

Alifia Rahmawati (23030630044) Matematika E

Alifia Rahmawati alifiarahmawati.2023

November 2024

1 Al Jabar

1.1 EMT untuk Perhitungan Al Jabar

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:

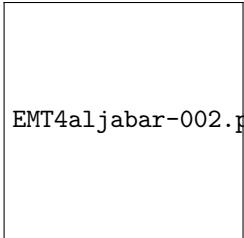
- Membaca secara cermat dan teliti notebook ini;
- Menerjemahkan teks bahasa Inggris ke bahasa Indonesia;
- Mencoba contoh-contoh perhitungan (perintah EMT) dengan cara meng-ENTER setiap perintah EMT yang ada (pindahkan kursor ke baris perintah)
- Jika perlu Anda dapat memodifikasi perintah yang ada 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 Soal

Menyederhanakan bentuk aljabar:

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

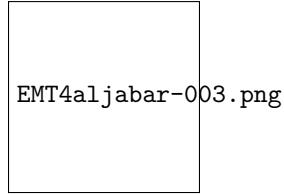
$$\$&6*x^(-3)*y^5*-7*x^2*y^(-9)$$



EMT4aljabar-002.png

Menjabarkan:

```
(6x-3 + y5)(-7x2 - y-9)  
$&showev('expand((6*x^(-3)+y^5)*(-7*x^2-y^(-9))))
```



1.2 Baris Perintah

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

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

```
r:=2; h:=4; pi*r^2*h/3  
16.7551608191
```

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

```
pi*2*r*h, %+2*pi*r*h  
50.2654824574 100.530964915
```

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

```
x := 1;  
x := cos(x)  
0.540302305868  
x := cos(x)  
0.857553215846
```

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

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

```
1.41666666667 1.41421568627 1.41421356237
```

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

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

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

Baris yang dimulai dengan

81

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

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

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

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

1.41421356237

Struktur kondisional juga berfungsi.

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

Thought so!

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

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

```

sqrt(sin(10°)/cos(20°))

0.429875017772

```

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

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

```

exp(log(2.5))

2.5

```

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

1.3 Sintaksis Dasar

uler mengetahui fungsi matematika yang umum. Seperti yang telah Anda lihat di atas, fungsi trigonometri bekerja dalam radian atau derajat. Untuk mengonversi ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilai, atau gunakan fungsi rad(x). Fungsi akar kuadrat disebut sqrt di Euler. Tentu saja, $x^{1/2}$ juga memungkinkan.

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

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

```

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

30.65625

```

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

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

```

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

8.77908249441

```

Untuk menghitung ekspresi rumit seperti
Anda perlu memasukkannya dalam bentuk baris.

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

23.2671801626

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

Hasil perhitungan ini adalah angka floating point. Secara default, angka ini dicetak dengan akurasi sekitar 12 digit. Pada baris perintah berikut, kita juga mempelajari cara merujuk ke hasil sebelumnya dalam baris yang sama.

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

0.47619047619 10/21

Perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi terdiri dari operator dan fungsi. Jika perlu, ekspresi harus berisi tanda kurung untuk memaksakan urutan eksekusi yang benar. Jika ragu, sebaiknya gunakan tanda kurung. Perhatikan bahwa EMT menampilkan tanda kurung buka dan tutup saat mengedit baris perintah.

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

14.4978445072

Operator numerik Euler meliputi

+ unary atau operator plus

- unary atau operator minus

*,/

. perkalian matriks

a^b pangkat untuk a positif atau integer b ($a^{**}b$ juga berfungsi)

n! operator faktorial

dan masih banyak lagi.

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

sin,cos,tan,atan,asin,acos,rad,deg

log,exp,log10,sqrt,logbase

sin,logbin,logfac,mod,floor,ceil,round,abs,sign

conj,re,im,arg,conj,real,complex

beta,betai,gamma,complexgamma,ellrf,ellf,ellrd,elle

bitand,bitor,bitxor,bitnot

Beberapa perintah memiliki alias, misalnya ln untuk log.

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

```

2 0.5
sin(30°)
0.5

```

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

```

$2^3^4, (2^3)^4, 2^(3^4)$$ ...
2.41785163923e+24 4096 2.41785163923e+24

```

1.4 Bilangan Rill

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

```

longest 1/3
0.3333333333333333

```

Representasi ganda internal membutuhkan 8 byte.

```

printdual(1/3)
1.01010101010101010101010101010101010101010101010101010101010101*2^-2
printhex(1/3)
5.55555555555554*16^-1

```

1.5 String

String dalam Euler didefinisikan dengan "...".

```

"A string can contain anything."
A string can contain anything.

```

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

```

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

```

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

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

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

```
none
```

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

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

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

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

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

```
w := [none]; w | v | v  
affe charlie bravo affe charlie bravo
```

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

String Unicode dapat dirangkai seperti string lainnya.

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

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

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

```

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

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

```

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

```

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

Ü is a German letter

```

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

```

s="We have &alpha;=&beta;; utf(s)

We have = .

```

Dimungkinkan juga untuk menggunakan entitas numerik.

```

u"\u00d6;hnliches"

Ähnliches

```

1.6 Nilai Boolean

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

2>1, "apel">1"banana"

0 1

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

2>E E>3

1

Operator Boolean mematuhi aturan bahasa matriks.

(1:10) 5, nonzeros(

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

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

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

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

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

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

Format Keluaran

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

```
deformat; pi  
3.14159265359
```

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

```
longest pi  
3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari angka ganda.

```
printhex(pi)  
3.243F6A8885A30*160
```

Format keluaran dapat diubah secara permanen dengan perintah format.

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

Format defaultnya adalah(12).

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

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

```
shortestformat; random(3,8)  
0.66 0.2 0.89 0.28 0.53 0.31 0.44 0.3 0.28 0.88 0.27 0.7 0.22 0.45 0.31 0.91  
0.19 0.46 0.095 0.6 0.43 0.73 0.47 0.32
```

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

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

Fungsi "longestformat" juga mengatur format skalar.

```
longestformat; pi  
3.141592653589793  
Sebagai referensi, berikut adalah daftar format output yang paling penting.  
shortestformat shortformat longformat, longestformat  
format(length,digits) goodformat(length)  
fracformat(length)  
deformat
```

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

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

```
longestformat; pi,  
3.141592653589793  
format(10,5); pi  
3.14159  
Standarnya adalah deformat().
```

```
deformat; // default
```

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

```
longest pi2/2
```

```
4.934802200544679
```

Ada juga operator pendek untuk mencetak hasil dalam format pecahan. Kami telah menggunakan di atas.

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

```
25/12
```

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

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

```
-1.110223024625157e-16
```

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

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

```
0
```

1.7 Ekspresi

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

Variabel global dapat digunakan dalam evaluasi.

```
r:=2; fx:="pi r2"; longest fx()
```

```
12.56637061435917
```

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

```
fx:="asin(x)2"; fx(5, a = -1)
```

```
-0.919535764538
```

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

```
at:=4; function f(expr,x,at) := expr(x); ... f("atx2", 3, 5)//computes 4 × 32not 532
```

```
36
```

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

```
at:=4; function f(expr,x,a) := expr(x,at=a); ... f("atx2", 3, 5)
```

```
45
```

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

```
at:=4; function f(expr,x) := expr(x); ...   f("atx2",at = 5,3)  
45
```

Ekspresi dalam x sering digunakan seperti fungsi.

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

```
f = 5x;  
function f(x) := 6x;  
f(2)  
12
```

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

```
fx = diff(xx,x);fx
```

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

```
"@(a,b) a2 + b2,  
@(a,b) a2 + b241
```

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

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

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

```
fx = x3 - ax; ... a = 1.2; fx(0.5)  
-0.475
```

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

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

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

1.8 Matematika Simbolis

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

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

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

44!

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

$44!/(34!10!)//nilaiC(44, 10)$

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

$binomial(44, 10)//menghitungC(44, 10) menggunakan fungsibinomial()$

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

Anda akan mempelajari bahwa hal berikut juga berfungsi.

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

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

$binomial(x, 3)withx = 10//substitusix = 10keC(x, 3)$

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

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

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

Untuk mencetak ekspresi simbolik dengan LaTeX, gunakan *didepanamp;* (atau *Andadapatmenghilangkanamp;*) jika Anda tidak menginstal LaTeX.

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

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

"v := 5; v²"

25

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

$expand((1 + x)^4), factor(diff($

Sekali lagi,

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

factor(diff(fx, x))

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

```
factor(20!)  
2432902008176640000  
::: factor(10!)  
8 4 2 2 3 5 7  
:: factor(20!)  
18 8 4 2 2 3 5 7 11 13 17 19
```

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

```
::: av:gav2;  
2 g  
fx = x3exp(x),fx  
3 x x E
```

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

```
(fx with x=5),  
5 125 E  
18551.64488782208  
fx(5)  
18551.6448878
```

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

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

```
(fx with x=10)-(fx with x=5), float(  
10 5 1000 E - 125 E  
2.20079141499189e+7  
factor(diff(fx, x, 2))
```

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

```
tex(fx)  
x3 ex
```

Ekspresi simbolik dapat dievaluasi seperti halnya ekspresi numerik.

```
fx(0.5)  
0.206090158838
```

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

fxwithx = 1/2

Penugasan tersebut juga dapat bersifat simbolis.

fxwithx = 1 + t

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

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

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

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

```
1.56155281281
```

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

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

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

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

```
solve( $x^2 + x = 1, x$ ), x2 = x with
```

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

```
sol = solve([x+y=3, x^2 + y^2 = 5], [x, y]);sol, xywithsol[1]
```

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

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

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

```
factor(
```

1.9 Fungsi

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

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

```
function f(x) := xsqrt(x^2 + 1)
```

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

```
f(2)
```

```
4.472135955
```

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

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

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

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

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

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

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

Anda masih dapat memanggil fungsi bawaan sebagai "...", jika itu adalah fungsi di *Euler*.

```
function overwrite sin (x) := _sin(x°) // redine sine in degrees
sin(45)
0.707106781187
Sebaiknya kita hilangkan pendefinisian ulang dosa ini.
forget sin; sin(pi/4)
0.707106781187
```

1.10 Parameter Default

Fungsi numerik dapat memiliki parameter default.

```
function f(x,a=1) := ax²
```

Mengabaikan parameter ini akan menggunakan nilai default.

```
f(4)
```

```
16
```

Mengaturnya akan menimpa nilai default.

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

```
80
```

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

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

```
16
```

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

```
function f(x) := ax²
```

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

```
24
```

Namun, parameter yang ditetapkan akan menggantikan nilai global.

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

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

```
20
```

Fungsi simbolik didefinisikan dengan "amp;=". Fungsi ini didefinisikan dalam Euler dan Maxima, dan berfungsi di kedua dunia. Ekspresi yang mendefinisikan dijalankan melalui Maxima sebelum definisi.

```
function g(x) = x³ - xexp(-x);g(x)
```

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

```
diff(g(x),x),
```

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

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

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

```
function G(x) = factor(integrate(g(x),x)); G(c)//integrate : mengintegralkan  
solve(g(x),0.5)  
0.703467422498
```

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

```
solve(g,0.5)  
0.703467422498  
function P(x,n) = (2x-1)^n;P(x,n)  
function Q(x,n) = (x+2)^n;Q(x,n)  
P(x, 4),expand(  
P(3,4)  
625  
P(x, 4) + Q(x, 3),expand(  
P(x, 4) - Q(x, 3),expand(  
P(x, 4)Q(x, 3),expand(  
P(x, 4)/Q(x, 1),expand(  
function f(x) = x^3 - x;f(x)
```

Dengan `amp` := fungsinya bersifat simbolis, dan dapat digunakan dalam ekspresi simbolis lainnya.

```
integrate(f(x),x)
```

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

yang tidak dapat dievaluasi secara simbolis.

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

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

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

2 6.7

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

2

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

```
function f([a,b]) = a2 + b2 - ab + b;f(a,b), f(x,y)
```

Such a symbolic function can be used for symbolic variables.

But the function can also be used for a numerical vector.

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

17

There are also purely symbolic functions, which cannot be used numerically.

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 + Iy)4),lapl(
```

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

```
function f(x,y) = factor(lapl((x+y2)5,x,y));f(x,y)
```

Singkatnya

* amp;= mendefinisikan fungsi simbolik,

* := mendefinisikan fungsi numerik,

* amp;amp;= mendefinisikan fungsi simbolik murni.

Menyelesaikan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolik.

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

```
solve("x2 - 2",1)
```

1.41421356237

Ini juga berlaku untuk ekspresi simbolik. Ambil fungsi berikut.

```
solve(x2 = 2, x)
```

```
solve(x2 - 2, x)
```

```
solve(ax2 + bx + c = 0, x)
```

```
solve([ax + by = c, dx + ey = f], [x, y])
```

```
px = 4x8 + x7 - x4 - x;px
```

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

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

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

0.966715594851 2

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

```
sol = solve(x2 - x - 1, x);sol
```

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

```
longest sol()
```

```
-0.6180339887498949 1.618033988749895
```

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

```
x^2 withsol[1], expand(x^2 - x - 1 withsol[2])
```

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

```
solve([x + y = 2, x^3 + 2y + x = 4], [x, y])
```

Fungsi f() dapat melihat variabel global. Namun, sering kali kita ingin menggunakan parameter lokal.

dengan a=3.

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

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

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

```
2.54116291558
```

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

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

```
2.54116291558
```

1.11 Menyelesaikan Pertidaksamaan

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

```
load(fourier_elim)
```

```
D:/New folder/Euler x64/maxima/share/maxima/5.35.1/share/fourier_elim/fourier_elim.lisp
```

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

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

```
fourier_elim([x^2 - 10], [x])//x^-1 < 0
```

```
fourier_elim([x^6], [x])
```

```
fourier_elim([x < 1, x > 1], [x])//tidak memilikipenyolesian
```

```
fourier_elim([minf < x, x < inf], [x])//solusinyaR
```

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

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

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

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

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

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

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

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

```
13 lt; y] or [8 lt; x, x lt; y, 13 lt; y] or [y lt; x, 13 lt; y]
```

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

1.12 Bahasa Matriks

Dokumentasi inti EMT berisi pembahasan terperinci tentang bahasa matriks Euler.

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

A=[1,2;3,4]

1 2 3 4

Produk matriks dilambangkan dengan sebuah titik.

b=[3;4]

3 4

b' // transpose b

[3, 4]

inv(A) //inverse A

-2 1 1.5 -0.5

A.b //perkalian matriks

11 25

A.inv(A)

1 0 0 1

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

A.A

7 10 15 22

A²//perpangkatanelemen2A

1 4 9 16

A.A.A

37 54 81 118

power(A,3) //perpangkatan matriks

37 54 81 118

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

1 1 1 1

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

0.333333 0.666667 0.75 1

A

b // hasil kali invers A dan b, A⁽⁻¹⁾b

-2 2.5

inv(A).b

-2 2.5

A

A //A⁽⁻¹⁾A

1 0 0 1

inv(A).A

1 0 0 1

AA //perkalin elemen-elemen matriks seletak

1 4 9 16

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

```
b2//perpangkatanelemen – elemenmatriks/vektor  
9 16
```

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

```
2A  
2 4 6 8
```

Jika salah satu operan merupakan vektor atau skalar, ia diekspansi dengan cara alami. Misalnya, jika operan merupakan vektor kolom, elemen-elemennya diterapkan ke semua baris A.

```
[1,2]A  
1 4 3 8
```

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

```
A[2,3]  
2 6 6 12
```

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

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

```
1 2 1 2  
Adup([1,2],2)  
1 4 3 8
```

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

```
(1:5)(1:5)' // hasilkali elemen-elemen vektor baris dan vektor kolom  
1 2 3 4 5 2 4 6 8 10 3 6 9 12 15 4 8 12 16 20 5 10 15 20 25
```

Sekali lagi, ingatlah bahwa ini bukan produk matriks!

```
(1:5).(1:5)' // hasilkali vektor baris dan vektor kolom  
55  
sum((1:5)(1:5)) // sama hasilnya  
55
```

Bahkan operator seperti lt; atau == bekerja dengan cara yang sama.

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

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

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

```
sum((1:10)\6) // banyak elemen yang kurang dari 6  
5
```

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

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

```
t=(1:10)2; t == 25//mengujielemen2tyangsamadengan25(hanyaada1)  
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
```

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

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

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

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

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

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

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

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

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

```
[4, 48, 95, 139, 147, 191, 238, 282, 290, 334, 381, 425, 433, 477, 524, 568,
```

```
576, 620, 667, 711, 719, 763, 810, 854, 862, 906, 953, 997]
```

EMT tidak sepenuhnya efektif untuk komputasi integer. Ia menggunakan floating point presisi ganda secara internal. Namun, ia sering kali sangat berguna.

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

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

```
112
```

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

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

```
0.765761 0.401188 0.406347 0.267829 0.13673 0.390567 0.495975 0.952814
```

```
0.548138 0.006085 0.444255 0.539246
```

Mengembalikan indeks elemen, yang bukan nol.

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

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

Indeks ini dapat digunakan untuk menetapkan elemen pada nilai tertentu.

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

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

```
0.539246
```

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

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

```
0.765761 0.401188 0.406347 -0.126917 -0.122404 -0.691673 0.495975 0.952814
```

```
0.548138 -0.483902 0.444255 0.539246
```

Dan adalah mungkin untuk mendapatkan unsur-unsur dalam sebuah vektor.

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

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

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

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

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

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

ex[,3]'

[0.765761, 0.952814, 0.548138]

Ini tentu saja sama dengan fungsi max().

max(A)'

[0.765761, 0.952814, 0.548138]

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

j=(1:rows(A))'—ex[,4], mget(-A,j)

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

Fungsi Matriks Lainnya (Membangun Matriks)

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

v=1:3; v_v

1 2 3 1 2 3

Dengan cara yang sama, kita dapat menempelkan suatu matriks ke sisi lain yang berdampingan, jika keduanya memiliki jumlah baris yang sama.

A=random(3,4); A—v'

0.032444 0.0534171 0.595713 0.564454 1 0.83916 0.175552 0.396988 0.83514

2 0.0257573 0.658585 0.629832 0.770895 3

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

Ada pengecualian untuk aturan ini. Bilangan riil yang dilampirkan ke matriks akan digunakan sebagai kolom yang diisi dengan bilangan riil tersebut.

A—1

0.032444 0.0534171 0.595713 0.564454 1 0.83916 0.175552 0.396988 0.83514

1 0.0257573 0.658585 0.629832 0.770895 1

Dimungkinkan untuk membuat matriks dari vektor baris dan kolom.

[v;v]

1 2 3 1 2 3

[v',v']

1 1 2 2 3 3

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

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

1 1 2 4 3 9

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut.

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

2 4 [2, 4] 4

Untuk vektor, ada length().

length(2:10)

9

Ada banyak fungsi lain yang menghasilkan matriks.

ones(2,2)

1 1 1 1

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

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

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

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

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

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

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

```
function rep(v,n) := redim(dup(v,n),1,ncols(v))
```

Mari kita menguji.

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

Fungsi multdup() menduplikasi elemen suatu vektor.

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

Fungsi flipx() dan flipy() membalikkan urutan baris atau kolom matriks.

Yaitu, fungsi flipx() membalik secara horizontal.

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

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

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

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

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

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

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

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

```
drop(1:10,shuffle([0,0,5,5,7,12,12]))  
[1, 2, 3, 4, 6, 8, 9, 10]
```

Ada beberapa fungsi khusus untuk mengatur diagonal atau membuat matriks diagonal.

Kita mulai dengan matriks identitas.

```

A=id(5) // matriks identitas 5x5
1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1
Kemudian kita atur diagonal bawah (-1) menjadi 1:4.
setdiag(A,-1,1:4) //mengganti diagonal di bawah diagonal utama
1 0 0 0 0 1 1 0 0 0 2 1 0 0 0 0 3 1 0 0 0 0 4 1
Perhatikan bahwa kita tidak mengubah matriks A. Kita mendapatkan matriks baru sebagai hasil dari setdiag().
Berikut ini adalah fungsi yang mengembalikan matriks tri-diagonal.
function tridiag (n,a,b,c) := setdiag(setdiag(bid(n),1,c),-1,a); ... tridiag(5,1,2,3)
2 3 0 0 0 1 2 3 0 0 0 1 2 3 0 0 0 1 2 3 0 0 0 1 2
Diagonal matriks juga dapat diekstraksi dari matriks. Untuk menunjukkan hal ini, kami merestrukturisasi vektor 1:9 menjadi matriks 3x3.
A=redim(1:9,3,3)
1 2 3 4 5 6 7 8 9
Sekarang kita dapat mengekstrak diagonalnya.
d=getdiag(A,0)
[1, 5, 9]
Misalnya, kita dapat membagi matriks berdasarkan diagonalnya. Bahasa matriks memastikan bahwa vektor kolom d diterapkan ke matriks baris demi baris.
fraction A/d'
1 2 3 4/5 1 6/5 7/9 8/9 1

```

1.13 Vektorisasi

Hampir semua fungsi di Euler juga berfungsi untuk masukan matriks dan vektor, jika ini masuk akal.

Misalnya, fungsi sqrt() menghitung akar kuadrat dari semua elemen vektor atau matriks.

```

sqrt(1:3)
[1, 1.41421, 1.73205]

```

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

```
x=1:0.01:5; y=log(x)/x^2;//terlalupanjanguntukditampilkan
```

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

Dalam contoh berikut, kita buat vektor nilai t[i] dengan spasi 0,1 dari -1 hingga 1. Kemudian kita buat vektor nilai fungsi

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

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

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

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

shortest (1:5)(1:5)'

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

Perhatikan bahwa ini sangat berbeda dari perkalian matriks. Perkalian matriks dilambangkan dengan titik ". " dalam EMT.

(1:5).(1:5)'

55

Secara default, vektor baris dicetak dalam format ringkas.

[1,2,3,4]

[1, 2, 3, 4]

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

v:=[1,2]; v.v',

5 25

Untuk mentranspos suatu matriks, kita menggunakan tanda apostrof.

v=1:4; v'

1 2 3 4

Jadi kita dapat menghitung matriks A dikali vektor b.

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

30 70

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

v'.v

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

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

v.v'

30

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

norm(v)²

30

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ini ringkasan aturannya.

* Fungsi yang diterapkan pada vektor atau matriks diterapkan pada * setiap elemen.

* Operator yang beroperasi pada dua matriks dengan ukuran yang sama * diterapkan secara berpasangan pada elemen-elemen matriks.

* Jika kedua matriks memiliki dimensi yang berbeda, keduanya * diekspansi dengan cara yang masuk akal, sehingga memiliki ukuran yang * sama.

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

Berikut ini adalah kasus sederhana dengan operator :

[1,2,3]²

[1, 4, 9]

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

v:=[1,2,3]; vv'

1 2 3 2 4 6 3 6 9

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan *!

v.v'

14

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

sum,prod menghitung jumlah dan hasil perkalian baris-baris

cumsum,cumprod melakukan hal yang sama secara kumulatif
menghitung nilai ekstrem dari setiap baris

extrema mengembalikan vektor dengan informasi ekstrem

diag(A,i) mengembalikan diagonal ke-i

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

id(n) matriks identitas

det(A) determinan

charpoly(A) polinomial karakteristik

eigenvalues(A) nilai eigen

vv, sum(vv), cumsum(vv)

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

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

1:4, 1:2:10

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

Untuk menggabungkan matriks dan vektor ada operator "—" dan "".

[1,2,3]—[4,5], [1,2,3]..1

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

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

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

6

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

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

6 [7, 8, 9]

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

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

[2, 4] 2 5 8

Bentuk singkat dari : adalah menghilangkan indeks sepenuhnya.

A[,2:3]

2 3 5 6 8 9

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

A4

4

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

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

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

```
deformat; w=0°:45°:360°; w=w'; deg(w)
0 45 90 135 180 225 270 315 360
```

Sekarang kita tambahkan kolom ke matriks.

```
M = deg(w)−w−cos(w)−sin(w)
0 0 1 0 45 0.785398 0.707107 0.707107 90 1.5708 0 1 135 2.35619 -0.707107
0.707107 180 3.14159 -1 0 225 3.92699 -0.707107 -0.707107 270 4.71239 0 -1 315
5.49779 0.707107 -0.707107 360 6.28319 1 0
```

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

Dalam contoh berikut, kita menghitung $t[j]^i$ untuk i dari 1 hingga n . Kita memperoleh matriks, yang setiap baris $a_{i,j} = t_j^i$, $1 \leq j \leq 101$, $1 \leq i \leq n$

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

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

```
function map f(x) := integrate("x^x", 1, x)
The "map" keyword vectorizes the function. The function will now work
for vectors of numbers.
f([1:5])
[0, 2.05045, 13.7251, 113.336, 1241.03]
```

1.14 Sub-Matriks dan Elemen Matriks

Untuk mengakses elemen matriks, gunakan notasi tanda kurung.

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

Kita dapat mengakses baris matriks yang lengkap.

```
A[2]
[4, 5, 6]
```

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

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

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

```
A[2,]
[4, 5, 6]
```

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

Di sini kita menginginkan baris pertama dan kedua dari A.

A[[1,2]]

1 2 3 4 5 6

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

A[[3,2,1]]

7 8 9 4 5 6 1 2 3

Trik indeks juga berfungsi dengan kolom.

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

A[1:3,2:3]

2 3 5 6 8 9

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

A[:,3]

3 6 9

Atau, biarkan indeks pertama kosong.

A[,2:3]

2 3 5 6 8 9

Kita juga bisa mendapatkan baris terakhir A.

A[-1]

[7, 8, 9]

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

A[1,1]=4

4 2 3 4 5 6 7 8 9

Kita juga dapat menetapkan nilai ke baris A.

A[1]=[-1,-1,-1]

-1 -1 4 5 6 7 8 9

Kita bahkan dapat menetapkannya ke submatriks jika ukurannya tepat.

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

5 6 -1 7 8 6 7 8 9

Selain itu, beberapa jalan pintas diperbolehkan.

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

0 0 -1 0 0 6 7 8 9

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

A[4]

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

1.15 Sortir dan Acak

Fungsi sort() mengurutkan vektor baris.

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

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

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

Mari kita acak sebuah vektor.

```
v=shuffle(1:10)  
[4, 5, 10, 6, 8, 9, 1, 7, 2, 3]
```

Indeks berisi urutan v yang tepat.

```
vs,ind=sort(v); v[ind]  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Ini juga berlaku untuk vektor string.

```
s=[”a”, ”d”, ”e”, ”a”, ”aa”, ”e”]  
a d e a aa e  
ss,ind=sort(s); ss
```

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

Seperti yang Anda lihat, posisi entri ganda agak acak.

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

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

```
intrandom(1,10,10), unique(  
[4, 4, 9, 2, 6, 5, 10, 6, 5, 1] [1, 2, 4, 5, 6, 9, 10]
```

Ini juga berlaku untuk vektor string.

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

1.16 Aljabar Linier

EMT memiliki banyak fungsi untuk memecahkan sistem linier, sistem renggang, atau masalah regresi.

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

```
A=[1,2;3,4]; b=[5;6]; A  
b  
-4 4.5
```

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

```
A=normal(200,200); b=sum(A); longest totalmax(abs(inv(A).b-1))  
8.790745908981989e-13
```

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

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

Determinan matriks ini adalah 0.

```
det(A)  
0
```

1.17 Matriks Simbolik

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

```
A = [a,1,1;1,a,1;1,1,a]; A  
det(A),factor(  
invert(A)witha = 0  
A = [1,a;b,2]; A
```

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

```
det(A - xident(2)),solve()
```

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

```
eigenvalues([a, 1; 1, a])
```

Untuk mengekstrak vektor eigen tertentu dibutuhkan pengindeksan yang cermat.

```
eigenvectors([a, 1; 1, a]),  
[1, - 1]
```

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

```
A(a=4,b=5)  
1 4 5 2
```

Dalam ekspresi simbolik, gunakan dengan.

```
Awith[a = 4, b = 5]
```

Akses terhadap baris matriks simbolik bekerja seperti halnya matriks numerik.

```
A[1]
```

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

```
A[1,1]:=t+1; A
```

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

```
v = makelist(1/(i+j),i,1,3); v  
B := [1,2;3,4]; B,invert(B)
```

The result can be evaluated numerically in Euler. For more information about Maxima, see the introduction to Maxima.

```
invert(B)()  
-2 1 1.5 -0.5
```

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

Perlu dicatat, bahwa dengan `amp;:=` matriks B telah didefinisikan sebagai simbolik dalam ekspresi simbolik dan sebagai numerik dalam ekspresi numerik. Jadi kita dapat menggunakan di sini.

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

```
1 0 0 1
```

Misalnya nilai eigen A dapat dihitung secara numerik.

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

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

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

```
eigenvalues(@A)
```

Nilai Numerik dalam Ekspresi Simbolik

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

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

```
1 3.14159 4 5
```

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

```
A
```

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

```
mxmset(A); A
```

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

```
bfloat(sqrt(2)),float(sqrt(2))
```

Ketepatan angka floating point besar dapat diubah.

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

```
3.14159265358979323846264338327950288419716939937510582097494 45923078164062862089986280348253
```

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

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

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

1.18 Demo - Suku Bunga

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

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

```
K=5000
```

```
5000
```

Sekarang kita asumsikan suku bunga 3satu suku bunga sederhana dan hitung hasilnya.

```

K1.03
5150
Euler juga akan memahami sintaksis berikut.
K+K3
5150
Namun lebih mudah menggunakan faktor
q=1+3
1.03 5150
Selama 10 tahun, kita cukup mengalikan faktor-faktornya dan mendapatkan
nilai akhir dengan suku bunga majemuk.
Kq10
6719.58189672
Untuk keperluan kita, kita dapat mengatur format menjadi 2 digit setelah
titik desimal.
format(12,2); Kq10
6719.58
Mari kita cetak angka tersebut dibulatkan menjadi 2 digit dalam kalimat
lengkap.
"Starting from " + K + " you get" + round(Kq10, 2) + "."
Starting from 5000 you get 6719.58.
Bagaimana jika kita ingin mengetahui hasil antara dari tahun 1 hingga tahun
9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu
menulis loop, tetapi cukup masukkan
Kq(0 : 10)
Real 1 x 11 matrix
5000.00 5150.00 5304.50 5463.64 ...
Bagaimana keajaiban ini bekerja? Pertama, ekspresi 0:10 menghasilkan vek-
tor bilangan bulat.
short 0:10
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Maka semua operator dan fungsi di Euler dapat diaplikasikan ke vektor ele-
men demi elemen. Jadi
short q(0 : 10)
[1, 1.03, 1.0609, 1.0927, 1.1255, 1.1593, 1.1941, 1.2299, 1.2668, 1.3048, 1.3439]
adalah vektor faktor  $q^0$  hingga  $q^{10}$ . Ini dikalikan dengan  $K$ , dan kita memperoleh vektor nilai.
VK=Kq(0 : 10);
Variable q not found! Error in Error in : VK = K * q(0 : 10);...
Tentu saja, cara realistik untuk menghitung suku bunga ini adalah dengan
membulatkannya ke sen terdekat setelah setiap tahun. Mari kita tambahkan
fungsi untuk ini.
function oneyear (K) := round(Kq,2)
Mari kita bandingkan kedua hasil, dengan dan tanpa pembulatan.
longest oneyear(1234.57), longest 1234.57q
1271.61 1271.6071
Sekarang tidak ada rumus sederhana untuk tahun ke-n, dan kita harus men-
gulangnya selama bertahun-tahun. Euler menyediakan banyak solusi untuk ini.

```

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

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

Real 1 x 11 matrix

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

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

```
VKr'
```

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

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

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

```
5150.00 5000.00 5150.00 5304.50
```

Anehnya, kita juga dapat menggunakan vektor indeks. Ingat bahwa 1:3 menghasilkan vektor [1,2,3].

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

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

```
6719.60 6719.58
```

Perbedaannya sangat kecil.

1.19 Menyelesaikan Persamaan

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

```
function onepay (K) := Kq+R
```

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

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

Real 1 x 11 matrix

```
5000.00 5350.00 5710.50 6081.82 ...
```

Bagaimana jika kita menghilangkan jumlah yang sama setiap tahun?

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

Real 1 x 11 matrix

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

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

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

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

Real 1 x 51 matrix

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

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

```
min(nonzeros(VKR[i0]))  
48.00
```

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

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

Fungsi iterate() memiliki satu trik lagi. Fungsi ini dapat mengambil kondisi akhir sebagai argumen. Kemudian, fungsi ini akan mengembalikan nilai dan jumlah iterasi.

```
x,n=iterate("onepay",5000,till="xj0"); x, n,  
-19.83 47.00
```

Mari kita coba menjawab pertanyaan yang lebih ambigu. Asumsikan kita tahu bahwa nilainya adalah 0 setelah 50 tahun. Berapa tingkat bunganya?

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

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

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

Iterasinya sama seperti di atas

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

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

Mari kita coba uji coba.

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

Sekarang kita bisa memecahkan masalah kita.

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

Rutin solve menyelesaikan ekspresi=0 untuk variabel x. Jawabannya adalah 3,15 tersebut. Fungsi solve() selalu membutuhkan nilai awal.

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

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

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

1.20 Solusi Simbolis untuk Masalah Suku Bunga

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

```
function op(K) = Kq+R; op(K)
```

Sekarang kita dapat mengulanginya.

```
op(op(op(op(K)))) ,expand()
```

Kita melihat suatu pola. Setelah n periode kita memiliki

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

```
sum(qk, k, 0, n - 1);
```

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

Mari kita buat fungsi untuk ini.

```
function fs(K,R,P,n) = (1+P/100)n K + ((1+P/100)n-1)/(P/100) R; fs(K,R,P,n)
```

Fungsi ini melakukan hal yang sama seperti fungsi f sebelumnya. Namun, fungsinya lebih efektif.

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

```
-19.82504734650985 -19.82504734652684
```

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

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

```
20.5061016552
```

Jawaban ini menyatakan bahwa akan negatif setelah 21 tahun.

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

Asumsikan kita mendapatkan pinjaman sebesar K, dan membayar n kali cicilan sebesar R (dimulai setelah tahun pertama) sehingga menyisakan utang residual sebesar Kn (pada saat pembayaran terakhir). Rumus untuk ini jelas

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

Biasanya rumus ini diberikan dalam bentuk

```
equ = (equ with P=100i); equ
```

Kita dapat mencari laju R secara simbolis.

```
solve(equ, R)
```

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

Tentu saja, kita memiliki limit berikut.

```
limit(R(5000,0,x,10),x,0)
```

Jelas, tanpa bunga, kita harus membayar kembali 10 suku bunga sebesar 500.

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

```
fn = solve(equ,n) — ratsimp; fn
```

R.2 Latihan Soal

Soal No 49

Menyederhanakan:

$((24a^10b^(-8)c^7)/(12a^6b^-3c^5))^(- 5)$
 Soal No 50
 Menyederhanakan:
 $((125p^12q^(-14)r^22)/25p^8q^6r^(-15))^(- 4)$
 Soal No 90
 calculate
 $2^62^{-3}/2^10/2^{-8}$
 2
 Soal No 91
 Calculate
 $(4(8-6)^2 - 43 + 28)/(3^1 + 19^0)$
 5
 Soal No 92
 Calculate
 $((4(8-6)^2 - 4)3 + 28)/(3^1 + 9^0)$
 13
 R.3 Latihan soal
 no 27
 $showev('expand((x + 3)^2))$
 no 29
 $showev('expand((y - 5)^2))$
 no 33
 $showev('expand((2x + 3y)^2))$
 no 39
 $showev('expand((3y + 4)(3y - 4)))$
 no 42
 $showev('expand((3x + 5y)(3x - 5y)))$
 R.4 Latihan Soal
 Faktor Trinomial
 Nomor 23
 $solve(t^2 + 8t + 15)$
 Nomor 24
 $solve(y^2 + 12y + 27)$
 Nomor 47
 $solve(z^2 - 81)$
 Nomor 48
 $solve(m^2 - 4)$
 Nomor 49
 $solve(16x^2 - 9)$
 R.5 Exercise Set
 soal no 36
 tentukan nilai y
 $solve(y^2 - 4y - 45, y)$
 soal no 38
 tentukan nilai y
 $solve(t^2 + 6t, t)$

soal no 41
 tentukan nilai x
 $solve(x^2 + 100 = 20x, x)$
 soal no 42
 tentukan nilai y
 $solve(y^2 + 25 = 10y, y)$
 soal no 45
 tentukan nilai y
 $solve(3y^2 + 8y + 4, y)$
 R.6 Exercise Set
 Nomor 9
 Menyederhanakan
 $((x^2)/(x^2 - 4x + 4)), factor()$
 Nomor 11
 Menyederhanakan
 $((x^3 - 6x^2 + 9x)/(x^3 - 3x^2)), factor()$
 Nomor 14
 Menyederhanakan
 $((2x^2 - 20x + 50)/(10x^2 - 30x - 100)), factor()$
 Nomor 15
 Menyederhanakan
 $((4 - x)/(x^2 + 4x - 32)), factor()$
 Nomor 16
 Menyederhanakan
 $((6 - x)/(x^2 - 36)), factor()$
 Latihan Ulangan
 Multiply
 Nomor 71
 function P(a,n) = $(t^a + t^c - a))^n; P(a,n)$
 $P(a, 2), expand()$
 Nomor 72
 $'((y^b - z^c)(y^b + z^c)) = expand((y^b - z^c)(y^b + z^c))$
 Nomor 73
 $'((a^n - b^n)^3) = expand((a^n - b^n)^3)$
 Nomor 32
 $'(((x^2 + x - 6)/(x^2 + 8x + 15))((x^2 - 25)/(x^2 - 4x + 4))) = simplify(((x^2 + x - 6)/(x^2 + 8x + 15))((x^2 - 25)/(x^2 - 4x + 4)))$
 $solve(((x^2 + x - 6)/(x^2 + 8x + 15))((x^2 - 25)/(x^2 - 4x + 4)), x)$
 Nomor 33
 $'(((x)/(x^2 - 1)) - ((3)/(x^2 + 4x - 5))) = simplify(((x)/(x^2 - 1)) - ((3)/(x^2 + 4x - 5)))$
 $solve(((x)/(x^2 - 1)) - ((3)/(x^2 + 4x - 5)))$
 2.3 Exercise Set
 Cari
 dan domain nya !
 Nomor 1

$gx := x - 3; fx := gx + 3; fx$

dengan domainnya

$fx := x + 3; gx := fx - 3; gx$

dengan domainnya

Nomor 2

$gx := 1/x; fx := 4/(1-5gx); fx$

dengan domain

$fx := 4/(1 - 5x); gx := 1/fx; gx$

dengan domainnya

Diberikan fungsi

cari

Nomor 3

$x := 1/3; gx := x^2 - 2x - 6; fx := 3gx + 1; fx$

Nomor 4

$x := 1/2; hx := x^3; gx := hx^2 - 2hx - 6; gx$

Nomor 5

$x := 5; gx := x^2 - 2x - 6; gx := x^2 - 2x - 6; gx$

3.1 Latihan Soal

Gunakan rumus kuadrat untuk menemukan solusi yang tepat

Nomor 39

$solve(5m^2 + 3m = 2, m)$

Nomor 40

$solve(2y^2 - 3y - 2 = 0, y)$

Nomor 48

$solve(2t^2 - 5t = 1, t)$

Solve.

Nomor 83

$solve(y^4 + 4y^2 - 5 = 0, y)$

Nomor 84

$solve(y^4 - 15y^2 - 16 = 0, y)$

3.4 Latihan Soal

Solve

Nomor 1

$solve(1/4 + 1/5 = 1/t, t)$

Nomor 4

$solve((t + 1)/1 - (t - 1)/2 = 1, t)$

Nomor 6

$solve(1/t + 1/2t + 1/3t = 5, t)$

Nomor 11

$solve((5/(y + 3)) + (2/y) = (5y - 3)/(y^2 - 9), y)$

Nomor 18

$solve(3y + 5/y^2 + 5y + y + 4/y + 5 = y + 1/y, y)$

3.5 Latihan Soal

load(fourier_elim)

D:/New folder/Euler x64/maxima/share/maxima/5.35.1/share/fourier_elim/fourier_elim.lisp

Nomor 44

```

kill(x);
fourier_elim([4x] 20, [x])//4x 20
Nomor 45
fourier_elim([x + 8] < 9, [x])//x + 8 < 9
Nomor 47
fourier_elim([x + 8] = 9, [x])//x + 8 = 9
Nomor 52
fourier_elim([x + 8] = 9, [x])//x + 8 = 9
Nomor 62
fourier_elim([3x + 5] < 0, [x])//3x + 5 < 0
Chapter 3 Test
kill(x);
load(fourier_elim)
D:/New folder/Euler x64/maxima/share/maxima/5.35.1/share/fourier_elim/fourier_elim.lisp
Nomor 8
solve(3/3x + 4 + 2/x - 1 = 2, x)
Nomor 11
fourier_elim([x + 4] = 7, [x])//x + 4 = 7
Nomor 12
fourier_elim([4y - 3] = 5, [x])//4y - 3 = 5
Solve
Nomor 13
fourier_elim([x + 3] <= 4, [x])//x + 3 <= 4
Nomor 15
fourier_elim([x + 5] 2, [x])//x + 5 2
Nomor 19
solve(x2 + 4x = 1, x)
4.1 Latihan Soal
Gunakan substitusi untuk menentukan apakah 2,3 dan -1 adalah nol dari
Nomor 23
function P(x) = (x3 - 9x2 + 14x + 24);P(x)
P(2)
24
P(3)
12
P(-1)
0
Jadi hasil substitusi yang mempunyai nilai nol adalah dengan mensubstisusi
angka -1
Nomor 24
function P(x) = (2x3 - 3x2 + x + 6);P(x)
P(2)
12
P(3)
36
P(-1)

```

0

Jadi hasil substitusi yang mempunyai nilai nol adalah dengan mensubstisusi angka -1

Nomor 25

function $P(x) = (x^4 - 6x^3 + 8x^2 + 6x - 9); P(x)$

P(2)

3

P(3)

0

P(-1)

0

Jadi hasil substitusi yang menghasilkan persamaan mempunyai nilai nol adalah dengan mensubstisusi angka 3 dan -1

Nomor 37

$solve(x^4 - 4x^2 + 3, x)$

Nomor 39

$solve(x^3 + 3x^2 - x - 3, x)$

4.3 Latihan Soal

Nomor 1

Untuk fungsi

Gunakan pembagian panjang untuk menentukan apakah masing-masing dari berikut ini merupakan faktor dari $f(x)$

a) $x+1$

b) $x-2$

c) $x + 5$

function $f(x) = (x^4 - 6x^3 + x^2 + 24x - 20); f(x)$

$f(x + 1), expand()$

$f(x - 2), expand()$

$f(x + 5), expand()$

Nomor 23

Gunakan pembagian sintetis untuk menemukan nilai fungsi.

cari $f(1)$, $f(-2)$, dan $f(3)$

function $f(x) = (x^3 - 6x^2 + 11x - 6); f(x)$

$f(1)$

0

$f(-2)$

-60

$f(3)$

0

Nomor 24

cari $f(-3), f(-2)$, dan $f(1)$

function $f(x) = (x^3 + 7x^2 - 12x - 3); f(x)$

$f(-3)$

69

$f(-2)$

41

$f(1)$
 -7
 Nomor 25
 cari $f(-1), f(4)$ dan $f(-5)$
 function $f(x) = (x^4 - 3x^3 + 2x + 8); f(x)$
 $f(-1)$
 10
 $f(4)$
 80
 $f(-5)$
 998
 Faktorkan fungsi polinomial $f(x)$. Kemudian selesaikan persamaan $f(x)=0$
 Nomor 39
 $fx = (x^3 + 4x^2 + x - 6 = 0); fx$
 $factor((fx, x^3 + 4x^2 + x - 6 = 0))$
 $solve(x^3 + 4x^2 + x - 6 = 0, x)$

1.21 Ulasan Campuran Tengah Bab

gunakan pembagian sintetis untuk menemukan nilai fungsi

Nomor 18
 cari $g(-5)$
 function $g(x) = (x^3 - 9x^2 + 4x - 10); g(x)$
 $g(-5)$
 -380
 Nomor 19
 cari $f(1/2)$
 function $f(x) = (20x^2 - 40x); f(x)$
 $f(1/2)$
 -15

Dengan menggunakan pembagian sintetis, tentukan apakah angka-angka tersebut adalah nol dari fungsi polinomial.

Angka 22
 -1,5;
 lateks: $f(x)=x^6 - 35x^4 + 259x^2 - 225$
 function $f(x) = (x^6 - 35x^4 + 259x^2 - 225); f(x)$
 $f(-1.5)$
 191.953125

Faktorkan fungsi polinomial $f(x)$. Kemudian selesaikan persamaan $f(x) = 0$.

Nomor 23
 $hx = (x^3 - 2x^2 - 55x + 56 = 0); hx$
 $factor((hx, x^3 - 2x^2 - 55x + 56 = 0))$
 $solve(x^3 - 2x^2 - 55x + 56 = 0, x)$
 Nomor 24
 $gx = (x^4 - 2x^3 - 13x^2 + 14x + 24 = 0); gx$
 $factor((gx, x^4 - 2x^3 - 13x^2 + 14x + 24 = 0))$

```
solve(x^4 - 2x^3 - 13x^2 + 14x + 24 = 0, x)
```

2 Plot 2D

2.1 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).

2.2 Plot Dasar

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

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

```
clg; // clear screen

window(0,0,1024,1024); // use all of the window

setplot(0,1,0,1); // set plot coordinates

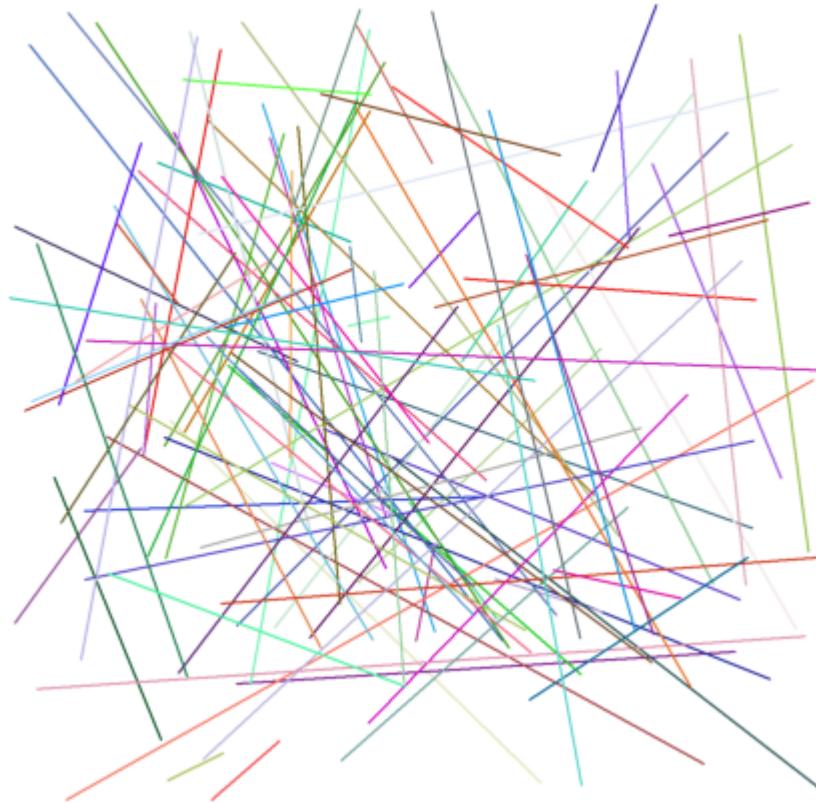
hold on; // start overwrite mode

n=100; X=random(n,2); Y=random(n,2); // get random points

colors=rgb(random(n),random(n),random(n)); // get random colors

loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot

insimg;
```



```
hold off;
reset;
```

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

Untuk membersihkan semua yang telah kita lakukan, kita menggunakan `reset()`.

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

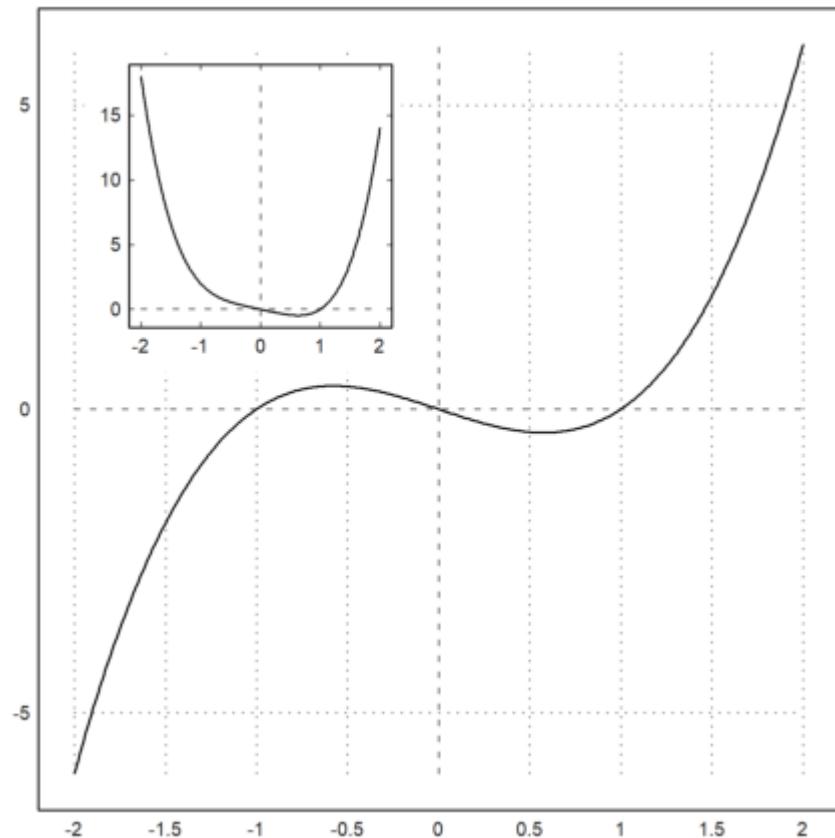
Sebagai contoh lain, kita menggambar plot sebagai inset di plot lain. Ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan beberapa margin untuk ini sesuai kebutuhan. Perhatikan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

```
plot2d("x3 - x");
```

```

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

```



```

hold off;
window(ow);

```

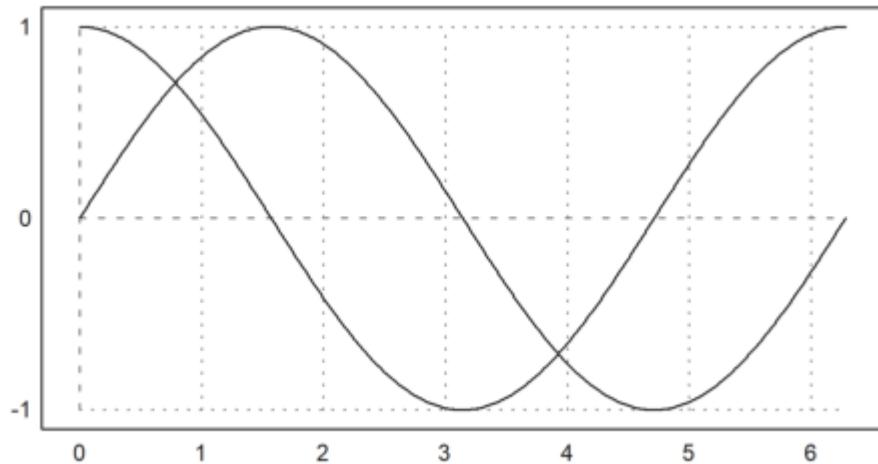
Plot dengan beberapa gambar dicapai dengan cara yang sama. Ada fungsi utilitas figure() untuk ini.

2.3 Plot Aspect

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

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

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



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

Fungsi reset() mengembalikan default plot termasuk rasio aspek.

2.4 Plot 2D di Euler

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

Dimungkinkan untuk 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 di bidang,
- * kurva implisit dengan level atau daerah level.
- * Fungsi kompleks

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

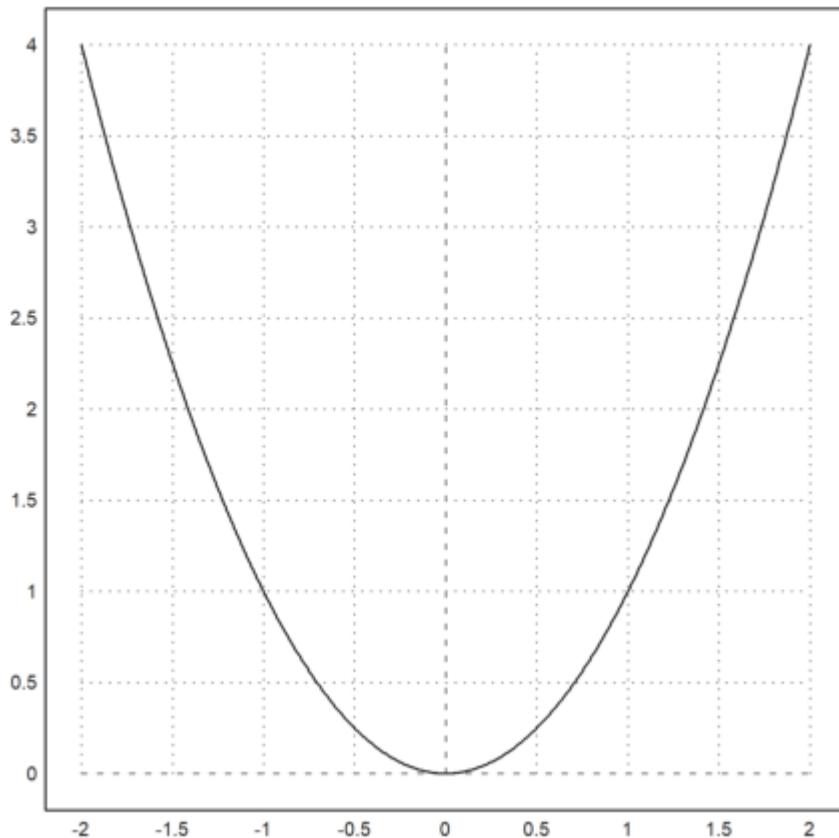
2.5 Plot Ekspresi atau Variabel

Ekspresi tunggal dalam "x" (misalnya "4*x²") atau nama fungsi (misalnya "f") menghasilkan grafik fungsi.

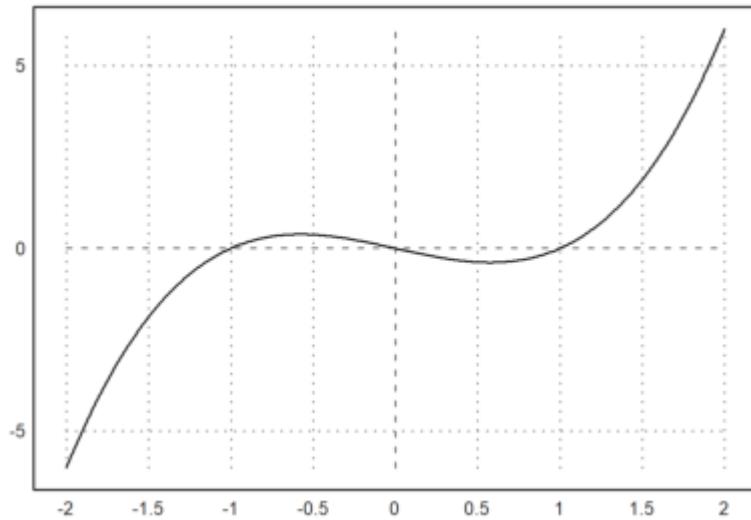
Berikut adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsi.

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

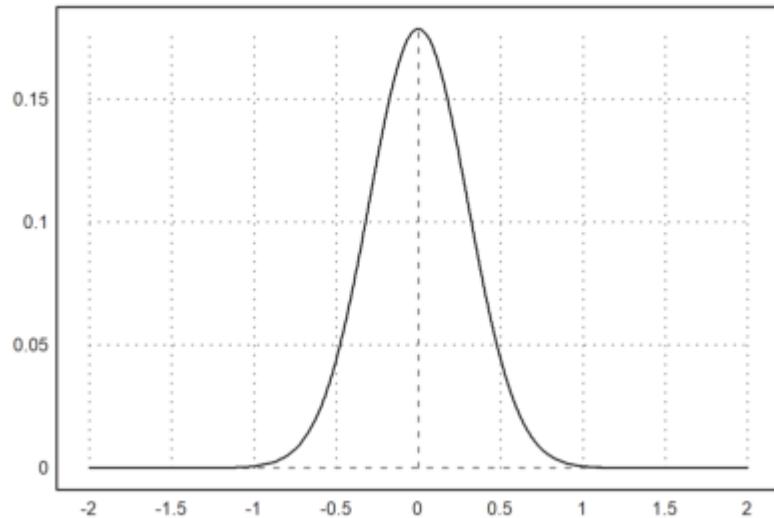
plot2d("x²") :



aspect(1.5); plot2d("x³ - x") :



```
a:=5.6; plot2d("exp(-ax^2)/a"); insimg(30);
```

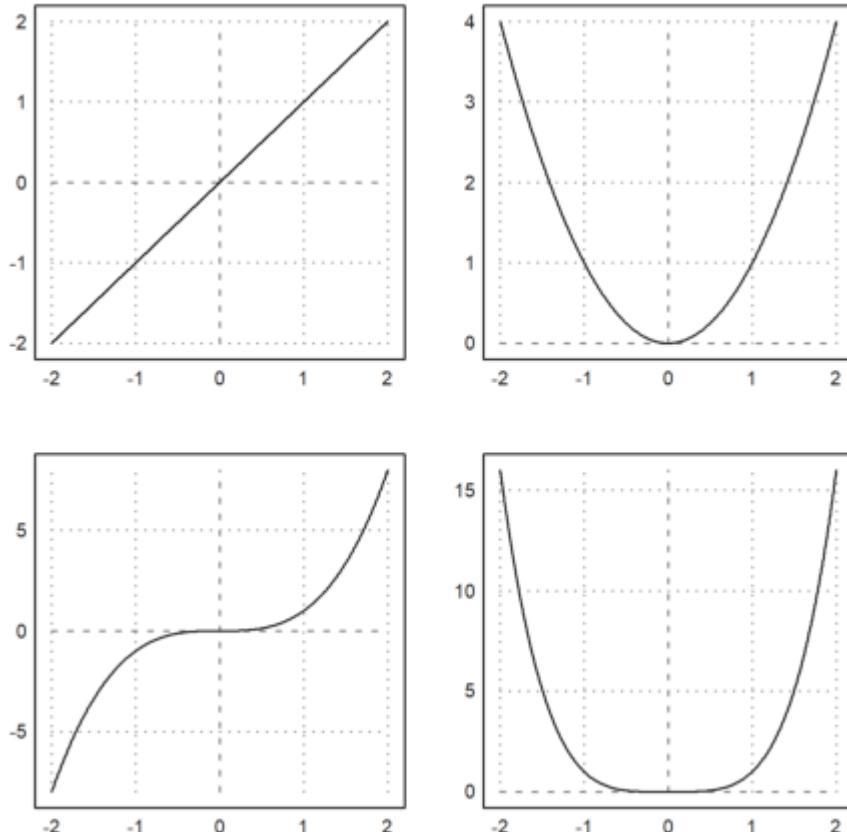


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

Rentang plot ditetapkan dengan parameter yang ditetapkan berikut

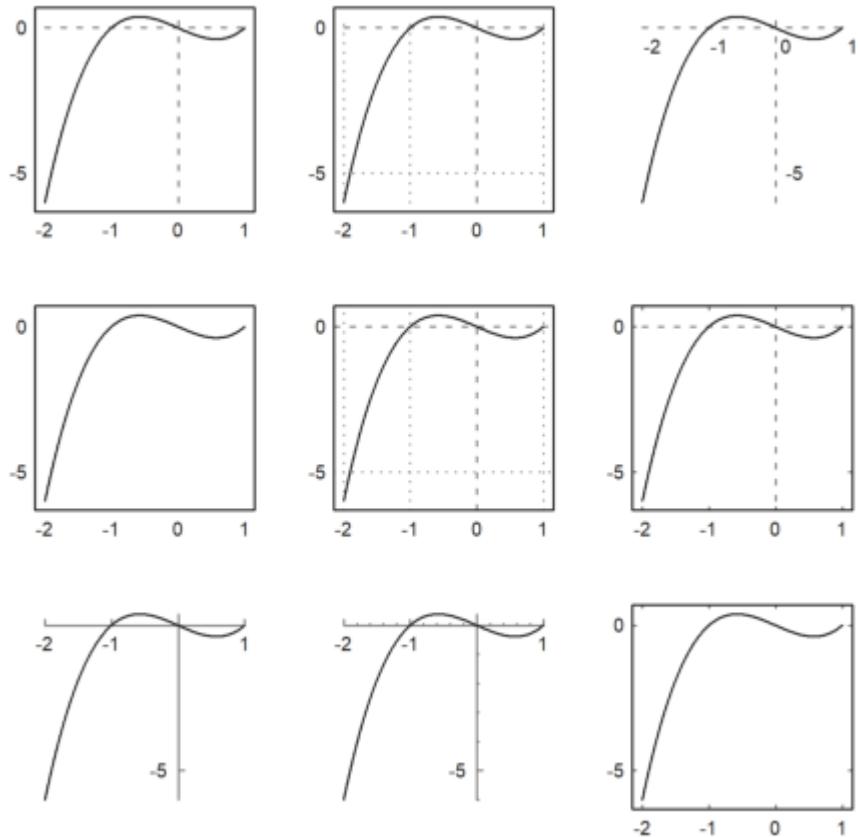
- * a,b: rentang x (default -2,2)
- * c,d: rentang y (default: skala dengan nilai)
- * r: alternatifnya radius di sekitar pusat plot

