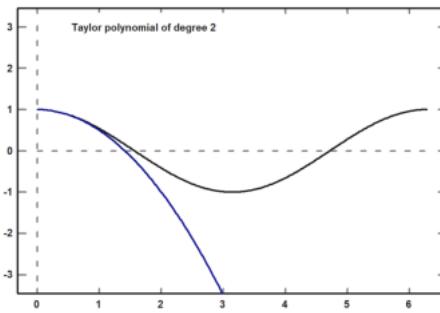
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-069.png

Anda dapat membatasi nilai yang diseret menjadi bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor dengan 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 kita biarkan derajat n bervariasi dari 0 sampai 20 dalam 20 stop. Hasil dari dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk menyisipkan plot ke dalam buku catatan.

```
>nd=dragvalues("plotf","degree",2,[0,20],20,y=0.8, ...
> heading="Drag the value:");
> plotf(nd);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-070.png

Berikut ini adalah peragaan sederhana dari fungsi ini. Pengguna dapat menggambar di atas jendela plot, meninggalkan jejak titik.

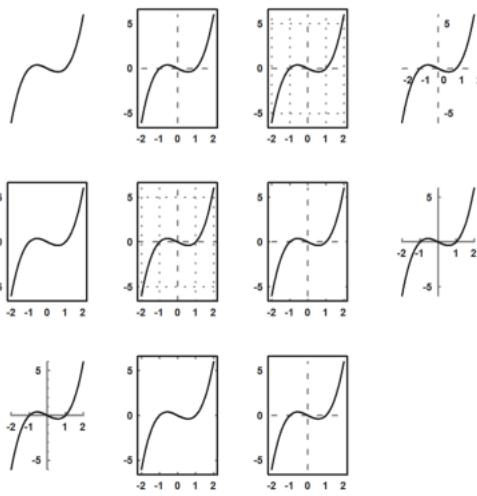
```
>function dragtest ...
plot2d(None,r=1,title="Drag with the mouse, or press any key!");
start=0;
repeat
{flag,m,time}=mousedrag();
if flag==0 then return; endif;
if flag==2 then
    hold on; mark(m[1],m[2]); hold off;
endif;
end
endfunction

>dragtest // lihat hasilnya dan cobalah lakukan!
```

## Gaya Plot 2D

Secara default, EMT menghitung tanda sumbu otomatis dan menambahkan label pada setiap tanda. Hal 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);
```

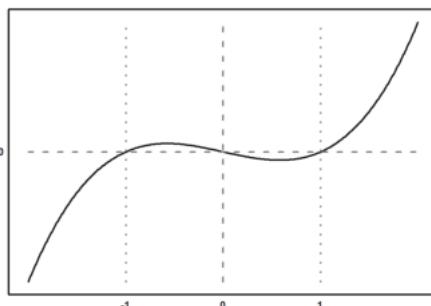


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-071.png

Parameter <frame mematikan bingkai, dan framecolor=blue menetapkan bingkai ke warna biru.

Jika Anda menginginkan tanda centang Anda sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

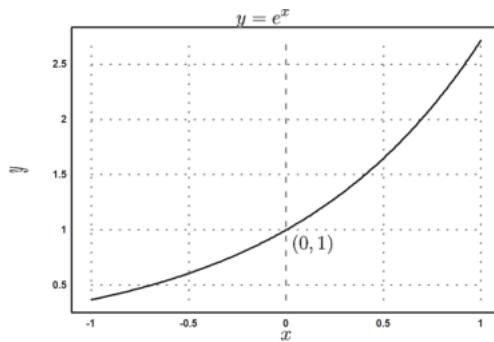
```
>aspect(1.5);
>plot2d("x^3-x",grid=0); // plot
>frame; xgrid([-1,0,1]); ygrid(0); // add frame and grid
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-072.png

Untuk judul plot dan label sumbu, lihat contoh berikut.

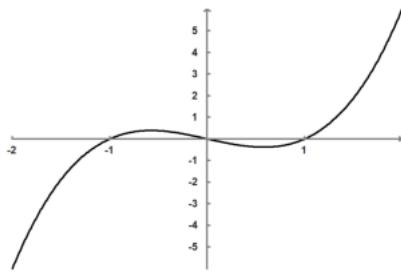
```
>plot2d("exp(x)",-1,1);
>textcolor(black); // set the text color to black
>title(latex("y=e^x")); // title above the plot
>xlabel(latex("x")); // "x" for x-axis
>ylabel(latex("y"),>vertical); // vertical "y" for y-axis
>label(latex("(0,1)'),0,1,color=blue); // label a point
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-073.png

Sumbu dapat digambar secara terpisah dengan sumbu x() dan sumbu y().

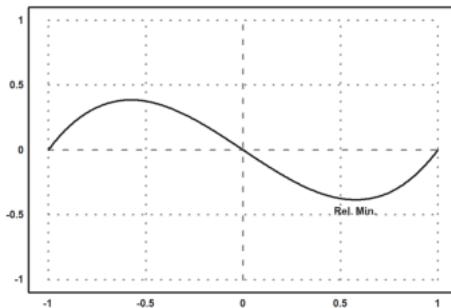
```
>plot2d("x^3-x",<grid,<frame);
>xaxis(0,xx=-2:1,style="->"); yaxis(0,yy=-5:5,style="->");
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-074.png

Teks pada plot dapat diatur dengan label(). Pada contoh berikut ini, ‘‘lc’’ berarti lower center. Ini mengatur posisi label relatif terhadap koordinat plot.

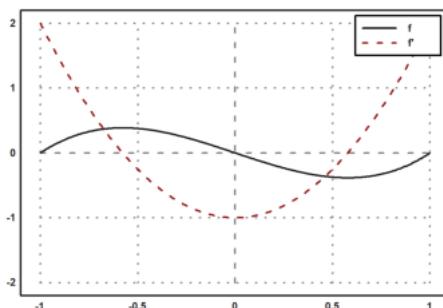
```
>function f(x) &= x^3-x
      3
      x  - x
>plot2d(f,-1,1,>square);
>x0=fmin(f,0,1); // compute point of minimum
>label("Rel. Min.",x0,f(x0),pos="lc"); // add a label there
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-075.png

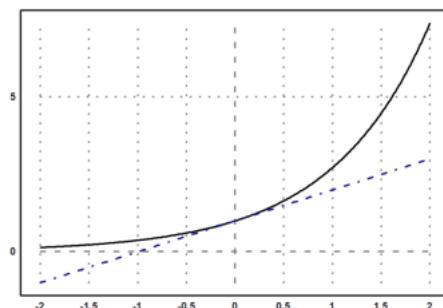
Terdapat juga kotak teks.

```
>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
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-076.png

```
>plot2d(["exp(x)","1+x"],color=[black,blue],style=["-","-."]):
```

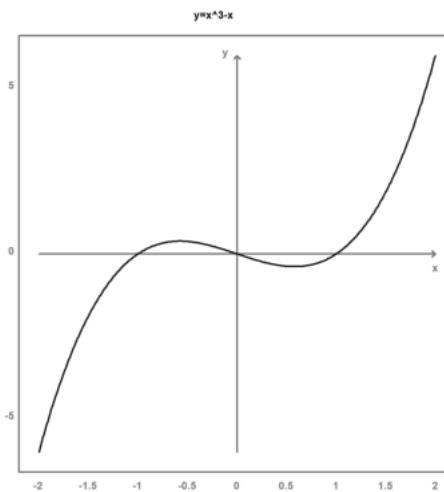


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-077.png

```

>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-078.png

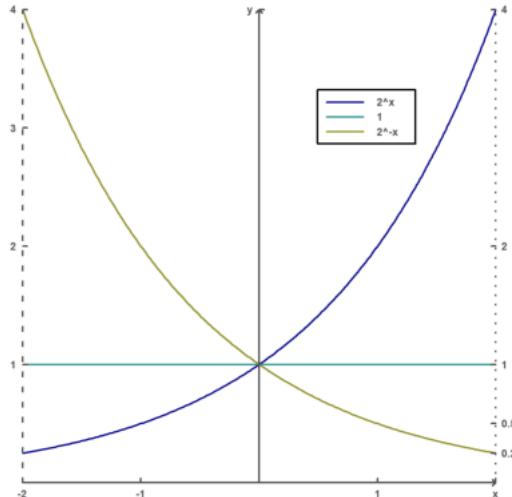
Untuk kontrol yang lebih besar lagi, sumbu x dan sumbu y dapat dilakukan secara manual.

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;

```



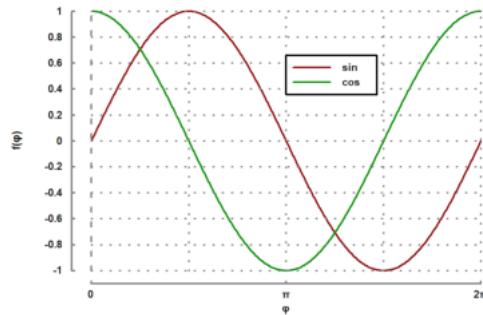
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-079.png

Berikut ini 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,(0.2)*pi,xt=[“0”,u”π”,u”2π”],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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-080.png

## Memplot Data 2D

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

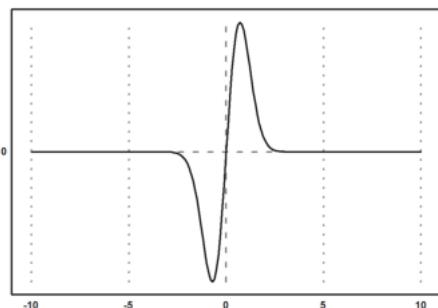
Memplot ekspresi hanyalah singkatan untuk plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai x, dan satu atau beberapa baris nilai y. Dari rentang dan nilai x, fungsi plot2d akan menghitung data untuk diplot, secara default dengan evaluasi adaptif dari fungsi tersebut. Untuk plot titik, gunakan ">points", untuk garis dan titik campuran gunakan ">addpoints".

Namun Anda dapat memasukkan data secara langsung.

- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y diplot baris demi baris.

Berikut adalah contoh dengan satu baris untuk x dan y.

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-081.png

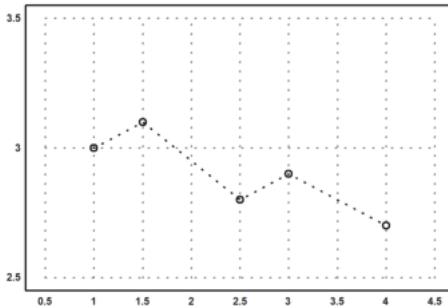
Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plot ini bekerja seperti poligon, namun hanya menggambar sudut-sudutnya saja.

- style = "...": Pilih dari "[]", "<>", "o", ".", "..", "+", "\*", "[", "[", "#", "<>#", "o#", ".#", "#", "|".

Untuk memplot kumpulan titik, gunakan >titik. Jika warna adalah sebuah vektor warna, setiap titik mendapatkan warna yang berbeda. Untuk sebuah matriks koordinat dan vektor kolom, warna berlaku pada baris-baris matriks.

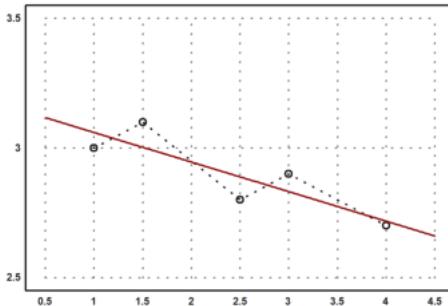
Parameter >addpoints menambahkan titik-titik pada segmen garis untuk plot data.

```
>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-082.png

```
>p=polyfit(xdata,ydata,1); // get regression line
>plot2d("polyval(p,x)",>add,color=red); // add plot of line
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-083.png

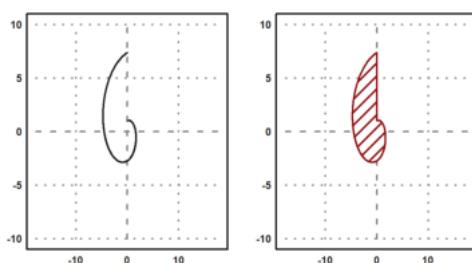
## Menggambar Daerah Yang Dibatasi Kurva

Plot data sebenarnya adalah poligon. Kita juga dapat memplot kurva atau kurva yang terisi.

- filled=true mengisi plot.
- style = "...": Pilih dari "#", "/", "", "/".
- fillcolor: Lihat di atas untuk warna yang tersedia.

Warna isian ditentukan oleh argumen "fillcolor", dan pada pilihan <outline mencegah menggambar batas untuk semua gaya kecuali gaya default.

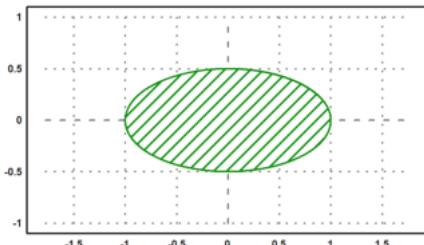
```
>t=linspace(0,2pi,1000); // parameter for curve
>x=sin(t)*exp(t/pi); y=cos(t)*exp(t/pi); // x(t) and y(t)
>figure(1,2); aspect(16/9)
>figure(1); plot2d(x,y,r=10); // plot curve
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // fill curve
>figure(0):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-084.png

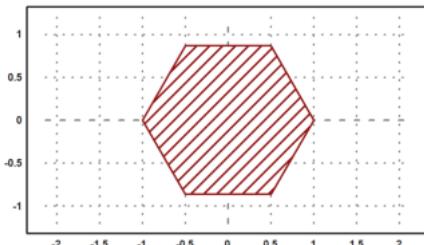
Pada contoh berikut ini, kami memplot clips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian yang berbeda.

```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/");
```



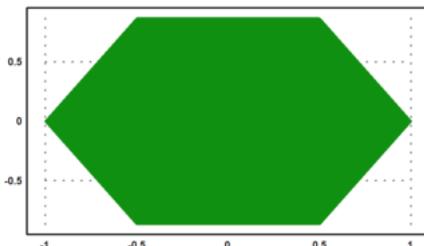
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-085.png

```
>t=linspace(0,2pi,6); ...
>plot2d(cos(t),sin(t),>filled,style='/',fillcolor=red,r=1.2);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-086.png

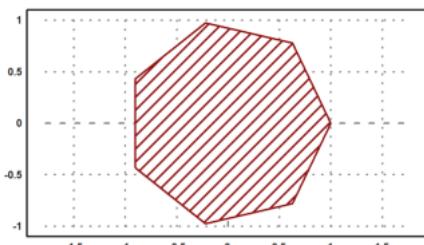
```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#");
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-087.png

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

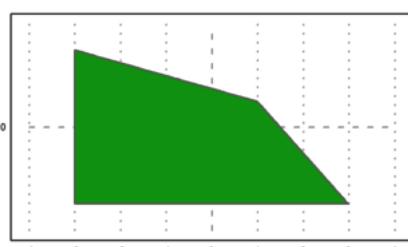
```
>t=linspace(0,2pi,7); ...
>plot2d(cos(t),sin(t),r=1,>filled,style='/',fillcolor=red);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-088.png

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 barisan  $A$ . Untuk mendapatkan sudut-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);
```



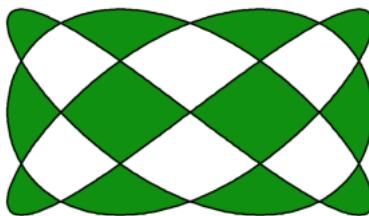
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-089.png

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

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);
```

Kita sekarang memiliki vektor nilai x dan y. plot2d() dapat memplot nilai-nilai ini sebagai sebuah kurva yang menghubungkan titik-titik. Plot dapat diisi. Dalam kasus ini memberikan hasil yang bagus karena aturan pengulangan, yang digunakan untuk pengisian.

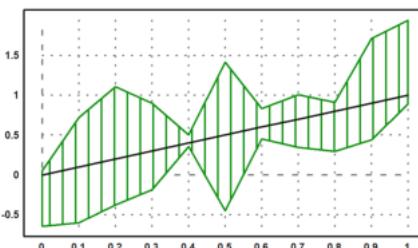
```
>plot2d(x,y,<grid,<frame,>filled):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-090.png

Vektor interval diplot terhadap nilai x sebagai wilayah yang terisi antara nilai bawah dan atas interval. Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga dapat digunakan untuk memplot kesalahan statistik.

```
>t=0:0.1:1; ...
>plot2d(t,interval(t-random(size(t)),t+random(size(t))),style="|"); ...
>plot2d(t,t,add=true):
```

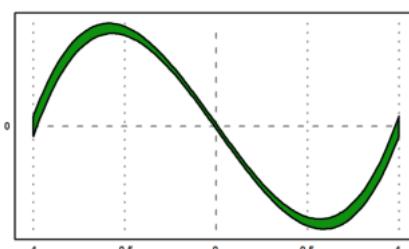


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-091.png

Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi pada 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):
```

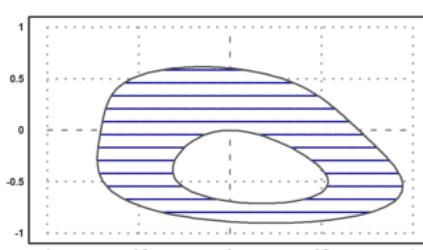


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-092.png

Diumungkinkan 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*y4+y"; // define an expression f(x,y)
```

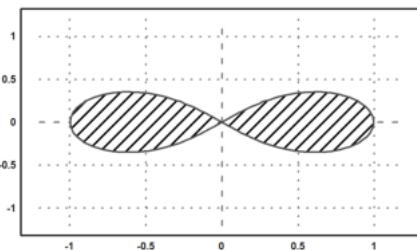
```
>plot2d(expr,level=[0;1],style="-",color=blue); // 0 <= f(x,y) <= 1
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-093.png

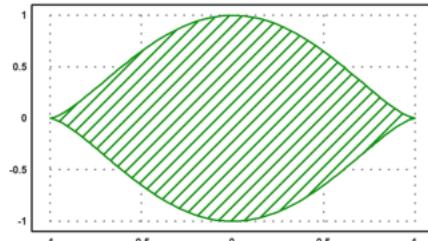
Kita juga dapat mengisi rentang nilai seperti

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot2D-094.png

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



# Menggambar Plot 3D

dengan EMT

Ini adalah pengenalan plot 3D di Euler. Kita memerlukan plot 3D untuk memvisualisasikan fungsi dua variabel.

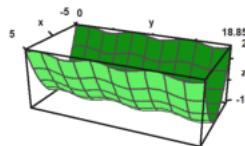
Euler menggambar fungsi tersebut menggunakan algoritma pengurutan untuk menyembunyikan bagian-bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Standarnya adalah dari kuadran x-y positif ke arah titik asal  $x=y=z=0$ , tetapi sudut=0° terlihat dari arah sumbu y. Sudut pandang dan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

Euler dapat memetakan

- permukaan dengan bayangan dan garis level atau rentang level,
- awan titik-titik,
- kurva parametrik,
- permukaan implisit.

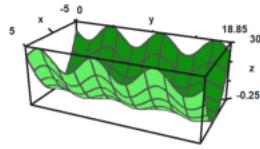
Plot 3D dari sebuah fungsi menggunakan plot3d. Cara termudah adalah memplot ekspresi dalam x dan y. Parameter r mengatur rentang plot di sekitar (0,0).

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-001.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-002.png

Silakan lakukan modifikasi agar gambar “talang bergelombang” tersebut tidak lurus melainkan melengkung/melingkar, baik melingkar secara mendatar maupun melingkar turun/naik (seperti papan peluncur pada kolam renang). Temukan rumusnya.

## Fungsi dari dua Variabel

Untuk grafik sebuah fungsi, gunakan

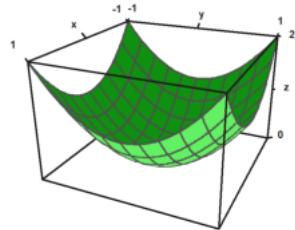
- ekspresi sederhana dalam x dan y,
- nama fungsi dari dua variabel
- atau matriks data.

Standarnya adalah kisi-kisi kawat yang terisi dengan warna yang berbeda di kedua sisi. Perhatikan bahwa jumlah default interval grid adalah 10, namun plot menggunakan jumlah default 40x40 persegi panjang untuk membangun permukaan. Hal ini dapat diubah.

- n=40, n=[40,40]: jumlah garis kisi di setiap arah
- grid=10, grid=[10,10]: jumlah garis grid di setiap arah.

Kami menggunakan default n=40 dan grid=10.

>plot3d("x^2+y^2");

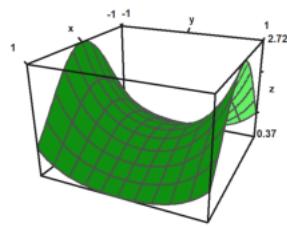


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-003.png

Interaksi pengguna dapat dilakukan dengan parameter >user. Pengguna dapat menekan tombol berikut ini.

- kiri, kanan, atas, bawah: memutar sudut pandang
- +,-: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l: beralih memutar sumber cahaya (lihat di bawah)
- spasi: mengatur ulang ke default
- kembali: mengakhiri interaksi

>plot3d("exp(-x^2-y^2)",>user,...  
>title="Turn with the vector keys (press return to finish)":



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-004.png

Rentang plot untuk fungsi dapat ditentukan dengan

- a, b: rentang x
- c, d: rentang y
- r: bujur sangkar simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

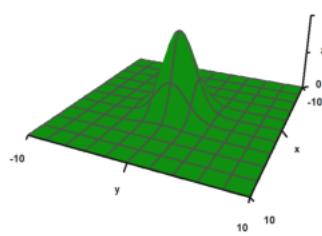
Terdapat beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: skala untuk nilai fungsi (standarnya adalah <fscale>).

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

frame: jenis bingkai (default 1).

```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-005.png

Tampilan dapat diubah dengan berbagai cara.

- jarak: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut ke sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.

Nilai default dapat diperiksa atau diubah dengan fungsi view(). Fungsi ini mengembalikan parameter dalam urutan di atas.

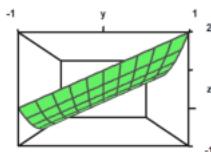
```
>view
```

```
[5, 2.6, 2, 0.4]
```

Jarak yang lebih dekat membutuhkan zoom yang lebih sedikit. Efeknya lebih seperti lensa sudut lebar.

Dalam contoh berikut ini, sudut = 0 dan tinggi = 0 terlihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

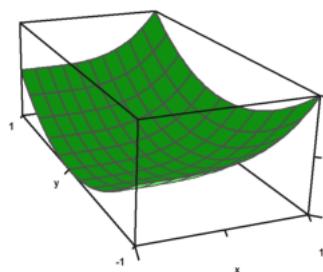
```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-006.png

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

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

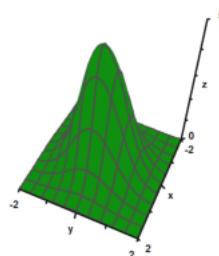


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-007.png

Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi, tidak perlu mengubah jarak atau melakukan zoom, tergantung pada ukuran plot. Namun demikian, label mengacu ke ukuran yang sesungguhnya.

Jika Anda menonaktifkannya dengan scale=false, Anda harus berhati-hati agar plot tetap muat di dalam jendela plotting, dengan mengubah jarak tampilan atau zoom, dan memindahkan bagian tengahnya.

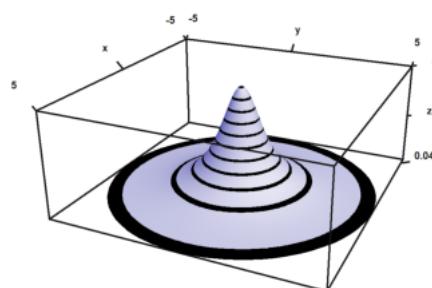
```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
>center=[0,0,-2],frame=3);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-008.png

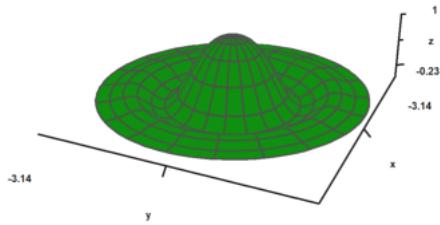
Plot polar juga tersedia. Parameter polar=true menggambar plot polar. Fungsi harus tetap merupakan fungsi dari x dan y. Parameter “fscale” menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi akan diskalakan agar sesuai dengan kubus.

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-009.png

```
>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):
```

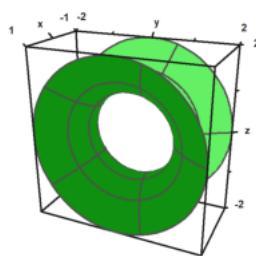


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-010.png

Parameter rotate memutar fungsi dalam x di sekitar sumbu x.

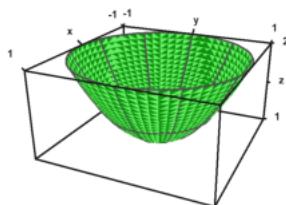
- rotate = 1: Menggunakan sumbu x
- rotate=2: Menggunakan sumbu z

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



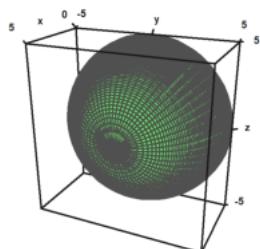
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-011.png

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



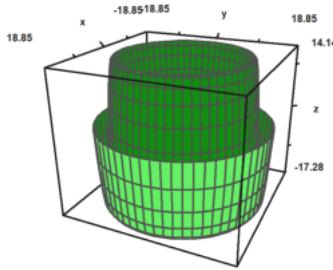
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-012.png

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



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-013.png

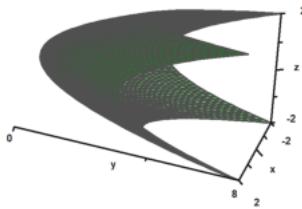
```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-014.png

Berikut ini adalah plot dengan tiga fungsi.

>plot3d("x","x<sup>2</sup>+y<sup>2</sup>","y",r=2,zoom=3.5,frame=3):



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-015.png

## Plot Kontur

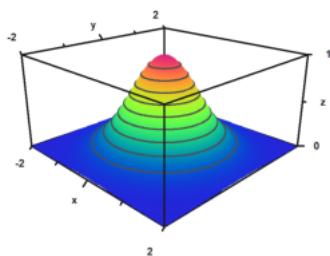
Untuk plot, Euler menambahkan garis kisi-kisi. Sebagai gantinya, dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona berwarna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan bayangan. Pada semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.

- Rona: Mengaktifkan bayangan cahaya, bukan kabel.
- >contour: Memplot garis kontur otomatis pada plot.
- level=... (atau level): Vektor nilai untuk garis kontur.

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

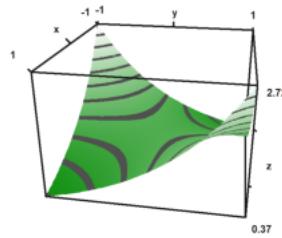
Gaya default dapat diubah. Untuk plot kontur berikut ini, kami menggunakan grid yang lebih halus untuk titik 100x100, skala fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

>plot3d("exp(-x<sup>2</sup>-y<sup>2</sup>)",r=2,n=100,level="thin", ...  
>>contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-016.png

>plot3d("exp(x\*y)",angle=100°,>contour,color=green):



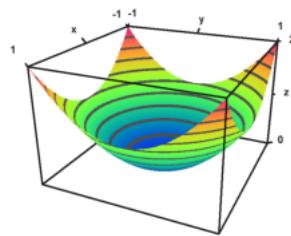
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-017.png

Bayangan default menggunakan warna abu-abu. Tetapi, kisaran warna spektral juga tersedia.

- >spektral: Menggunakan skema spektral default
- color =...: Menggunakan warna khusus atau skema spektral

Untuk plot berikut ini, kami menggunakan skema spektral default dan menambah jumlah titik untuk mendapatkan tampilan yang sangat mulus.

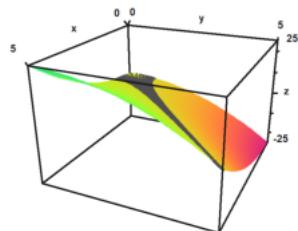
>plot3d("x^2+y^2",>spectral,>contour,n=100);



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-018.png

Alih-alih garis level otomatis, kita juga dapat menetapkan nilai garis level. Hal ini akan menghasilkan garis level yang tipis, alih-alih rentang level.

>plot3d("x^2-y^2",0.5,0.5,level=-1:0.1:1,color=redgreen);

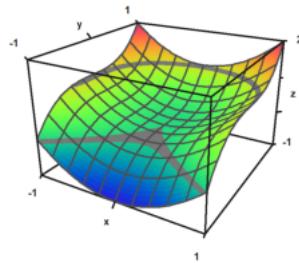


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-019.png

Pada plot berikut ini, 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-batas level sebagai kolom.

Selain itu, kami menghamparkan grid 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);

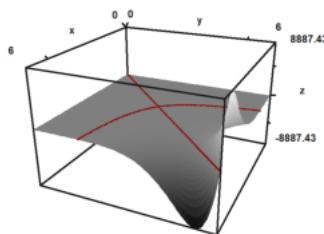


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-020.png

Pada contoh berikut, kami memplot himpunan, di mana

Kita menggunakan satu garis tipis untuk garis level.

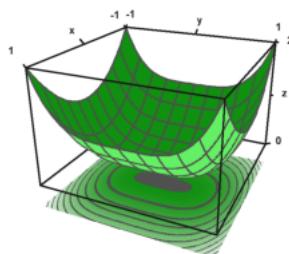
>plot3d("x<sup>y</sup>-y<sup>x</sup>",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100);



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-021.png

Dimungkinkan untuk menampilkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

>plot3d("x<sup>2</sup>+y<sup>4</sup>",>cp,cpcolor=green,cpdelta=0.2);



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-022.png

Berikut ini beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi-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);
```

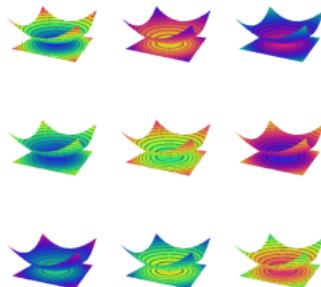


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-023.png

Ada beberapa skema spektral lainnya, yang diberi nomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan color=value, di mana value

- spektral: untuk rentang dari biru ke merah
- putih: untuk rentang yang lebih redup
- kuningbiru, ungu-hijau, biru-kuning, hijau-merah
- biru-kuning, hijau-ungu, kuning-biru, merah-hijau

```
>figure(3,3); ...
>for i=1:9; ...
> figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4); ...
> end; ...
> figure(0);
```



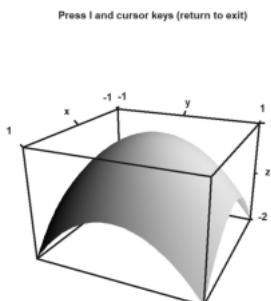
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-024.png

Sumber cahaya dapat diubah dengan l dan tombol kursor selama interaksi pengguna. Ini juga dapat ditetapkan dengan parameter.

- light: arah cahaya
- amb: cahaya sekitar antara 0 dan 1

Perhatikan, bahwa program ini tidak membuat perbedaan di antara sisi-sisi plot. Tidak ada bayangan. Untuk ini Anda akan membutuhkan Povray.

```
>plot3d("-x^2-y^2", ...
> hue=true,light=[0,1,1],amb=0,user=true, ...
> title="Press l and cursor keys (return to exit)":
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-025.png

Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

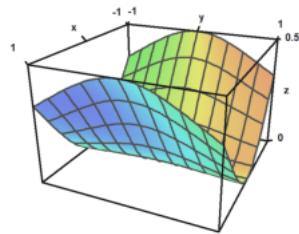
```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
> zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-026.png

Warna 0 memberikan efek pelangi yang istimewa.

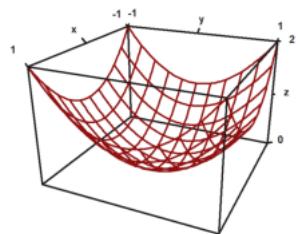
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10);



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-027.png

Permukaannya juga bisa transparan.

>plot3d("x^2+y^2",>transparent,grid=10,wirecolor=red);



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-028.png

## Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari sebuah fungsi dalam tiga variabel.

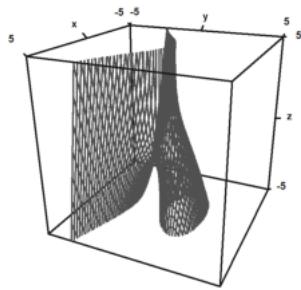
Solusi dari

dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, bidang x-z, dan bidang y-z.

- implisit = 1: potong sejajar dengan bidang-y-z
- implicit = 2: memotong sejajar dengan bidang x-z
- implicit=4: memotong sejajar dengan bidang x-y

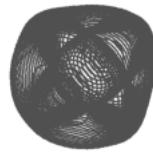
Tambahkan nilai-nilai ini, jika Anda mau. Pada contoh, kami memplot

>plot3d("x^2+y^3+z\*y-1",r=5,implicit=3);



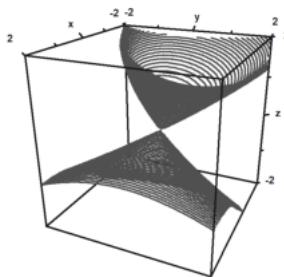
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-029.png

```
>c=1; d=1;
>plot3d(~((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)-d",r=2,
<frame,>implicit,>user):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-030.png

```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-031.png

## Memplot Data 3D

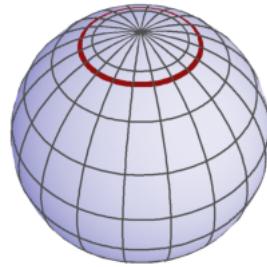
Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu menyediakan matriks nilai x, y, dan z, atau tiga fungsi atau ekspresi  $f_x(x,y)$ ,  $f_y(x,y)$ ,  $f_z(x,y)$ .

Karena x,y,z adalah matriks, kita mengasumsikan bahwa  $(t,s)$  berjalan melalui kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang dalam ruang.

Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

Pada contoh berikut, kita menggunakan vektor nilai  $t$  dan vektor kolom nilai  $s$  untuk memparameterkan permukaan bola. Pada gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

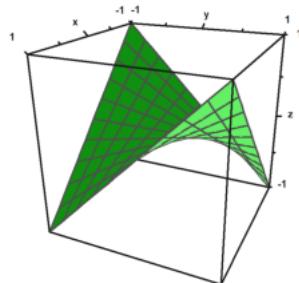
```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...
>plot3d(x,y,z,>hue, ...
>color=blue,<frame,grid=[10,20], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.4,height=50°):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-032.png

Berikut ini adalah contoh, yang merupakan grafik suatu fungsi.

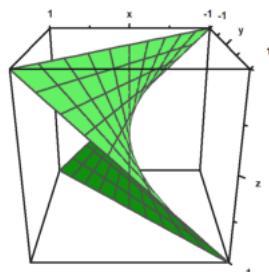
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-033.png

Namun demikian, kita bisa membuat segala macam permukaan. Berikut ini adalah permukaan yang sama dengan fungsi

```
>plot3d(t*s,t,s,angle=180°,grid=10);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-034.png

Dengan lebih banyak upaya, kita bisa menghasilkan banyak permukaan.

Dalam contoh berikut ini, kami membuat tampilan berbayang dari bola yang terdistorsi. Koordinat yang biasa digunakan untuk bola adalah

dengan

Kami mengubahnya 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);
```

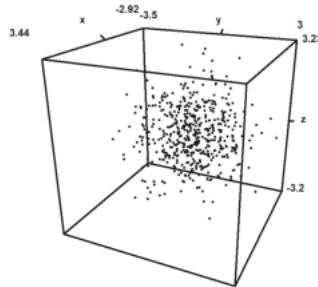


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-035.png

Tentu saja, awan titik juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor untuk koordinat titik.

Gaya-gayanya sama seperti pada plot2d dengan points=true;

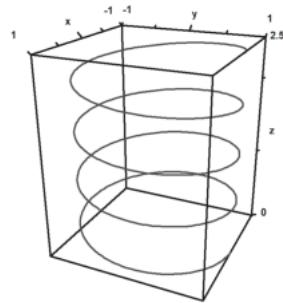
```
>n=500; ...
>plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=".");
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-036.png

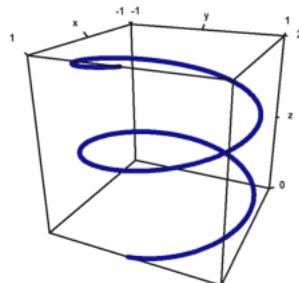
Anda juga dapat memplot kurva dalam bentuk 3D. Dalam hal ini, akan lebih mudah untuk menghitung titik-titik kurva. Untuk kurva pada bidang, kami menggunakan urutan koordinat dan parameter wire = true.

```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3);
```



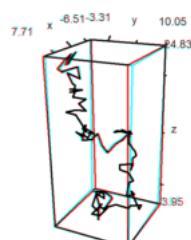
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-037.png

```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
> linewidth=3, wirecolor=blue);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-038.png

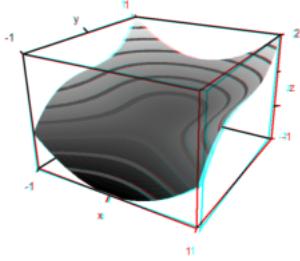
```
>X=cumsum(normal(3,100)); ...
>plot3d(X[1],X[2],X[3],>anaglyph,>wire);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-039.png

EMT juga dapat membuat plot dalam mode anaglyph. Untuk melihat plot semacam itu, Anda memerlukan kacamata merah/cyan.

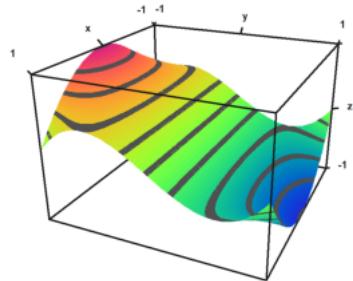
```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-040.png

Sering kali, skema warna spektral digunakan untuk plot. Hal ini menekankan ketinggian fungsi.

```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```

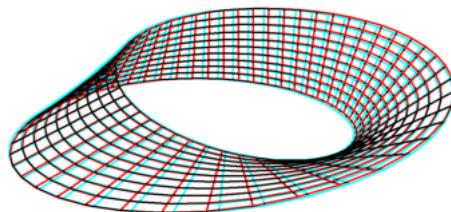


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-041.png

Euler juga dapat memplot permukaan yang diparameterkan, ketika parameternya adalah nilai x, y, dan z dari gambar kisi-kisi persegi panjang di dalam ruang.

Untuk demo berikut ini, kami menyiapkan parameter u dan v, dan menghasilkan koordinat ruang dari parameter ini.

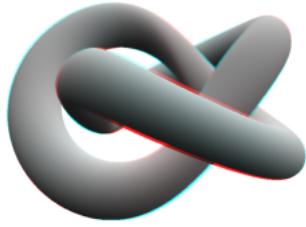
```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...
> X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
> plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-042.png

Berikut ini contoh yang lebih rumit, yang tampak megah dengan kacamata merah/cyan.

```
>u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)'; ...
> x:=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...
> y:=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...
> z=sin(u)+2*cos(3*v); ...
> plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-043.png

## Plot Statistik

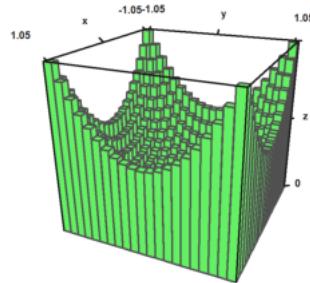
Plot batang juga dapat digunakan. Untuk ini, kita harus menyediakan

- x: vektor baris dengan n+1 elemen
- y: vektor kolom dengan n+1 elemen
- z: matriks nilai berukuran nxn.

z dapat lebih besar, tetapi hanya nilai nxn yang akan digunakan.

Pada contoh, pertama-tama kita menghitung nilainya. Kemudian kita sesuaikan x dan y, sehingga vektor berada di tengah-tengah nilai yang digunakan.

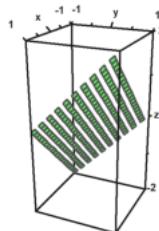
```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...
>plot3d(xa,ya,z,bar=true);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-044.png

Dimungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

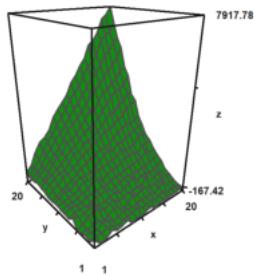
```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-045.png

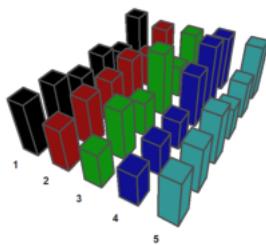
Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan scale(M), atau menskalakan matriks dengan >zscale. Hal ini dapat dikombinasikan dengan faktor penskalaan individual yang diterapkan sebagai tambahan.

```
>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-046.png

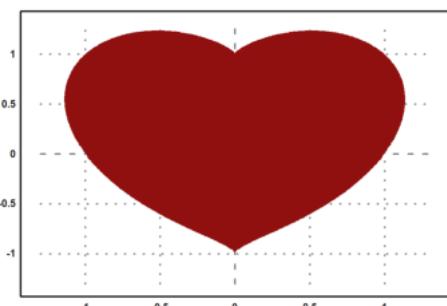
```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
> loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
> columnsplot3d(v',scols=1:5,ccols=[1:5]);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-047.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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-048.png

```
>ekspresso &= (x^2+y^2-1)^3-x^2*y^3; $ekspresso
```

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

Kami ingin memutar kurva jantung di sekitar sumbu y. Berikut ini adalah ekspresso yang mendefinisikan jantung:

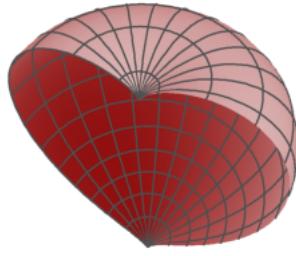
Selanjutnya kita atur

```
>function fr(r,a) &= ekspresso with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2\sin a)r^5}{16}$$

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang menyelesaikan untuk r, jika a diberikan. Dengan fungsi tersebut kita dapat memplotkan jantung yang diputar sebagai permukaan parametrik.

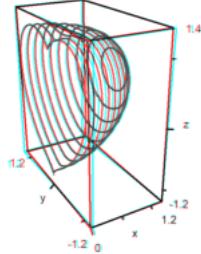
```
>function map f(a) := bisect("fr",0,2;a); ...
> t=linspace(-pi/2,pi/2,100); r=f(t); ...
> s=linspace(pi,2pi,100)';
> plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
> >hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°);
```



Berikut ini adalah plot 3D dari gambar di atas yang diputar mengelilingi sumbu-z. Kami mendefinisikan fungsi, yang menggambarkan objek.

```
>function f(x,y,z) ...
r=x^2+y^2;
return (r+z^2-1)^3-r*z^3;
endfunction

>plot3d("f(x,y,z)", ...
> xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...
> implicit=1,angle=-30°,zoom=2.5,n=[10,100,60],>anaglyph):
```



[images/23030630004\\_Siti%20Faltipah%20Hayani\\_EMT4Plot3D-052.png](#)

## Plot 3D Khusus

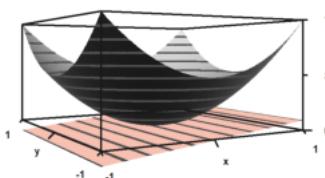
Fungsi plot3d memang bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda juga bisa mendapatkan plot berbingkai dari objek apa pun yang Anda sukai.

Meskipun Euler bukan program 3D, namun dapat menggabungkan beberapa objek dasar. Kami mencoba memvisualisasikan parabola dan garis singgungnya.

```
>function myplot ...
y=-1:0.01:1; x=(-1:0.01:1)';
plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ..
    hues=0.5,>contour,color=orange);
h=holding(1);
plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
holding(h);
endfunction
```

Sekarang framedplot() menyediakan frame, dan mengatur tampilan.

```
>framedplot("myplot",[-1,1,-1,1,0,1],height=0,angle=-30°, ...
> center=[0,0,-0.7],zoom=3):
```



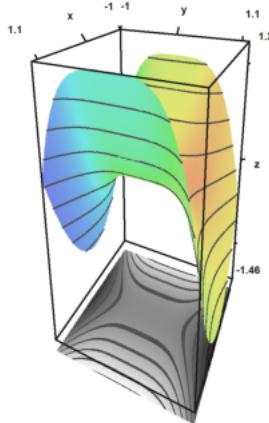
[images/23030630004\\_Siti%20Faltipah%20Hayani\\_EMT4Plot3D-053.png](#)

Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d() mengatur jendela ke fullwindow() secara default, namun plotcontourplane() mengasumsikannya.

```

>x=-1:0.02:1.1; y=x'; z=x^2-y4;
>
>function myplot (x,y,z) ...
>
    zoom(2);
    wi=fullwindow();
    plotcontourplane(x,y,z,level="auto",hue,add,color=white
    window(wi);
    reset();
endfunction
◀ ▶
>myplot(x,y,z);

```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-054.png

## Animasi

Euler dapat menggunakan frame untuk melakukan pra-komputasi animasi.

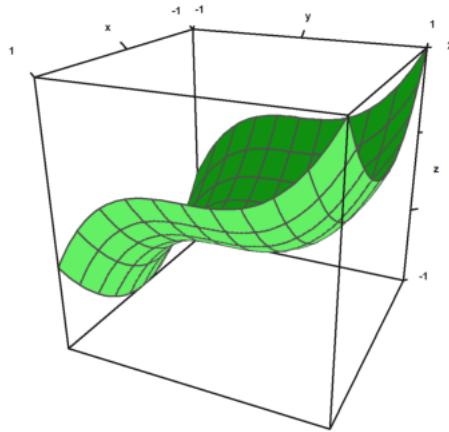
Salah satu fungsi yang memanfaatkan teknik ini adalah rotate. Fungsi ini dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi ini memanggil addpage() untuk setiap plot baru. Akhirnya fungsi ini menganimasikan plot tersebut.

Silakan pelajari sumber dari rotate untuk melihat lebih detail.

```

>function testplot () := plot3d("x^2+y3"); ...
>rotate("testplot"); testplot();

```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-055.png

## Menggambar Povray

With the help of the Euler file povray.e, Euler can generate Povray files. The results are very nice to look at.

You need to install Povray (32bit or 64bit) from <http://www.povray.org/>, and put the sub-directory "bin" of Povray into the environment path, or set the variable "defaultpovray" with full path pointing to "povengine.exe".

put the sub-directory “bin” of Povray into the environment path, or set the variable “defaultpovray” with full path pointing to “pvengine.exe”.

The Povray interface of Euler generates Povray files in the home directory of the user, and calls Povray to parse these files. The default file name is current.pov, and the default directory is eulerhome(), usually c:. Povray generates a PNG file, which can be loaded by Euler into a notebook. To clean up these files, use povclear().

The pov3d function is in the same spirit as plot3d. It can generate the graph of a function f(x,y), or a surface with coordinates X,Y,Z in matrices, including optional level lines. This function starts the raytracer automatically, and loads the scene into the Euler notebook.

Besides pov3d(), there are many functions, which generate Povray objects. These functions return strings, containing the Povray code for the objects. To use these functions, start the Povray file with povstart(). Then use writeln(...) to write the objects to the scene file. Finally, end the file with povend(). By default, the raytracer will start, and the PNG will be inserted into the Euler notebook.

The object functions have a parameter called “look”, which needs a string with Povray code for the texture and the finish of the object. The function povlook() can be used to produce this string. It has parameters for the color, the transparency, Phong Shading etc.

Note that the Povray universe has another coordinate system. This interface translates all coordinates to the Povray system. So you can keep thinking in the Euler coordinate system with z pointing vertically upwards, and x,y,z axes in right hand sense.

You need to load the povray file.

## 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” dari Povray ke dalam jalur lingkungan, atau mengatur 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 defaultnya adalah eulerhome(), biasanya c:. Povray menghasilkan sebuah file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan berkas-berkas ini, gunakan povclear().

Fungsi pov3d memiliki semangat yang sama dengan plot3d. Fungsi ini dapat menghasilkan grafik dari sebuah fungsi f(x,y), atau sebuah permukaan dengan koordinat X,Y,Z dalam bentuk matriks, termasuk garis-garis level yang bersifat opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat adegan ke dalam notebook Euler.

Selain pov3d(), ada banyak fungsi yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string, yang berisi kode Povray untuk objek. Untuk menggunakan fungsi-fungsi ini, mulai file Povray dengan povstart(). Kemudian gunakan writeln(...) untuk menulis objek ke file scene. Terakhir, akhiri file dengan povend(). Secara default, raytracer akan dimulai, dan PNG akan dimasukkan ke dalam buku catatan Euler.

Fungsi objek memiliki parameter yang disebut “look”, yang membutuhkan string dengan kode povray untuk tekstur dan hasil akhir objek. Fungsi povlook() dapat digunakan untuk menghasilkan string ini. Fungsi ini memiliki parameter untuk warna, transparansi, Phong Shading, dll.

Perhatikan bahwa Povray universe memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat tetap berpikir dalam sistem koordinat Euler dengan z yang mengarah vertikal ke atas, dan sumbu x, y, z di tangan kanan.

Anda perlu memuat file povray.

>load povray;

Pastikan direktori bin povray berada di dalam path. Jika tidak, edit variabel berikut sehingga berisi jalur ke povray yang dapat dieksekusi.

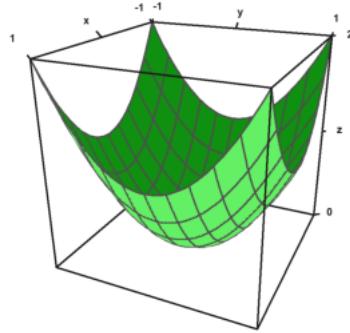
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"

C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe

Untuk kesan pertama, kita plot sebuah fungsi sederhana. Perintah berikut ini menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk melacak sinar pada file ini.

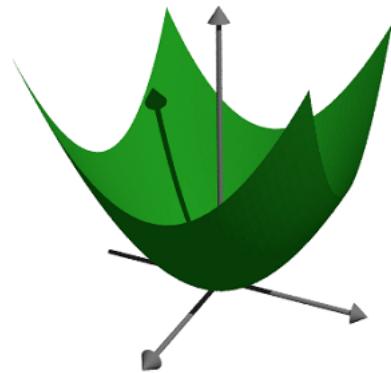
Jika Anda memulai perintah berikut, GUI Povray akan terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya, apakah Anda ingin mengizinkan file exe dijalankan. Anda dapat menekan cancel untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK pada jendela Povray untuk mengetahui dialog awal Povray.

>plot3d("x^2+y^2",zoom=2);



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-056.png

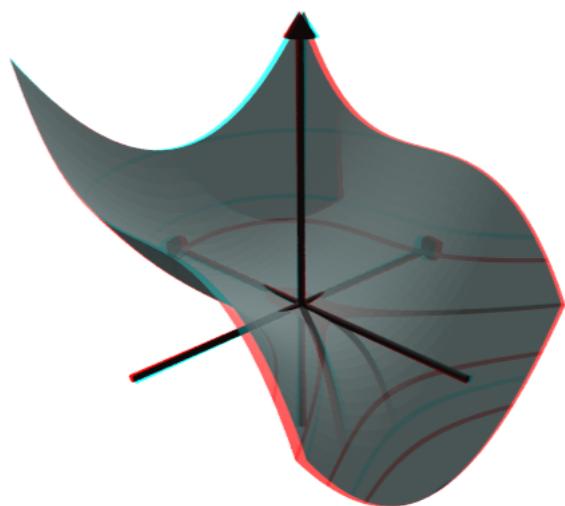
```
>pov3d("x^2+y^2",zoom=3);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-057.png

Kita dapat membuat fungsi menjadi transparan dan menambahkan hasil akhir lainnya.  
Kita juga dapat menambahkan garis level ke plot fungsi.

```
>pov3d("x^2+y^2",axiscolor=red,angle=-45°,>anaglyph, ...  
>look=povlook(cyan,0.2),level=-1:0.5:1,zoom=3.8);
```

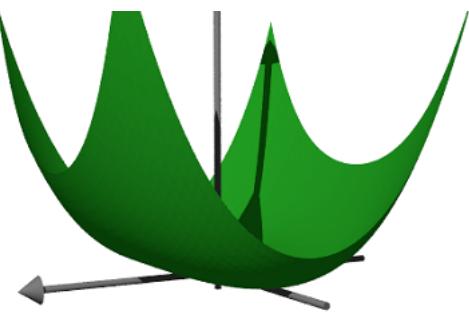


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-058.png

Kadang-kadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.

Kami memplot kumpulan titik pada bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y2)*((x+1)^2+y2)/40",r=2, ...
> angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=10°,n=50, ...
><fscale,zoom=3.8);
```



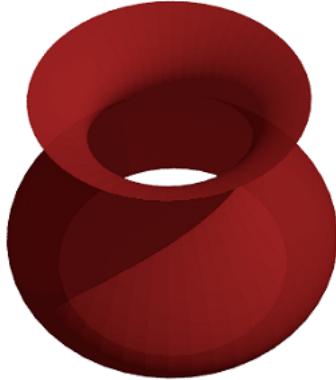
images/23030630004\_\_Siti%20Faltipah%20Hayani\_\_EMT4Plot3D-059.png

## Merencanakan dengan Koordinat

Sebagai pengganti fungsi, kita dapat membuat plot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objek.

Pada contoh, kita memutar sebuah fungsi pada sumbu z.

```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,50)';
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light=
[10,5,15]);
```

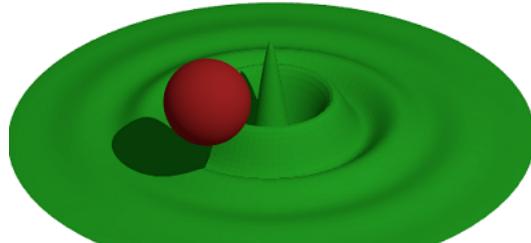


images/23030630004\_\_Siti%20Faltipah%20Hayani\_\_EMT4Plot3D-060.png

Pada contoh berikut, kita 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=6,axis=0,height=30°,add=povsphere([0.5,0,0.25],0.15,povlook(red)),
...
>w=500,h=300);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-061.png

Dengan metode bayangan canggih Povray, hanya sedikit titik yang bisa menghasilkan permukaan yang sangat halus. Hanya pada batas-batas dan bayangan, trik ini bisa terlihat jelas.

Untuk itu, 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 terhadap x dan y dari persamaan ini dan mengambil hasil perkalian silang sebagai normal.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

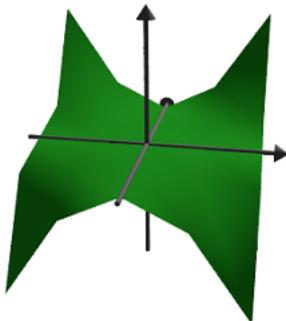
Kami mendefinisikan normal sebagai hasil kali silang dari turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

$$\begin{matrix} 3 & 2 & 2 \\ [-2 \times y, -3 \times y, 1] \end{matrix}$$

Kami hanya menggunakan 25 poin.  $>x=-1:0.5:1; y=x^*$ ;

```
>pov3d(x,y,Z(x,y),angle=10°, ...
>xv=NX(x,y),yv=NY(x,y),zv=NZ(x,y),<shadow);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-062.png

Berikut ini adalah simpul Trefoil yang dibuat oleh A. Busser di Povray. Ada versi yang lebih baik dari ini dalam contoh.

#### Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kita tambahkan vektor normal di sini. Kita menggunakan Maxima untuk menghitung normal untuk kita. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolik.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
>Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
>Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunan terhadap x dan y.

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang yang normal, yang merupakan produk silang dari dua turunan.

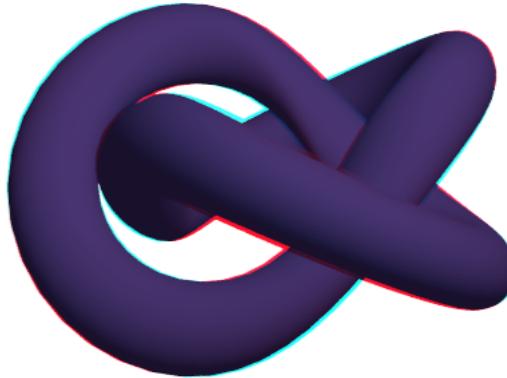
```
>dn &= crossproduct(dx,dy);
```

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 &“ekspresi”(parameter). Ini adalah sebuah alternatif dari metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik  $NX, NY, NZ$  terlebih dahulu.

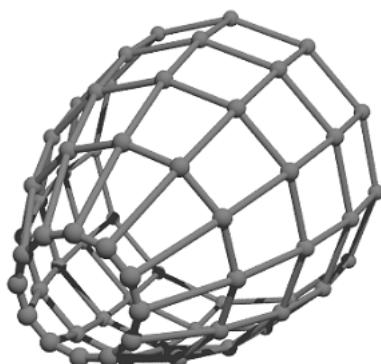
```
>pov3d(X(x,y),Y(x,y),Z(x,y),>anaglyph,axis=0,zoom=5,w=450,h=350, ...
><shadow,look=povlook(blue), ...
>xv=&“dn[1]”(x,y), yv=&“dn[2]”(x,y), zv=&“dn[3]”(x,y));
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-063.png

Kami juga dapat menghasilkan kisi-kisi dalam bentuk 3D.

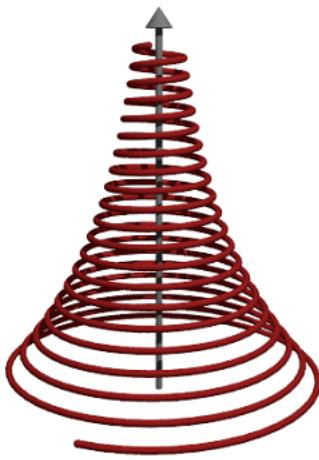
```
>povstart(zoom=4); ...
>x=-1:0.5:1; r=1-(x+1)^2/6; ...
>t=(0°:30°:360°); y=r*cos(t); z=r*sin(t); ...
>writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...
>povend();
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-064.png

Dengan povgrid(), kurva dapat dibuat.

```
>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-065.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.

Kita mulai output dengan povstart().

```
>povstart(zoom=4);
```

Pertama, kita mendefinisikan tiga silinder, dan menyimpannya dalam bentuk string di Euler.

Fungsi povx() dll. hanya mengembalikan vektor [1,0,0], yang dapat digunakan sebagai gantinya.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...
>c2=povcylinder(-povy,povy,1,povlook(yellow)); ...
>c3=povcylinder(-povz,povz,1,povlook(blue)); ...
>
```

String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

```
>c2
```

```
cylinder { <0,0,-1>, <0,0,1>, 1
    texture { pigment { color rgb <0.941176,0.941176,0.392157> } }
    finish { ambient 0.2 }
}
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.

Hal ini dilakukan dengan povlook(), yang mengembalikan sebuah string dengan kode Povray yang relevan. Kita dapat menggunakan warna default Euler, atau menentukan warna kita sendiri. Kita juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

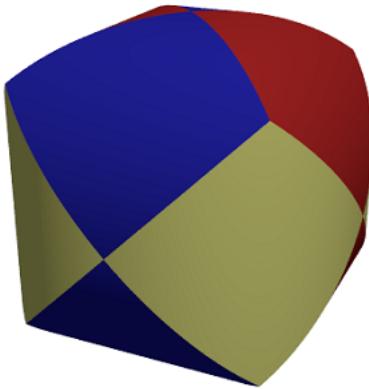
```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> } }
finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek perpotongan, dan menulis hasilnya ke file.

```
>writeln(povintersection([c1,c2,c3]));
```

Perpotongan tiga silinder sulit untuk divisualisasikan, jika Anda belum pernah melihatnya.

```
>povend;
```



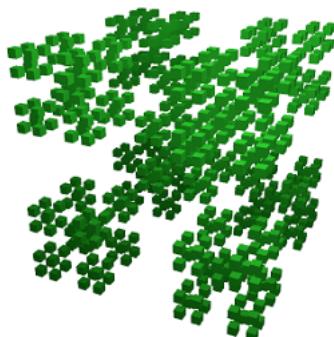
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-066.png

Fungsi-fungsi berikut ini menghasilkan fractal secara rekursif.

Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana.  
Fungsi povbox() mengembalikan sebuah string, yang 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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-067.png

Perbedaan memungkinkan pemotongan satu objek dari objek lainnya. Seperti persimpangan, ada bagian dari objek CSG Povray.

```
>povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kita akan mendefinisikan sebuah objek di Povray, alih-alih menggunakan sebuah string di Euler. Definisi akan langsung dituliskan ke file.

Koordinat kotak -1 berarti [-1,-1,-1].

```
>povdefine("mycube",povbox(-1,1));
```

Kita dapat menggunakan objek ini dalam povobject(), yang mengembalikan sebuah string seperti biasa.

```
>c1=povobject("mycube",povlook(red));
```

Kami menghasilkan kubus kedua, dan memutar serta menskalakannya sedikit.

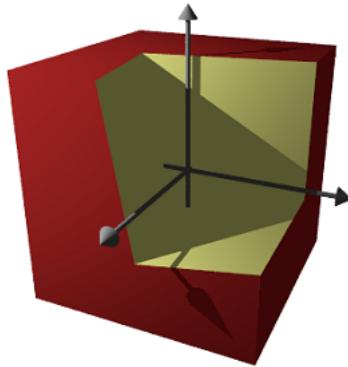
```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...  
>rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita ambil selisih dari 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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-068.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.

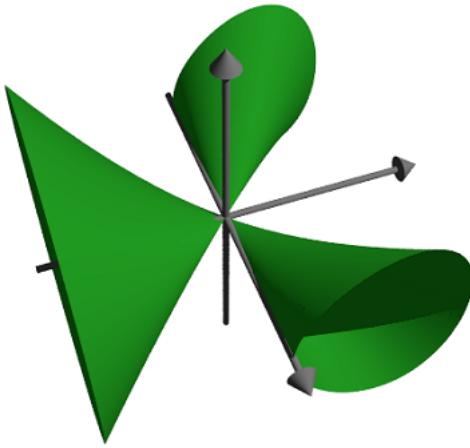
Sintaks untuk fungsi-fungsi ini sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

```
>povstart(angle=70°,height=50°,zoom=4);  
  
>c=0.1; d=0.1; ...  
> writeln(povsurface("(pow(pow(x,2)+pow(y,2)-pow(c,2),2)+pow(pow(z,2)-1,2))*  
(pow(pow(y,2)+pow(z,2)-pow(c,2),2)+pow(pow(x,2)-1,2))*(pow(pow(z,2)+pow(x,2)-  
pow(c,2),2)+pow(pow(y,2)-1,2))-d",povlook(red))); ...  
> povend();  
  
Error : Povray error!  
  
Error generated by error() command  
  
povray:  
    error("Povray error!");  
Try "trace errors" to inspect local variables after errors.  
povend:  
    povray(file,w,h,aspect,exit);  
  
>povstart(angle=25°,height=10°);  
>writeln(povsurface("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,2,"")));  
>povend();
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-069.png

```
>povstart(angle=70°,height=50°,zoom=4);
Membuat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi.
>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...
> writeAxes(); ...
> povend();
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-070.png

## Objek Jaring

Pada contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.

Kami ingin memaksimalkan xy di bawah kondisi  $x+y = 1$  dan mendemonstrasikan sentuhan tangensial dari garis level.

```
>povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kita tidak dapat menyimpan objek dalam sebuah string seperti sebelumnya, karena ukurannya terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan #declare. Fungsi povtriangle() melakukan hal ini secara otomatis. Fungsi ini dapat menerima vektor normal seperti halnya pov3d().

Berikut ini mendefinisikan objek mesh, dan menuliskannya langsung 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 kita tentukan dua cakram, 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);
```

Tuliskan permukaan dikurangi kedua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tuliskan kedua perpotongan tersebut.

```

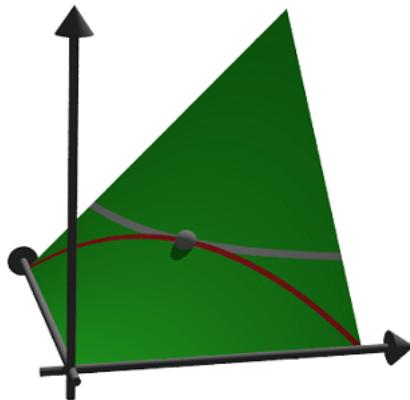
>writeln(povintersection([mesh,cl],povlook(red))); ...
> writeln(povintersection([mesh,ll],povlook(gray)));
Tulislah satu titik secara maksimal.

>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));

Tambahkan sumbu dan selesaikan.

>writeAxes(0,1,0,1,0,1,d=0.015); ...
> povend();

```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-071.png

### Anaglyph dalam 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().

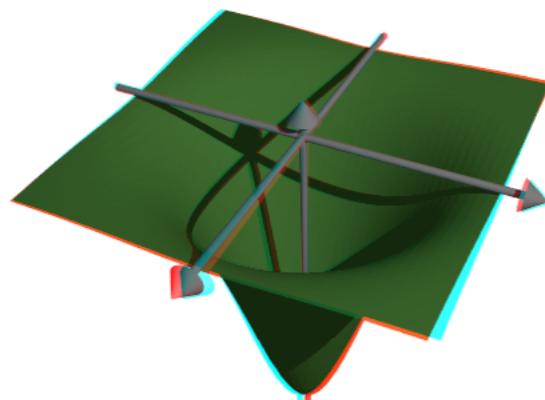
Tentu saja, Anda membutuhkan kacamata merah/cyan untuk melihat contoh berikut dengan benar.

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);

```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-072.png

Jika Anda membuat scene dengan objek, Anda harus menempatkan pembuatan scene ke dalam suatu fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

```

>function myscene ...
s=povsphere(povc,1);
c1=povcylinder(-povz,povz,0.5);
c1x=povobject(c1,rotate=xrotate(90°));

```

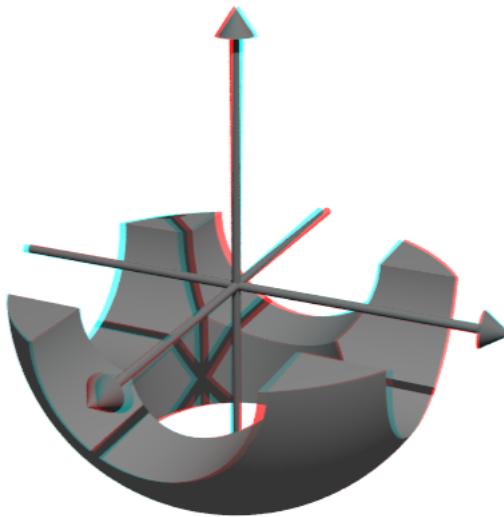
```

c1y=povobject(c1,rotate=yrotate(90°));
c=povbox([-1,-1,0],1);
un=povunion([c1,c1x,c1y,c]);
obj=povdifference(s,un,povlook(red));
writeln(obj);
writeAxes();
endfunction

```

Fungsi povanaglyph() melakukan semua ini. Parameter-parameternya seperti pada povstart() dan povend() yang digabungkan.

```
>povanaglyph("myscene",zoom=4.5);
```



[images/23030630004\\_Siti%20Faltipah%20Hayani\\_EMT4Plot3D-073.png](#)

## Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak objek. Namun Anda tidak dibatasi pada objek-objek tersebut. Anda dapat membuat objek sendiri, yang menggabungkan objek-objek lain, atau objek yang benar-benar baru.

Kami mendemonstrasikan sebuah torus. Perintah Povray untuk ini adalah “torus”. Jadi kita mengembalikan sebuah string dengan perintah ini dan parameteranya. Perhatikan bahwa torus selalu berpusat pada titik asal.

```

>function povdonat (r1,r2,look="") ...
    return "torus {" + r1 + "," + r2 + look + "}";
endfunction

```

Inilah torus pertama kami.

```
>t1=povdonat(0.8,0.2)
```

```
torus {0.8,0.2}
```

Mari kita gunakan objek ini untuk membuat torus kedua, ditranslasikan dan diputar.

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
object { torus {0.8,0.2}
    rotate 90 *x
    translate <0.8,0,0>;
}
```

Sekarang, kita tempatkan semua benda ini ke dalam suatu pemandangan. Untuk tampilannya, kami menggunakan Phong Shading.

```

>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...
> writeln(povobject(t1,povlook(green,phong=1))); ...
> writeln(povobject(t2,povlook(green,phong=1))); ...
>
>povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, program ini tidak menampilkan kesalahan. Oleh karena itu, Anda harus menggunakan

```
>povend(<exit>);
```

jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray tetap terbuka.