```
* cx,cy: koordinat pusat plot (default 0,0)
reset;
figure(2,2); ... for n=1 to 4; figure(n); plot2d("x" + n); end; ... figure(0) :
```



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

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

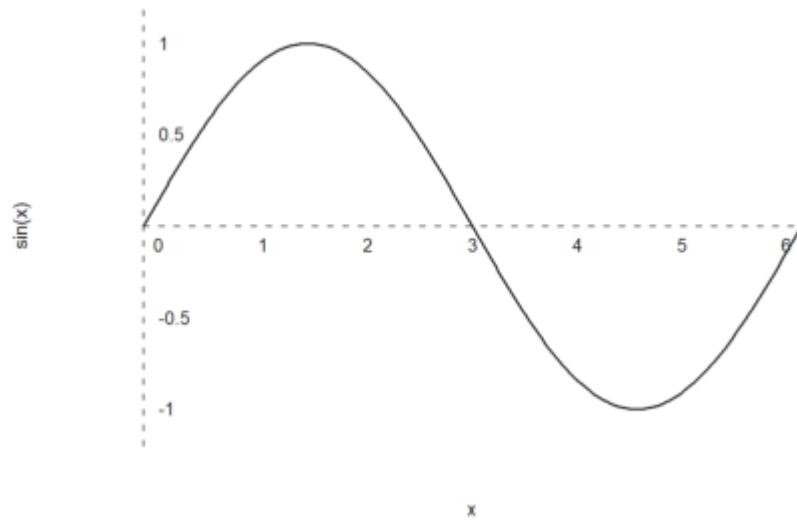


Jika argumen untuk `plot2d()` adalah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot.

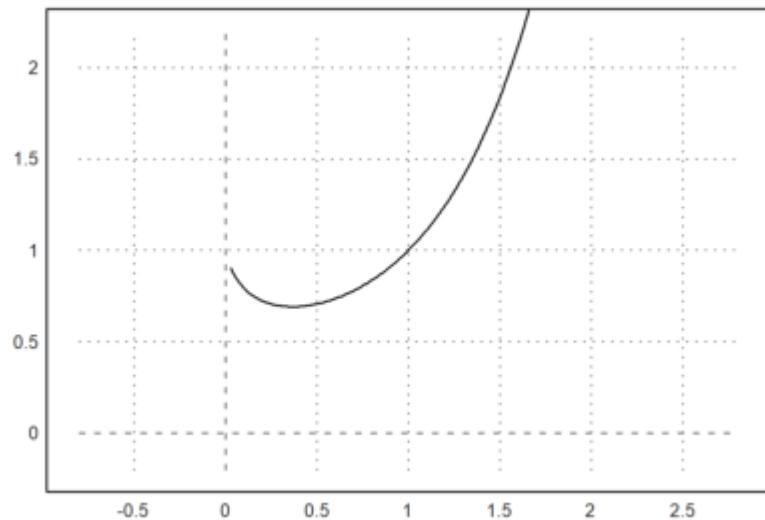
Atau, a, b, c, d dapat ditetapkan sebagai parameter yang ditetapkan sebagai `a=... dst.`

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

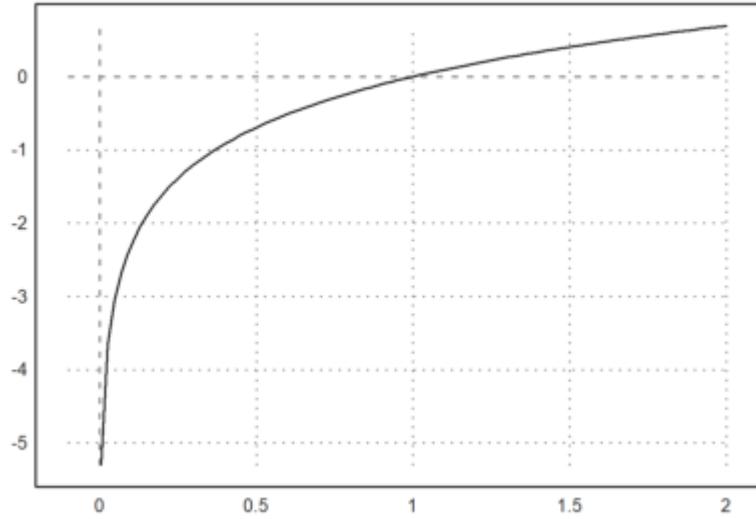
```
aspect(1.5); plot2d("sin(x)",0,2pi,-1.2,1.2,grid=3,xl="x",yl="sin(x)"):
```



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

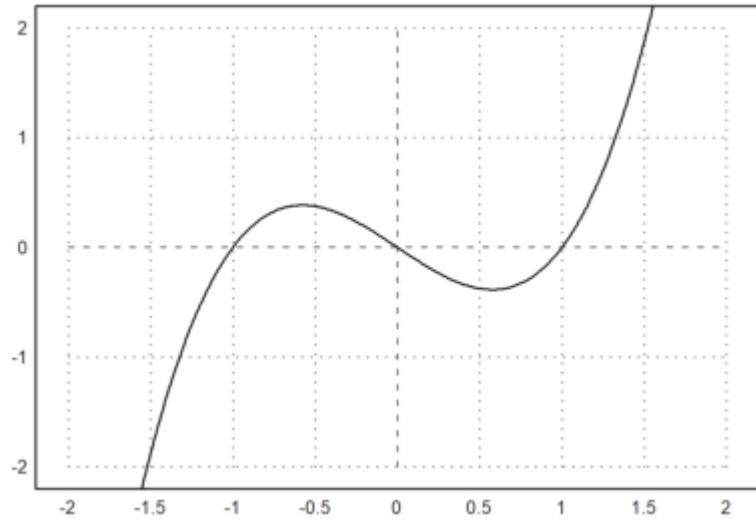


Perhatikan bahwa x^x tidak didefinisikan untuk $x \leq 0$. Fungsi plot2d menangkap kesalahan ini, dan mulai me

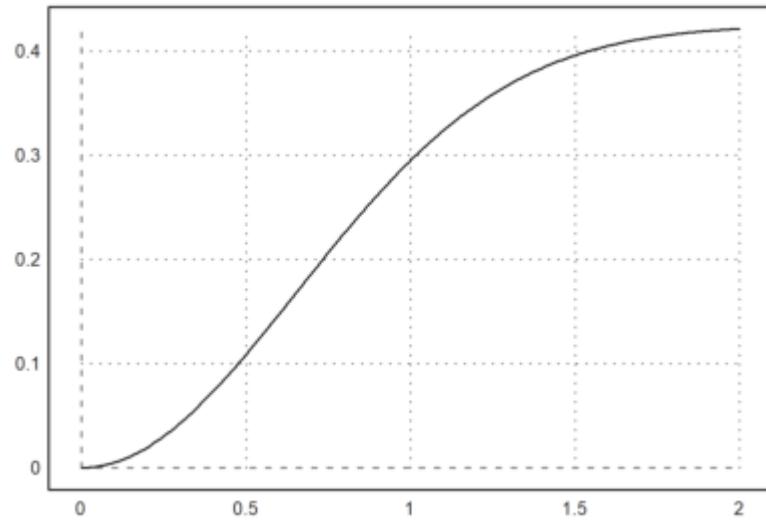


Parameter `square=true` (atau `gt;square`) memilih rentang y secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

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

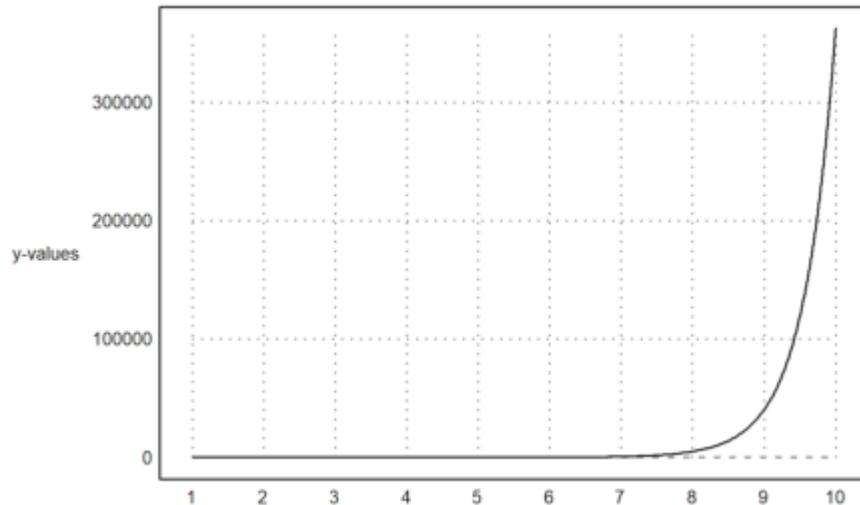


`plot2d("integrate("sin(x)exp(-x^2)", 0, x)", 0, 2) : //plotintegral`



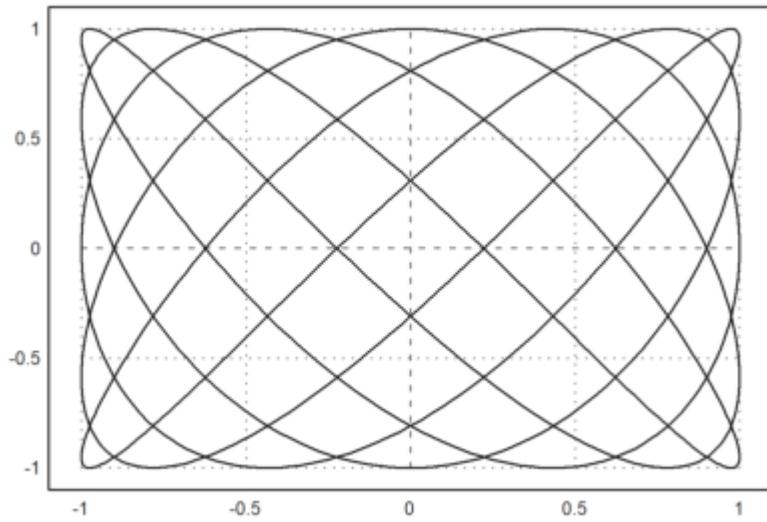
Jika Anda memerlukan lebih banyak ruang untuk label-y, panggil shrinkwindow() dengan parameter yang lebih kecil, atau tetapkan nilai positif untuk "lebih kecil" di plot2d().

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

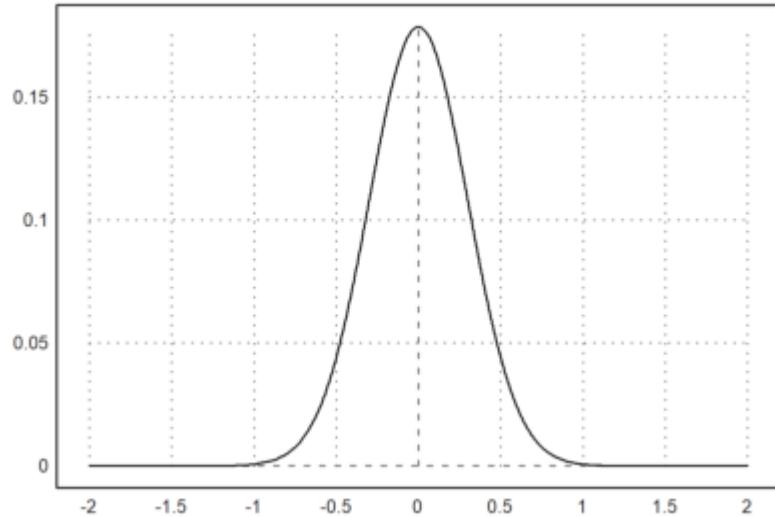


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

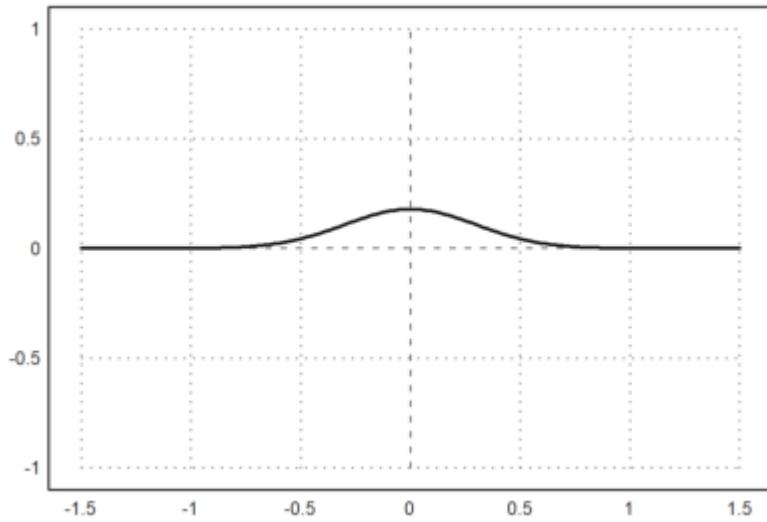
```
x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



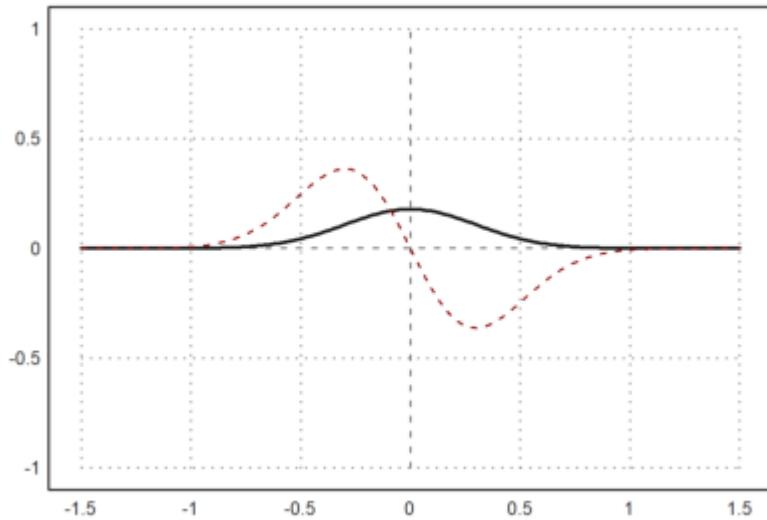
```
a:=5.6; expr = exp(-ax^2)/a; //define expression
plot2d(expr,-2,2); // plot from -2 to 2
```



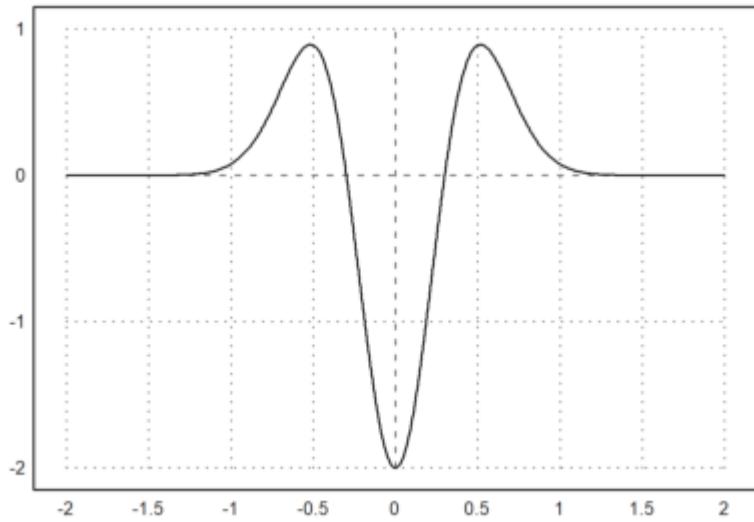
```
plot2d(expr,r=1,thickness=2); // plot in a square around (0,0)
```



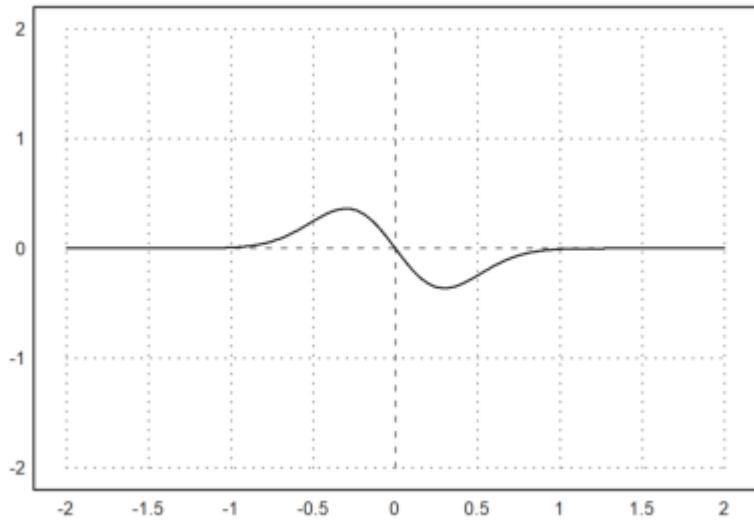
```
plot2d(diff(expr,x), add,style="--",color=red); // add another plot
```



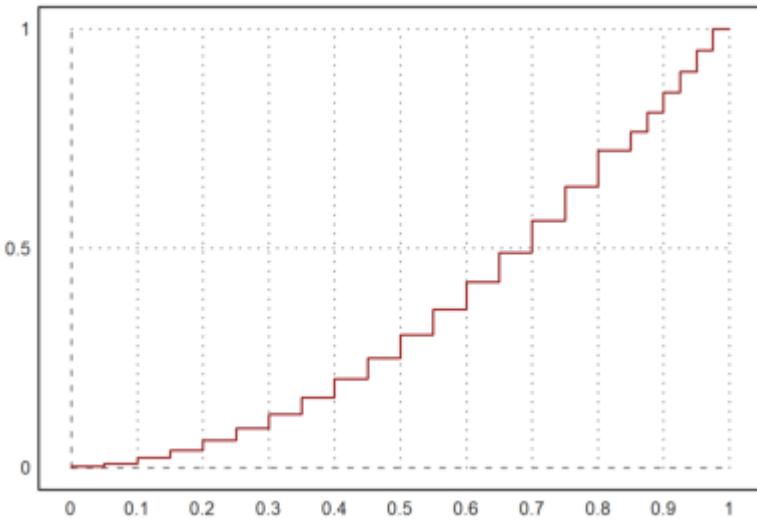
```
plot2d(diff(expr,x,2),a=-2,b=2,c=-2,d=1); // plot in rectangle
```



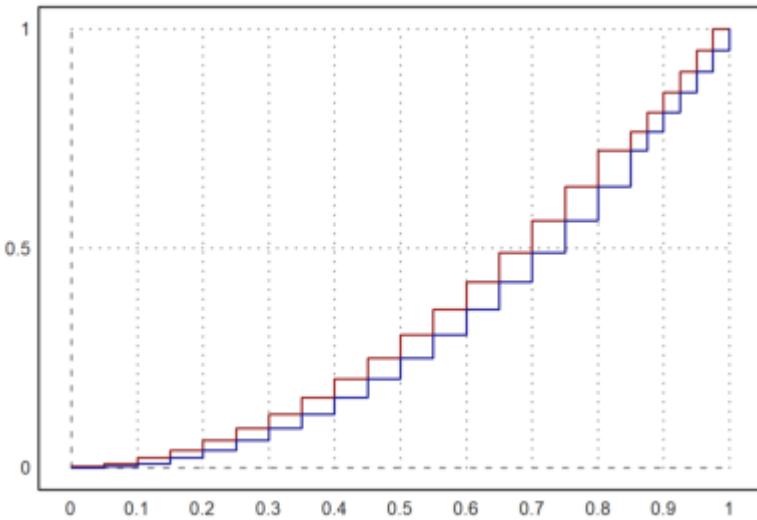
```
plot2d(diff(expr,x),a=-2,b=2, square): // keep plot square
```



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



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



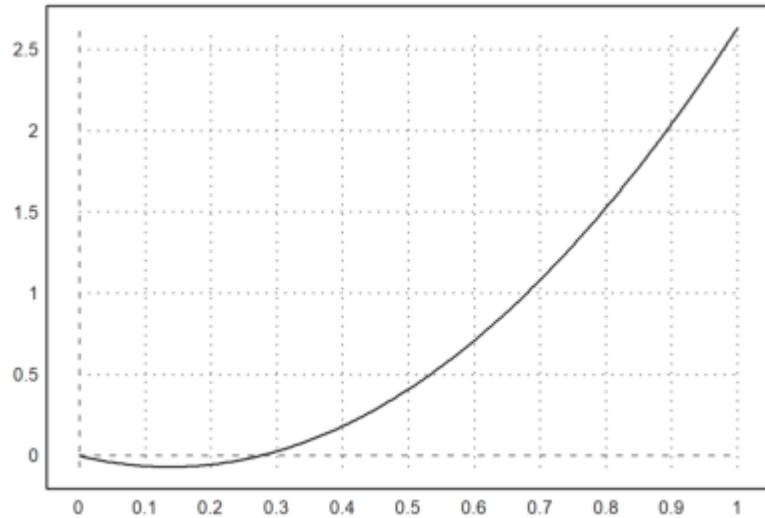
2.6 Fungsi dalam Satu Parameter

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

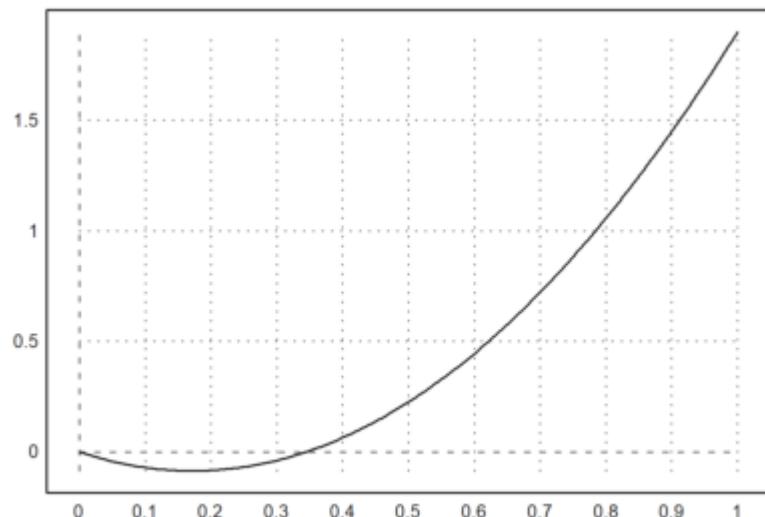
Berikut ini beberapa contoh penggunaan fungsi. Seperti biasa dalam EMT, fungsi yang berfungsi untuk fungsi atau ekspresi lain, Anda dapat meneruskan

parameter tambahan (selain x) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan koleksi panggilan.

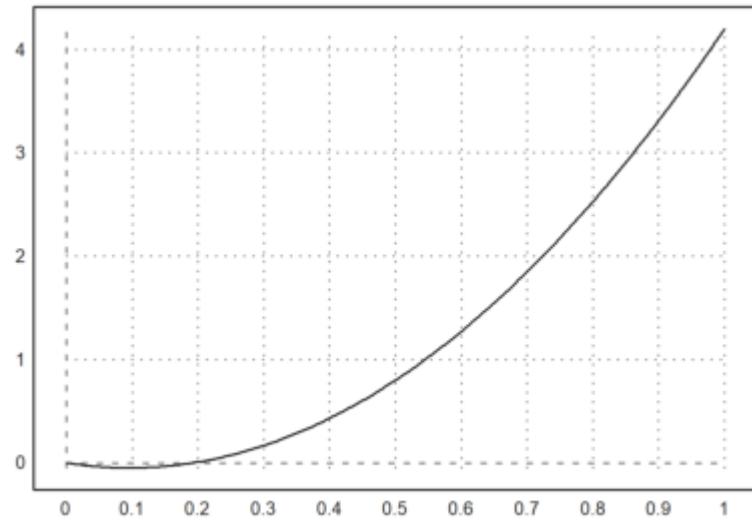
```
function f(x,a) := x2/a + ax2 - x; //define a function  
a=0.3; plot2d("f",0,1;a); // plot with a=0.3
```



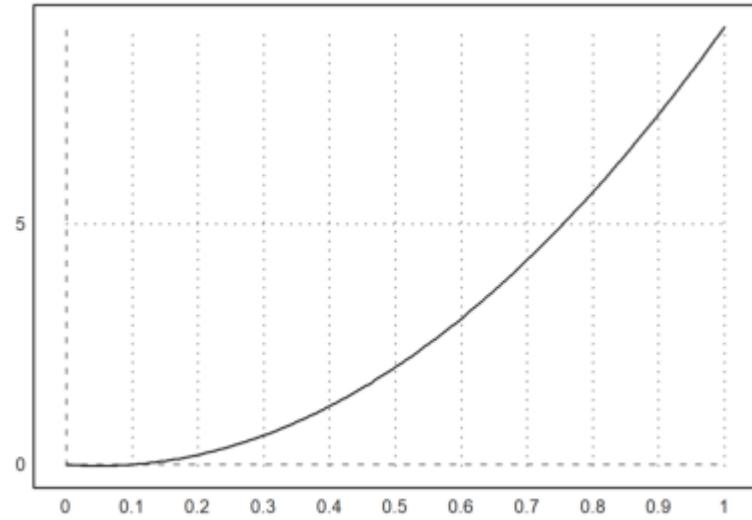
```
plot2d("f",0,1;0.4); // plot with a=0.4
```



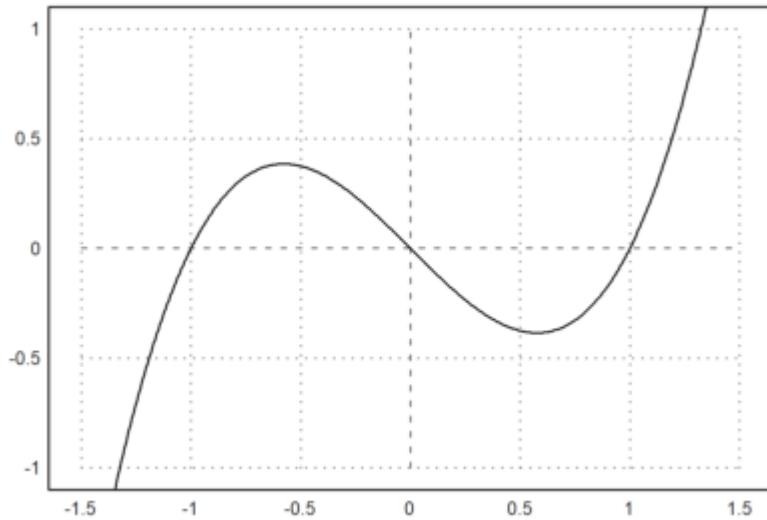
```
plot2d("f",0,1;0.2); // plot with a=0.2
```



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



```
function f(x) := x3 - x; ... plot2d("f",r = 1) :
```

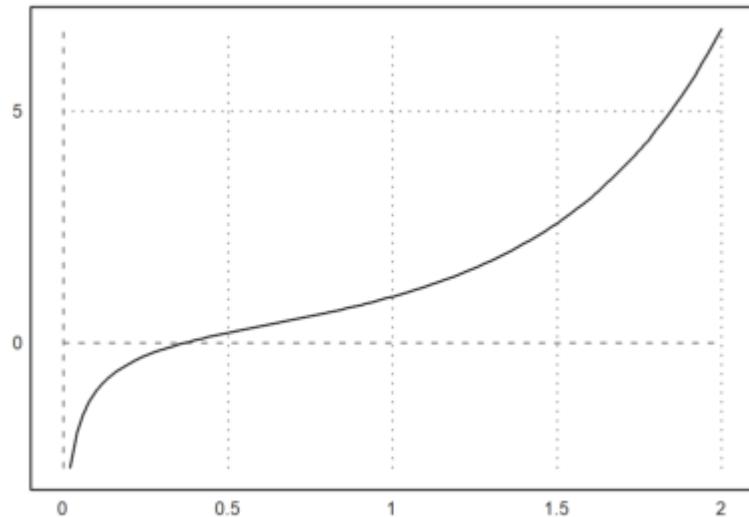


Berikut ini adalah ringkasan fungsi yang diterima

- * ekspresi atau ekspresi simbolik dalam x
- * fungsi atau fungsi simbolik berdasarkan nama seperti "f"
- * fungsi simbolik hanya berdasarkan nama f

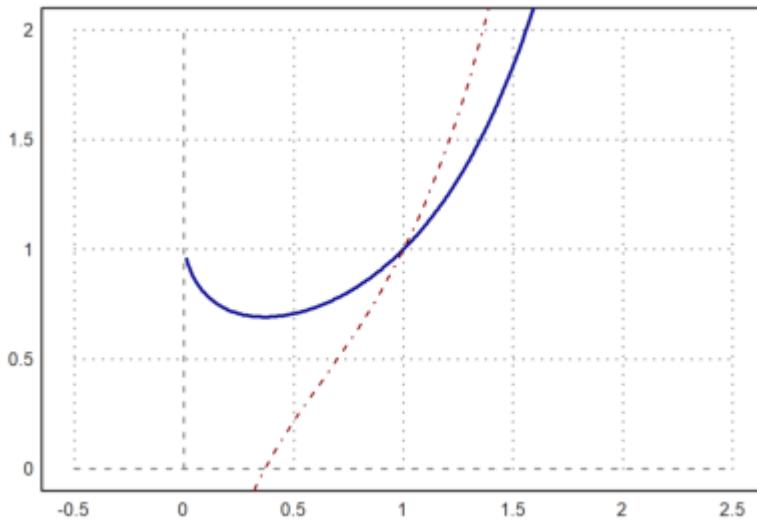
Fungsi `plot2d()` juga menerima fungsi simbolik. Untuk fungsi simbolik, hanya nama yang berfungsi.

```
function f(x) = diff(x^x, x)
x x (log(x) + 1)
plot2d(f,0,2);
```



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

```
expr = sin(x)exp(-x)
- x E sin(x)
plot2d(expr,0,3pi):
function f(x) = xx;
plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
plot2d(diff(f(x),x), add,color=red,style="--"):
```



Untuk gaya garis, ada berbagai pilihan.

* style="...". Pilih dari "-", "--", "-.", ".-", "-.-", "-.-".

* color: Lihat di bawah untuk warna.

thickness: Default adalah 1.

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

* 0..15: indeks warna default.

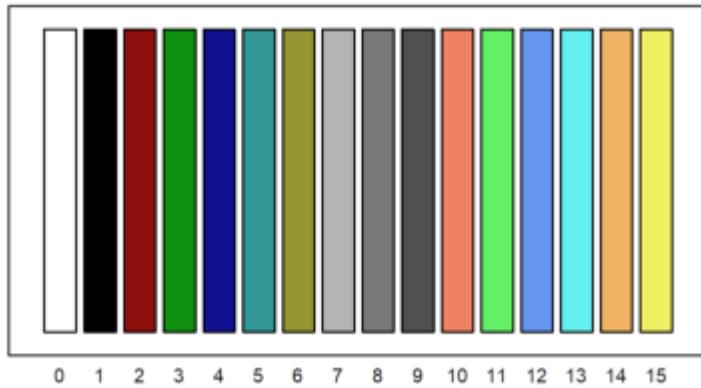
* konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, * abu-abu muda, abu-abu, abu-abu tua, oranye, hijau muda, biru * kehijauan, biru muda, oranye muda, kuning

* rgb(merah,hijau,biru): parameter adalah bilangan real dalam [0,1].

plot2d("exp(-x²)", r = 2, color = red, thickness = 3, style = " -- ") :

Berikut ini tampilan warna EMT yang telah ditetapkan sebelumnya.

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



Namun Anda dapat menggunakan warna apa pun.

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

![images/23030630044_Alfia

Menggambar Beberapa Kurva pada bidang koordinat yang sama

Plotting lebih dari satu fungsi (multifungsi) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu metodenya adalah menggunakan `gt;add` untuk beberapa panggilan ke `plot2d` secara keseluruhan, kecuali panggilan pertama. Kami telah menggunakan fitur ini pada contoh di atas.

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

![images/23030630044_Alfia

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

![images/23030630044_Alfia

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

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

![images/23030630044_Alfia

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

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

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

![images/23030630044_Alfia

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

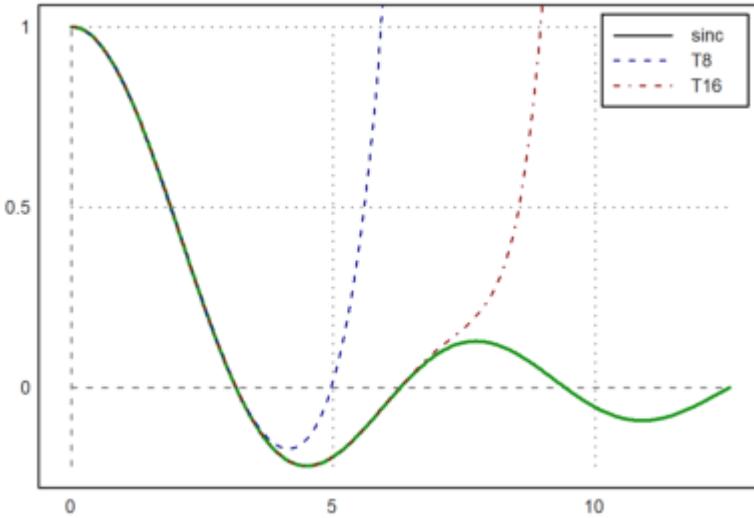
Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke `plot2d()`. Yang kedua dan ketiga memiliki set flag `gt;add`, yang membuat plot menggunakan rentang sebelumnya.

Kami menambahkan 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]):
```



Dalam contoh berikut, kami menghasilkan Polinomial Bernstein.

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

```
plot2d("(1-x)^10", 0, 1); //plot first function
for i=1 to 10; plot2d("bin(10,i)x^i(1-x)^(10-i)", add); end;
insimg;
![images/23030630044_Alfia]
```

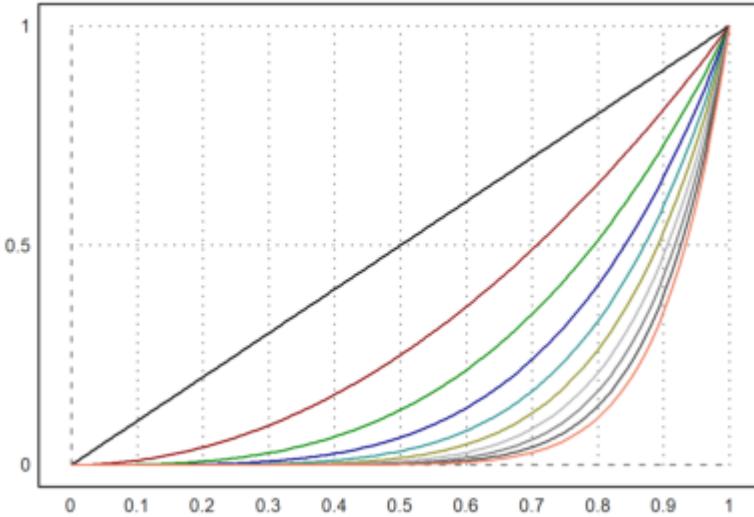
Metode kedua menggunakan sepasang matriks nilai-x dan matriks nilai-y dengan ukuran yang sama.

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

```
x=linspace(0,1,500);
n=10; k=(0:n)'; // n is row vector, k is column vector
y=bin(n,k)x^k(1-x)^(n-k); // y is a matrix then
plot2d(x,y);
![images/23030630044_Alfia]
```

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

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

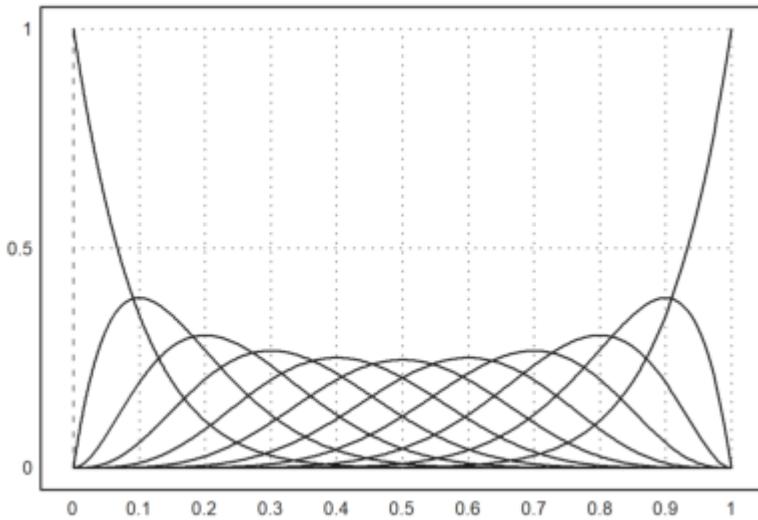


Metode lain adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan array warna, array gaya, dan array ketebalan dengan panjang yang sama.

```
plot2d(["sin(x)","cos(x")],0,2pi,color=4:5);
![images/23030630044_Alfia
plot2d(["sin(x)","cos(x")],0,2pi); // plot vector of expressions
![images/23030630044_Alfia
```

Kita bisa mendapatkan vektor tersebut dari Maxima menggunakan makelist() dan mxm2str().

```
v = makelist(binomial(10,i)x^i(1 - x)^(10 - i), i, 0, 10)//makelist
10 9 8 2 7 3 [(1 - x) , 10 (1 - x) x, 45 (1 - x) x , 120 (1 - x) x , 6 4 5 5 4 6 3
7 210 (1 - x) x , 252 (1 - x) x , 210 (1 - x) x , 120 (1 - x) x , 2 8 9 10 45 (1 - x)
x , 10 (1 - x) x , x ]
mxm2str(v) // get a vector of strings from the symbolic vector
(1-x)^1010*(1-x)^9*x45*(1-x)^8*x^2120*(1-x)^7*x^3210*(1-x)^6*x^4252*
(1-x)^5*x^5210*(1-x)^4*x^6120*(1-x)^3*x^745*(1-x)^2*x^810*(1-x)*x^9x^10
plot2d(mxm2str(v),0,1); // plot functions
```



Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

Jika suatu ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, semua fungsi ini akan diplot menjadi satu plot.

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

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

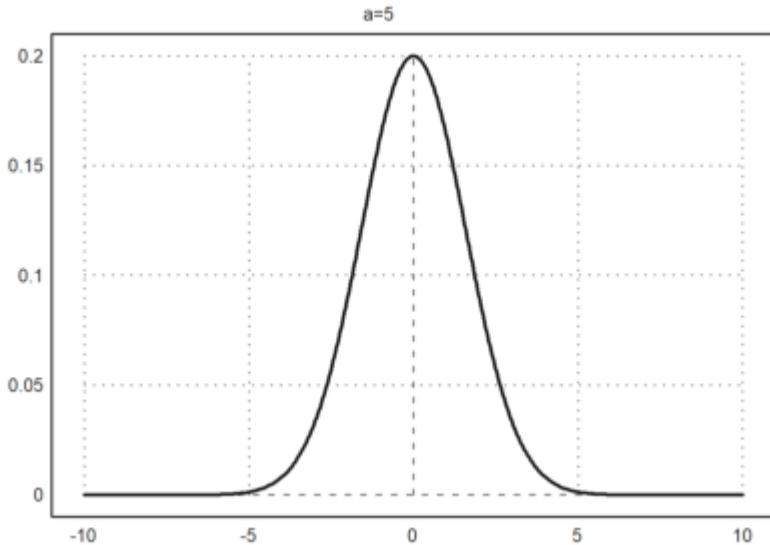
```
![images/23030630044Alifia
```

Ekspresi dan fungsi satu baris dapat melihat variabel global.

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

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

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



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

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

```
plot2d("f",1,-10,10); ... for a=2:10; plot2d("f",a, add); end;
![images/23030630044_Ali_ifia]
```

Kita dapat memperoleh hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris matriks $f(x,a)$ adalah satu fungsi. Selain itu, kita dapat mengatur warna untuk setiap baris matriks. Klik dua kali pada fungsi getspectral() untuk penjelasannya.

```
x=-10:0.01:10; a=(1:10)'; plot2d(x,f(x,a),color=getspectral(a/10));
![images/23030630044_Ali_ifia]
```

Label Teks

Dekorasi sederhana dapat berupa

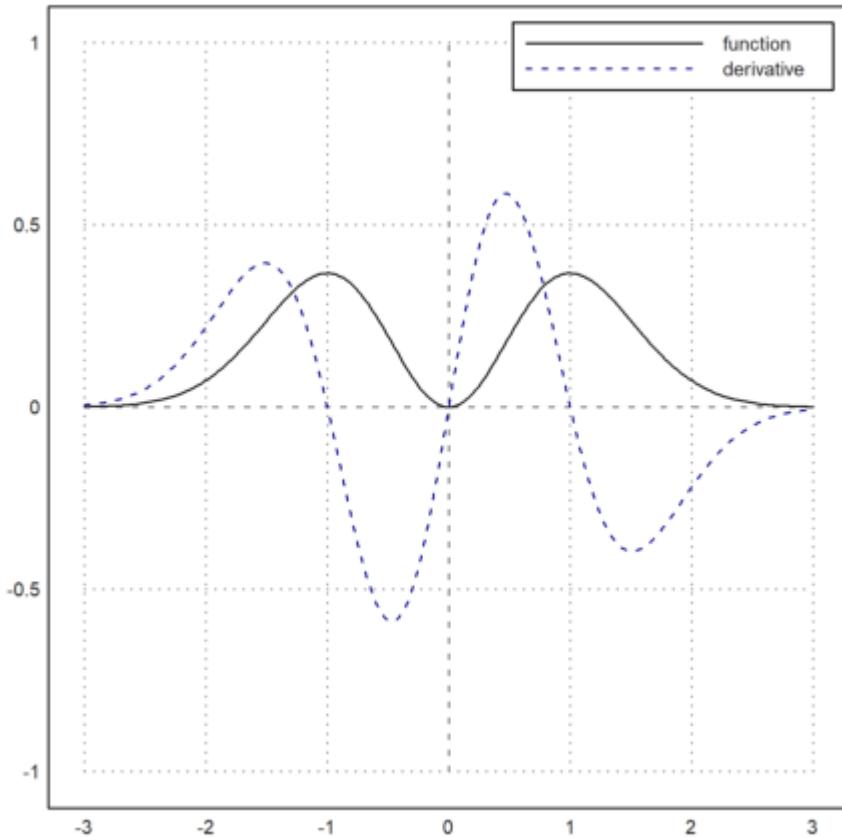
- * judul dengan title="..."
- * label x dan y dengan xl="...", yl="..."
- * label teks lain dengan label("...",x,y)

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

```
plot2d("x^3 - x", -1, 2, title = "y = x^3 - x", yl = "y", xl = "x") :
![images/23030630044_Ali_ifia]
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/23030630044_Ali_ifia]
```

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

```
function f(x) = x2exp(-x2); ... plot2d(f(x),a = -3,b = 3,c = -1,d = 1); ...plot2d(diff(f(x),x),add,color = blue,style = "—"); ...labelbox(["function","derivative"],styles = ["—","—"],... colors = [black,blue],w = 0.4) :
```



Kotak tersebut ditambahkan di kanan atas secara default, tetapi `gt:left` menambahkannya di kiri atas. Anda dapat memindahkannya ke tempat mana pun yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angka-angkanya adalah pecahan dari ukuran jendela grafik. Lebarnya otomatis.

Untuk plot titik, kotak label juga berfungsi. Tambahkan parameter `gt;points`, atau vektor bendera, satu untuk setiap label.

Dalam contoh berikut, hanya ada satu fungsi. Jadi, kita dapat menggunakan string alih-alih vektor string. Kita tetapkan 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/23030630044_Alfia
```

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

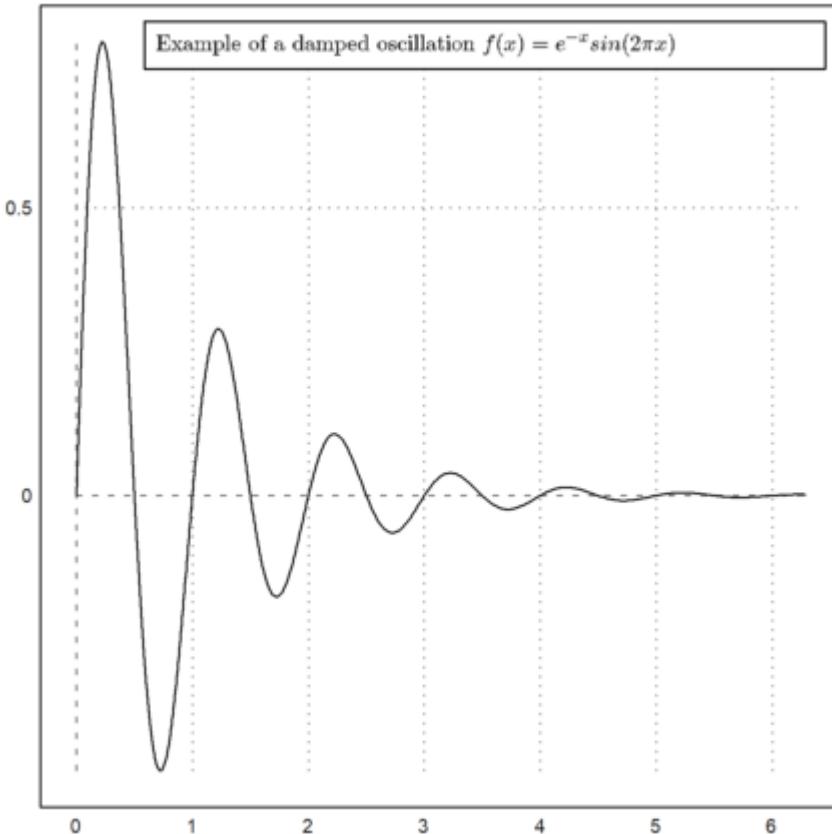
```
statplot(1:10,random(2,10),color=[red,blue]):
```

```
![images/23030630044_Alfia
```

Fitur serupa adalah fungsi textbox().

Lebar secara default adalah lebar maksimal baris teks. Namun, pengguna juga dapat mengurnanya.

```
function f(x) = exp(-x)sin(2pix); ... plot2d("f(x)",0,2pi); ... textbox(latex("textExample of a damped oscillation  
f(x)=e-xsin(2  
pix)'),w = 0.85) :
```



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

```
plot2d("x3 - x",title = "xrarr;x3-x") :
```

```
![images/23030630044_Alfia
```

Label pada sumbu x dan y dapat vertikal, begitu pula sumbunya.

```
plot2d("sinc(x)",0,2pi, xl="x",yl=u"x rarr; sinc(x)", vertical);
![images/23030630044Alifia
```

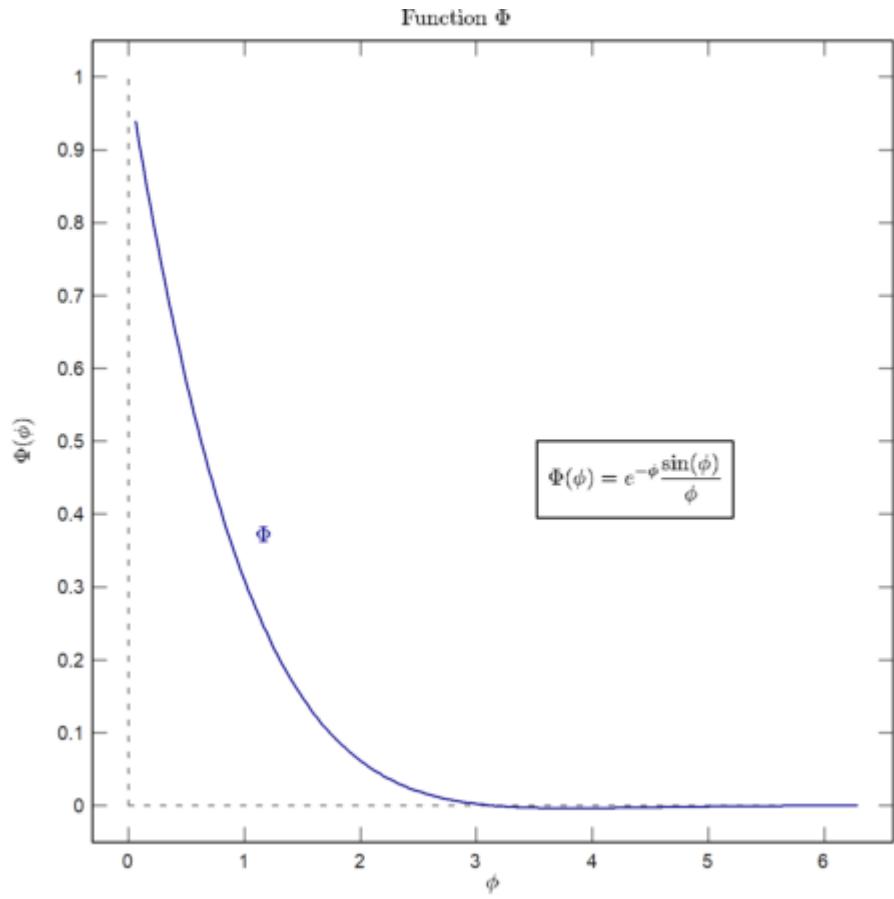
2.7 LaTex

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

Perlu dicatat, bahwa penguraian LaTeX lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil latex() sebelum loop sekali dan menggunakan hasilnya (gambar dalam matriks RGB).

Dalam plot berikut, kami menggunakan LaTeX untuk label x dan y, label, kotak label, dan judul plot.

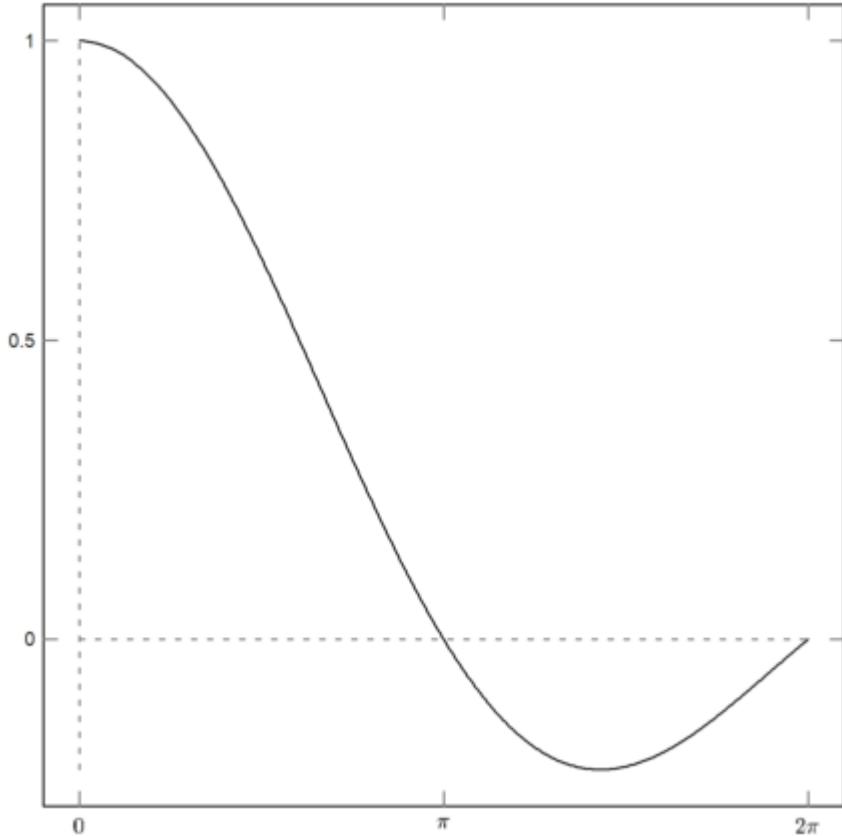
```
plot2d("exp(-x)sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ... title=latex("textFunction Phi"), ... xl=latex("phi"),yl=latex("Phi(phi)"); ... textbox( ... latex("Phi(phi) = e^{-phi} frac{sin(phi)}{phi}, x = 0.8, y = 0.5); ... label(latex("Phi", color = blue), 1, 0.4) :
```



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

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

```
plot2d("sinc(x)",0,2pi,grid=4,xticks); ... ygrid(-2:0.5:2,grid=6); ... xgrid([0:2] × pi,xticks,grid=6); ... xtick([0,pi,2pi],["0","pi","2pi"], latex);
```



Tentu saja, fungsi juga dapat digunakan.

```
function map f(x) ...
if x<0 then return x4else return x2endifendfunction
```

Parameter "peta" membantu menggunakan fungsi $f(x)$. Jika $x = -1$, $f(-1) = 1$; jika $x = 0$, $f(0) = 0$; dan jika $x = 1$, $f(1) = 1$.

Dalam plot berikut, kami juga memasukkan beberapa kode LaTeX. Kami menggunakanannya untuk dua label dan 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("x3"),0.72,f(0.72)); ...
label(latex("x2"),-0.52,f(-0.52),pos =
"ll");
... textbox(... latex("f(x) =
begin{cases} x^3 & x \leq 0 \\ x^2 & x > 0 \end{cases}"))
... x = 0.7, y = 0.2) :
```

Variable f not found! Use global variables or parameters for string evaluation.
Error in expression: $f y0=f(x[1], args())$; adaptiveevalone : $s =$ Try "traceerrors" to inspect local variables after evalone.

```
 $dw/n, dw/n^2, dw/n, auto; args());$ 
```

2.8 Interaksi Pengguna

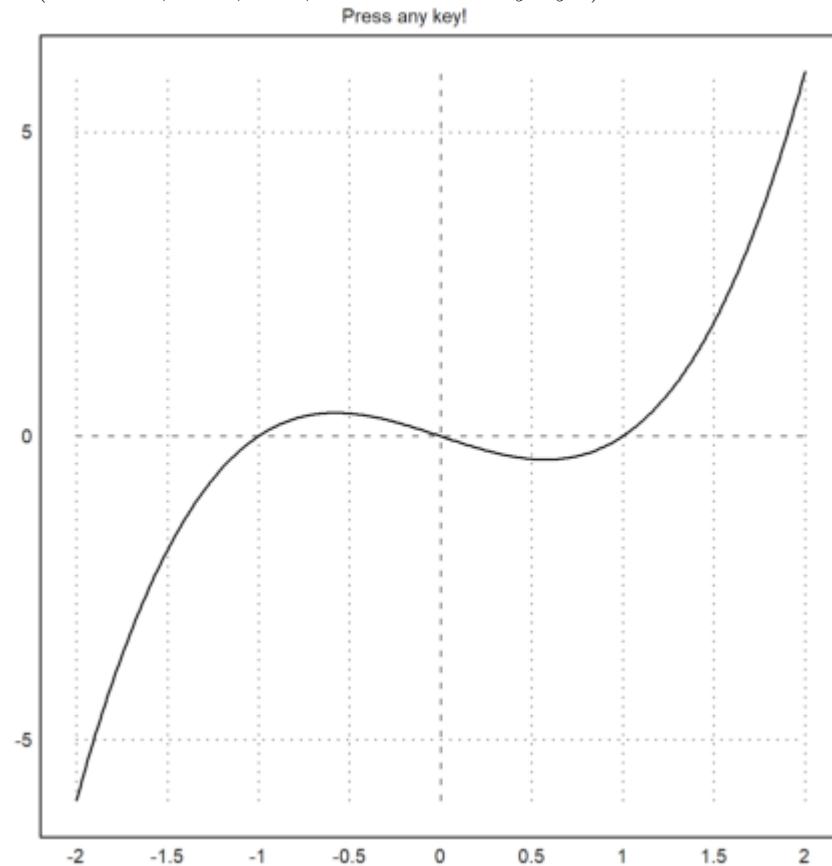
Saat memplot fungsi atau ekspresi, parameter `gt;user` memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kurSOR atau tetikus. Pengguna dapat

- * memperbesar dengan + atau -
- * memindahkan plot dengan tombol kurSOR
- * memilih jendela plot dengan tetikus
- * mengatur ulang tampilan dengan spasi
- * keluar dengan kembali

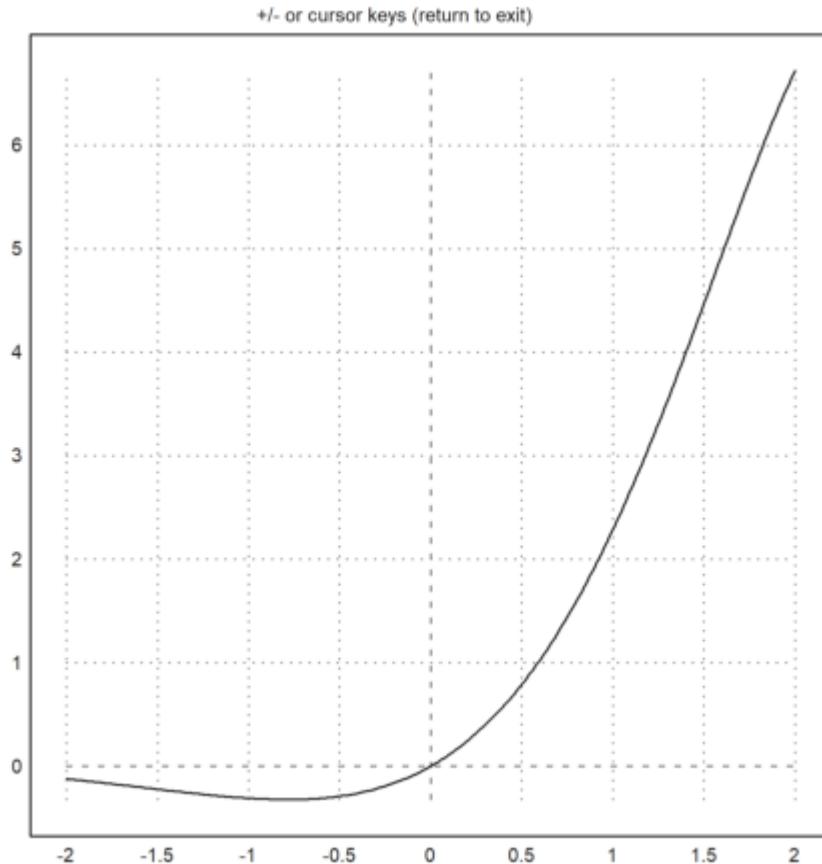
Tombol spasi akan mengatur ulang plot ke jendela plot asli.

Saat memplot data, tanda `gt;user` akan menunggu penekanan tombol.

```
plot2d("x3 - ax", a = 1, user, title = "Press anykey"):
```



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



The following demonstrates an advanced way of user interaction (see the tutorial about programming for details).

The built-in function `mousedown()` waits for mouse or keyboard events. It reports mouse down, mouse moved or mouse up, and key presses. The function `dragpoints()` makes use of this, and lets the user drag any point in a plot.

We need a plot function first. For an example, we interpolate in 5 points with a polynomial. The function should plot into a fixed plot area.

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

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

Sekarang kita buat beberapa nilai acak, dan biarkan pengguna menyeret titik-titiknya.

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

Variable plotf not found! Use global variables or parameters for string evaluation. Error in expression: plotf Try "trace errors" to inspect local variables after errors. dragpoints: f(x,y,select,args());

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

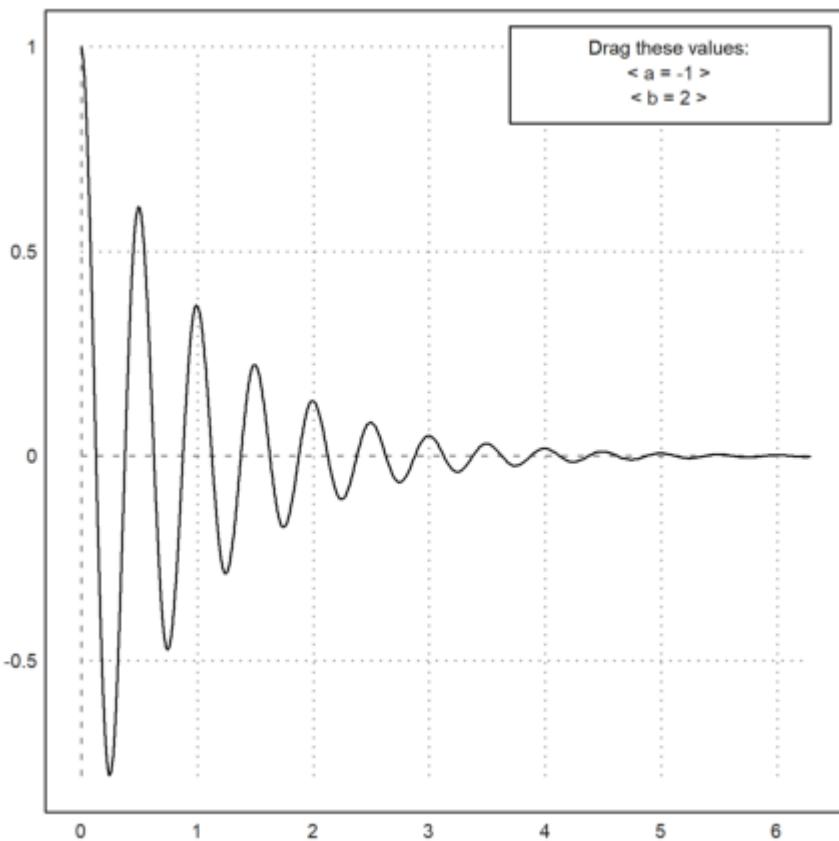
Pertama, kita perlu fungsi plot.

```
function plotf([a,b]) := plot2d("exp(ax)cos(2pix)",0,2pi;a,b);
```

Kemudian kita perlu nama untuk parameter, nilai awal, dan matriks rentang nx2, secara opsional baris judul.

Ada slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

```
dragvalues("plotf","a","b],[-1,2],[-2,2];[1,10], ... heading="Drag these values:",hcolor=black);
```



Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Misalnya, kita menulis fungsi plot, yang memplot polinomial Taylor berderajat n ke fungsi kosinus.

```
function plotf(n) ...
```

```

plot2d("cos(x)",0,2pi,;square,grid=6); plot2d("taylor(cos(x),x,0,@n)",color=blue,;add);
textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",;left); endfunction
;j/pre;
Sekarang kita biarkan derajat n bervariasi dari 0 hingga 20 dalam 20
stop. Hasil dragvalues() digunakan untuk memplot sketsa dengan n ini, dan
untuk memasukkan plot ke dalam buku catatan.

```

```

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

```

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

```
function dragtest ...

```

```

plot2d(null,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
;j/pre;
dragtest // lihat hasilnya dan cobalah lakukan!

```

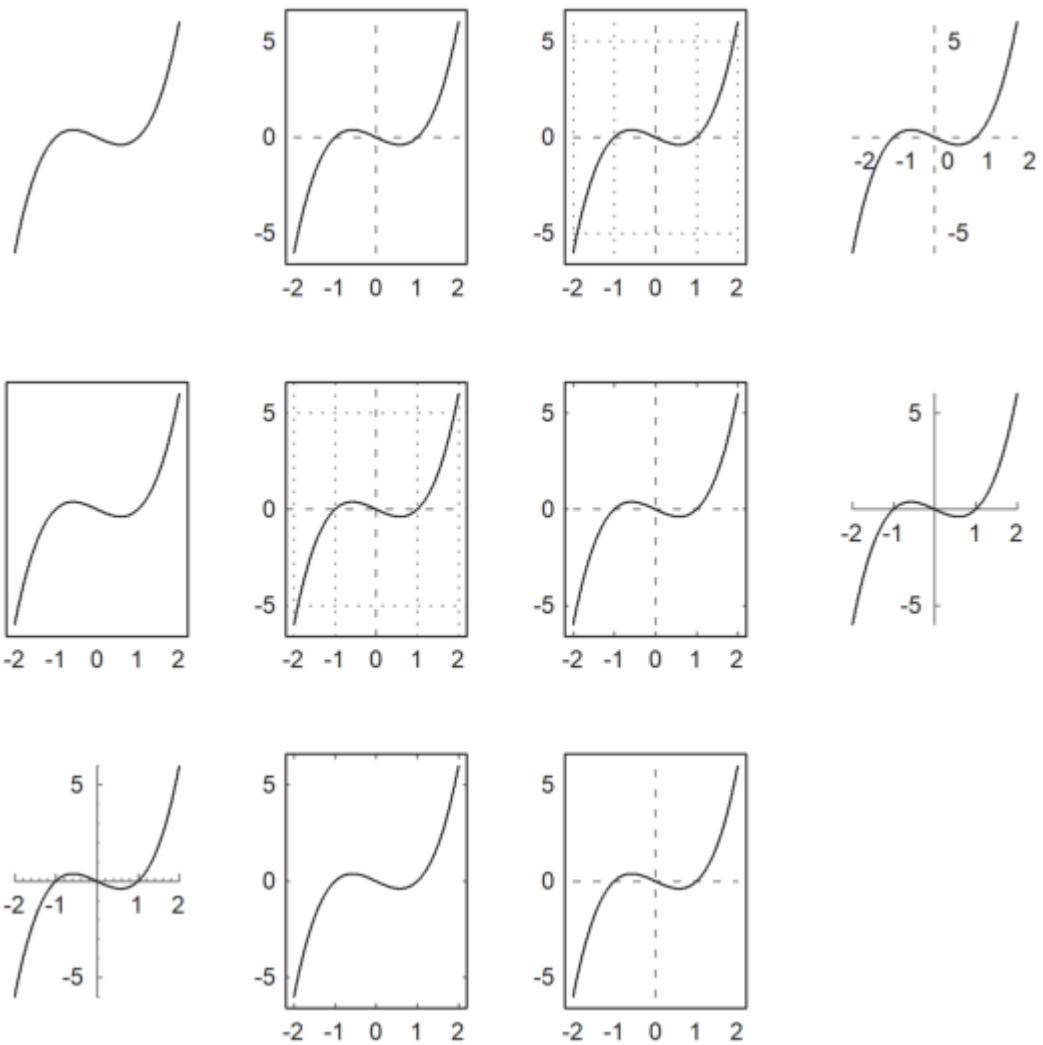
Gaya Plot 2D

Secara default, EMT menghitung tanda centang sumbu otomatis dan menambahkan label ke setiap tanda centang. 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); ...//nogrid, frameoraxis
figure(2); plot2d("x^3 - x",grid = 1); ...//x - y - axis
figure(3); plot2d("x^3 - x",grid = 2); ...//defaultticks
figure(4); plot2d("x^3 - x",grid = 3); ...//x - y - axiswithlabelsinside
figure(5); plot2d("x^3 - x",grid = 4); ...//noticks, onlylabels
figure(6); plot2d("x^3 - x",grid = 5); ...//default, butnomargin
figure(7); plot2d("x^3 - x",grid = 6); ...//axesonly
figure(8); plot2d("x^3 - x",grid = 7); ...//axesonly, ticksataxis
figure(9); plot2d("x^3 - x",grid = 8); ...//axesonly, finerticksataxis
figure(10); plot2d("x^3 - x",grid = 9); ...//default, smallticksinside
figure(11); plot2d("x^3 - x",grid = 10); ...//noticks, axesonly
figure(0):

```



Parameter `lt:frame` menonaktifkan bingkai, dan `framecolor=blue` menyetel bingkai ke warna biru.

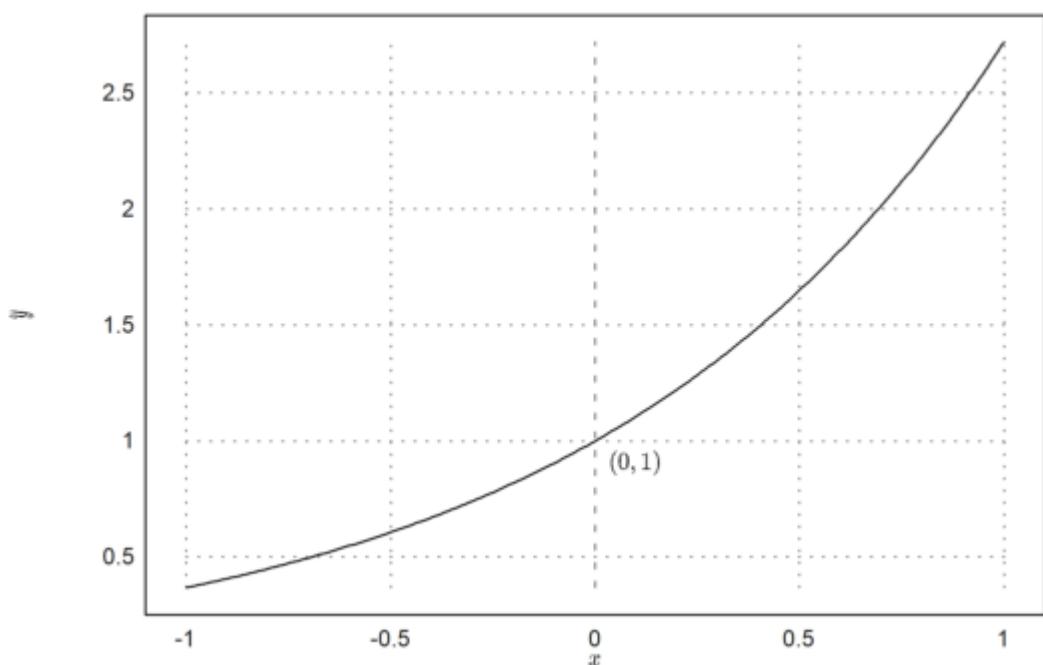
Jika Anda menginginkan tanda centang Anda sendiri, Anda dapat menggunakan `style=0`, dan menambahkan semuanya nanti.

```
aspect(1.5);
plot2d("x^3 - x",grid = 0); //plot
frame; xgrid([-1,0,1]); ygrid(0); // add frame and grid
![images/23030630044_Alfia]
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/23030630044_Alfia

Sumbu dapat digambar secara terpisah dengan xaxis() dan yaxis().

```

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

```

![images/23030630044_Alfia

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

```

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

```

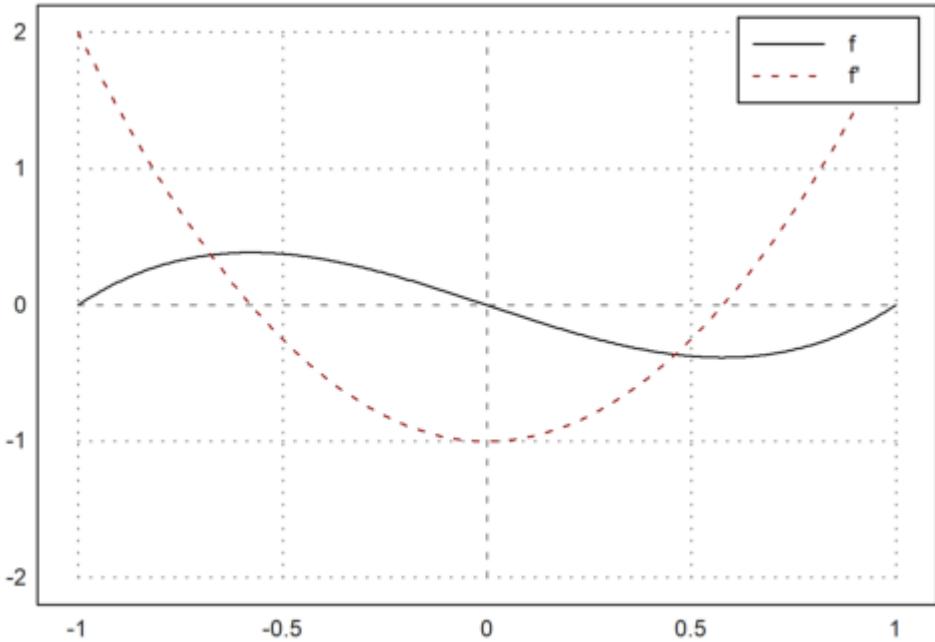
![images/23030630044_Alfia

Ada juga kotak teks.

```

plot2d(f(x),-1,1,-2,2); // function
plot2d(diff(f(x),x),add,style="--",color=red); // derivative
labelbox(["f","f'"],["-","-"],[black,red]); // label box

```



```

plot2d(["exp(x)","1+x"],color=[black,blue],style=["-","-.-"]):
![images/23030630044Alifia
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/23030630044Alifia

```

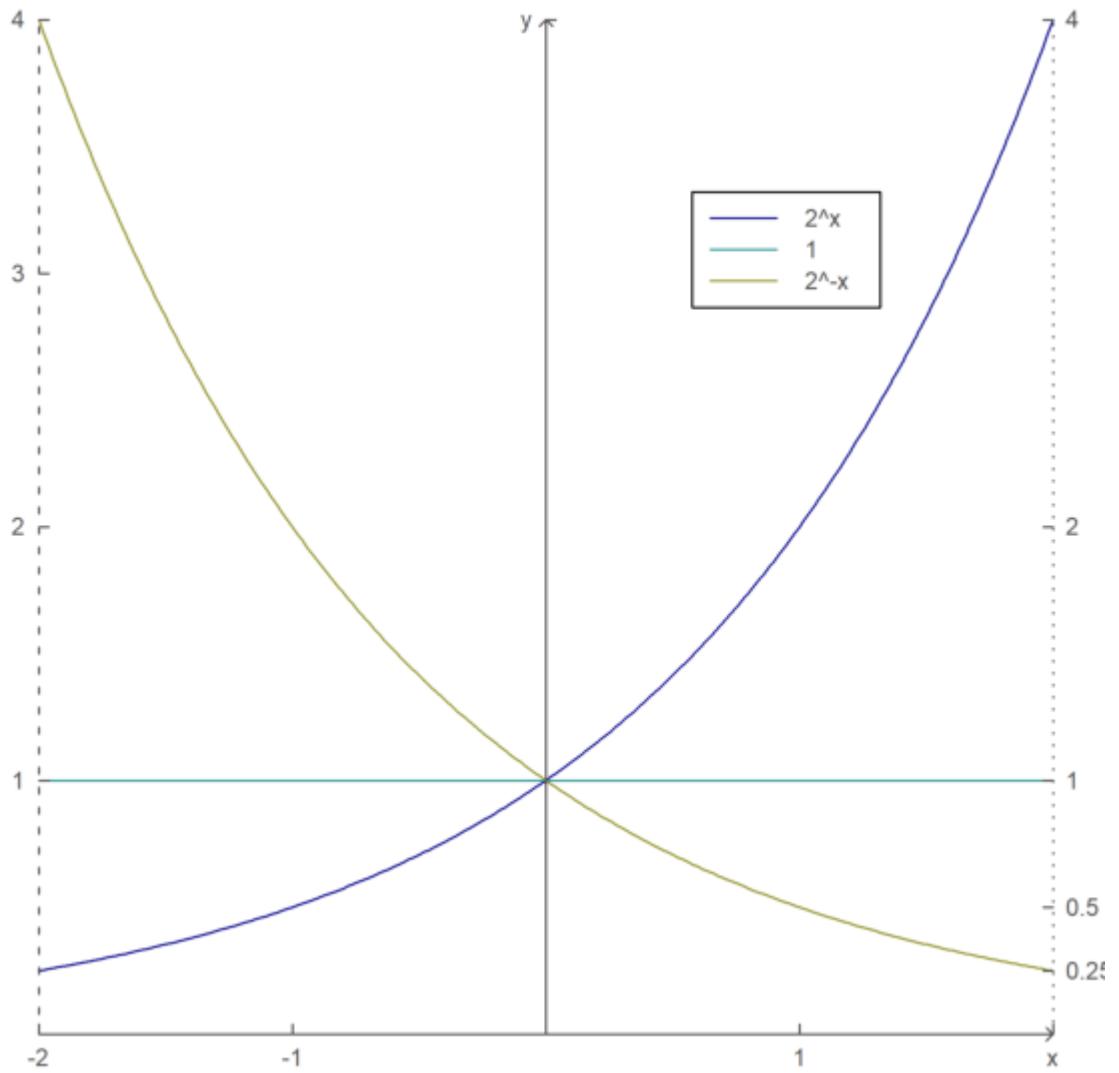
Untuk kontrol yang lebih baik, sumbu x dan sumbu y dapat dilakukan secara manual.

Perintah fullwindow() memperluas jendela plot karena kita tidak lagi memerlukan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk 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/23030630044_Alfia

Berikut contoh lain, di mana string Unicode digunakan dan sumbu berada di luar area plot.

```
aspect(1.5);
plot2d(["sin(x)","cos(x)],0,2pi,color=[red,green],grid,frame); ... xaxis(-1.1,(0:2)pi,xt=["0",u"\u03c0;","2\u03c0;"],style="-", ticks, zero); ... xgrid((0:0.5:2) \times pi,ticks); ... yaxis(-0.1pi,-1:0.2:1,style="-", zero, grid); ... labelbox(["sin","cos"],colors=[red,green],x=0.5,y=0.5);
xlabel(u"\u03c6"); ylabel(u"f(\u03c6)");
```

![images/23030630044_Alfia

Plotting Data 2D

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

Plotting ekspresi hanyalah singkatan untuk plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai- x , dan satu atau beberapa baris nilai- y . Dari rentang dan nilai- x , fungsi `plot2d` akan menghitung data yang akan diplot, secara default dengan evaluasi adaptif fungsi. Untuk plot titik gunakan "gt;points", untuk garis dan titik campuran gunakan "gt;addpoints".

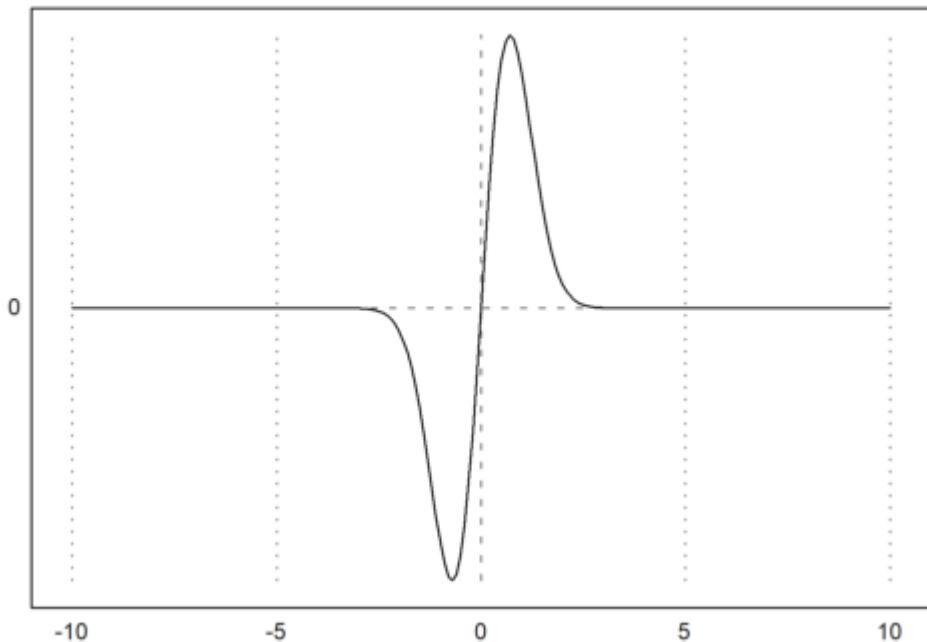
Namun, Anda dapat memasukkan data secara langsung.

- * 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/23030630044_Alfia

`ipre class="udf" i/prei` Data juga dapat diplot sebagai titik. Gunakan `points=true` untuk ini. Plot bekerja seperti poligon, tetapi hanya menggambar sudutnya.

`* style="...": Pilih dari "[]", "lt;gt;", "o", ".", "..", "+", "*", "[]", * "lt;gt;", "o", "..", "", "-".`

Untuk memplot kumpulan titik, gunakan `gt;points`. Jika warnanya adalah vektor warna, setiap titik

mendapatkan warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna berlaku untuk baris matriks.

Parameter `gt;addpoints` menambahkan titik ke segmen garis untuk plot data.
`xdata=[1,1.5,2.5,3,4]; ydata=[3,3.1,2.8,2.9,2.7]; // data`
`plot2d(xdata,ydata,a=0.5,b=4.5,c=2.5,d=3.5,style="."); // lines`
`plot2d(xdata,ydata, points, add,style="o"); // add points`
`p=polyfit(xdata,ydata,1); // get regression line`
`plot2d("polyval(p,x)", add,color=red); // add plot of line`
Variable or function `p` not found. Error in expression: `polyval(p,x) y0=f(x[1],args()); adaptiveevalone : s = Try "traceerrors" to inspect local variables after errors. plot2d : dw/n, dw/n^2, dw/n, auto; args()`

Menggambar Daerah Yang Berbatasan Kurva

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

- * `terisi=benar` mengisi plot.
- * `style="..."`: Pilih dari "", "/", ";" .
- * `fillcolor`: Lihat di atas untuk warna yang tersedia.

Warna isian ditentukan oleh argumen "`fillcolor`", dan pada `lt;outline` opsional mencegah penggambaran batas untuk semua gaya kecuali yang default.

```
t=linspace(0,2pi,1000); // parameter for curve
x=sin(t)exp(t/pi); y=cos(t)exp(t/pi); // x(t) and y(t)
figure(1,2); aspect(16/9)
figure(1); plot2d(x,y,r=10); // plot curve
figure(2); plot2d(x,y,r=10, filled,style="/",fillcolor=red); // fill curve
figure(0);
![images/23030630044_Alfia]
```

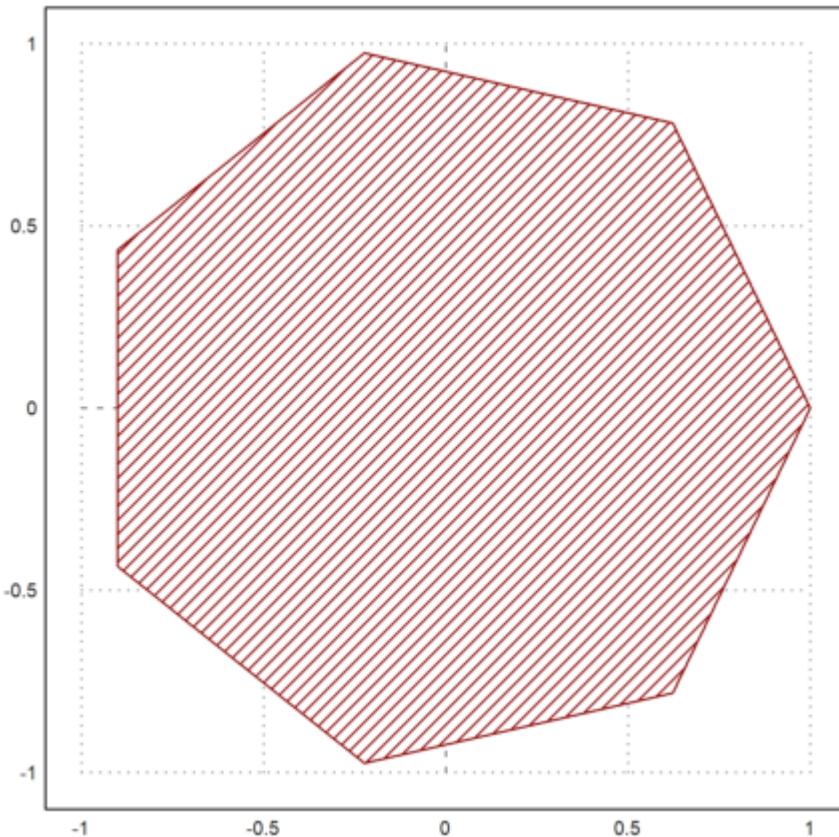
Pada contoh berikut ini kami memplot elips yang terisi dan dua segi enam yang terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian yang berbeda.

```
dx=linspace(0,2pi,1000); plot2d(sin(x),cos(x)0.5,r=1, filled,style="/");
insimg;
![images/23030630044_Alfia]
The : allows only real arguments! Error in: ... (sin(x),cos(x)*0.5,r=1,gt;filled,style="/");
insimg; ...
t=linspace(0,2pi,6); ... plot2d(cos(t),sin(t), filled,style="/",fillcolor=red,r=1.2);
![images/23030630044_Alfia]
```

```
t=linspace(0,2pi,6); plot2d(cos(t),sin(t), filled,style="");
![images/23030630044_Alfia]
```

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

```
t=linspace(0,2pi,7); ... plot2d(cos(t),sin(t),r=1, filled,style="/",fillcolor=red);
```



Berikut ini adalah himpunan nilai maksimal dari empat kondisi linier yang kurang dari atau sama dengan 3. Ini adalah $A[k].vlt;=3$ untuk semua baris A. Untuk mendapatkan sudut yang bagus, kita menggunakan n yang relatif besar.

```
A=[2,1;1,2;-1,0;0,-1];
function f(x,y) := max([x,y].A');
plot2d("f",r=4,level=[0;3],color=green,n=111);
![images/23030630044_Ali
```

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

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

Sekarang kita memiliki vektor x dan y dari nilai-nilai. `plot2d()` dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plot dapat diisi. Dalam hal ini menghasilkan hasil yang bagus karena aturan lilitan, yang digunakan untuk pengisian.

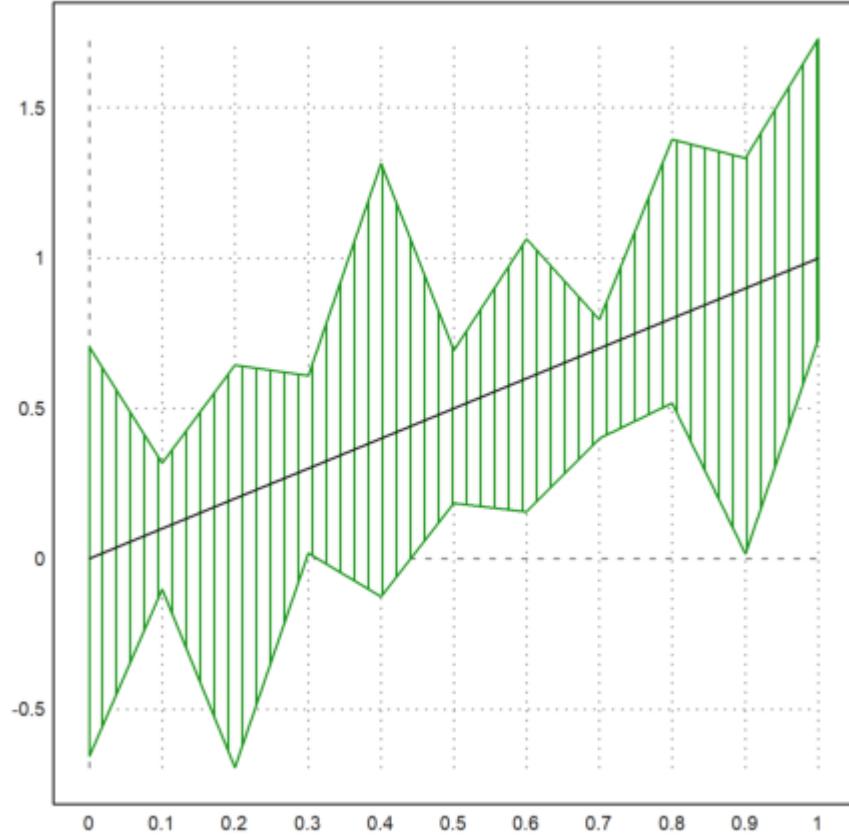
```
plot2d(x,y,jgrid,jframe,filled);
![images/23030630044_Ali
```

Vektor interval diplot terhadap nilai x sebagai daerah terisi antara nilai

interval yang lebih rendah dan lebih tinggi.

Hal ini dapat berguna untuk memplot kesalahan perhitungan. Namun, hal ini juga dapat digunakan untuk memplot kesalahan statistik.

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



Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka `plot2d` akan memplot rentang interval yang terisi pada bidang. Gaya isianya sama dengan gaya poligon.

```
t=-1:0.01:1; x= t-0.01,t+0.01 ; y=x^3 - x;
```

```
plot2d(t,y);
```

```
![images/23030630044_Alfia]
```

Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks $2 \times n$. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

```
expr := "2x^2 + xy + 3y^4 + y"; //define an expression f(x, y)
```

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

```
![images/23030630044_Alfia]
```

Kita juga dapat mengisi rentang nilai seperti

$$-1 \leq (x^2 + y^2)^2 - x^2 + y^2 \leq 0.$$

```
plot2d("(x^2 + y^2)^2 - x^2 + y^2", r = 1.2, level = [-1;0], style = "/") :  
![images/23030630044_Alfia  
plot2d("cos(x)", "sin(x)^3", xmin = 0, xmax = 2pi, filled, style = "/") :  
![images/23030630044_AlfiaRahmawati_M_E23Plot2D20(1)-087.png](images/23030630044_Alfia
```

2.9 Grafik Fungsi Parametrik

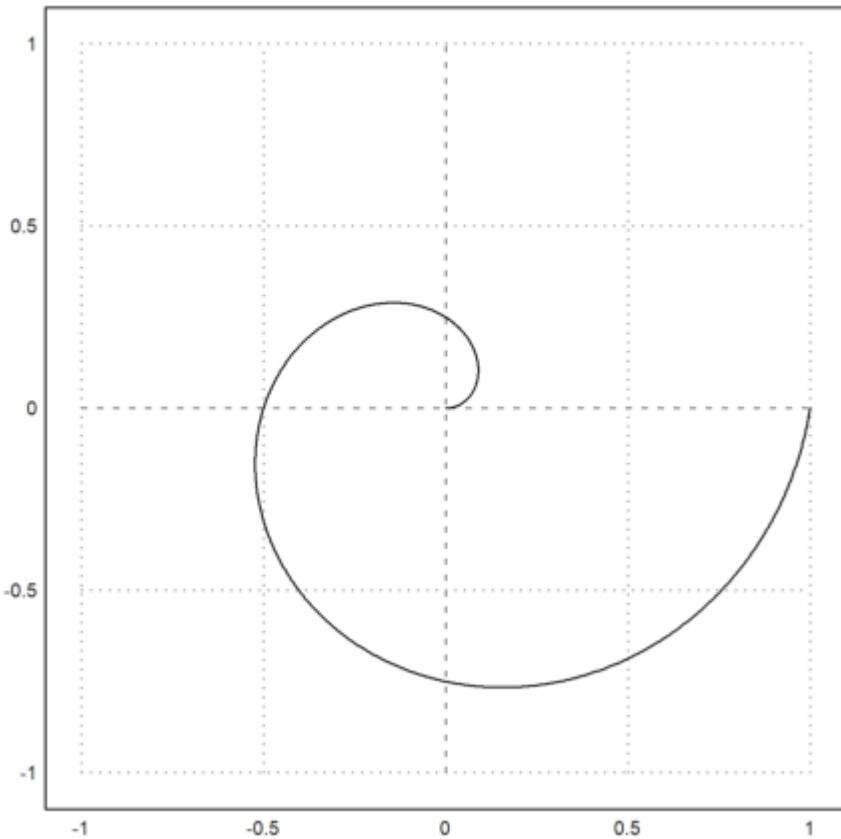
Nilai-nilai x tidak perlu diurutkan. (x,y) hanya menggambarkan kurva. Jika x diurutkan, kurva tersebut adalah grafik fungsi.

Dalam contoh berikut, kita memplot spiral

$$\gamma(t) = t \cdot (\cos(2\pi t), \sin(2\pi t))$$

Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk detail lebih lanjut).

```
t=linspace(0,1,1000); ... plot2d(tcos(2pit),tsin(2pit),r=1):
```



Atau, ada kemungkinan untuk menggunakan dua ekspresi untuk kurva.
Berikut ini memplot kurva yang sama seperti di atas.

```
plot2d("x*cos(2pix)","xsin(2pix)",xmin=0,xmax=1,r=1);
![images/23030630044Alifia
```

```
t=linspace(0,1,1000); r=exp(-t); x=rcos(2pit); y=rsin(2pit);
plot2d(x,y,r=1);
![images/23030630044Alifia
```

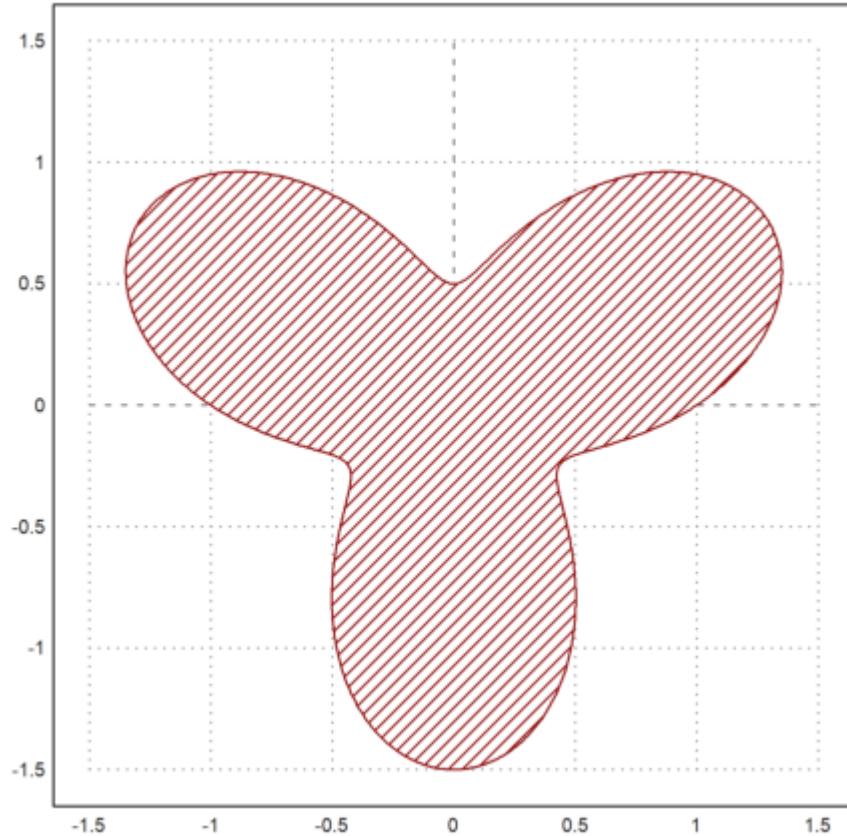
Pada contoh berikut, kita plot kurva

$$\gamma(t) = (r(t) \cos(t), r(t) \sin(t))$$

dengan

$$r(t) = 1 + \sin(3t)/2.$$

```
t=linspace(0,2pi,1000); r=1+sin(3t)/2; x=rcos(t); y=rsin(t); ... plot2d(x,y,filled,fillcolor=red,style="/"',r=1);
```



Menggambar Grafik Bilangan Kompleks

Susunan bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan dihubungkan. Jika sejumlah garis grid ditentukan (atau vektor garis grid 1×2) dalam argumen cgrid, hanya garis grid tersebut yang terlihat.

Matriks bilangan kompleks akan secara otomatis diplot sebagai grid dalam bidang kompleks.

Dalam contoh berikut, kami memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

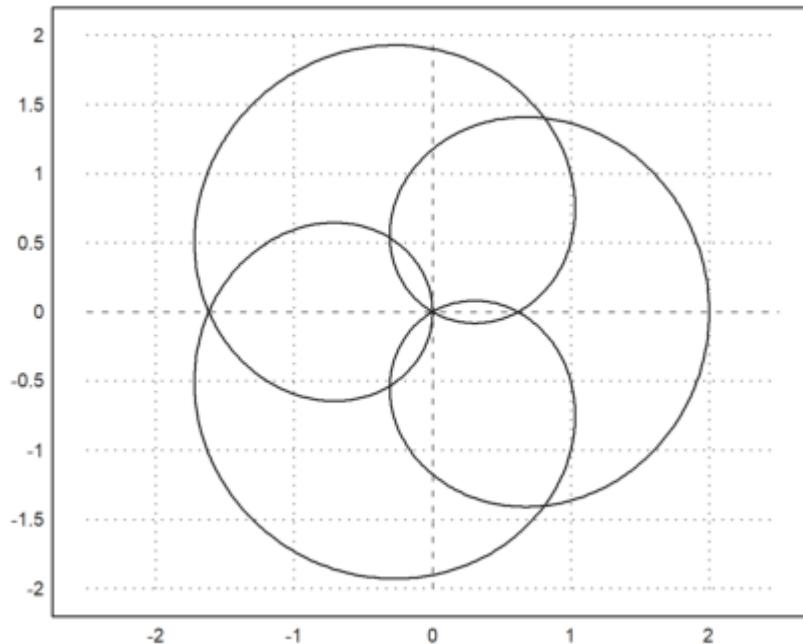
```
aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=rexp(Ia);... plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10);
![images/23030630044_Alifia
aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=rexp(Ia);
plot2d(exp(z),cgrid=[40,10]);
![images/23030630044_Alifia
r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=rexp(Ia);
plot2d(exp(z), points, add);
![images/23030630044_Alifia
```

Vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian riil dan bagian imajiner.

Dalam contoh, kita memplot lingkaran satuan dengan

$$\gamma(t) = e^{it}$$

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



2.10 Plot Statistik

Ada banyak fungsi yang dikhkususkan pada plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

Penjumlahan kumulatif dari nilai-nilai berdistribusi normal 0-1 menghasilkan pergerakan acak.

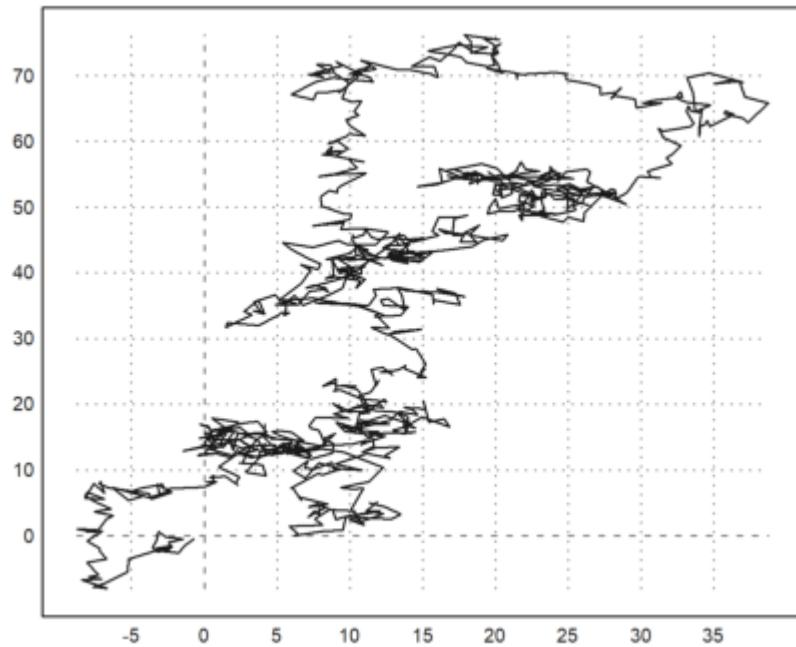
```
plot2d(cumsum(randnormal(1,1000))):
```



![images/23030630044_Alifia

Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

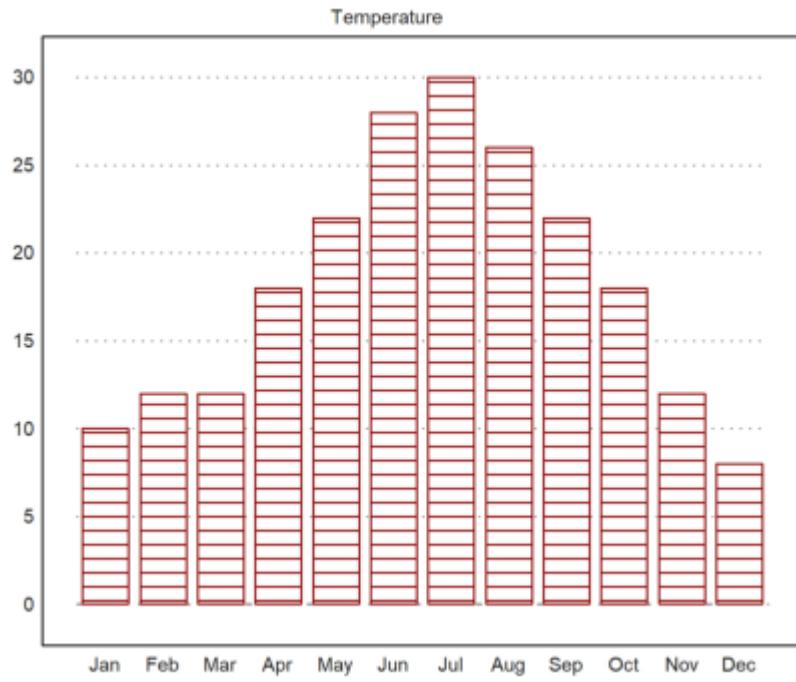
X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):



```

columnsplot(cumsum(random(10)),style="/" ,color=blue);
![images/23030630044_Alfia
Ia juga dapat menampilkan string sebagai label.
months=["Jan","Feb","Mar","Apr","May","Jun", ... "Jul","Aug","Sep","Oct","Nov","Dec"];
values=[10,12,12,18,22,28,30,26,22,18,12,8];
columnsplot(values,lab=months,color=red,style="-");
title("Temperature");

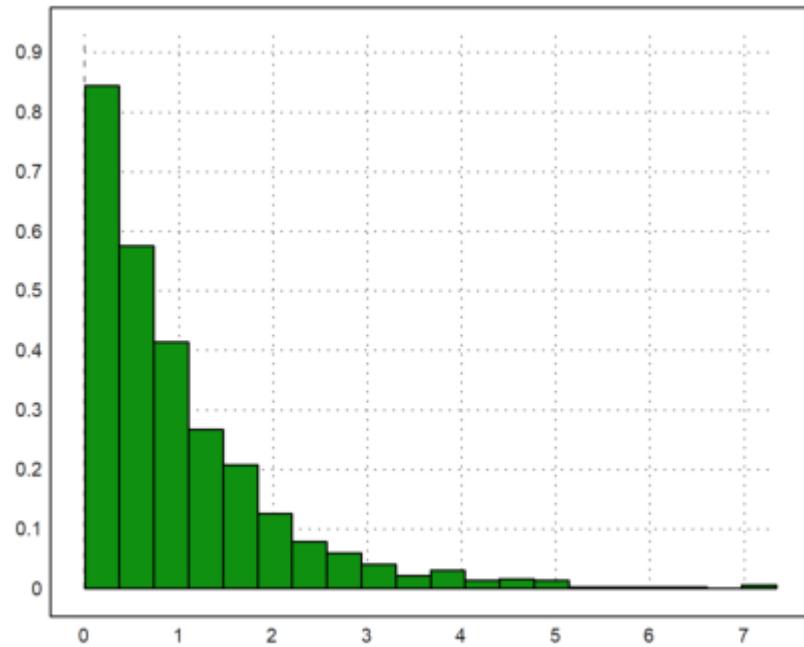
```



```

k=0:10;
plot2d(k,bin(10,k), bar):
![images/23030630044_Alfia
plot2d(k,bin(10,k)); plot2d(k,bin(10,k), points, add):
![images/23030630044_Alfia
plot2d(normal(1000),normal(1000), points,grid=6,style=".."):
![images/23030630044_Alfia
plot2d(normal(1,1000), distribution,style="O"):
![images/23030630044_Alfia
plot2d("qnormal",0,5;2.5,0.5, filled):
![images/23030630044_Alfia
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan
distribution=n dengan plot2d.
w=randexponential(1,1000); // exponential distribution
plot2d(w, distribution): // or distribution=n with n intervals

```



Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan `gt;bar` di `plot3d`, atau dengan plot kolom.

```
w=normal(1000); // 0-1-normal distribution
x,y=histo(w,10,v=[-6,-4,-2,-1,0,1,2,4,6]); // interval bounds v
plot2d(x,y, bar):
![images/23030630044_Alfia]
Fungsi statplot() mengatur gaya dengan string sederhana.
statplot(1:10,cumsum(random(10)),"b"):
![images/23030630044_Alfia]
n=10; i=0:n; ... plot2d(i,bin(n,i)/2^n, a = 0, b = 10, c = 0, d = 0.3); ...plot2d(i, bin(n, i)/2^n, points =
true, style = "ow", add = true, color = blue) :
![images/23030630044_Alfia]
```

Selain itu, data dapat diplot sebagai batang. Dalam kasus ini, x harus diurutkan dan satu elemen lebih panjang dari y. Batang akan memanjang dari $x[i]$ ke $x[i+1]$ dengan nilai $y[i]$. Jika x memiliki ukuran yang sama dengan y, maka akan memanjang satu elemen dengan spasi terakhir.

Gaya isian dapat digunakan seperti di atas.

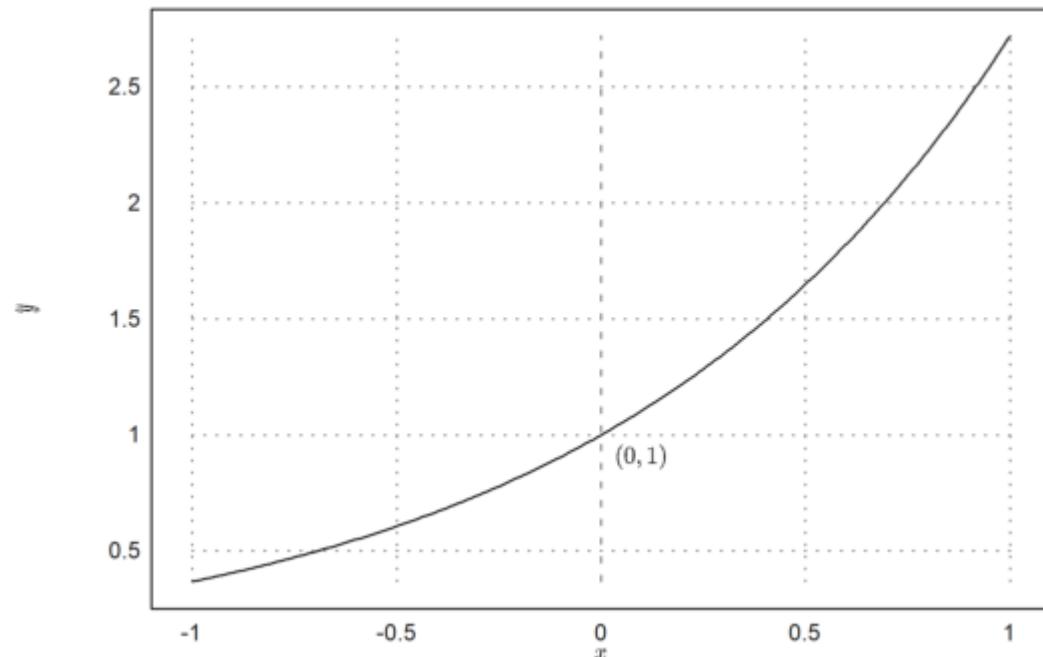
```
n=10; k=bin(n,0:n); ... plot2d(-0.5:n+0.5,k,bar=true,fillcolor=lightgray):
![images/23030630044_Alfia]
```

Data untuk diagram batang (`batang=1`) dan histogram (`histogram=1`) dapat diberikan secara eksplisit dalam `xv` dan `yv`, atau dapat dihitung dari distribusi empiris dalam `xv` dengan `gt;distribusi` (atau `distribusi=n`). Histogram nilai `xv` akan dihitung secara otomatis dengan `gt;histogram`. Jika `gt;even` ditentukan,

nilai xv akan dihitung dalam interval integer.

```
plot2d(normal(10000),distribution=50);
![images/23030630044_Alfia
k=0:10; m=bin(10,k); x=(0:1)-0.5; plot2d(x,m, bar);
![images/23030630044_Alfia
columnsplot(m,k);
```

$$y = e^x$$



```
plot2d(random(600)6,histogram=6);
![images/23030630044_Alfia
```

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

```
plot2d(normal(1,1000),distribution=10,style=""
/"):
![images/23030630044_Alfia
```

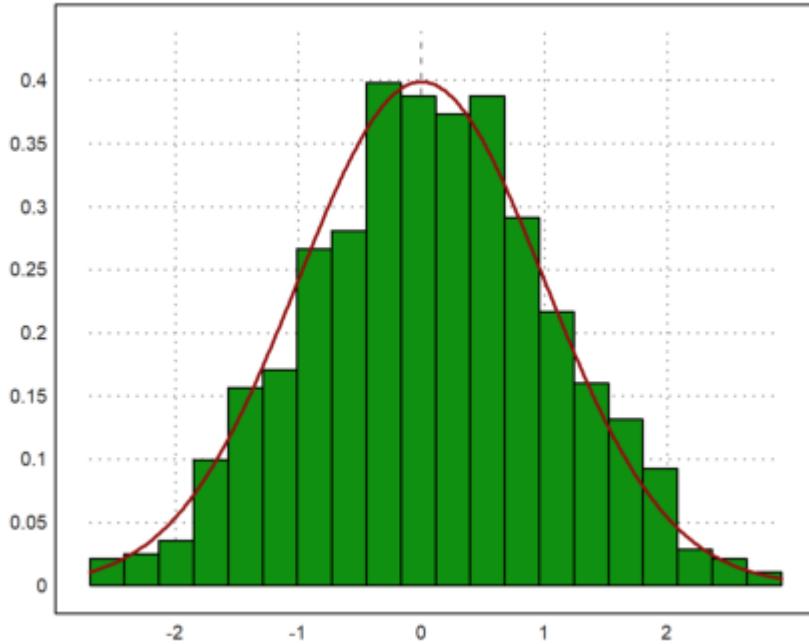
Dengan parameter even=true, ini akan menggunakan interval integer.

```
plot2d(intrandom(1,1000,10),distribution=10,even=true);
![images/23030630044_Alfia
```

Perhatikan bahwa ada banyak plot statistik yang mungkin berguna. Lihatlah tutorial tentang statistik.

```
columnsplot(getmultiplicities(1:6,intrandom(1,6000,6))):
![images/23030630044_Alfia
```

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



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

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

```
![images/23030630044_Alfia
```

Fungsi Implisit

Plot implisit menunjukkan garis level yang menyelesaikan $f(x,y)=\text{level}$, di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika $\text{level}=\text{"auto"}$, akan ada garis level nc, yang akan menyebar antara minimum dan maksimum fungsi secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan gt;hue untuk menunjukkan nilai fungsi. Untuk fungsi implisit, xv harus berupa fungsi atau ekspresi parameter x dan y, atau, sebagai alternatif, xv dapat berupa matriks nilai.

Euler dapat menandai garis level

$$f(x, y) = c$$

dari fungsi apa pun.

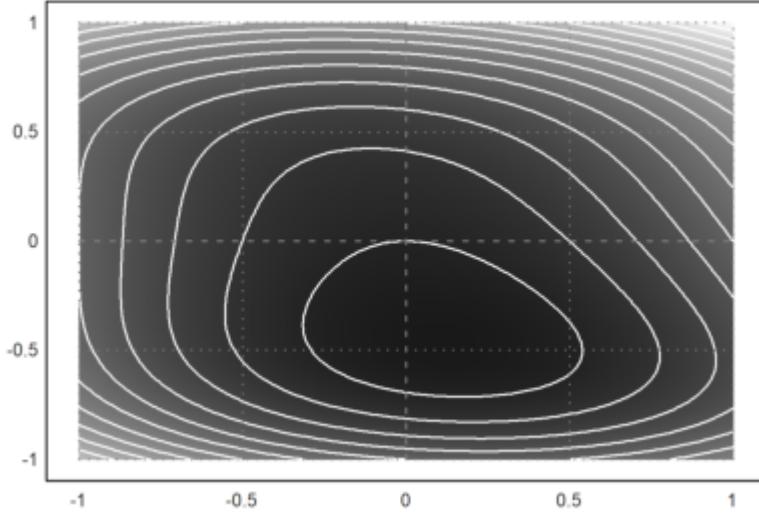
Untuk menggambar himpunan $f(x,y)=c$ untuk satu atau lebih konstanta c, Anda dapat menggunakan `plot2d()` dengan plot implisitnya di bidang. Parameter untuk c adalah `level=c`, di mana c dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi untuk setiap titik dalam plot. Parameter "n" menentukan kehalusan plot.

```
aspect(1.5);
```

```

plot2d("x^2 + y^2 - xy - x", r = 1.5, level = 0, contourcolor = red) :
![images/23030630044_Alfia
expr := "2x^2 + xy + 3y^4 + y"; //define an expression f(x,y)
plot2d(expr,level=0); // Solutions of f(x,y)=0
![images/23030630044_Alfia
plot2d(expr,level=0:0.5:20, hue,contourcolor=white,n=200); // nice

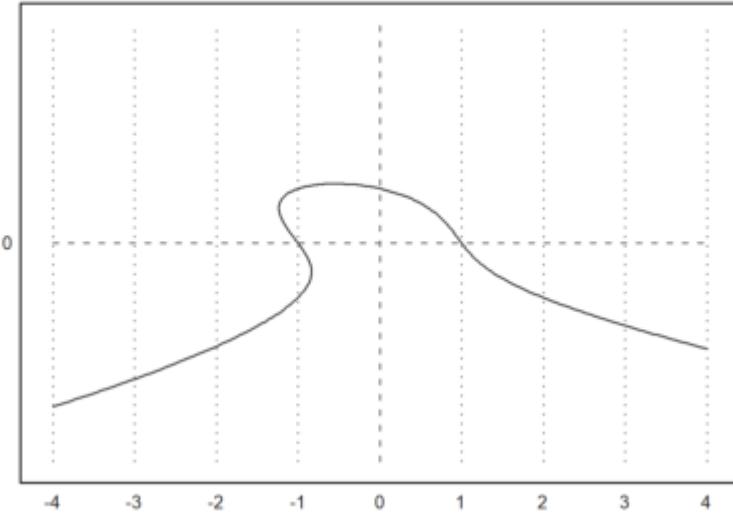
```



```

plot2d(expr,level=0:0.5:20, hue, spectral,n=200,grid=4); // nicer
![images/23030630044_Alfia
Ini juga berlaku untuk plot data. Namun, Anda harus menentukan rentang
untuk label sumbu.
x=-2:0.05:1; y=x'; z=expr(x,y);
plot2d(z,level=0,a=-1,b=2,c=-2,d=1, hue):
![images/23030630044_Alfia
plot2d("x^3 - y^2", contour, hue, spectral) :
![images/23030630044_Alfia
plot2d("x^3 - y^2", level = 0, contourwidth = 3, add, contourcolor = red) :
![images/23030630044_Alfia
z=z+normal(size(z))0.2;
plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
![images/23030630044_Alfia
plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
![images/23030630044_Alfia
plot2d("x^2 + y^3 + xy",level = 1,r = 4,n = 100) :

```



```
plot2d("x^2 + 2y^2 - xy",level = 0 : 0.1 : 10,n = 100,contourcolor = white, hue) :
```

![images/23030630044_Alfia

Dimungkinkan juga untuk mengisi himpunan

$$a \leq f(x, y) \leq b$$

dengan rentang level.

Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

![images/23030630044_Alfia

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

```
plot2d("x^4 + y^4",r = 1.5,level = [0;1],color = blue,style = "/") :
```

![images/23030630044_Alfia

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

![images/23030630044_Alfia

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

![images/23030630044_Alfia

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

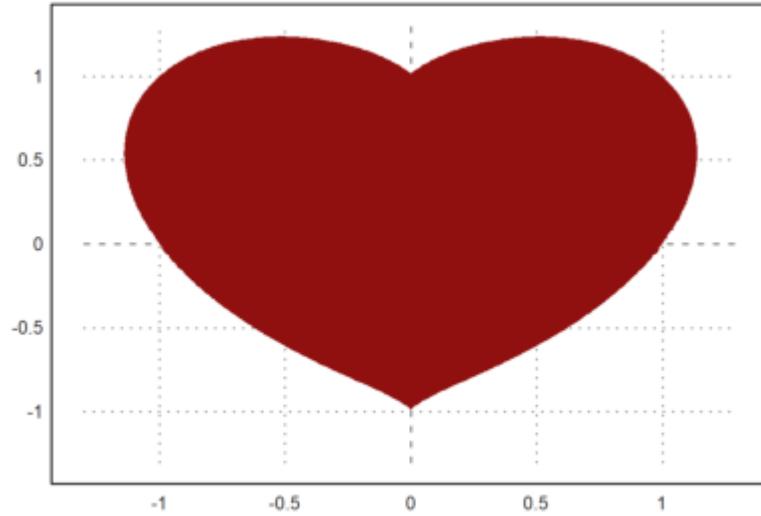
![images/23030630044_Alfia

Dimungkinkan juga untuk menandai suatu wilayah

$$a \leq f(x, y) \leq b.$$

Hal ini dilakukan dengan menambahkan level dengan dua baris.

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



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

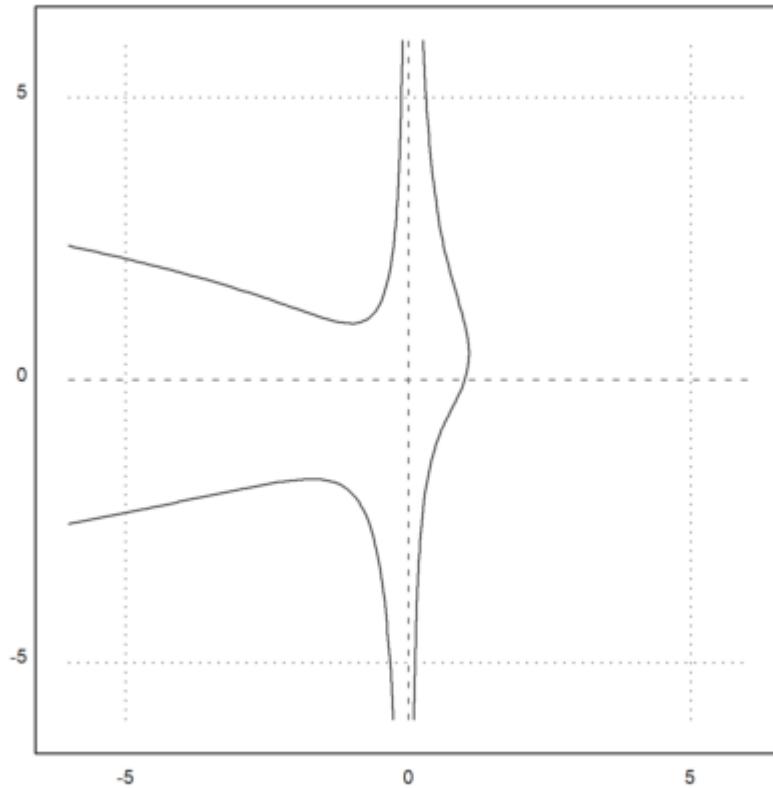
$$x^3 - xy + x^2y^2 = 6$$

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

```
![images/23030630044_Alfia
function starplot1 (v, style="/" , color=green, lab=none) ...
if !holding() then clg; endif; w=window(); window(0,0,1024,1024); h=holding(1);
r=max(abs(v))*1.2; setplot(-r,r,-r,r); n=cols(v); t=linspace(0,2pi,n); v=v-v[1];
c=v*cos(t); s=v*sin(t); cl=barcolor(color); st=barstyle(style); loop 1 to n polygon([0,c[],c[+1]], [0,s[],s[+1]],1); if lab!=none then rlab=v[]+r*0.1; col,row=toscreen(cos(t[])*rlab,sin(t[])*rlab);
ctext(""+lab[],col,row-textheight()/2); endif; end; barcolor(cl); barstyle(st); holding(h); window(w); endfunction ]
Tidak ada tanda centang pada grid atau sumbu di sini. Selain itu, kami menggunakan jendela penuh untuk plot.
```

Kami memanggil reset sebelum menguji plot ini untuk mengembalikan grafik ke default. Ini tidak perlu, jika Anda yakin bahwa plot Anda berfungsi.

```
reset; starplot1(normal(1,10)+5,color=red,lab=1:10);
```



Terkadang, Anda mungkin ingin memplot sesuatu yang tidak dapat dilakukan oleh plot2d, tetapi hampir.

Dalam fungsi berikut, kita melakukan plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

```
function logimpulseplot1 (x,y) ...
x0,y0=makeimpulse(x,log(y)/log(10)); plot2d(x0,y0,bar,grid=0); h=holding(1);
frame(); xgrid(ticks(x)); p=plot(); for i=-10 to 10; if i=p[4] and i=p[3] then
ygrid(i,yt="10"+i); endif; end; holding(h); endfunction </pre > Marikitamengujinyadengannilai-nilaiyangterdistribusise secaraeksponensial.
```

```
aspect(1.5); x=1:10; y=-log(random(size(x)))200; ... logimpulseplot1(x,y);
![images/23030630044_Alfia]
```

Mari kita animasikan kurva 2D menggunakan plot langsung. Perintah plot(x,y) cukup memplot kurva ke dalam jendela plot. setplot(a,b,c,d) mengatur jendela ini.

Fungsi wait(0) memaksa plot untuk muncul di jendela grafik. Jika tidak, penggambaran ulang akan dilakukan dalam interval waktu yang jarang.