```
>povend(h=320,w=480);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-074.png

Berikut adalah contoh yang lebih rumit. Kami menyelesaikan

lateks: Ax b, x , c.x

dan menunjukkan titik-titik yang layak dan optimal dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
```

```
>b=[10,10,10]';
```

```
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini memiliki solusi atau tidak.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0, 1, 0.5]
```

Ya, benar.

Selanjutnya kita mendefinisikan dua objek. Yang pertama adalah pesawat

```
>function oneplane (a,b,look=""") ...
```

```
    return povplane(a,b,look)
endfunction
```

Kemudian kita tentukan perpotongan semua setengah ruang dan kubus.

```
>function adm (A, b, r, look=""") ...
```

```
    ol=[];
    loop 1 to rows(A); ol=ol|oneplane(A[#,b[#]); end;
    ol=ol|povbox([0,0,0],[r,r,r]);
    return povintersection(ol,look);
endfunction
```

Sekarang, kita bisa merencanakan adegan tersebut.

```
>povstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); ...
> writeln(adm(A,b,2,povlook(green,0.4))); ...
> writeAxes(0,1.3,0,1.6,0,1.5); ...
>
```

Berikut ini adalah lingkaran di sekeliling optimal.

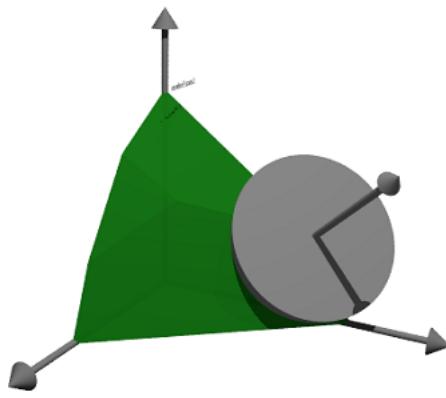
```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
> povlook(red,0.9)));
```

Dan kesalahan pada arah yang optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah sebuah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

```
>writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=5°)); ...
> povend();
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-075.png

## More Examples

You can find some more examples for Povray in Euler in the following files.

[Examples/Dandelin Spheres](#)

[Examples/Donut Math](#)

[Examples/Trefoil Knot](#)

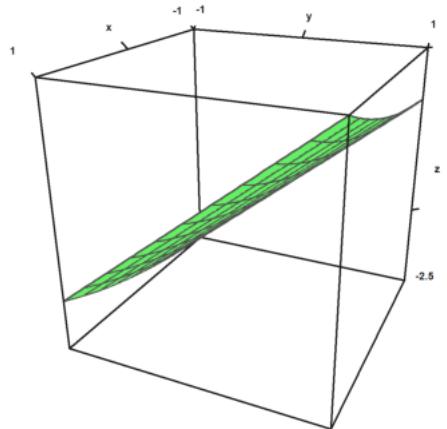
[Examples/Optimization by Affine Scaling](#)

### Latihan soal

Menggambar grafik fungsi berikut

```
>function f(x,y):= 2^x+3*y
```

```
>plot3d("f"):
```

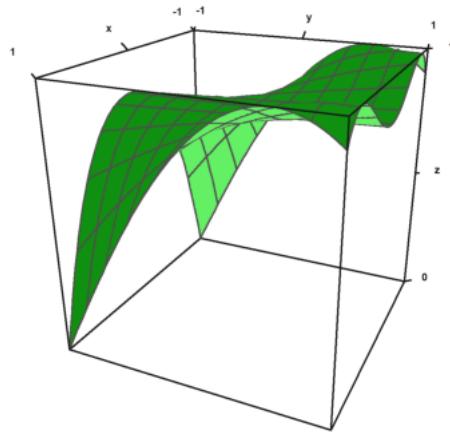


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-076.png

Gambarlah grafik fungsi berikut :

```
>function g(x,y):= sin(x^2*y+1);
```

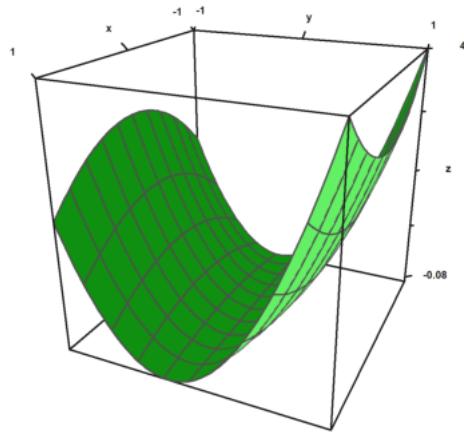
```
>plot3d("g"):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-077.png

Gambarlah plot3d dari persamaan berikut:

```
>function A(x,y)&= x^2*y+3*y^2;
>plot3d("A");
```

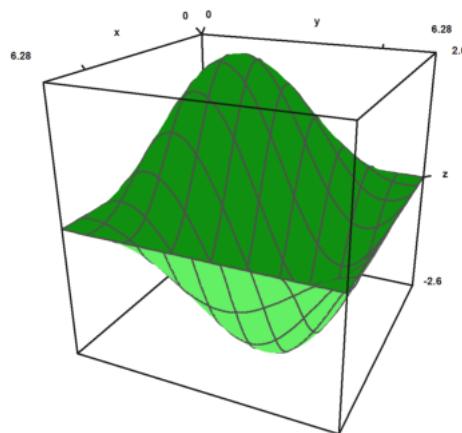


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-078.png

Plotlah grafik

pada  $0 \leq x, y \leq 2\pi$

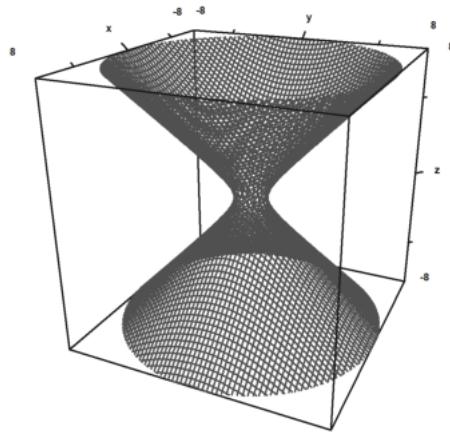
```
>plot3d("sin(x)+sin(y)-sin(x+y)",0,2*pi,0,2*pi);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-079.png

Carilah fungsi implisit berikut dalam 3D

```
>plot3d("x^2+y^2-z^2-1",r=8,implicit=3);
```

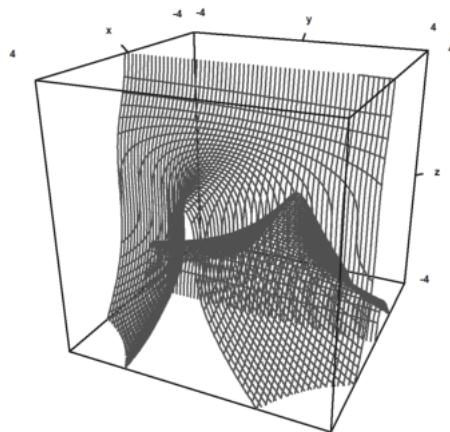


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-080.png

Gambarkan fungsi 3D dari fungsi implisit berikut ini

dengan  $r=4$

`>plot3d("x*y+x^3*y^2+x*z^3-9",r=4,implicit=3);`

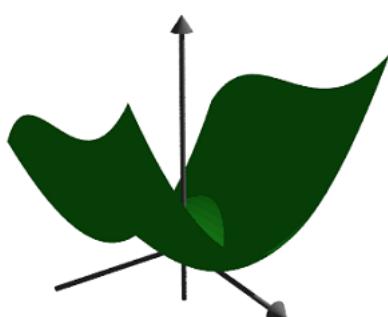


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-081.png

Buatlah plot 3D dari fungsi

dengan zoom 3 dan angle 55 derajat menggunakan povray

`>pov3d("x^3+y^2",zoom=3,angle=55°);`



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Plot3D-082.png

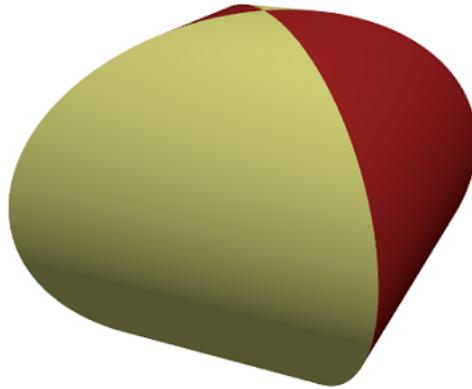
Buatlah gabungan 2 silinder dengan fungsi povx() berwarna merah dan povz() berwarna kuning dan zoom 4

`>povstart(zoom=4)`

```

>c1 = povcylinder(-povx,povx,1,povlook(red));
>c2 = povcylinder(-povz,povz,1,povlook(yellow));
>writeln(povintersection([c1,c2]));
>povend();

```



# Kalkulus dengan EMT —

Nama : Siti Faltipah Hayani

NIM : 23030630004

Materi Kalkulus mencakup di antaranya: \* Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi)

- Limit Fungsi,
- Turunan Fungsi,
- Integral Tak Tentu,
- Integral Tentu dan Aplikasinya,
- Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

## Mendefinisikan Fungsi

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:

- Menggunakan format `nama_fungsi := rumus fungsi` (untuk fungsi numerik),
- Menggunakan format `nama_fungsi &= rumus fungsi` (untuk fungsi simbolik, namun dapat dihitung secara numerik),
- Menggunakan format `nama_fungsi &&= rumus fungsi` (untuk fungsi simbolik murni, tidak dapat dihitung langsung),
- Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah `function` (bukan sebagai ekspresi).

Berikut adalah beberapa contoh cara mendefinisikan fungsi:

```

f(x) = 2x2 + esin(x).
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
1
4.31977682472
20.7392088022
>f(a) // tidak dapat dihitung nilainya
Variable or function a not found.
Error in:
f(a) // tidak dapat dihitung nilainya ...
^

```

Silakan Anda plot kurva fungsi di atas!

Berikutnya kita definisikan fungsi:

```
>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) ...
^
```

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);
>function f(x) &= 2*E^x // fungsi simbolik
```

```

>$f(a) // nilai fungsi secara simbolik
>f(E) // nilai fungsi berupa bilangan desimal
30.308524483
>$f(E), $float(%)
>function g(x) &= 3*x+1
3 x + 1
>function h(x) &= f(g(x)) // komposisi fungsi
3 x + 1
2 E
>plot2d("h(x)",-1,1);

```

## Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan fungsi-fungsi tersebut dan komposisinya di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik fungsi-fungsi tersebut dan komposisi-komposisi 2 fungsi.

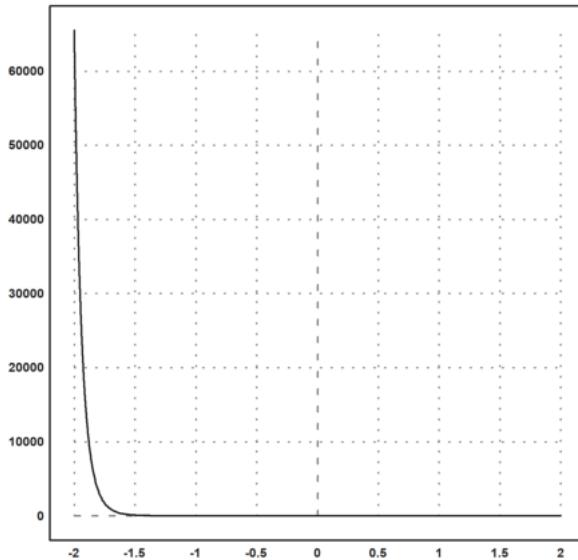
Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

$$a(x) = x + 5$$

```

>function a(x):= x+5;
terdapat fungsi
b(x) = x^3
>function b(x) := x^3;
terdapat fungsi
c(x) = 1/(4^x)
>function c(x) := 1/(4^x)
terdapat fungsi
d(x) = \sqrt{-}(9x)
>function d(x) := sqrt(9x)
terdapat fungsi
e(x) = sin(2x)
>function e(x):= sin(2x)
>function n(x):= c(b(x)) //mendefinisikan komposisi fungsi
>function m(x):= d(a(x)) //mendefinisikan komposisi fungsi
>function o(x):= e(b(x)) //mendefinisikan komposisi fungsi
>n(1)//fungsi komposisi terhadap satu nilai
0.25
>n(1:5)//fungsi komposisi terhadap vektor
[0.25, 1.52588e-05, 0, 0, 0]
>plot2d("n(x)": //Menggambar grafik fungsi

```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-007.png

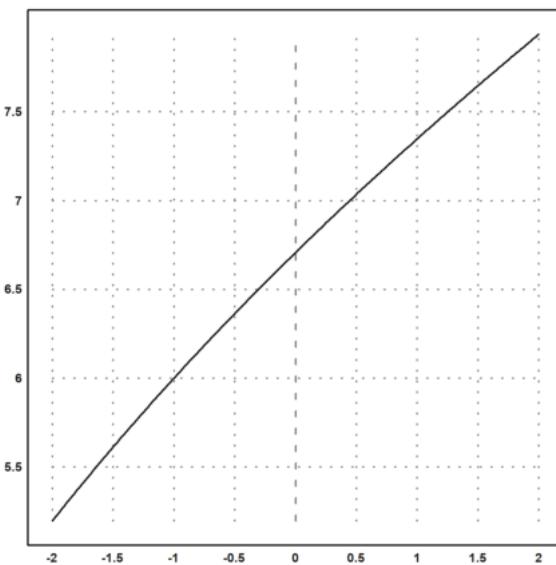
>m(2)

7.93725393319

>m(3:7)

[8.48528, 9, 9.48683, 9.94987, 10.3923]

>plot2d("m(x)":



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-008.png

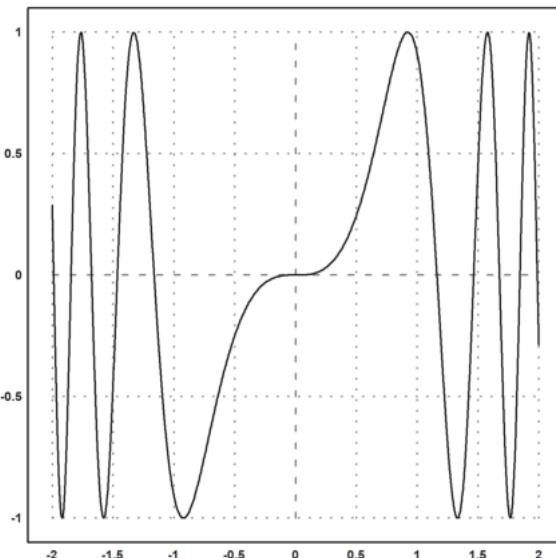
>o(5)

-0.970528019542

>o(5:10)

[-0.970528, -0.999519, 0.905604, -0.158533, 0.296484, 0.93004]

>plot2d("o(x)":



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-009.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))

$$\lim_{x \rightarrow \infty} \frac{\sqrt{x^2 - 3x}}{x + 1} = 1$$

>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)

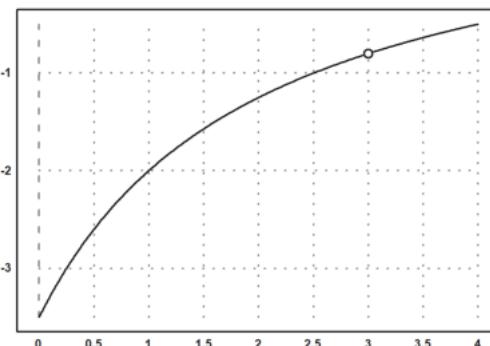
$$-\frac{4}{5}$$

```

maxima: 'limit((x^3-13\*x^2+51\*x-63)/(x^3-4\*x^2-3\*x+18),x,3)=limit((x^3-13\*x^2+51\*x-63)/(x^3-4\*x^2-3\*x+18),x,3)

Fungsi tersebut diskontinu di titik  $x=3$ . Berikut adalah grafik fungsinya.

```
>aspect(1.5); plot2d("(x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18)",0,4);
plot2d(3,-4/5,>points,style="ow",>add);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-012.png

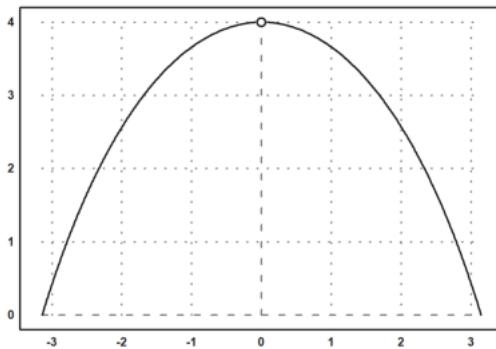
```
>$limit(2*x*sin(x)/(1-cos(x)),x,0)
```

4

maxima: 'limit(2\*x\*sin(x)/(1-cos(x)),x,0)=limit(2\*x\*sin(x)/(1-cos(x)),x,0)

Fungsi tersebut diskontinu di titik  $x=0$ . Berikut adalah grafik fungsinya.

```
>plot2d("2*x*sin(x)/(1-cos(x))",-pi,pi); plot2d(0,4,>points,style="ow",>add);
```



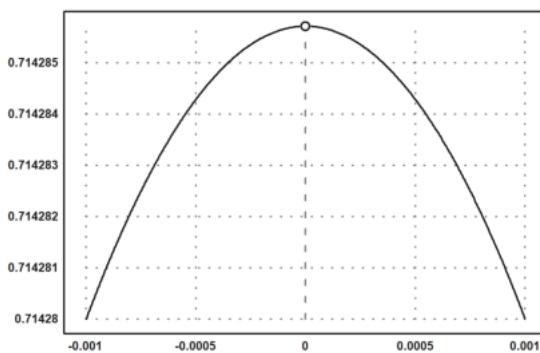
images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-014.png  
 $\Rightarrow \lim(\cot(7*h)/\cot(5*h), h, 0)$

$$\frac{5}{7}$$

maxima: showev('limit(cot(7\*h)/cot(5\*h),h,0))

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di  $x=0$ . Berikut adalah grafiknya.

$\Rightarrow \text{plot2d}(\text{"cot}(7*x)/\text{cot}(5*x)", -0.001, 0.001); \text{plot2d}(0, 5/7, >\text{points}, \text{style}=\text{"ow"}, >\text{add})$ :



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-016.png  
 $\Rightarrow \$\text{showev}(\text{'limit}((x/8)^{(1/3)}-1)/(x-8), x, 8)$

$$\lim_{x \rightarrow 8} \frac{\frac{x^{\frac{1}{3}}}{8} - 1}{x - 8} = \frac{1}{24}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

$\Rightarrow \$\text{showev}(\text{'limit}(1/(2*x-1), x, 0))$

$$\lim_{x \rightarrow 0} \frac{1}{2x - 1} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

$\Rightarrow \$\text{showev}(\text{'limit}((x^2-3*x-10)/(x-5), x, 5))$

$$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

$\Rightarrow \$\text{showev}(\text{'limit}(\sqrt{x^2+x} - x, x, inf))$

$$\lim_{x \rightarrow \infty} \sqrt{x^2 + x} - x = \frac{1}{2}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

$\Rightarrow \$\text{showev}(\text{'limit}(\text{abs}(x-1)/(x-1), x, 1, minus))$

$$\lim_{x \uparrow 1} \frac{|x - 1|}{x - 1} = -1$$

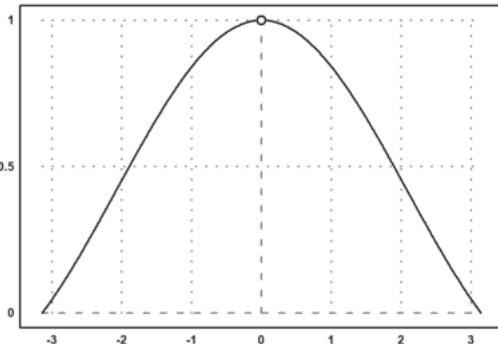
Hitung limit di atas untuk  $x$  menuju 1 dari kanan.

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

$\Rightarrow \$\text{showev}(\text{'limit}(\sin(x)/x, x, 0))$

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

```
>plot2d("sin(x)/x",-pi,pi); plot2d(0,1,>points,style="ow",>add):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-023.png

```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(log(x), x, minf))
```

$$\lim_{x \rightarrow -\infty} \log x = \text{infinity}$$

```
>$showev('limit((-2)^x,x, inf))
```

$$\lim_{x \rightarrow \infty} (-2)^x = \text{infinity}$$

```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t \downarrow 2} t - \sqrt{2-t} = 2$$

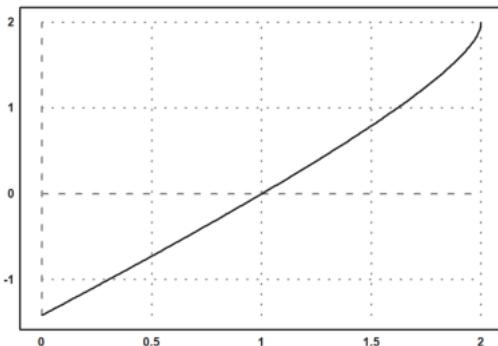
```
>$showev('limit(t-sqrt(2-t),t,2,plus))
```

$$\lim_{t \downarrow 2} t - \sqrt{2-t} = 2$$

```
>$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/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-030.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.

```
>$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.

```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

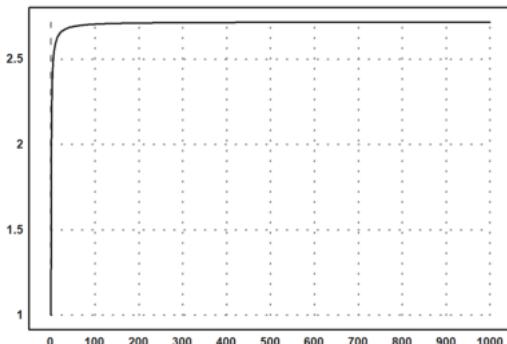
$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>showev('limit((1+1/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{1}{x} + 1 \right)^x = e$$

```
>plot2d("((1+1/x)^x",0,1000);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-035.png

```
>showev('limit((1+k/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{k}{x} + 1 \right)^x = e^k$$

```
>showev('limit((1+x)^(1/x),x,0))
```

$$\lim_{x \rightarrow 0} (x + 1)^{\frac{1}{x}} = e$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>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.

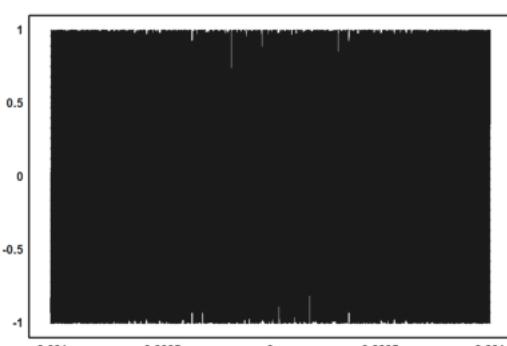
```
>showev('limit(sin(1/x),x,0))
```

$$\lim_{x \rightarrow 0} \sin \left( \frac{1}{x} \right) = \text{ind}$$

```
>showev('limit(sin(1/x),x,inf))
```

$$\lim_{x \rightarrow \infty} \sin \left( \frac{1}{x} \right) = 0$$

```
>plot2d("sin(1/x)",-0.001,0.001);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-042.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.

>\$showev('limit((4\*x-2),x,a)) // untuk setiap a adalah bilangan rill

$$\lim_{x \rightarrow a} 4x - 2 = 4a - 2$$

>\$showev('limit((4\*x-2),x,0)) // untuk a = 0

$$\lim_{x \rightarrow 0} 4x - 2 = -2$$

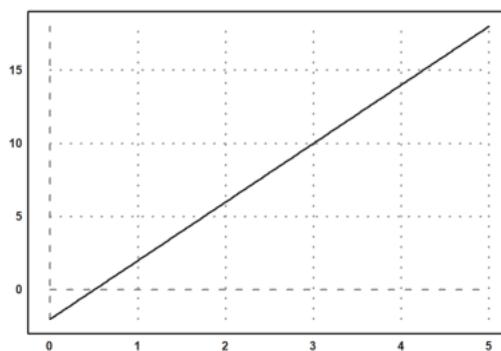
>\$showev('limit((4\*x-2),x,3)) // untuk a = 3

$$\lim_{x \rightarrow 3} 4x - 2 = 10$$

>\$showev('limit((4\*x-2),x,inf)) // untuk a = tak hingga

$$\lim_{x \rightarrow \infty} 4x - 2 = \infty$$

>plot2d("4\*x-2",0,5);



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-047.png

>\$showev('limit(((x^4-1)/(x^2-1)),x,a)) // untuk semua a bilangan real

$$\lim_{x \rightarrow a} \frac{x^4 - 1}{x^2 - 1} = a^2 + 1$$

>\$showev('limit(((x^4-1)/(x^2-1)),x,0))

$$\lim_{x \rightarrow 0} \frac{x^4 - 1}{x^2 - 1} = 1$$

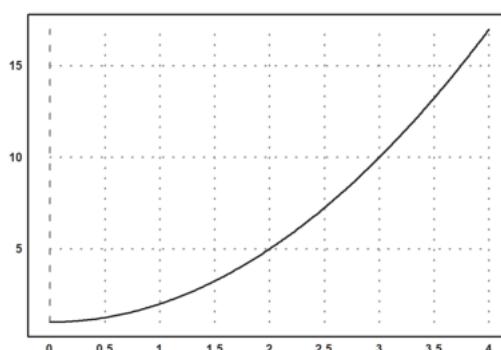
>\$showev('limit(((x^4-1)/(x^2-1)),x,5))

$$\lim_{x \rightarrow 5} \frac{x^4 - 1}{x^2 - 1} = 26$$

>\$showev('limit(((x^4-1)/(x^2-1)),x,inf))

$$\lim_{x \rightarrow \infty} \frac{x^4 - 1}{x^2 - 1} = \infty$$

>plot2d("x^4-1/(x^2-1)",0,4);



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-052.png

```

>$showev('limit((tan(x)),x,a))

$$\lim_{x \rightarrow a} \tan x = \tan a$$

>$showev('limit((tan(x)),x,0))

$$\lim_{x \rightarrow 0} \tan x = 0$$

>$showev('limit((tan(x)),x,pi))

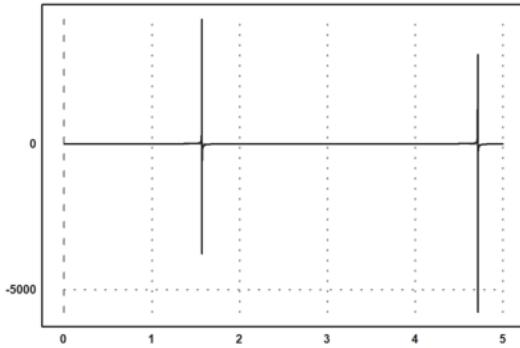
$$\lim_{x \rightarrow \pi} \tan x = 0$$

>$showev('limit((tan(x)),x,inf))

$$\lim_{x \rightarrow \infty} \tan x = \text{und}$$

>plot2d("(\tan(x))",0,5):

```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-057.png

```

>$showev('limit(((abs(x))),x,a))

$$\lim_{x \rightarrow a} |x| = |a|$$

>$showev('limit(((abs(x))),x,0))

$$\lim_{x \rightarrow 0} |x| = 0$$

>$showev('limit(((abs(x))),x,2))

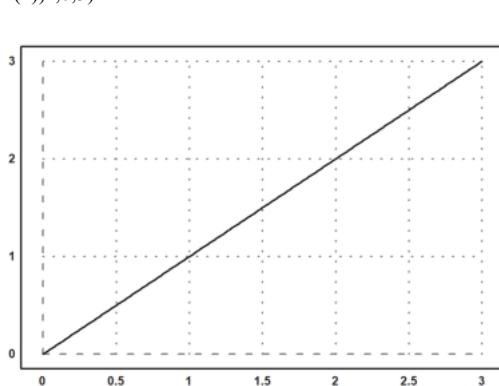
$$\lim_{x \rightarrow 2} |x| = 2$$

>$showev('limit(((abs(x))),x,inf))

$$\lim_{x \rightarrow \infty} |x| = \infty$$

>plot2d("(|x|)",0,3):

```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-062.png

```

>$showev('limit(((sqrt(x))),x,a))

$$\lim_{x \rightarrow a} \sqrt{x} = \sqrt{a}$$

>$showev('limit(((sqrt(x))),x,4))

$$\lim_{x \rightarrow 4} \sqrt{x} = 2$$

>$showev('limit(((sqrt(x))),x,9))

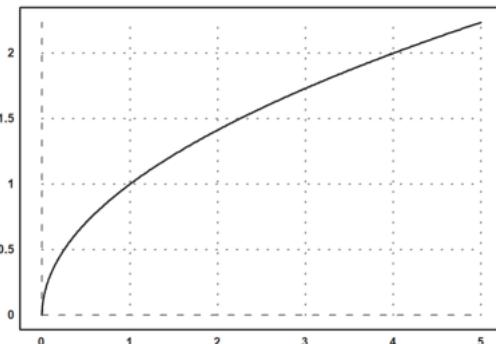
$$\lim_{x \rightarrow 9} \sqrt{x} = 3$$

>$showev('limit(((sqrt(x))),x,inf))

```

$$\lim_{x \rightarrow \infty} \sqrt{x} = \infty$$

>plot2d("sqrt(x)",0,5);



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-067.png

## Turunan Fungsi

Definisi turunan:

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

>\$showev('limit((x+h)^2-x^2)/h,h,0)) // turunan x^2

$$\lim_{h \rightarrow 0} \frac{(x + h)^2 - x^2}{h} = 2x$$

>p &= expand((x+h)^2-x^2)|simplify; \$p // pembilang dijabarkan dan disederhanakan

$$2hx + h^2$$

>q &= ratsimp(p/h); \$q // ekspresi yang akan dihitung limitnya disederhanakan

$$2x + h$$

>\$limit(q,h,0) // nilai limit sebagai turunan

$$2x$$

>\$showev('limit((x+h)^n-x^n)/h,h,0)) // turunan x^n

$$\lim_{h \rightarrow 0} \frac{(x + h)^n - x^n}{h} = nx^{n-1}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan  $(x+h)^n$  dengan menggunakan teorema binomial.

>\$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)

$$\lim_{h \rightarrow 0} \frac{\sin(x + h) - \sin x}{h} = \cos x$$

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.

>\$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)

$$\lim_{h \rightarrow 0} \frac{\log(x + h) - \log x}{h} = \frac{1}{x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, gunakan sifat-sifat logaritma dan hasil limit pada bagian sebelumnya di atas.

>\$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x

$$\lim_{h \rightarrow 0} \frac{\frac{1}{x+h} - \frac{1}{x}}{h} = -\frac{1}{x^2}$$

>\$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x

```

Answering "Is x an integer?" with "integer"
Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is x an integer?

```

```

Use assume!
Error in:
$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x ...
^

```

Maxima bermasalah dengan limit:

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{e^h - 1}{h} = 1$$

```
>$showev('factor(E^(x+h)-E^x))
```

$$\text{factor } (e^{x+h} - e^x) = (e^h - 1) e^x$$

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

$$\left( \lim_{h \rightarrow 0} \frac{e^h - 1}{h} \right) e^x = e^x$$

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

$$\begin{matrix} x \\ x \end{matrix}$$

```
>$showev('limit(f(x),x,0))
```

$$\lim_{x \rightarrow 0} x^x = 1$$

Silakan Anda gambar kurva

```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x + h)^{x+h} - x^x}{h} = \text{infinity}$$

Di sini Maxima juga bermasalah terkait limit:

Dalam hal ini diperlukan asumsi nilai x.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x + h)^{x+h} - x^x}{h} = x^x (\log x + 1)$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

```
[x > 0]
```

```
>&forget(x<0)
```

```
[x < 0]
```

```
>&facts()
```

```
[]
```

```
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
```

$$\lim_{h \rightarrow 0} \frac{\arcsin(x + h) - \arcsin x}{h} = \frac{1}{\sqrt{1 - x^2}}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

$$\lim_{h \rightarrow 0} \frac{\tan(x + h) - \tan x}{h} = \frac{1}{\cos^2 x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar,

sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

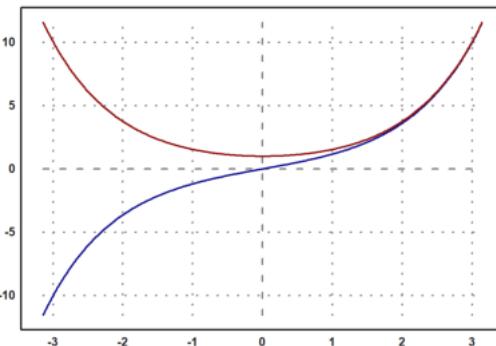
```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
          sinh(x)

>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
          
$$\frac{e^{-x} (e^{2x} + 1)}{2}$$

```

Hasilnya adalah cosh(x), karena

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-085.png

```
>function f(x) &= sin(3*x^5+7)
          
$$\sin(3x^5 + 7)$$

```

```
>diff(f,3), diffc(f,3)
```

```
1198.32948904  
1198.72863721
```

Apakah perbedaan diff dan diffc?

```
>$showev('diff(f(x),x))
```

$$\frac{d}{dx} \sin^2(3x^5 + 7) = 30x^4 \cos(3x^5 + 7) \sin(3x^5 + 7)$$

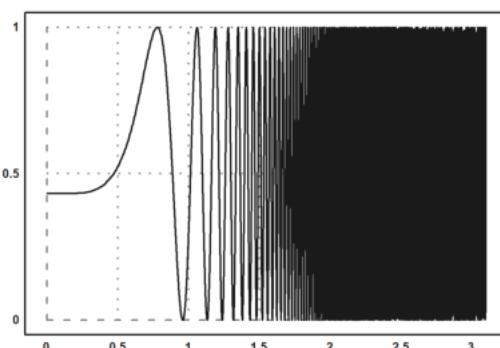
```
>% with x=3
```

$$\text{at } \left( \frac{d}{dx} \sin^2(3x^5 + 7), x = 3 \right) = 2430 \cos 736 \sin 736$$

```
>$float(%)
```

$$\text{at } \left( \frac{d^{1.0}}{dx^{1.0}} \sin^2(3.0x^5 + 7.0), x = 3.0 \right) = 1198.728637211748$$

```
>plot2d(f,0,3.1):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-089.png