```
function animliss (n,m) ...
```

```
t=linspace(0,2pi,500); f=0; c=framecolor(0); l=linewidth(2); setplot(-1,1,-1,1); repeat clg; plot(sin(n*t),cos(m*t+f)); wait(0); if testkey() then break; endif; f=f+0.02; end; framecolor(c); linewidth(l); endfunction i/pre; Tekan tombol apa saja untuk menghentikan animasi ini.
```

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

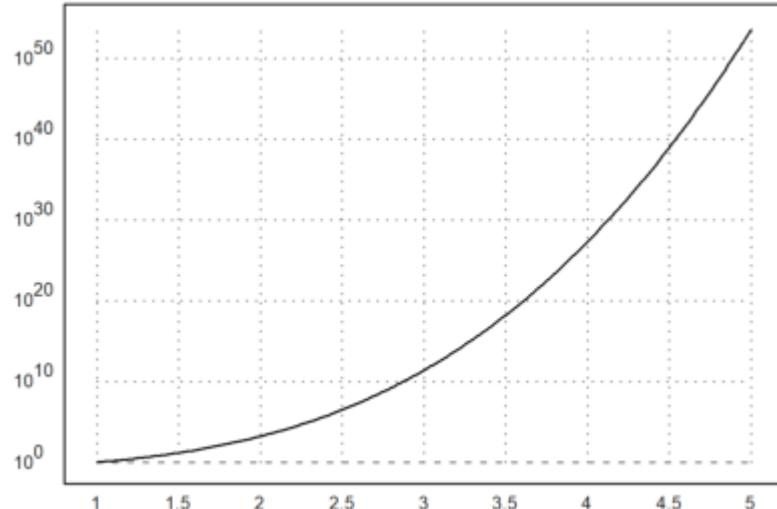
Plot Logaritmik

EMT menggunakan parameter "logplot" untuk skala logaritmik.

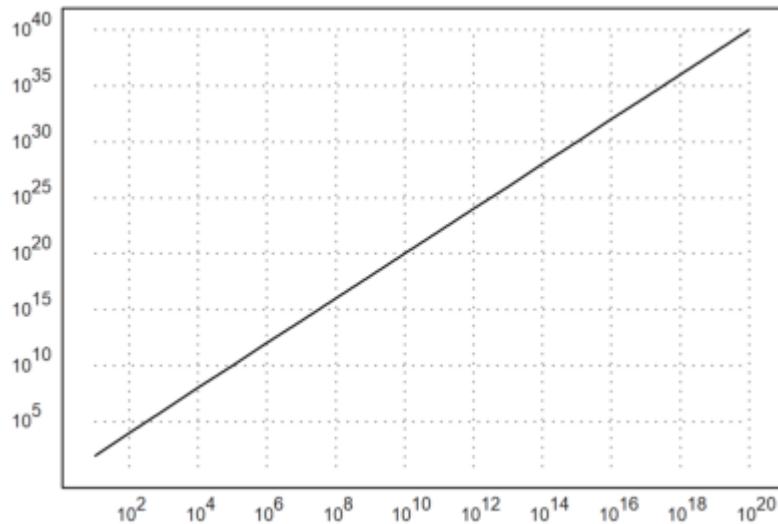
Plot logaritmik dapat diplot menggunakan skala logaritmik dalam y dengan logplot=1, atau menggunakan skala logaritmik dalam x dan y dengan logplot=2, atau dalam x dengan logplot=3.

- * logplot=1: y-logaritmik
- * logplot=2: x-y-logaritmik
- * logplot=3: x-logaritmik

plot2d("exp(x³ - x)x²", 1, 5, logplot = 1) :



```
plot2d("exp(x+sin(x))",0,100,logplot=1);
![images/23030630044_Alfia
plot2d("exp(x+sin(x))",10,100,logplot=2);
![images/23030630044_Alfia
plot2d("gamma(x)",1,10,logplot=1);
![images/23030630044_Alfia
plot2d("log(x(2+sin(x/100)))",10,1000,logplot=3);
![images/23030630044_Alfia
Hal ini juga berlaku untuk plot data.
x=10^(1 : 20); y = x2 - x;
plot2d(x,y,logplot=2);
```



```
x = -10:0.01:10; a = (1:5)'; plot2d(x, a*sin(x), color=getspectral(a/5));
```

2.11 Rujukan Lengkap Fungsi Plot 2D

fungsi plot2d (xv, yv, btest, a, b, c, d, xmin, xmax, r, n, logplot, grid, bingkai, framecolor, kotak, warna, ketebalan, gaya, tomatis, tambah, pengguna, delta, titik, addpoints, pointstyle, batang, histogram, istribusi, merata, langkah, sendiri, adaptif, rona, level, kontur, nc, terisi, warna isian, garis besar, judul, xl, yl, peta, warna kontur, lebar kontur, tanda centang, margin, kliping, cx, cy, insimg, spektral, cgrid, vertikal, lebih kecil, dl, niveau, level)

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

Parameter

x,y : persamaan, fungsi atau vektor data

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

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

r dapat berupa vektor [rx,ry] atau vektor [rx1,rx2,ry1,ry2].

xmin,xmax : rentang parameter untuk kurva

auto : Menentukan rentang y secara otomatis (default)

aquare : jika benar, cobalah untuk mempertahankan rentang x-y persegi

n : jumlah interval (standarnya adaptif)

grid : 0 = tidak ada kisi dan label,

1 = sumbu saja,

2 = grid normal (lihat di bawah untuk jumlah garis grid)

3 = sumbu dalam

4 = tidak ada grid

5 = grid penuh termasuk margin
6 = centang pada bingkai
7 = sumbu saja
8 = sumbu saja, sub-centang
frame : 0 = tanpa bingkai
framecolor: warna bingkai dan grid
margin : angka antara 0 dan 0,4 untuk margin di sekitar plot
color : Warna kurva. Jika ini adalah vektor warna,
itu akan digunakan untuk setiap baris matriks plot. Dalam
kasus plot titik, itu harus berupa vektor kolom. Jika
suatu vektor baris atau a matriks warna penuh
digunakan untuk plot titik, itu akan digunakan untuk
setiap titik data.
thickness : ketebalan garis untuk kurva
Nilai ini bisa lebih kecil dari 1 untuk garis yang sangat
tipis.
style : Gaya plot untuk garis, penanda, dan isian.
Untuk penggunaan poin
"[]", "lt;gt;", ".", "..", "...",
"*", "+", "--", "-", "o"
"[]", "lt;gt;", "o" (bentuk terisi)
"[]w", "lt;gt;w", "ow" (tidak transparan)
Untuk penggunaan garis
"-", "--", "-.", ".-", "-.-", "-gt;"
Untuk penggunaan poligon atau plot batang yang terisi
"\"", "O", "O", "/", ";",
"+", "--", "-", "t"
points : plot titik tunggal, bukan segmen garis
addpoints : jika benar, buat plot segmen garis dan titik
add : menambahkan plot ke plot yang ada
user : memungkinkan interaksi pengguna untuk fungsi
delta : ukuran langkah untuk interaksi pengguna
bar : plot batang (x adalah batas interval, y adalah nilai interval)
histogram : memplot frekuensi x dalam n subinterval
distribution=n : memplot distribusi x dengan n subinterval
even : gunakan nilai antar untuk histogram otomatis.
steps : memplot fungsi sebagai fungsi langkah (langkah=1,2)
adaptive : gunakan plot adaptif (n adalah jumlah langkah minimal)
level : garis tingkat plot dari fungsi implisit dua variabel
outline : menarik batas rentang level
Jika nilai level adalah matriks 2xn, rentang level akan digambarkan dalam
warna menggunakan gaya isian yang diberikan. Jika garis besarnya benar, maka
itu benar akan digambar dalam warna kontur. Dengan menggunakan fitur ini,
wilayah f(x,y) antar batas dapat ditandai.
hue : tambahkan warna rona ke plot level untuk menunjukkan nilai
fungsi

contour : Gunakan plot level dengan level otomatis
nc : jumlah garis level otomatis
title : judul plot (default "")
xl, yl : label untuk sumbu x dan y
smaller : jika $gt;0$, akan ada lebih banyak ruang di sebelah kiri untuk label.
vertical : Mengaktifkan atau menonaktifkan label vertikal. Ini mengubah label vertikal variabel global secara lokal untuk satu plot. Nilai 1 hanya menetapkan teks vertikal, nilai 2 menggunakan label numerik vertikal pada sumbu y.
filled : mengisi plot kurva
fillcolor : warna isian untuk batang dan kurva yang terisi
outline : batas poligon terisi
logplot : mengatur plot logaritmik
 1 = logplot in y,
 2 = logplot in xy,
 3 = logplot in x
own :
 Sebuah string, yang menunjuk ke rutinitas plotnya sendiri. Dengan itu pengguna, Anda mendapatkan interaksi pengguna yang sama seperti di plot2d. Kisarannya akan ditentukan sebelum setiap panggilan ke fungsi Anda.
maps : map expressions (0 is faster), functions are always mapped.
contourcolor : color of contour lines
contourwidth : width of contour lines
clipping : toggles the clipping (default is true)
title :
 Ini dapat digunakan untuk mendeskripsikan plot. Judul akan muncul di atas plot. Selain itu, label untuk sumbu x dan y dapat ditambahkan dengan $xl="string"$ atau $yl="string"$. Label lain dapat ditambahkan dengan fungsi `label()` atau `labelbox()`. Judulnya bisa berupa unicode string atau gambar rumus Lateks.
cgrid :
 Menentukan jumlah garis grid untuk plot grid yang kompleks.
 Harus berupa pembagi ukuran matriks dikurangi 1 (jumlah subinterval).
cgrid dapat berupa vektor [cx,cy].
Ringkasan
 Fungsinya dapat diplot
 * ekspresi, kumpulan panggilan atau fungsi dari satu variabel,
 * kurva parametrik,
 * x data versus y data,
 * fungsi implisit,
 * plot batang,
 * jaringan kompleks,
 * poligon.
 Jika fungsi atau ekspresi untuk xv diberikan, `plot2d()` akan menghitung nilai dalam rentang tertentu menggunakan fungsi atau ekspresi tersebut. Ekspresinya

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

Perhatikan bahwa plot adaptif selalu dihitung elemen demi elemen.

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

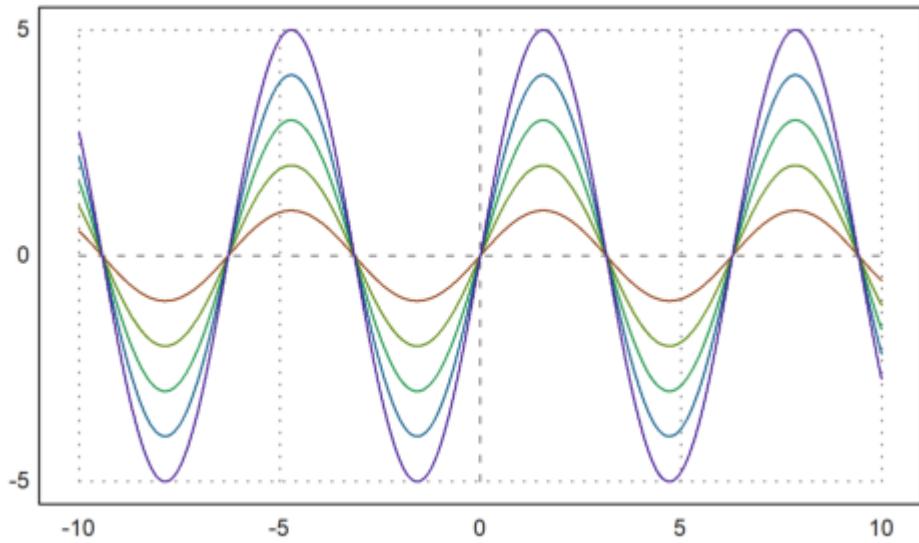
2.12 Latihan Soal

1. Gunakan Euler Math Toolbox untuk memplot grafik fungsi

$$f(x) = a * \sin$$

dengan nilai parameter aa yang bervariasi dari 1 hingga 5. Buat grafik dalam rentang x=-10 hingga x=10, dengan setiap grafik diberi warna yang berbeda berdasarkan spektrum warna.

```
x = -10:0.01:10;  
a = (1:5)';  
plot2d(x, asin(x), color=getspectral(a/5));
```



3 Plot 3D

3.1 Menggambar Plot 3D dengan EMT

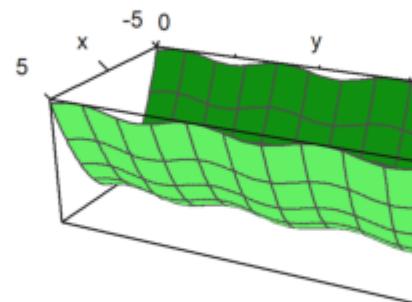
Ini adalah pengantar plot 3D di Euler. Kita memerlukan plot 3D untuk memvisualisasikan fungsi dua variabel.

Euler menggambar fungsi tersebut menggunakan algoritma pengurutan untuk menyembunyikan bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Defaultnya 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 tinggi dapat diubah.

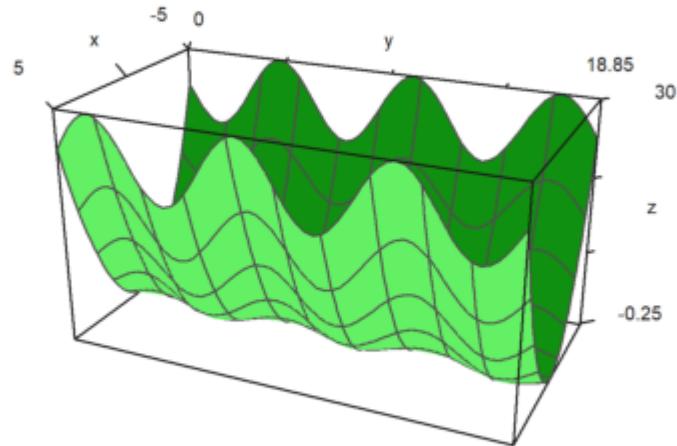
Euler dapat memplot

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

Plot 3D suatu 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, 6pi) :
```



`plot3d("x2+xsin(y)", -5, 5, 0, 6pi) :`

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.

3.2 Fungsi Dua Variabel

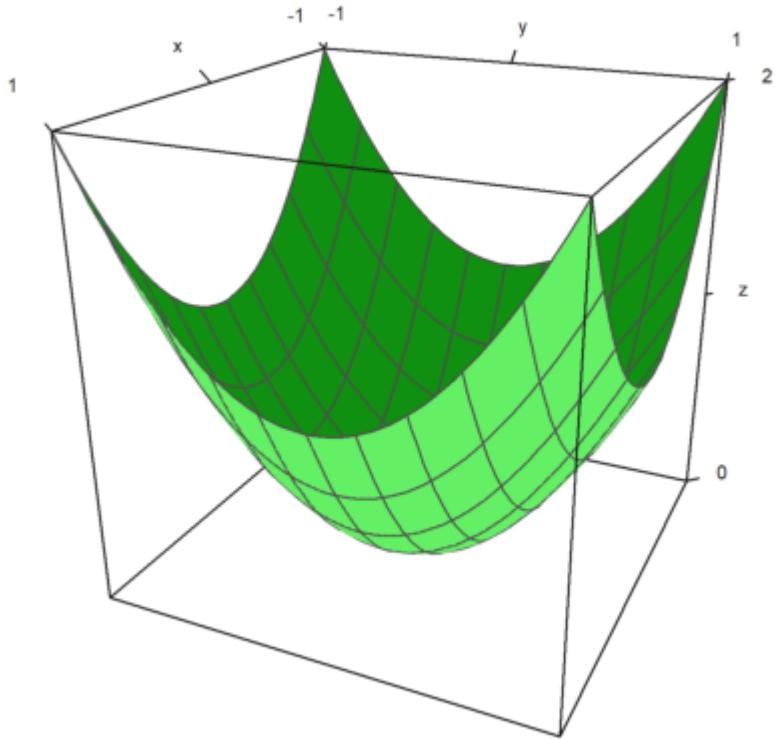
Untuk grafik fungsi, gunakan

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

Defaultnya adalah kisi kawat yang terisi dengan warna berbeda di kedua sisinya. Perhatikan bahwa jumlah interval kisi default adalah 10, tetapi plot menggunakan jumlah persegi panjang 40x40 default untuk membuat permukaan. Ini dapat diubah.

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

Kami menggunakan default n=40 dan kisi=10.



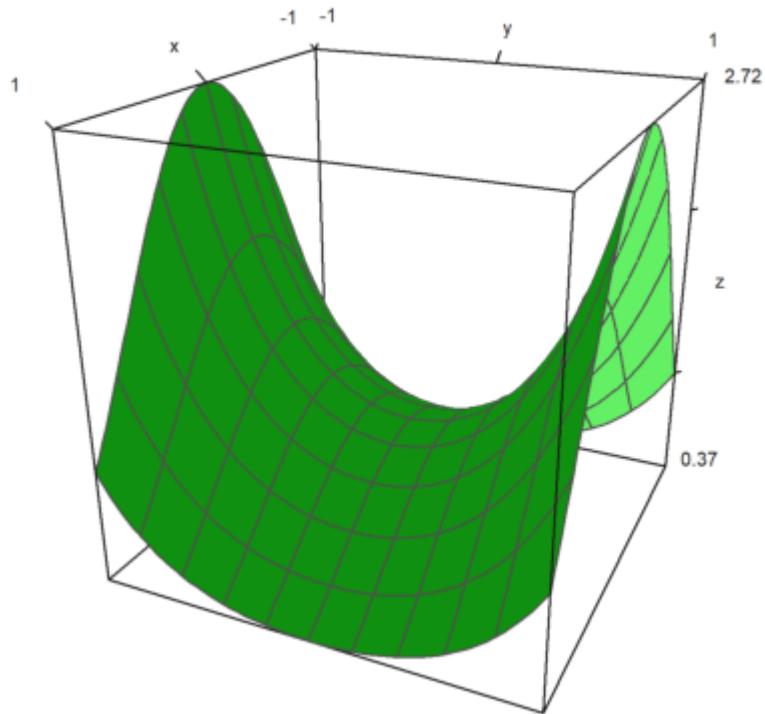
plot3d("x²+y²") :

Interaksi pengguna dimungkinkan dengan parameter gt;user. Pengguna dapat menekan tombol berikut.

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

plot3d("exp(-x²+y²)", user, ...title = "Turn with the vector keys (press return to finish)") :

Turn with the vector keys (press return to finish)



Rentang plot untuk fungsi dapat ditentukan dengan

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

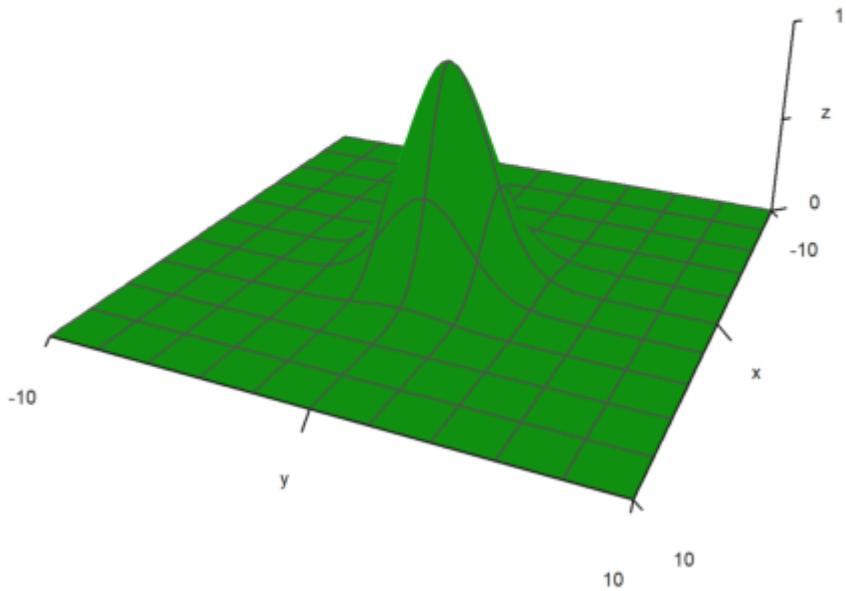
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: skala ke nilai fungsi (default adalah 1t;fscale).

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

frame: jenis frame (default 1).

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



3, user) :

Tampilan dapat diubah dengan berbagai cara.

- * jarak: jarak pandang ke plot.
- * perbesaran: nilai perbesaran.
- * sudut: sudut ke sumbu y negatif dalam radian.
- * tinggi: tinggi tampilan dalam radian.

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

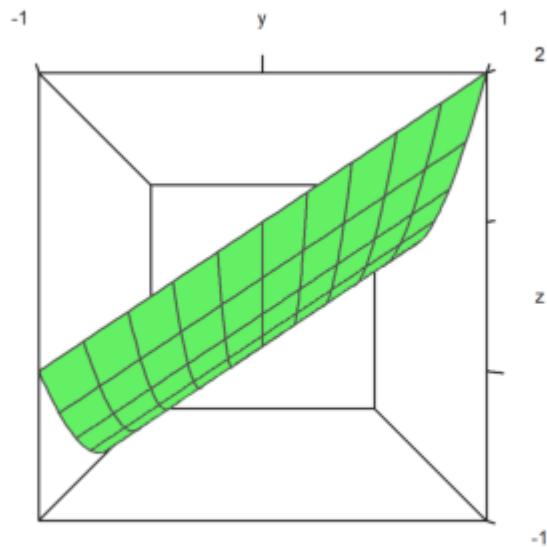
view

[5, 2.6, 2, 0.4]

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

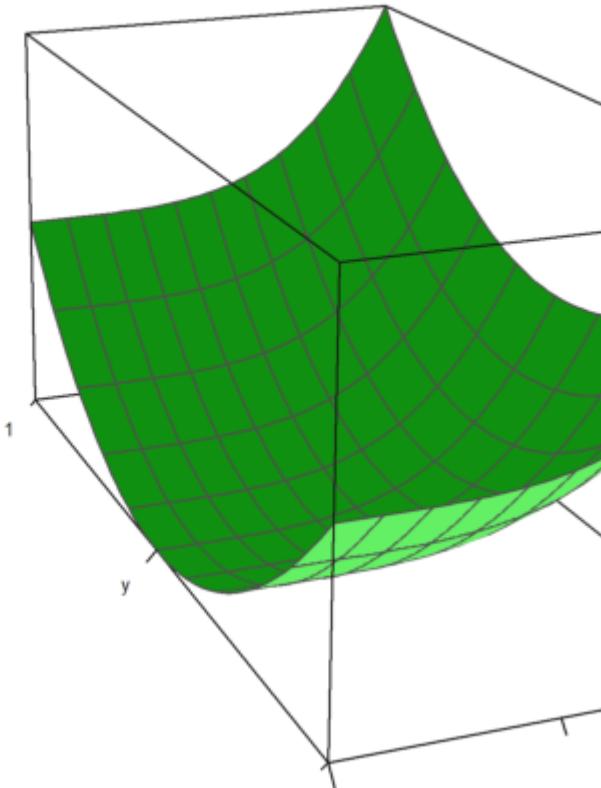
Dalam contoh berikut, sudut=0 dan tinggi=0 terlihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

plot3d("x² + y", distance = 3, zoom = 1, angle = pi/2, height = 0) :



Plot selalu mengarah ke 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 =`

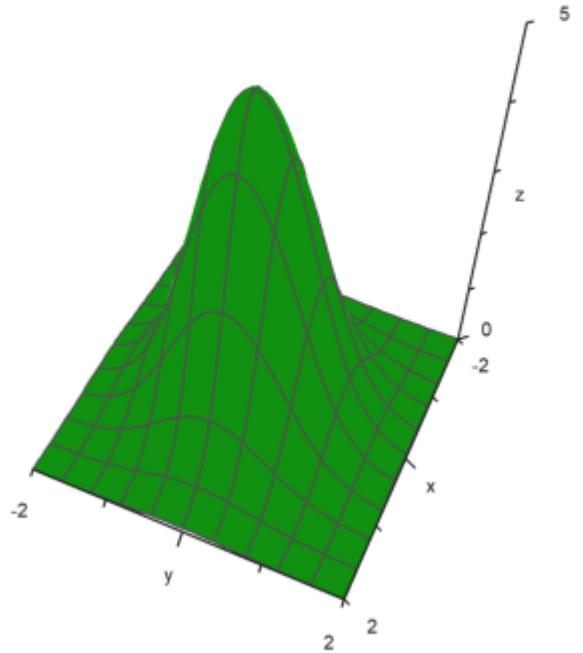


20,...*center* = [0.4, 0, 0], *zoom* = 5) :

Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung pada ukuran plot. Namun, label merujuk pada ukuran sebenarnya.

Jika Anda menonaktifkannya dengan *scale*=false, Anda perlu berhati-hati agar plot tetap sesuai dengan jendela plot, dengan mengubah jarak tampilan atau zoom, dan memindahkan bagian tengah.

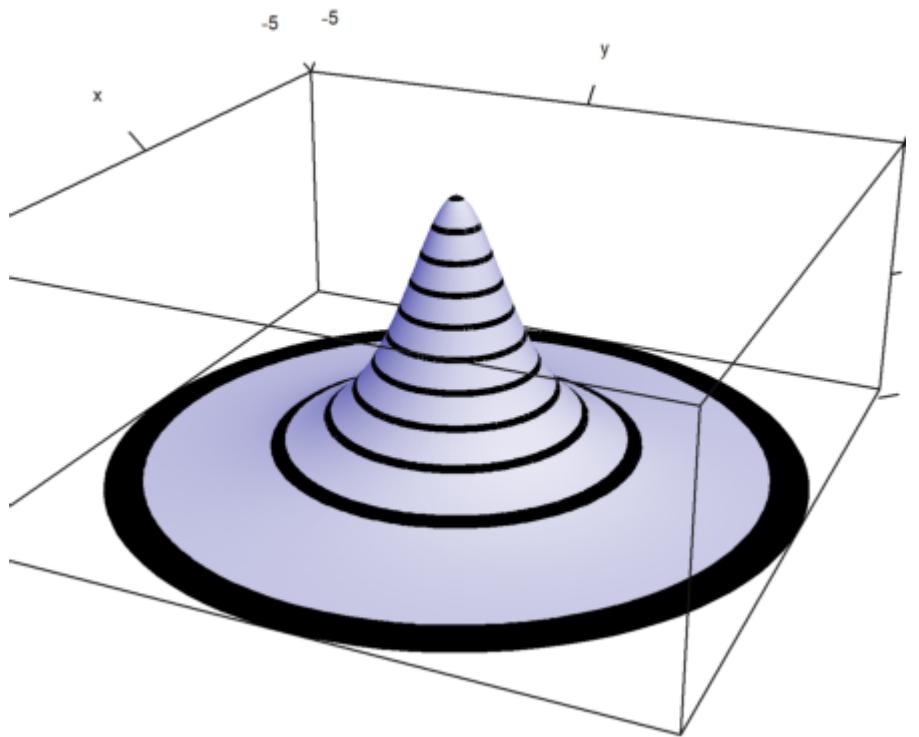
`plot3d("5exp(-x2 - y2)", r = 2, < fscale, < scale, distance = 13, height =`



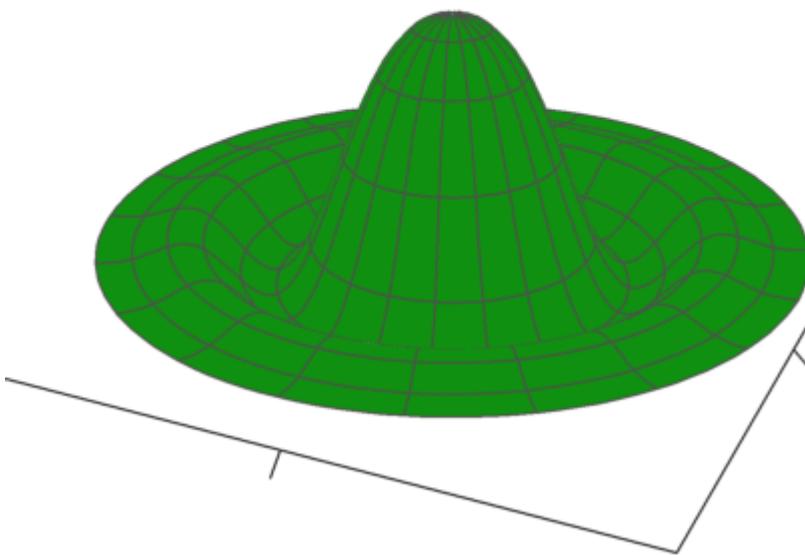
`50,...center = [0, 0, -2], frame = 3) :`

Plot polar juga tersedia. Parameter `polar=true` menggambar plot polar. Fungsi tersebut harus tetap berupa fungsi x dan y. Parameter `"fscale"` menskalakan fungsi dengan skalanya sendiri. Jika tidak, fungsi tersebut diskalakan agar sesuai dengan kubus.

`plot3d("1/(x2 + y2 + 1)", r = 5, polar,... fscale = 2, hue, n = 100, zoom =`



```
4, contour, color = blue) :  
function f(r) := exp(-r/2)cos(r); ...      plot3d("f(x2 + y2)", polar, scale =
```

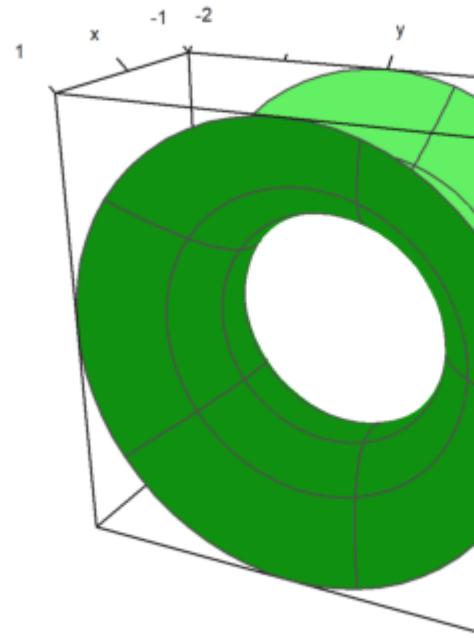


[1, 1, 0.4], $r = pi$, frame = 3, zoom = 4) :

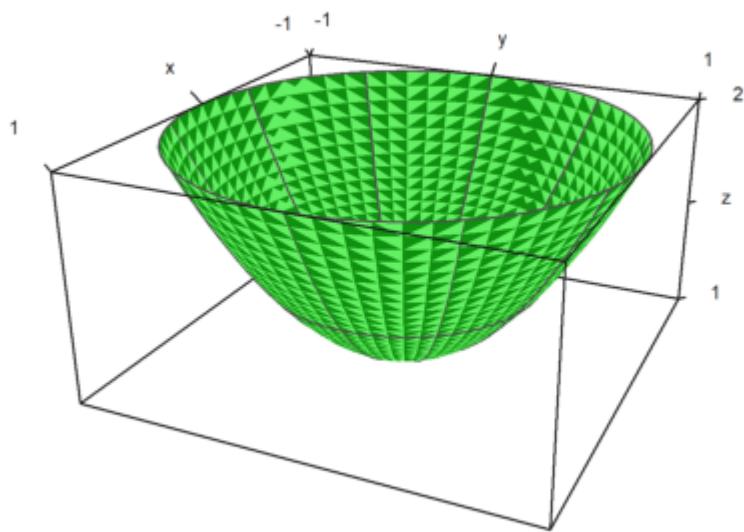
Parameter rotate memutar fungsi dalam x di sekitar sumbu x.

* rotate=1: Menggunakan sumbu x

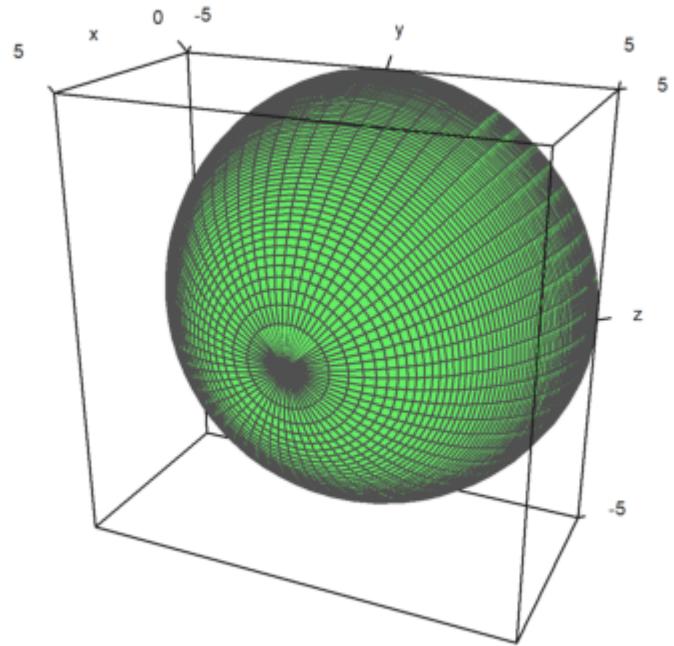
* rotate=2: Menggunakan sumbu z



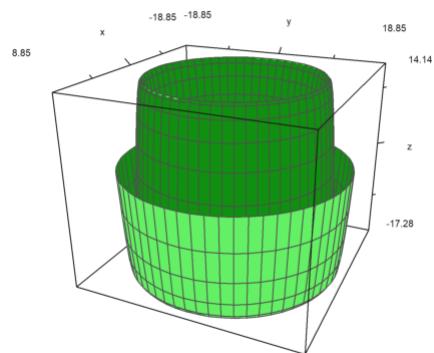
```
plot3d("x^2+1",a = -1,b = 1,rotate = true,grid = 5) :
```



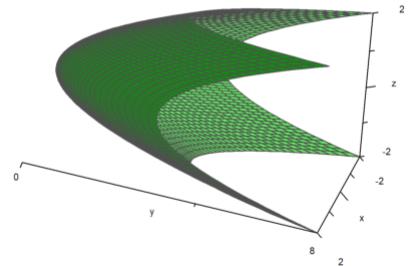
1", $a = -1, b = 1, rotate = 2, grid = 5$) :



$x^2)$, $a = 0, b = 5, \text{rotate} = 1$) :



plot3d("xsin(x)",a=0,b=6pi,rotate=2):
Berikut adalah plot dengan tiga fungsi.



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

3.3 Plot Kontur

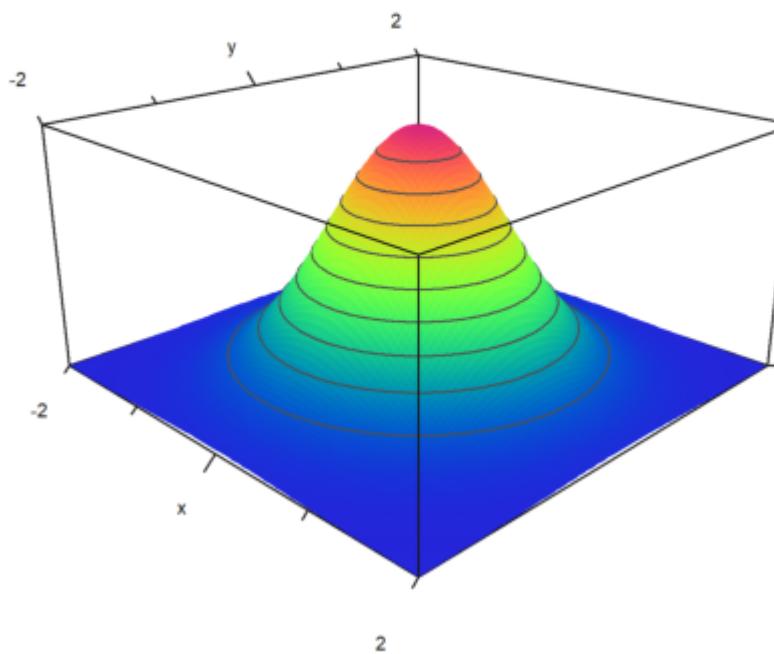
Untuk plot, Euler menambahkan garis kisi. Sebagai gantinya, dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona warna spektral. Euler dapat menggambar tinggi fungsi pada plot dengan bayangan. Dalam semua plot 3D, Euler dapat menghasilkan anaglif merah/sian.

- * gt;hue: Mengaktifkan bayangan terang alih-alih kabel.
- * gt;contour: Memplot garis kontur otomatis pada plot.
- * level=... (atau level): Vektor nilai untuk garis kontur.

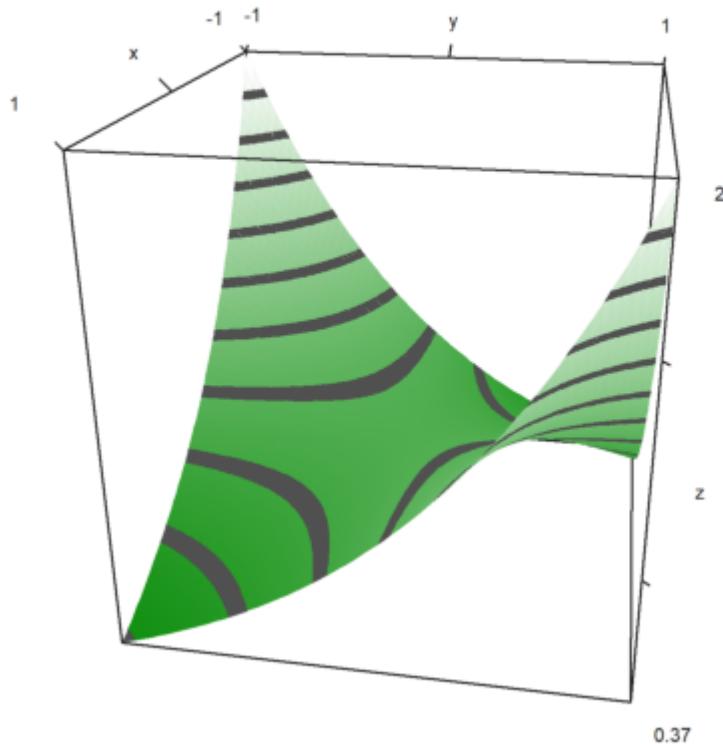
Nilai default adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

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

```
plot3d("exp(-x^2-y^2)",r = 2,n = 100,level = "thin",...contour,spectral,fscale =
```



1, $scale = 1.1$, $angle = 45$, $height = 20$) :



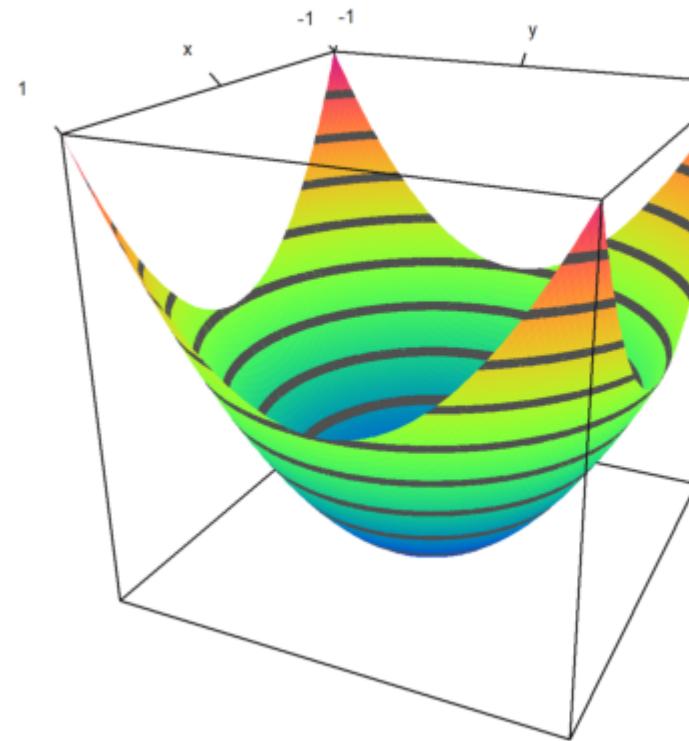
y)" , angle = 100, contour, color = green) :

Shading default menggunakan warna abu-abu. Namun, rentang warna spektral juga tersedia.

* gt:spectral: Menggunakan skema spektral default

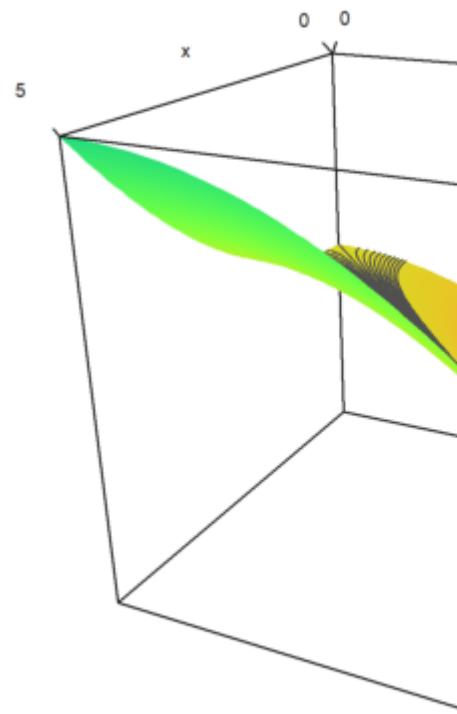
* color=...: Menggunakan warna khusus atau skema spektral

Untuk plot berikut, kami menggunakan skema spektral default dan menambah jumlah titik untuk mendapatkan tampilan yang sangat halus.



plot3d("x²+y²", *spectral, contour, n = 100*) :

Alih-alih garis level otomatis, kita juga dapat mengatur nilai garis level. Ini akan menghasilkan garis level tipis alih-alih rentang level.

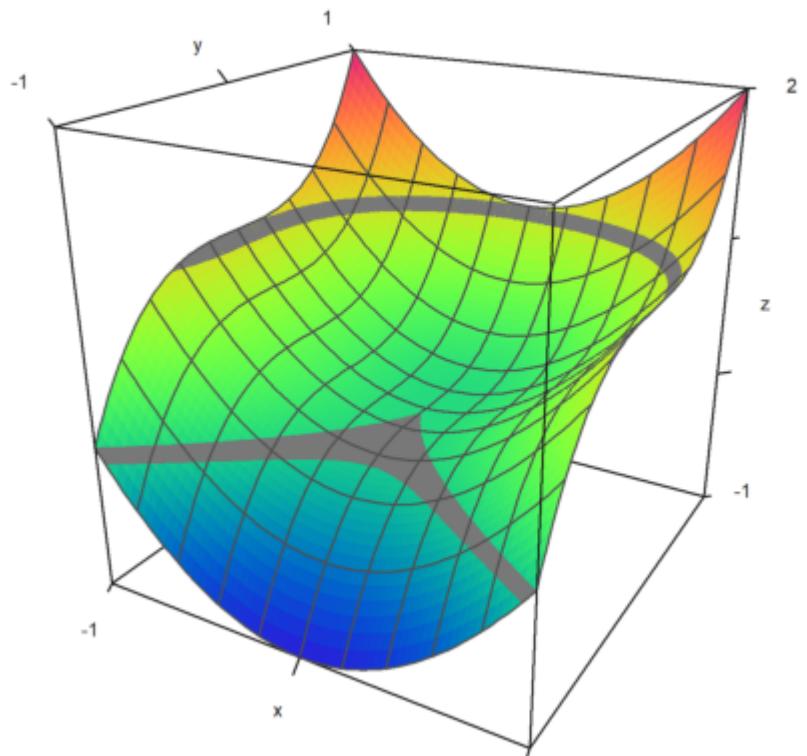


`plot3d("x^2-y^2", 0, 5, 0, 5, level = -1 : 0.1 : 1, color = redgreen) :`

Dalam plot berikut, kami menggunakan dua pita level yang sangat lebar dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas level sebagai kolom.

Selain itu, kami melapisi kisi dengan 10 interval di setiap arah.

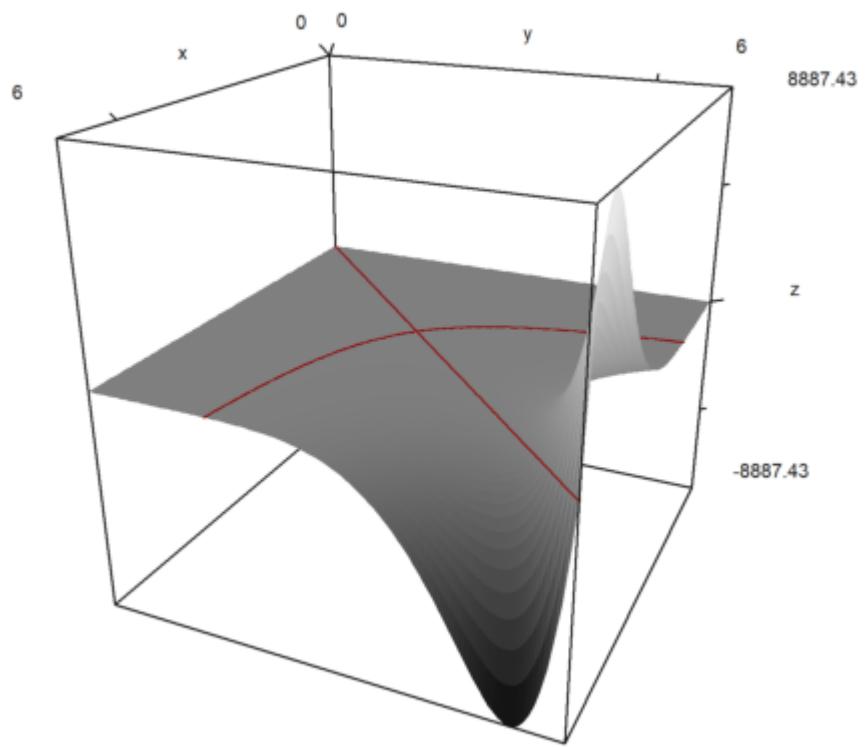
`plot3d("x^2 + y^3", level = [-0.1, 0.9; 0, 1], ... spectral, angle = 30, grid =`



10, *contourcolor = gray*) :

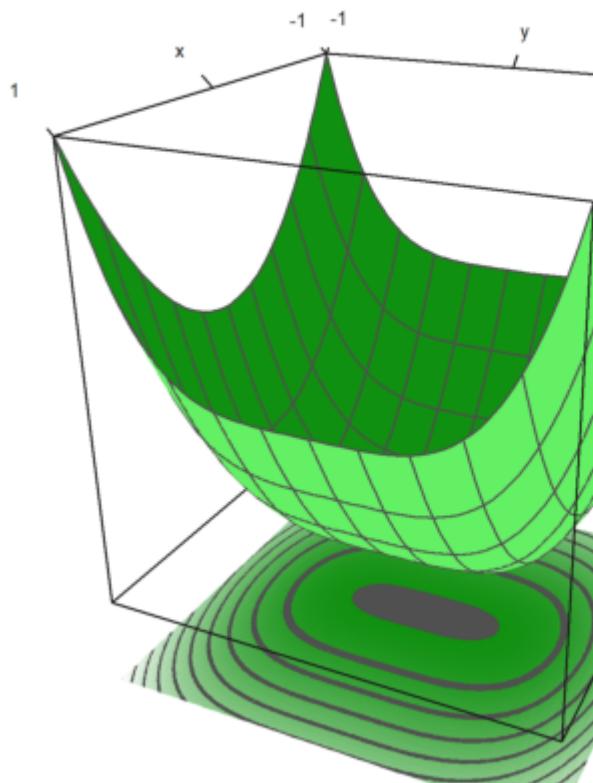
Dalam contoh berikut, kami memplot himpunan, di mana
 $f(x, y) = x^y - y^x = 0$ Kami menggunakan satu garis tipis untuk garis datar.

`plot3d("x^y - y^x", level = 0, a = 0, b = 6, c = 0, d = 6, contourcolor =`



red, $n = 100$) :

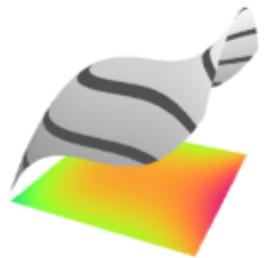
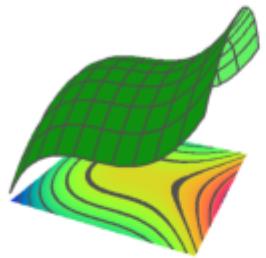
Dimungkinkan untuk menunjukkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.



```
plot3d("x^2+y^4", cp, cpcolor = green, cpdelta = 0.2) :
```

Berikut ini beberapa gaya lainnya. Kami selalu menonaktifkan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi.

```
figure(2,2); ... expr="y^3-x^2"; ... figure(1); ... plot3d(expr, < frame, cp, cpcolor = spectral); ... figure(2); ... plot3d(expr, < frame, spectral, grid = 10, cp = 2); ... figure(3); ... plot3d(expr, < frame, contour, color = gray, nc = 5, cp = 3, cpcolor = greenred); ... figure(4); ... plot3d(expr, <
```

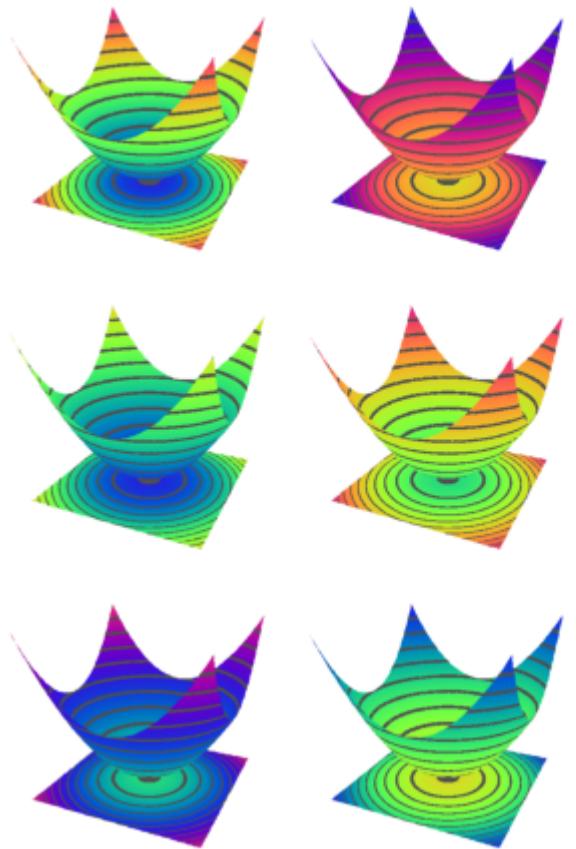


frame, hue, grid = 10, transparent, cp, cpcolor = gray); ... figure(0) :

Ada beberapa skema spektral lain, yang diberi nomor dari 1 hingga 9. Namun, Anda juga dapat menggunakan *color=value*, di mana value

- * spektral: untuk rentang dari biru hingga merah
- * putih: untuk rentang yang lebih redup
- * kuning biru, 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) :

Sumber cahaya dapat diubah dengan 1 dan tombol kurSOR selama interaksi pengguna. Sumber cahaya juga dapat diatur dengan parameter.

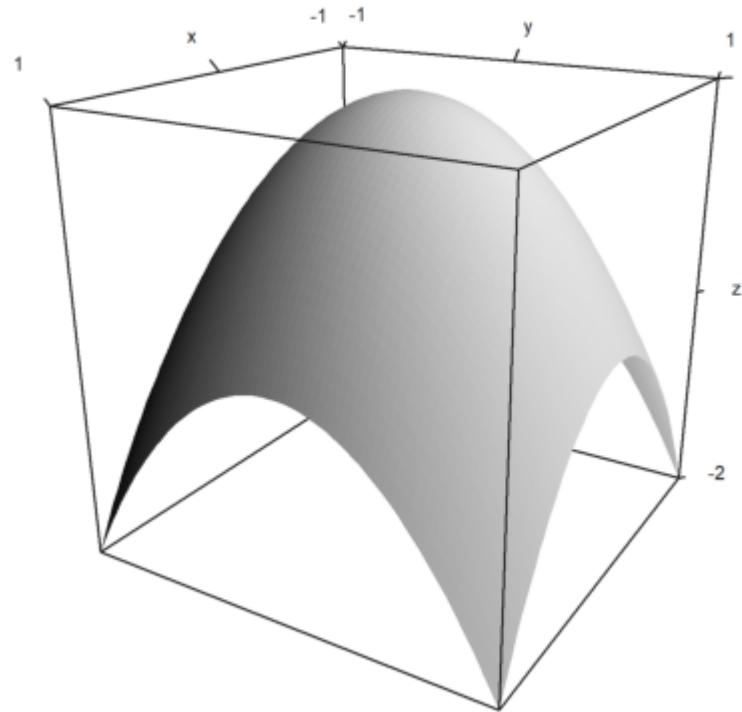
* light: arah cahaya

* amb: cahaya sekitar antara 0 dan 1

Perlu dicatat bahwa program tidak membuat perbedaan antara sisi plot. Tidak ada bayangan. Untuk ini, Anda memerlukan Povray.

plot3d("-x²-y²",...hue = true, light = [0,1,1], amb = 0, user = true, ...title =

Press l and cursor keys (return to exit)



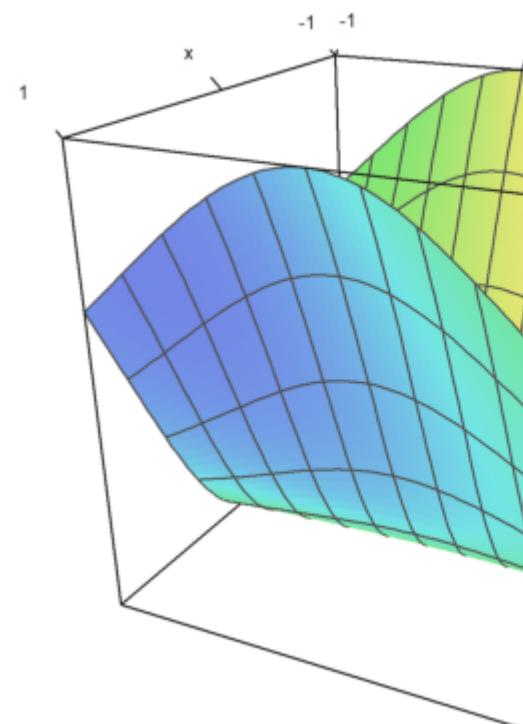
”Presslandcursorkeys(returntoexit)” :

Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

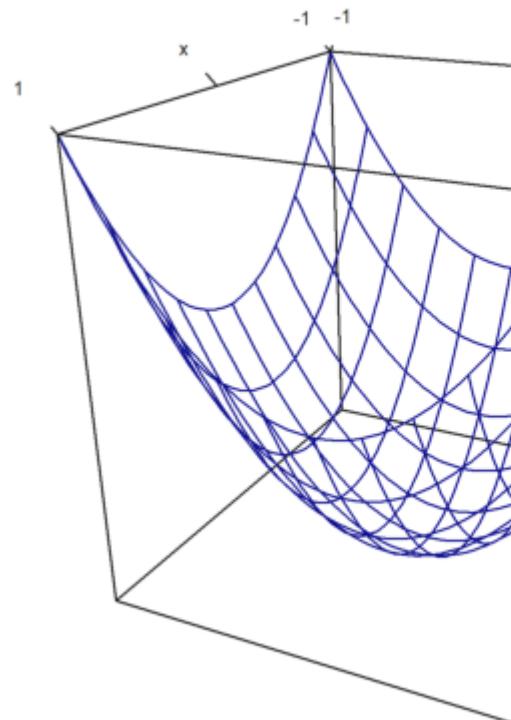
```
plot3d(" -x2-y2", color = rgb(0.7, 0.7, 0), hue = true, frame = false, ...zoom =
```



$3, \text{contourcolor} = \text{red}, \text{level} = -2 : 0.1 : 1, \text{dl} = 0.01$) :
Warna 0 memberikan efek pelangi khusus.



```
plot3d("x^2/(x^2+y^2+1)", color = 0, hue = true, grid = 10) :  
Permukaannya juga bisa transparan.
```



```
plot3d("x^2+y^2", transparent, grid = 10, wirecolor = blue) :
```

3.4 Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d mencakup plot implisit. Plot ini menunjukkan himpunan nol dari suatu fungsi dalam tiga variabel.

Solusi dari $f(x, y, z) = 0$ dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, x-z, dan y-z.

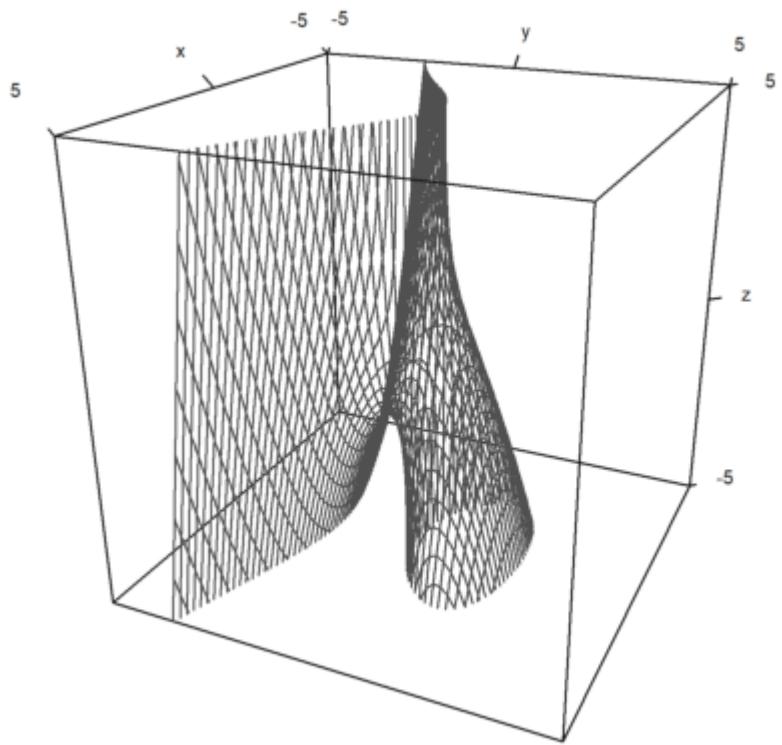
- * implisit=1: potongan sejajar dengan bidang y-z

- * implisit=2: potongan sejajar dengan bidang x-z

- * implisit=4: potongan sejajar dengan bidang x-y

Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh ini, kami memplot

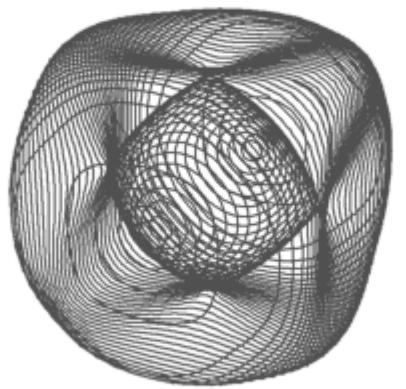
$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$ `plot3d("x^2+y^3+zy-1", r = 5, implicit =`



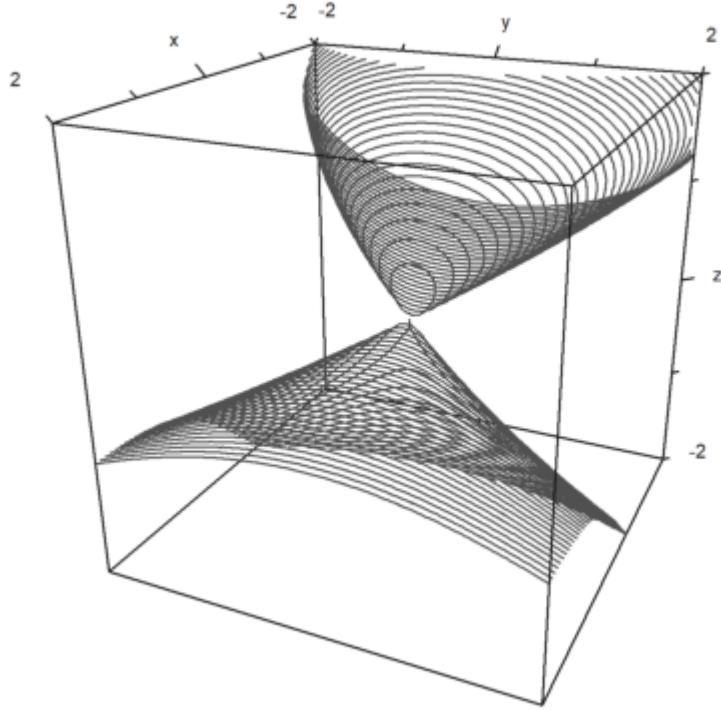
3) :
1; $d = 1;$

$c =$

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+



$(y^2 - 1)^2 - d^r, r = 2, < frame, implicit, user) :$



$y^2 + 4xz + z^3$, implicit, $r = 2$, zoom = 2.5) :
 $y^2 + 4xz + z^3$, implicit, $r = 5$, zoom = 3) :

3.5 Membuat Plot 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 $fx(x,y)$, $fy(x,y)$, $fz(x,y)$. $\gamma(t,s) = (x(t,s), y(t,s), z(t,s))$

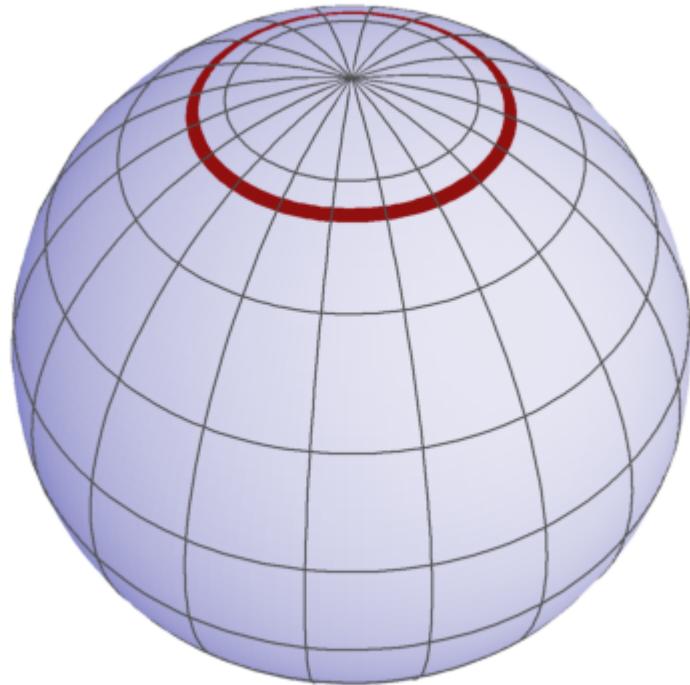
Karena x,y,z adalah matriks, kami berasumsi bahwa (t,s) berjalan melalui kisi persegi. Hasilnya, Anda dapat membuat plot gambar persegi panjang di ruang angkasa.

Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

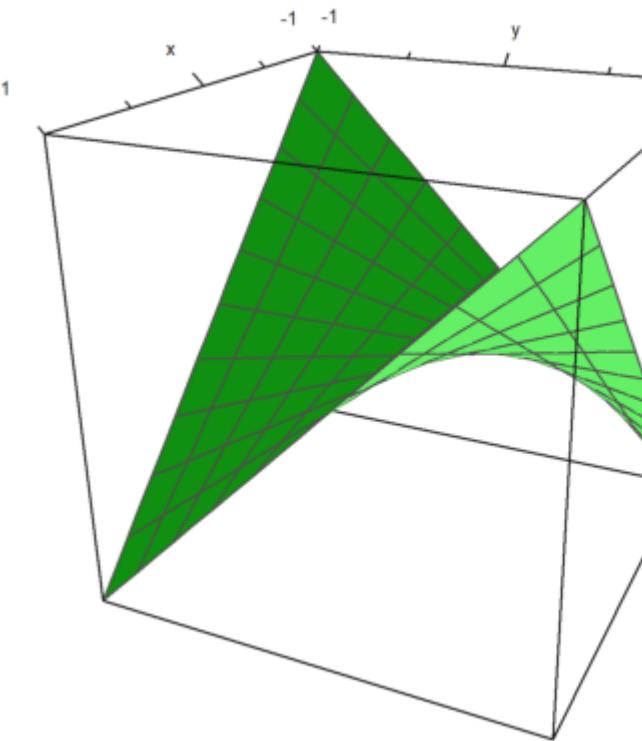
Dalam contoh berikut, kami menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar, kami dapat menandai wilayah, dalam kasus kami wilayah kutub.

```
t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
x=cos(s)cos(t); y=cos(s) * sin(t); z=sin(s); ...
plot3d(x,y,z, hue, ...
color=blue,frame,grid=[10,20], ...)
```

values=s,contourcolor=red,level=[90°-24°;90°-22°], ... scale=1.4,height=50°):



Berikut adalah contoh, yang merupakan grafik suatu fungsi.



```
t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,ts,grid=10);
```

However, we can make all sorts of surfaces. Here is the same surface as a function

```
plot3d(ts,t,s,angle=180°,grid=10);
```

Dengan usaha lebih, kita dapat menghasilkan banyak permukaan.

Dalam contoh berikut, kita membuat tampilan berbayang dari bola yang terdistorsi. Koordinat yang biasa untuk bola adalah

dengan

Kita mendistorsi ini dengan faktor

```
t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ... d=1+0.2(cos(4 ×
t)+cos(8s)); ... plot3d(cos(t)cos(s)d,sin(t)cos(s)d,sin(s)d,hue=1, ... light=[1,0,1],frame=0,zoom=5);
 $\gamma(t, s) = (\cos(t) \cos(s), \sin(t) \sin(s), \cos(s))$ 
```

Tentu saja, titik awan juga memungkinkan. Untuk memplot data titik di ruang, kita memerlukan tiga vektor untuk koordinat titik.

Gayanya sama seperti di plot2d dengan points=true;

```
n=500; ... plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=".");
```

Anda juga dapat memplot kurva dalam 3D. Dalam kasus ini, lebih mudah untuk menghitung titik-titik kurva terlebih dahulu. Untuk kurva dalam bidang,

kami menggunakan urutan koordinat dan parameter wire=true.

```
t=linspace(0,8pi,500); ... plot3d(sin(t),cos(t),t/10, wire,zoom=3);
t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi, wire, ... linewidth=3, wirecolor=blue);
X=cumsum(normal(3,100)); ... plot3d(X[1],X[2],X[3], anaglyph, wire);
```

EMT juga dapat membuat grafik dalam mode anaglif. Untuk melihat grafik tersebut, Anda memerlukan kacamata merah/sian.

```
plot3d("x2 + y3", anaglyph, contour, angle = 30) :
```

Seringkali, skema warna spektral digunakan untuk plot. Ini menekankan tinggi fungsi.

```
plot3d("x2y3 - y", spectral, contour, zoom = 3.2) :
```

Euler juga dapat memplot permukaan berparameter, ketika parameternya adalah nilai x, y, dan z dari gambar kotak persegi panjang di ruang tersebut.

Untuk demo berikut, kami menyiapkan parameter u dan v, dan menghasilkan koordinat ruang dari parameter tersebut.

```
u=linspace(-1,1,10); v=linspace(0,2pi,50)'; ... X=(3+ucos(v/2))cos(v);
Y=(3+ucos(v/2))sin(v); Z=usin(v/2); ... plot3d(X,Y,Z, anaglyph, iframe, wire, scale=2.3);
```

Berikut adalah contoh yang lebih rumit, yang tampak megah dengan kaca merah/cyan.

```
u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)'; ... x:=(4(1+.25sin(3 ×
v))+cos(u))cos(2v); ... y:=(4(1+.25sin(3v))+cos(u))sin(2v); ... z=sin(u)+2 ×
cos(3v); ... plot3d(x,y,z, frame=0, scale=1.5, hue=1, light=[1,0,-1], zoom=2.8, anaglyph);
```

3.6 Plot Statistik

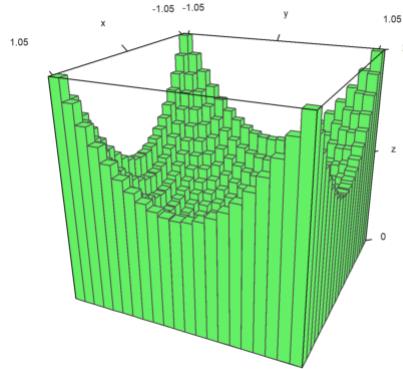
Plot batang juga dimungkinkan. Untuk ini, kita harus menyediakan

- * x: vektor baris dengan n+1 elemen
- * y: vektor kolom dengan n+1 elemen
- * z: matriks nilai nxn.

z dapat lebih besar, tetapi hanya nilai nxn yang akan digunakan.

Dalam contoh, pertama-tama kita menghitung nilai. Kemudian kita menye-suaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

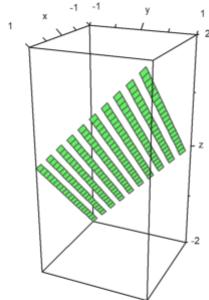
```
x=-1:0.1:1; y=x'; z=x2+y2; ... xa = (x|1.1)-0.05; ya = (y-1.1)-0.05; ... plot3d(xa, ya, z, bar =
```



true) :

Dimungkinkan untuk membagi bidang permukaan menjadi dua bagian atau lebih.

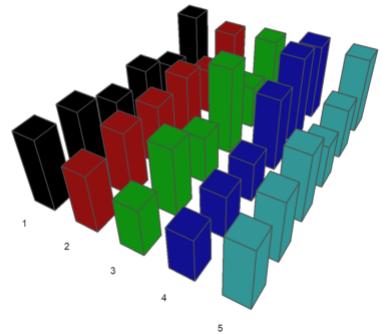
```
x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ... plot3d(x,y,z,disconnect=2:2:20);
```



Jika memuat atau membuat matriks data M dari sebuah file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan scale(M), atau menskalakan matriks dengan gt;zscale. Ini dapat dikombinasikan dengan faktor penskalaan individual yang diterapkan sebagai tambahan.

```
i=1:20; j=i'; ... plot3d(ij^2+100normal(20,20),zscale,scale=[1,1,1.5],angle=-40,zoom=1.8);
```

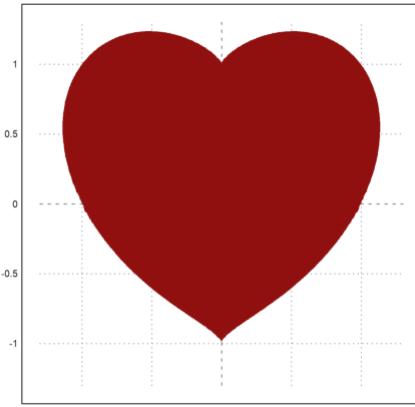
```
Z=intrandom(5,100,6); v=zeros(5,6); ... loop 1 to 5; v[] = getmultiplicities(1:6,Z[]);
```



```
end; ... columnsplot3d(v',scols=1:5,ccols=[1:5]):
```

3.7 Permukaan Benda Putar

```
plot2d("x^2+y^2-1)^3-x^2y^3",r = 1.3,...style = "",color = red,<outline,...level =
```



$[-2; 0], n = 100$:
 $1)^3 - x^2y^3$; ekspresi

$$\text{ekspresi} = (x^2 + y^2 -$$

Kita ingin memutar kurva jantung di sekitar sumbu y. Berikut ini adalah ekspresi yang mendefinisikan jantung:

Selanjutnya kita tetapkan

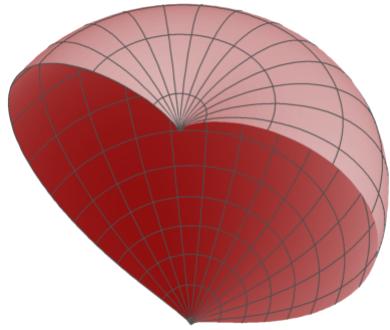
```
function fr(r,a) = ekspresi with [x=rcos(a),y=rsin(a)] — trigreduce; fr(r,a)
```

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang memecahkan r, jika a diberikan. Dengan fungsi itu kita dapat memplot jantung yang diputar sebagai permukaan parametrik.

```
function map f(a) := bisect("fr",0,2;a); ... t=linspace(-pi/2,pi/2,100);  

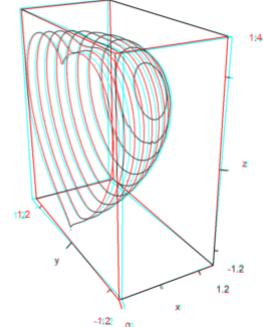
r=f(t); ... s=linspace(pi,2pi,100)'; ... plot3d(rcos(t)sin(s),rcos(t)cos(s),r ×  

sin(t), ... hue,frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```



Berikut ini adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kami mendefinisikan fungsi yang menggambarkan objek tersebut.

```
function f(x,y,z) ...
r=x^2+y^2;return(r+z^2-1)^3-r*z^3;endfunction </pre>
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) :
```

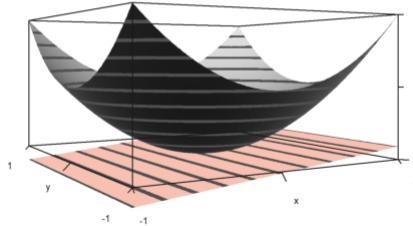
3.8 Plot 3D Khusus

Fungsi plot3d memang bagus, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda dapat memperoleh plot berbingkai dari objek apa pun yang Anda suka.

Meskipun Euler bukanlah program 3D, ia 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,yscale,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 </pre>
Sekarang framedplot() menyediakan bingkai dan mengatur tampilan.
```

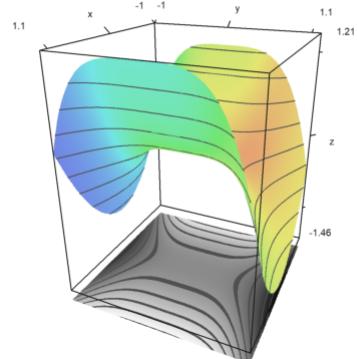
```
framedplot("myplot",[-1,1,-1,1,0,1],height=0,angle=-30°, ... center=[0,0,-
```



0.7],zoom=3):

Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa `plot3d()` menetapkan jendela ke `fullwindow()` secara default, tetapi `plotcontourplane()` mengasumsikannya.

```
x=-1:0.02:1.1; y=x'; z=x^2 - y^4;
function myplot (x,y,z) ... jpre class="udf" ; zoom(2); wi=fullwindow();
plotcontourplane(x,y,z,level="auto",jscale); plot3d(x,y,z,hue,jscale,add,color=white,level="thin");
window(wi); reset(); endfunction j/pre;
```



`j/pre`

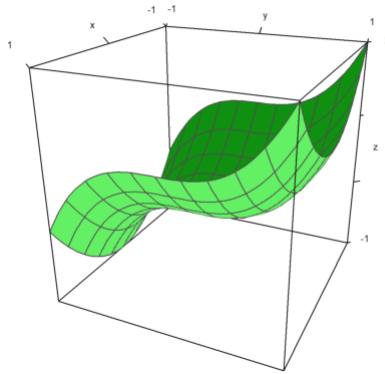
3.9 Animasi

Euler dapat menggunakan bingkai 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. Terakhir, fungsi ini menganimasikan plot tersebut.

Silakan pelajari sumber `rotate` untuk melihat detail selengkapnya.

```
function testplot () := plot3d("x2 + y3");... rotate("testplot");testplot() :
```



3.10 Menggambar Povray

Dengan bantuan file Euler povray.e, Euler dapat membuat file Povray. Hasilnya sangat bagus untuk dilihat.

Anda perlu menginstal Povray (32bit atau 64bit) dari <http://www.povray.org/>, dan meletakkan subdirektori "bin" dari Povray ke dalam jalur lingkungan, atau mengatur variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe". <http://www.povray.org/>, dan meletakkan subdirektori "bin" dari Povray ke dalam jalur lingkungan, atau mengatur variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe".

Antarmuka Povray dari Euler membuat file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah current.pov, dan direktori default adalah eulerhome(), biasanya c:. Povray membuat file PNG, yang dapat dimuat oleh Euler ke dalam buku catatan. Untuk membersihkan file-file ini, gunakan povclear().

Fungsi pov3d memiliki semangat yang sama dengan plot3d. Fungsi ini dapat menghasilkan grafik fungsi $f(x,y)$, atau permukaan dengan koordinat X,Y,Z dalam matriks, termasuk garis level opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat adegan ke dalam buku catatan 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 adegan. Terakhir, akhiri file dengan povend(). Secara default, raytracer akan mulai, dan PNG akan dimasukkan ke dalam buku catatan Euler.

Fungsi objek memiliki parameter yang disebut "look", yang memerlukan string dengan kode Povray untuk tekstur dan penyelesaian objek. Fungsi povlook() dapat digunakan untuk menghasilkan string ini. Fungsi ini memiliki parameter untuk warna, transparansi, Phong Shading, dll.

Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk vertikal ke atas, dan sumbu x, y, z dalam arah kanan.

Anda perlu memuat berkas povray.

```
load povray;
```

Pastikan direktori bin Povray ada di jalur tersebut. Jika tidak, edit variabel berikut sehingga berisi jalur ke povray yang dapat dieksekusi.

```
defaultpovray="C:
```

Program Files

POV-Ray

v3.7

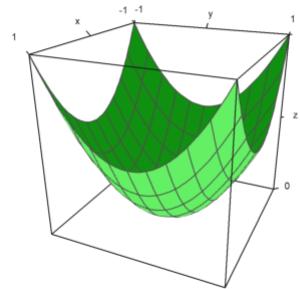
bin

pengine.exe"

C:\Files-Ray3.7.exe

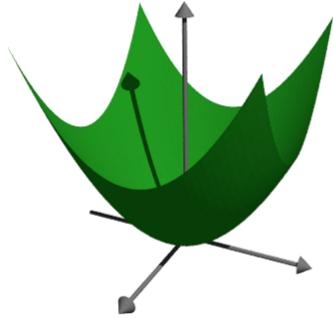
Untuk kesan pertama, kami membuat fungsi sederhana. Perintah berikut menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk melakukan ray tracing pada file ini.

Jika Anda menjalankan perintah berikut, GUI Povray akan terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya apakah Anda ingin mengizinkan file exe untuk berjalan. Anda dapat menekan batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK di jendela Povray untuk mengakui dialog awal Povray.



plot3d("x²+y²", zoom = 2) :

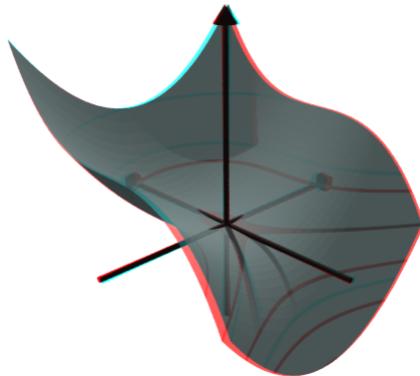
pov3d("x²+



$y^2", zoom = 3);$

Kita dapat membuat fungsi tersebut transparan dan menambahkan penyelesaian lainnya. Kita juga dapat menambahkan garis level pada plot fungsi.

`pov3d("x2+y3", axiscolor = red, angle = -45, anaglyph, ...look = povlook(cyan, 0.2), level =`

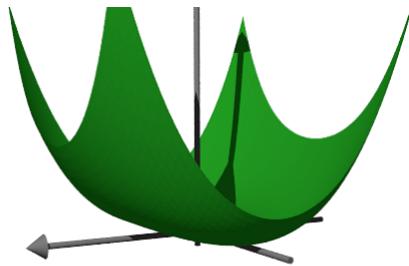


$-1 : 0.5 : 1, zoom = 3.8);$

Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi secara manual.

Kami memplot himpunan 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);

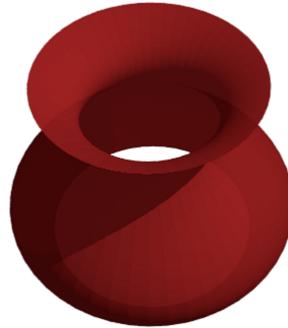
3.11 Membuat Plot dengan Koordinat

Alih-alih menggunakan fungsi, kita dapat membuat plot dengan koordinat. Seperti pada plot3d, kita memerlukan tiga matriks untuk menentukan objek.

Dalam contoh ini, kita memutar fungsi di sekitar sumbu z.

```
function f(x) := x^3 - x + 1; ... x = -1 : 0.01 : 1; t = linspace(0, 2pi, 50)'; ... Z =
x; X = cos(t)f(x); Y = sin(t)f(x); ... pov3d(X, Y, Z, angle = 40, look = povlook(red, 0.1), height =

```

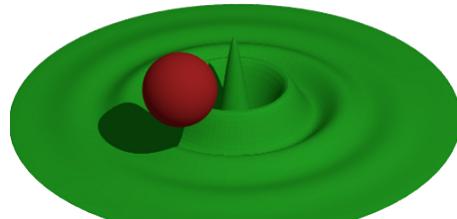


50, axis = 0, zoom = 4, light = [10, 5, 15]);

Dalam contoh berikut, kami memplot gelombang yang direndam. 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=rcos(phi); y=rsin(phi);
z=exp(-5r)cos(8pir)/3; ... pov3d(x,y,z,zoom=6, axis=0, height=30°, add=povsphere([0.5,0,0.25], 0.15, povlook(r,
```



... w=500,h=300);

Dengan metode shading Povray yang canggih, hanya sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya pada batas dan bayangan, triknya mungkin menjadi jelas.

Untuk ini, kita perlu menambahkan vektor normal di setiap titik matriks.

$$Z = x^2y^3$$

$$2 \ 3 \ x \ y$$

Persamaan permukaannya adalah $[x,y,Z]$. Kita hitung dua turunan x dan y dari persamaan ini dan ambil perkalian silang sebagai normalnya.

$$dx = \text{diff}([x,y,Z],x); dy = \text{diff}([x,y,Z],y);$$

Kami mendefinisikan normal sebagai perkalian silang turunan-turunan ini dan mendefinisikan fungsi koordinat.

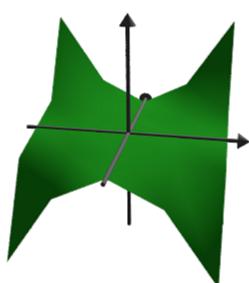
$$N = \text{crossproduct}(dx,dy); NX = N[1]; NY = N[2]; NZ = N[3]; N,$$

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

Kami hanya menggunakan 25 poin.

$$x=-1:0.5:1; y=x';$$

$$\text{pov3d}(x,y,Z(x,y),\text{angle}=10^\circ, \dots \ xv=NX(x,y),yv=NY(x,y),zv=NZ(x,y),\text{ishadow});$$



Berikut ini adalah simpul Trefoil yang dibuat oleh A. Busser di Povray. Ada versi yang lebih baik dari simpul ini dalam contoh-contohnya.

Lihat: ContohTrefoil — Simpul Trefoil

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normal bagi kami. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolik.

```

X = ((4+sin(3y))+cos(x))cos(2y); ...    Y = ((4+sin(3y))+cos(x))sin(2y);
...  Z = sin(x)+2cos(3y);
Kemudian dua vektor turunan ke x dan y.
dx = diff([X,Y,Z],x); dy = diff([X,Y,Z],y);
Sekarang normal, yang merupakan perkalian silang dari dua turunan.
dn = crossproduct(dx,dy);
Sekarang mari kita evaluasi semua ini secara numerik.
x:=linspace(-
Vektor normal adalah evaluasi ekspresi simbolik dn[i] untuk i=1,2,3. Sintaks
untuk ini adalah amp;"ekspresi"(parameter). Ini adalah alternatif untuk 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, ...
jshadow,look=povlook(blue), ... xv="dn[1]"(x,y), yv="dn[2]"(x,y), zv="dn[3]"(x,y));

```

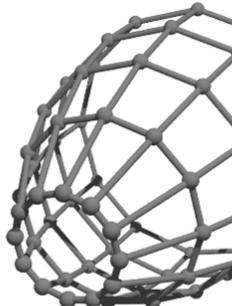


Kita juga dapat membuat grid dalam 3D.

```

povstart(zoom=4); ...  x=-1:0.5:1; r=1-(x+1)^2/6; ... t = (0 : 30 : 360)'; y =

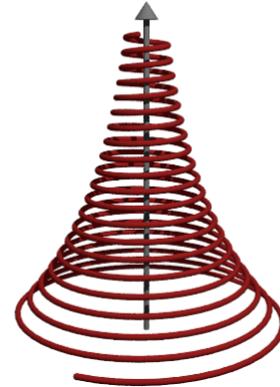
```



```

rcos(t); z = rsin(t); ... writeln(povgrid(x, y, z, d = 0.02, dballs = 0.05)); ... povend();
Dengan povgrid(), kurva dimungkinkan.
povstart(center=[0,0,1],zoom=3.6); ... t=linspace(0,2,1000); r=exp(-t); ...
x=cos(2pi10t)r; y=sin(2pi10t)r; z=t; ... writeln(povgrid(x,y,z,povlook(red)));

```



```
... writeAxis(0,2,axis=3); ... povend();
```

3.12 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 berkas Povray.

Kami memulai output dengan povstart().

```
povstart(zoom=4);
```

Pertama, kita mendefinisikan tiga silinder, dan menyimpannya dalam 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)); ... Rangkaian tersebut berisi kode Povray, yang tidak perlu kita pahami saat itu.
```

```
c2
```

```
cylinder lt;0,0,-1gt;, lt;0,0,1gt;; 1 texture pigment color rgb lt;0.941176,0.941176,0.392157gt;
finish ambient 0.2
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.

Hal itu dilakukan oleh povlook(), yang mengembalikan string dengan kode Povray yang relevan. Kita dapat menggunakan warna Euler default, atau menentukan warna kita sendiri. Kita juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

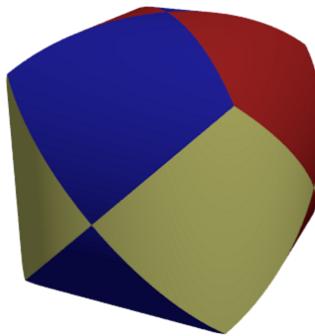
```
povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

```
texture pigment color rgbf lt;0.101961,0.2,0.301961,0.1gt; finish ambient 0.5
```

Sekarang kita mendefinisikan objek persimpangan dan menulis hasilnya ke berkas.

```
writeln(povintersection([c1,c2,c3]));
```

Persimpangan tiga silinder sulit dibayangkan, jika Anda belum pernah melihatnya sebelumnya.

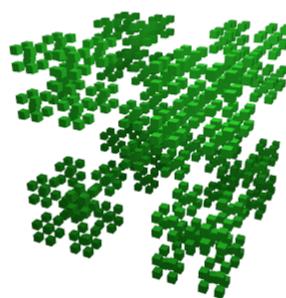


povend;

Fungsi-fungsi berikut menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan bagaimana Euler menangani objek-objek Povray sederhana. Fungsi povbox() mengembalikan 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) ... i/pre class="udf" ; 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 i/pre; povstart(fade=10,ishadow);  
fractal(-1,-1,1,2,4);
```



povend();

Perbedaan memungkinkan pemisahan satu objek dari objek lainnya. Seperti halnya persimpangan, ada beberapa objek CSG dari Povray.

```
povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kami mendefinisikan objek dalam Povray, alih-alih

menggunakan string dalam Euler. Definisi langsung ditulis ke berkas.

Koordinat kotak -1 berarti [-1,-1,-1].

```
povdefine("mycube",povbox(-1,1));
```

Kita dapat menggunakan objek ini dalam povobject(), yang mengembalikan string seperti biasa.

```
c1=povobject("mycube",povlook(red));
```

Kita buat kubus kedua, lalu putar dan ubah skalanya sedikit.

```
c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ... rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Lalu kita ambil selisih kedua benda 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();
```

3.13 Fungsi Implisit

Povray dapat memplot himpunan di mana $f(x,y,z)=0$, sama seperti parameter implisit dalam plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsi-fungsi tersebut 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();
povstart(angle=70°,height=50°,zoom=4);
Buat permukaan implisit. Perhatikan sintaksis yang berbeda dalam ekspresi.
writeln(povsurface("pow(x,2)y-pow(y,3)-pow(z,2)",povlook(green))); ... writeAxes();
... povend();
```

3.14 Objek Mesh

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.

Kami ingin memaksimalkan xy dalam kondisi $x+y=1$ dan menunjukkan sentuhan tangensial garis-garis level.

```
povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kita tidak dapat menyimpan objek dalam string seperti sebelumnya, karena terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan declare. Fungsi povtriangle() melakukan ini secara otomatis. Fungsi ini dapat menerima vektor normal seperti pov3d().

Berikut ini mendefinisikan objek mesh, dan langsung menuliskannya ke dalam file.

```
x=0:0.02:1; y=x'; z=xy; vx=-y; vy=-x; vz=1;  
mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kita definisikan 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 dua titik potongnya.

```
writeln(povintersection([mesh,cl],povlook(red))); ... writeln(povintersection([mesh,ll],povlook(gray)));  
Tuliskan titik maksimumnya.
```

```
writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2defaultpointsize));
```

Tambahkan sumbu dan selesaikan.

```
writeAxes(0,1,0,1,0,1,d=0.015); ... povend();
```

Anaglif dalam Povray

Untuk menghasilkan anaglif untuk kacamata merah/sian, Povray harus dijalankan dua kali dari posisi kamera yang berbeda. Ia menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph().

Tentu saja, Anda memerlukan kacamata merah/sian untuk melihat contoh berikut dengan benar.

Fungsi pov3d() memiliki sakelar sederhana untuk menghasilkan anaglif.

```
pov3d("-exp(-x2-y2)/2",r = 2,height = 45,anaglyph,...center = [0,0,0.5],zoom = 3.5);
```

Jika Anda membuat suatu pemandangan dengan objek, Anda perlu memasukkan pembuatan pemandangan tersebut ke dalam suatu fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

```
function myscene ...
```

```
s=povsphere(povc,1); cl=povcylinder(-povz,povz,0.5); clx=povobject(cl,rotate=xrotate(90°));  
cly=povobject(cl,rotate=yrotate(90°)); c=povbox([-1,-1,0],1); un=povunion([cl,clx,cly,c]);  
obj=povdifference(s,un,povlook(red)); writeln(obj); writeAxes(); endfunction  
j/pre; Fungsi povanaglyph() melakukan semua ini. Parameternya seperti pada povstart() dan povend() yang digabungkan.
```

```
povanaglyph("myscene",zoom=4.5);
```

3.15 Menentukan Objek Sendiri

Antarmuka povray Euler berisi banyak objek. Namun, Anda tidak terbatas pada objek-objek ini. Anda dapat membuat objek sendiri, yang menggabungkan

objek lain, atau objek yang sama sekali baru.

Kami mendemonstrasikan sebuah torus. Perintah Povray untuk ini adalah "torus". Jadi, kami mengembalikan string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat di titik asal.

```
function povdonat (r1,r2,look="") ...
return "torus "+r1+"," +r2+look+""; endfunction ;/pre;
Inilah torus pertama kita.
```

```
t1=povdonat(0.8,0.2)
```

```
torus 0.8,0.2
```

Mari kita gunakan objek ini untuk membuat torus kedua, diterjemahkan dan diputar.

```
t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
object torus 0.8,0.2 rotate 90 *x translate lt;0.8,0,0gt;
```

Sekarang kita tempatkan objek-objek ini ke dalam sebuah scene. Untuk tampilannya, kita gunakan 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))); ... gt;povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, program tersebut tidak menampilkan kesalahan tersebut. Oleh karena itu, Anda harus menggunakan

```
gt;povend(lt;exit);
```

jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray terbuka.
povend(h=320,w=480);

Berikut adalah contoh yang lebih rinci. Kami memecahkan
dan menunjukkan titik-titik yang layak dan titik-titik optimum dalam plot
3D.

```
A=[10,8,4;5,6,8;6,3,2;9,5,6];
```

```
b=[10,10,10,10]';
```

```
c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini punya solusi.

```
x=simplex(A,b,c, max, check)'
```

```
[0, 1, 0.5]
```

Ya, benar.

Berikutnya kita mendefinisikan dua objek. Yang pertama adalah bidang datar

```
function oneplane (a,b,look="") ...
return povplane(a,b,look) endfunction ;/pre;

```

Kemudian kita mendefinisikan irisan semua ruang setengah dan sebuah kubus.

```
function adm (A, b, r, look="") ...
ol=[]; loop 1 to rows(A); ol=ol—oneplane(A[],b[]); end; ol=ol—povbox([0,0,0],[r,r,r]);
```

return povintersection(ol,look); endfunction ;/pre;
Sekarang, kita dapat merencanakan adegannya.

```
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 sekitar
```

titik optimum.

```
writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ... povlook(red,0.9)));
```

Dan kesalahan dalam arah yang optimum.

```
writeln(povarrow(x,c0.5,povlook(red)));
```

Kita menambahkan teks ke layar. Teks hanyalah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

```
writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=5)); ... povend();
```

Contoh Lainnya

Anda dapat menemukan beberapa contoh lainnya untuk Povray di Euler dalam berkas berikut.

Lihat: Contoh/Bola Dandelin

Lihat: Contoh/Donat Math

Lihat: Contoh/Simpul Trefoil

Lihat: Contoh/Optimasi dengan Penskalaan Afinitas

Contoh Soal Diberikan fungsi kuadrat $g(x) := x^2 - 2x + 3$

lakukan visualisasi grafik 3D dari fungsi tersebut dengan langkah-langkah berikut:

1. Definisikan fungsi $g(x)$ dan variabel $x = -2:0.02:2$ serta $u = \text{linspace}(0, 2\pi, 100)'$.

2. Buat grid 3D berdasarkan variabel:

$Z = x$,

$X = \cos(u) \cdot g(x)$,

$Y = \sin(u) \cdot g(x)$.

Definisikan fungsi $g(x)$

```
function g(x) := x2 - 2x + 3;
```

Rentan nilai x dan u

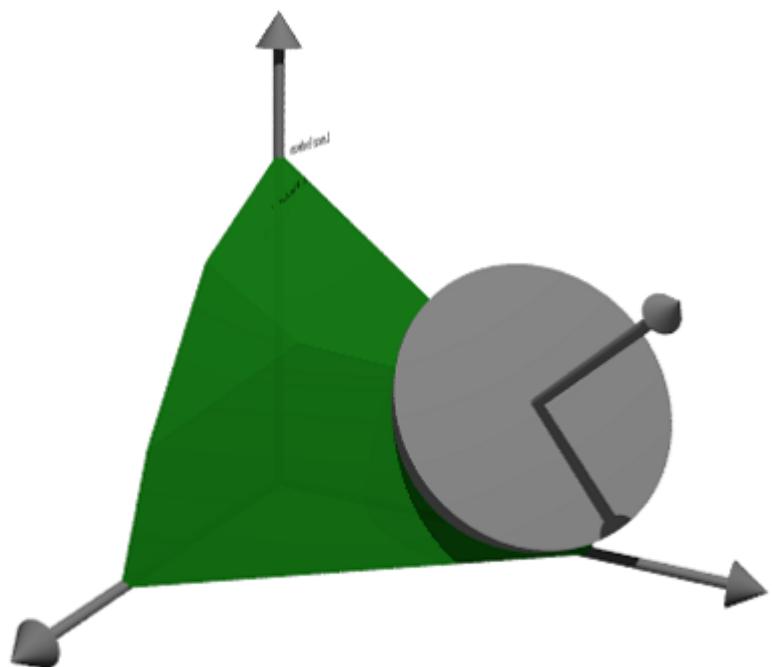
```
x = -2:0.02:2;
```

```
u = linspace(0, 2pi, 100)';
```

Buat grid untuk X dan U

```
Z = x;
```

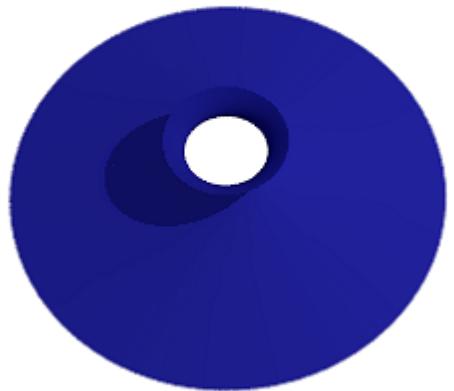
```
X = cos(u)g(x);
```



$$Y = \sin(u)g(x);$$

Visualisasi 3D

pov3d(X, Y, Z, angle=30°, look=povlook(blue, 0.2), height=60°, axis=0,



zoom=3, light=[15,10,20]);

4 Kalkulus

4.1 Kalkulus dengan EMT

Materi Kalkulus mencakup di antaranya:

- * Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, * komposisi fungsi)
- * Limit Fungsi,
- * Turunan Fungsi,
- * Integral Tak Tentu,
- * Integral Tentu dan Aplikasinya,
- * Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

4.2 Mendefinisikan Fungsi

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:

- * Menggunakan format $\text{nama}_f\text{ungsi} := \text{rumusfungsi}(\text{untuk fungsi*numerik}),$
- * Menggunakan format $\text{nama}_f\text{ungsi}amp; = \text{rumusfungsi}(\text{untuk fungsi * simbolik}, \text{namun dapat dihitung secara numerik}),$
- * Menggunakan format $\text{nama}_f\text{ungsi}amp; amp; = \text{rumusfungsi}(\text{untuk fungsi* simbolik murni}, \text{tidak dapat dihitung langsung}),$
- * Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah function (bukan sebagai ekspresi).

Berikut adalah beberapa contoh cara mendefinisikan fungsi:

```
function f(x) := 2x2 + exp(sin(x))//fungsinumerik
```

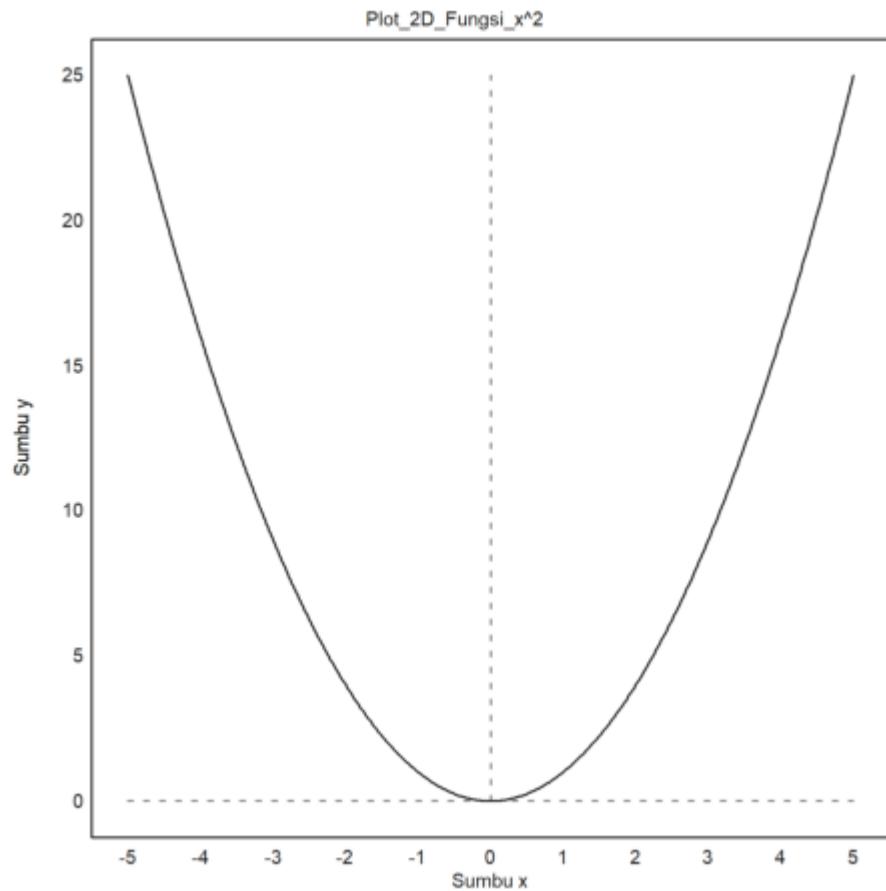
```
f(0), f(1), f(pi)
```

```
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!



```
plot2d("2x^2+exp(sin(x))") :
```

Berikutnya kita definisikan fungsi:

```
function g(x) := sqrt(x^2 - 3x)/(x + 1)
```

```
g(3)
```

```
0
```

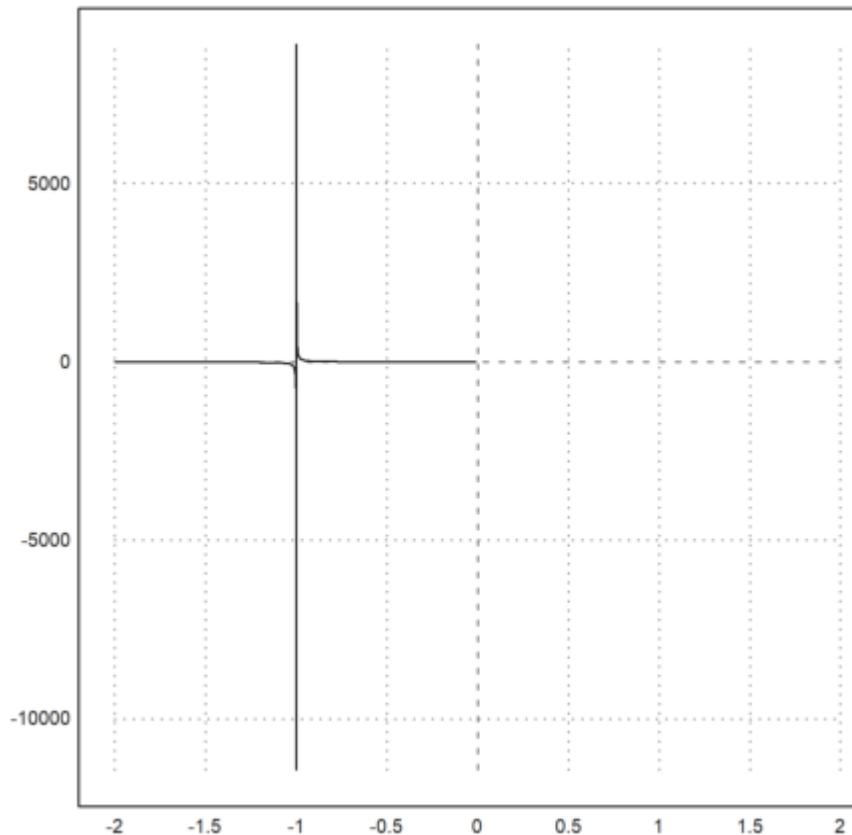
```
g(0)
```

```
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(x² - 3 * x)/(x + 1)
Error in : g(1)//kompleks, tidak dapat dihitung oleh fungsi numerik...

Silakan Anda plot kurva fungsi di atas!



```

plot2d("sqrt(x^2-3x)/(x+1)") :
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 x3 else return x2 endif; endfunction </pre>
```

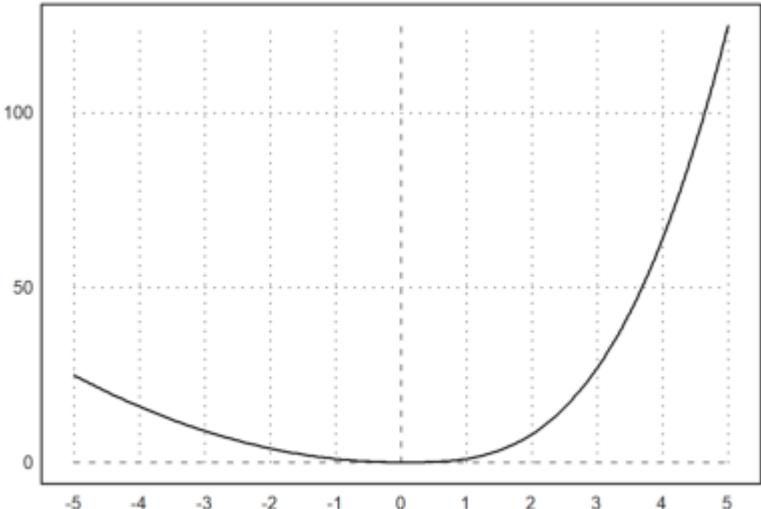
```
1
```

```
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) = 2Ex // fungsi simbolik
x 2 E
f(a) // nilai fungsi secara simbolik
f(E) // nilai fungsi berupa bilangan desimal
30.308524483
f(E),float(
function g(x) = 3x+1
3 x + 1
function h(x) = f(g(x)) // komposisi fungsi
3 x + 1 2 E
plot2d("h(x)", -1, 1);
```

4.3 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.

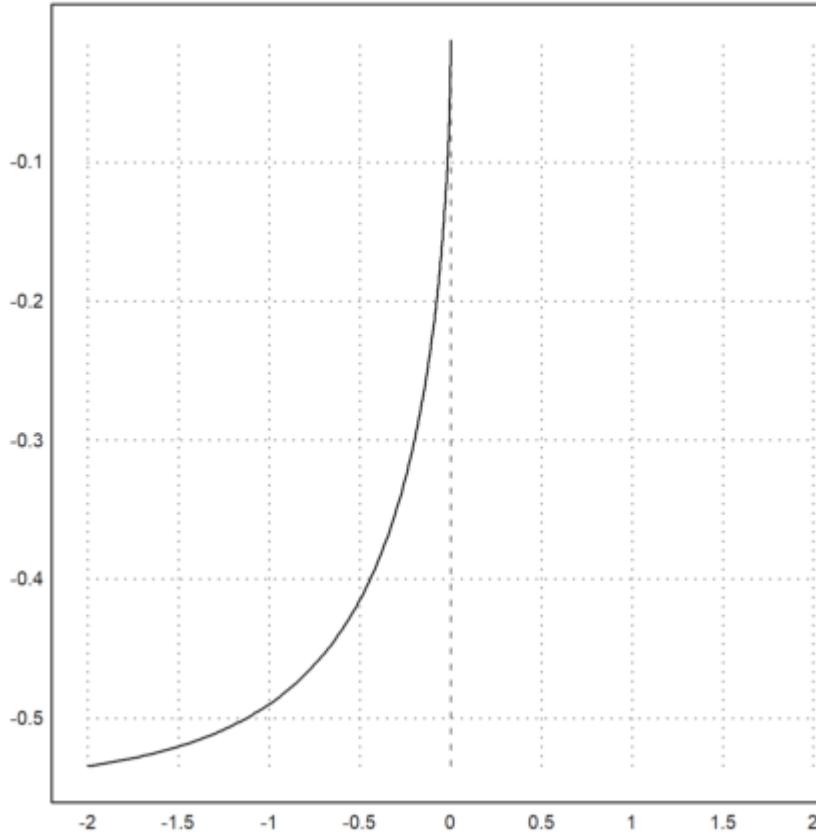
1. Misalkan kita punya fungsi aljabar:

cari $g(0)$, $g(5)$ dan buatkan plot 2d

function $g(x) := \sqrt{x^2 - 5x} / (2x - 3)$

$g(0)$, $g(5)$

0 0



plot2d("sqrt(x^2-5x)/(2x-3)") :

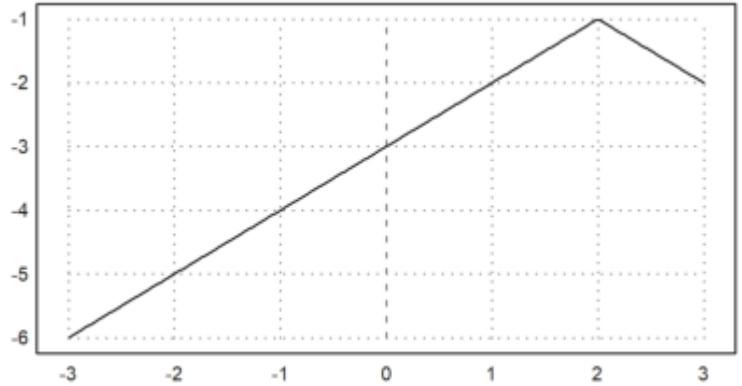
2.

function $g(x) := (\sqrt{x^2 + 36}) // fungsinumerik$
 $g(6)$

```

8.48528137424
g=(-3:3)
[-3, -2, -1, 0, 1, 2, 3]
aspect(3); plot2d("g(x)",-3,3):
3.
function d(x) := ((x^2 + 6)/(2x))//fungsinumerik
d(2)
2.5
d=(-2:2)
[-2, -1, 0, 1, 2]
aspect(3); plot2d("d(x)",-2,2):
4.
function f(x) = (cos(x)) // fungsi numerik
cos(x)
f(pi)
-1
f(3pi)
-1
function g(x) = (sin(x)) // fungsi numerik
sin(x)
g(pi)
0
g(2pi)
0
f(g(pi)) // komposisi fungsi
1
g(f(pi))
-0.841470984808
function h(x) = f(g(pi))
1
plot2d("h(x)",-4,4):
5.
function map e(x) ...
if x<2 then return x-3 else return 1-x endif;e(5) endfunction i/prei_ e(5)
-4
e=(-3:3)
[-3, -2, -1, 0, 1, 2, 3]

```



```
aspect(2); plot2d("e(x)",-3,3);
```

4.4 Menghitung Limit

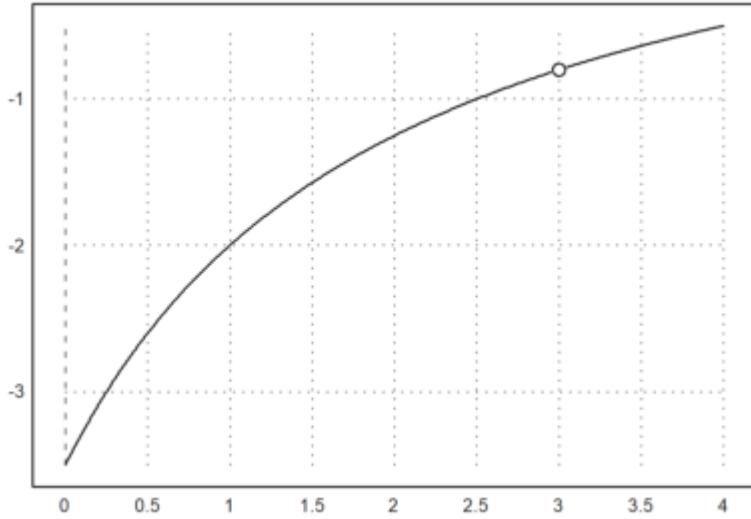
Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, 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 - 3x)/(x + 1), x, inf))
limit((x^3 - 13x^2 + 51x - 63)/(x^3 - 4x^2 - 3x + 18), x, 3)
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.

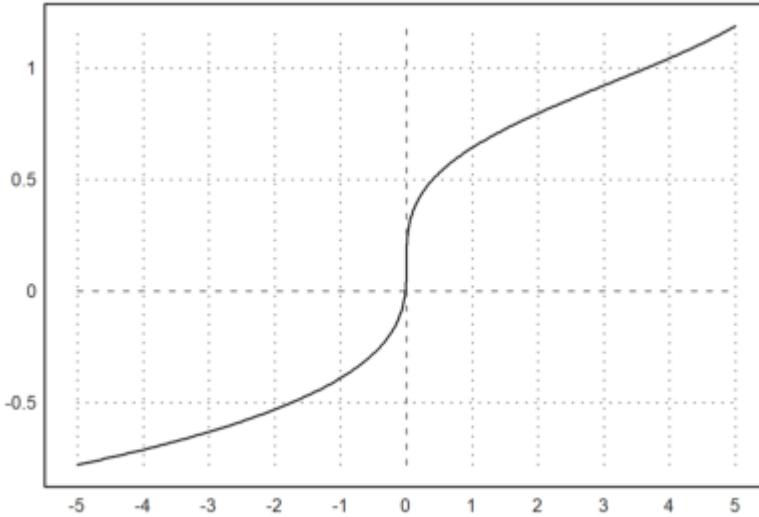
```
aspect(1.5); plot2d("(x^3-13x^2+51x-63)/(x^3-4x^2-3x+18)", 0, 4); plot2d(3, -4/5, points, style =
```



```

"ow", add) : limit(2 ×
xsin(x)/(1-cos(x)),x,0)
maxima: 'limit(2*x*sin(x)/(1-cos(x)),x,0)=limit(2*x*sin(x)/(1-cos(x)),x,0)
Fungsi tersebut diskontinu di titik x=0. Berikut adalah grafik fungsinya.
plot2d("2xsin(x)/(1-cos(x))",-pi,pi); plot2d(0,4, points,style="ow", add):
limit(cot(7h)/cot(5h), h, 0)
maxima: showev('limit(cot(7*h)/cot(5*h),h,0))
Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di x=0. Berikut
adalah grafiknya.
plot2d("cot(7x)/cot(5x)",-0.001,0.001); plot2d(0,5/7, points,style="ow", add):
showev('limit(((x/8)^(1/3) - 1)/(x - 8), x, 8))
Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.
plot2d("(x/8)^(1/3)-1/(x-8)",-5,5); plot2d(8, 1/24, points, style = "ow", add) :

```



showev('limit(1/(2 ×
x-1),x,0))

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

plot2d("1/(2x-1)",-5,5); plot2d(0,-1, points,style="ow", add):

showev('limit((x² - 3x - 10)/(x - 5),x,5))

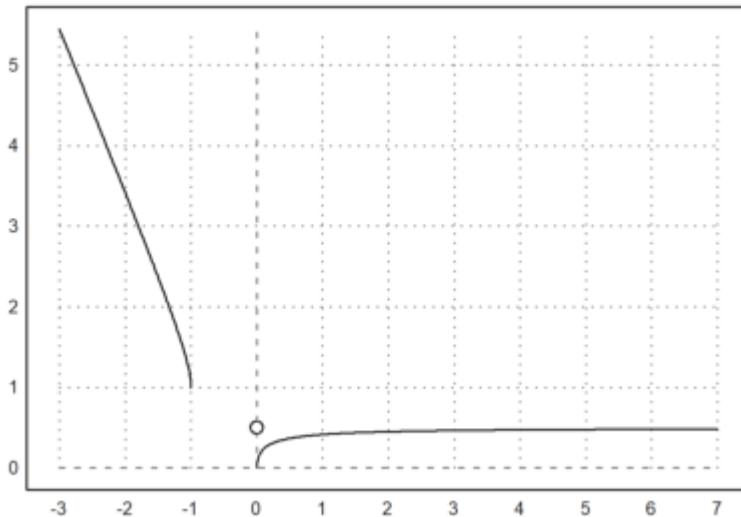
Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

plot2d("(x²-3x-10)/(x-5)",0,10); plot2d(5,7, points, style = "ow", add) :

showev('limit(sqrt(x² + x) - x,x,inf))

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

plot2d("sqrt(x² + x) - x",-3,7); plot2d(0,1/2, points, style = "ow", add) :



showev('limit(abs(x-

```
1)/(x-1),x,1,minus))
```

Hitung limit di atas untuk x menuju 1 dari kanan.

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

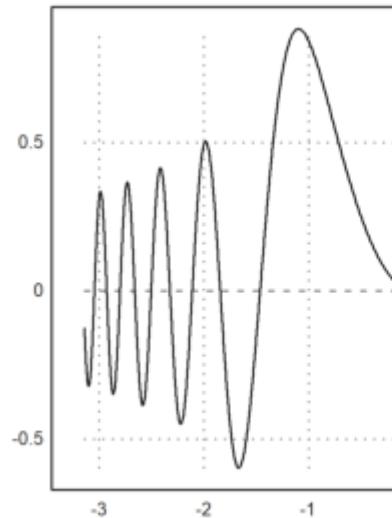
```
plot2d("abs(x-1)/(x-1)",-1,5); plot2d(1,-1, points,style="ow", add):
```

```
showev('limit(sin(x)/x,x,0))
```

```
plot2d("sin(x)/x",-pi,pi); plot2d(0,1, points,style="ow", add):
```

```
showev('limit(sin(x^3)/x,x,0))
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.



```
plot2d("sin(x^3)/x",-pi,pi); plot2d(0,0, points, style = "ow", add) :  
x, minf))
```

```
showev('limit((-2)^x,x,infinity))
```

```
showev('limit(t - sqrt(2 - t),t,2,minus))
```

```
showev('limit(t - sqrt(2 - t),t,2,plus))
```

```
showev('limit(t - sqrt(2 - t),t,5,plus))//Perhatikan hasilnya
```

```
plot2d("x-sqrt(2-x)",0,2):
```

```
showev('limit((x^2 - 9)/(2x^2 - 5x - 3),x,3))
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
plot2d("(x^2 - 9)/(2x^2 - 5x - 3)",0,7) :
```

```
showev('limit((1 - cos(x))/x,x,0))
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
plot2d("((1-cos(x))/x)",-1,7):
```

```
showev('limit((x^2 + abs(x))/(x^2 - abs(x)),x,0))
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
plot2d("(x^2 + abs(x))/(x^2 - abs(x))",-1,6) :
```

```
showev('limit((1 + 1/x)^x,x,infinity))
```

```
plot2d("(1+1/x)^x",0,1000) :
```

```
showev('limit((1 + k/x)^x,x,infinity))
```

```
showev('limit((1 + x)^(1/x),x,0))
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

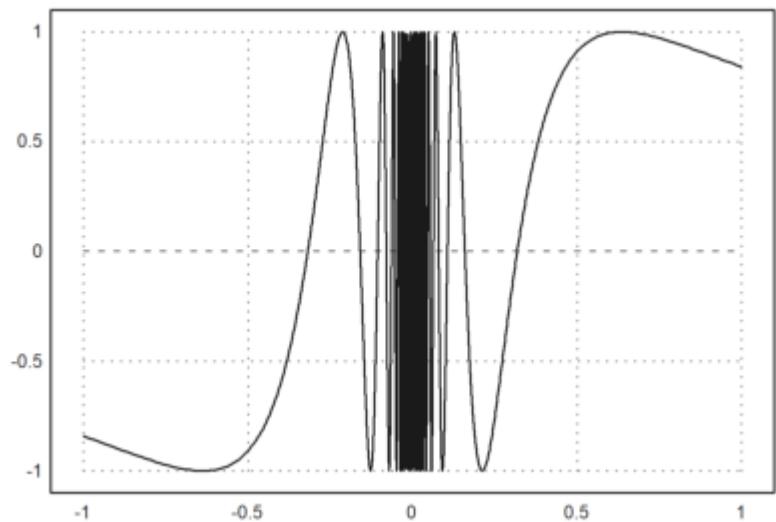
plot2d("\$(1+x)^{1/x}\$", 0, 10) :

```
showev('limit((x/(x + k))x, x, inf))  
showev('limit((Ex - E2)/(x - 2), x, 2))
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

plot2d("\$(E^x - E^2)/(x - 2)\$", 0, 10) :

```
showev('limit(sin(1/x), x, 0))  
showev('limit(sin(1/x), x, inf))
```



plot2d("sin(1/x)", -1, 1):

4.5 Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

1.

```
showev('limit((x2 + 2x + 6), x, 3))  
showev('limit((x2 + 2x + 6), x, 2))  
showev('limit((x2 + 2x + 6), x, inf))  
plot2d("$(x^2 + 2x + 6)$", -6, 6); plot2d(3, 21, points, style = "ow", add) :
```

2.

```
showev('limit((x4 + 2x3 - x2)/(x2), x, 0))  
showev('limit((x4 + 2x3 - x2)/(x2), x, 2))  
showev('limit((x4 + 2x3 - x2)/(x2), x, inf))  
plot2d("$(x^4 + 2x^3 - x^2)$", -2, 2); plot2d(0, 0, points, style = "ow", add) :
```

3.

```

showev('limit(sqrt(x + 16), x, inf))
showev('limit(sqrt(x + 16), x, 0))
showev('limit(sqrt(x + 16), x, 4))
plot2d("sqrt(x+16)", -7, 7); plot2d(0, 4, points, style="ow", add):
4.
showev('limit(((x^2) + (2x)), x, 0, plus))
showev('limit(((x^2) + (2x)), x, 2, minus))
showev('limit(((x^2) + (2x)), x, inf))
plot2d("((x^2) + (2x))", 0, 8) :
5.
showev('limit((1 - cos(x))/x^2, x, 0))
showev('limit((1 - cos(x))/x^2, x, inf))
showev('limit((1 - cos(x))/x^2, x, 2))
plot2d("(1-cos(x))/x^2", -3, 3) :

```

4.6 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))//turunanx^2
p = expand((x+h)^2 - x^2)|simplify;p //pembilang dijabarkan dan disederhanakan
q = ratsimp(p/h); q//ekspresiyangakandihitunglimitnyadisederhanakan
limit(q, h, 0)//nilailimitsebagaiaturunan
showev('limit(((x + h)^n - x^n)/h, h, 0))//turunanx^n

```

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))//turunansin(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))//turunanlog(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))//turunan1/x
showev('limit((E^(x + h) - E^x)/h, h, 0))//turunansf(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?

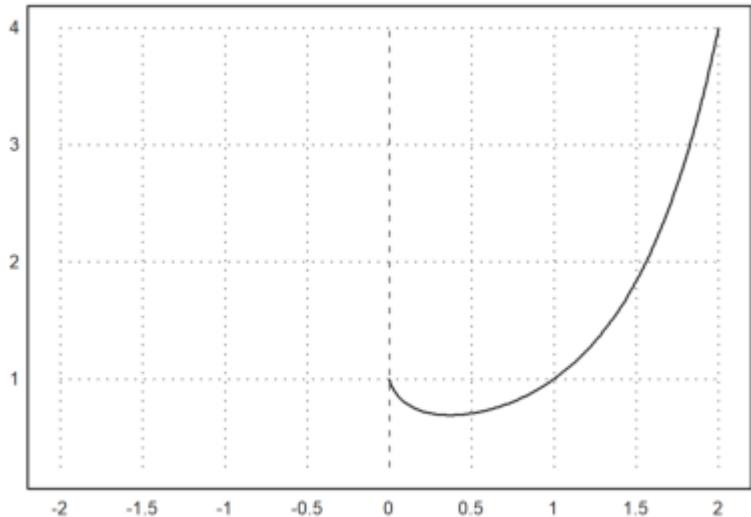
Use assume! Error in: `showev('limit((E^(x+h)-E^x)/h,h,0))//turunanf(x) = e^x...`

Maxima bermasalah dengan limit:

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
showev('limit((E^h - 1)/h, h, 0))
showev('factor(E^(x + h) - E^x))
showev('limit(factor((E^(x + h) - E^x)/h), h, 0))//turunanf(x) = e^x
function f(x) = x^x
x x
showev('limit(f(x), x, 0))
```

Silakan Anda gambar kurva



`plot2d("x^x", -2, 2) :`

`showev('limit((f(x + h) - f(x))/h, h, 0))//turunanf(x) = x^x`

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))//turunanf(x) = x^x`

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Fungsi

yang membantu dalam proses turunan menggunakan aturan rantai:

kemudian, menggunakan aturan rantai untuk turunan:

Dengan demikian, hasil akhirnya adalah

yang mengkonfirmasi bahwa limit yang ada pada gambar tersebut benar setelah asumsi $x > 0$ ditetapkan.

```
forget(x 0) // jangan lupa, lupakan asumsi untuk kembali ke semula
[x gt; 0]
forget(x>0)
[x lt; 0]
facts()
[]

showev('limit((asin(x + h) - asin(x))/h, h, 0))//turunanarcsin(x)
```

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))//turunantan(x)
```

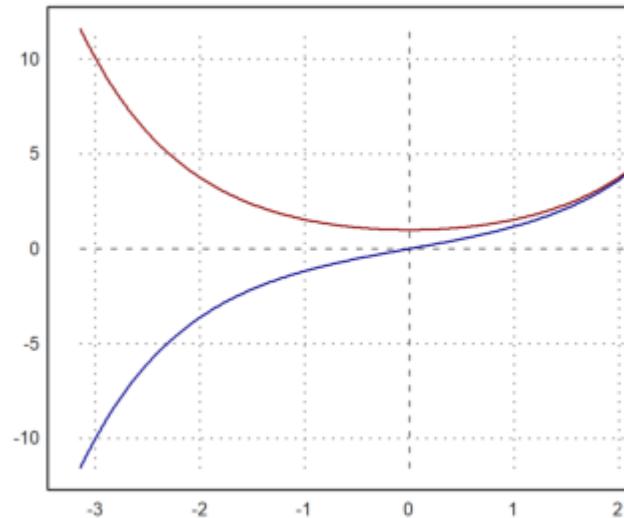
Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
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)
```

Hasilnya adalah $\cosh(x)$, karena



```
plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]):
```

```
function f(x) = sin(3x^5 + 7)^2
```

```
2 5 sin (3 x + 7)
```

```
diff(f,3), diffc(f,3)
```

```
1198.32948904 1198.72863721
```

Apakah perbedaan `diff` dan `diffc`?

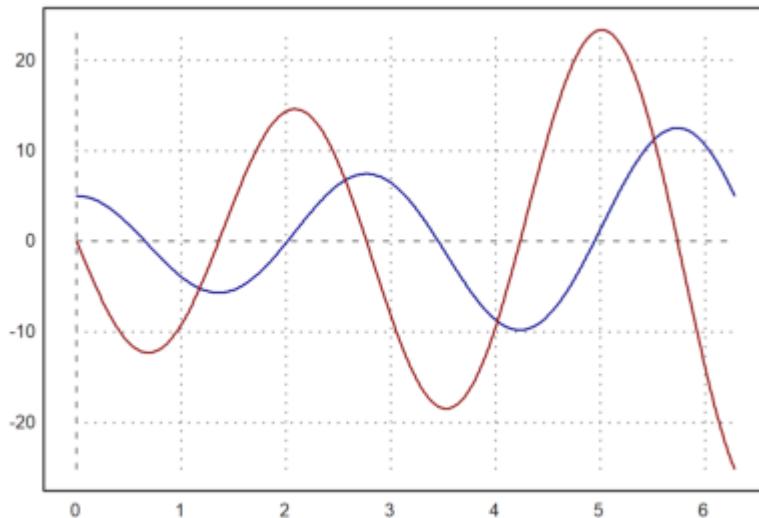
`diff` digunakan untuk turunan biasa, sedangkan `diffc` digunakan untuk turunan total dalam konteks yang lebih kompleks.

```

showev('diff(f(x),x))

float(
plot2d(f,0,3.1):
function f(x) =5cos(2x)-2xsin(2x) // mendefinisikan fungsi f
5 cos(2 x) - 2 x sin(2 x)
function df(x) =diff(f(x),x) // fd(x) = f'(x)
- 12 sin(2 x) - 4 x cos(2 x)
'f(1) = f(1),float(f(1)), 'f(2) = f(2),float(f(2)) // nilai f(1) dan f(2)
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")],0,2pi,color=[blue,red]): //grafik fungsi dan turunan-

```



nya

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.

Nomor 1

```

function f(x) := cos(x2)
showev('limit((cos((x + h)2) - cos(x2))/h, h, 0))

```

```

function df(x) =diff(cos(x2),x)//fd(x) = f'(x)
2 - 2 x sin(x )
plot2d(["f(x)", "df(x)"],-pi,pi,color=[blue,red]):
Nomor 2
function f(x) := sqrt(x2 + 4)
showev('limit((sqrt((x + h)2 + 4) - sqrt(x2 + 4))/h,h,0))
Nomor 3
function f(x) :=(3-x)3
showev('limit(((3 - (x + h))3 - (3 - x)3)/h,h,0))
Nomor 4
function f(x) :=sin(x)+2cos(x)
showev('limit((sin(x + h) + 2cos(x + h) - (sin(x) + 2cos(x)))/h,h,0))
Nomor 5
function f(x) :=10x-2
showev('limit(((10(x + h) - 2) - (10x - 2))/h,h,0))

```

4.7 Integral

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).

Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak integrabel, kita tidak akan menggunakan metode ini).