```
>function f(x) &= 5*cos(2*x)-2*x*sin(2*x) // mendefinisikan fungsi f
          
$$5 \cos(2x) - 2x \sin(2x)$$

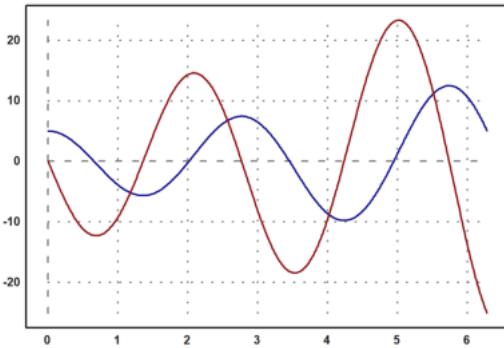

>function df(x) &= diff(f(x),x) // fd(x) = f'(x)
          
$$- 12 \sin(2x) - 4x \cos(2x)$$

```

```

>$'f(1)=f(1), $float(f(1)), '$'f(2)=f(2), $float(f(2)) // nilai f(1) dan f(2)
- 0.2410081230863468
-3.899329036387075
f (2) = 5 cos 4 - 4 sin 4
images/Siti%20Faltipah%20Hayani_23030630004_EMT4kalkulus-092.png
-0.2410081230863468
images/Siti%20Faltipah%20Hayani_23030630004_EMT4kalkulus-093.png
>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

```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-094.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.

### SOAL 1

$$f(x) = x^2 - 8$$

Turunkan dengan menggunakan definisi limit

```
>$showev('limit(((x+h)^2-8)-(x^2-8))/h,h,0)) // turunan fungsi x^2-8
```

$$\lim_{h \rightarrow 0} \frac{(x + h)^2 - x^2}{h} = 2x$$

Turunkan dengan menggunakan perintah diff

```
>function f(x) &=x^2-8 // mendefinisikan fungsi f
```

$$\begin{matrix} 2 \\ x \\ - 8 \end{matrix}$$

```
>function df(x) &=diff(f(x),x) // fd(x) = f'(x)
```

$$2 \ x$$

### SOAL 2

$$g(x) = x^5$$

Turunkan dengan menggunakan definisi limit

```
>$showev('limit(((x+h)^5-(x^5))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{(x + h)^5 - x^5}{h} = 5x^4$$

Turunkan dengan menggunakan fungsi diff

```

>function g(x)&=x^5
      5
      x
>function df(x) &=diff(g(x),x) // fd(x) = f'(x)
      4
      5 x

```

### SOAL 3

$$\log(x)$$

Turunkan dengan menggunakan definisi limit

```
>$showev('limit(((log(x+h))-log(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{\log(x + h) - \log x}{h} = \frac{1}{x}$$

Turunkan dengan menggunakan perintah diff

```
>function f(x)&=log(x)
```

$$\log(x)$$

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

$$\begin{matrix} 1 \\ - \\ x \end{matrix}$$

### SOAL 4

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

Turunkan dengan menggunakan definisi limit

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{\sin(x + h) - \sin x}{h} = \cos x$$

Turunkan dengan menggunakan perintah diff

```
>function f(x) &=sin(x)
```

$$\sin(x)$$

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

$$\cos(x)$$

### SOAL 5

$$f(x) = \frac{1}{x^3}$$

```
>$showev('limit((1/(x+h)^3-1/x^3)/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{\frac{1}{(x+h)^3} - \frac{1}{x^3}}{h} = -\frac{3}{x^4}$$

```
>function f(x) &=(1/(x^3))
```

$$\begin{matrix} 1 \\ -- \\ 3 \\ x \end{matrix}$$

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

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

## 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 mplementasikan 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))
```

$$\int \frac{1}{x+1} dx = \log(x+1)$$

```
>$showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2+1} dx = \arctan x$$

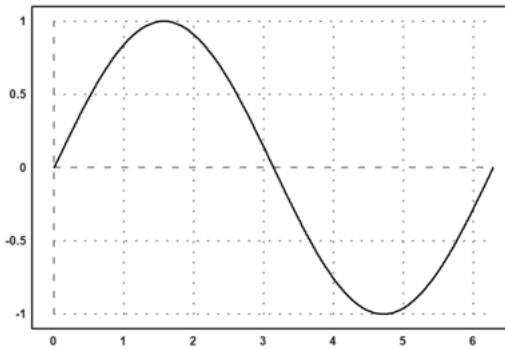
```
>$showev('integrate(1/sqrt(1-x^2),x))
```

$$\int \frac{1}{\sqrt{1-x^2}} dx = \arcsin x$$

```
>$showev('integrate(sin(x),x,0,pi))
```

$$\int_0^\pi \sin x dx = 2$$

```
>plot2d("sin(x)",0,2*pi);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-110.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,pi/2))
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^\pi - 1) \sin a + (e^\pi + 1) \cos a$$

>factor(%)

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x} + a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^\pi - 1) (\sin a - \cos a)$$

>function map f(x) &= E^{(-x^2)}

$$E^{-x^2}$$

>showev('integrate(f(x),x))

$$\int e^{-x^2} dx = \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

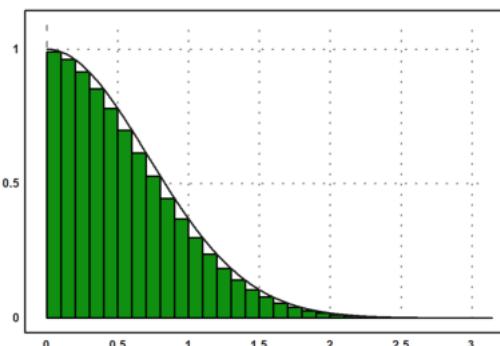
Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

maxima: 'integrate(f(x),x,0,pi)

>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add);



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-119.png

Integral tentu

maxima: 'integrate(f(x),x,0,pi)

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva  $y=f(x)$  tersebut. Langkah-langkahnya adalah sebagai berikut.

>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-nilai x

>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)

>/ jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!

Hasilnya adalah:

maxima: 'integrate(f(x),x,0,pi) = 0.1\*sum(fx[i],i,1,length(fx))

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai  $f(x)$  untuk  $x = 0.1, 0.2, 0.3, \dots, 3.2$ .

>0.1\*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT

0.836219610253

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

>showev('integrate(f(x),x,0,inf))

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

Tunjukkan kebenaran hasil di atas!

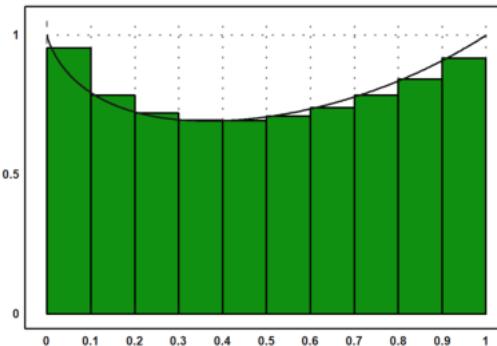
Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

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

```
x
x
>$showev('integrate(f(x),x,0,1))

\int_0^1 x^x dx = \int_0^1 x^x dx

>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-122.png

Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);
```

```
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
```

```
maxima: 'integrate(f(x),x,0,1) = 0.01*sum(fx[i],i,1,length(fx))
```

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
>function f(x) &= sin(3*x^5+7)2
```

$$\sin(3x^5 + 7)^2$$

```
>integrate(f,0,1)
```

```
0.542581176074
```

```
>&showev('integrate(f(x),x,0,1))
```

```
1
/
[      2      5          1          pi
I  sin (3 x  + 7) dx = -----
]                               1/5
/
0
4/5          1          4/5          1
- ((6 gamma_incomplete(-, 6 I) + 6 gamma_incomplete(-, - 6 I))
5           5
sin(14) + (6 I gamma_incomplete(-, 6 I)
5
4/5          1          pi
- 6 I gamma_incomplete(-, - 6 I)) cos(14) sin(--)
5           10
- 60)/120
```

```
>&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)

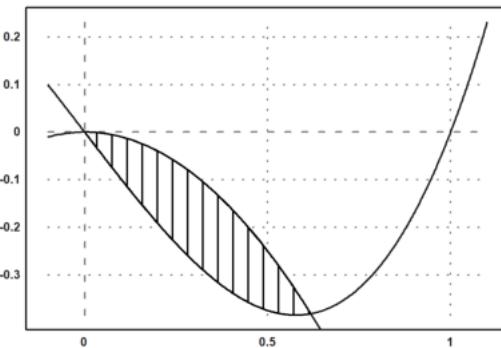
>$showev(`integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)


$$\int_0^1 x e^{-x} dx = 1 - 2 e^{-1}$$


# 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

```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-124.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)-(x^3-x)",a,b) // luas daerah yang diarsir

0.0758191713542

```

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```
>a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik potong kedua kurva secara eksak
```

$$[ x = \frac{-\sqrt{5}-1}{2}, x = \frac{\sqrt{5}-1}{2}, x = 0 ]$$

```
>$showev(`integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara eksak
```

$$\int_0^{\frac{\sqrt{5}-1}{2}} -x^3 - x^2 + x \, dx = \frac{13 - 5^{\frac{1}{2}}}{24}$$

```
>$float(%)
```

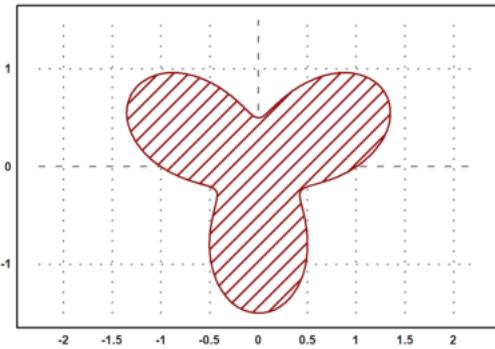
$$\int_{0.0}^{0.6180339887498949} -1.0 x^3 - 1.0 x^2 + x \, dx = 0.07581917135421037$$

## 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=1.5); // Kita gambar kurvanya terlebih dahulu
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-128.png

```
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
```

$$r(t) = \frac{\sin(3t)}{2} + 1$$

```
>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
```

$$fx(t) = \cos t \left( \frac{\sin(3t)}{2} + 1 \right)$$

```
>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
```

$$fy(t) = \sin t \left( \frac{\sin(3t)}{2} + 1 \right)$$

```
>function ds(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t)
```

$$ds(t) = \frac{\sqrt{4 \cos(6t) + 4 \sin(3t) + 9}}{2}$$

```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
```

$$\int_0^{2\pi} \frac{\sqrt{4 \cos(6x) + 4 \sin(3x) + 9}}{2} dx$$

Maxima gagal melakukan perhitungan eksak integral tersebut.

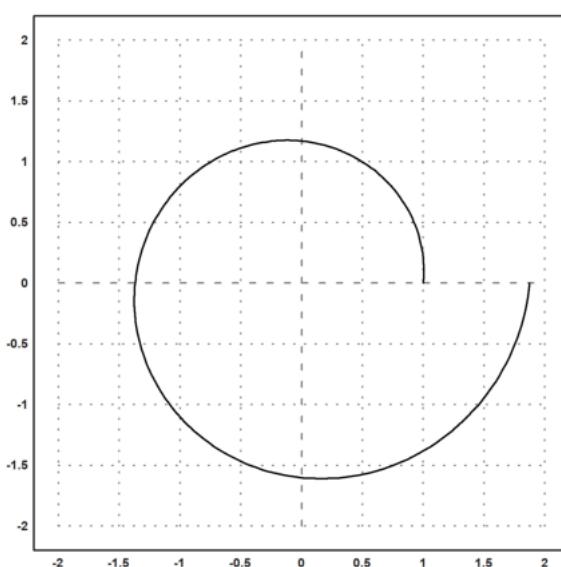
Berikut kita hitung integralnya secara umerik dengan perintah EMT.

```
>integrate("ds(x)",0,2*pi)
```

9.0749467823

Spiral Logaritmik

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-134.png

```
>&kill(a) // hapus expresi a
```

done

```

>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
    fx(t) = e^a t cos t
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
    fy(t) = e^a t sin t
>function df(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t)
    df(t) = sqrt(a^2 + 1) e^a t
>S &=integrate(df(t),t,0,2*pi); $S // panjang kurva (spiral)
    sqrt(a^2 + 1) (e^(2 pi a) - 1/a)
>S(a=0.1) // Panjang kurva untuk a=0.1
8.78817491636

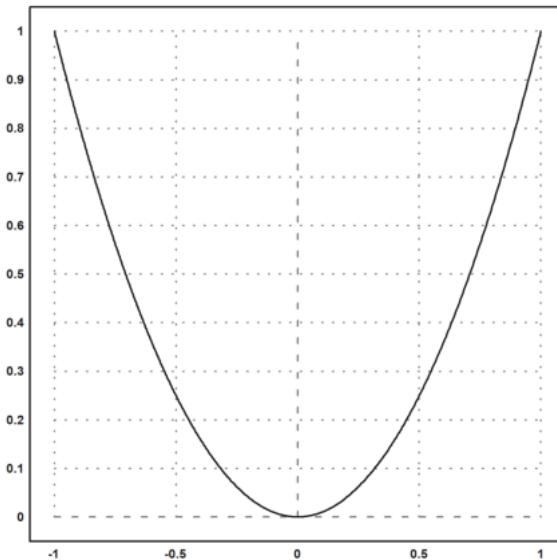
```

Soal:

Tunjukkan bahwa keliling lingkaran dengan jari-jari r adalah  $K=2\pi r$ .

Berikut adalah contoh menghitung panjang parabola.

```
>plot2d("x^2",xmin=-1,xmax=1);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-139.png

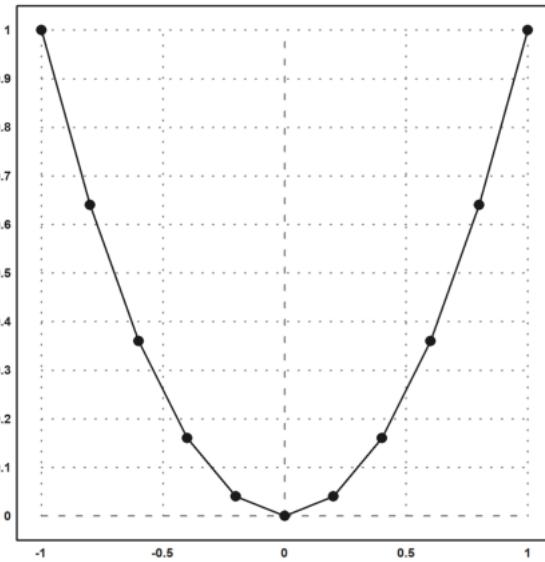
```
>$showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))
```

$$\int_{-1}^1 \sqrt{4x^2 + 1} dx = \frac{\operatorname{asinh} 2 + 2\sqrt{5}}{2}$$

```
>$float(%)
```

$$\int_{-1.0}^{1.0} \sqrt{4.0x^2 + 1.0} dx = 2.957885715089195$$

```
>x=-1:0.2:1; y=x^2; plot2d(x,y); ...
>plot2d(x,y,points=1,style="o#",add=1);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-142.png

Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```
>i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])2+(y[i+1]-y[i])2))
```

2.95191957027

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

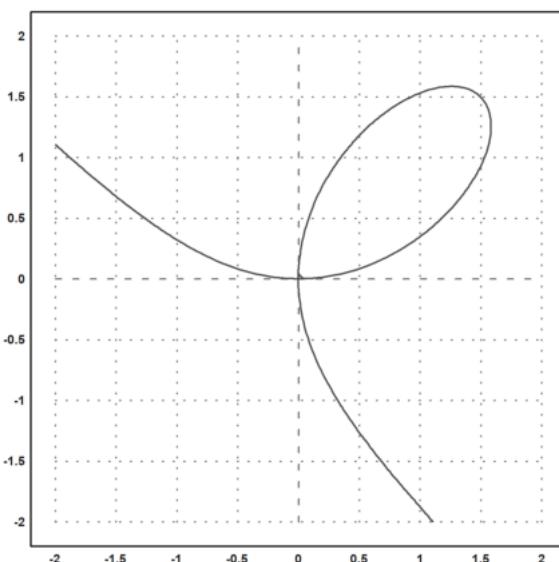
## Koordinat Kartesius

Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

```
>z &= x3+y3-3*x*y; $z
```

$$y^3 - 3xy + x^3$$

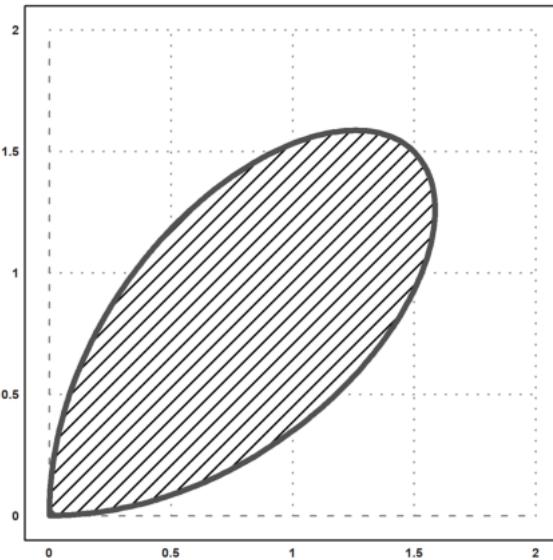
```
>plot2d(z,r=2,level=0,n=100);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-144.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="/");
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-145.png

Kita selesaikan persamaannya untuk x.

>\$z with  $y=l*x$ , sol &= solve(%x); \$sol

$$[x = \frac{3l}{l^3 + 1}, x = 0]$$

$$\left[ x = \frac{3l}{l^3 + 1}, x = 0 \right]$$

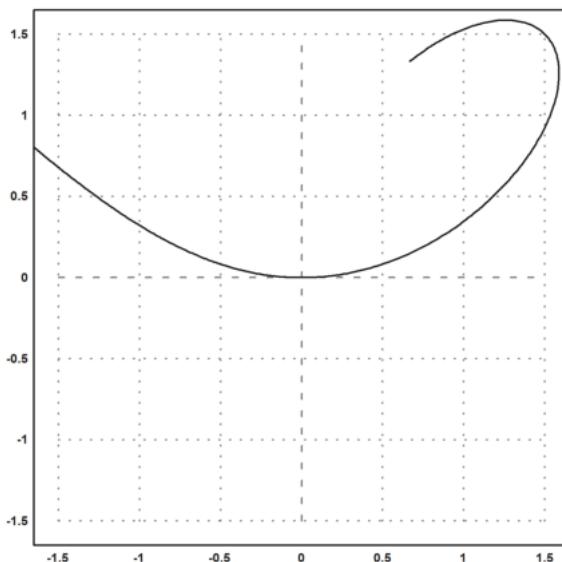
Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

>function f(l) &= rhs(sol[1]); \$'f(l)=f(l)

$$f(l) = \frac{3l}{l^3 + 1}$$

Fungsi tersebut juga dapat digunakan untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai  $y=l*x$ , yakni  $x=f(l)$  dan  $y=l*f(l)$ .

>plot2d(&f(x),&x\*f(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-149.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)

$$ds(l) = \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}}$$

>\$integrate(ds(l),l,0,1)

$$\int_0^1 \frac{\sqrt{91^8 + 361^6 - 361^5 - 361^3 + 361^2 + 9}}{\sqrt{1^{12} + 41^9 + 61^6 + 41^3 + 1}} dl$$

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral tersebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
>2*integrate("ds(x)",0,1)
```

```
4.91748872168
```

```
>2*romberg(&ds(x),0,1)// perintah Euler lain untuk menghitung nilai hampiran integral yang sama
```

```
4.91748872168
```

Perhitungan di 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
```

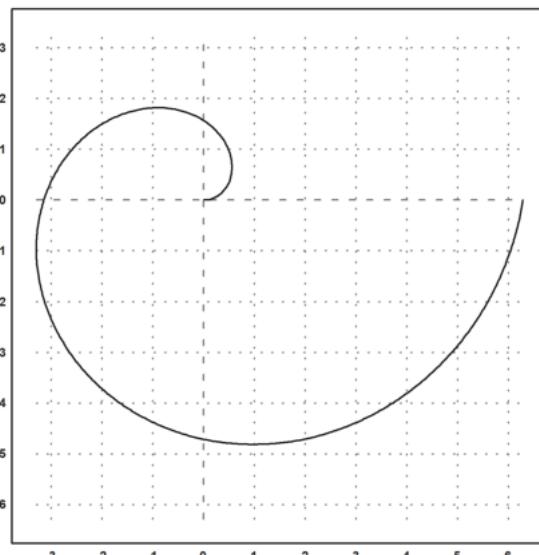
Bandangkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)",mxm("x*f(x)",0,1))// cek contoh terakhir, bandangkan hasilnya!
```

```
4.91748872168
```

Kita hitung panjang spiral Archimedes berikut ini dengan fungsi tersebut.

```
>plot2d("x*cos(x)","x*sin(x)",xmin=0,xmax=2*pi,square=1):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-152.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(diff (fy, x) ^ 2 + 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(% ,x,0,2*pi), %()

$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$


$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$

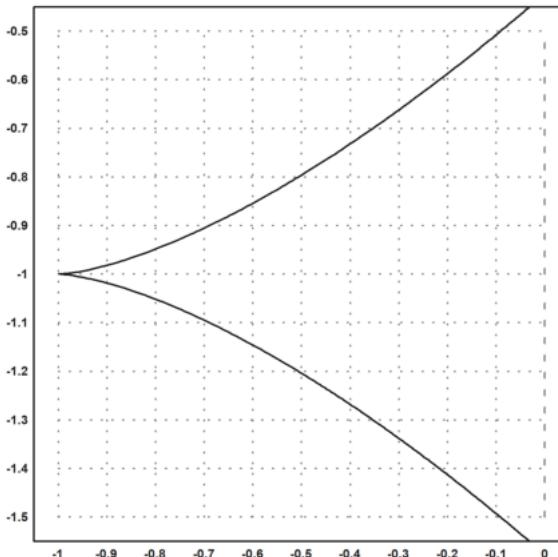
21.2562941482

```

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.

Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
>plot2d("3*x^2-1","3*x^3-1",xmin=-1/sqrt(3),xmax=1/sqrt(3),square=1):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-156.png

```

>sol &= radcan(ds(3*x^2-1,3*x^3-1)); $sol

$$3x\sqrt{9x^2 + 4}$$

>$showev(`integrate(sol,x,0,1/sqrt(3))), $2*float(%)) // panjang kurva di atas

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0x^2 + 4.0} dx = 2.337835372767141$$


$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0x^2 + 4.0} dx = 2.337835372767141$$


```

## Sikloid

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar ke kanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah  $r$ . Posisi titik pusat lingkaran pada saat  $t$  adalah:

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  $t=0$ ,  $t=\pi/2$ ,  $t=r*\pi$ .

```

>x &= r*(t-sin(t))
r (t - sin(t))
>y &= r*(1-cos(t))
r (1 - cos(t))

```

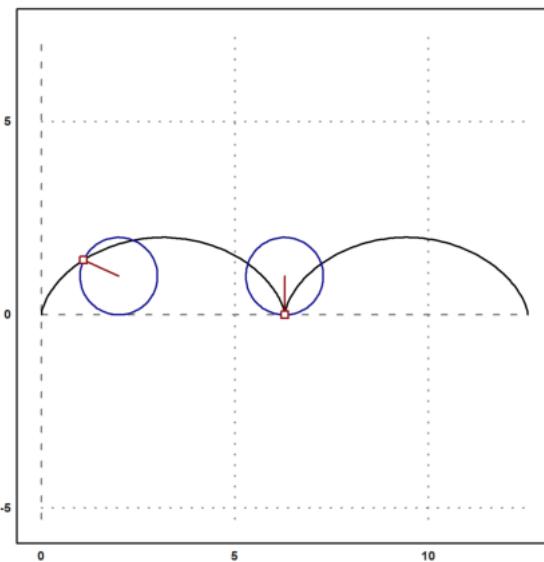
Berikut kita gambar sikloid untuk  $r=1$ .

```

>ex &= x-sin(x); ey &= 1-cos(x); aspect(1);
>plot2d(ex,ey,xmin=0,xmax=4pi,square=1); ...
>plot2d("2+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
>plot2d([2,ex(2)], [1,ey(2)],color=red,>add); ...
>plot2d(ex(2),ey(2),>points,>add,color=red); ...

```

```
> plot2d("2pi+cos(x)", "1+sin(x)", xmin=0, xmax=2pi, >add, color=blue); ...
> plot2d([2pi, ex(2pi)], [1, ey(2pi)], color=red, >add); ...
> plot2d(ex(2pi), ey(2pi), >points, >add, color=red);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-160.png

Berikut dihitung panjang lintasan untuk 1 putaran penuh. (Jangan salah menduga bahwa panjang lintasan 1 putaran penuh sama dengan keliling lingkaran!)

```
>ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds) // elemen panjang kurva sikloid
```

$$\sqrt{\sin^2 x + \cos^2 x - 2 \cos x + 1} = \sqrt{2 - 2 \cos x}$$

```
>ds &= trigsimp(ds); $ds
```

$$\sqrt{2 - 2 \cos x}$$

```
>$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu putaran penuh
```

$$\int_0^{2\pi} \sqrt{2 - 2 \cos x} dx = 8$$

```
>integrate(mxm("ds"),0,2*pi) // hitung secara numerik
```

8

```
>romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik
```

8

Perhatikan, seperti terlihat pada gambar, panjang sikloid lebih besar daripada keliling lingkarannya, yakni:

## Kurvatur (Kelengkungan) Kurva

image: Osculating.png

Aslinya, kelengkungan kurva diferensiabel (yakni, kurva mulus yang tidak lancip) di titik P didefinisikan melalui lingkaran oskulasi (yaitu, lingkaran yang melalui titik P dan terbaik memperkirakan, paling banyak menyentuh 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  $2\pi$  sejauh  $2\pi R$ .)

Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekivalen.

## Definisi Kurvatur dengan Fungsi Parametrik Panjang Kurva

Setiap kurva diferensiabel dapat dinyatakan dengan persamaan parametrik terhadap panjang kurva s:

dengan x dan y adalah fungsi riil yang diferensiabel, yang memenuhi:

Ini berarti bahwa vektor singgung

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):

Nilai disebut jari-jari (radius) kelengkungan kurva.

Bilangan riil

disebut nilai kelengkungan bertanda.

Contoh:

Akan ditentukan kurvatur lingkaran

```
>fx &= r*cos(t); fy &= r*sin(t);
>&assume(t>0,r>0); s &=integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s // elemen panjang
kurva, panjang busur lingkaran (s)
```

$r \ t$

```
>&kill(s); fx &= r*cos(s/r); fy &= r*sin(s/r); // definisi ulang persamaan parametrik
terhadap s dengan substitusi t=s/r
```

```
>k &= trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur lingkaran dengan
menggunakan definisi di atas
```

$$\frac{1}{r}$$

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

$$\begin{aligned} \kappa &= \frac{\frac{d\phi}{dt}}{\frac{ds}{dt}} = \frac{\frac{d\phi}{dt}}{\frac{dy}{dx}} \quad (\phi \text{ adalah sudut kemiringan garis singgung dan } s \text{ adalah panjang kurva}) \\ &= \frac{\frac{d\phi}{dt}}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}} = \frac{\frac{d\phi}{dt}}{\sqrt{x'(t)^2 + y'(t)^2}}. \end{aligned}$$

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

$$\sec^2 \phi \frac{d\phi}{dt} = \frac{d}{dt}(\tan \phi) = \frac{d}{dt}\left(\frac{dy}{dx}\right) = \frac{d}{dt}\left(\frac{dy/dt}{dx/dt}\right) = \frac{d}{dt}\left(\frac{y'(t)}{x'(t)}\right) = \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2}.$$

$$\begin{aligned} \frac{d\phi}{dt} &= \frac{1}{\sec^2 \phi} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1 + \tan^2 \phi} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1 + \left(\frac{y'(t)}{x'(t)}\right)^2} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2 + y'(t)^2}. \end{aligned}$$

Jadi, rumus kurvatur untuk kurva parametrik

$$x = x(t), y = y(t)$$

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 kurvurnya adalah

$$\kappa(\theta) = \frac{r(\theta)^2 + 2r'(\theta)^2 - r(\theta)r''(\theta)}{(r'(\theta)^2 + r''(\theta)^2)^{3/2}}.$$

(Silakan Anda turunkan rumus tersebut!)

Contoh:

Lingkaran dengan pusat (0,0) dan jari-jari r dapat dinyatakan dengan persamaan parametrik

$$x = r \cos t, y = r \sin t.$$

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 kurvurnya 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)

$$y = a x^2 + b x + c$$

>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)(3/2); \$'k(x)=k(x) // kelengkungan parabola

$$k(x) = \frac{2a}{((2ax + b)^2 + 1)^{\frac{3}{2}}}$$

>function f(x) &= x^2+x+1; \$y=f(x) // akan kita plot kelengkungan parabola untuk a=b=c=1

$$y = x^2 + x + 1$$

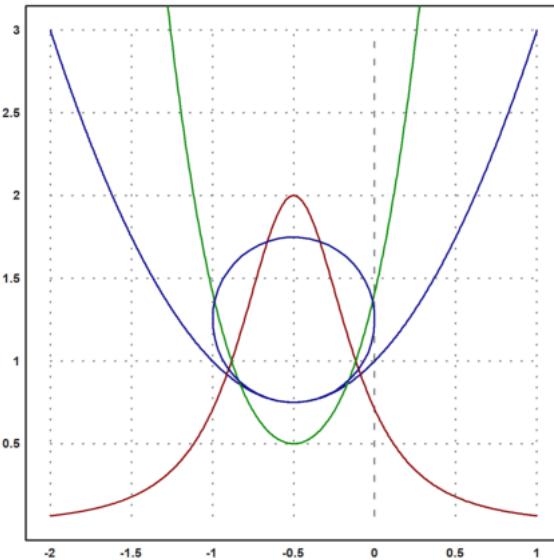
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)(3/2); \$'k(x)=k(x) // kelengkungan parabola

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

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);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-182.png

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 kurvaturnya adalah

$$\kappa = \frac{F_y^2 F_{xx} - 2F_x F_y F_{xy} + F_x^2 F_{yy}}{(F_x^2 + F_y^2)^{3/2}}.$$

(Silakan Anda turunkan sendiri!)

Contoh 1:

Parabola

$$y = ax^2 + bx + c$$

dapat dinyatakan ke dalam persamaan implisit

$$ax^2 + bx + c - y = 0.$$

>function F(x,y) &=a\*x^2+b\*x+c-y; \$F(x,y)

$$- y + a x^2 + b x + c$$

>Fx &= diff(F(x,y),x), Fxx &= diff(F(x,y),x,2), Fy &= diff(F(x,y),y), Fxy &= diff(diff(F(x,y),x),y), Fyy &= diff(F(x,y),y,2)

$$2 a x + b$$

$$2 a$$

$$- 1$$

$$\theta$$

$$\theta$$

>function k(x) &= (Fy^2\*Fxx-2\*Fx\*Fy\*Fxy+Fx2\*Fyy)/(Fx^2+Fy^2)^(3/2); \$'k(x)=k(x) // kurvatur parabola tersebut

$$k(x) = \frac{2 a}{((2 a x + b)^2 + 1)^{\frac{3}{2}}}$$

Hasilnya sama dengan sebelumnya yang menggunakan persamaan parabola biasa.

## Latihan

- Bukalah buku Kalkulus.
- Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi).
- Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah integral tentu dengan batas-batas yang menarik (Anda tentukan sendiri), seperti contoh-contoh tersebut.
- Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat diintegralkan (cari sedikitnya 3 fungsi).
- Gambar grafik fungsi dan daerah integrasinya pada sumbu koordinat yang sama.
- Gunakan integral tentu untuk mencari luas daerah yang dibatasi oleh dua kurva yang berpotongan di dua titik. (Cari dan gambar kedua kurva dan arsir (warnai) daerah yang dibatasi oleh keduanya.)
- Gunakan integral tentu untuk menghitung volume benda putar kurva  $y = f(x)$  yang diputar mengelilingi sumbu x dari  $x=a$  sampai  $x=b$ , yakni

$$V = \int_a^b \pi(f(x))^2 dx.$$

(Pilih fungsinya dan gambar kurva dan benda putar yang dihasilkan. Anda dapat mencari contoh-contoh bagaimana cara menggambar benda hasil perputaran suatu kurva.)

- Gunakan integral tentu untuk menghitung panjang kurva  $y=f(x)$  dari  $x=a$  sampai  $x=b$  dengan menggunakan rumus:

$$S = \int_a^b \sqrt{1 + (f'(x))^2} dx.$$

(Pilih fungsi dan gambar kurvanya.)

- Apabila fungsi dinyatakan dalam koordinat kutub  $x=f(r,t)$ ,  $y=g(r,t)$ ,  $r=h(t)$ ,  $x=a$  bersesuaian dengan  $t=t_0$  dan  $x=b$  bersesuaian dengan  $t=t_1$ , maka rumus di atas akan menjadi:

$$S = \int_{t_0}^{t_1} \sqrt{x'(t)^2 + y'(t)^2} dt.$$

\* Pilih beberapa kurva menarik (selain lingkaran dan parabola) dari buku kalkulus. Nyatakan setiap kurva tersebut dalam bentuk: a. koordinat Kartesius (persamaan  $y=f(x)$ ) b. koordinat kutub ( $r=r(\theta)$ ) c. persamaan parametrik  $x=x(t)$ ,  $y=y(t)$  d. persamaan implisit  $F(x,y)=0$

- Tentukan kurvatur masing-masing kurva dengan menggunakan keempat representasi tersebut (hasilnya harus sama).
- Gambarlah kurva asli, kurva kurvatur, kurva jari-jari lingkaran oskulasi, dan salah satu lingkaran oskulasinya.

## Fungsi

1. Tentukan panjang kurva dan volume benda putar kurva  $y=f(x)$  yang diputar mengelilingi sumbu x dari  $x=0$  sampai  $x=4$

$$f(x) = x^3$$

>function f(x)&=x^3; \$f(x) // Mendefinisikan fungsi

$$x^3$$

>\$showev(`integrate(pi\*(f(x))^2,x,0,4))// Menentukan antiturunan

$$\pi \int_0^4 x^6 dx = \frac{16384\pi}{7}$$

>function df(x) &= limit((f(x+h)-f(x))/h,h,0); \$df(x) // df(x) = f'(x) // menentukan turuna fungsi

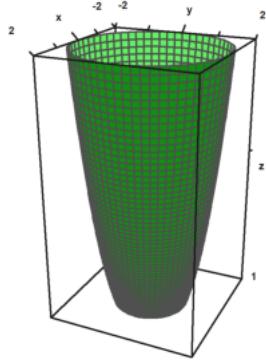
$$3x^2$$

p

>\$showev(`integrate(sqrt((1+df(x)^2)),x,0,4)) // panjang kurva

$$\int_0^4 \sqrt{9x^4 + 1} dx = \int_0^4 \sqrt{9x^4 + 1} dx$$

>plot3d("x^3",2,0,2,rotate=2); // grafik benda putar mengelilingi sumbu x



Tentukan panjang kurva dan volume benda putar kurva  $y=f(x)$  yang diputar mengelilingi sumbu x dari  $x=-1$  sampai  $x=2$

$$f(x) = 3x^2 + 2x + 1$$

>function f(x)&=3\*x^2+2\*x+1; \$f(x) // volume benda putar

$$3x^2 + 2x + 1$$

>\$showev('integrate(pi\*(f(x))^2,x,-1,2))

$$\pi \int_{-1}^2 (3x^2 + 2x + 1)^2 dx = \frac{717\pi}{5}$$

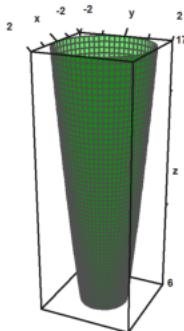
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); \$df(x) // df(x) = f'(x) // menentukan turunan fungsinya

$$6x + 2$$

>\$showev('integrate(sqrt((1+df(x)^2)),x,-1,2)) // menentukan panjang kurva

$$\int_{-1}^2 \sqrt{(6x + 2)^2 + 1} dx = \frac{\operatorname{asinh} 14 + 14\sqrt{197}}{12} + \frac{\operatorname{asinh} 4 + 4\sqrt{17}}{12}$$

>plot3d("3x^2+2x+1",2,-1,2,rotate=2); // menggambar plot benda putar mengelilingi sumbu x



[-1,1] dari fungsi berikut

Integral tentu dengan batas

$$f(x) = x^2 + 8x - 9$$

>function map f(x) &= (x^2+8\*x-9); \$f(x)

$$x^2 + 8x - 9$$

>\$showev('integrate(f(x), x, -1, 1)// mencari nilai dari integral tentu fungsi f(x) dengan batas [-1,1]

$$\int_{-1}^1 x^2 + 8x - 9 dx = -\frac{52}{3}$$

4. Tentukan panjang kurva dan volume benda putar kurva  $y=f(x)$  yang diputar mengelilingi sumbu x dari  $x=0$  sampai  $x=2$

```

f(x) = 4x^2 + 1
>$showev(`integrate(pi*(f(x))^2,x,0,2))


$$\pi \int_0^2 (x^2 + 8x - 9)^2 dx = \frac{1006\pi}{15}$$


>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)

$$2x + 8$$


>$showev(`integrate (k(x),x,0,2))// tidak dapat diintegralkan

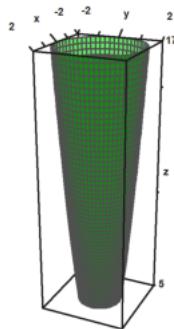
$$\int_0^2 \frac{1}{\log x} dx = \int_0^2 \frac{1}{\log x} dx$$


>$showev(`integrate(sqrt((1+df(x)^2)),x,0,2))// menentukan panjang kurva

$$\int_0^2 \sqrt{(2x + 8)^2 + 1} dx = \frac{\operatorname{asinh} 12 + 12\sqrt{145}}{4} - \frac{\operatorname{asinh} 8 + 8\sqrt{65}}{4}$$


>plot3d("4x^2+1",2,0,2,rotate=2); // menentukan panjang kurva

```



Tentukan integral fungsi berikut.

$$f(x) = \frac{\sqrt{2x}}{x} + \frac{3}{x^5}$$

```

>function f(x) &= (sqrt(2*x))/x + 3/x^5 ; $f(x)

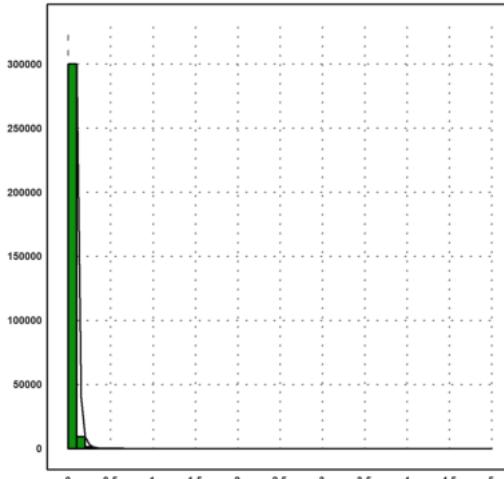
$$\frac{\sqrt{2}}{\sqrt{x}} + \frac{3}{x^5}$$

>$showev(`integrate(((sqrt(2*x))/x)+(3/x^5),x))

$$\int \frac{\sqrt{2}}{\sqrt{x}} + \frac{3}{x^5} dx = 2^{\frac{1}{2}} \sqrt{x} - \frac{3}{4x^4}$$


```

```
>x=0:0.1:5-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,5,>add);
```



>

## Barisan dan Deret

(Cataatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan (menggunakan titik dua “:”);
- menggunakan perintah “sequence” dan rumus barisan (suku ke -n);
- menggunakan perintah “iterate” atau “niterate”;
- menggunakan fungsi Maxima “create\_list” atau “makelist” untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) “sum”. Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

>1:10 // barisan sederhana

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

>1:2:30

[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]

## Iterasi dan Barisan

EMT menyediakan fungsi iterate(“g(x)”, x0, n) untuk melakukan iterasi

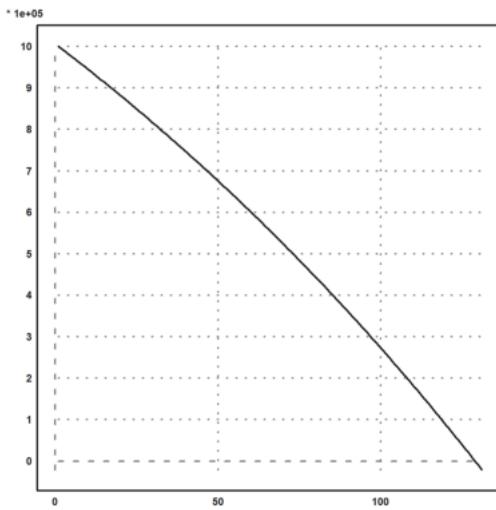
Berikut ini disajikan contoh-contoh penggunaan iterasi dan rekursi dengan EMT. Contoh pertama menunjukkan pertumbuhan dari nilai awal 1000 dengan laju pertambahan 5%, selama 10 periode.

>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/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-219.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]

[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]
```

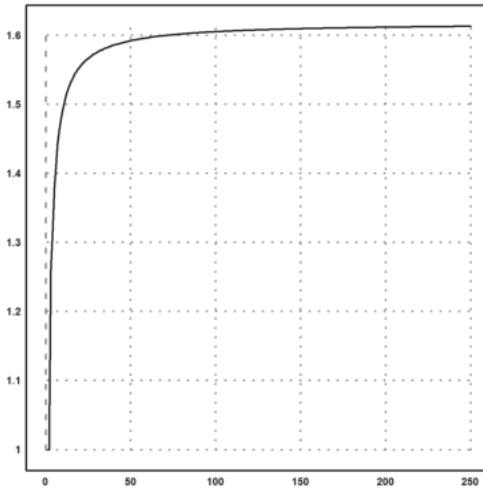
Barisan Fibonacci memiliki banyak sifat menarik, salah satunya adalah akar pangkat ke-n suku ke-n akan konvergen ke pecahan emas:

```
>$(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))):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-223.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, ....

```
>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 2x2, dan setiap x[n] merupakan matriks/vektor 2x1.

```
>A=[1,1;1,2]; function suku(x,n) := A.x[,n-1]
```

```
>sequence("suku",[1;1],6)
```

```
Real 2 x 6 matrix
```

|   |   |   |    |     |
|---|---|---|----|-----|
| 1 | 2 | 5 | 13 | ... |
| 1 | 3 | 8 | 21 | ... |

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "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

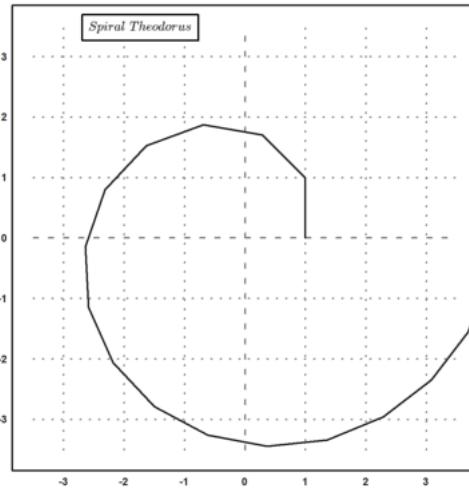
Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

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.

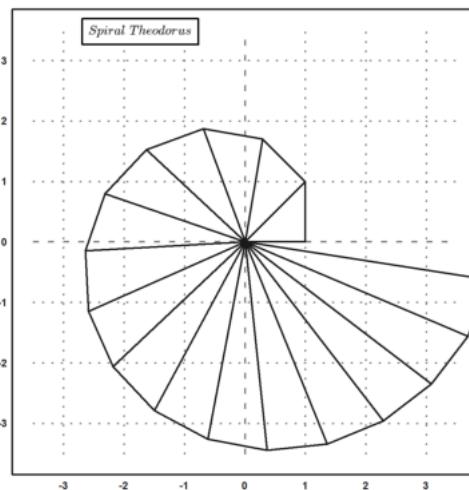
```
>x=sequence("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox(latex("Spiral\\ Theodorus"),0.4);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-224.png

Selanjutnya dihubungkan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end;
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-225.png

>

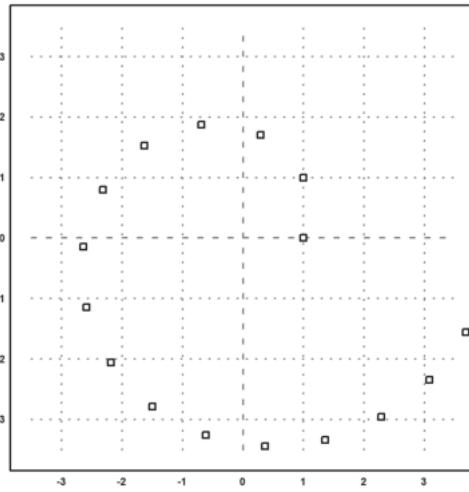
Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memerlukan indeks dan bilangan kompleks. Dalam hal ini diigunakan vektor kolom pada bidang.

```
>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);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-226.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, 0.739085133213~

Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan  $x=\cos(x)$ .

>h=expand(hasil,100), cos(h) << h

~0.73908513309, 0.73908513333~  
1

Iterasi juga dapat digunakan pada fungsi yang didefinisikan.

>function f(x) := (x+2/x)/2

Iterasi  $x(n+1)=f(x(n))$  akan konvergen ke akar kuadrat 2.

>iterate("f",2), sqrt(2)

1.41421356237  
1.41421356237

Jika pada perintah iterate diberikan tambahan parameter n, maka hasil iterasinya akan ditampilkan mulai dari iterasi pertama sampai ke-n.

>iterate("f",2,5)

[2, 1.5, 1.41667, 1.41422, 1.41421, 1.41421]

Untuk iterasi ini tidak dapat dilakukan terhadap interval.

>niterate("f",1,2,5)

[ ~1,2~, ~1,2~, ~1,2~, ~1,2~, ~1,2~, ~1,2~ ]

Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya adalah perhitungan dengan interval bersifat terlalu longgar. Untuk meingkatkan perhitungan pada ekspresi dapat digunakan pembagian intervalnya, menggunakan fungsi ieval().

>function s(x) := ieval("(x+2/x)/2",x,10)

Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal, dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang memuat akar persamaan:

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

```
~0.99999999999999778,1.0000000000000022~ ...  
~4.999999999999911,5.000000000000089~ ...
```

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
```

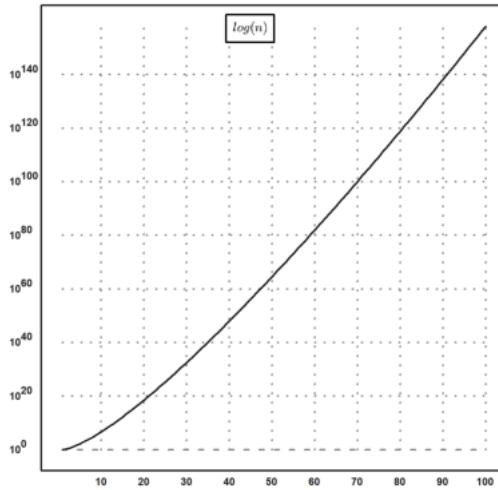
Penggabungan matriks menggunakan tanda “|” dapat digunakan untuk menyimpan semua hasil iterasi.

```
>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(latex(&log(n)),x=0.5);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-227.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
```

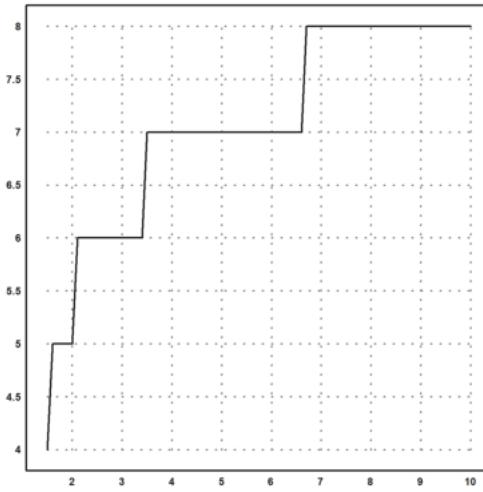
Misalnya, berikut adalah iterasi untuk mendapatkan hampiran akar kuadrat 2, cukup cepat, konvergen pada iterasi ke-5, jika dimulai dari hampiran awal 2.

```
>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);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-228.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
```

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

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);
```

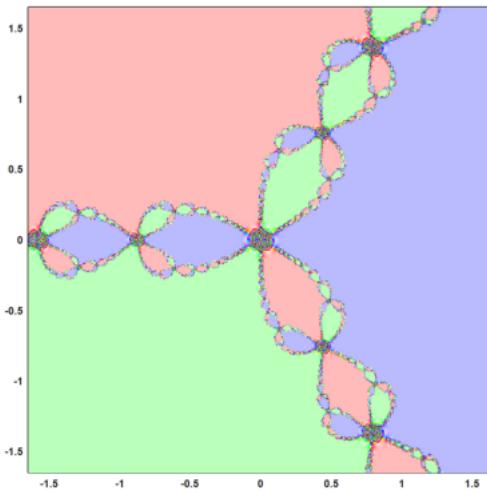
Berikut adalah akar-akar polinomial di atas.

```
>z=&solve(p)()
[ -0.5+0.866025i, -0.5-0.866025i, 1+0i ]
```

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);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-230.png

## 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

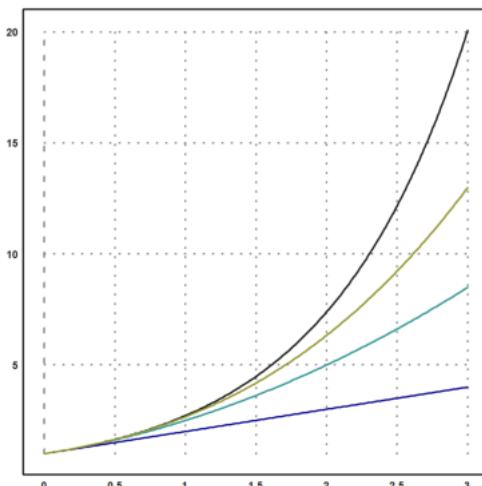
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); \$deret // barisan deret Taylor untuk  $e^x$

$$[ 1 + x, 1 + x + \frac{x^2}{2}, 1 + x + \frac{x^2}{2} + \frac{x^3}{6} ]$$

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



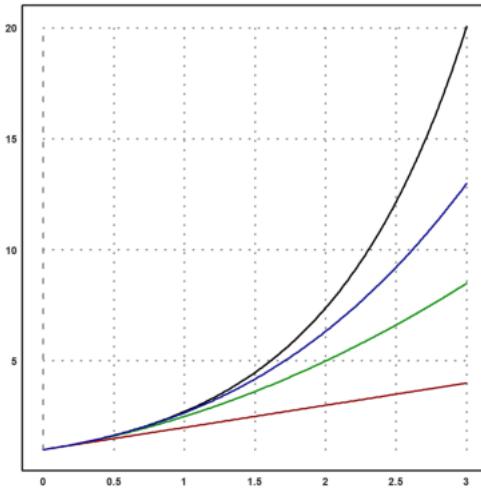
images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-232.png

Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

>\$deret[3]

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6}$$

>plot2d(["exp(x)",&deret[1],&deret[2],&deret[3]],0,3,color=1:4);



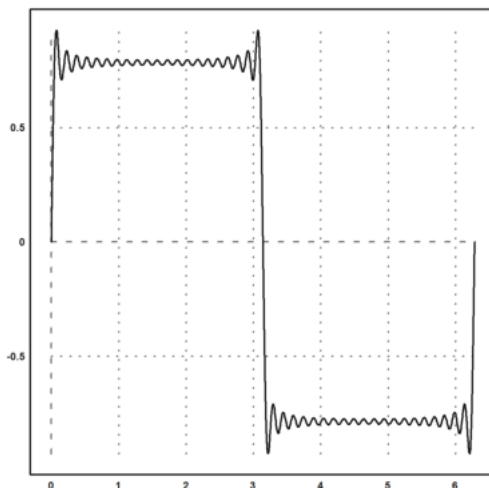
images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-234.png

```
>sum(sin(k*x)/k,k,1,5)
```

$$\sin x + \frac{\sin (2x)}{2} + \frac{\sin (3x)}{3} + \frac{\sin (4x)}{4} + \frac{\sin (5x)}{5}$$

Berikut adalah cara menggambar kurva

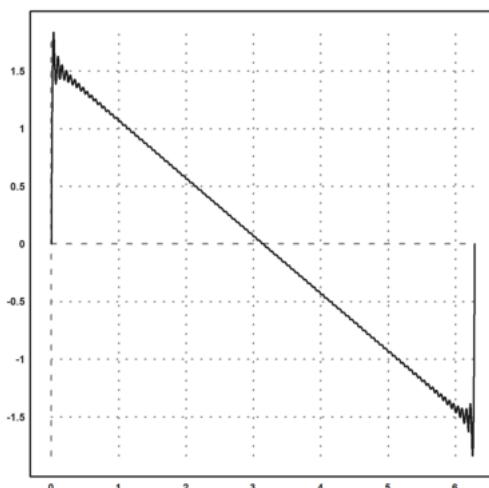
```
>plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-236.png

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);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4kalkulus-237.png

## Tabel Fungsi

Terdapat cara menarik untuk menghasilkan barisan dengan ekspresi Maxima. Perintah mxmtable() berguna untuk menampilkan dan menggambar barisan dan menghasilkan barisan sebagai vektor kolom.

Sebagai contoh berikut adalah barisan turunan ke-n  $x^n$  di  $x=1$ .

```
>imxmtable("diffat(x^n,x=1,n)","n",1,8,frac=1);

Function imxmtable not found.
Try list ... to find functions!
Error in:
imxmtable("diffat(x^n,x=1,n)","n",1,8,frac=1); ...

>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:menghitung
deret secara simbolik
```

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

```
>$'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))
```

$$\sum_{k=0}^{\infty} \frac{1}{k+3^k} = \sum_{k=0}^{\infty} \frac{1}{k+3^k}$$

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);
```

$$a \sum_{k=0}^{\infty} x^k = \frac{a}{1-x}$$

Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1.

```
>$'sum(x^k/k!,k,0,inf)=ev(sum(x^k/k!,k,0,inf),simpsum=true)
```

$$\sum_{k=0}^{\infty} \frac{x^k}{k!} = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

```
>$limit(sum(x^k/k!,k,0,n),n,inf)
```

$$\lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{x^k}{k!}$$

```
>function d(n) &= sum(1/(k^2-k),k,2,n); $'d(n)=d(n)
```

$$d(n) = \sum_{k=2}^n \frac{1}{-k+k^2}$$

```
>d(10)=ev(d(10),simpsum=true)
```

```


$$\sum_{k=2}^{10} \frac{1}{-k+k^2} = \frac{9}{10}$$

>$d(100)=ev(d(100),simpsum=true)


$$\sum_{k=2}^{100} \frac{1}{-k+k^2} = \frac{99}{100}$$

# Deret Taylor

Deret Taylor suatu fungsi f yang diferensiabel sampai tak hingga di sekitar x=a adalah:

>$'e^x=taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai suku ke-11


$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040} + \frac{x^8}{40320} + \frac{x^9}{362880} + \frac{x^{10}}{3628800}$$

>$'log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1


$$\log x = -1 - \frac{(-1+x)^2}{2} + \frac{(-1+x)^3}{3} - \frac{(-1+x)^4}{4} + \frac{(-1+x)^5}{5} - \frac{(-1+x)^6}{6} + \frac{(-1+x)^7}{7} - \frac{(-1+x)^8}{8} + \frac{(-1+x)^9}{9}$$


```

## 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:=texheight()*1.5: nilai asli untuk parameter d
setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang koordinat
setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas sumbu-x dan y adalah -r sd r
plotPoint (P, "P"): menggambar titik P dan diberi label "P"
plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label "AB" sejauh d
plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d
plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"
plotLabel (label, P, V, d): menuliskan label pada posisi P

```

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

```

turn(v, phi): memutar vektor v sejauh phi
turnLeft(v): memutar vektor v ke kiri
turnRight(v): memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang vektorv dan w.
lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh. ax+by=c.
lineWithDirection(A,v): garis melalui A searah vektor v
getLineDirection(g): vektor arah (gradien) garis g
getNormal(g): vektor normal (tegak lurus) garis g
getPointOnLine(g): titik pada garis g
perpendicular(A, g): garis melalui A tegak lurus garis g
parallel (A, g): garis melalui A sejajar garis g
lineIntersection(g, h): titik potong garis g dan h
projectToLine(A, g): proyeksi titik A pada garis g
distance(A, B): jarak titik A dan B
distanceSquared(A, B): kuadrat jarak A dan B
quadrance(A, B): kuadrat jarak A dan B
areaTriangle(A, B, C): luas segitiga ABC
computeAngle(A, B, C): besar sudut <ABC
angleBisector(A, B, C): garis bagi sudut <ABC
circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r
getCircleCenter(c): pusat lingkaran c
getCircleRadius(c): jari-jari lingkaran c
circleThrough(A,B,C): lingkaran melalui A, B, C
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan lingkaran c
circleCircleIntersections (c1, c2): titik potong lingkaran c1 dan c2
planeThrough(A, B, C): bidang melalui titik A, B, C

```

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

getLineEquation (g,x,y): persamaan garis g dinyatakan dalam x dan y  
 getHesseForm (g,x,y,A): bentuk Hesse garis g dinyatakan dalam x dan y dengan titik A pada  
 sisi positif (kanan/atas) garis  
 quad(A,B): kuadrat jarak AB  
 spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni  $\sin(\alpha)^2$  dengan alpha sudut yang menghadap sisi a.  
 crosslaw(a,b,c,sa): persamaan 3 quads dan 1 spread pada segitiga dengan panjang sisi a, b, c.  
 triplespread(sa,sb,sc): persamaan 3 spread sa,sb,sc yang memebntuk suatu segitiga  
 doublespread(sa): Spread sudut rangkap Spread  $2\phi$ , dengan  $sa=\sin(\phi)^2$  spread a.

## Contoh 1: Luas, Lingkaran Luar, Lingkaran Dalam Segitiga

Untuk menggambar objek-objek geometri, langkah pertama adalah menentukan rentang sumbu-sumbu koordinat. Semua objek geometri akan digambar pada satu bidang koordinat, sampai didefinisikan bidang koordinat yang baru.

```
>setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
```

Sekarang, tetapkan tiga titik dan plotlah.

```
>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
```

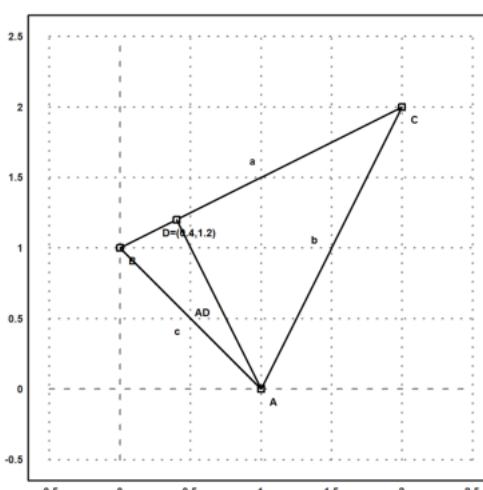
Dan persimpangannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Plot itu.

```
>plotPoint(D,value=1); // koordinat D ditampilkan
```

```
>aspect(1); plotSegment(A,D); // tampilkan semua gambar hasil plot...()
```



[images/23030630004\\_Siti%20Faltipah%20Hayani\\_EMT4Geometry-001.png](images/23030630004_Siti%20Faltipah%20Hayani_EMT4Geometry-001.png)

Hitung luas ABC:

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
```

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.

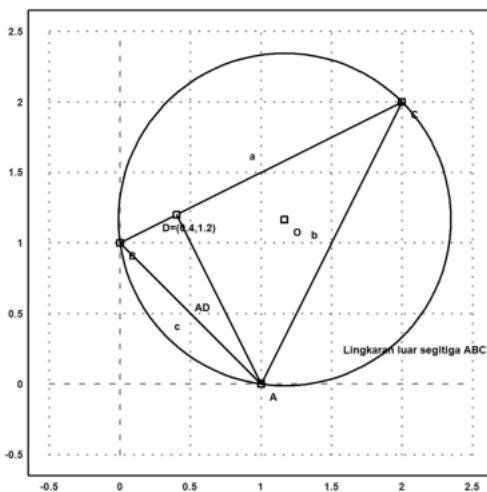
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC

>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-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.

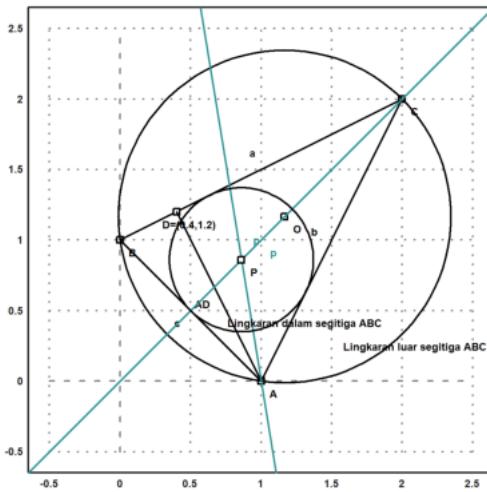
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut

>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-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?

```
>setPlotRange(-2.5,4.5,-2.5,4.5);
```

```
>A=[-2,1]; plotPoint(A,"A");
```

```
>B=[1,-2]; plotPoint(B,"B");
```

```
>C=[4,4]; plotPoint(C,"C");
```

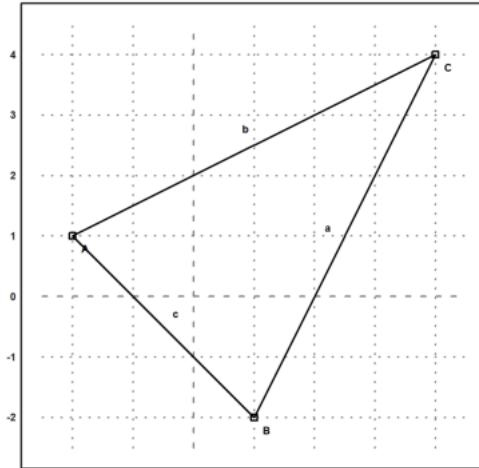
2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. Merupakan segitiga apakah itu?

```
>plotSegment(A,B,"c");
```

```
>plotSegment(B,C,"a");
```

```
>plotSegment(A,C,"b");
```

```
>aspect(1):
```



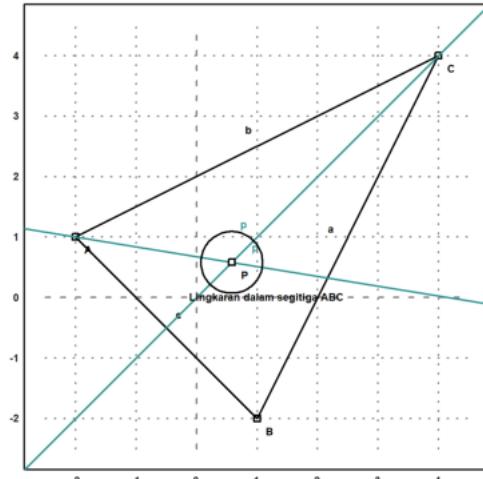
3. Hitung luas segitiga tersebut.

```
>distance(A,D)*distance(B,C)/2
```

8.0777472107

4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

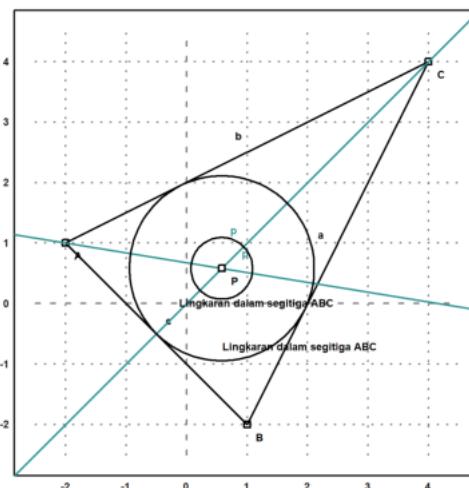
```
>l=angleBisector(A,C,B);
>g=angleBisector(C,A,B);
>P=lineIntersection(l,g)
[0.581139, 0.581139]
>color(5); plotLine(l); plotLine(g); color(1);
>plotPoint(P,"P");
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC");
```



5. Gambar jari-jari

lingkaran dalam.

```
>r=norm(P-projectToLine(P,lineThrough(A,B)))
1.52896119631
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC");
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-006.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
```

$$2y - x = 2$$

$$\left[ y = \frac{x}{2} + 1 \right]$$

```
>$getLineEquation(lineThrough([x1,y1],[x2,y2]),x,y), $solve(% ,y) // persamaan garis melalui (x1, y1) dan (x2, y2)
```

$$x(y_1 - y_2) + (x_2 - x_1)y = x_1(y_1 - y_2) + (x_2 - x_1)y_1$$

$$\left[ y = \frac{-(x_1 - x)y_2 - (x - x_2)y_1}{x_2 - x_1} \right]$$

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
```

$$(x_1 - 1)y - x y_1 = -y_1$$

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

$$\left[ 2, 1, 2 \right]$$

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

$$\begin{bmatrix} 2 & 6 \\ -, - \\ 5 & 5 \end{bmatrix}$$

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\left[ \frac{2}{5}, \frac{6}{5} \right]$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[ \frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}} \right]$$

```
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

$$\frac{5}{3\sqrt{2}}$$

$$1.178511301977579$$

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 garis bagi sudut
```

$$\left[ \frac{\sqrt{2}\sqrt{5} + 2}{6}, \frac{\sqrt{2}\sqrt{5} + 2}{6} \right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

$$[0.86038, 0.86038]$$

## Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga bisa memotong garis dengan lingkaran, dan lingkaran dengan lingkaran.

```
>A &:= [1,0]; c=circleWithCenter(A,4);
```

```
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
```

```
>setPlotRange(5); plotCircle(c); plotLine(l);
```

Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik perpotongan.

```
>{P1,P2,f}=lineCircleIntersections(l,c);
```

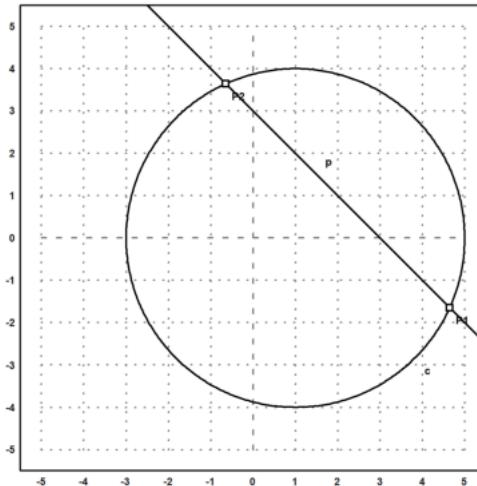
```
>P1, P2, f
```

```
[4.64575, -1.64575]
```

```
[-0.645751, 3.64575]
```

```
2
```

```
>plotPoint(P1); plotPoint(P2);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-020.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)/2 + 1]]
```

Akan ditunjukkan bahwa sudut-sudut yang menghadap bsuusr yang sama adalah sama besar.

```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
```

```
>deprint(computeAngle(P1,C,P2))
```

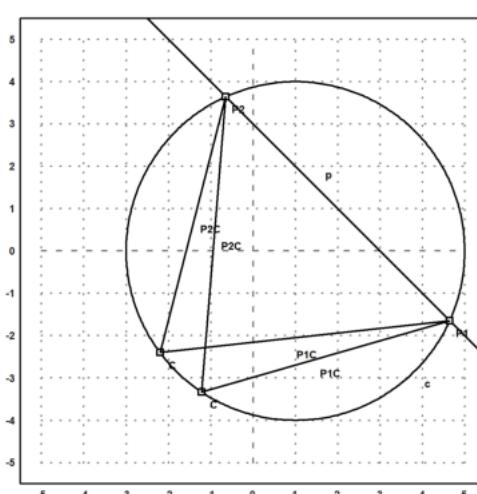
```
69°17'42.68''
```

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
```

```
>deprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

```
>insimg;
```

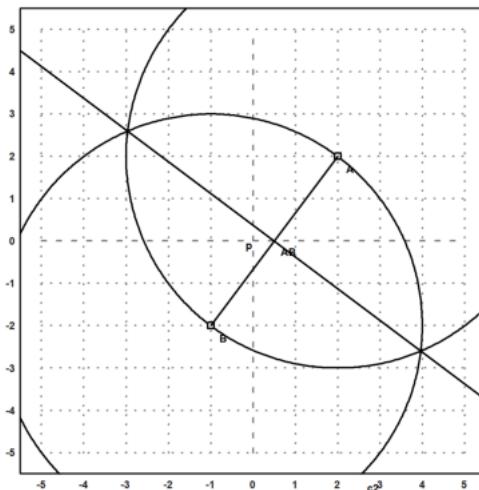


## Garis Sumbu

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)
```

$$y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2}$$

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sama sekali berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2}$$

```
>h &= getLineEquation(lineThrough(A,B),x,y);
```

```
>$solve(h,y)
```

$$[y = \frac{(b_2 - a_2)x - a_1 b_2 + a_2 b_1}{b_1 - a_1}]$$

Perhatikan hasil kali gradien garis g dan h adalah:

Artinya kedua garis tegak lurus.

### Contoh 3: Rumus Heron

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a, b dan c adalah:

atau bisa dituliskan dalam bentuk lain:

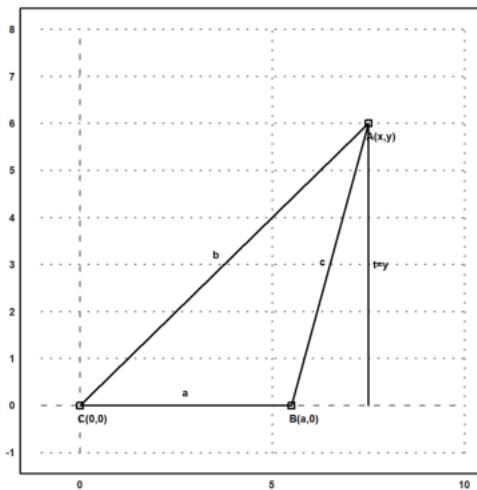
Untuk membuktikan hal ini kita misalkan C(0,0), B(a,0) dan A(x,y), b=AC, c=AB. Luas segitiga ABC adalah

Nilai y didapat dengan menyelesaikan sistem persamaan:

```
>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-027.png

```
>&assume(a>0); sol &= solve([x^2+y^2=b^2,(x-a)^2+y^2=c^2],[x,y])
[]
```

Ekstrak solusi dari 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)
```

$$H(a, b, c) = \frac{a |ysol|}{2}$$

```
>$'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.

```
>$'Luas=H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

$$\text{Luas} = \frac{3 |ysol|}{2}$$

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 sengan panjang sisi 3, 4, x
(1<= x <=7)

Variable or function ysol not found.
Error in expression: 3*abs(ysol)/2
%ploteval:
y0=f$(x[1],args());
adaptiveevalone:
s=%ploteval(g$,t,args());
Try "trace errors" to inspect local variables after errors.
plot2d:
dw/n,dw/n^2,dw/n,auto;args());

```

Kasus umum juga bisa digunakan.

```
>$solve(diff(H(a,b,c)^2,c)=0,c)
```

all

Sekarang mari kita cari himpunan semua titik di mana  $b+c=d$  untuk suatu konstanta  $d$ . Sudah diketahui bahwa ini adalah sebuah ellips.

```

>s1 &= subst(d-c,b,sol[2]); $s1

Maxima said:
part: invalid index of list or matrix.
-- an error. To debug this try: debugmode(true);

Error in:
s1 &= subst(d-c,b,sol[2]); $s1 ...
^

```

Dan membuat fungsi-fungsi ini.