```

showev('integrate(xn,x))
Answering "Is n equal to -1?" with "no"
showev('integrate(1/(1 + x),x))
showev('integrate(1/(1 + x2),x))
showev('integrate(1/sqrt(1 - x2),x))
showev('integrate(sin(x),x,0,pi))
plot2d("sin(x)",0,2pi):
showev('integrate(sin(x),x,a,b))
showev('integrate(xn,x,a,b))
Answering "Is n positive, negative or zero?" with "positive"
showev('integrate(x2sqrt(2x + 1),x))
showev('integrate(x2sqrt(2x + 1),x,0,2))
ratsimp(
showev('integrate((sin(sqrt(x) + a)Essqrt(x))/sqrt(x),x,0,pi2))
factor(
function map f(x) = E^( - x2)

```

2 - x E

showev('integrate(f(x),x))

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):

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 $x = 0.1, 0.2, 0.3, \dots, 3.2$.

0.1sum(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))

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))

x=0:0.1:1-0.01; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,1, add):

Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah *integrate*. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

t = makelist(a,a,0,1-0.01,0.01);

fx = makelist(f(t[i]+0.01),i,1,length(t));

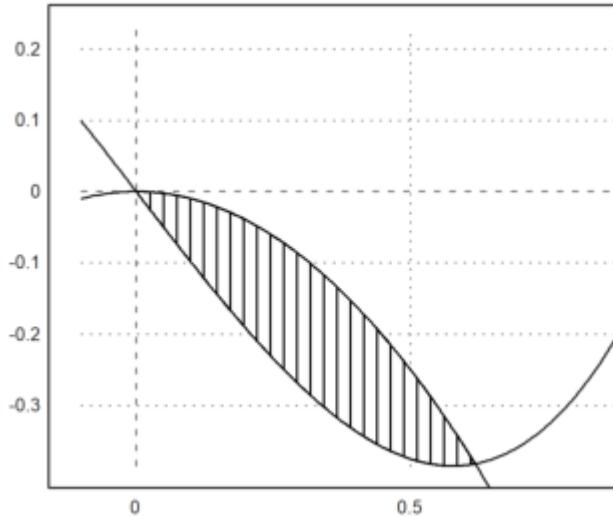
maxima: *'integrate(f(x),x,0,1) = 0.01*sum(fx[i],i,1,length(fx))*

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
function f(x) = sin(3x5 + 7)2
2 5 sin (3 x + 7)
integrate(f,0,1)
0.542581176074
showev('integrate(f(x), x, 0, 1))
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(0.2,6.0I) - 4.192962712629476 I gamma
showev('integrate(xexp(-x), x, 0, 1))//Integraltentu(eksak)
```

4.8 Aplikasi Integral Tentu

```
plot2d("x3-x", -0.1, 1.1); plot2d("-x2, add); ... b = solve("x3-x+x2", 0.5); x =
linspace(0, b, 200); xi = flipx(x); ... plot2d(x|xi, x3 - x| - xi2, filled, style =
```



```
""|", fillcolor = 1, add) : //Plotdaerahantara2kurva
solve("x3-x+x2, 0), b = solve("x3-x+x2, 1)//absis titik-potong kedua kurva
```

0 0.61803398875

integrate("(-x²) - (x³ - x)", a, b)//luas daerah yang diarsir

0.0758191713542

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

a = solve((-x²) - (x³ - x), x); a // menentukan absis titik potong kedua kurva secara eksak

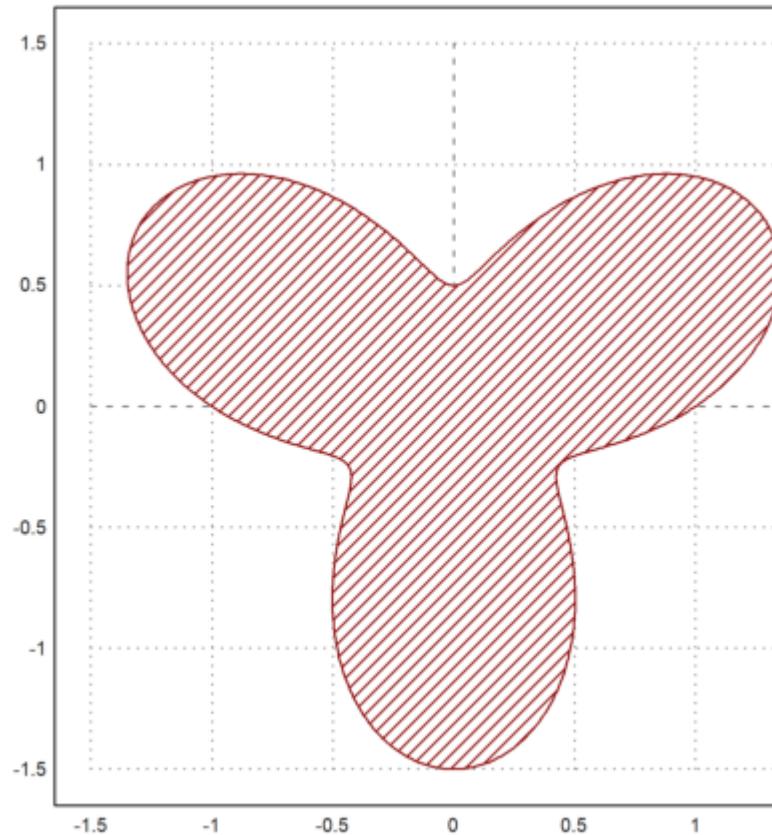
```
showev('integrate(-x2-x3+x, x, 0, (sqrt(5)-1)/2))//Nilai integral secara eksak
float(
```

4.9 Panjang Kurva

Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

dengan

```
t=linspace(0,2pi,1000); r=1+sin(3t)/2; x=r*cos(t); y=r*sin(t); ... plot2d(x,y,filled,fillcolor=red,style="/" ,r)
```

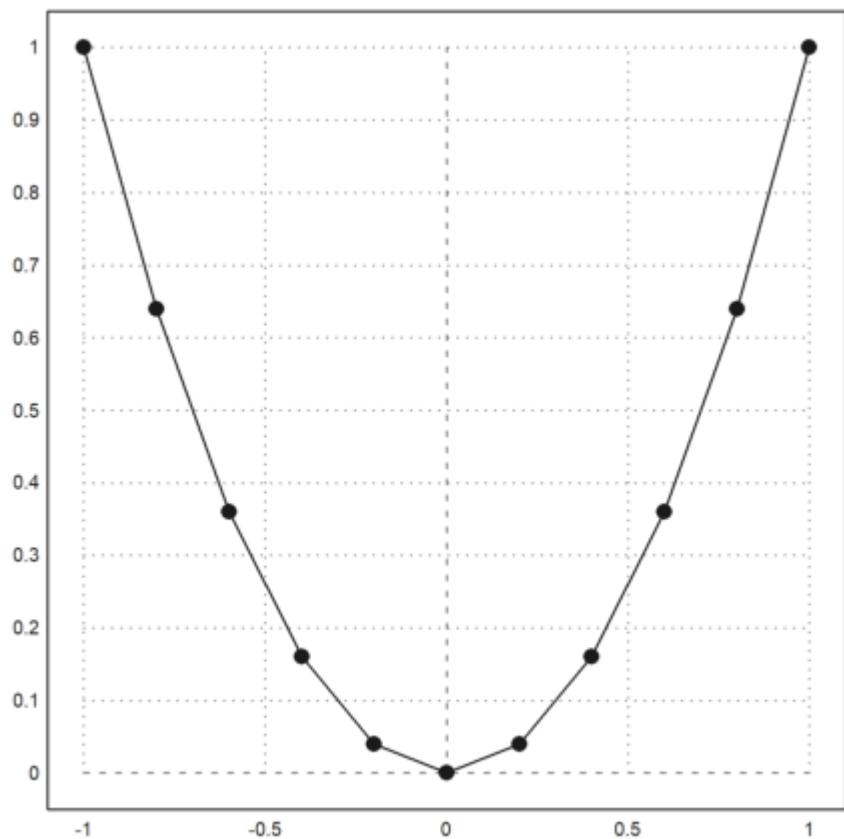


```
// Kita gambar kurvanya terlebih dahulu
function r(t) = 1+sin(3t)/2; 'r(t) = r(t)
    function fx(t) = r(t)cos(t); 'fx(t) = fx(t)
    function fy(t) = r(t)sin(t); 'fy(t) = fy(t)
    function ds(t) = trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2)));'ds(t)=ds(t)
    integrate(ds(x), x, 0, 2pi)//panjang(keliling)kurva
    Maxima gagal melakukan perhitungan eksak integral tersebut.
Berikut kita hitung integralnya secara umerik dengan perintah EMT.
integrate("ds(x)",0,2pi)
9.0749467823
Spiral Logaritmik
a=0.1; plot2d("exp(ax)cos(x)","exp(ax)sin(x)",r=2,xmin=0,xmax=2pi):
kill(a) // hapus expresi a
done
```

```

function fx(t) = exp(at)cos(t); 'fx(t) = fx(t)
function fy(t) = exp(at)sin(t); 'fy(t) = fy(t)
function df(t) = trigreduce(radcan(sqrt(diff(fx(t),t)2+diff(fy(t),t)2)));'df(t)=df(t)
S =integrate(df(t),t,0,2
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("x2", xmin = -1, xmax = 1):
showev('integrate(sqrt(1 + diff(x2, x)2), x, -1, 1))
float(
x=-1:0.2:1; y=x2; plot2d(x, y); ... plot2d(x, y, points = 1, style = "o", add =

```



1) :

Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```

i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])2 + (y[i + 1] - y[i])2))
2.95191957027

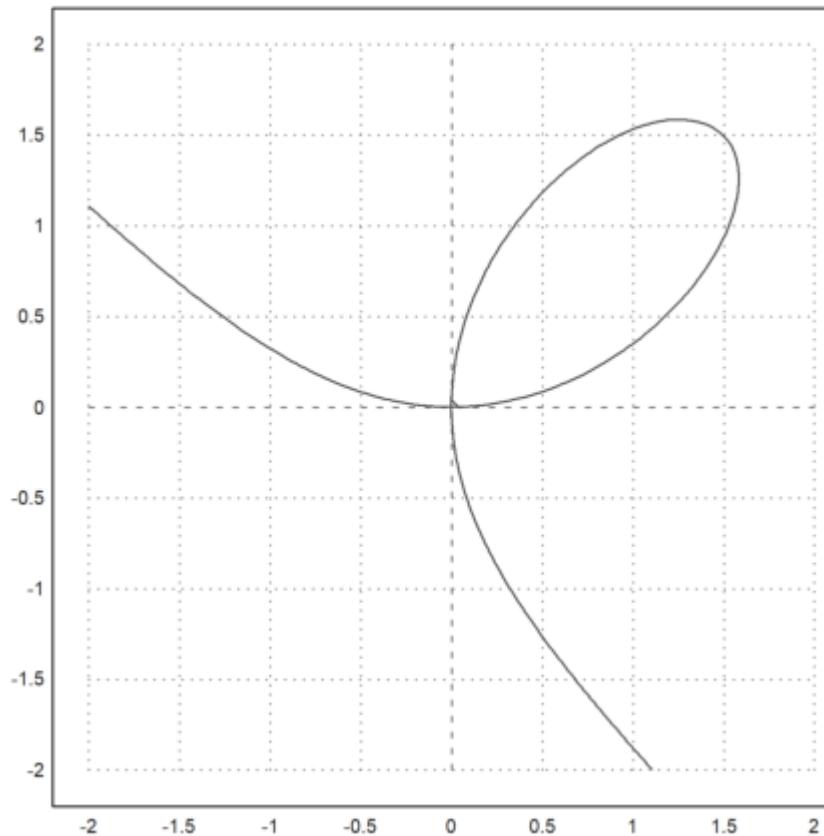
```

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya.

4.10 Koordinat Kartesius

Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

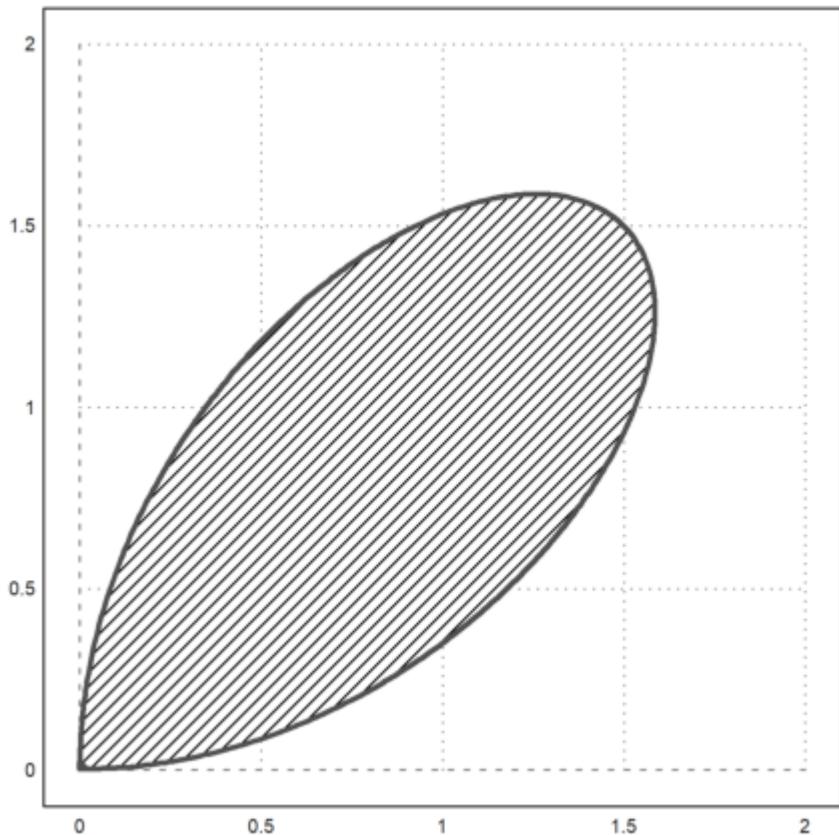
$$z = x^3 + y^3 - 3xy; z$$



`plot2d(z,r=2,level=0,n=100):`

Kita tertarik pada kurva di kuadran pertama.

`plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="/"':`



Kita selesaikan persamaannya untuk x.

zwithy = lx, sol = solve(

Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

function f(l) = rhs(sol[1]); 'f(l) = f(l)

Fungsi tersebut juga dapat digunakan untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai $y=l \cdot x$, yakni $x=f(l)$ dan $y=l \cdot f(l)$.

plot2d(f(x),xf(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):

Elemen panjang kurva adalah:

*function ds(l) = ratsimp(sqrt(diff(f(l),l)^2 + diff(l*f(l),l)^2));'ds(l)=ds(l)
integrate(ds(l),l,0,1)*

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral tersebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

2integrate("ds(x)",0,1)

4.91748872168

2romberg(ds(x),0,1)// perintah Euler lain untuk menghitung nilai hampiran

integral yang sama

4.91748872168

Perhitungan di datas dapat dilakukan untuk sebarang fungsi x dan y dengan mendefinisikan fungsi EMT, misalnya kita beri nama panjangkurva. Fungsi ini selalu memanggil Maxima untuk menurunkan fungsi yang diberikan.

function panjangkurva(fx,fy,a,b) ...

```
ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2));returnromberg(ds,a,b);endfunction <
/pre> panjangkurva("x","x^2",-1,1)//cekuntukmenghitungpanjangkurvaparabolasebelumnya
2.95788571509
```

Bandingkan dengan nilai eksak di atas.

```
2panjangkurva(mxm("f(x)",mxm("xf(x)",0,1)) // cek contoh terakhir, bandingkan hasilnya!
```

4.91748872168

Kita hitung panjang spiral Archimedes berikut ini dengan fungsi tersebut.

```
plot2d("xcos(x)","xsin(x)",xmin=0,xmax=2pi,square=1):
```

```
panjangkurva("xcos(x)","xsin(x)",0,2pi)
```

21.2562941482

Berikut kita definisikan fungsi yang sama namun dengan Maxima, untuk perhitungan eksak.

```
kill(ds,x,fx,fy)
```

```
done
```

```
function ds(fx,fy) = sqrt(diff(fx,x)^2 + diff(fy,x)^2)
```

```
2 2 sqrt(diff (fy, x) + diff (fx, x))
```

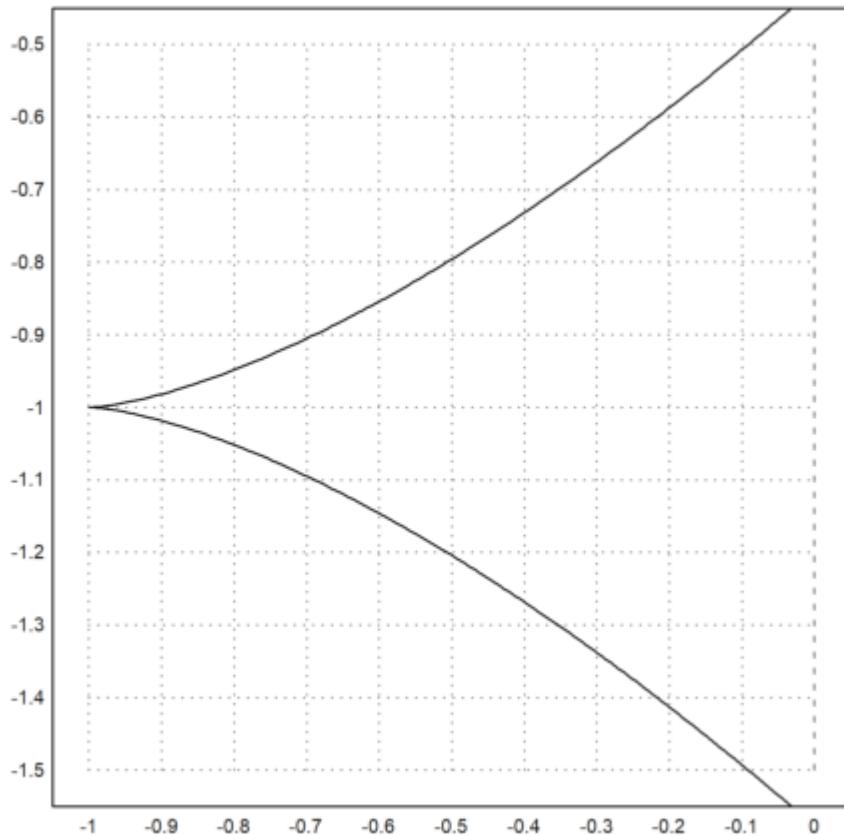
```
sol = ds(xcos(x),xsin(x)); sol//Kitagunakanuntukmenghitungpanjangkurvaterakhirdiatas
sol|trigreduce|expand,integrate(
```

21.2562941482

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.

Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
plot2d("3x^2-1","3x^3-1",xmin = -1/sqrt(3),xmax = 1/sqrt(3),square =
```



1) : $sol =$
 $radcan(ds(3x^2 - 1, 3x^3 - 1));sol$
 $showev('integrate(sol, x, 0, 1/sqrt(3))),2float($

4.11 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 $t=0$, $t=\pi/2$, $t=r*\pi$.

$$\begin{aligned} x &= r(t-\sin(t)) \\ r(t - \sin(t)) \\ y &= r(1-\cos(t)) \\ r(1 - \cos(t)) \\ t &= 0 \end{aligned}$$

```

0
r=1
1

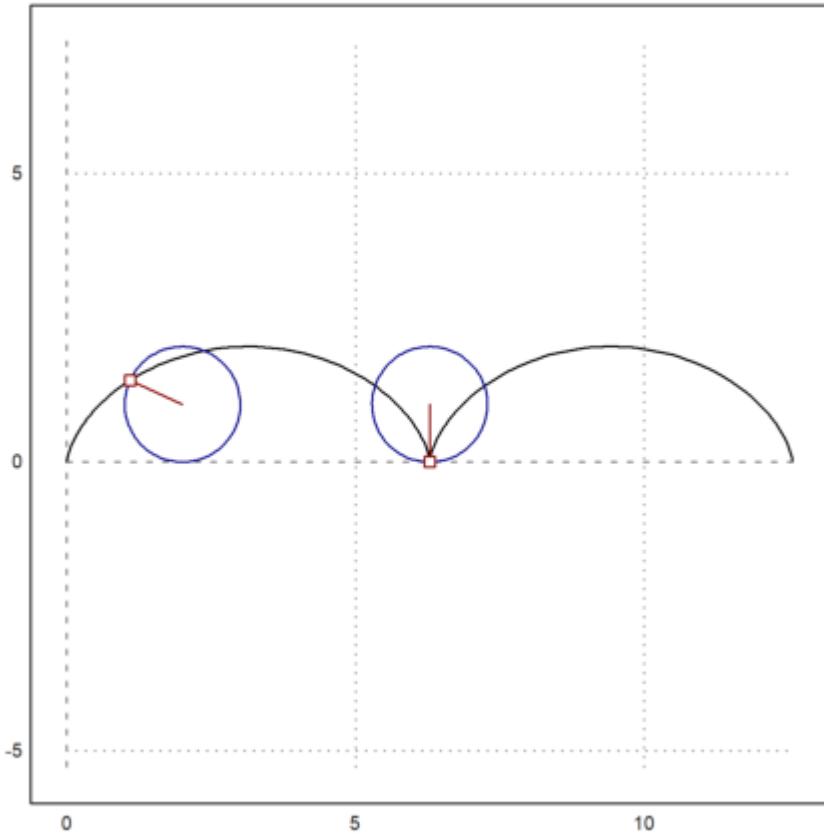
```

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,col
... 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):

```



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 pan-
jang kurva sikloid
ds = trigsimp(ds); ds
showev('integrate(ds, x, 0, 2pi))//hitungpanjangsikloidsatuputaranpenuh
integrate(mxm("ds"),0,2pi) // hitung secara numerik
0

```

```
romberg(mxm("ds"),0,2pi) // cara lain hitung secara numerik  
0
```

Perhatikan, seperti terlihat pada gambar, panjang sikloid lebih besar dari pada keliling lingkarannya, yakni:

4.12 Kurvatur (Kelengkungan) Kurva

image: Osculating.png

Aslinya, kelengkungan kurva diferensiabel (yakni, kurva mulus yang tidak lancip) di titik P didefinisikan melalui lingkaran oskulasi (yaitu, lingkaran yang melalui titik P dan terbaik memperkirakan, paling banyak menyinggung kurva di sekitar P). Pusat dan radius kelengkungan kurva di P adalah pusat dan radius lingkaran oskulasi. Kelengkungan adalah kebalikan dari radius kelengkungan:

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar 2π sejauh $2\pi R$.)

Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang 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 // elemenpanjangkurva, panjangbusur  
r t
```

```
kill(s); fx = rcos(s/r); fy = rsin(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
```

Untuk representasi parametrik umum, misalkan

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

Jadi, rumus kurvatur untuk kurva parametrik
adalah

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kartesius

maka rumus kurvaturnya adalah

(Silakan Anda turunkan rumus tersebut!)

Contoh:

Lingkaran dengan pusat $(0,0)$ dan jari-jari r dapat dinyatakan dengan persamaan parametrik

Nilai kelengkungan lingkaran tersebut adalah

Hasil cocok dengan definisi kurvatur suatu kelengkungan.

Kurva

dapat dinyatakan ke dalam persamaan parametrik

sehingga kurvaturnya adalah

Contoh:

Akan ditentukan kurvatur parabola

function $f(x) = ax^2 + bx + c; y=f(x)$

function $k(x) = (\text{diff}(f(x),x,2))/(1+\text{diff}(f(x),x)^2)^{(3/2)}; k(x)=k(x) // \text{kelengkungan parabola}$

function $f(x) = x^2 + x + 1; y=f(x) // \text{akan kita plot kelengkungan parabola}$
untuk $a=b=c=1$

function $k(x) = (\text{diff}(f(x),x,2))/(1+\text{diff}(f(x),x)^2)^{(3/2)}; k(x)=k(x) // \text{kelengkungan parabola}$

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);

Untuk kurva yang dinyatakan dengan fungsi implisit

dengan turunan-turunan parsial

berlaku

sehingga kurvaturnya adalah

(Silakan Anda turunkan sendiri!)

Contoh 1:

Parabola

dapat dinyatakan ke dalam persamaan implisit

function $F(x,y) = ax^2 + bx + c - y; F(x,y)$

$F_x = \text{diff}(F(x,y),x)$, $F_{xx} = \text{diff}(F(x,y),x,2)$, $F_y = \text{diff}(F(x,y),y)$, $F_{xy} = \text{diff}(\text{diff}(F(x,y),x),y)$,
 $F_{yy} = \text{diff}(F(x,y),y,2)$

$2ax + b$

$2a$

-1

0

0

function $k(x) = (F_y^2 F_{xx} - 2F_x F_y F_{xy} + F_x^2 F_{yy})/(F_x^2 + F_y^2)^{(3/2)}; k(x)=k(x)$

// kurvatur parabola tersebut

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

(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:

(Pilih fungsi dan gambar kurvanya.)

- Apabila fungsi dinyatakan dalam koordinat kutub $x=f(r,t)$, $y=g(r,t)$, $r=h(t)$, $x=a$ bersesuaian dengan $t=t_0$ dan $x=b$ bersesuaian dengan $t=t_1$, maka rumus di atas akan menjadi:

* Pilih beberapa kurva menarik (selain lingkaran dan parabola) dari * buku kalkulus. Nyatakan setiap kurva tersebut dalam bentuk: * a. koordinat Kartesius (persamaan $y=f(x)$) * b. koordinat kutub ($r=r(\theta)$) * c. persamaan parametrik $x=x(t)$, $y=y(t)$ * d. persamaan 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.

Nomor 1

```
function f(x):=sin(2x)
showev('integrate(sin(3x),x))
showev('integrate(sin(3x),x,0,pi/4))
x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),bar); plot2d("f(x)",0,1,add):
```

Nomor 2

```
function f(x):=sqrt(2x^2 + 4x)
showev('integrate(f(x),x))
showev('integrate(sqrt(2x^2 + 4x),x,0,2))
x=0:0.1:pi-0.01; plot2d(x,f(x+0.01),bar); plot2d("f(x)",0,1,add):
```

Nomor 3

```
function f(x):=(x4 + x2)
showev('integrate((x4 + x2), x))
showev('integrate(x4 + x2, x, 0, 2))
x=0:0.1:pi-0.01; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,1, add):
```

Nomor 4

```
function f(x):=sin(6x)
showev('integrate(sin(2x), x))
showev('integrate(sin(2x), x, 0, pi/4))
x=0:0.1:pi-0.01; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,1, add):
```

Nomor 5

```
function f(x):=x3 + 3x + 2
showev('integrate(x3 + 3x + 2, x))
showev('integrate(x3 + 3x + 2, x, 0, pi/4))
x=0:0.1:pi-0.01; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,1, add):
```

Barisan dan Deret

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- * dengan cara yang sama seperti mendefinisikan vektor dengan * elemen-elemen beraturan (menggunakan titik dua ":");
 - * menggunakan perintah "sequence" dan rumus barisan (suku ke -n);
 - * menggunakan perintah "iterate" atau "niterate";
 - * menggunakan fungsi Maxima "create_list" atau "makelist" untuk menghasilkan barisan simbolik;
 - * menggunakan fungsi biasa yang inputnya vektor atau barisan;
 - * menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- * sum: menghitung jumlah semua elemen suatu barisan
- * cumsum: jumlah kumulatif suatu barisan
- * differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
1:10 // barisan sederhana
```

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

```
1:2:30
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
```

Iterasi dan Barisan

EMT menyediakan fungsi iterate("g(x)", x0, n) untuk melakukan iterasi

Berikut ini disajikan contoh-contoh penggunaan iterasi dan rekursi dengan EMT. Contoh pertama menunjukkan pertumbuhan dari nilai awal 1000 dengan laju pertambahan 5

```
q=1.05; iterate("xq",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 610000 per bulan, tabungan sebesar 1 juta tanpa diambil selama sekitar 10 tahunan akan habis diambil oleh bank!

```
r=0.005; plot2d(iterate("(1+0.8r)x-10000",1000000,n=130));
```

Silakan Anda coba-coba, dengan tabungan minimal berapa agar tidak akan habis diambil oleh bank dengan ketentuan bunga dan biaya administrasi seperti di atas.

Berikut adalah perhitungan minimal tabungan agar aman di bank dengan bunga sebesar r dan biaya administrasi a, pajak bunga 20

```
solve(0.8rA - a, A),
```

Berikut didefinisikan fungsi untuk menghitung saldo tabungan, kemudian dilakukan iterasi.

```
function saldo(x,r,a) := round((1+0.8r)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)
plot2d(sequence("x[n-1]+x[n-2"],[1,1],250)(1/(1 : 250))) :
```

Barisan yang sama juga dapat dihasilkan dengan menggunakan loop.

```
x=ones(500); for k=3 to 500; x[k]=x[k-1]+x[k-2]; end;
```

Rekursi dapat dilakukan dengan menggunakan rumus yang tergantung pada semua elemen sebelumnya. Pada contoh berikut, elemen ke-n merupakan jumlah $(n-1)$ elemen sebelumnya, dimulai dengan 1 (elemen ke-1). Jelas, nilai elemen ke-n adalah $2^{(n-2)}$, untuk $n = 2, 4, 5, \dots$

```
sequence("sum(x)",1,10)
[1, 1, 2, 4, 8, 16, 32, 64, 128, 256]
```

Selain menggunakan ekspresi dalam x dan n, kita juga dapat menggunakan fungsi.

Pada contoh berikut, digunakan iterasi

dengan A suatu matriks 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.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);
```

Selanjutnya dihubungan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
for i=1:cols(x); plot2d([0,x[i]], add); end;
```

Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memerlukan indeks dan bilangan kompleks. Dalam hal ini diigunakan vektor kolom pada bidang.

```
function gstep (v) ...
```

```
w=[-v[2];v[1]]; return v+w/norm(w); endfunction j/pre; Jika dilakukan iterasi 16 kali dimulai dari [1;0] akan didapatkan matriks yang memuat vektor-vektor dari setiap iterasi.
```

```
x=iterate("gstep",[1;0],16); plot2d(x[1],x[2],r=3.5, points):
```

Kekonvergenan

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```

iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1.
0.739085133216
Iterasi tersebut konvergen ke penyelesaian persamaan
Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.
    hasil := iterate("cos(x)", 1,2 ) //iterasi x(n+1)=cos(x(n)), dengan interval awal (1, 2)
    0.739085133211,0.7390851332133
    Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan  $x=\cos(x)$ .
    h=expand(hasil,100), cos(h) jj h
    0.73908513309,0.73908513333 1
    Iterasi juga dapat digunakan pada fungsi yang didefinisikan.
    function f(x) := (x+2/x)/2
    Iterasi  $x(n+1)=f(x(n))$  akan konvergen ke akar kuadrat 2.
    iterate("f",2), sqrt(2)
    1.41421356237 1.41421356237
    Jika pada perintah iterate diberikan tambahan parameter n, maka hasil iterasinya akan ditampilkan mulai dari iterasi pertama sampai ke-n.
    iterate("f",2,5)
    [2, 1.5, 1.41667, 1.41422, 1.41421, 1.41421]
    Untuk iterasi ini tidak dapat dilakukan terhadap interval.
    niterate("f", 1,2 ,5)
    [ 1,2 , 1,2 , 1,2 , 1,2 , 1,2 , 1,2 ]
    Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya adalah perhitungan dengan interval bersifat terlalu longgar. Untuk meningkatkan perhitungan pada ekspresi dapat digunakan pembagian intervalnya, menggunakan fungsi ieval().
    function s(x) := ieval("(x+2/x)/2",x,10)
    Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal, dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang memuat akar persamaan:
    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
0.99999999999999778,1.00000000000000022 ... 4.999999999999911,5.00000000000000089 ...
Iterasi berikut konvergen sangat lambat.
iterate("sqrt(x)",2,10)
[2, 1.41421, 1.18921, 1.09051, 1.04427, 1.0219, 1.01089, 1.00543, 1.00271,
1.00135, 1.00068]
```

Kekonvergenan iterasi tersebut dapat dipercepat dengan percepatan Steffensen:

```
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 x2 = 2; end; x,
1.41421356237
```

Penggabungan matriks menggunakan tanda "—" dapat digunakan untuk menyimpan semua hasil iterasi.

```
v=[1]; for i=2 to 8; v=v—(v[i-1]i); end; v,
[1, 2, 6, 24, 120, 720, 5040, 40320]
hasil iterasi juga dapat disimpan pada vektor yang sudah ada.
v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]i; end; ... plot2d(v,logplot=1);
textbox(latex(log(n)),x=0.5):
A =[0.5,0.2;0.7,0.1]; b=[2;2]; ... x=[1;1]; repeat xnew=A.x-b; until all(xnew =x);
x=xnew; end; ... x,
-7.09677 -7.74194
```

Iterasi di dalam Fungsi

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.

Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

```
function map hiter(f, x0)...
x=x0; maxiter=0; repeat xnew=f(x); maxiter = maxiter + 1; until xnew =
x; x = xnew; end; return maxiter;
endfunction </pre > Misalnya, berikut adalah iterasi untuk mendapatkan 5, jika dimulai dari  $x_0 = 2$ .
hiter("(x+2/x)/2",2)
```

5

Karena fungsinya didefinisikan menggunakan "map". maka nilai awalnya dapat berupa vektor.

```
x=1.5:0.1:10; hasil=hiter("(x+2/x)/2",x); ... plot2d(x,hasil);
```

Dari gambar di atas terlihat bahwa kekonvergenan iterasinya semakin lambat, untuk nilai awal semakin besar, namun penambahannya tidak kontinu. Kita dapat menemukan kapan maksimum iterasinya bertambah.

hasil[1:10]

```
[4, 5, 5, 5, 5, 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.

$p = x^3 - 1; newton = x - p/diff(p, x); newton$

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 </pre> Kitamula idengan menentuk antitiktitik grid pada bidang kompleksnya.

r=1.5; x=linspace(-r,r,501); Z=x+Ix'; 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);

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 // baris anderet Taylor untuk e^x

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

Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

deret[3]

plot2d(["exp(x)",deret[1],deret[2],deret[3]],0,3,color=1:4);

sum(sin(kx)/k, k, 1, 5)

Berikut adalah cara menggambar kurva

```
plot2d(sum(sin((2k+1)x)/(2k+1),k,0,20),0,2pi);
```

Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan kita akan menggambar kurva

```
x=linspace(0,2pi,1000); k=1:100; y=sum(sin(kx'))/k'; plot2d(x,y);
```

Tabel Fungsi

Terdapat cara menarik untuk menghasilkan barisan dengan ekspresi Maxima. Perintah mxmttable() berguna untuk menampilkan dan menggambar barisan dan menghasilkan barisan sebagai vektor kolom.

Sebagai contoh berikut adalah barisan turunan ke-n $x^k d^k x = 1$.

```
mxmttable("diffat(x^k,x=1,n)","n",1,8,frac=1);
```

```
1 2 3 8 10 54 -42 944
```

```
'sum(k,k,1,n)=factor(ev(sum(k,k,1,n),simpsum=true))/simpsum :  
menghitungderetsecara simbolik
```

```
'sum(1/(3^k+k),k,0,inf)=factor(ev(sum(1/(3^k+k),k,0,inf),simpsum=true))
```

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((-1)^(k-1)/k,k,1,inf)=factor(ev(sum((-1)^(k-1)/k,k,1,inf),simpsum=true))
```

Di sini masih gagal, hasilnya tidak dihitung.

```
'sum((-1)^k/(2k-1),k,1,inf)=factor(ev(sum((-1)^k/(2k-1),k,1,inf),simpsum=true))
```

```
ev(sum(1/n!,n,0,inf),simpsum=true)
```

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
assume(abs(x)>1);'sum(ax^k,k,0,inf)=ev(sum(ax^k,k,0,inf),simpsum=true),forget(abs(x)<1);
```

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)
```

```
limit(sum(x^k/k!,k,0,n),n,inf)
```

```
function d(n)=sum(1/(k^2-k),k,2,n);d(n)=d(n)
```

```
d(10)=ev(d(10),simpsum=true)
```

```
d(100)=ev(d(100),simpsum=true)
```

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 Taylore^x di sekitar x=0, sampai suku ke-
```

11

```
exp(1)
```

```
2.71828182846
```

```
'log(x)=taylor(log(x),x,1,10)//deret log(x) di sekitar x=1
```

```
log(2)
```

```
0.69314718056
```

```
'sin(x)=taylor(sin(x),x,1,5)
```

```
sin(1)
```

```
0.841470984808
```

FUNGSI MULTIVARIABEL

Sebuah fungsi multivariabel adalah pemetaan matematis yang menghubungkan beberapa variabel independen dengan satu variabel dependen. Fungsi ini umumnya dinotasikan sebagai

Fungsi multivariabel dapat diwakili dalam bentuk peta atau grafik tiga dimensi.

Contoh fungsi multivariabel :

Dimana variabel bebasnya yaitu x dan y.

Grafik Fungsi Multivariabel

Pada fungsi multivariabel, grafik fungsinya merupakan grafik tiga dimensi.

Ruang dimensi tiga dilambangkan dengan

Permukaan dalam R^3 terdapat dua macam yakni permukaan linear dan kuadratik. Setiap permukaan linear ber-

Untuk membuat grafik fungsi tiga dimensi, maka kita dapat menggunakan perintah "plot3d()"

Grafik Fungsi Persamaan Linear Dalam Dimensi Tiga

Bentuk umum persamaan permukaan linear adalah

Contoh grafik persamaan linear di ruang tiga dimensi

plot3d("x-1"):

plot3d("3x+4y-12"):

plot3d("7x-1"):

Grafik Fungsi Kuadratik di Ruang Dimensi Tiga

Persamaan kuadratik mempunyai rumus umum :

Permukaan-permukaan kuadratik dapat berupa permukaan bola, ellipsoida, paraboloida, tabung ellips, tabung lingkaran, atau tabung parabola.

Contoh grafik fungsi kuadratik di ruang dimensi tiga:

plot3d("x² + y²") :

Pada contoh diatas, digunakan ekspresi langsung dalam fungsi plot3d dan permukaan tersebut berupa paraboloida.

function f(x,y) := exp(-(x² + y²))

plot3d("f", r=5):

plot3d("y² - x²", r = 2,) :

Gambar di atas merupakan paraboloida hiperbolik

Menggambar kurva perpotongan dari dua persamaan

Di EMT kita juga dapat menggabungkan dua kurva pada satu bidang untuk menggambarkan perpotongan. Untuk masalah ini kita gunakan fungsi gt;add dalam prosesnya.

Contoh 1:

plot3d("x² + y² + z - 4", r = 5, implicit = 3) :

plot3d("x² + y² - 1", implicit = 3, r = 5, add) :

Dari persamaan diatas, kita dapatkan perpotongan antara paraboloida dengan tabung lingkaran.

contoh 2:

plot3d("x² + y² + 2z - 9", r = 5, implicit = 3) :

plot3d("y-1",r=5, implicit=3, add):

Dari persamaan diatas, didapatkan perpotongan antara bidang datar dan bidang lengkung.

contoh 3:

```
plot3d("x^2+y^2-4", r = 5, implicit = 3); plot3d("y^2+z^2-4", r = 5, implicit = 3, add) :
```

Didapatkan perpotongan antara dua tabung lingkaran.

Turunan Fungsi Multivariabel

Turunan Fungsi Dua Variabel

Turunan parsial, yaitu turunan fungsi terhadap satu variabel bebas semestara variabel bebas lainnya dianggap tetap atau konstan.

* Turunan Parsial terhadap f terhadap x di (x0,y0)

* Turunan Parsial f terhadap y di (x0,y0)

Contoh soal untuk turunan parsial :

$$z = 2xy - (1-x^2)$$

$$2 - 1 + x + 2 x y$$

$$\text{showev}(\text{limit}(((2(x+h)y) - (1 - (x+h)^2))/h, h, 0))$$

Perhitungan akan dilakukan menggunakan diff

diff(z,x) // z akan diturunkan terhadap x

$$2 x + 2 y$$

Karena pada fungsi z terdapat $2xy$ yang merupakan perkalian, untuk menghitung turunannya kita gunakan $u'v+uv'$, sehingga turunan dari $2xy$:

Kemudian turunan dari variabel berpangkat yaitu:

Jadi turunan dari x^2 :

Turunan dari konstanta adalah 0

sehingga, turunan dari fungsi

terhadap x adalah

diff(z,y) // z akan diturunkan terhadap y

$$2 x$$

Seperti pada turunan terhadap x, kita gunakan langkah yang sama namun kita anggap x konstan.

Jadi, turunan terhadap y dari $f(x,y)$ adalah $2x$.

Sehingga, turunan parsial dari

Apabila kita gambarkan grafik untuk masing-masing turunan tersebut adalah sebagai berikut:

```
plot3d("2x+2y"): // turunan terhadap x
```

```
plot3d("2x"): // turunan terhadap y
```

Integral

Integral Lipat Dua

Integral lipat dua f pada R, diberikan oleh

atau bisa ditulis dengan

Misalkan R adalah sebuah persegi panjang dengan sisi-sisi sejajar sumbu-sumbu koordinat; yaitu, misalkan

Urutan dx dan dy penting karena ia merinci integrasi mana yang akan dilakukan pertama.

SIFAT-SIFAT INTEGRAL LIPAT DUA:

1. Integral lipat dua bersifat linear, yaitu:
2. Integral lipat dua bersifat aditif(dapat dijumlahkan) pada persegi panjang yang saling berhimpit pada hanya sebuah ruas garis
3. Sifat perbandingan berlaku jika

CONTOH :

1. Hitung

Dengan memperhatikan urutan dx dan dy , pada soal kali ini kita hitung integralnya dari sebelah dalam yaitu kita integralkan terhadap x terlebih dahulu.

$$qx = \text{integrate}((2xy),x)$$

$$2 x y$$

Setelah kita integralkan terhadap x , lalu hasilnya kita integralkan terhadap y .

$$qy = \text{integrate}(qx,y)$$

$$2 2 x y — 2$$

Jadi, hasil dari

adalah

2. Hitung

$$f = 2x+3y$$

$$2 x + 3 y$$

Dalam integral sebelah dalam, y berupa konstanta, sehingga

$$fx = \text{integrate}((2x+3y),x,1,2)$$

$$3 + 3 y$$

Setelah mencari integral terhadap x , lalu kita integralkan hasil tersebut terhadap y , akibatnya

$$fy = \text{integrate}((3+3y),y,0,3)$$

$$45 - 2$$

Aplikasi Fungsi Multivariabel

Penerapan Plot Kontur

Peta kontur seringkali digunakan untuk memperlihatkan kondisi cuaca atau lainnya dari berbagai titik dalam peta.

1.Buatlah plot kontur dari fungsi berikut

$$f = 1/2(x^2 + y^2)$$

$$2 2 x + y — 2$$

aspect(1.5);

plot2d(f,level=0:2:20, hue, spectral,n=200,grid=4,r=5):

2. Seorang ahli geologi sedang melakukan penelitian di daerah vulkanik. Dia tertarik untuk memahami bagaimana ketinggian permukaan tanah di sekitar gunung berapi berubah. Ketinggian di suatu titik (x, y) dalam kilometer di sekitar gunung berapi tersebut dapat dijelaskan oleh fungsi ketinggian $H(x, y)$, di mana x dan y adalah koordinat dalam kilometer. Fungsi ketinggian $H(x, y)$ diberikan oleh persamaan:

Gambarkan plot kontur dari fungsi ketinggian $H(x, y)$ untuk memvisualisasikan perubahan ketinggian di sekitar gunung berapi dengan level ketinggian 1 sampai 7 km.

$$h = 3x^2 - 2y^2$$

$$2 2 3 x - 2 y$$

aspect(1.5);

plot2d(h,level=1:1:7, hue, spectral,n=200,grid=4,r=2):

5 Geometri

5.1 Visualisasi dan Perhitungan Geometri dengan EMT

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan geometri tersebut disimpan di dalam file program "geometry.e", sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

load geometry

Numerical and symbolic geometry.

Fungsi-fungsi Geometri

Fungsi-fungsi untuk Menggambar Objek Geometri:

defaultd:=textheight()*1.5: nilai asli untuk parameter d setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang koordinat setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas sumbu-x dan y adalah -r sd r plotPoint (P, "P"): menggambar titik P dan diberi label "P" plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label "AB" sejauh d plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c" plotLabel (label, P, V, d): menuliskan label pada posisi P

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

turn(v, phi): memutar vektor v sejauh phi turnLeft(v): memutar vektor v ke kiri turnRight(v): memutar vektor v ke kanan normalize(v): normal vektor v crossProduct(v, w): hasil kali silang vektor v dan w. lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh. ax+by=c. lineWithDirection(A,v): garis melalui A searah vektor v getLineDirection(g): vektor arah (gradien) garis g getNormal(g): vektor normal (tegak lurus) garis g getPointOnLine(g): titik pada garis g perpendicular(A, g): garis melalui A tegak lurus garis g parallel (A, g): garis melalui A sejajar garis g lineIntersection(g, h): titik potong garis g dan h projectToLine(A, g): proyeksi titik A pada garis g distance(A, B): jarak titik A dan B distanceSquared(A, B): kuadrat jarak A dan B quadrance(A, B): kuadrat jarak A dan B areaTriangle(A, B, C): luas segitiga ABC computeAngle(A, B, C): besar sudut lt;ABC angleBisector(A, B, C): garis bagi sudut lt;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/atasi) garis quad(A,B): kuadrat jarak AB spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni $\sin(\alpha)^2 / \text{dengan alpha Sudut yang menghadap sisi a}$. crossla

persamaan3quads dan 1spread pada segitiga dengan panjang sisi a, b, c. *triplespread(sa, sb, sc) : persamaan3spread sa, sb, sc yang memebntuk suatu segitiga double spread(sa) : Spread sudut rangkap Spread2* phi, dengan sa = sin(phi)^2 spreada.*

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");
```

Lalu 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]`

Hitunglah garis tegak lurus melalui A pada BC.

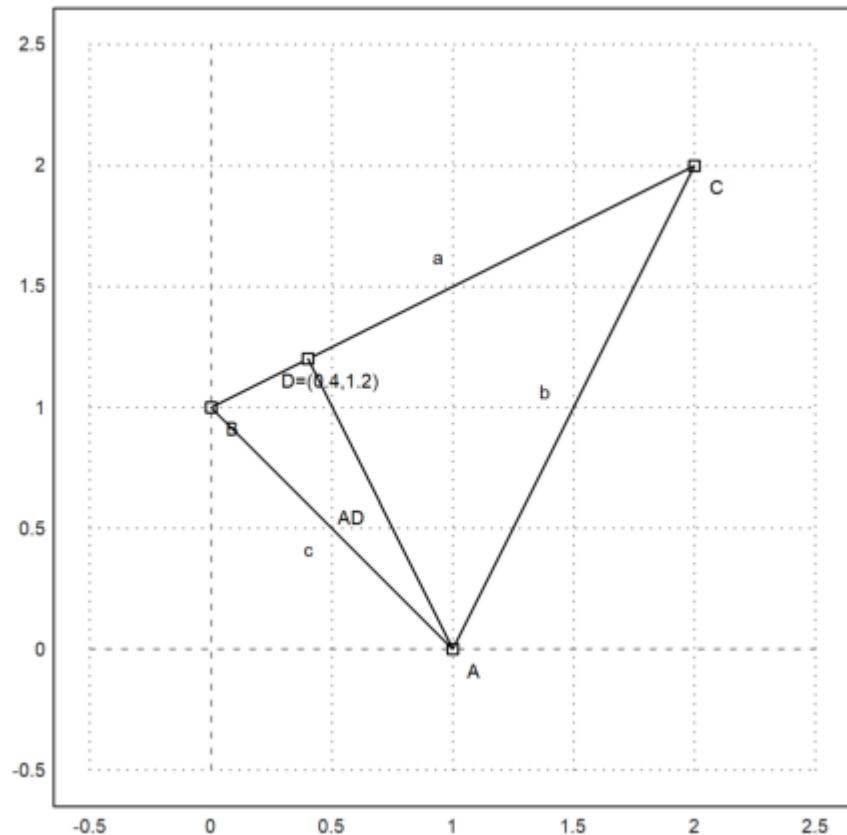
`h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui
A`

Dan persimpangannya dengan BC.

`D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan
BC`

Gambarkan itu.

```
plotPoint(D,value=1); // koordinat D ditampilkan
aspect(1); plotSegment(A,D); // tampilkan semua gambar hasil plot...()
```



Hitung luas ABC:

```
norm(A-D)norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
```

1.5

Compare with determinant formula.

```
areaTriangle(A,B,C) // hitung luas segitiga langsung dengan fungsi
```

1.5

Cara lain menghitung luas segitiga ABC:

```
distance(A,D)distance(B,C)/2
```

1.5

Sudut di C.

```
deprint(computeAngle(B,C,A))
```

$36^\circ 52' 11.63''$

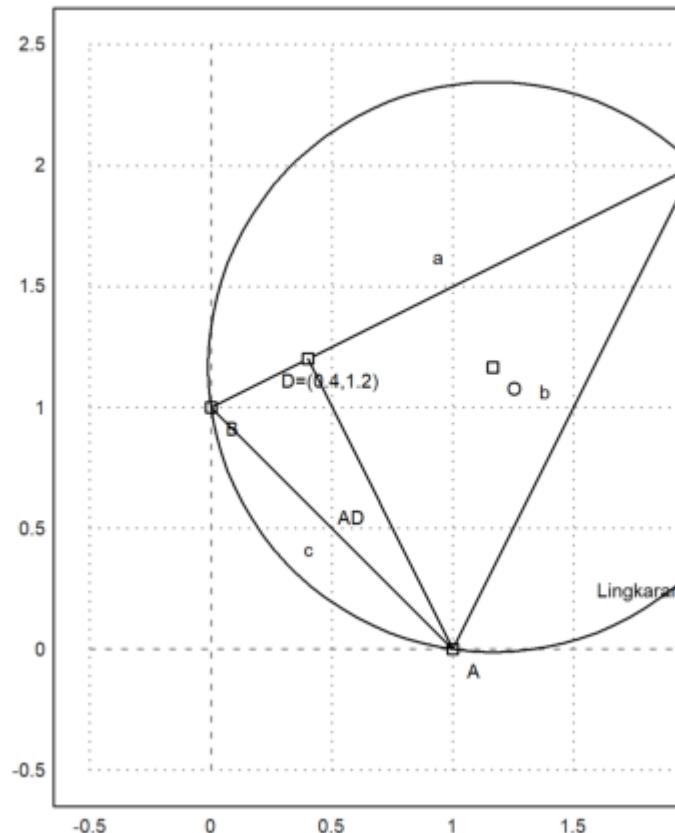
Sekarang lingkaran luar segitiga.

```
c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
```

```
R=getCircleRadius(c); // jari2 lingkaran luar
```

```
O=getCircleCenter(c); // titik pusat lingkaran c
```

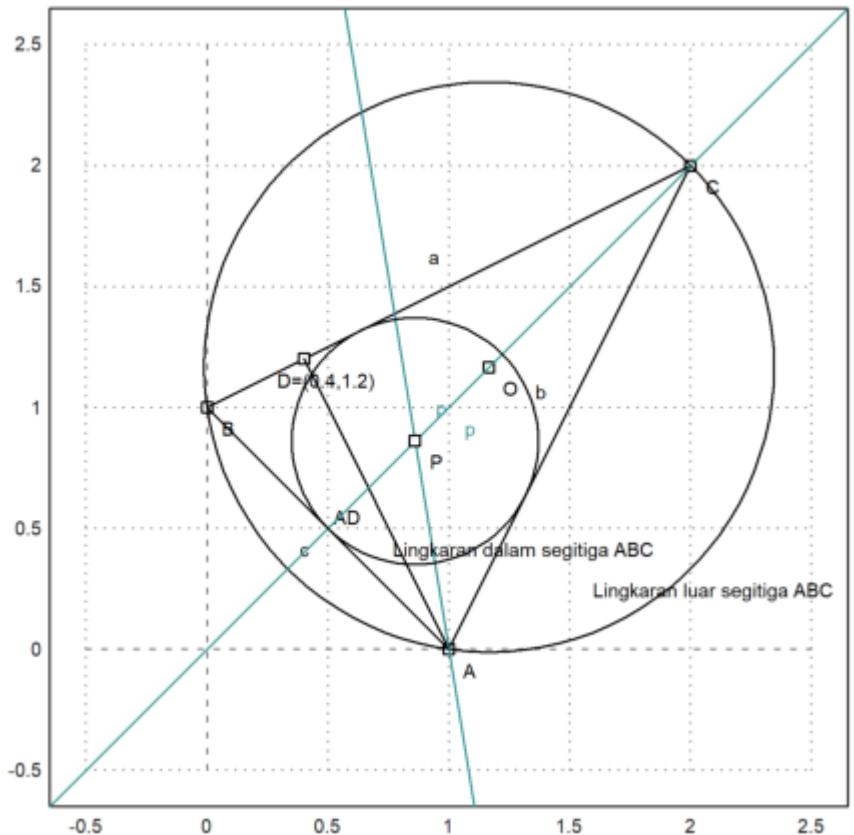
```
plotPoint(O,"O"); // gambar titik "O"
```



```

plotCircle(c,"Lingkaran luar segitiga ABC"):
Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.
O, R
[1.166666666667, 1.166666666667] 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 dalam alur cerita.
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"): // gam-

```



bar lingkaran dalam

Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

```
setPlotRange(-2.5,4.5,-2.5,4.5);
```

```
A=[-2,1]; plotPoint(A,"A");
```

```
B=[1,-2]; plotPoint(B,"B");
```

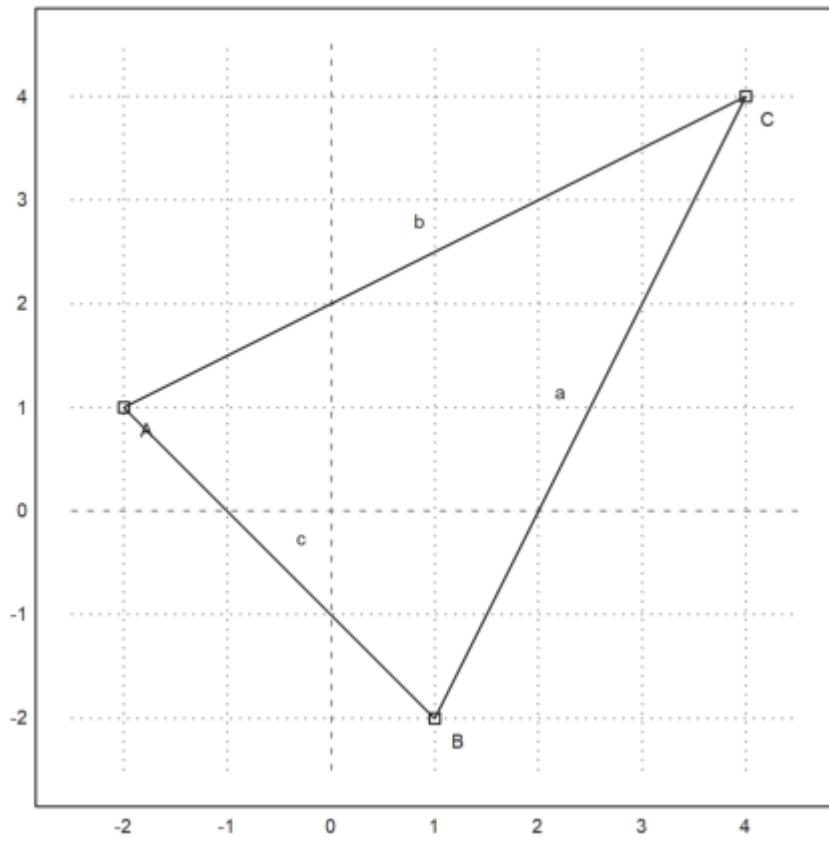
```
C=[4,4]; plotPoint(C,"C");
```

2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.

```
plotSegment(A,B,"c")
```

```
plotSegment(B,C,"a")
```

```
plotSegment(A,C,"b")
```



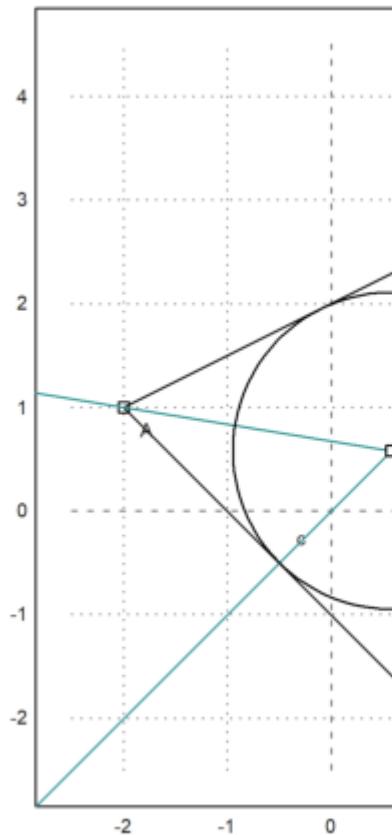
aspect(1):

3. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

```

l=angleBisector(A,C,B);
g=angleBisector(C,A,B);
P=lineIntersection(l,g)
[0.581139, 0.581139]
color(5); plotLine(l); plotLine(g); color(1);
plotPoint(P,"P");
r=norm(P-projectToLine(P,lineThrough(A,B)))
1.52896119631

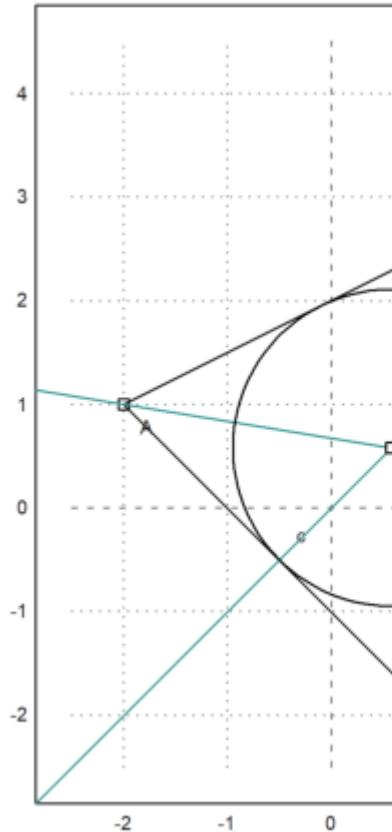
```



plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):

Jadi, terbukti bahwa garis bagi sudut yang ketiga juga melalui titik pusat lingkaran dalam.

4. Gambar jari-jari lingkaran dalam.
- ```
r=norm(P-projectToLine(P,lineThrough(A,B)))
```
- 1.52896119631



plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):

Contoh 2: Geometri Smbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometry.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Akan tetapi, kita sekarang dapat menggunakan perhitungan simbolik.

$A = [1,0]; B = [0,1]; C = [2,2]; //$  menentukan tiga titik A, B, C

Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi menyediakan perhitungan simbolis.

$c = \text{lineThrough}(B,C) // c=BC$

$[-1, 2, 2]$

Kita dapat memperoleh persamaan garis dengan mudah.

$\text{getLineEquation}(c, x, y), \text{solve}($

$\text{getLineEquation}(\text{lineThrough}([x1, y1], [x2, y2]), x, y), \text{solve}($

$\text{getLineEquation}(\text{lineThrough}(A, [x1, y1]), x, y) // \text{persamaan garis melalui } A \text{ dan } (x1, y1)$

$h = \text{perpendicular}(A, \text{lineThrough}(B, C)) // h \text{ melalui } A \text{ tegak lurus } BC$

$[2, 1, 2]$

$Q = \text{lineIntersection}(c, h) // Q \text{ titik potong garis } c=BC \text{ dan } h$

$2 \ 6 \ [-, -] \ 5 \ 5$

```

projectToLine(A, lineThrough(B, C)) // proyeksi Apada BC
distance(A, Q) // jarak AQ
cc = circleThrough(A, B, C); cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
r = getCircleRadius(cc); r, float(r) // tampilkan nilai jari-jari
computeAngle(A, C, B) // nilai < ACB
solve(getLineEquation(angleBisector(A, C, B), x, y), y)[1] // persamaan garis bagi <
ACB
P = lineIntersection(angleBisector(A, C, B), angleBisector(C, B, A)); P // titik potong 2 garis bagi sudut
P() // hasilnya sama dengan perhitungan sebelumnya
[0.86038, 0.86038]

```

## 5.2 Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga dapat membuat garis berpotongan dengan lingkaran, dan lingkaran dengan lingkaran.

```

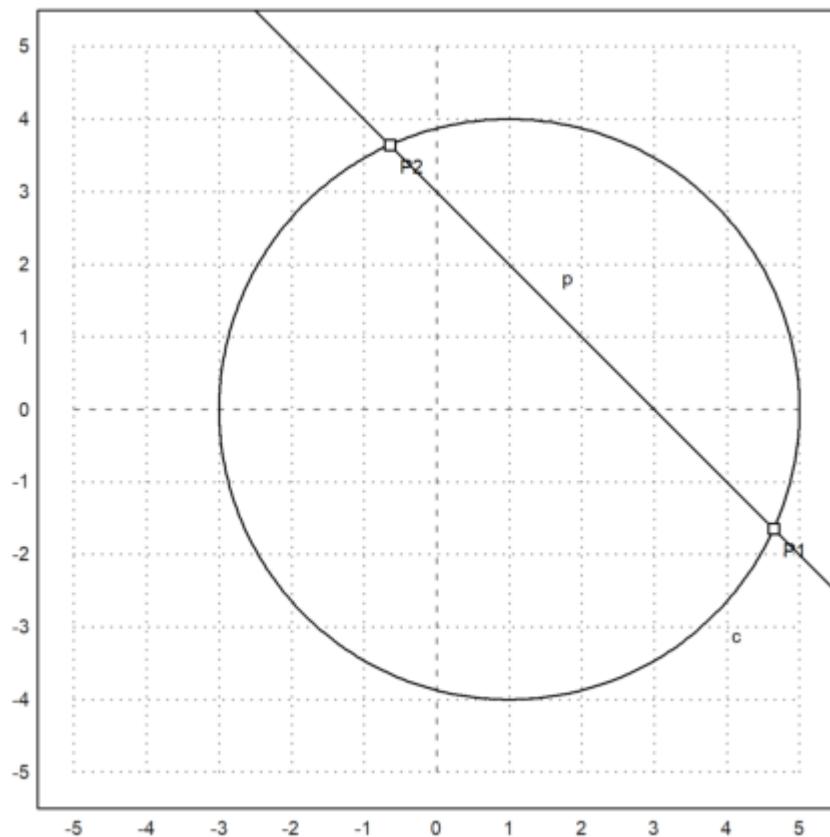
A := [1,0]; c = circleWithCenter(A, 4);
B := [1,2]; C := [2,1]; l = lineThrough(B, C);
setPlotRange(5); plotCircle(c); plotLine(l);
Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik
perpotongan.

```

```

P1, P2, f = lineCircleIntersections(l, c);
P1, P2, f
[4.64575, -1.64575] [-0.645751, 3.64575] 2

```



```
plotPoint(P1); plotPoint(P2);
```

Sama halnya di Maxima.

```
c = circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
[1, 0, 4]
```

```
l = lineThrough(B,C) // garis l melalui B dan C
[1, 1, 3]
```

*lineCircleIntersections(l,c)|radcan, //titikpotonglingkarancdangarisl*

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"
```

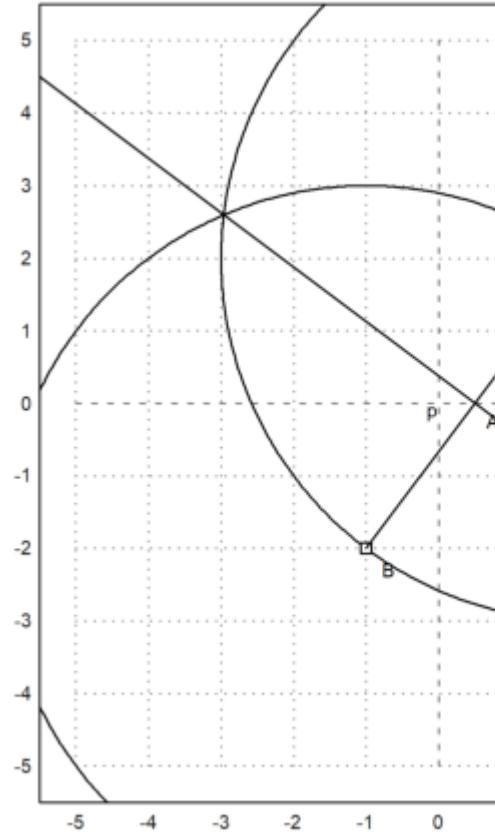
insimg;

### 5.3 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, kita 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 perpotongan cukup rumit. Namun, kita dapat menyederhanakannya, jika kita mencari nilai  $y$ .

```
g = getLineEquation(lineThrough(P1,P2),x,y);
solve(g,y)
```

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sepenuhnya berbeda.

```
solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
h =getLineEquation(lineThrough(A,B),x,y);
solve(h,y)
```

Perhatikan hasil kali gradien garis  $g$  dan  $h$  adalah:

Artinya kedua garis tegak lurus.

Contoh 3: Rumus Heron

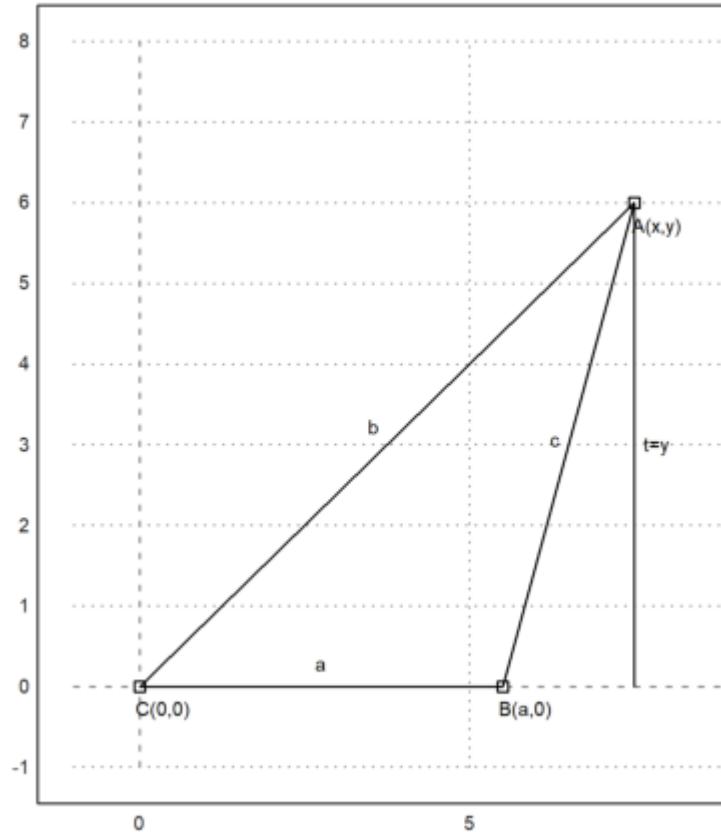
Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi  $a$ ,  $b$  dan  $c$  adalah:

atau bisa ditulis 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);
assume(a>0); sol = solve([x^2 + y^2 = b^2, (x - a)^2 + y^2 = c^2], [x, y])

$$[x = \frac{2ac + b^2}{a^2 + c^2}, y = \frac{2a^2c + 2b^2c + 2ab^2 - a^2b^2}{2a^2c + 2b^2c + 2ab^2 - a^2b^2}]$$

$$[x = \frac{2ac + b^2}{a^2 + c^2}, y = \frac{2a^2c + 2b^2c + 2ab^2 - a^2b^2}{2a^2c + 2b^2c + 2ab^2 - a^2b^2}]$$


```

Ekstrak solusi y.

$y_{sol} = y$  with  $sol[2][2]$ ;  $'y = sqrt(factor(ysol^2))'$

Kita mendapatkan rumus Heron.

function  $H(a,b,c) = sqrt(factor((ysol/2)^2));$   $H(a,b,c)=H(a,b,c)$

'Luas =  $H(2,5,6)/luassegitigadenganpanjangsisi - sisi2,5,6$

Tentu saja, setiap segitiga siku-siku adalah kasus yang terkenal.

$H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5$

6

Dan jelas pula, bahwa ini adalah segitiga dengan luas maksimal dan dua sisinya 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)

Kasus umum juga berfungsi.

```
solve(diff(H(a,b,c)^2,c) = 0, c)
```

Sekarang mari kita cari himpunan semua titik di mana  $b+c=d$  untuk suatu konstanta  $d$ . Diketahui bahwa ini adalah elips.

```
s1 = subst(d-c,b,sol[2]); s1
```

Dan buat fungsi ini.

```
function fx(a,c,d) = rhs(s1[1]); fx(a,c,d), function fy(a,c,d) = rhs(s1[2]); fy(a,c,d)
```

Sekarang kita dapat menggambar himpunannya. Sisi  $b$  bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan elips.

```
aspect(1); plot2d(fx(3,x,5),fy(3,x,5),xmin=1,xmax=4,square=1):
```

Kita dapat memeriksa persamaan umum untuk elips ini, yaitu:

di mana  $(xm,ym)$  adalah pusat, dan  $u$  dan  $v$  adalah sumbu setengah.

```
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])
```

Kita melihat bahwa tinggi dan luas segitiga tersebut adalah maksimum untuk  $x=0$ . Jadi luas segitiga dengan  $a+b+c=d$  adalah maksimum, jika segitiga tersebut sama sisi. Kita ingin memperolehnya secara analitis.

```
eqns = [diff(H(a,b,d-(a+b))^2,a) = 0, diff(H(a,b,d-(a+b))^2,b) = 0]; eqns
```

Kita memperoleh beberapa nilai minimum, yang dimiliki oleh segitiga dengan satu sisi 0, dan solusinya  $a=b=c=d/3$ .

```
solve(eqns,[a,b])
```

Ada juga metode Lagrange, yang memaksimalkan  $H(a,b,c)^2$  terhadap  $a+b+d=d$ .

```
solve([diff(H(a,b,c)^2,a) = la, diff(H(a,b,c)^2,b) = la, ... diff(H(a,b,c)^2,c) = la, a + b + c = d], [a, b, c, la])
```

d d [[a = 0, b = -, c = -, la = 0], 2 2 d d d d [a = -, b = 0, c = -, la = 0], [a = -, b = -, c = 0, la = 0], 2 2 2 2 3 d d d d [a = -, b = -, c = -, la = -]] 3 3 3  
108

Kita bisa membuat plot dari situasinya

Pertama-tama atur titik di Maxima.

```
A = at([x,y],sol[2]); A
```

```
B = [0,0]; B, C = [a, 0]; C
```

Kemudian atur rentang plot dan plot titik-titiknya.

```
setPlotRange(0,5,-2,3); ... a=4; b=3; c=2; ... plotPoint(mxmeval("B"), "B");
plotPoint(mxmeval("C"), "C"); ... plotPoint(mxmeval("A"), "A");
```

Plot segmen.

```
splotSegment(mxmeval("A"),mxmeval("C")); ... plotSegment(mxmeval("B"),mxmeval("C"));
... plotSegment(mxmeval("B"),mxmeval("A"));
```

Hitunglah garis tegak lurus tengah di Maxima.

```
h = middlePerpendicular(A,B); g = middlePerpendicular(B,C);
```

Dan pusat kelilingnya.

```
U = lineIntersection(h,g);
```

Kita mendapatkan rumus untuk jari-jari lingkaran luar.

```
assume(a > 0,b > 0,c > 0); distance(U, B)|radcan
```

Mari kita tambahkan ini ke dalam alur cerita.

```
plotPoint(U()); ... plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)")));
```

Dengan menggunakan geometri, kita peroleh rumus sederhana untuk jari-jari. Kita dapat memeriksa apakah ini benar dengan Maxima. Maxima akan memfaktorkan ini hanya jika kita mengkuadratkannya.

$$c^2 / \sin(\text{computeAngle}(A, B, C))^2 | \text{factor}$$

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari sembarang segitiga yang tidak sama sisi. Garis ini merupakan garis pusat segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga tersebut, termasuk orthocenter, circumcenter, centroid, titik Exeter, dan titik pusat lingkaran sembilan titik pada segitiga tersebut.

Sebagai contoh, kita hitung dan plot garis Euler dalam sebuah segitiga.

Pertama, kita definisikan sudut-sudut segitiga dalam Euler. Kita gunakan definisi, yang terlihat dalam ekspresi simbolik.

$$A ::= [-1, -1]; B ::= [2, 0]; C ::= [1, 2];$$

Untuk memplot objek geometris, kita menyiapkan area plot, dan menambahkan titik-titik ke dalamnya. Semua plot objek geometris ditambahkan ke plot saat ini.

$$\text{setPlotRange}(3); \text{plotPoint}(A, "A"); \text{plotPoint}(B, "B"); \text{plotPoint}(C, "C");$$

Kita juga dapat menambahkan sisi-sisi segitiga.

$$\text{plotSegment}(A, B, ""); \text{plotSegment}(B, C, ""); \text{plotSegment}(C, A, "");$$

Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

$$\text{areaTriangle}(A, B, C)$$

Kita dapat menghitung koefisien sisi c.

$$c = \text{lineThrough}(A, B)$$

$$[-1, 3, -2]$$

Dan dapatkan juga rumus untuk garis ini.

$$\text{getLineEquation}(c, x, y)$$

Untuk bentuk Hesse, kita perlu menentukan suatu titik, sehingga titik tersebut berada di sisi positif bentuk Hesse. Memasukkan titik akan menghasilkan jarak positif ke garis.

$$\text{getHesseForm}(c, x, y, C).at($$

Sekarang kita hitung lingkaran luar ABC.

$$LL = \text{circleThrough}(A, B, C); \text{getCircleEquation}(LL, x, y)$$

$$O = \text{getCircleCenter}(LL); O$$

Gambarkan lingkaran dan titik pusatnya. Cu dan U adalah simbol. Kita evaluasi ekspresi ini untuk Euler.

$$\text{plotCircle}(LL()); \text{plotPoint}(O(), "O"):$$

Kita dapat menghitung perpotongan tinggi di ABC (orthocenter) secara numerik dengan perintah berikut.

$$H = \text{lineIntersection}(\text{perpendicular}(A, \text{lineThrough}(C, B)), \dots, \text{perpendicular}(B, \text{lineThrough}(A, C))); H$$

Sekarang kita dapat menghitung garis Euler dari segitiga tersebut.

$$el = \text{lineThrough}(H, O); \text{getLineEquation}(el, x, y)$$

Tambahkan ke plot kita.

$$\text{plotPoint}(H(), "H"); \text{plotLine}(el(), "Garis Euler"):$$

Pusat gravitasi seharusnya berada pada garis ini.

```
M = (A+B+C)/3; getLineEquation(el,x,y)with[x = M[1],y = M[2]]
```

```
plotPoint(M(),"M"); // titik berat
```

Teori ini memberi tahu kita  $MH=2^*MO$ . Kita perlu menyederhanakannya dengan radcan untuk mencapainya.

```
distance(M,H)/distance(M,O)|radcan
```

Fungsinya termasuk fungsi untuk sudut juga.

```
computeAngle(A,C,B), degprint()
```

```
60°15'18.43"
```

Persamaan untuk pusat lingkaran dalam tidak terlalu bagus.

```
Q = lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))—radcan;
```

*Q*

Mari kita hitung juga ekspresi untuk jari-jari lingkaran dalam.

```
r = distance(Q,projectToLine(Q,lineThrough(A,B)))—ratsimp; r
```

```
LD = circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke dalam alur cerita.

```
color(5); plotCircle(LD());
```

Parabola

Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
p = getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); p =' 0
```

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
plot2d(p,level=0,add=1,contourcolor=6);
```

Ini seharusnya merupakan suatu fungsi, tetapi penyelesaian default Maxima hanya dapat menemukan solusinya, jika kita mengquadratkan persamaannya. Akibatnya, kita mendapatkan solusi palsu.

```
akar = solve(getHesseForm(lineThrough(A,B),x,y,C)^2 - distance([x,y],C)^2, y)
```

```
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26, y = - 3 x + sqrt(70) sqrt(9 - 2 x) +
```

26]

Solusi pertama adalah

```
maxima: akar[1]
```

Dengan menambahkan solusi pertama ke dalam plot, terlihat bahwa itu memang jalur yang kita cari. Teori tersebut memberi tahu kita bahwa itu adalah parabola yang diputar.

```
plot2d(rhs(akar[1]),add=1);
```

```
function g(x) = rhs(akar[1]); 'g(x) = g(x)//fungsiyangmendefinisikan kurvadiatas
```

```
T =[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
```

```
dTC = distance(T,C); fullratsimp(dTC),float(
```

```
U = projectToLine(T,lineThrough(A,B)); U//proyeksi T pada garis AB
```

```
dU2AB = distance(T,U); fullratsimp(dU2AB),float(
```

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

Contoh 5: Trigonometri Rasional

Hal ini terinspirasi dari ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti konsep klasik jarak dan sudut dengan kuadran dan sebaran. Dengan menggunakan konsep ini, memang memungkinkan untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

Berikut ini, saya memperkenalkan konsep-konsep tersebut, dan memecahkan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keuntungan utama trigonometri rasional bahwa perhitungan dapat dilakukan hanya dengan kertas dan pensil. Anda diundang untuk memeriksa hasilnya tanpa komputer.

Intinya adalah bahwa perhitungan simbolik rasional sering kali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

load geometry;

Untuk pengenalan pertama, kami menggunakan segitiga siku-siku dengan proporsi Mesir yang terkenal yaitu 3, 4, dan 5. Perintah berikut adalah perintah Euler untuk memplot geometri bidang yang terdapat dalam file Euler "geometry.e".

```
C:=[0,0]; A:=[4,0]; B:=[0,3]; ... setPlotRange(-1,5,-1,5); ... plot-
Point(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ... plotSegment(B,A,"c");
plotSegment(A,C,"b"); plotSegment(C,B,"a"); ... insimg(30);
```

Tentu saja,

di mana wa adalah sudut di A. Cara yang biasa untuk menghitung sudut ini adalah dengan mengambil kebalikan dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak secara perkiraan.

```
wa := arcsin(3/5); degprint(wa)
36°52'11.63"
```

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadran, yang mengantikan jarak. Faktanya, itu hanyalah kuadrat jarak. Dalam persamaan berikut, a, b, dan c menunjukkan kuadran sisi-sisi.

Teorema Pythagoras menjadi  $a+b=c$ .

$$a = 3^2; b = 4^2; c = 5^2; a + b = c \\ 25 = 25$$

Gagasan kedua trigonometri rasional adalah sebaran. Sebaran mengukur bukaan antara garis. Nilainya 0, jika garis sejajar, dan 1, jika garis persegi panjang. Nilainya adalah kuadrat sinus sudut antara dua garis.

Sebaran garis AB dan AC pada gambar di atas didefinisikan sebagai di mana a dan c adalah kuadran dari setiap segitiga persegi panjang dengan satu sudut di A.

$$sa = a/c; sa$$

Tentu saja, ini lebih mudah dihitung daripada sudut. Namun, Anda kehilangan sifat bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengubah nilai perkiraan untuk sudut wa menjadi sprad, dan mencetaknya sebagai pecahan.

$$\text{fracprint}(\sin(wa)^2)$$

9/25

Hukum kosinus dari trigonometri klasik diterjemahkan ke dalam "hukum silang" berikut.

Di sini  $a$ ,  $b$ , dan  $c$  adalah kuadran sisi-sisi segitiga, dan  $\alpha$  adalah sebaran di sudut  $A$ . Sisi  $a$ , seperti biasa, berseberangan dengan sudut  $A$ .

Hukum-hukum ini diimplementasikan dalam berkas geometry.e yang kami muat ke Euler.

*crosslaw(aa, bb, cc, saa)*

Dalam kasus kami, kami mendapatkan

*crosslaw(a, b, c, sa)*

Mari kita gunakan hukum silang ini untuk menemukan sebaran di  $A$ . Untuk melakukannya, kita buat hukum silang untuk kuadran  $a$ ,  $b$ , dan  $c$ , dan selesaikan untuk sebaran yang tidak diketahui  $\alpha$ .

Anda dapat melakukannya dengan mudah secara manual, tetapi saya menggunakan Maxima. Tentu saja, kita mendapatkan hasil yang sudah kita miliki.

*crosslaw(a, b, c, x), solve()*

Kita sudah tahu ini. Definisi sebaran adalah kasus khusus dari hukum silang.

Kita juga dapat memecahkan ini untuk  $a, b, c$  umum. Hasilnya adalah rumus yang menghitung sebaran sudut segitiga yang diberikan kuadran ketiga sisinya.

*solve(crosslaw(aa, bb, cc, x), x)*

Kita dapat membuat fungsi dari hasil tersebut. Fungsi tersebut telah didefinisikan dalam berkas geometry.e milik Euler.

*spread(a, b, c)*

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi-sisi

Hasilnya rasional, yang tidak mudah didapat jika kita menggunakan trigonometri klasik.

*spread(a, a, 4a/7)*

Ini adalah sudut dalam derajat.

*deprint(arcsin(sqrt(6/7)))*

$67^\circ 47' 32.44''$

Contoh Lain

Sekarang, mari kita coba contoh yang lebih maju.

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 fungsi *distance* dari file Euler untuk geometri. Fungsi *distance* menggunakan geometri klasik.

*distance(A, B)*

Euler juga memuat fungsi untuk kuadran antara dua titik.

Dalam contoh berikut, karena  $c+b$  bukan  $a$ , segitiga tersebut bukan persegi panjang.

$c = \text{quad}(A, B); c, b = \text{quad}(A, C); b, a = \text{quad}(B, C); a,$

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan perkalian titik dua vektor. Hasilnya adalah beberapa perkiraan floating point.

```
wb = computeAngle(A,B,C); wb,(wb/pi180)()
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(
```

Yaitu, apa yang dilakukan fungsi sebaran yang didefinisikan dalam "geometry.e".

```
sb = spread(b,a,c); sb
```

Maxima memperoleh hasil yang sama dengan menggunakan trigonometri biasa, jika kita memaksakannya. Maxima memang menyelesaikan suku  $\sin(\arccos(...))$  menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

```
sin(computeAngle(A,B,C))^2
```

Setelah kita memiliki sebaran di B, kita dapat menghitung tinggi ha pada sisi a. Ingat bahwa

menurut definisi.

```
ha = csb; ha
```

Gambar berikut ini dibuat dengan program geometri C.a.R., yang dapat menggambar kuadran dan sebaran.

```
image: (20) RationalGeometryCar.png
```

Menurut definisi, panjang ha adalah akar kuadrat dari kuadrannya.

```
sqrt(ha)
```

Sekarang kita bisa menghitung luas segitiga. Jangan lupa, bahwa kita sedang membahas tentang kuadran!

```
sqrt(ha)sqrt(a)/2
```

Rumus determinan yang biasa menghasilkan hasil yang sama.

```
areaTriangle(B,A,C)
```

Rumus Heron

Sekarang, mari kita selesaikan masalah ini secara umum!

```
remvalue(a,b,c,sb,ha);
```

Pertama-tama kita hitung sebaran di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita hitung luas kuadrat (yang disebut "quadrea"?), faktorkan dengan Maxima, dan kita dapatkan rumus Heron yang terkenal.

Memang, ini sulit dilakukan dengan pensil dan kertas.

```
spread(b^2,c^2,a^2),factor(
```

Aturan Triple Spread

Kerugian spread adalah tidak lagi sekadar menambahkan sudut yang sama.

Namun, tiga spread segitiga memenuhi aturan "triple spread" berikut.

```
remvalue(sa,sb,sc); triplespread(sa,sb,sc)
```

Aturan ini berlaku untuk tiga sudut yang jumlahnya mencapai  $180^\circ$ .

Karena sebaran

sama, aturan sebaran rangkap tiga juga berlaku, jika

Karena sebaran sudut negatif sama, aturan sebaran rangkap tiga juga berlaku, jika

Misalnya, kita dapat menghitung sebaran sudut  $60^\circ$ . Yaitu  $3/4$ . Persamaan tersebut memiliki solusi kedua, di mana semua sebarannya adalah 0.

```
solve(triplespread(x,x,x),x)
```

Sebaran  $90^\circ$  jelas adalah 1. Jika dua sudut dijumlahkan menjadi  $90^\circ$ , sebarannya memecahkan persamaan sebaran rangkap tiga dengan a,b,1. Dengan perhitungan berikut kita memperoleh  $a+b=1$ .

```
triplespread(x,y,1),solve(
```

Karena sebaran  $180^\circ-t$  sama dengan sebaran  $t$ , rumus sebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau selisih dari dua sudut lainnya.

Jadi kita dapat menemukan sebaran sudut yang digandakan. Perhatikan bahwa ada dua solusi lagi. Kita buat ini menjadi fungsi.

```
solve(triplespread(a,a,x),x),functiondoublespread(a)=factor(rhs(-4 (a - 1) a
```

Garis Bagi Sudut

Kita sudah tahu situasinya seperti ini.

```
C:=[0,0]; A:=[4,0]; B:=[0,3]; ... setPlotRange(-1,5,-1,5); ... plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ... plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ... insimg;
```

Mari kita hitung panjang garis bagi sudut di A. Namun, kita ingin menyelesaikannya untuk a,b,c umum.

```
remvalue(a,b,c);
```

Jadi pertama-tama kita hitung sebaran sudut yang dibagi dua di A, menggunakan rumus sebaran rangkap tiga.

Masalah dengan rumus ini muncul lagi. Rumus ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut yang dibagi dua  $180^\circ-wa$ .

```
triplespread(x,x,a/(a + b)),solve(
```

Mari kita periksa persegi panjang Mesir.

```
sa2with[a = 3^2, b = 4^2]
```

Kita dapat mencetak sudut dalam Euler, setelah mentransfer sebaran ke radian.

```
wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

$18^\circ 26' 5.82''$

Titik P merupakan 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-sudut pada contoh spesifik kita.

```
computeAngle(C,A,P), computeAngle(P,A,B)
```

```
0.321750554397 0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.

Kita gunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

berlaku di sembarang segitiga. Kuadratkan, maka akan menghasilkan apa yang disebut "hukum sebaran"

di mana  $a,b,c$  menunjukkan kuadran.

Karena CPA sebaran adalah  $1-sa^2$ , kita peroleh darinya  $bisa/1=b/(1-sa^2)$  dan dapat menghitung bisa (kuadran garis bagi sudut).

$factor(ratsimp(b/(1-sa^2))); bisa =$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

$sqrt(mxmeval("at(bisa,[a=3^2, b = 4^2])")), distance(A, P)$

4.21637021356 4.21637021356

Kita juga dapat menghitung P menggunakan rumus spread.

$py=factor(ratsimp(sa2bisa)); py$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

$sqrt(mxmeval("at(py,[a=3^2, b = 4^2])"))$

1.333333333333

Sudut Tali

Perhatikan situasi berikut.

```
setPlotRange(1.2); ... color(1); plotCircle(circleWithCenter([0,0],1)); ...
A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ... plotPoint(A,"A");
plotPoint(B,"B"); plotPoint(C,"C"); ... color(3); plotSegment(A,B,"c"); plot-
Segment(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 memecahkan rumus penyebaran rangkap tiga untuk sudut-sudut di pusat O untuk r. Dengan demikian, kita memperoleh rumus untuk jari-jari kuadrat pericircle dalam bentuk kuadran sisi-sisinya.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

```
remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
rabc = rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]);
```

$rabc$

Kita dapat menjadikannya fungsi Euler.

$function periradius(a,b,c) = rabc;$

Mari kita periksa hasilnya untuk titik A, B, C.

```
a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Radiusnya memang 1.

$periradius(a,b,c)$

1

Faktanya, sebaran CBA hanya bergantung pada b dan c. Ini adalah teorema sudut tali busur.

$spread(b, a, c)rabc|ratsimp$

Faktanya, sebarannya adalah  $b/(4r)$ , dan kita melihat bahwa sudut tali busur b adalah setengah sudut pusat.

$doublespread(b/(4r)) - spread(b, r, r)|ratsimp$

Contoh 6: Jarak Minimal pada Bidang

Catatan awal

Fungsi yang, pada titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis datar yang agak 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"):
```

dan grafiknya cukup sederhana: bagian atas kerucut:

```
plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```

Tentu saja minimum 0 dicapai di A.

Dua titik

Sekarang kita lihat fungsi MA+MB di mana A dan B adalah dua titik (tetap). Merupakan "fakta yang diketahui" bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali untuk minimum AB yang konstan pada segmen [AB]:

```
B=[1,-1];
function d2(x,y):=d1(x,y)+sqrt((x-B[1])2 + (y - B[2])2)
fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```

Grafiknya lebih menarik:

```
plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```

Pembatasan pada garis (AB) lebih terkenal:

```
plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

Tiga poin

Sekarang semuanya menjadi kurang sederhana: Tidak banyak yang tahu bahwa MA+MB+MC mencapai nilai minimumnya di satu titik bidang, tetapi menentukannya tidaklah sesederhana itu:

- 1) Jika salah satu sudut segitiga ABC lebih dari  $120^\circ$  (misalkan di A), maka nilai minimumnya tercapai di titik ini (misalkan AB+AC).

Contoh:

```
C=[-4,1];
function d3(x,y):=d2(x,y)+sqrt((x-C[1])2 + (y - C[2])2)
plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
insimg;
fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The mini-
mum is on A");
```

```
P=(A_B_C_A); plot2d(P[1],P[2],add=1,color=12);
insimg;
```

- 2) Namun jika semua sudut segitiga ABC kurang dari  $120^\circ$ , maka nilai minimumnya berada di titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang sudut-sudut sisi ABC-nya sama (masing-masing sudutnya  $120^\circ$ ):

```
C=[-0.5,1];
plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2);
fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat
point");
```

```
P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
insimg;
```

Merupakan aktivitas yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya mengetahui perangkat lunak yang ditulis dalam Java yang memiliki instruksi "garis kontur" ...

Semua ini ditemukan oleh seorang hakim Prancis bernama Pierre de Fermat; ia menulis surat kepada para dilettan lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di pajak penghasilan. Jadi titik unik F sehingga  $FA+FB+FC$  minimal, disebut titik Fermat dari segitiga tersebut. Namun tampaknya beberapa tahun sebelumnya, Torricelli dari Italia telah menemukan titik ini sebelum Fermat menemukannya! Bagaimanapun tradisinya adalah mencatat titik F ini...

Empat titik

Langkah berikutnya adalah menambahkan titik ke-4 D dan mencoba meminimalkan  $MA+MB+MC+MD$ ; katakanlah Anda adalah operator TV kabel dan ingin mencari di bidang mana Anda harus meletakkan antena Anda sehingga Anda dapat menyalurkan sinyal ke empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
D=[1,1];
function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2 + (y - D[2])^2)
plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5);
fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
```

insimg;

Masih terdapat nilai minimum dan tidak tercapai di titik A, B, C, maupun D:

```
function f(x):=d4(x[1],x[2])
neldermin("f",[0.2,0.2])
[0.142858, 0.142857]
```

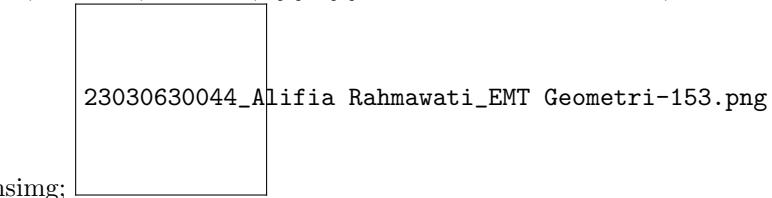
Tampaknya dalam kasus ini, koordinat titik optimal bersifat rasional atau mendekati rasional...

Sekarang ABCD adalah persegi, kita mengharapkan bahwa titik optimal akan menjadi pusat ABCD:

```
C=[-1,1];
```

plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1); fcontour("d4",xmin=-

```
1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);
```



insimg;

Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan pvengine.exe di jalur program.

Pertama, kita hitung jari-jari bola.

Jika Anda melihat gambar di bawah, Anda melihat bahwa kita memerlukan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file geometry.e milik Euler untuk ini.

load geometry;

Pertama dua garis membentuk kerucut.

```
g1 = lineThrough([0,0],[1,a])
[- a, 1, 0]
```

```
g2 = lineThrough([0,0],[-1,a])
[- a, - 1, 0]
```

Lalu baris ketiga.

```
g = lineThrough([-1,0],[1,1])
[- 1, 2, 1]
```

Kita merencanakan segalanya sejauh ini.

```
setPlotRange(-1,1,0,2);
color(black); plotLine(g(),"")
```

```
a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```

Sekarang kita ambil titik umum pada sumbu y.

P = [0,u]

[0, u]

Hitunglah jarak ke g1.

d1 = distance(P,projectToLine(P,g1)); d1

Hitunglah jarak ke g.

d = distance(P,projectToLine(P,g)); d

Dan temukan pusat kedua lingkaran, yang jaraknya sama.

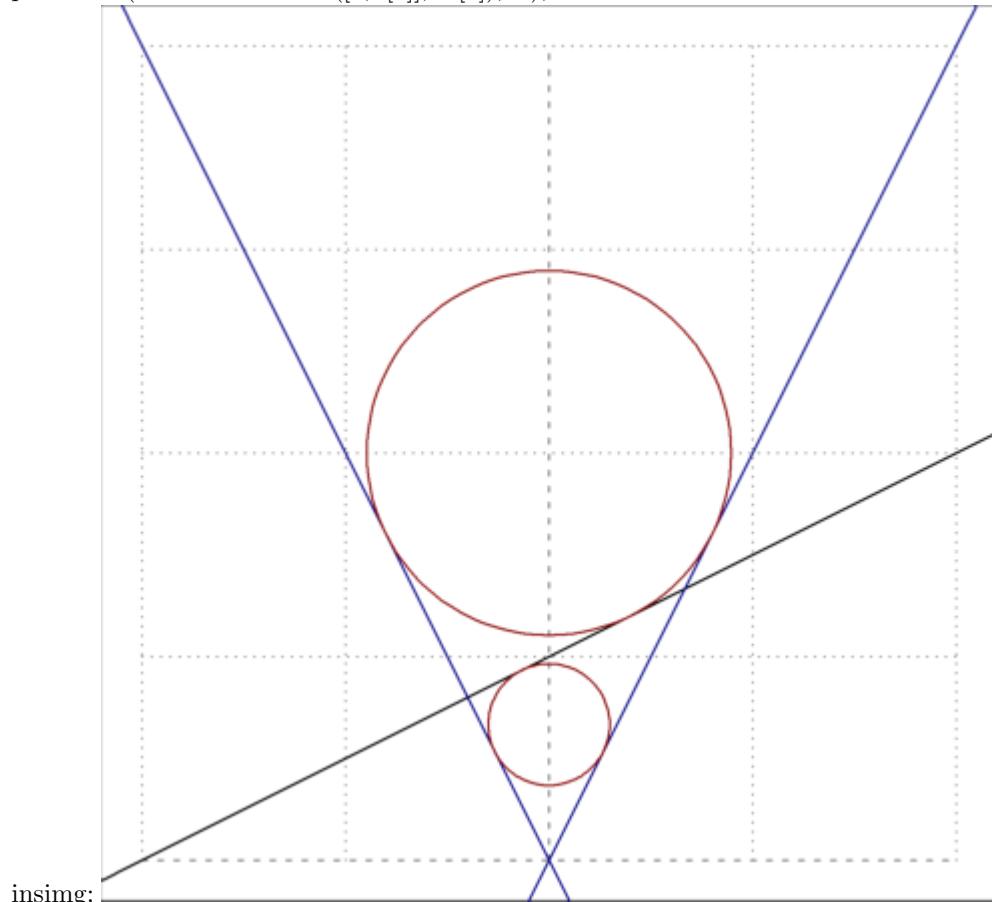
sol = solve(d1^2 = d^2, u);sol

Ada dua solusi.

The image is a small square frame containing the text "23030630044\_Alifia Rahmawati\_EMT Geometri-153.png".

Kita mengevaluasi solusi simbolik, dan menemukan kedua pusat, dan kedua jarak.

```
u := sol()
[0.333333, 1]
dd := d()
[0.149071, 0.447214]
Gambarkan lingkaran-lingkaran tersebut ke dalam gambar.
color(red);
plotCircle(circleWithCenter([0,u[1]],dd[1]),"");
plotCircle(circleWithCenter([0,u[2]],dd[2]),"");
```



Plot dengan Povray

Selanjutnya kita plot semuanya dengan Povray. Perhatikan bahwa Anda mengubah perintah apa pun dalam urutan perintah Povray berikut, dan menjalankan kembali semua perintah dengan Shift-Return.

Pertama-tama kita memuat fungsi povray.  
load povray;  
defaultpovray="C:

Program Files

POV-Ray

v3.7

bin

pengine.exe"

C:\Files-Ray3.7.exe

Kami menyiapkan suasannya dengan tepat.

```
povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Berikutnya kita menulis kedua bola itu ke dalam 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)));
```

Kita buat bidang yang dibatasi 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 buat dua titik pada lingkaran, di mana bola menyentuh kerucut.

```
function turnz(v) := return [-v[2],v[1],v[3]]
```

```
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 buat dua titik tempat bola-bola tersebut menyentuh bidang.

Titik-titik ini adalah fokus elips.

```
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
```

```
writeln(povpoint(P3,povlook(yellow)));
```

```
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
```

```
writeln(povpoint(P4,povlook(yellow)));
```

Berikutnya kita hitung perpotongan P1P2 dengan bidang.

```
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)(P2-P1);
```

```
writeln(povpoint(P5,povlook(yellow)));
```

Kita 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 buat pita abu-abu, di mana bola-bola menyentuh kerucut.

```
pcw=povcone([0,0,0],0,[0,0,a],1.01);
```

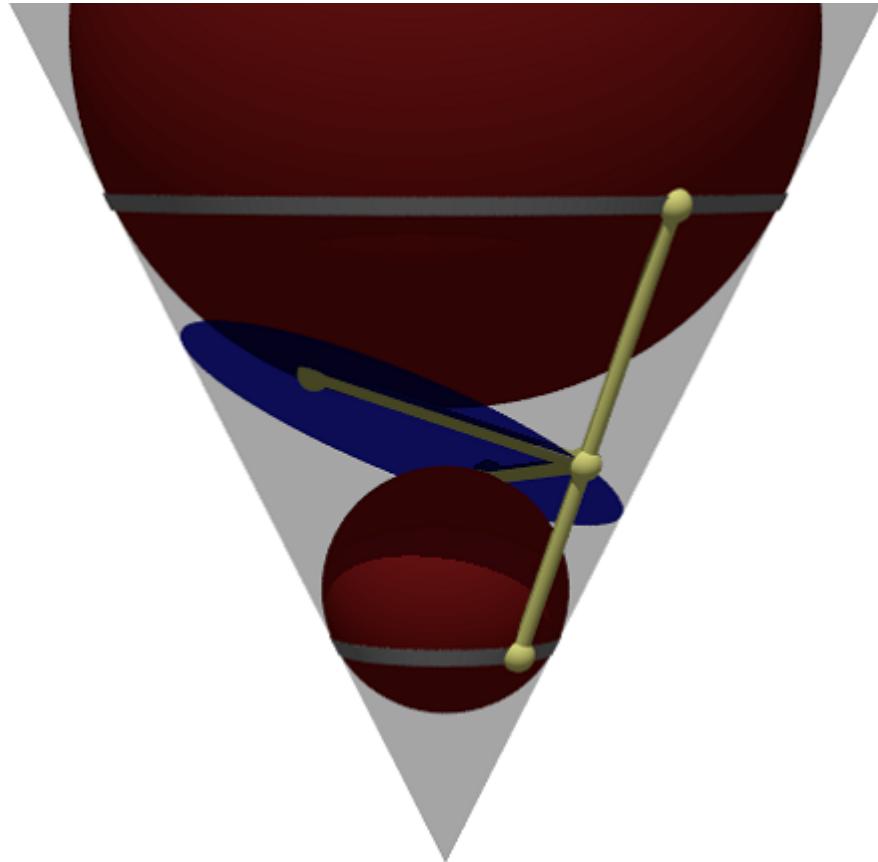
```
pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
```

```
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();
```

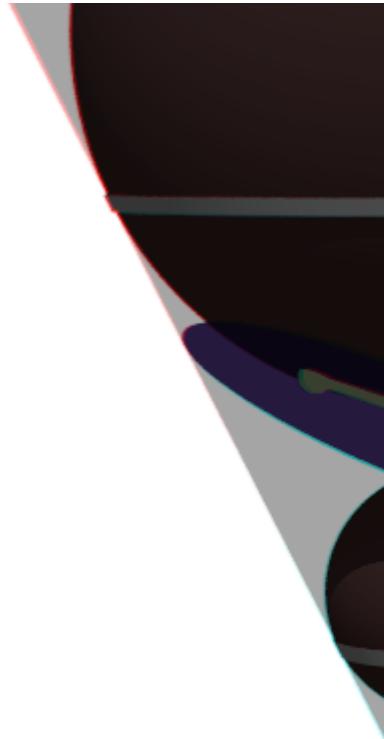
Untuk mendapatkan Anaglyph ini, kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali nanti.

```
function scene () ...
```

```
global a,u,dd,g,g1,defaultPointSize; writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
writeln(povsphere([0,0,u[2]],dd[2],povlook(red))); writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
gp=g(); pc=povcone([0,0,0],0,[0,0,a],1,""); vp=[gp[1],0,gp[2]]; dp=gp[3]; writeln(povplane(vp,dp,povlook(blue,0));
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]); writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]); writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]]; writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]]; writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1); writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow))); writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow))); pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-defaultPointSize/2],[0,0,P1[3]+defaultPointSize/2],1);
writeln(povintersection([pcw,pc1],povlook(gray))); pc2=povcylinder([0,0,P2[3]-
defaultPointSize/2],[0,0,P2[3]+defaultPointSize/2],1); writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction
```

*i/pre\_* Anda memerlukan kacamata merah/cyan untuk menghargai

efek berikut.



```
povaganlyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Contoh 8: Geometri Bumi

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan sferis. Fungsi-fungsi tersebut terdapat dalam berkas "spherical.e" di folder contoh. Kami perlu memuat berkas tersebut terlebih dahulu.

```
load "spherical.e";
```

Untuk memasukkan posisi geografis, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut 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 bulat).

```
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  $7^{\circ}34.333'$  E  $110^{\circ}49.683'$  S  $6^{\circ}59.050'$  E  $110^{\circ}24.533'$

Pertama, kita hitung vektor dari satu ke yang lain pada bola ideal. Vektor ini adalah [arah, jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang  $7^{\circ}$ .

```
br=svector(FMIPA,Solo); degprint(br[1]), br[2]rearth(7°)- km // perkiraan
jarak FMIPA-Solo
```

$65^{\circ}20'26.60''$  53.8945384608

Ini adalah perkiraan yang bagus. Rutin berikut menggunakan perkiraan yang lebih baik lagi. Pada jarak yang pendek, hasilnya hampir sama.

```
esdist(FMIPA, Semarang)- " km"; // perkiraan jarak FMIPA-Semarang
```

Ada fungsi untuk judul, yang memperhitungkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
sdegrprint(esdir(FMIPA,Solo))
```

$65.34^{\circ}$

Sudut suatu segitiga melebihi  $180^{\circ}$  pada bola.

```
asum=sangle(Solo,FMIPA, Semarang)+sangle(FMIPA,Solo, Semarang)+sangle(FMIPA, Semarang, Solo);
degprint(asum)
```

$180^{\circ}0'10.77''$

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, ini tidak akurat karena kesalahan pengurangan dalam asum-pi.

```
(asum-pi)rearth(48°) 2 - "km 2 "; // perkiraan luas segitiga FMIPA - Solo -
Semarang
```

Ada fungsi untuk ini, yang menggunakan lintang rata-rata segitiga untuk menghitung jari-jari bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

```
esarea(Solo,FMIPA, Semarang)- " km 2 ", // perkiraan yang sama dengan fungsi esarea()
2123.64310526 km 2
```

Kita juga dapat menambahkan vektor ke posisi. Vektor berisi arah dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kita menggunakan svector. Untuk menambahkan vektor ke posisi, kita menggunakan sposprint.

```
v=svector(FMIPA,Solo); sposprint(saddvector(FMIPA,v)), sposprint(Solo),
S $7^{\circ}34.333'$ E $110^{\circ}49.683'$ S $7^{\circ}34.333'$ E $110^{\circ}49.683'$
```

Fungsi-fungsi ini mengasumsikan bentuk bola yang ideal. Sama halnya di bumi.

```
sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(Solo),
S $7^{\circ}34.333'$ E $110^{\circ}49.683'$ S $7^{\circ}34.333'$ E $110^{\circ}49.683'$
```

Mari kita lihat contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];
```

```
sposprint(Tugu), sposprint(Monas)
```

$S 7^{\circ}46.998'$  E  $110^{\circ}21.966'$  S  $6^{\circ}10.500'$  E  $106^{\circ}48.717'$

Menurut Google Earth, jaraknya adalah 429,66 km. Kami memperoleh perkiraan yang baik.

```
esdist(Tugu,Monas)- " km"; // perkiraan jarak Tugu Jogja - Monas Jakarta
Judulnya sama dengan yang dihitung di Google Earth.
```

```
degprint(esdir(Tugu,Monas))
```

$294^\circ 17' 2.85''$

Akan tetapi, kita tidak lagi memperoleh posisi target yang tepat, jika kita menambahkan arah dan jarak ke posisi awal. Hal ini terjadi karena kita tidak menghitung fungsi invers secara tepat, tetapi mengambil perkiraan radius bumi di sepanjang lintasan.

```
sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

S  $6^\circ 10.500'$  E  $106^\circ 48.717'$

Namun, kesalahannya tidak besar.

```
sposprint(Monas),
```

S  $6^\circ 10.500'$  E  $106^\circ 48.717'$

Tentu saja, kita tidak dapat berlayar dengan arah yang sama dari satu tujuan ke tujuan lain, jika kita ingin mengambil jalur terpendek. Bayangkan, Anda terbang ke arah timur laut mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti arah yang konstan!

Perhitungan berikut menunjukkan bahwa kita jauh dari tujuan yang benar, jika kita menggunakan arah yang sama selama perjalanan kita.

```
dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 dikalikan sepersepuluh jaraknya, dengan memakai arah ke Monas, kita sampai di Tugu.

```
p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya sangat jauh.

```
sposprint(p), skmpprint(esdist(p,Monas))
```

S  $6^\circ 11.250'$  E  $106^\circ 48.372'$  1.529km

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

```
P1=[$30^\circ, 10^\circ$]; P2=[$30^\circ, 50^\circ$];
```

Lintasan terpendek dari P1 ke P2 bukanlah lingkaran lintang  $30^\circ$ , tetapi lintasan yang lebih pendek yang dimulai  $10^\circ$  lebih jauh ke utara di P1.

```
sdeprint(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 total jarak.

```
p=P1; dist=esdist(P1,P2); ... loop 1 to 10; dir=esdir(p,P2); sdeprint(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 arah yang sama terlalu lama.

```
skmpprint(esdist(p,P2))
```

0.203km

Kita memperoleh perkiraan yang baik, jika kita menyesuaikan arah setelah setiap  $1/100$  jarak total dari Tugu ke Monas.

```
p=Tugu; dist=esdist(Tugu,Monas); ... loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100);
end;
```

```
skmpprint(esdist(p,Monas))
```

0.000km

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS sepanjang lingkaran besar menuju Monas dengan fungsi navigasi.

```
load spherical; v=navigate(Tugu,Monas,10); ... loop 1 to rows(v); sposprint(v[]),
end;
```

```
S 7°46.998' E 110°21.966' S 7°37.422' E 110°0.573' S 7°27.829' E 109°39.196'
S 7°18.219' E 109°17.834' S 7°8.592' E 108°56.488' S 6°58.948' E 108°35.157' S
6°49.289' E 108°13.841' S 6°39.614' E 107°52.539' S 6°29.924' E 107°31.251' S
6°20.219' E 107°9.977' S 6°10.500' E 106°48.717'
```

Kita menulis suatu 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 i/pre; Sekarang rencanakan semuanya.
```

```
plot3d("testplot",angle=25, height=6, own, user,zoom=4);
```

Atau gunakan plot3d untuk mendapatkan tampilan anaglifnya. Ini tampak sangat bagus dengan kaca mata merah/biru kehijauan.

```
plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4);
```

Latihan

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

Petunjuk:

\* Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah \*  $(360/n)$ .

\* Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n \* dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar \* kelipatan  $(360/n)$ .

\* Untuk n ganjil, pilih salah satu titik sudut adalah di atas.

\* Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik \* pusat.

\* Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

```
load geometry
```

Numerical and symbolic geometry.

```
setPlotRange(-3.5,3.5,-3.5,3.5);
```

```
A=[-2,-2]; plotPoint(A,"A");
```

```
B=[2,-2]; plotPoint(B,"B");
```

```
C=[0,3]; plotPoint(C,"C");
```

```
plotSegment(A,B,"c");
```

```
plotSegment(B,C,"a");
```

```
plotSegment(A,C,"b");
```

```
aspect(1);
```

```
c=circleThrough(A,B,C);
```

```
R=getCircleRadius(c);
```

```
O=getCircleCenter(c);
```

```
plotPoint(O,"O");
```

```
l=angleBisector(A,C,B);
```

```
color(2); plotLine(l); color(1);
```

plotCircle(c,"Lingkaran luar segitiga ABC"):

2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:

- Misalkan persamaan parabolanya  $y = ax^2 + bx + c$ .

- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.

- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

load geometry;

setPlotRange(5); P=[2,0]; Q=[4,0]; R=[0,-4];

plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R");

sol = solve([a+b=-c,16a+4b=-c,c=-4],[a,b,c])

[[a = - 1, b = 5, c = - 4]]

function  $y = -x^2 + 5x - 4$

$2 - x + 5x - 4$

plot2d("-x^2 + 5x - 4", -5, 5, -5, 5) :

3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.

- Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung (sisinya merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).

- Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya bertemu di satu titik.

- Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya.

- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

load geometry

Numerical and symbolic geometry.

setPlotRange(-4.5,4.5,-4.5,4.5);

A=[-3,-3]; plotPoint(A,"A");

B=[3,-3]; plotPoint(B,"B");

C=[3,3]; plotPoint(C,"C");

D=[-3,3]; plotPoint(D,"D");

plotSegment(A,B,"");

plotSegment(B,C,"");

plotSegment(C,D,"");

plotSegment(A,D,"");

aspect(1):

l=angleBisector(A,B,C);

m=angleBisector(B,C,D);

P=lineIntersection(l,m);

color(5); plotLine(l); plotLine(m); color(1);

plotPoint(P,"P");

Dari gambar diatas terlihat bahwa keempat garis bagi sudutnya bertemu di satu titik yaitu titik P.

r=norm(P-projectToLine(P,lineThrough(A,B)));

plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat ABCD"):

Dari gambar diatas, terlihat bahwa sisi-sisinya merupakan garis singgung lingkaran yang sama yaitu lingkaran dalam segiempat.

Akan ditunjukkan bahwa hasil kali panjang sisi-sisi yang berhadapan sama.

$$AB = \text{norm}(A-B) // \text{panjang sisi } AB$$

6

$$CD = \text{norm}(C-D) // \text{panjang sisi } CD$$

6

$$AD = \text{norm}(A-D) // \text{panjang sisi } AD$$

6

$$BC = \text{norm}(B-C) // \text{panjang sisi } BC$$

6

$$AB \cdot CD$$

36

$$AD \cdot BC$$

36

Terbukti bahwa hasil kali panjang sisi-sisi yang berhadapan sama yaitu 36. Jadi dapat dipastikan bahwa segiempat tersebut merupakan segiempat garis singgung.

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).

Penyelesaian :

Diketahui kedua titik fokus  $P = [-1, -1]$  dan  $Q = [1, -1]$

$$P = [-1, -1]; Q = [1, -1];$$

$$\text{function } d1(x,y) := \sqrt{(x-P[1])^2 + (y - P[2])^2}$$

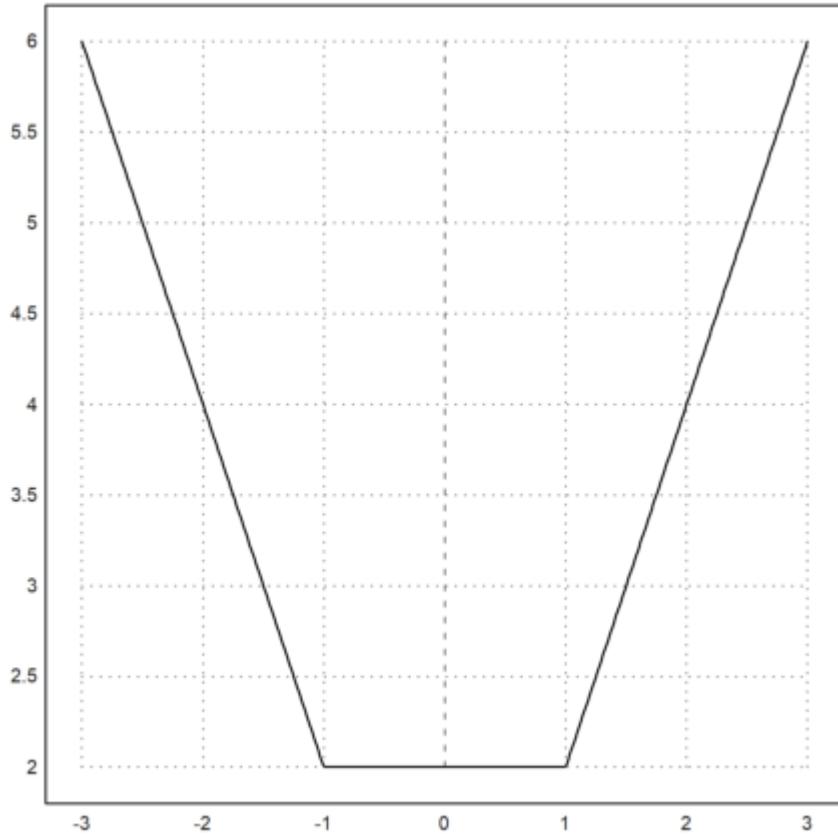
$$Q = [1, -1]; \text{ function } d2(x,y) := \sqrt{(x-P[1])^2 + (y - P[2])^2} + \sqrt{(x-Q[1])^2 + (y - Q[2])^2}$$

fcontour("d2", xmin=-2, xmax=2, ymin=-3, ymax=1, hue=1):

Grafik yang lebih menarik

plot3d("d2", xmin=-2, xmax=2, ymin=-3, ymax=1):

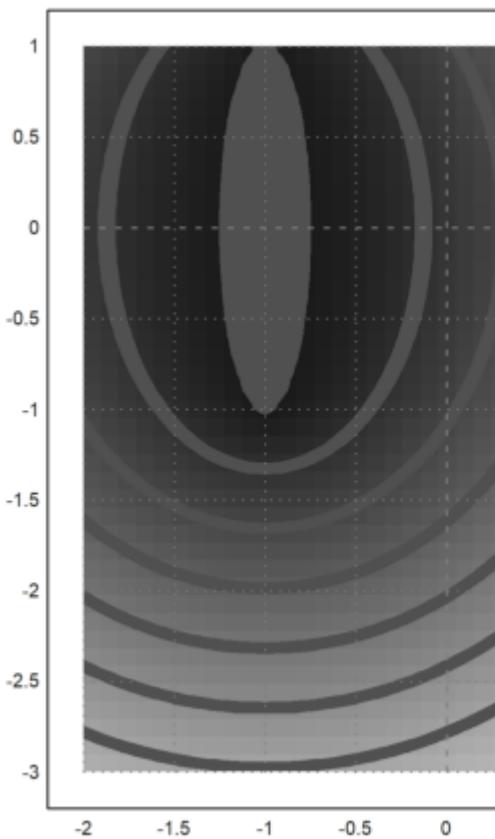
Batasan ke garis PQ



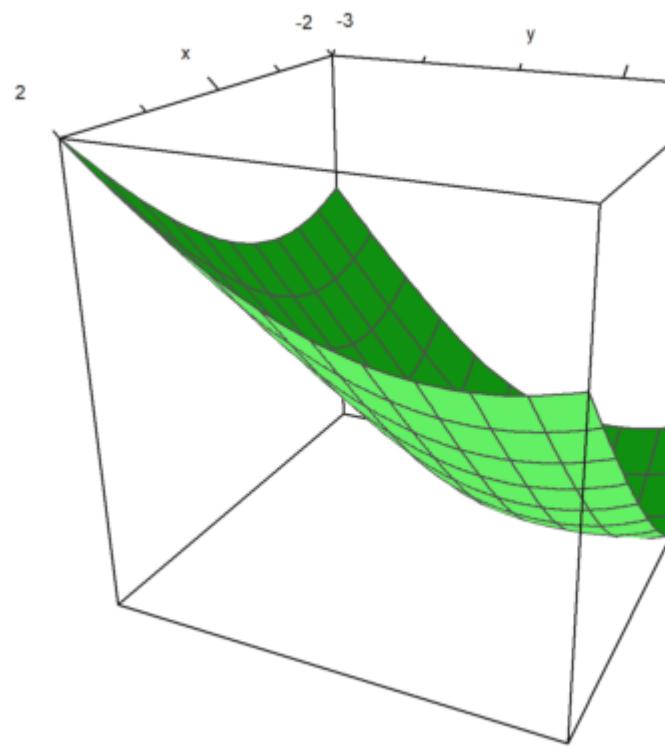
```
plot2d("abs(x+1)+abs(x-1)"
3,xmax=3);
```

5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

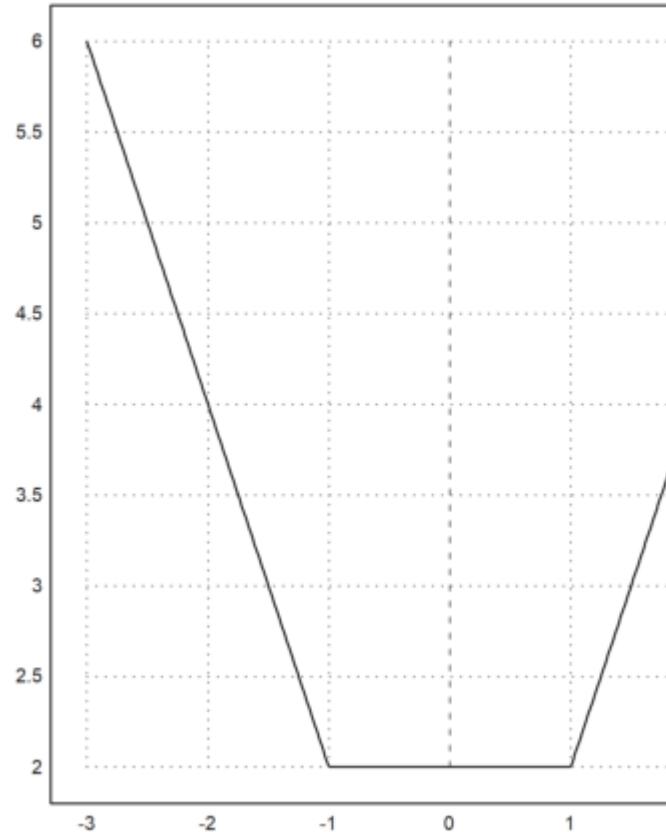
```
P=[-1,-1]; Q=[1,-1];
function d1(x,y):=sqrt((x-p[1])2+(y-p[2])2)
Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])2+(y-P[2])2)+sqrt((x+Q[1])2+
(y+Q[2])2)
```



fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):



plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):



plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):

## 6 Statistika

### 6.1 EMT untuk Statistik

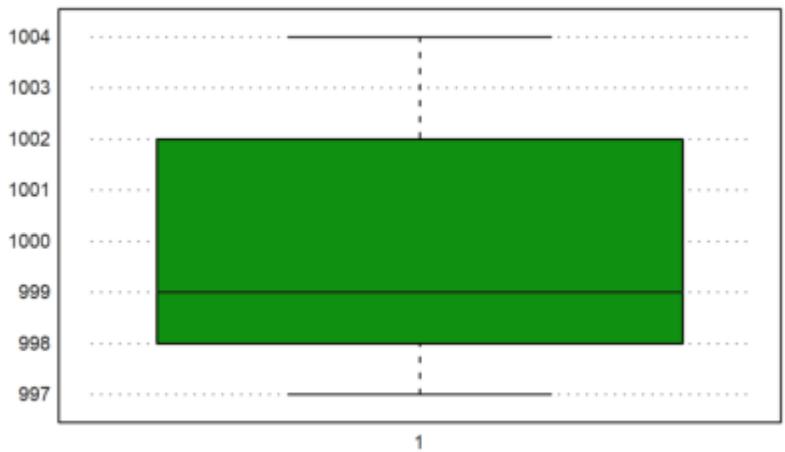
Dalam buku catatan ini, kami menunjukkan plot, pengujian, dan distribusi statistik utama dalam Euler.

Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar statistik. Jadi, Anda mungkin memerlukan beberapa latar belakang untuk memahami detailnya.

Asumsikan pengukuran berikut. Kami ingin menghitung nilai rata-rata dan deviasi standar yang diukur.

$M=[1000,1004,998,997,1002,1001,998,1004,998,997]; \dots$  median(M), mean(M), dev(M),  
 999 999.9 2.72641400622

Kita dapat membuat diagram kotak dan kumis untuk data tersebut. Dalam kasus kita, tidak ada outlier.



aspect(1.75); boxplot(M):

Kami menghitung probabilitas bahwa suatu nilai lebih besar dari 1005, dengan asumsi nilai terukur dari distribusi normal.

Semua fungsi untuk distribusi dalam Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

$$normaldis(x, m, d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} dt.$$

Kami mencetak hasil dalam cetak.

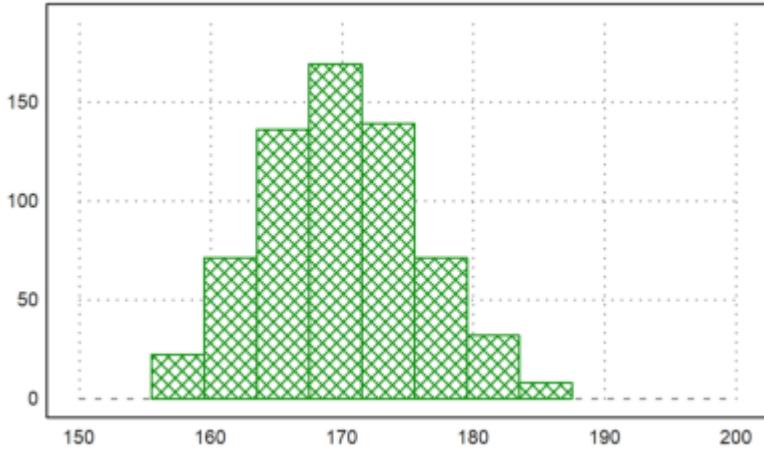
```
print((1-normaldis(1005,mean(M),dev(M))))100,2,unit=""
3.07
```

Untuk contoh berikutnya, kami mengasumsikan jumlah pria berikut dalam rentang ukuran tertentu.

r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];

Berikut adalah plot distribusinya.

```
plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style=""')
":)
```



Kita dapat memasukkan data mentah tersebut ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal rentang, akhir rentang, jumlah orang dalam rentang.

Tabel dapat dicetak dengan tajuk. Kita menggunakan vektor string untuk mengatur tajuk.

```
T:=r[1:8]' — r[2:9]' — v'; writetable(T,labc=[“BB”, “BA”, “Frek”])
BB BA Frek
155.5 159.5 22
159.5 163.5 71
163.5 167.5 136
167.5 171.5 169
171.5 175.5 139
175.5 179.5 71
179.5 183.5 32
183.5 187.5 8
```

Jika kita memerlukan nilai rata-rata dan statistik ukuran lainnya, kita perlu menghitung titik tengah rentang. Kita dapat menggunakan dua kolom pertama tabel kita untuk ini.

Simbol “—” digunakan untuk memisahkan kolom, fungsi “writetable” digunakan untuk menulis tabel, dengan opsi ”labc” untuk menentukan tajuk 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
```

Namun 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,
```

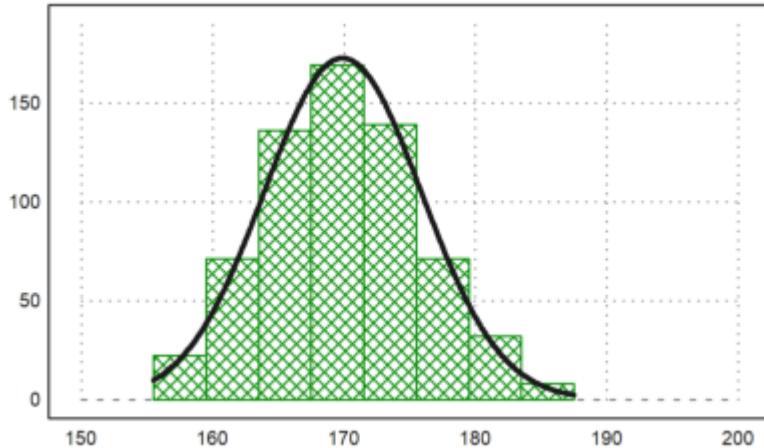
```
169.901234568 5.98912964449
```

Mari kita tambahkan distribusi normal nilai-nilai tersebut ke diagram batang di atas. Rumus untuk distribusi normal dengan rata-rata  $m$  dan simpangan baku  $d$  adalah:

$$y = \frac{1}{d\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2d^2}}.$$

Karena nilainya berada di antara 0 dan 1, untuk memplotnya pada diagram batang, nilainya harus dikalikan dengan 4 kali jumlah total data.