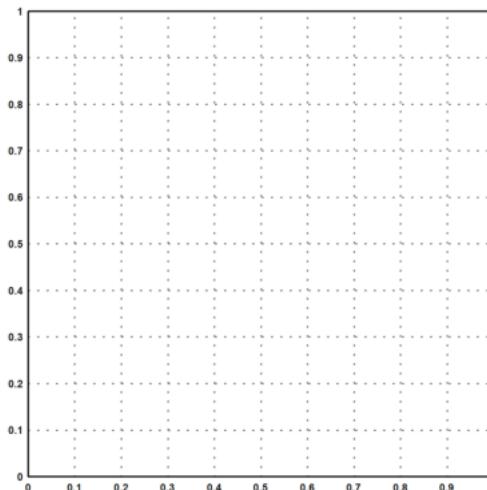
```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); $fy(a,c,d)
```

0

0

Sekarang kita dapat menggambar himpunan tersebut. Sisi  $b$  bervariasi dari 1 hingga 4. Sudah diketahui bahwa kita mendapatkan sebuah ellips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```



[images/23030630004\\_Siti%20Faltipah%20Hayani\\_EMT4Geometry-034.png](#)

Kita dapat memeriksa persamaan umum untuk ellips ini, yaitu

$$\frac{(x - x_m)^2}{u^2} + \frac{(y - y_m)^2}{v^2} = 1,$$

di mana  $(x_m, y_m)$  adalah pusat, dan  $u$  serta  $v$  adalah setengah sumbu.

```
>$ratsimp((fx(a,c,d)-a/2)^2/u^2+fy(a,c,d)^2/v^2 with [u=d/2,v=sqrt(d^2-a^2)/2])
```

$$\frac{a^2}{d^2}$$

Kita melihat bahwa tinggi dan luas segitiga adalah maksimal untuk  $x=0$ . 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)),a)=0,diff(H(a,b,d-(a+b)),b)=0]; $eqns
```

$$[ \frac{a \text{ysol}^2}{2} = 0, 0 = 0 ]$$

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  $a+b+d=d$ .

```
>&solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la, ...
> diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la])
```

Maxima said:

```
diff: second argument must be a variable; found [1,0,4]
-- an error. To debug this try: debugmode(true);
```

Error in:

```
... la,      diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la]) ...
```

Kita dapat 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
```

```
[0, 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 segmen-segmen tersebut.

```
>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 lingkar.

```
>U &= lineIntersection(h,g);
```

Kita mendapatkan rumus untuk jari-jari lingkaran.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

$$\frac{\sqrt{a_2^2 + a_1^2} \sqrt{a_2^2 + a_1^2 - 2a a_1 + a^2}}{2 |a_2|}$$

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 untuk radius. Kita bisa mengecek, apakah hal ini benar dengan Maxima. Maxima akan memperhitungkannya hanya jika kita mengkuadratkannya.

```
>$c^2/sin(computeAngle(A,B,C))2 | factor
```

$$[\frac{a_2^2 + a_1^2}{a_2^2}, 0, \frac{16(a_2^2 + a_1^2)}{a_2^2}]$$

# 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.

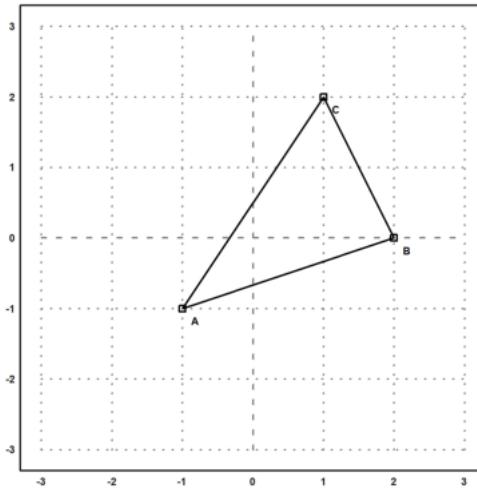
```
>A:=[-1,-1]; B:=[2,0]; C:=[1,2];
```

Untuk memplot objek geometris, kita menyiapkan area plot, dan menambahkan titik-titiknya. 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/23030630004\\_Siti%20Faltipah%20Hayani\\_EMT4Geometry-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(%,[x=C[1],y=C[2]])
```

$$\frac{3y - x + 2}{\sqrt{10}}$$

$$\frac{7}{\sqrt{10}}$$

Sekarang kita menghitung keliling ABC.

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

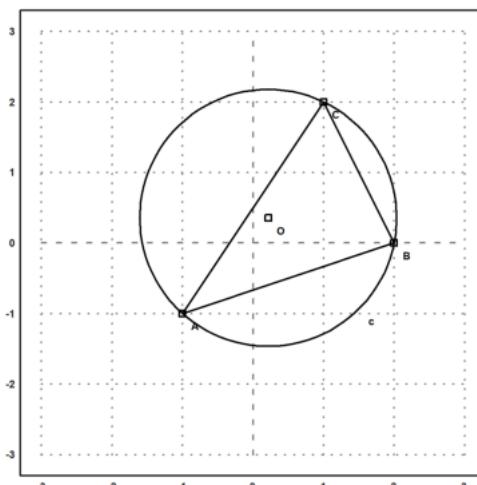
$$\left( y - \frac{5}{14} \right)^2 + \left( x - \frac{3}{14} \right)^2 = \frac{325}{98}$$

```
>O &= getCircleCenter(LL); $O
```

$$\left[ \frac{3}{14}, \frac{5}{14} \right]$$

Plot lingkaran dan pusatnya. Cu dan U adalah simbolik. Kami mengevaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O");
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-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
```

$$\left[ \frac{11}{7}, \frac{2}{7} \right]$$

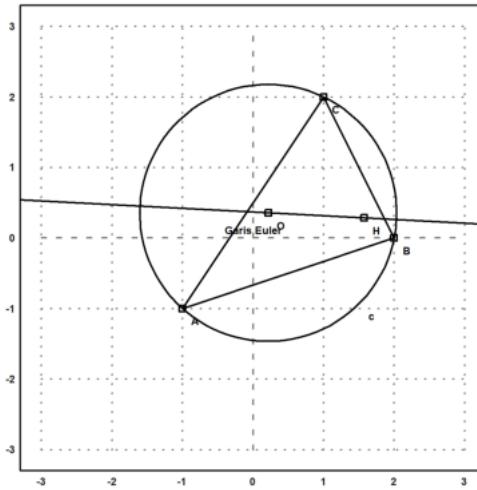
Sekarang kita dapat menghitung garis Euler dari segitiga tersebut.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$\$ - \frac{19y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan ke plot kami.

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler");
```



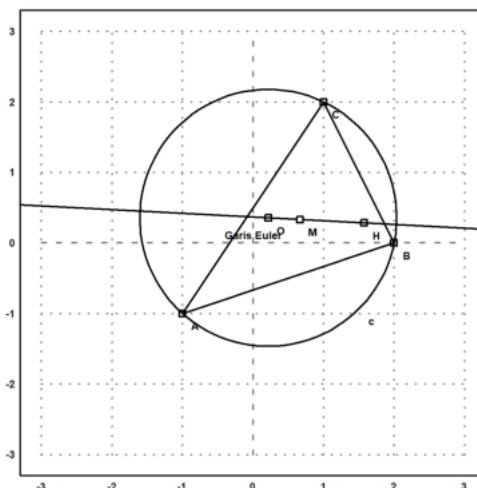
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-053.png

Pusat gravitasi harus berada pada garis ini.

>M &= (A+B+C)/3; \$getLineEquation(el,x,y) with

$$-\frac{1}{2} = -\frac{1}{2}$$

>plotPoint(M(),"M"); // titik berat



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-055.png

Teori mengatakan bahwa  $MH = 2 \cdot MO$ . Kita perlu menyederhanakan dengan radcan untuk mencapai hal ini.

>\$distance(M,H)/\$distance(M,O)|radcan

$2$

Fungsi-fungsi ini juga mencakup fungsi untuk sudut.

>\$computeAngle(A,C,B), deprint(%())

$$\arccos\left(\frac{4}{\sqrt{5}\sqrt{13}}\right)$$

$60^\circ 15' 18.43''$

Persamaan untuk bagian tengah lingkaran tidak terlalu bagus.

>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; \$Q

$$\left[ \frac{(2^{\frac{3}{2}} + 1)\sqrt{5}\sqrt{13} - 15\sqrt{2} + 3}{14}, \frac{(\sqrt{2} - 3)\sqrt{5}\sqrt{13} + 52^{\frac{3}{2}} + 5}{14} \right]$$

Mari kita hitung juga ekspresi untuk jari-jari lingkaran yang tertulis.

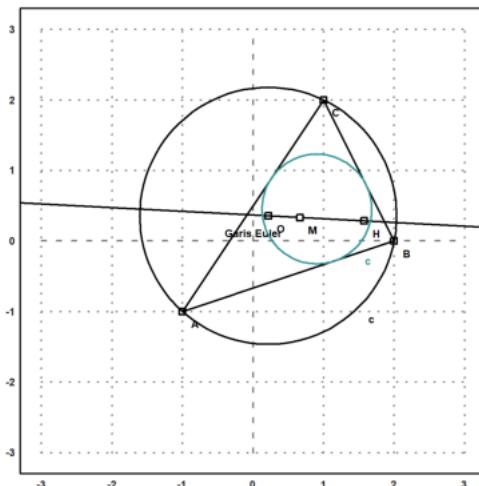
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; \$r

$$\frac{\sqrt{(-41\sqrt{2} - 31)\sqrt{5}\sqrt{13} + 115\sqrt{2} + 614}}{7\sqrt{2}}$$

>LD &= circleWithCenter(Q,r); // Lingkaran dalam

Mari kita tambahkan ini ke dalam plot.

>color(5); plotCircle(LD());



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-060.png

## Parabola

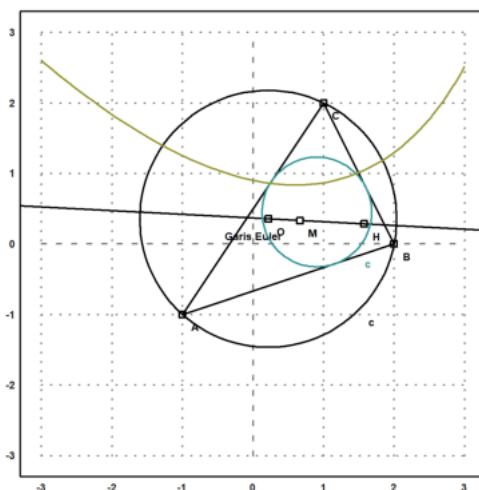
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-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)

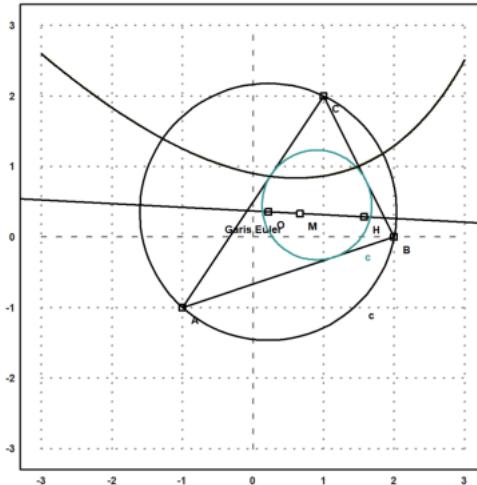
$$[y = -3x - \sqrt{70}\sqrt{9 - 2x} + 26, \\ y = -3x + \sqrt{70}\sqrt{9 - 2x} + 26]$$

Solusi pertama adalah

maxima: akar[1]

Dengan menambahkan solusi pertama ke dalam plot, maka akan terlihat bahwa ini adalah jalur yang kita cari. Teori mengatakan bahwa ini adalah sebuah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1);
```



[images/23030630004\\_Siti%20Faltipah%20Hayani\\_EMT4Geometry-063.png](images/23030630004_Siti%20Faltipah%20Hayani_EMT4Geometry-063.png)

```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan kurva di atas
```

$$g(x) = -3x - \sqrt{70} \sqrt{9 - 2x + 26}$$

```
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
```

```
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%)) // jarak T ke C
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

2.135605779339061

```
>U &= projectToLine(T,lineThrough(A,B)); $U // proyeksi T pada garis AB
```

$$\left[ \frac{80 - 3\sqrt{11}\sqrt{70}}{10}, \frac{20 - \sqrt{11}\sqrt{70}}{10} \right]$$

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%)) // jarak T ke AB
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

2.135605779339061

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

## Contoh 5: Trigonometri Rasional

Ini terinspirasi dari sebuah ceramah N.J. Wildberger. Dalam bukunya “Proporsi Ilahi”, Wildberger mengusulkan untuk mengganti gagasan klasik tentang jarak dan sudut dengan kuadrans dan penyebaran. Dengan menggunakan ini, memang memungkinkan untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap “rasional”.

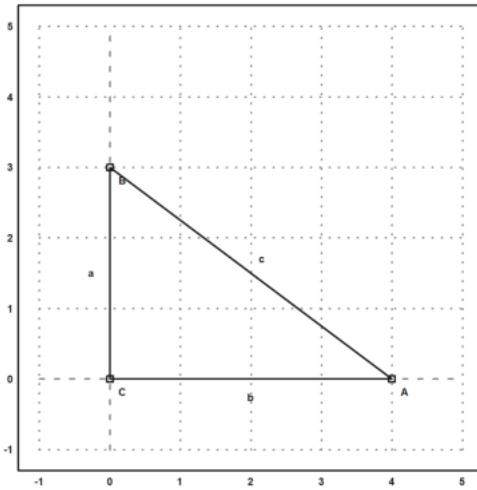
Berikut ini, saya akan memperkenalkan konsep-konsep tersebut, dan memecahkan beberapa masalah. Saya menggunakan komputasi simbolik Maxima di sini, yang menyembunyikan keuntungan utama dari trigonometri rasional yaitu komputasi dapat dilakukan dengan kertas dan pensil saja. Anda dipersilakan untuk memeriksa 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);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-070.png

Tentunya,

di mana wa 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 kuadran 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 kuadran 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 wa 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 kuadran 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)

$$[(bb - aa + \frac{7}{6})^2, (bb - aa + \frac{7}{6})^2, (bb - aa + \frac{5}{3\sqrt{2}})^2] = [\frac{14bb(1 - saa)}{3}, \frac{14bb(1 - saa)}{3}, \frac{52\frac{3}{2}bb(1 - saa)}{3}]$$

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 kuadran 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,x)
```

$$1024 = 1600 (1 - x)$$

$$[ x = \frac{9}{25} ]$$

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)
```

$$\left[ \left[ \frac{168 bb x + 36 bb^2 + (-72 aa - 84) bb + 36 aa^2 - 84 aa + 49}{36}, \frac{168 bb x + 36 bb^2 + (-72 aa - 84) bb + 36 aa^2 - 84 aa + 49}{36} \right] \right]$$

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 menggunakan 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)))
```

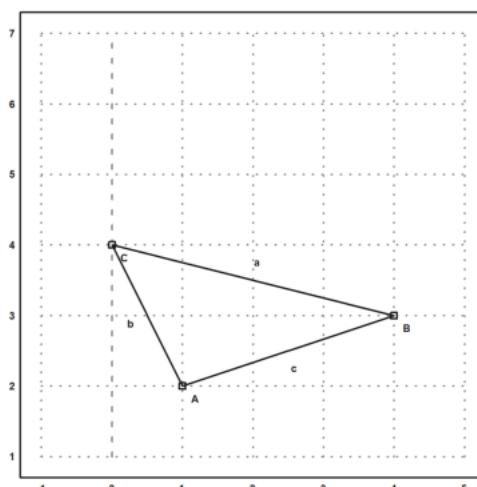
$67^\circ 47' 32.44''$

## Contoh Lain

Sekarang, mari kita coba contoh yang lebih lanjut.

Kita tentukan 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;
```



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 kuadran 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,

$$\begin{matrix} 10 \\ 5 \\ 17 \end{matrix}$$

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)()

$$\arccos\left(\frac{11}{\sqrt{10}\sqrt{17}}\right)$$

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(%), //((b+c-a)^=4b.c(1-x))

$$4 = 200(1 - x)$$

$$[x = \frac{49}{50}]$$

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))^2

$$\frac{49}{170}$$

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.

>\$sqrt(ha)

$$\frac{7}{\sqrt{17}}$$

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita berurusan dengan kuadran!

>\$sqrt(ha)\*sqrt(a)/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(b^2.c^2,a^2), \$factor(%\*c^2\*a^2/4)

$$\frac{-c^4 - (-2b^2 - 2a^2)c^2 - b^4 + 2a^2b^2 - a^4}{4a^2c^2}$$

$$\frac{(-c + b + a)(c - b + a)(c + b - a)(c + b + a)}{16}$$

## 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)

$$(sc + sb + sa)^2 = 2(sc^2 + sb^2 + sa^2) + 4sa sb sc$$

Aturan ini berlaku untuk tiga sudut yang berjumlah  $180^\circ$ .

$$\alpha + \beta + \gamma = \pi$$

Karena spread dari

$$\alpha, \pi - \alpha$$

sama, aturan triple spread juga benar, jika

$$\alpha + \beta = \gamma$$

Karena penyebaran sudut negatifnya sama, aturan penyebaran tiga kali lipat juga berlaku, jika

$$\alpha + \beta + \gamma = 0$$

Contohnya, kita bisa menghitung penyebaran sudut  $60^\circ$ . Hasilnya 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  $90^\circ$  jelas adalah 1. Jika dua sudut ditambahkan ke  $90^\circ$ , penyebarannya akan menyelesaikan persamaan penyebaran tiga dengan a, b, 1. Dengan perhitungan berikut, kita mendapatkan  $a + b = 1$ .

>\$triplespread(x,y,1), \$solve(% ,x)

$$(y + x + 1)^2 = 2(y^2 + x^2 + 1) + 4xy$$

$$[x = 1 - y]$$

Karena penyebaran  $180^\circ$ -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.

>\$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))

$$[x = 4a - 4a^2, x = 0]$$

$$-4(a - 1)a$$

## Angle Bisectors

Ini adalah situasi yang sudah kita ketahui.

```
> 8C:=[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;

Commands must be separated by semicolon or comma!
Found: &:=;:=[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; (character 38)
You can disable this in the Options menu.
Error in:
  8C&:=;:=[0,0]; A&:=[4,0]; B&:=[0,3]; setPlotRange(-1,5,-1,5); pl
...
^
```

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  $180^\circ$ -wa.

```
>$triplespread(x,x,a/(a+b)), $solve(%), sa2 &= rhs(%[1]); $sa2
```

$$\left(2x + \frac{a}{b+a}\right)^2 = 2\left(2x^2 + \frac{a^2}{(b+a)^2}\right) + \frac{4ax^2}{b+a}$$

$$[x = \frac{-\sqrt{b}\sqrt{b+a} + b+a}{2b+2a}, x = \frac{\sqrt{b}\sqrt{b+a} + b+a}{2b+2a}]$$

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

Mari kita periksa persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

Kita bisa mencetak sudut dalam Euler, setelah mentransfer penyebaran ke radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

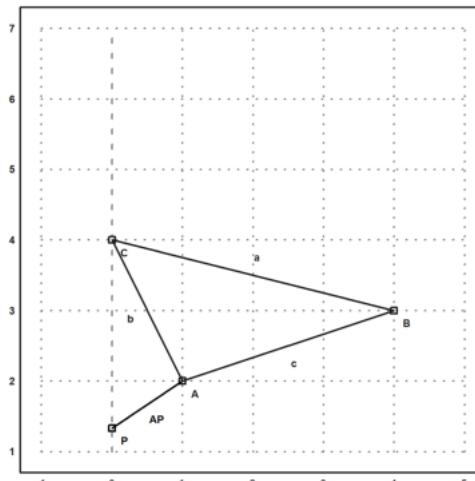
$18^\circ 26' 5.82''$

Titik P adalah perpotongan garis bagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
```

[0, 1.33333]

```
>plotPoint(P,"P"); plotSegment(A,P);
```



Mari kita periksa sudut-sudutnya dalam contoh spesifik kita.

>computeAngle(C,A,P), computeAngle(P,A,B)

1.69515132134  
2.87534060444

Sekarang kita hitung panjang garis bagi AP.

Kita menggunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa berlaku dalam segitiga apa pun. Kuadratkan, ini diterjemahkan ke dalam apa yang disebut “hukum penyebaran”

where  $a, b, c$  denote quadrances. di mana  $a, b, c$  menunjukkan kuadrannya.

Karena spread CPA adalah  $1 - sa^2$ , kita mendapatkan  $bisa / l = b / (1 - sa^2)$  dan bisa menghitung bisa (kuadrant dari pembagi sudut).

>&factor(ratsimp(b/(1-sa2))); bisa &= %; \$bisa

$$\frac{2b(b+a)}{\sqrt{b}\sqrt{b+a+b+a}}$$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2]))), distance(A,P)

4.21637021356  
1.20185042515

Kita juga dapat menghitung P dengan menggunakan rumus penyebaran.

>py&=factor(ratsimp(sa2\*bisa)); \$py

$$-\frac{b(\sqrt{b}\sqrt{b+a} - b-a)}{\sqrt{b}\sqrt{b+a+b+a}}$$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

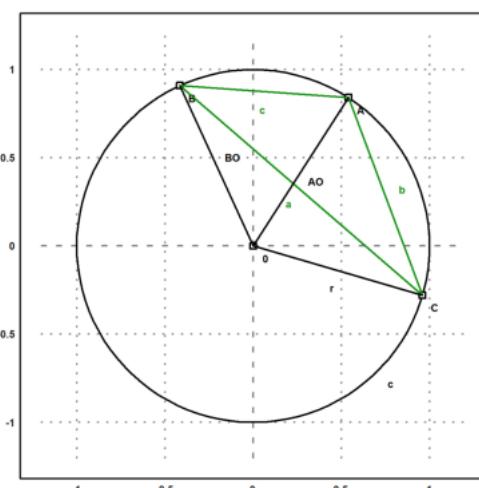
>sqrt(mxmeval("at(py,[a=3^2,b=4^2]))

1.33333333333

## 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;
```



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
- 
$$\frac{a b c}{c^2 - 2 b c + a (-2 c - 2 b) + b^2 + a^2}$$

```

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 akor.

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

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

## Preliminary remark

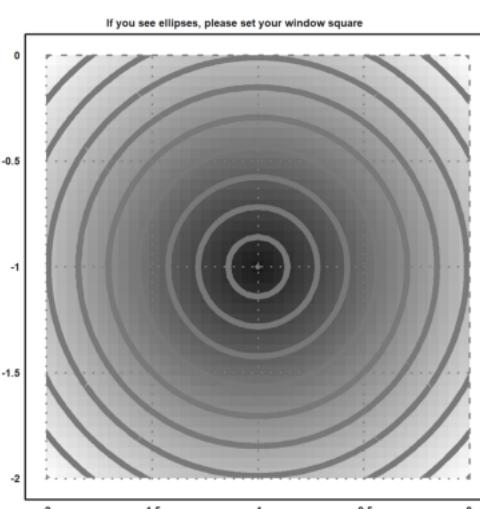
Fungsi yang, pada sebuah titik M pada bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis-garis tingkat yang cukup sederhana: lingkaran yang berpusat di A.

```
>&remvalue();
```

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

```
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)
```

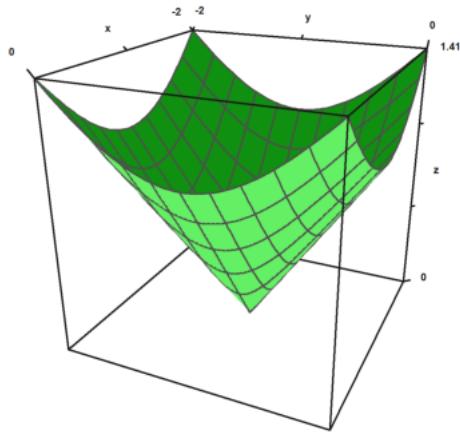
```
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...
> title="If you see ellipses, please set your window square");
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-115.png

dan grafiknya juga cukup sederhana: bagian atas kerucut:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0);
```



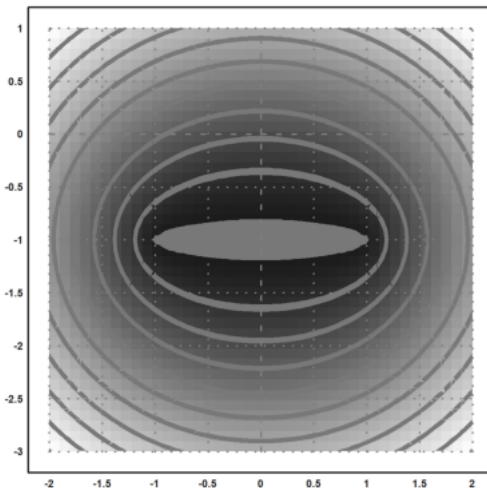
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-116.png

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 ellips, titik fokusnya adalah A dan B; kecuali AB minimum yang konstan pada segmen [AB]:

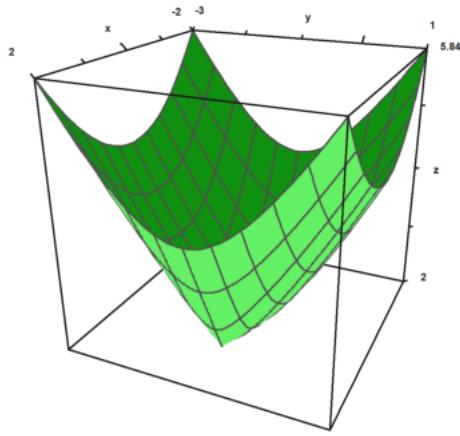
```
>B=[1,-1];
>function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2+(y-B[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-117.png

Grafiknya lebih menarik:

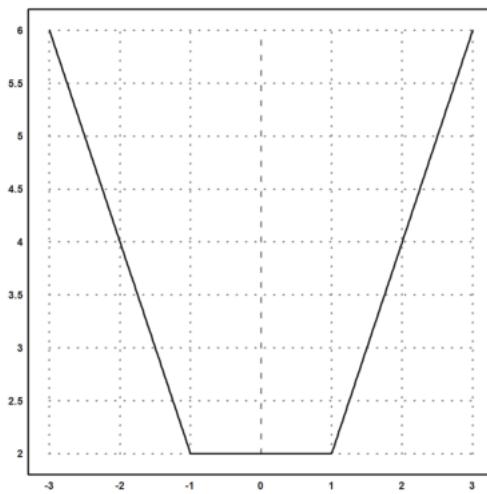
```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-118.png

Pembatasan pada garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-119.png

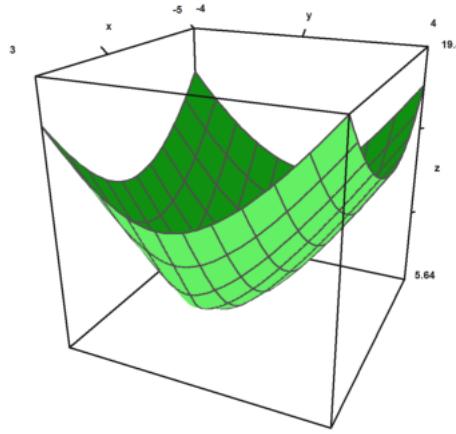
### Three points

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  $120^\circ$  (katakanlah di A), maka minimum dicapai pada titik ini (katakanlah AB+AC).

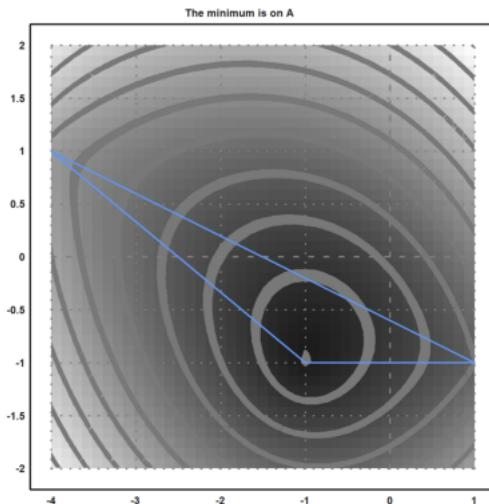
Contoh:

```
>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;
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-120.png

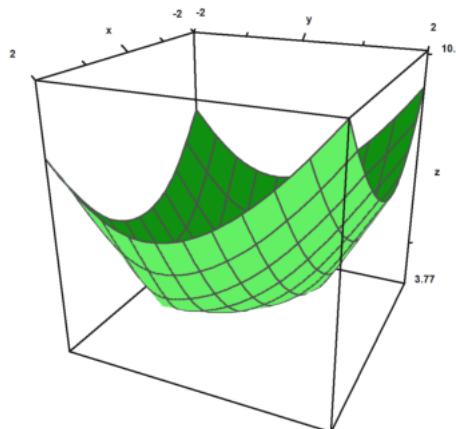
```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A); plot2d(P[1],P[2],add=1,color=12);
>insimg;
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-121.png

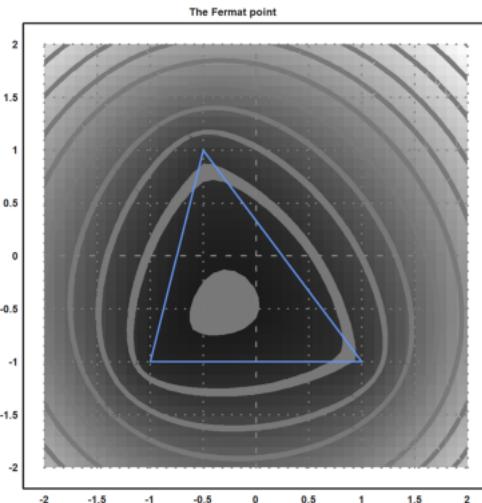
Tetapi jika semua sudut segitiga ABC kurang dari  $120^\circ$ , minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (masing-masing  $120^\circ$ ):

```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-122.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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-123.png

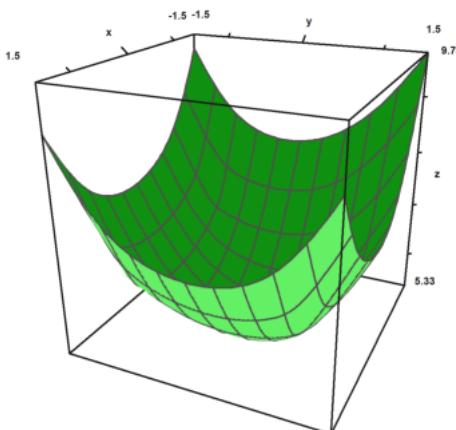
Merupakan kegiatan yang menarik untuk merealisasikan gambar di atas dengan perangkat lunak geometri; sebagai contoh, saya tahu sebuah perangkat lunak yang ditulis dalam bahasa Java yang memiliki instruksi “garis kontur”...

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...

## Four points

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!

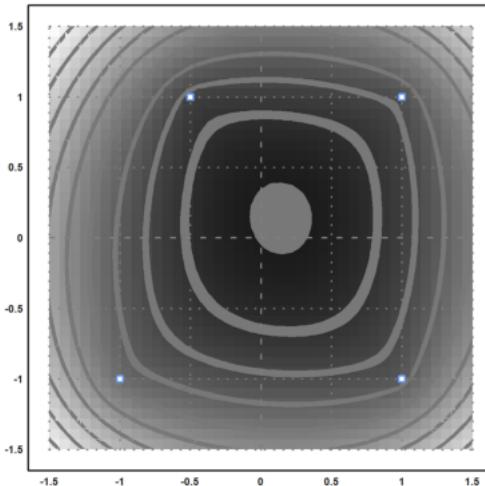
```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-124.png

```
>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-125.png

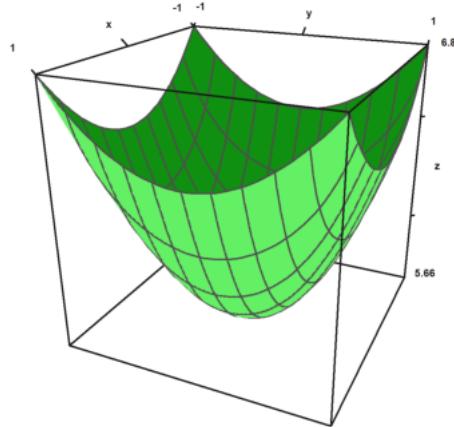
Masih ada nilai minimum dan tidak ada simpul A, B, C, maupun D:

```
>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...

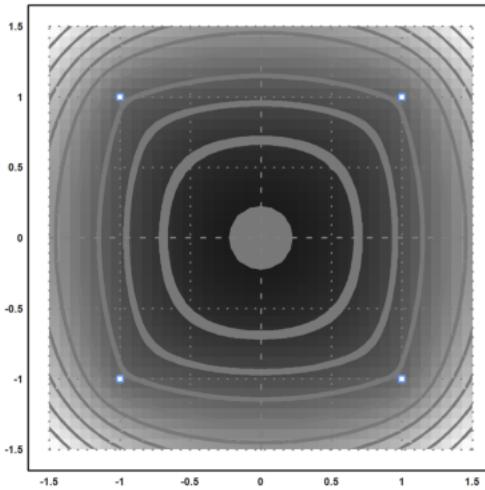
Karena ABCD adalah sebuah bujur sangkar, kita berharap bahwa titik optimalnya adalah pusat dari ABCD:

```
>C=[-1,1];  
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1);
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-126.png

```
>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-127.png

### Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan pvengine.exe pada jalur 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]
```

Kemudian baris ketiga.

```
>g &= lineThrough([-1,0],[1,1])
```

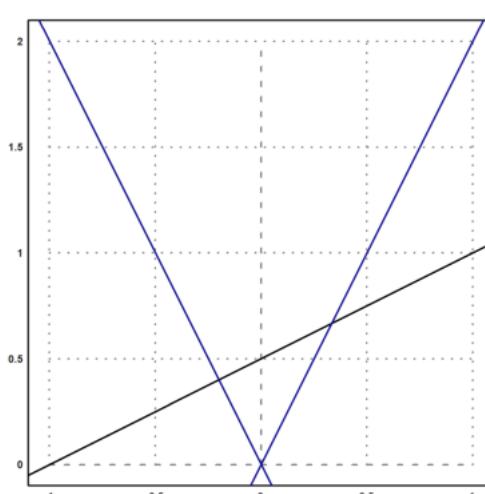
```
[ - 1, 2, 1]
```

Kami merencanakan semuanya sejauh ini.

```
>setPlotRange(-1,1,0,2);
```

```
>color(black); plotLine(g(), "")
```

```
>a:=2; color(blue); plotLine(g1(),""); plotLine(g2(),"");
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-128.png

Sekarang, kita ambil titik umum pada sumbu y.

```
>P &= [0,u]
```

```
[0, u]
```

Hitung jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

$$\sqrt{\left(\frac{a^2 u}{a^2 + 1} - u\right)^2 + \frac{a^2 u^2}{(a^2 + 1)^2}}$$

Hitung jarak ke g.

```
>d &= distance(P,projectToLine(P,g)); $d
```

$$\sqrt{\left(\frac{u + 2}{5} - u\right)^2 + \frac{(2u - 1)^2}{25}}$$

Dan temukan pusat kedua lingkaran, yang jaraknya sama.

```
>sol &= solve(d1^2=d2,u); $sol
```

$$[u = \frac{-\sqrt{5}\sqrt{a^2 + 1} + 2a^2 + 2}{4a^2 - 1}, u = \frac{\sqrt{5}\sqrt{a^2 + 1} + 2a^2 + 2}{4a^2 - 1}]$$

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]
```

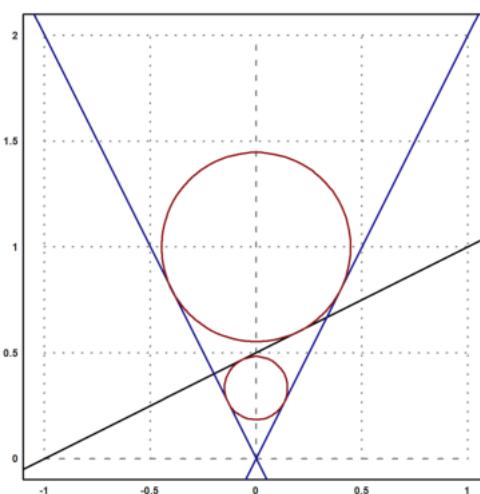
Plot lingkaran-lingkaran tersebut ke dalam gambar.

```
>color(red);
```

```
>plotCircle(circleWithCenter([0,u[1]],dd[1]),"");
```

```
>plotCircle(circleWithCenter([0,u[2]],dd[2]),"");
```

```
>insimg;
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-132.png

## Plot with 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.

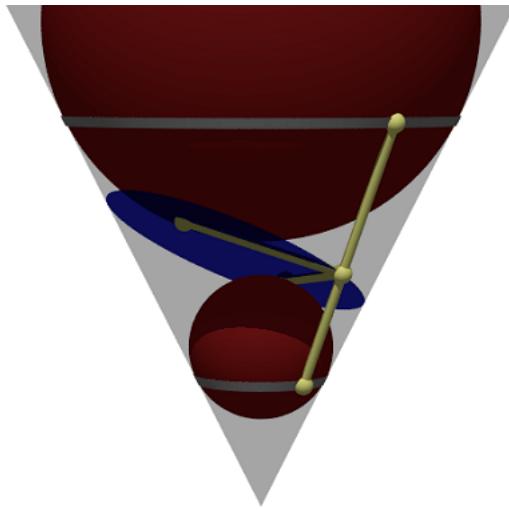
Pertama kita memuat fungsi povray.

```
>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]],g());
P1=turnz([P1[1],0,P1[2]]);
>writeln(povpoint(P1,povlook(yellow)));
>P2=projectToLine([0,u[2]],g());
P2=turnz([P2[1],0,P2[2]]);
>writeln(povpoint(P2,povlook(yellow)));
Kemudian, kita menghasilkan dua titik di mana bola-bola tersebut menyentuh bidang. Ini adalah fokus ellips.
>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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-133.png

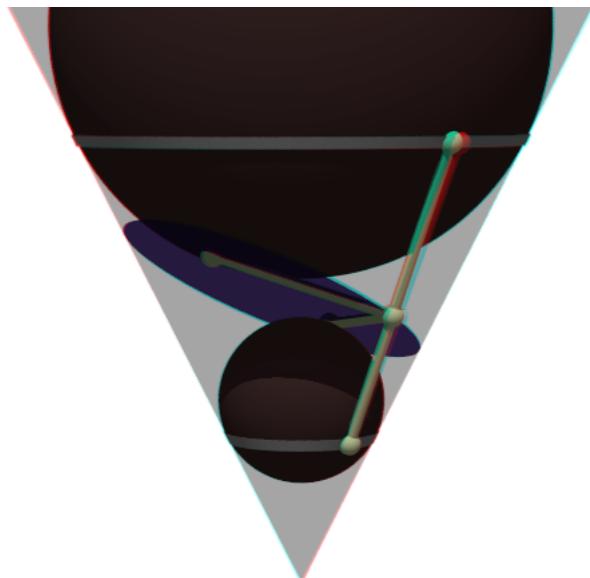
Untuk mendapatkan Anaglyph ini, kita harus memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali nanti.

```
>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,dp[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(P4,P1,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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-134.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 begitu 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)-&gt;" 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  $180^\circ$  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 segitiga. Catatan: Untuk segitiga kecil, cara ini tidak akurat karena kesalahan pengurangan dalam asum- $\pi$ .

```
>(asum- $\pi$ )*rearth( $48^\circ$ )^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( $48^\circ$ )^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 - Monas ...  
^
```

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 menyesuaikannya pada 1/10 dari jarak total.

```
>p=P1; dist=esdist(P1,P2); ...  
> loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
79.69°  
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'
```

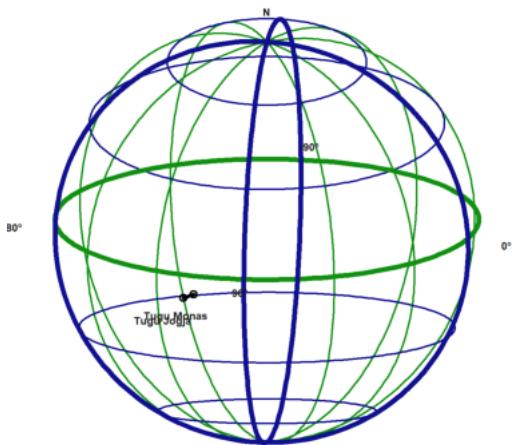
Kami menulis sebuah fungsi, yang memplot bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
```

```
useglobal;  
plotearth;  
plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");  
plotposline(v);  
endfunction
```

Sekarang rencanakan semuanya.

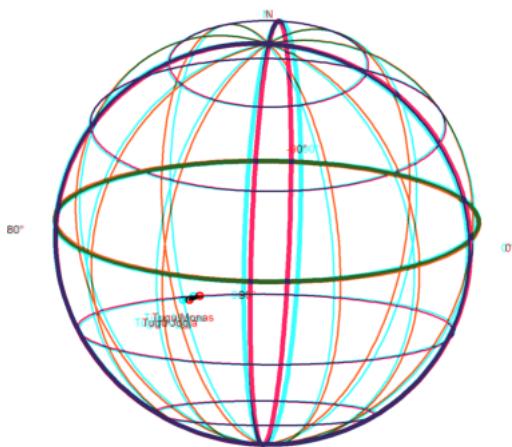
>plot3d("testplot",angle=25, height=6,own>user,zoom=4):



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-135.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):



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-136.png

### Latihan

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah  $(360/n)$ .
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n
- dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar
- kelipatan  $(360/n)$ .
- Untuk  $n$  ganjil, pilih salah satu titik sudut adalah di atas.
- Untuk  $n$  genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat.
- Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

Penyelesaian :

>load geometry

Numerical and symbolic geometry.

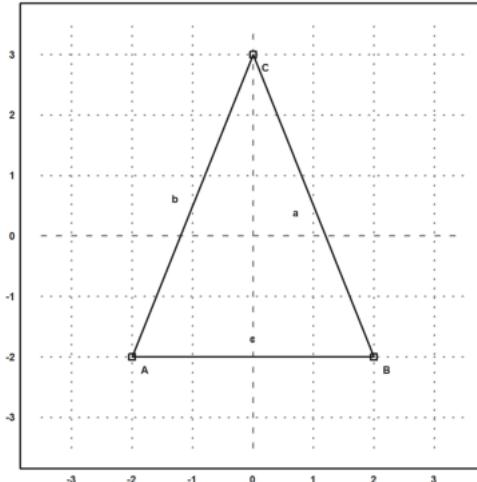
>setPlotRange(-3.5,3.5,-3.5,3.5);

>A=[-2,-2]; plotPoint(A,"A");

```

>B=[2,-2]; plotPoint(B,"B");
>C=[0,3]; plotPoint(C,"C");
>plotSegment(A,B,"c");
>plotSegment(B,C,"a");
>plotSegment(A,C,"b");
>aspect(1);

```

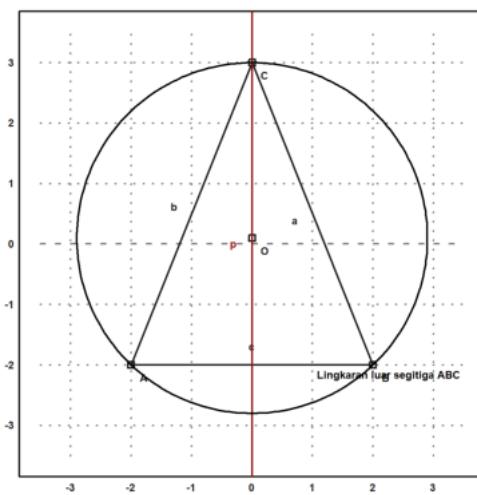


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-137.png

```

>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");

```



2. Gambarlah suatu

parabola yang melalui 3 titik yang diketahui.

Petunjuk:

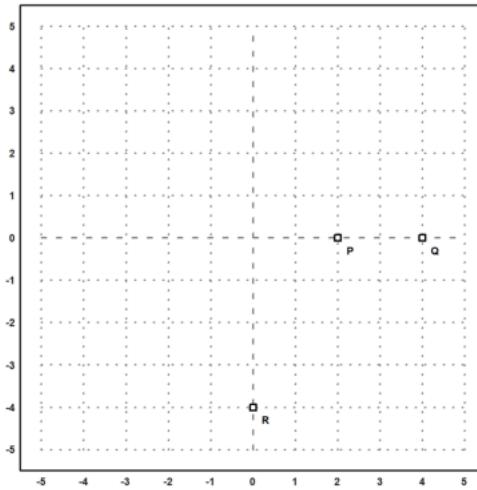
- Misalkan persamaan parabolanya  $y = ax^2 + bx + c$ .
- Subsitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

Penyelesaian :

```

>load geometry;
>setPlotRange(5); P=[2,0]; Q=[4,0]; R=[0,-4];
>plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R");

```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-139.png

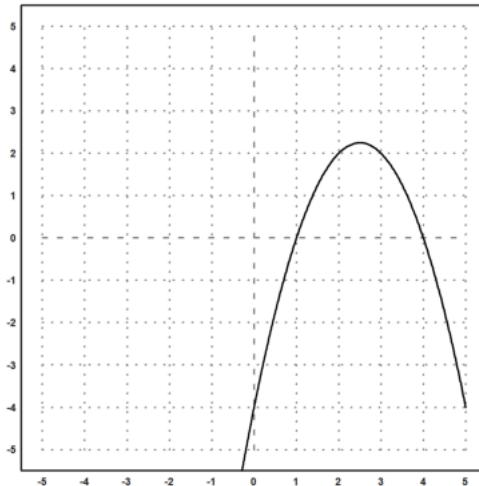
```
>sol &= solve([a+b=-c,16*a+4*b=-c,c=-4],[a,b,c])
[[a = - 1, b = 5, c = - 4]]
```

Sehingga didapat nilai  $a = -1$ ,  $b = 5$  dan  $c = -4$

```
>function y&=-x^2+5*x-4
```

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

```
>plot2d("-x^2+5*x-4",-5,5,-5,5);
```



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.

Penyelesaian :

```
>load geometry
```

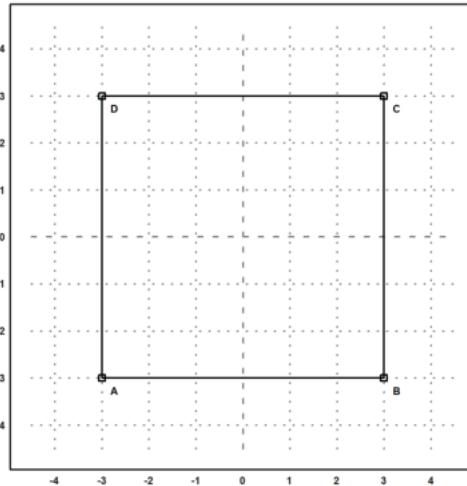
Numerical and symbolic geometry.

```
>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);

```

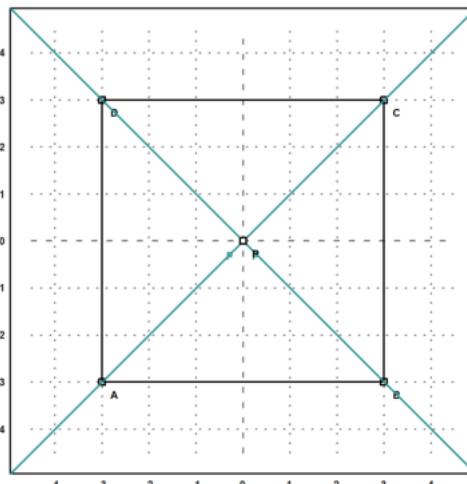


images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-141.png

```

>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");

```



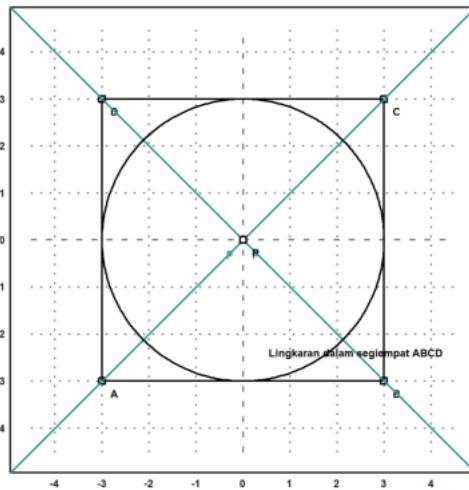
images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-142.png

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/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-143.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

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).

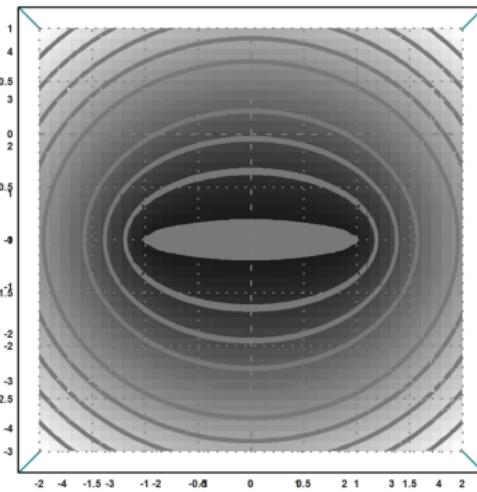
Penyelesaian :

>P=[-1,-1]; Q=[1,-1];

>function d1(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)

>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)+sqrt((x-Q[1])^2+(y-Q[2])^2)

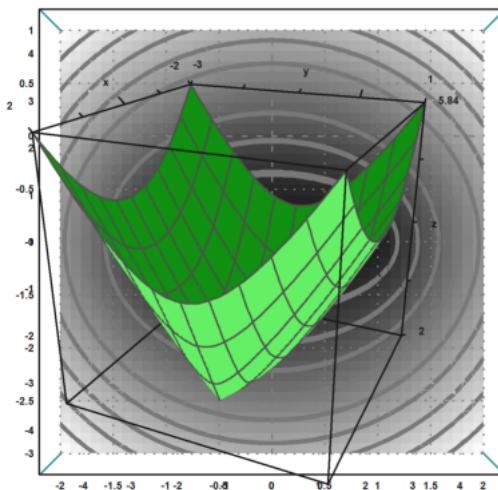
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1);



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-144.png

Grafik menjadi lebih menarik

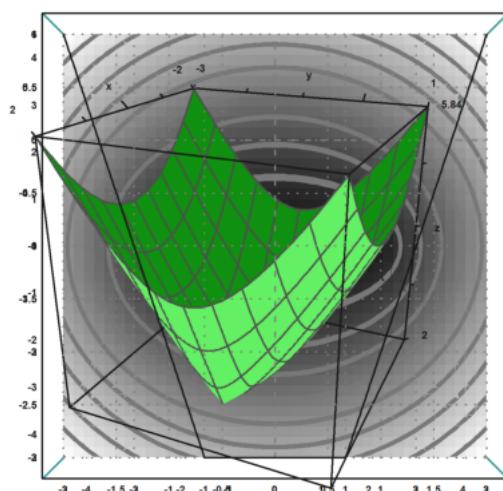
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1);



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-145.png

Batasan ke garis PQ

>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3);



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).

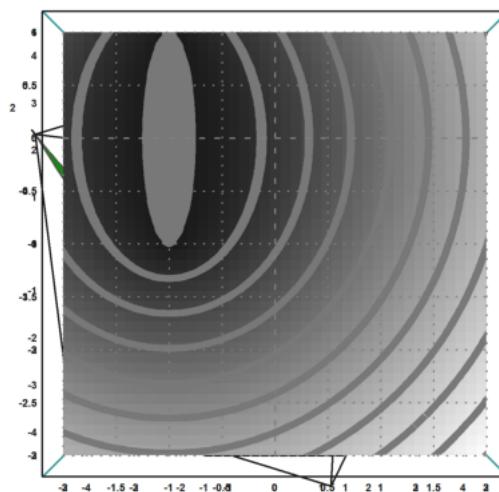
Penyelesaian :

>P=[-1,-1]; Q=[1,-1];

```

>function d1(x,y):=sqrt((x-p[1])2+(y-p[2])2)
>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])2+(y-P[2])2)+sqrt((x+Q[1])2+(y+Q[2])2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):

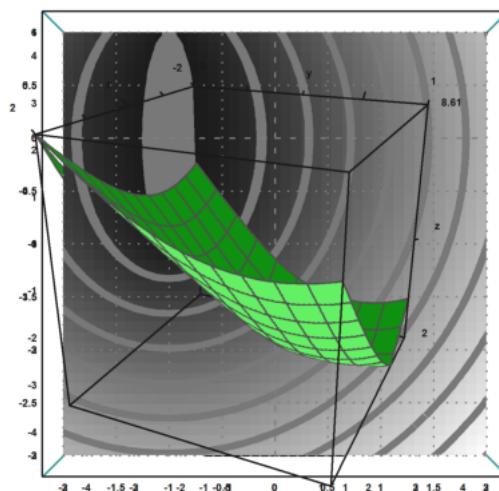
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-147.png

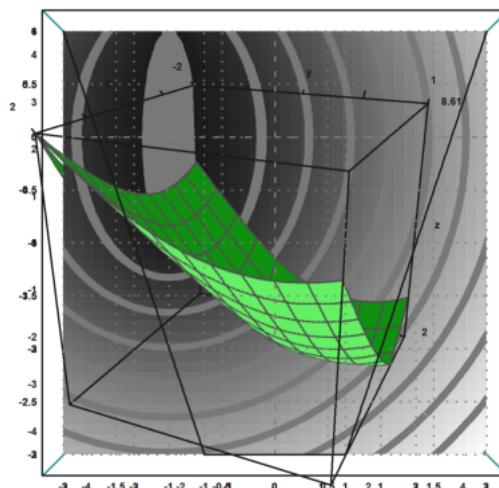
Grafik yang lebih menarik

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-148.png

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



images/23030630004\_Siti%20Faltipah%20Hayani\_EMT4Geometry-149.png

# EMT untuk Statistika

Nama : Siti Faltipah Hayani

NIM : 23030630004

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.

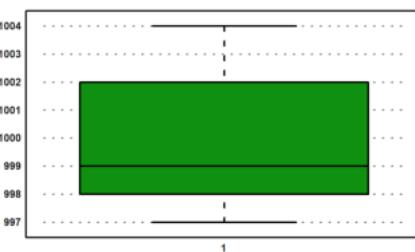
Asumsikan pengukuran berikut ini. Kita ingin menghitung nilai rata-rata dan deviasi standar yang diukur.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...
> median(M), mean(M), dev(M),
```

```
999
999.9
2.72641400622
```

Kita dapat memplot plot kotak dan kumis untuk data tersebut. Dalam kasus kami, tidak ada pencilan.

```
>aspect(1.75); boxplot(M);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-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).

Kami mencetak hasilnya dalam % dengan akurasi 2 digit menggunakan fungsi cetak.

```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
```

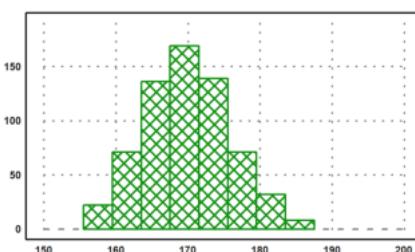
```
3.07 %
```

Untuk contoh berikutnya, kami mengasumsikan jumlah pria berikut 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/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-002.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.

Simbol “|” 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]
```

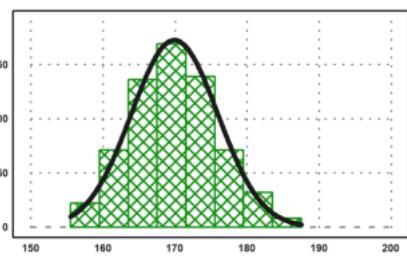
Sekarang kita dapat menghitung rata-rata dan deviasi sampel dengan frekuensi yang diberikan.

```
>{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:

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):
```



## Tables

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);
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
open(filename,"r");
```

Tabel berisi 7 kolom angka atau token (string). 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 harus menyediakannya dengan “:=”.

```
>{MT,hd}=readtable(“table.dat”,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```

Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
  if filename!=none then open(filename,"r"); endif;

>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);

MT is not a variable!
Error in:
writetable(MT[1:10],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,to ...
^
```

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);

Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
  if filename!=none then open(filename,"r"); endif;

Fungsi readtable() sekarang mengembalikan satu set token.

>tok

Variable tok not found!
Error in:
tok ...
^
```

Tabel berisi entri dari file dengan token yang diterjemahkan menjadi angka.

String khusus NA=“.” ditafsirkan sebagai “Tidak Tersedia”, dan mendapatkan NAN (bukan angka) dalam tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAvail.

```

>MT[1]

MT is not a variable!
Error in:
MT[1] ...
^
```

Berikut ini adalah isi tabel dengan angka yang tidak diterjemahkan.

```

>writetable(MT,wc=5)

Variable or function MT not found.
Error in:
writetable(MT,wc=5) ...
^
```

Untuk kenyamanan, Anda dapat menaruh keluaran dari readtable() ke dalam sebuah daftar.

```

>Table={{readtable("table.dat",ctok=ctok)}};

Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
  if filename!=none then open(filename,"r"); endif;
```

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);

Variable or function Table not found.
Error in:
writetable(Table,ctok=ctok,wc=5); ...
^
```

Fungsi tablecol() mengembalikan nilai kolom dari tabel, melewaskan setiap baris dengan nilai NAN (“.” dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

```
>{c,i}=tablecol(MT,[5,6]);
```

```
Variable or function MT not found.  
Error in:  
{c,i}=tablecol(MT,[5,6]); ...  
^
```

Kita dapat menggunakan ini untuk mengekstrak kolom dari tabel untuk tabel baru.

```
>j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok)  
  
Variable or function i not found.  
Error in:  
j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok) ...  
^
```

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 menggunakan 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',labc=tok[xu])  
  
Variable count not found!  
Error in:  
writetable('count',labc=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”])  
  
Variable or function tok not found.  
Error in:  
v:=indexof(tok,[“g”,“vg”]) ...  
^
```

Sekarang kita dapat memilih baris-baris dari tabel, yang memiliki salah satu nilai dalam v di baris ke-5.

```
>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);  
  
Variable or function i not found.  
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)

Could not open the file
test.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
  if filename!=none then open(filename,"r"); endif;

```

Dan hapus file tersebut.

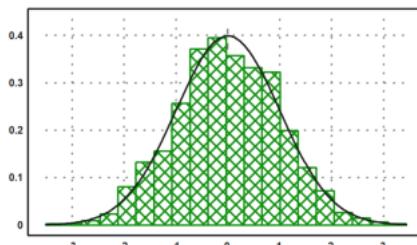
```
>fileremove(filename); ## Distribusi
```

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

```

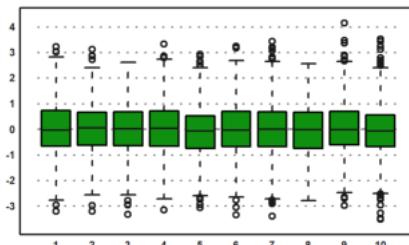


images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-004.png

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);
```

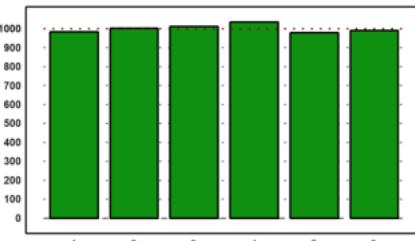


images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-005.png

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 `columnsplot()`.

```
>k=intrandom(1,6000,6); ...
>columnsplot(getmultiplicities(1:6,k)); ...
>ygrid(1000,color=red);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-006.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()`.

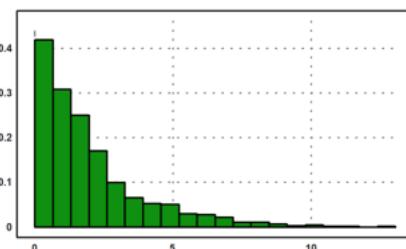
Pada contoh berikut, probabilitas untuk 1,2,3 adalah 0.4, 0.1, 0.5 secara berurutan.

```
>randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,%)
[378, 102, 520]
```

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 with parameter

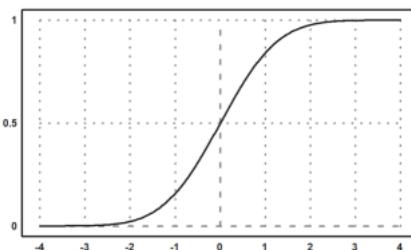
```
>plot2d(randexponential(1,1000,2),>distribution):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-007.png

Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan kebalikannya.

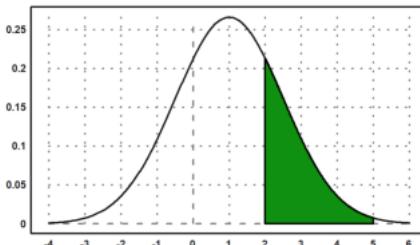
```
>plot2d("normaldis",-4,4):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-008.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):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-009.png

Probabilitas untuk berada di area hijau adalah sebagai berikut.

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

0.248662156979

Hal ini dapat dihitung secara numerik dengan integral berikut ini.

```
>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.

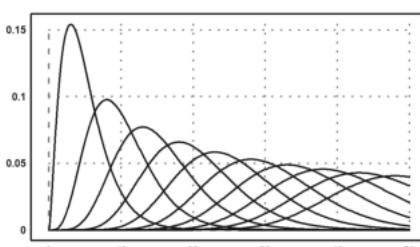
```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

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):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-010.png

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]

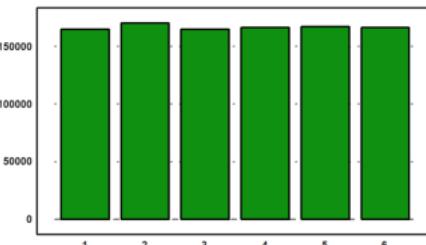
Artinya, dengan probabilitas wd[i+1]-wd[i] kita menghasilkan nilai acak i.

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/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-011.png

Berikut ini adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai 1...K dalam v. Kami menerima hasilnya, jika untuk semua frekuensi

```
>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 built-in.

```
>checkrandom(intrandom(1,1000000,6))
```

1

Kita dapat menghitung distribusi binomial. Pertama, ada binomials() yang mengembalikan probabilitas i atau kurang dari n percobaan.

```
>bindis(410,1000,0.4)
```

0.751401349654

Fungsi Beta invers digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p. Tingkat defaultnya adalah alpha.

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)*p^i*(1-p)^(n-i))
```

0.751401349655

Omong-omong, invbinsum() menghitung kebalikan dari binomials().

```
>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])
```

|     |     |     |
|-----|-----|-----|
| 381 | 100 | 519 |
| 376 | 91  | 533 |
| 417 | 80  | 503 |
| 440 | 94  | 466 |
| 406 | 112 | 482 |
| 408 | 94  | 498 |
| 395 | 107 | 498 |
| 399 | 96  | 505 |
| 428 | 87  | 485 |
| 400 | 99  | 501 |

## 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-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 |

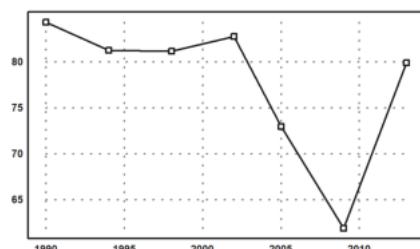
Perkalian matriks berikut ini mengekstrak jumlah persentase dua partai besar yang menunjukkan bahwa partai-partai kecil telah memperoleh suara di parlemen hingga tahun 2009.

```
>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 menggunakanannya untuk menampilkan garis dan titik secara bersamaan. Alternatif lainnya adalah memanggil `plot2d` dua kali dengan `>add`.

```
>statplot(YT,BT1,“b”):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-012.png

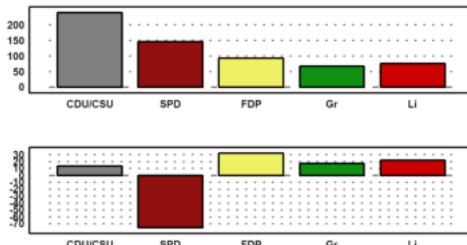
Tentukan beberapa warna untuk masing-masing pihak.

```
>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); ...
```

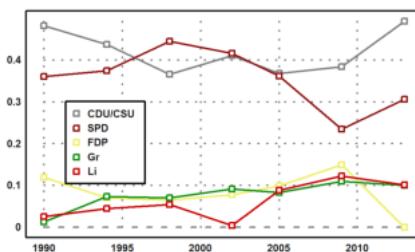
> figure(0):



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-013.png

Plot data menggabungkan baris data statistik dalam satu plot.

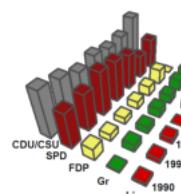
```
>J:=BW[,1]'; DP:=BW[,3:7]'; ...
> dataplot(YT,BT',color=CP); ...
> labelbox(P,colors=CP,styles="[]");>points,w=0.2,x=0.3,y=0.4);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-014.png

Plot kolom 3D menunjukkan deretan data statistik dalam bentuk kolom. Kami menyediakan label untuk baris dan kolom. angle adalah sudut pandang.

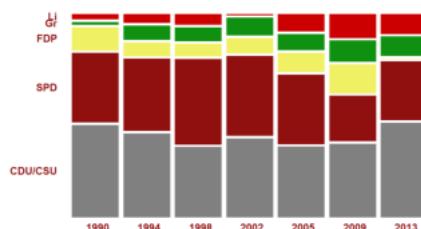
```
>columnsplot3d(BT,scols=P,srows=YT, ...
> angle=30°,ccols=CP);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-015.png

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.

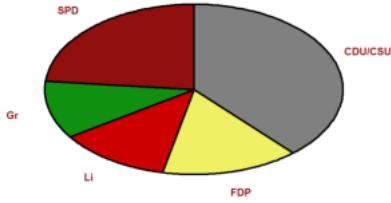
```
>shrinkwindow(>smaller); ...
> mosaicplot(BT',srows=YT,scols=P,color=CP,style="#"); ...
> shrinkwindow();
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-016.png

Kita juga bisa membuat diagram lingkaran. Karena warna hitam dan kuning membentuk koalisi, kita menyusun ulang elemen-elemennya.

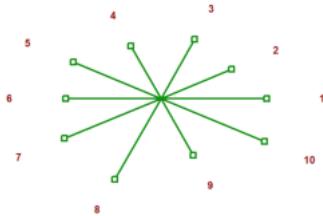
```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-017.png

Berikut ini jenis plot yang lain.

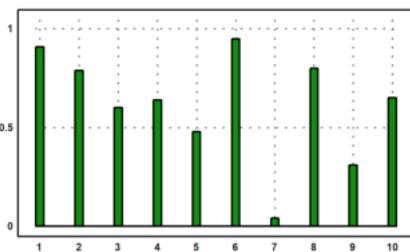
>starplot(normal(1,10)+4,lab=1:10,>rays):



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-018.png

Beberapa plot dalam plot2d bagus untuk statika. Berikut ini adalah plot impuls dari data acak, yang terdistribusi secara seragam dalam  $[0,1]$ .

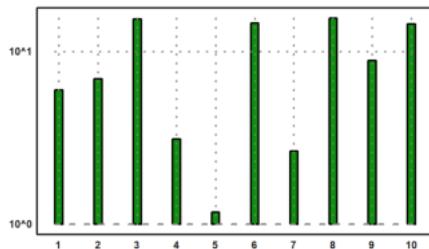
>plot2d(makeimpulse(1:10,random(1,10)),>bar):



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-019.png

Tetapi untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

>logimpulseplot(1:10,-log(random(1,10))\*10):

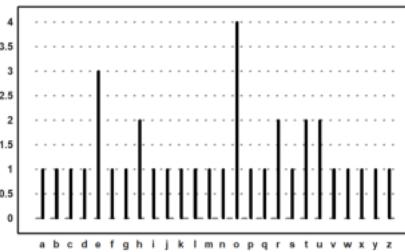


images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-020.png

Fungsi columnsplot() 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.

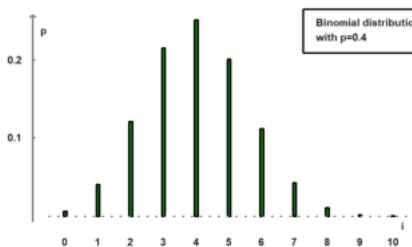
```
>v=strtochar("the quick brown fox jumps over the lazy dog"); ...
>w=ascii("a"):ascii("z"); x=getmultiplicities(w,v); ...
>cw=[]; for k=w; cw=cw|char(k); end; ...
>columnsplot(x,lab=cw,width=0.05):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-021.png

Anda juga dapat menetapkan sumbu secara manual.

```
>n=10; p=0.4; i=0:n; x=bin(n,i)*p^i*(1-p)^(n-i); ...
>columnsplot(x,lab=i,width=0.05,<frame,<grid); ...
>yaxis(0.0:0.1:1,style=">>">left); xaxis(0,style="."); ...
>label("p",0,0.25), label("i",11,0); ...
>textbox(["Binomial distribution","with p=0.4"]);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-022.png

Berikut ini adalah cara untuk memplot frekuensi angka dalam vektor. Kami membuat vektor angka acak bilangan bulat 1 hingga 6.

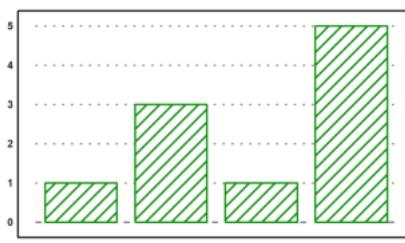
```
>v:=intrandom(1,10,10)
[8, 5, 8, 8, 6, 8, 8, 3, 5, 5]
```

Kemudian ekstrak nomor unik dalam v.

```
>vu:=unique(v)
[3, 5, 6, 8]
```

Dan memplot frekuensi dalam plot kolom.

```
>columnsplot(getmultiplicities(vu,v),lab=vu,style="/");
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-023.png

Kami ingin mendemonstrasikan fungsi untuk distribusi nilai empiris.

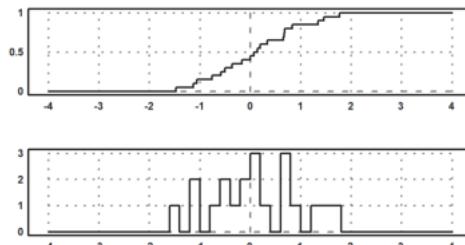
```
>x=normal(1,20);
```

Fungsi empdist(x,vs) membutuhkan larik nilai yang telah diurutkan. Jadi kita harus mengurutkan x sebelum dapat menggunakananya.

```
>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.

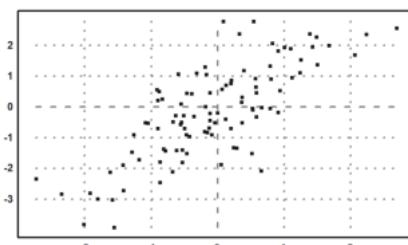
```
>figure(2,1); ...
>figure(1); plot2d("empdist",-4,4;xs); ...
>figure(2); plot2d(histo(x,v=-4:0.2:4,<bar)); ...
>figure(0);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-024.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="..");
```

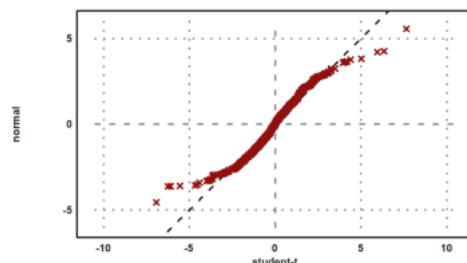


images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-025.png

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",t=6,style="--",yl="normal",xl="student-t",>vertical); ...
> plot2d(sort(x),sort(y),>points,color=red,style="x",>add);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-026.png

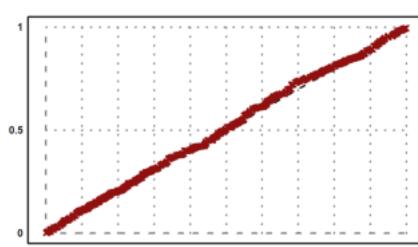
Plot tersebut dengan jelas menunjukkan bahwa nilai yang terdistribusi normal cenderung lebih kecil pada ujung yang ekstrim.

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/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-027.png

## Regresi dan Korelasi

Regresi linier dapat dilakukan dengan fungsi polyfit() atau berbagai fungsi kecocokan.

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.

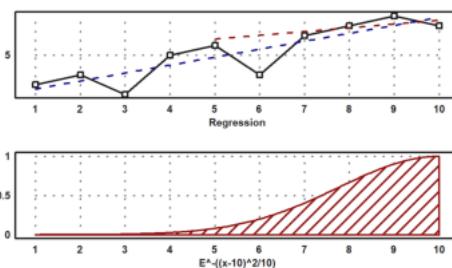
```
>p=polyfit(x,y,1)
[0.733333, 0.812121]
```

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/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-028.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”]);

Could not open the file
table1.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
if filename!=none then open(filename,"r"); endif;
```

Bagaimana usia saling bergantung satu sama lain? Kesan pertama datang dari scatterplot berpasangan.

```
>scatterplots(tablecol(MS,3:5),hd[3:5]):
```

Variable or function MS not found.  
Error in:  
scatterplots(tablecol(MS,3:5),hd[3:5]): ...  
^

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)
```

```

MS is not a variable!
Error in:
cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1) ...
^

```

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, kita 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])

cs is not a variable!
Error in:
da:=mean(cs[2]-cs[1]) ...
^

```

Mari kita plotkan ini ke dalam satu scatter plot.

```

>plot2d(cs[1],cs[2],>points); ...
> plot2d("evalpoly(x,ps)",color=red,style=".",>add); ...
> plot2d("x+da",color=blue,>add);

cs is not a variable!
Error in:
plot2d(cs[1],cs[2],&gt;points); plot2d("evalpoly(x,ps)",color=re ...
^

```

Berikut ini adalah plot kotak dari kedua usia tersebut. Ini hanya menunjukkan, bahwa usia keduanya berbeda.

```

>boxplot(cs,[“mothers”,“fathers”]);

Variable or function cs not found.
Error in:
boxplot(cs,[“mothers”,“fathers”]): ...
^

```

Sangat menarik bahwa perbedaan dalam median tidak sebesar perbedaan dalam mean.

```

>median(cs[2])-median(cs[1])

cs is not a variable!
Error in:
median(cs[2])-median(cs[1]) ...
^

```

Koefisien korelasi menunjukkan korelasi positif.

```

>correl(cs[1],cs[2])

cs is not a variable!
Error in:
correl(cs[1],cs[2]) ...
^

```

Korelasi peringkat adalah ukuran untuk urutan yang sama dalam kedua vektor. Korelasi ini juga cukup positif.

```

>rankcorrel(cs[1],cs[2])

cs is not a variable!
Error in:
rankcorrel(cs[1],cs[2]) ...
^

```

## 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.

```

>function skew (x:vector) ...

m=mean(x);
return sqrt(cols(x)*sum((x-m)^3)/(sum((x-m)^2))^(3/2));
endfunction

```

Seperti yang Anda lihat, kita dapat dengan mudah menggunakan bahasa matriks untuk mendapatkan implementasi yang sangat singkat dan efisien. Mari kita coba fungsi ini.

```

>data=normal(20); skew(normal(10))

-0.198710316203

```

Berikut ini adalah fungsi lain, yang disebut koefisien kemencengan Pearson.

```
>function skew1(x) := 3*(mean(x)-median(x))/dev(x)
>skew1(data)
-0.0801873249135
```

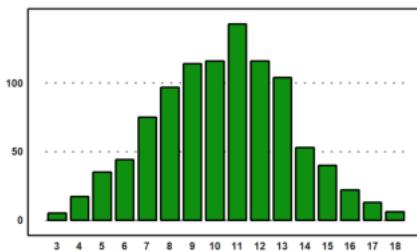
## Simulasi Monte Carlo

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.

```
>ds:=sum(intrandom(1000,3,6)); fs=getmultiplicities(3:18,ds)
[5, 17, 35, 44, 75, 97, 114, 116, 143, 116, 104, 53, 40,
22, 13, 6]
```

Kita bisa merencanakan ini sekarang.

```
>columnsplot(fs,lab=3:18);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-029.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.

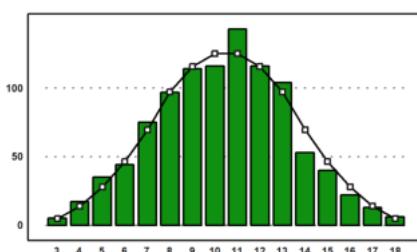
```
>function map countways (k; n, m) ...
  if n==1 then return k>=1 && k<=m
  else
    sum=0;
    loop 1 to m; sum=sum+countways(k-#,n-1,m); end;
    return sum;
  end;
endfunction
```

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,
```

Kami menambahkan nilai yang diharapkan ke plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-030.png

Untuk simulasi lainnya, deviasi nilai rata-rata dari n variabel acak berdistribusi normal 0-1 adalah  $1/\sqrt{n}$ .

```
>longformat; 1/sqrt(10)
```

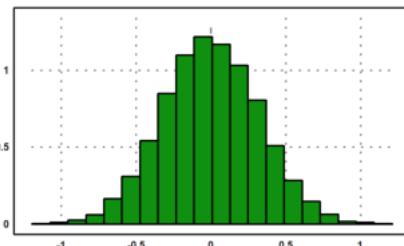
0.316227766017

Mari kita periksa hal ini dengan sebuah simulasi. Kami menghasilkan 10.000 kali 10 vektor acak.

```
>M=normal(10000,10); dev(mean(M))'
```

```
0.319493614817
```

```
>plot2d(mean(M)',>distribution):
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-031.png

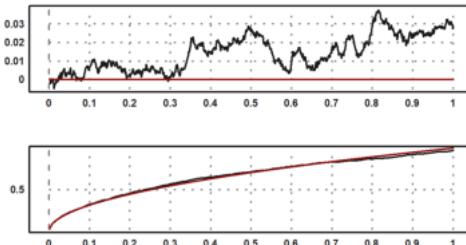
Median dari 10 bilangan acak berdistribusi normal 0-1 memiliki deviasi yang lebih besar.

```
>dev(median(M))'
```

```
0.374460271535
```

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/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-032.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)
```

```
0.526
```

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.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

```
0.528028118442
```

Mari kita uji nilai rata-rata 100 dengan uji-t.

```
>s=200+normal([1,100])*10; ...
>ttest(mean(s),dev(s),100,200)
```

0.0218365848476

Fungsi ttest() membutuhkan nilai rata-rata, deviasi, jumlah data, dan nilai rata-rata untuk diujii.

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.

>tcomparedata(normal(1,10),normal(1,10))

0.38722000942

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)

5.60009101758e-07

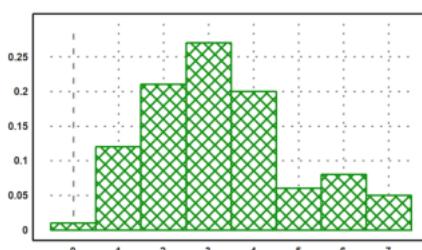
Pada contoh berikut, kita membuat 20 lemparan dadu secara acak sebanyak 100 kali dan menghitung jumlah dadu yang muncul. Rata-rata harus ada  $20/6 = 3,3$  mata dadu.

>R=random(100,20); R=sum(R\*6<=1); mean(R)

3.28

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/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-033.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.

>t1=sum(t[1:2])|t[3:7]|sum(t[8:21]); ...  
> b1=sum(b[1:2])|b[3:7]|sum(b[8:21]);

Uji chi-square menolak hipotesis bahwa distribusi kita adalah distribusi binomial, jika hasilnya <0,05.

>chitest(t1,b1)

0.53921579764

Contoh berikut ini berisi hasil dari dua kelompok orang (laki-laki dan perempuan, katakanlah) yang memberikan suara untuk satu dari enam partai.

>A=[23,37,43,52,64,74;27,39,41,49,63,76]; ...  
> writetable(A,wc=6,labr=[“m”,“f”],labc=1:6)

|   | 1  | 2  | 3  | 4  | 5  | 6  |
|---|----|----|----|----|----|----|
| m | 23 | 37 | 43 | 52 | 64 | 74 |
| f | 27 | 39 | 41 | 49 | 63 | 76 |

Kami ingin menguji independensi suara dari jenis kelamin. Uji tabel chi^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),
```

```
106.545454545
```

```
>x2=[120,124,115,139,114,110,113,120,117]; mean(x2),
```

```
119.111111111
```

```
>x3=[120,112,115,110,105,134,105,130,121,111]; mean(x3)
```

```
116.3
```

```
>varanalysis(x1,x2,x3)
```

```
0.0138048221371
```

Ini berarti, kami menolak hipotesis 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)
```

```
0.129608141484
```

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)
```

```
0.0068706451766
```

```
>>wilcoxon(normal(1,20),normal(1,20))
```

## 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].

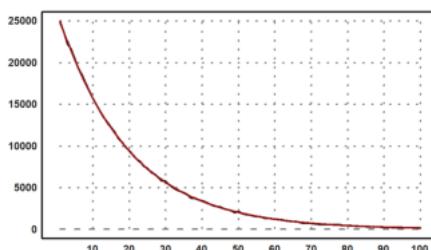
```
>n:=10000000; r:=random(1,n);
```

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^2",color=red,>add);
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-034.png

Menghapus 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 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 "Interoduksi 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 menyesatkan, 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, u“...” dapat berisi entitas HTML.

```
>u"© René 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])
```

```
Ā
```

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 dalam 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.

```
>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;
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 berbagai hal 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.

```
>function tapply (t:vector,tf,f$call) ...
## Makes a table of data and factors
## tf : output of makef()
## See: makef
uf=tf[2]; f=tf[3]; x=zeros(length(uf));
for i=1 to length(uf);
```

```

ind=nonzeros(f==i);
if length(ind)==0 then x[i]=NAN;
else x[i]=f$(t[ind]);
endif;
end;
return {{x,uf}};
endfunction

```

Kami tidak menambahkan banyak 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)
```

| act  | nsw   | nt   | qld  | sa | tas  | vic | wa    |
|------|-------|------|------|----|------|-----|-------|
| 44.5 | 57.33 | 55.5 | 53.6 | 55 | 60.5 | 56  | 52.25 |

## Arrays

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Akan lebih mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau 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 ini untuk menunjukkan bahwa matriks dilewatkan dengan referensi di EMT. Jika Anda tidak ingin mengubah matriks asli M, Anda perlu menyalininya dalam fungsi.

```
>setmatrixvalue(X,1:3,3:-1:1,0); X,
```

| 1  | 2  | 0  | 4  | 5  |
|----|----|----|----|----|
| 6  | 0  | 8  | 9  | 10 |
| 0  | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |

Produk luar di EMT hanya dapat dilakukan antar vektor. Ini otomatis karena bahasa matriks. Satu vektor harus menjadi vektor kolom dan yang lainnya vektor baris.

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

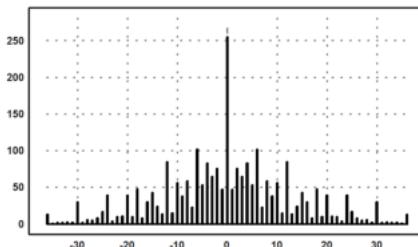
| 1 | 2  | 3  | 4  | 5  |
|---|----|----|----|----|
| 2 | 4  | 6  | 8  | 10 |
| 3 | 6  | 9  | 12 | 15 |
| 4 | 8  | 12 | 16 | 20 |
| 5 | 10 | 15 | 20 | 25 |

Dalam PDF pengantar untuk R ada contoh, yang menghitung distribusi ab-cd untuk a, b, c, d yang dipilih dari 0 ke n secara acak. Solusi dalam R adalah membentuk matriks 4 dimensi dan jalankan table() di atasnya.

Tentu saja, ini dapat dicapai dengan loop. Tetapi loop tidak efektif di EMT atau R. Di EMT, kita bisa menulis loop dalam C dan itu akan menjadi solusi tercepat.

Tapi kami ingin meniru perilaku R. Untuk ini, kita perlu meratakan perkalian ab dan membuat matriks ab-cd.

```
>a=0:6; b=a'; p=flatten(a*b); q=flatten(p-p'); ...
> u=sort(unique(q)); f=getmultiplicities(u,q); ...
> statplot(u,f,"h");
```



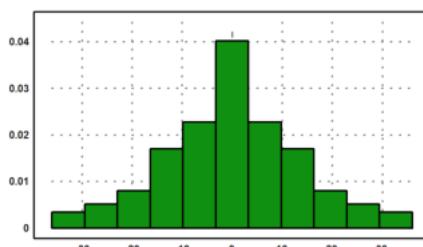
images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-035.png

Selain multiplisitas yang tepat, EMT dapat menghitung frekuensi dalam vektor.

```
>getfrequencies(q,-50:10:50)
[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]
```

Cara paling mudah untuk memplot ini sebagai distribusi adalah sebagai berikut.

```
>plot2d(q,distribution=11);
```

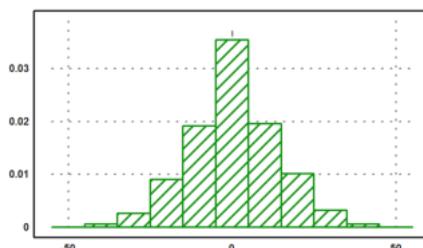


images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-036.png

Tetapi dimungkinkan juga untuk menghitung hitungan terlebih dahulu dalam interval yang dipilih sebelumnya. Tentu saja, berikut ini menggunakan getfrequency() secara internal.

Karena fungsi histo() mengembalikan frekuensi, kita perlu menskalakan ini 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/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-037.png

## Lists

EMT memiliki dua jenis daftar. Salah satunya 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 di EMT. Ini 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 elemen-elemen tersebut tidak memiliki nama, meskipun nama dapat diatur untuk tujuan khusus. Mereka diakses dengan angka.

```
>(L[4])[2]
```

## Input dan Output File (Membaca dan Menulis Data)

Anda akan sering ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberi tahu Anda tentang banyak cara untuk mencapai ini. Fungsi sederhana adalah writematrix() dan readmatrix().

Mari kita tunjukkan cara membaca dan menulis vektor real ke file.

```
>a=random(1,100); mean(a), dev(a),
```

```
0.49815  
0.28037
```

Untuk menulis data ke file, kita menggunakan fungsi writematrix(). Karena pengantar ini kemungkinan besar berada di direktori, di mana pengguna tidak memiliki akses tulis, kami menulis data ke direktori beranda pengguna. Untuk buku catatan sendiri, ini tidak perlu, karena file data akan ditulis ke dalam direktori yang sama.

```
>filename="test.dat";
```

Sekarang kita tulis vektor kolom a' ke file. Ini menghasilkan satu angka di setiap baris file.

```
>writematrix(a',filename);
```

Untuk membaca data, kita menggunakan readmatrix().

```
>a=readmatrix(filename');
```

Dan hapus file.

```
>fileremove(filename);
```

```
>mean(a), dev(a),
```

```
0.49815  
0.28037
```

Fungsi writematrix() atau writetable() dapat dikonfigurasi untuk bahasa lain.

Misalnya, jika Anda memiliki sistem bahasa Indonesia (titik desimal dengan koma), Excel Anda membutuhkan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan koma). File berikut "test.csv" akan muncul di folder cuurent Anda.

```
>filename="test.csv"; ...  
> writematrix(random(5,3),file=filename,separator=",");
```

Anda sekarang dapat membuka file ini dengan Excel Indonesia secara langsung.

```
>fileremove(filename);
```

Terkadang kita memiliki string dengan token seperti berikut.

```
>s1:="f m m f m m f f f m m f"; ...  
> s2:="f f f m m f f";
```

Untuk mendefinisikan token, kita mendefinisikan vektor token.

```
>tok:=[“f”,“m”]
```

```
f  
m
```

Kemudian kita dapat menghitung berapa kali setiap token muncul dalam string, dan memasukkan hasilnya ke dalam tabel.

```
>M:=getmultiplicities(tok,strtokens(s1))_ ...  
> getmultiplicities(tok,strtokens(s2));
```

Tulis tabel dengan header token.

```
>writetable(M,labc=tok,labr=1:2,wc=8)
```

|   | f | m |
|---|---|---|
| 1 | 6 | 7 |
| 2 | 5 | 2 |

Untuk statis, EMT dapat membaca dan menulis tabel.

```
>file="test.dat"; open(file,"w"); ...  
> writeln("A,B,C"); writematrix(random(3,3)); ...  
> close();
```

File terlihat seperti ini.

```
>printfile(file)
```

```
A,B,C  
0.7003664386138074,0.1875530821001213,0.3262339279660414
```

```
0.5926249243193858,0.1522927283984059,0.368140583062521  
0.8065535209872989,0.7265910840408142,0.7332619844597152
```

Fungsi readtable() dalam bentuknya yang paling sederhana dapat membaca ini dan mengembalikan kumpulan nilai dan baris judul.

```
>L=readtable(file,>list);
```

Koleksi ini dapat dicetak dengan writetable() ke buku catatan, atau ke file.

```
>writetable(L,wc=10,dc=5)
```

| A       | B       | C       |
|---------|---------|---------|
| 0.70037 | 0.18755 | 0.32623 |
| 0.59262 | 0.15229 | 0.36814 |
| 0.80655 | 0.72659 | 0.73326 |

Matriks nilai adalah elemen pertama dari L. Perhatikan bahwa mean() di EMT menghitung nilai rata-rata baris matriks.

```
>mean(L[1])
```

|         |
|---------|
| 0.40472 |
| 0.37102 |
| 0.75547 |

## File CSV

Pertama, mari kita tulis matriks ke dalam file. Untuk output, kami membuat file di direktori kerja saat ini.

```
>file="test.csv"; ...  
> M=random(3,3); writematrix(M,file);
```

Berikut adalah isi file ini.

```
>printfile(file)
```

|  |
|--|
| 0.8221197733097619,0.821531098722547,0.7771240608094004  |
| 0.8482947121863489,0.3237767724883862,0.6501422353377985 |
| 0.1482301827518109,0.3297459716109594,0.6261901074210923 |

CSV ini dapat dibuka pada sistem bahasa Inggris ke Excel dengan mengklik dua kali. Jika Anda mendapatkan file seperti itu pada sistem Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal.

Tetapi titik desimal adalah format default untuk EMT juga. Anda dapat membaca matriks dari file dengan readmatrix().

```
>readmatrix(file)
```

|         |         |         |
|---------|---------|---------|
| 0.82212 | 0.82153 | 0.77712 |
| 0.84829 | 0.32378 | 0.65014 |
| 0.14823 | 0.32975 | 0.62619 |

Dimungkinkan untuk menulis beberapa matriks ke satu file. Perintah open() dapat membuka file untuk ditulis dengan parameter "w". Defaultnya adalah "r" untuk membaca.

```
>open(file,"w"); writematrix(M); writematrix(M'); close();
```

Matriks dipisahkan oleh garis kosong. Untuk membaca matriks, buka file dan panggil readmatrix() beberapa kali.

```
>open(file); A=readmatrix(); B=readmatrix(); A==B, close();
```

|   |   |   |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

Di Excel atau spreadsheet serupa, Anda dapat mengekspor matriks sebagai CSV (nilai yang dipisahkan koma). Di Excel 2007, gunakan "simpan sebagai" dan "format lain", lalu pilih "CSV". Pastikan, tabel saat ini hanya berisi data yang ingin Anda ekspor.

Berikut contohnya.

```
>printfile("excel-data.csv")
```

```
Could not open the file  
excel-data.csv  
for reading!  
Try "trace errors" to inspect local variables after errors.  
printfile:  
    open(filename,"r");
```

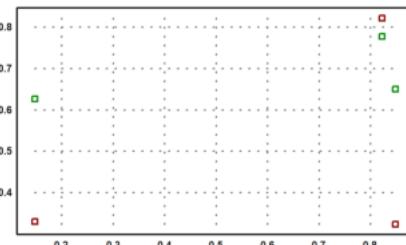
Seperti yang Anda lihat, sistem Jerman saya telah menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubahnya di pengaturan sistem atau di Excel, tetapi tidak perlu untuk membaca matriks ke EMT.

Cara termudah untuk membacanya ke dalam Euler adalah readmatrix(). Semua koma diganti dengan titik-titik dengan parameter >koma. Untuk CSV bahasa Inggris, cukup hilangkan parameter ini.

```
>M=readmatrix("excel-data.csv",>comma)
Could not open the file
excel-data.csv
for reading!
Try "trace errors" to inspect local variables after errors.
readmatrix:
if filename<>""
then open(filename,"r");
endif;
```

Mari kita rencanakan ini.

```
>plot2d(M'[1],M'[2:3],>points,color=[red,green]');
```



images/Siti%20Faltipah%20Hayani\_23030630004\_EMT4Statistika-038.png

Ada cara yang lebih mendasar untuk membaca data dari file. Anda dapat membuka file dan membaca angka baris demi baris. Fungsi getvectorline() akan membaca angka dari baris data. Secara default, ia mengharapkan titik desimal. Tetapi itu juga dapat menggunakan koma desimal, jika Anda memanggil setdecimaldot(“,”) sebelum Anda menggunakan fungsi ini.

Fungsi berikut adalah contoh untuk ini. Ini akan berhenti di akhir file atau baris kosong.

```
>function myload (file) ...
open(file);
M=[];
repeat
    until eof();
    v=getvectorline(3);
    if length(v)>0 then M=M_v; else break; endif;
end;
return M;
close(file);
endfunction

>myload(file)

0.82212      0   0.82153      0   0.77712
0.84829      0   0.32378      0   0.65014
0.14823      0   0.32975      0   0.62619
```

Dimungkinkan juga untuk membaca semua angka dalam berkas itu dengan getvector().

```
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)

0.82212      0   0.82153
0   0.77712   0.84829
0   0.32378      0
```

Dengan demikian sangat mudah untuk menyimpan vektor nilai, satu nilai di setiap baris dan membaca kembali vektor ini.

```
>v=random(1000); mean(v)

0.50303

>writematrix(v,file); mean(readmatrix(file)')

0.50303
```

## Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Sebagai contoh, kita menulis tabel dengan header baris dan kolom ke file.

```
>file="test.tab"; M=random(3,3); ...
>open(file,"w"); ...
>writetable(M,separator=",",labc=["one","two","three"]); ...
>close(); ...
>printfile(file)
```

```

one,two,three
0.09,      0.39,      0.86
0.39,      0.86,      0.71
0.2,       0.02,      0.83

```

Ini dapat diimpor ke Excel.

Untuk membaca file di EMT, kita menggunakan readtable().

```

>{M,headings}=readtable(file,>clabs); ...
> writetable(M,labc=headings)

```

```

one      two      three
0.09    0.39    0.86
0.39    0.86    0.71
0.2     0.02    0.83

```

## Menganalisis Garis

Anda bahkan dapat mengevaluasi setiap baris dengan tangan. Misalkan, kita memiliki baris dengan format berikut.

```
>line="2020-11-03,Tue,1'114.05"
```

```
2020-11-03,Tue,1'114.05
```

Pertama kita dapat menandai garis.

```
>vt=strtoks(line)
```

```

2020-11-03
Tue
1'114.05

```

Kemudian kita dapat mengevaluasi setiap elemen garis menggunakan evaluasi yang sesuai.

```

>day(vt[1]), ...
> indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])), ...
> strrep(vt[3], "", ",")()

```

```

7.3816e+05
2
1114

```

Dengan menggunakan ekspresi reguler, dimungkinkan untuk mengekstrak hampir semua informasi dari baris data.

Asumsikan kita memiliki baris berikut 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&
```

Untuk mengekstrak ini, kami menggunakan ekspresi reguler, yang mencari

- tanda kurung penutup >,
- string apa pun yang tidak mengandung tanda kurung dengan sub-kecocokan “(...)”,
- braket pembuka dan penutup menggunakan solusi terpendek,
- sekali lagi string apa pun yang tidak mengandung tanda kurung,
- dan braket pembuka <.

Ekspresi reguler agak sulit dipelajari tetapi sangat kuat.

```
>{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;
```

```

1145.5
5.6

```

Berikut adalah fungsi, yang membaca semua item numerik antara <td> dan . </td>

```
>function readtd (line) ...
```

```

v=[]; cp=0;
repeat
    {pos,s,vt}=strxfind(line,"<td.*?>(.+?)</td>",cp);
    until pos==0;
    if length(vt)>0 then v=v|vt[1]; endif;
    cp=cp+strlen(s);
end;
return v;
endfunction

```

```

>readtd(line+“<td>non-numerical</td>”)

1145.45
5.6
-4.5
non-numerical

```

## Membaca dari Web

Situs web atau file dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris.

Dalam contoh tersebut, kami membaca versi saat ini 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 file atau ke baris perintah.

```

>writevar(pi,“mypi”);

mypi = 3.141592653589793;

```

Untuk pengujian, kami menghasilkan file Euler di direktori kerja EMT.

```

>file=“test.e”; ...
> writevar(random(2,2),“M”,file); ...
> printfile(file,3)

M = [ ..
0.5991820585590205, 0.7960280262224293;
0.5167243983231363, 0.2996684599070898];

```

Sekarang kita dapat memuat file. Ini akan mendefinisikan matriks M.

```
>load(file); show M,
```

```

M =
0.59918   0.79603
0.51672   0.29967

```

Ngomong-ngomong, jika writevar() digunakan pada variabel, itu akan mencetak definisi variabel dengan nama variabel ini.

```

>writevar(M); writevar(inch$)

M = [ ..
0.5991820585590205, 0.7960280262224293;
0.5167243983231363, 0.2996684599070898];
inch$ = 0.0254;

```

Kita juga dapat membuka file baru atau menambahkan ke file yang sudah ada. Dalam contoh ini kita menambahkan ke file yang dibuat sebelumnya.

```

>open(file,“a”); ...
> writevar(random(2,2),“M1”); ...
> writevar(random(3,1),“M2”); ...
> close();

>load(file); show M1; show M2;

M1 =
0.30287   0.15372
0.7504   0.75401
M2 =
0.27213
0.053211
0.70249

```

Untuk menghapus file apa pun, gunakan fileremove().

```
>fileremove(file);
```

Vektor baris dalam file tidak memerlukan koma, jika setiap angka berada di baris baru. Mari kita buat file seperti itu, menulis setiap baris satu per satu dengan writeln().

```
>open(file,“w”); writeln(“M = [”); ...
> for i=1 to 5; writeln(“”+random()); end; ...
> writeln(“];”); close(); ...
> printfile(file)

M = [
0.344851384551
0.0807510017715
0.876519562911
0.754157709472
0.688392638934
];

>load(file); M

[0.34485, 0.080751, 0.87652, 0.75416, 0.68839]
```

### Latihan

1. Misalkan anda memiliki vektor  $x = [2 \ 4 \ 6 \ 8 \ 10]$

- Buatkan vektor yang menggabungkan vektor  $x$ , angka dan vektor  $x$  lagi
- Tentukan apakah setiap elemen vektor  $x$  lebih besar dari 5 (Tarik logika untuk benar dan 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. Seorang Analisis memiliki data penjualan harian selama 5 hari  
(150,200,250,300,350) yang disimpan dalam bentuk vektor sebagai berikut :

- mean(rata-rata)

- deviasi standar

```
>penjualan=[150,200,250,300,350]

[150, 200, 250, 300, 350]

>filename=“penjualan.dat”;

>writematrix(penjualan’,filename)

>penjualan=readmatrix(filename’)

[150, 200, 250, 300, 350]

>mean(penjualan)

250

>dev(penjualan)

79.057
```

3. Buatlah fungsi yang membuka URL

“[https://en.wikipedia.org/wiki/Euler\\_\(software\)](https://en.wikipedia.org/wiki/Euler_(software))”

dan mencari kata “versi” di dalam URL tersebut, dan tampilkan hasilnya.

```
>function readversionwebsite () ...
```

```
endfunction
```

```
urlopen(...);
repeat
    ...
    ...
    ...
    ...
end;
urclose();
endfunction

>readversion...
>
```