```
plot2d("qnormal(x,m,d)sum(v)4", ... xmin=min(r),xmax=max(r),thickness=3,add=1):
```



Tabel

Di direktori buku catatan ini, Anda akan menemukan berkas dengan tabel. Data tersebut merupakan hasil survei. Berikut adalah empat baris pertama berkas tersebut. Data tersebut berasal dari buku daring Jerman "Einführung in die Statistik mit R" karya A. Handl.

```
printfile("table.dat",4);
```

Person Sex Age Titanic Evaluation Tip Problem 1 m 30 n . 1.80 n 2 f 23 y g 1.80 n 3 f 26 y g 1.80 y

Tabel berisi 7 kolom angka atau token (string). Kita ingin membaca tabel dari file. Pertama, kita menggunakan terjemahan kita sendiri untuk token.

Untuk ini, 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 baca tabel dengan terjemahan ini.

Argumen tok2, tok4, dst. adalah terjemahan kolom-kolom tabel. Argumen ini tidak ada dalam daftar parameter readtable(), jadi Anda perlu menyediakannya dengan “:=”.

```
MT,hd=readtable(“table.dat”,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

load over statistics;

Untuk mencetak, kita perlu menentukan set token yang sama. Kita cetak empat baris pertama saja.

```
writetable(MT[1:10],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

Person Sex Age Titanic Evaluation Tip Problem 1 m 30 n . 1.8 n 2 f 23 y g 1.8 n 3 f 26 y g 1.8 y 4 m 33 n . 2.8 n 5 m 37 n . 1.8 n 6 m 28 y g 2.8 y 7 f 31 y vg 2.8 n 8 m 23 n . 0.8 n 9 f 24 y vg 1.8 y 10 m 26 n . 1.8 n

Titik ”.” mewakili nilai yang tidak tersedia.

Jika kita tidak ingin menentukan token untuk penerjemahan terlebih dahulu, kita hanya perlu menentukan kolom mana yang berisi token dan bukan angka.

```
ctok=[2,4,5,7]; MT,hd,tok=readtable("table.dat",ctok=ctok);
```

Fungsi readtable() sekarang mengembalikan serangkaian token.

```
tok
```

```
m n f y g vg
```

Tabel berisi entri dari berkas dengan token yang diterjemahkan ke angka.

String khusus NA="." ditafsirkan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) dalam tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAval.

```
MT[1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Berikut ini adalah isi tabel dengan angka yang belum diterjemahkan.

```
writetable(MT,wc=5)
```

```
1 1 30 2 . 1.8 2 2 3 23 4 5 1.8 2 3 3 26 4 5 1.8 4 4 1 33 2 . 2.8 2 5 1 37 2 .
1.8 2 6 1 28 4 5 2.8 4 7 3 31 4 6 2.8 2 8 1 23 2 . 0.8 2 9 3 24 4 6 1.8 4 10 1 26 2
. 1.8 2 11 3 23 4 6 1.8 4 12 1 32 4 5 1.8 2 13 1 29 4 6 1.8 4 14 3 25 4 5 1.8 4 15
3 31 4 5 0.8 2 16 1 26 4 5 2.8 2 17 1 37 2 . 3.8 2 18 1 38 4 5 . 2 19 3 29 2 . 3.8
2 20 3 28 4 6 1.8 2 21 3 28 4 1 2.8 4 22 3 28 4 6 1.8 4 23 3 38 4 5 2.8 2 24 3 27
4 1 1.8 4 25 1 27 2 . 2.8 4
```

Demi kenyamanan, Anda dapat memasukkan output readtable() ke dalam daftar.

```
Table=readtable("table.dat",ctok=ctok);
```

Dengan menggunakan kolom token yang sama dan token yang dibaca dari berkas, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dll. atau menggunakan daftar Tabel.

```
writetable(Table,ctok=ctok,wc=5);
```

```
Person Sex Age Titanic Evaluation Tip Problem 1 m 30 n . 1.8 n 2 f 23 y g
1.8 n 3 f 26 y g 1.8 y 4 m 33 n . 2.8 n 5 m 37 n . 1.8 n 6 m 28 y g 2.8 y 7 f 31
y vg 2.8 n 8 m 23 n . 0.8 n 9 f 24 y vg 1.8 y 10 m 26 n . 1.8 n 11 f 23 y vg 1.8
y 12 m 32 y g 1.8 n 13 m 29 y vg 1.8 y 14 f 25 y g 1.8 y 15 f 31 y g 0.8 n 16 m
26 y g 2.8 n 17 m 37 n . 3.8 n 18 m 38 y g . n 19 f 29 n . 3.8 n 20 f 28 y vg 1.8
n 21 f 28 y m 2.8 y 22 f 28 y vg 1.8 y 23 f 38 y g 2.8 n 24 f 27 y m 1.8 y 25 m
27 n . 2.8 y
```

Fungsi tablecol() mengembalikan nilai kolom tabel, melewati baris mana pun dengan nilai NAN("." dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

```
c,i=tablecol(MT,[5,6]);
```

Kita dapat menggunakan ini untuk mengekstrak kolom dari tabel untuk tabel baru.

```
j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok)
```

```
Person Evaluation Tip 2 g 1.8 3 g 1.8 6 g 2.8 7 vg 2.8 9 vg 1.8 11 vg 1.8 12
g 1.8 13 vg 1.8 14 g 1.8 15 g 0.8 16 g 2.8 20 vg 1.8 21 m 2.8 22 vg 1.8 23 g 2.8
24 m 1.8
```

Tentu saja, kita perlu mengekstrak tabel itu sendiri dari daftar Table dalam kasus ini.

```
MT=Table[1];
```

Tentu saja, kita juga dapat menggunakan untuk menentukan nilai rata-rata kolom atau nilai statistik lainnya.

```
mean(tablecol(MT,6))
2.175
```

Fungsi getstatistics() mengembalikan elemen dalam vektor dan jumlahnya. Kita menerapkannya pada nilai "m" dan "f" di kolom kedua tabel kita.

```
xu,count=getstatistics(tablecol(MT,2)); xu, count,
[1, 3] [12, 13]
```

Kita dapat mencetak hasilnya di tabel baru.

```
writetable(count',labr=tok[xu])
m 12 f 13
```

Fungsi selecttable() mengembalikan tabel baru dengan nilai-nilai dalam satu kolom yang dipilih dari vektor indeks. Pertama-tama kita mencari indeks dari dua nilai kita di tabel token.

```
v:=indexof(tok,[“g”, “vg”])
[5, 6]
```

Sekarang kita dapat memilih baris tabel, yang memiliki salah satu nilai dalam v di baris ke-5.

```
MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5);
```

Sekarang kita dapat mencetak tabel, dengan nilai yang diekstraksi dan diurutkan di kolom ke-5.

```
writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7);
```

Person Sex Age Titanic Evaluation Tip Problem 2 f 23 y g 1.8 n 3 f 26 y g  
1.8 y 6 m 28 y g 2.8 y 18 m 38 y g . n 16 m 26 y g 2.8 n 15 f 31 y g 0.8 n 12 m  
32 y g 1.8 n 23 f 38 y g 2.8 n 14 f 25 y g 1.8 y 9 f 24 y vg 1.8 y 7 f 31 y vg 2.8  
n 20 f 28 y vg 1.8 n 22 f 28 y vg 1.8 y 13 m 29 y vg 1.8 y 11 f 23 y vg 1.8 y

Untuk statistik 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)
m n m n m n m n m n m n m y m y m y m y f n f y f y f y f y
f y f y f y f y f y f y
```

Dengan getstatistics(), kita juga dapat menghubungkan jumlah pada dua kolom tabel satu sama lain.

```
MT24=tablecol(MT,[2,4]); ... xu1,xu2,count=getstatistics(MT24[1],MT24[2]);
... writetable(count,labr=tok[xu1],labc=tok[xu2])
n y m 7 5 f 1 12
```

Suatu tabel dapat ditulis ke dalam suatu berkas.

```
filename=“test.dat”; ... writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
```

Lalu kita dapat membaca tabel dari berkas tersebut.

```
MT2,hd,tok2,hdr=readtable(filename, clabs, rlabs); ... writetable(MT2,labr=hdr,labc=hd)
n y m 7 5 f 1 12
```

Dan hapus berkasnya.

```
fileremove(filename);
```

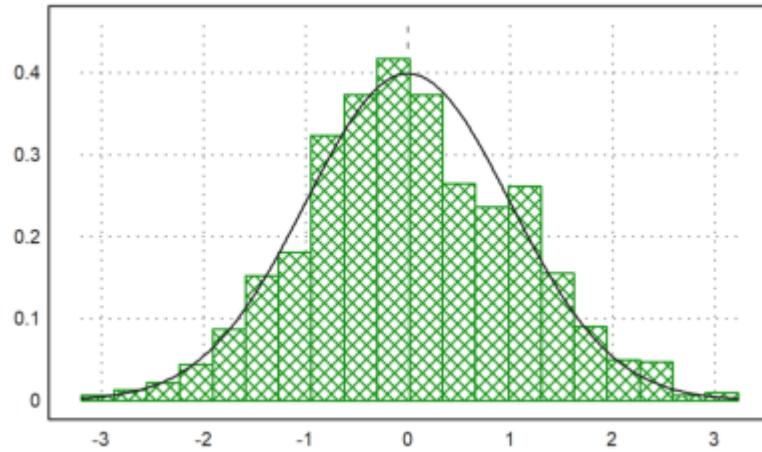
Distribusi

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="
/");
// plot the random sample p
plot2d("qnormal(x,0,1)",add=1); // add the standard normal distribution
plot

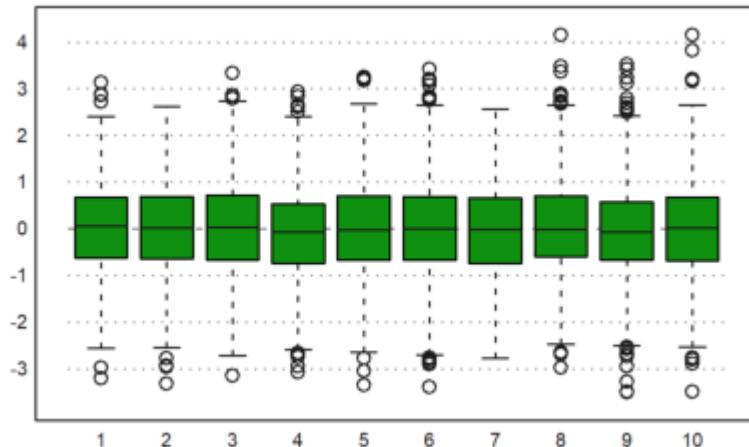
```



Harap perhatikan perbedaan antara diagram batang (sampel) dan kurva normal (distribusi riil). Masukkan kembali ketiga perintah tersebut untuk melihat hasil sampel lainnya.

Berikut ini adalah perbandingan 10 simulasi dari 1000 nilai yang didistribusikan secara normal menggunakan apa yang disebut diagram kotak. Diagram ini menunjukkan median, kuartil 25minimal dan maksimal, dan outlier.

```
p=normal(10,1000); boxplot(p);
```

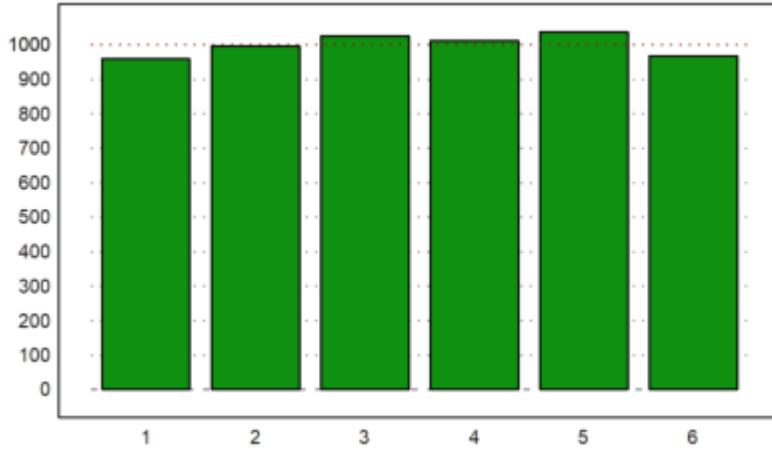


Untuk menghasilkan bilangan bulat acak, Euler memiliki inrandom. Mari

kita simulasiakan lemparan dadu dan plot distribusinya.

Kita menggunakan fungsi getmultiplicities(v,x), yang menghitung seberapa sering elemen v muncul di x. Kemudian kita plot hasilnya menggunakan column-splot().

```
k=intrandom(1,6000,6); ... columnsplot(getmultiplicities(1:6,k)); ...
ygrid(1000,color=red);
```



Sementara intrandom(n,m,k) mengembalikan bilangan bulat yang terdistribusi seragam dari 1 hingga k, dimungkinkan untuk menggunakan distribusi bilangan bulat lain yang diberikan dengan randpint().

Dalam contoh berikut, probabilitas untuk 1,2,3 masing-masing adalah 0,4,0,1,0,5.  
`randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,`  
`[394, 107, 499]`

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Lihat referensinya.

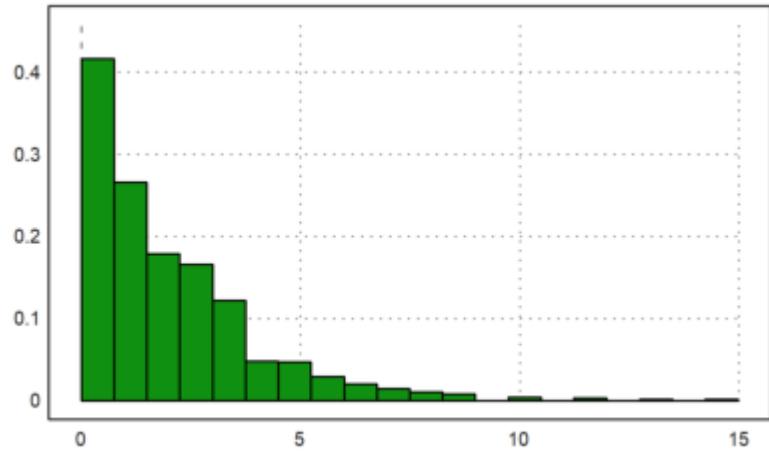
Misalnya, kita coba distribusi eksponensial. Variabel acak kontinu X dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

dengan parameter

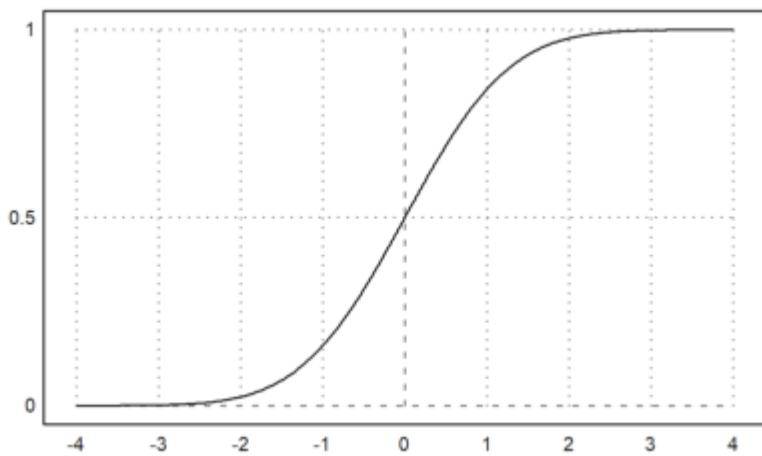
$$\lambda = \frac{1}{\mu}, \quad \mu \text{ adalah mean, dan dilambangkan dengan } X \sim \text{Eksponensial}(\lambda).$$

```
plot2d(randexponential(1,1000,2), distribution);
```



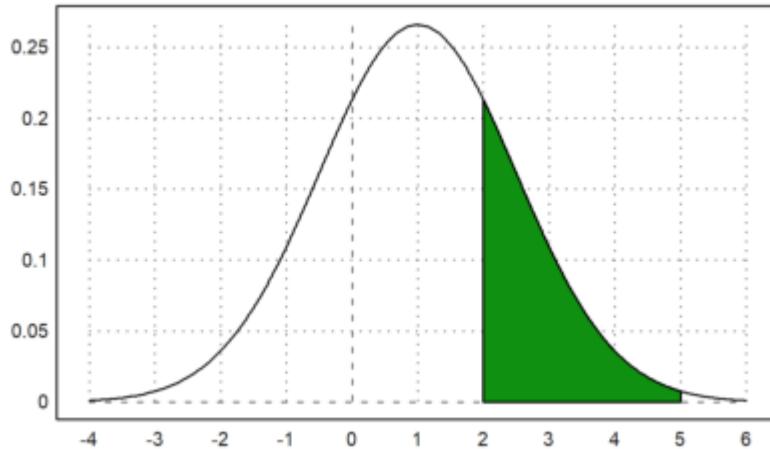
Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversnya.

plot2d("normaldis",-4,4);



Berikut ini adalah salah satu cara untuk memplot kuantil.

plot2d("qnormal(x,1,1.5)",-4,6); ... plot2d("qnormal(x,1,1.5)",a=2,b=5, add, filled);



lateks:  $\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-m}{d})^2} dt.$

Peluang 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.

$$\int_2^5 \frac{1}{1.5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-1.5}{1.5})^2} dx.$$

`gauss("qnormal(x,1,1.5)",2,5)`

0.248662156979

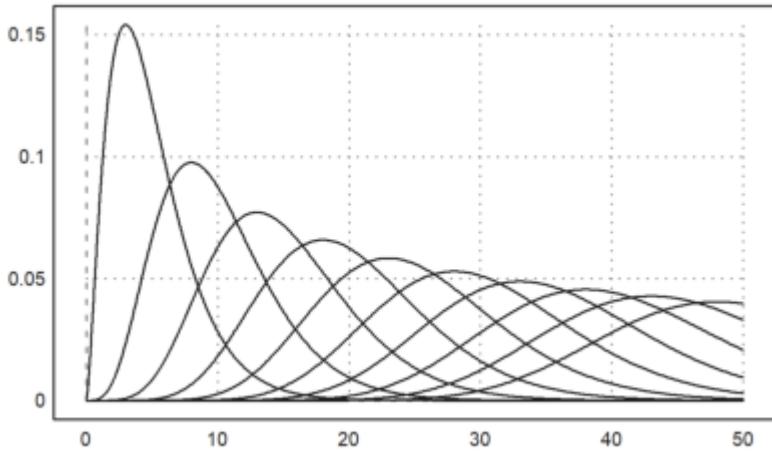
Mari kita bandingkan distribusi binomial dengan distribusi normal dengan nilai rata-rata dan deviasi yang sama. Fungsi `invbindis()` menyelesaikan interpolasi linier antara nilai integer.

`invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5sqrt(1000))`

525.516721219 526.007419394

Fungsi `qdis()` adalah kerapatan distribusi chi-kuadrat. Seperti biasa, Euler memetakan vektor ke fungsi ini. Jadi, kita memperoleh plot semua distribusi chi-kuadrat dengan derajat 5 hingga 30 dengan mudah dengan cara berikut.

`plot2d("qchidis(x,(5:5:50))",0,50):`



Euler memiliki fungsi yang akurat untuk mengevaluasi distribusi. Mari kita periksa chidis() dengan integral.

Penamaannya mencoba agar konsisten. Misalnya,

- \* distribusi chi-kuadrat adalah chidis(),
- \* fungsi inversnya adalah invchidis(),
- \* densitasnya adalah qchidis().

Komplemen 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 distribusi.

```
wd = 0—((1:6)+[-0.01,0.01,0,0,0,0])/6
[0, 0.165, 0.335, 0.5, 0.666667, 0.833333, 1]
```

Artinya adalah bahwa dengan probabilitas  $wd[i+1]-wd[i]$  kita menghasilkan nilai acak i.

Ini hampir merupakan distribusi seragam. Mari kita definisikan generator angka acak untuk ini. Fungsi find(v,x) menemukan nilai x dalam vektor v. Fungsi ini juga berfungsi untuk vektor x.

```
function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahannya begitu halus sehingga kita hanya melihatnya pada pengulangan yang sangat banyak.

```
columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```

```
![images/23030630044_Aliifia
```

Berikut ini adalah fungsi sederhana untuk memeriksa distribusi seragam nilai 1...K dalam v. Kita terima hasilnya, jika untuk semua frekuensi

$$\left| f_i - \frac{1}{K} \right| < \frac{\delta}{\sqrt{n}}.$$

```

function checkrandom (v, delta=1) ...
K=max(v); n=cols(v); fr=getfrequencies(v,1:K); return max(fr/n-1/K)|delta/sqrt(n);
endfunction i/pre; Memang fungsi tersebut menolak distribusi seragam.
checkrandom(wrongdice(1,1000000))
0
Dan menerima generator acak bawaan.
checkrandom(intrandom(1,1000000,6))
1
Kita dapat menghitung distribusi binomial. Pertama ada binomials() ,
yang mengembalikan probabilitas i atau kurang dari n kali percobaan.
bindis(410,1000,0.4)
0.751401349654
Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-
Pearson untuk parameter p. Level default adalah alpha.
Arti dari interval ini adalah jika p berada di luar interval, hasil yang diamati
sebesar 410 dalam 1000 adalah langka.
clopperpearson(410,1000)
[0.37932, 0.441212]
Perintah berikut adalah cara langsung untuk mendapatkan hasil di atas.
Namun untuk n yang besar, penjumlahan langsung tidak akurat dan lambat.
p=0.4; i=0:410; n=1000; sum(bin(n,i)pi(1 - p)(n - i))
0.751401349655
Omong-omong, invbinsum() menghitung kebalikan dari binomials().
invbindis(0.75,1000,0.4)
409.932733047
Dalam Bridge, kita mengasumsikan 5 kartu yang beredar (dari 52) dalam
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).
2hypergeomsum(1,5,13,26)
0.321739130435
Ada juga simulasi distribusi multinomial.
randmultinomial(10,1000,[0.4,0.1,0.5])
394 82 524 418 92 490 445 90 465 403 114 483 405 95 500 384 107 509 414
93 493 419 90 491 394 101 505 382 103 515
Merencanakan Data
Untuk merencanakan data, kami mencoba hasil pemilu Jerman sejak 1990,
yang diukur dalam jumlah kursi.
BW := [... 1990,662,319,239,79,8,17; ... 1994,672,294,252,47,49,30; ...
1998,669,245,298,43,47,36; ... 2002,603,248,251,47,55,2; ... 2005,614,226,222,61,51,54;
... 2009,622,239,146,93,68,76; ... 2013,631,311,193,0,63,64];
Untuk para pihak, kami menggunakan serangkaian nama.
P:=["CDU/CSU","SPD","FDP","Gr","Li"];
Mari kita cetak persentasenya dengan baik.
Pertama-tama kita ekstrak kolom-kolom yang diperlukan. Kolom 3 hingga
7 adalah kursi masing-masing partai, dan kolom 2 adalah jumlah total kursi.
Kolom 3 adalah tahun pemilihan.

```

```
BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Kemudian kami mencetak statistik dalam bentuk tabel. Kami menggunakan nama sebagai tajuk kolom, dan tahun sebagai tajuk untuk baris. Lebar default untuk kolom adalah `wc=10`, tetapi kami lebih suka keluaran yang lebih padat. Kolom akan diperluas untuk label kolom, jika perlu.

```
writetable(BT100,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
```

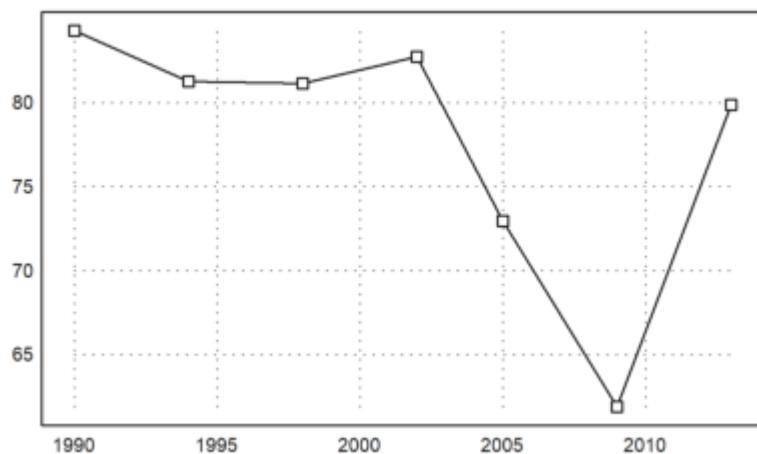
Perkalian matriks berikut ini mengekstrak jumlah persentase dari dua partai besar yang menunjukkan bahwa partai-partai kecil telah memperoleh dukungan di parlemen hingga tahun 2009.

```
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 menggunakan untuk menampilkan garis dan titik secara bersamaan. Alternatifnya adalah memanggil `plot2d` dua kali dengan `gt;add`.

```
statplot(YT,BT1,"b"):
```



Tentukan beberapa warna untuk setiap pihak.

```
CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

Sekarang kita dapat memetakan hasil pemilu 2009 dan perubahannya ke dalam satu plot menggunakan gambar. Kita dapat menambahkan vektor kolom ke setiap plot.

```
figure(2,1); ... figure(1); columnsplot(BW[6,3:7],P,color=CP); ... figure(2); columnsplot(BW[6,3:7]-BW[5,3:7],P,color=CP); ... figure(0):
```

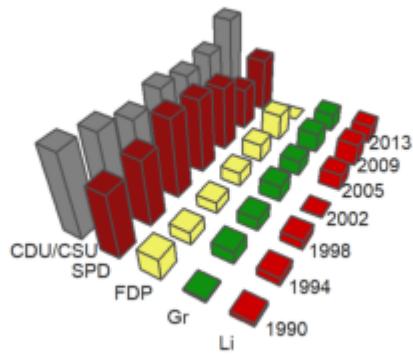
```
![images/23030630044_Alifia
```

Plot data menggabungkan baris-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/23030630044_Alifia
```

Plot kolom 3D menunjukkan baris data statistik dalam bentuk kolom. Kami memberikan label untuk baris dan kolom. Angle adalah sudut pandang.  
`columnsplot3d(BT,scols=P,srows=YT, ... angle=30°,ccols=CP):`



Representasi lainnya adalah plot mosaik. Perhatikan bahwa kolom-kolom plot mewakili kolom-kolom matriks di sini. Karena panjang label CDU/CSU, kami mengambil jendela yang lebih kecil dari biasanya.

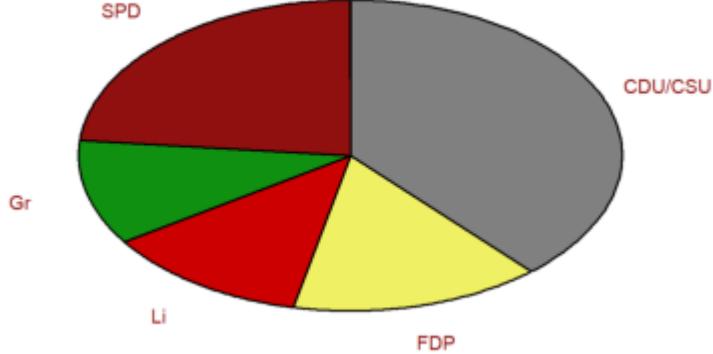
```
shrinkwindow(smaller); ... mosaicplot(BT',srows=YT,scols=P,color=CP,style="");

... shrinkwindow():

![images/23030630044_Alfia]
```

Kita juga bisa membuat diagram lingkaran. Karena hitam dan kuning membentuk koalisi, kita susun ulang unsur-unsurnya.

```
i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```



Berikut adalah jenis plot yang lain.

```
starplot(normal(1,10)+4,lab=1:10, rays):
```

```
![images/23030630044_Alfia
```

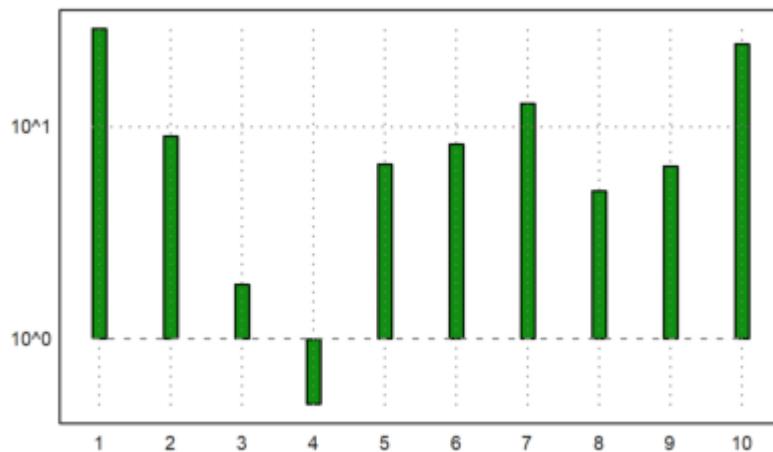
Beberapa plot dalam plot2d bagus untuk statika. Berikut adalah plot impuls data acak, yang didistribusikan secara seragam dalam [0,1].

```
plot2d(makeimpulse(1:10,random(1,10)), bar):
```

```
![images/23030630044_Alfia
```

Namun untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

```
logimpulseplot(1:10,-log(random(1,10))10):
```



Fungsi columnsplot() lebih mudah digunakan, karena hanya memerlukan vektor nilai. Selain itu, fungsi ini dapat mengatur labelnya sesuai keinginan kita, kami telah menunjukkannya dalam tutorial ini.

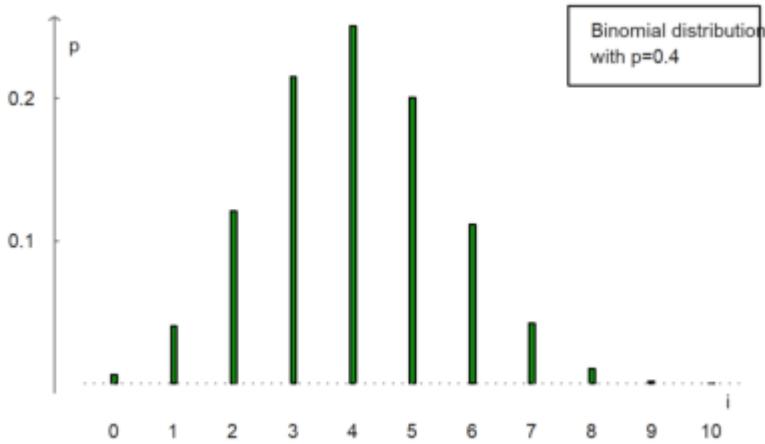
Berikut adalah aplikasi lain, tempat kita menghitung karakter dalam kalimat dan memplot statistik.

```
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/23030630044_Alfia
```

Anda juga dapat mengatur sumbu secara manual.

```
n=10; p=0.4; i=0:n; x=bin(n,i)p^i(1-p)^(n-i); ... columnsplot(x, lab =
i, width = 0.05, < frame, < grid); ...yaxis(0, 0 : 0.1 : 1, style = "-"; left); xaxis(0, style =
"."); ...label("p", 0, 0.25), label("i", 11, 0); ...textbox(["Binomial distribution", "with p =
0.4"]);
```



Berikut ini adalah cara untuk memetakan frekuensi angka dalam sebuah vektor.

Kita buat sebuah vektor bilangan acak integer 1 hingga 6.

```
v:=intrandom(1,10,10)
[5, 1, 8, 6, 6, 10, 9, 6, 8, 3]
```

Lalu ekstrak angka-angka unik dalam v.

```


```

Dan plot frekuensi pada kolom plot.

```
columnsplot(getmultiplicities(vu,v),lab=vu,style="/")
![images/23030630044_Alfia
```

Kami ingin menunjukkan fungsi untuk distribusi nilai empiris.

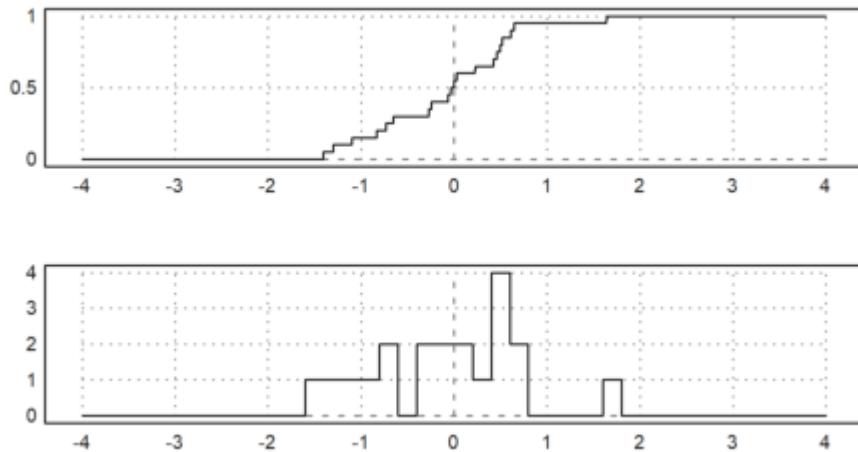
```
x=normal(1,20);
```

Fungsi empdist(x,vs) memerlukan array nilai yang diurutkan. Jadi, kita harus mengurutkan x sebelum dapat menggunakaninya.

```
xs=sort(x);
```

Kemudian kami memetakan distribusi empiris dan beberapa batang kepadatan ke dalam satu petak. Alih-alih menggunakan petak batang untuk distribusi, kali ini kami menggunakan petak gigi gergaji.

```
figure(2,1); ... figure(1); plot2d("empdist",-4,4;xs); ... figure(2); plot2d(histo(x,v=-4:0.2:4,1bar)); ... figure(0);
```



Plot sebaran mudah dibuat di Euler dengan plot titik biasa. Grafik berikut menunjukkan bahwa  $X$  dan  $X+Y$  jelas berkorelasi positif.

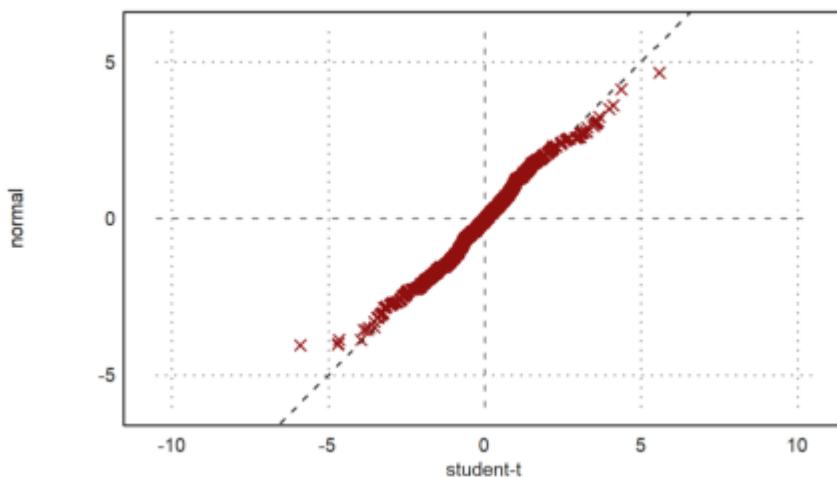
```
x=normal(1,100); plot2d(x,x+rotright(x), points,style=".."):
```

![images/23030630044\_Alfia

Sering kali, kita ingin membandingkan dua sampel dengan distribusi yang berbeda. Hal ini dapat dilakukan dengan plot kuantil-kuantil.

Untuk pengujian, kita mencoba distribusi t-student dan distribusi eksponensial.

```
x=randt(1,1000,5); y=randnormal(1,1000,mean(x),dev(x)); ... plot2d("x",r=6,style="-",yl="normal",xl="student-t", vertical); ... plot2d(sort(x),sort(y), points,color=red,style="x", add):
```



Plot tersebut dengan jelas menunjukkan bahwa nilai-nilai yang terdistribusi normal cenderung lebih kecil di ujung-ujung ekstrem.

Jika kita memiliki dua distribusi dengan ukuran yang berbeda, kita dapat

memperluas yang lebih kecil atau mengecilkan 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/23030630044_Alfia
```

#### Regresi dan Korelasi

Regresi linier dapat dilakukan dengan fungsi polyfit() atau berbagai fungsi fit.

Sebagai permulaan, kita mencari garis regresi untuk data univariat dengan polyfit(x,y,1).

```
x=1:10; y=[2,3,1,5,6,3,7,8,9,8]; writetable(x—y',labc=[“x”, “y”])
x y 1 2 2 3 3 1 4 5 5 6 6 3 7 7 8 8 9 9 10 8
```

Kami ingin membandingkan kecocokan yang tidak tertimbang dan tertimbang. Pertama, koefisien kecocokan linier.

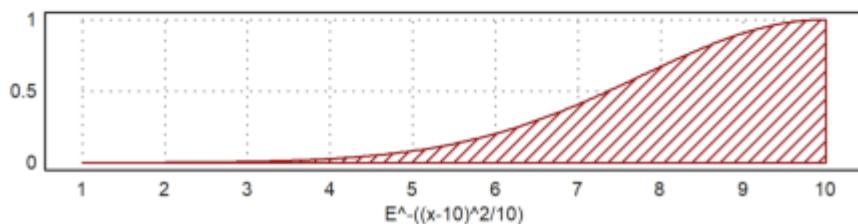
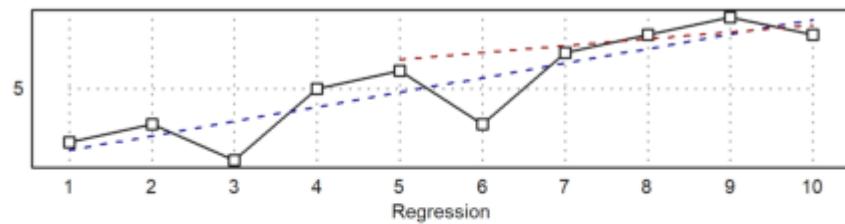
```
p=polyfit(x,y,1)
[0.733333, 0.812121]
```

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 memasukkan semuanya ke dalam satu plot untuk titik dan garis regresi, dan untuk bobot yang digunakan.

```
figure(2,1); ... figure(1); statplot(x,y,"b",xl="Regression"); ... plot2d("evalpoly(x,p)", add,color=blue,style="—");
... plot2d("evalpoly(x,pw)",5,10, add,color=red,style="--"); ... figure(2);
plot2d(w,1,10, filled,style="/",fillcolor=red, xl=w); ... figure(0):
```



Untuk contoh lain, kami membaca survei siswa, usia mereka, usia orang tua mereka, dan jumlah saudara kandung dari sebuah berkas.

Tabel ini berisi "m" dan "f" di kolom kedua. Kami menggunakan variabel tok2 untuk mengatur terjemahan yang tepat alih-alih membiarkan readtable() mengumpulkan terjemahan.

```
MS,hd:=readtable("table1.dat",tok2:=["m" , "f"]); ... writetable(MS,labc=hd,tok2:=["m" , "f"]);
Person Sex Age Mother Father Siblings 1 m 29 58 61 1 2 f 26 53 54 2 3 m
24 49 55 1 4 f 25 56 63 3 5 f 25 49 53 0 6 f 23 55 55 2 7 m 23 48 54 2 8 m 27 56
58 1 9 m 25 57 59 1 10 m 24 50 54 1 11 f 26 61 65 1 12 m 24 50 52 1 13 m 29
54 56 1 14 m 28 48 51 2 15 f 23 52 52 1 16 m 24 45 57 1 17 f 24 59 63 0 18 f 23
52 55 1 19 m 24 54 61 2 20 f 23 54 55 1
```

Bagaimana usia saling bergantung? Kesan pertama datang dari diagram sebaran berpasangan.

```
scatterplots(tablecol(MS,3:5),hd[3:5]);
![images/23030630044_Aliifia]
```

Jelas bahwa usia ayah dan ibu saling bergantung. Mari kita tentukan dan gambarkan garis regresinya.

```
cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1)
[17.3789, 0.740964]
```

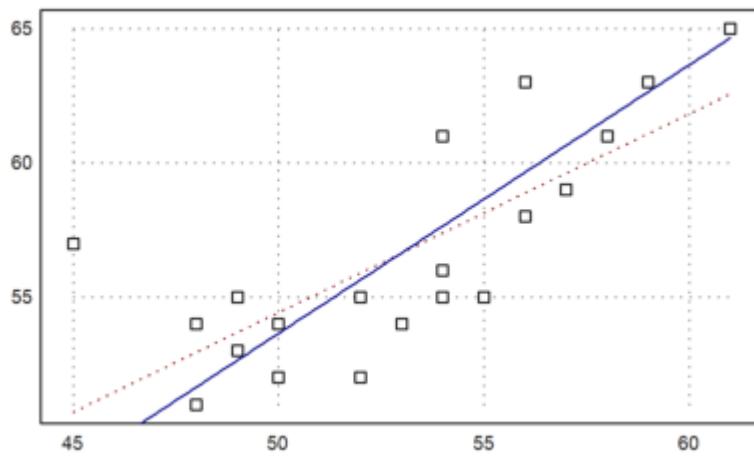
Ini jelas model yang salah. Garis regresi adalah  $s=17+0,74t$ , di mana t adalah usia ibu dan s adalah usia ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak terlalu banyak.

Sebaliknya, kami menduga fungsi seperti  $s=a+t$ . Maka a adalah rata-rata s-t. Itu adalah perbedaan usia rata-rata antara ayah dan ibu.

```
da:=mean(cs[2]-cs[1])
3.65
```

Mari kita gambarkan ini menjadi satu diagram sebar.

```
plot2d(cs[1],cs[2], points); ... plot2d("evalpoly(x,ps)",color=red,style=". ", add);
... plot2d("x+da",color=blue, add);
```



Berikut ini adalah diagram kotak dari dua zaman tersebut. Ini hanya menunjukkan bahwa zamannya berbeda.

```

boxplot(cs,"mothers","fathers"]):
![images/23030630044_Alfia
Menariknya bahwa perbedaan median tidak sebesar perbedaan mean.
median(cs[2])-median(cs[1])
1.5
Koefisien korelasi menunjukkan korelasi positif.
correl(cs[1],cs[2])
0.7588307236
Korelasi peringkat adalah ukuran untuk urutan yang sama di kedua vektor.
Korelasi ini juga cukup positif.
rankcorrel(cs[1],cs[2])
0.758925292358
Membuat Fungsi baru
Tentu saja, bahasa EMT dapat digunakan untuk memprogram fungsi baru.
Misalnya, kita mendefinisikan fungsi kemiringan.

```

$$sk(x) = \sqrt{n} \sum_i (x_i - m)^3 \left( \sum_i (x_i - m)^2 \right)^{3/2}$$

di mana m adalah rata-rata x.

```

function skew (x:vector) ...
m=mean(x); return sqrt(cols(x))*sum((x-m)^3)/(sum((x-m)^2))^(3/2); endfunction <
/pre > Seperti yang Andalihat, kita dapat dengan mudah menggunakan bahasa matriks untuk mendapatkan impl
data=normal(20); skew(normal(10))
0.521806329961

```

Berikut adalah fungsi lainnya, yang disebut koefisien kemiringan Pearson.

```

function skew1 (x) := 3(mean(x)-median(x))/dev(x)
skew1(data)
0.573102925949

```

Simulasi Monte Carlo

Euler dapat digunakan untuk mensimulasikan kejadian acak. Kita telah melihat contoh sederhana di atas. Berikut ini contoh lain, yang mensimulasikan 1000 kali lemparan 3 dadu, dan menanyakan distribusi jumlahnya.

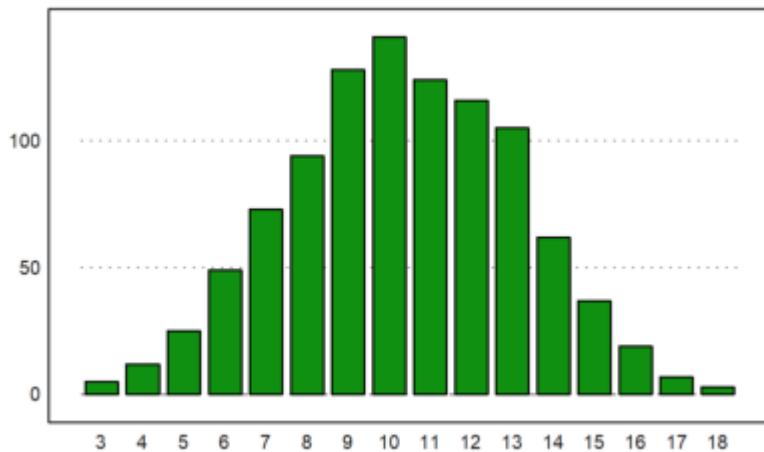
```

ds:=sum(intrandom(1000,3,6)); fs=getmultiplicities(3:18,ds)
[5, 12, 25, 49, 73, 94, 128, 141, 124, 116, 105, 62, 37, 19, 7, 3]

```

Kita bisa merencanakannya sekarang.

```
columnsplot(fs,lab=3:18):
```

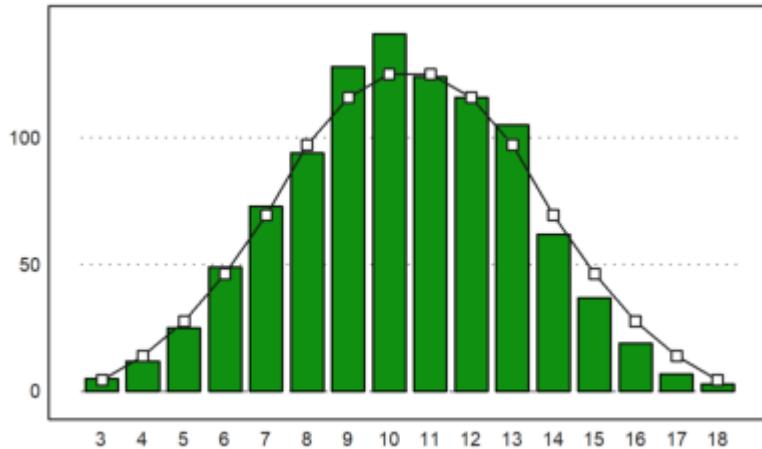


Menentukan distribusi yang diharapkan tidaklah mudah. ??Kami menggunakan rekursi tingkat lanjut untuk ini.

Fungsi berikut menghitung jumlah cara bilangan k dapat direpresentasikan sebagai jumlah n bilangan 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 kj=1 kj=m else sum=0; loop 1 to m; sum=sum+countways(k-
,n-1,m); end; return sum; end; endfunction i/prej Berikut ini hasil dari tiga kali
lemparan dadu.
 countways(5:25,5,5)
 [1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320, 255, 185, 121, 70, 35,
15, 5, 1]
 cw=countways(3:18,3,6)
 [1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3, 1]
 Kami menambahkan nilai yang diharapkan ke plot.
 plot2d(cw/631000, add); plot2d(cw/631000, points, add) :
```



Untuk simulasi lain, 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 ini dengan simulasi. Kita hasilkan 10000 kali 10 vektor acak.

```
M=normal(10000,10); dev(mean(M)')
0.318861419326
plot2d(mean(M)', distribution):
![images/23030630044_Alfia]
```

Median dari 10 bilangan acak berdistribusi normal 0-1 memiliki deviasi yang lebih besar.

```
dev(median(M)')
0.376651504162
```

Karena kita dapat dengan mudah menghasilkan lintasan acak, kita dapat mensimulasikan proses Wiener. Kita mengambil 1000 langkah dari 1000 proses. Kemudian kita memetakan deviasi standar dan rata-rata langkah ke-n dari 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/23030630044_Alfia]
```

Pengujian

Pengujian merupakan alat penting dalam statistik. Dalam Euler, banyak pengujian yang diterapkan. Semua pengujian ini menghasilkan galat yang kita terima jika kita menolak hipotesis nol.

Sebagai contoh, kita menguji lemparan dadu untuk distribusi seragam. Pada 600 lemparan, kita memperoleh 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 modus, yang menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya harus hampir sama. Parameter  $gt;p$  menginterpretasikan vektor  $y$  sebagai vektor probabilitas.

```
chitest([90,103,114,101,103,89],dup(1/6,6)', p, montecarlo)
0.517
```

Kesalahan ini terlalu besar. Jadi kita tidak dapat menolak distribusi seragam. Ini tidak membuktikan bahwa dadu kita adil. Namun, kita tidak dapat menolak hipotesis kita.

Selanjutnya, kita menghasilkan 1000 lemparan dadu menggunakan generator angka acak, dan melakukan pengujian yang sama.

```
n=1000; t=random([1,n6]); chitest(count(t6,6),dup(n,6)')
0.159910723681
```

Mari kita uji nilai rata-rata 100 dengan uji-t.

```
s=200+normal([1,100])10; ... ttest(mean(s),dev(s),100,200)
0.353414299977
```

Fungsi `ttest()` memerlukan nilai rata-rata, deviasi, jumlah data, dan nilai rata-rata yang akan diuji.

Sekarang mari kita periksa dua pengukuran untuk nilai rata-rata yang sama. Kita tolak hipotesis bahwa keduanya memiliki nilai rata-rata yang sama, jika hasilnya  $lt;0.05$ .

```
tcomparedata(normal(1,10),normal(1,10))
0.268166832384
```

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)
2.17064628392e-05
```

Pada contoh berikutnya, kita buat 20 lemparan dadu acak sebanyak 100 kali dan hitung angka-angka yang ada di dalamnya. Rata-rata harus ada  $20/6=3,3$  angka.

```
R=random(100,20); R=sum(R6j=1)'; mean(R)
3.05
```

Sekarang kita bandingkan jumlah angka satu dengan distribusi binomial. Pertama kita gambarkan distribusi angka satu.

```
plot2d(R,distribution=max(R)+1,even=1,style="")
/":
```

```
![images/23030630044_Alfia
t=count(R,21);
```

Lalu kami hitung nilai yang diharapkan.

```
n=0:20; b=bin(20,n)(1/6)n(5/6)(20-n)100;
```

Kita harus mengumpulkan beberapa angka untuk mendapatkan kategori yang cukup besar.

```
t1=sum(t[1:2])-t[3:7]-sum(t[8:21]); ... b1=sum(b[1:2])-b[3:7]-sum(b[8:21]);
```

Uji chi-kuadrat menolak hipotesis bahwa distribusi kami adalah distribusi binomial, jika hasilnya  $lt; 0.05$ .

```
chitest(t1,b1)
```

```
0.262911182138
```

Contoh berikut berisi hasil dari dua kelompok orang (misalnya pria dan wanita) yang memilih satu dari enam partai.

```
A=[23,37,43,52,64,74;27,39,41,49,63,76]; ... writetable(A,wc=6,labr=[”m”,”f”],labc=1:6)
```

```
1 2 3 4 5 6 m 23 37 43 52 64 74 f 27 39 41 49 63 76
```

Kami ingin menguji independensi suara dari jenis kelamin. Uji tabel chi<sup>2</sup> melakukan hal ini. Hasilnya terlalu besar.

```
tabletest(A)
```

```
0.990701632326
```

Berikut ini adalah tabel yang diharapkan, jika kita mengasumsikan frekuensi pemungutan suara yang diamati.

```
writetable(expectedtable(A),wc=6,dc=1,labr=[”m”,”f”],labc=1:6)
```

```
1 2 3 4 5 6 m 24.9 37.9 41.9 50.3 63.3 74.7 f 25.1 38.1 42.1 50.7 63.7 75.3
```

Kita dapat menghitung koefisien kontingensi yang dikoreksi. Karena sangat mendekati 0, kita simpulkan bahwa pemungutan suara tidak bergantung pada jenis kelamin.

```
contingency(A)
```

```
0.0427225484717
```

Beberapa Pengujian Lainnya

Selanjutnya, kami menggunakan analisis varians (uji F) untuk menguji tiga sampel data berdistribusi normal untuk 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
```

Artinya, kita menolak hipotesis nilai rata-rata yang sama. Kita 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 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 kesetaraan lainnya adalah uji peringkat. Uji peringkat jauh lebih tajam daripada uji median.

```
ranktest(a,b)
```

```
0.00199969612469
```

Dalam contoh berikut, kedua distribusi memiliki rata-rata yang sama.

```
ranktest(random(1,100),random(1,50)3-1)
```

```
0.119780211001
```

Sekarang, mari kita coba simulasikan 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 dari b.

```
signtest(a,b)
```

```
0.0546875
```

Ini adalah kesalahan yang sangat besar. Kita tidak dapat menolak bahwa a sama baiknya dengan b.

Uji Wilcoxon lebih tajam daripada uji ini, tetapi bergantung pada nilai kuantitatif perbedaannya.

```
wilcoxon(a,b)
```

```
0.0296680599405
```

Mari kita coba dua pengujian lagi menggunakan seri yang dihasilkan.

```
wilcoxon(normal(1,20),normal(1,20)-1)
```

```
0.0309763942686
```

```
wilcoxon(normal(1,20),normal(1,20))
```

```
0.588619799108
```

Angka Acak

Berikut ini adalah pengujian untuk generator angka acak. Euler menggunakan generator yang sangat bagus, jadi kita tidak perlu mengharapkan masalah apa pun.

Pertama-tama kita menghasilkan sepuluh juta angka acak dalam [0,1].

```
n:=10000000; r:=random(1,n);
```

Berikutnya kita hitung jarak antara dua angka kurang dari 0,05.

```
a:=0.05; d:=differences(nonzeros(r));
```

Terakhir, kami memplot berapa kali setiap jarak terjadi, dan membandingkannya dengan nilai yang diharapkan.

```
m=getmultiplicities(1:100,d); plot2d(m); ... plot2d("n(1-a)^(x-1)a^2",color = red, add) :
```

```
![images/23030630044_Alfia
```

Hapus data.

```
remvalue n;
```

Pendahuluan bagi Pengguna Proyek R

Jelas, EMT tidak bersaing dengan R sebagai paket statistik. Akan tetapi, ada banyak prosedur dan fungsi statistik yang tersedia di EMT juga. Jadi, EMT dapat memenuhi kebutuhan dasar. Lagi pula, EMT dilengkapi dengan paket numerik dan sistem aljabar komputer.

Buku catatan ini ditujukan bagi Anda yang sudah familier dengan R, tetapi perlu mengetahui perbedaan sintaksis EMT dan R. Kami mencoba memberikan gambaran umum tentang hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami melihat cara untuk bertukar data antara kedua sistem.

Harap dicatat bahwa ini adalah pekerjaan yang masih dalam tahap pengembangan.

Sintaksis Dasar

Hal pertama yang Anda pelajari di R adalah membuat vektor. Dalam EMT, perbedaan utamanya adalah operator : dapat mengambil ukuran langkah. Selain itu, operator ini memiliki daya pengikatan 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. Dimungkinkan untuk menggunakan vektor guna menggabungkan berbagai hal.

Contoh berikut ini, seperti banyak contoh lainnya, berasal dari "Introduction to R" yang disertakan dalam 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() di 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 ditulis 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 "-gt;" digunakan untuk unit dalam EMT.

```
125km - " miles"
```

```
77.6713990297 miles
```

Operator "lt;-" 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, "alt;-4lt;3" berfungsi, tetapi "alt;-4lt;-3" tidak. Saya juga mengalami ambiguitas serupa dalam EMT, tetapi mencoba menghilangkannya sedikit demi sedikit.

EMT dan R memiliki vektor bertipe boolean. Namun dalam EMT, angka 0 dan 1 digunakan untuk mewakili false dan true. Dalam R, nilai true dan false tetap dapat digunakan dalam aritmatika biasa seperti dalam EMT.

```
x|5,
```

```
[0, 0, 1, 0, 0] [0, 0, 3.1, 0, 0]
```

EMT memunculkan kesalahan atau menghasilkan NAN, tergantung pada tanda "kesalahan".

```
errors off; 0/0, isNaN(sqrt(-1)), errors on;
```

```
NAN 1
```

String sama di R dan EMT. Keduanya berada di lokal saat ini, bukan di Unicode.

Di R ada paket untuk Unicode. Di EMT, string dapat berupa string Unicode. String unicode dapat diterjemahkan ke pengodean lokal dan sebaliknya. Selain itu, u"..." dapat berisi entitas HTML.

```
u"169; Reneacut; Grothmann"
```

© René Grothmann

Berikut ini mungkin atau mungkin tidak ditampilkan dengan benar pada sistem Anda sebagai A dengan titik dan garis di atasnya. Hal ini bergantung pada font yang Anda gunakan.

chartoutf([480])

Penggabungan string dilakukan dengan "+" atau "—". String dapat menyertakan angka, yang akan dicetak dalam format saat ini.

"pi = "+pi

pi = 3.14159265359

Pengindeksan

Sering kali, ini akan berfungsi seperti di R.

Namun EMT akan menginterpretasikan indeks negatif dari belakang vektor, sementara R menginterpretasikan x[n] sebagai x tanpa elemen ke-n.

x, x[1:3], x[-2]

[-55, -45, -35, -25, -15, -5, 5, 15, 25, 35, 45, 55] [-55, -45, -35] 45

Perilaku R dapat dicapai dalam EMT dengan drop().

drop(x,2)

[-55, -35, -25, -15, -5, 5, 15, 25, 35, 45, 55]

Vektor logika tidak diperlakukan secara berbeda sebagai indeks dalam EMT, berbeda dengan R. Anda perlu mengekstrak elemen bukan nol terlebih dahulu dalam EMT.

x, x 5, x[nonzeros(x 5)]

[-55, -45, -35, -25, -15, -5, 5, 15, 25, 35, 45, 55] [0, 0, 0, 0, 0, 0, 1, 1, 1,

1] [15, 25, 35, 45, 55]

Sama seperti di R, vektor indeks dapat berisi pengulangan.

x[[1,2,2,1]]

[-55, -45, -45, -55]

Namun, nama untuk indeks tidak dimungkinkan dalam EMT. Untuk paket statistik, hal ini mungkin sering diperlukan untuk memudahkan akses ke elemen vektor.

Untuk meniru perilaku ini, kita dapat mendefinisikan 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)]; ... ...

Trying to overwrite protected function sel! Error in: function sel (v,i,s) :=  
v[indexof(s,i)]; ... ...

Trying to overwrite protected function sel! Error in: function sel (v,i,s) :=  
v[indexof(s,i)]; ... ...

Trying to overwrite protected function sel! Error in: function sel (v,i,s) :=  
v[indexof(s,i)]; ... ... [10.4, 3.1]

Tipe Data

EMT memiliki lebih banyak tipe data tetap daripada R. Jelas, di R terdapat vektor yang terus bertambah. Anda dapat menetapkan vektor numerik kosong

v dan menetapkan nilai ke elemen v[17]. Hal ini tidak mungkin dilakukan di EMT.

Berikut ini agak tidak efisien.

```
v=[]; for i=1 to 10000; v=v—i; end;
```

EMT sekarang akan membuat vektor dengan v dan i yang ditambahkan pada tumpukan dan menyalin vektor itu kembali ke variabel global v.

Yang lebih efisien mendefinisikan vektor terlebih dahulu.

```
v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Yang lebih efisien mendefinisikan vektor terlebih dahulu.

```
complex(1:4)
```

```
[1+0i , 2+0i , 3+0i , 4+0i]
```

Konversi ke string hanya dimungkinkan untuk tipe data dasar. Format saat ini digunakan untuk penggabungan string sederhana. Namun, ada fungsi seperti print() atau frac().

Untuk vektor, Anda dapat dengan mudah menulis fungsi Anda sendiri.

```
function tostr (v) ...
```

```
s=""; loop 1 to length(v); s=s+print(v[],2,0); if !length(v) then s=s+",";
endif; end; return s+""; endfunction
```

```
i/prej_ tostr(linspace(0,1,10))
```

```
[0.00,0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1.00]
```

Untuk komunikasi dengan Maxima, terdapat fungsi convertmxm(), yang juga dapat digunakan untuk memformat vektor untuk keluaran.

```
convertm xm(1:10)
```

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

Untuk Latex perintah tex dapat digunakan untuk mendapatkan perintah Latex.

```
tex([1,2,3])
```

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

Faktor dan Tabel

Dalam pengantar R terdapat 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 program statistik, R memiliki factor() dan tapply() untuk ini.

EMT dapat melakukan ini dengan menemukan indeks wilayah dalam daftar wilayah yang unik.

```
auterr=sort(unique(austates)); f=indexofsorted(auterr,austates)
[6, 5, 4, 2, 2, 3, 8, 8, 4, 7, 2, 7, 4, 4, 5, 6, 5, 3, 8, 7, 4, 2, 2, 8, 5, 1, 2, 7, 7, 1]
```

Pada titik tersebut, kita dapat menulis fungsi loop kita sendiri untuk melakukan sesuatu hanya untuk satu faktor.

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 <
/pre > Agak tidak efisien, karena namanya menghitung wilayah unik untuk setiap i, tetapi berhasil.
```

tappl(auterr,"mean",austates,incomes)

[44.5, 57.3333333333, 55.5, 53.6, 55, 60.5, 56, 52.25]

Perhatikan bahwa ini berfungsi untuk setiap vektor wilayah.

tappl(["act","nsw"],"mean",austates,incomes)

[44.5, 57.3333333333]

Sekarang, paket statistik EMT mendefinisikan tabel seperti di R. Fungsi readtable() dan writetable() dapat digunakan untuk input dan output.

Jadi kita dapat mencetak pendapatan negara rata-rata di wilayah dengan cara yang mudah.

```
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 dalam suatu koleksi dengan jenis dan kategori (negara bagian dan teritori dalam contoh kita). Untuk EMT, kita tambahkan indeks yang telah dihitung sebelumnya.

```
function makef (t) ...
```

Factor data Returns a collection with data t, unique data, indices. See: tap-  
ply u=sort(unique(t)); return t,u,indexofs(t); endfunction </pre>

Sekarang elemen ketiga dari koleksi akan berisi indeks.

statef[3]

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

Sekarang kita dapat meniru tapply() dengan cara berikut. Fungsi ini akan mengembalikan tabel sebagai kumpulan data tabel dan judul kolom.

```
function tapply (t:vector,tf,f: call)...
```

Makes a table of data and factors tf : output of makef() See: makef uf=tf[2];  
f=tf[3]; x=zeros(length(uf)); for i=1 to length(uf); ind=nonzeros(f==i); if length(ind)==0  
then x[i]=NAN; else x[i]=f(ind); endif; end; return x,uf; endfunction </pre>

Kami tidak menambahkan banyak pemeriksaan tipedisini. Satu-satunya tindakan pencegahan menyangkut kata

Tabel ini dapat dicetak sebagai tabel dengan writetable().

```
writetable(tapply(incomes,statef,"mean"),wc=7)
```

act nsw nt qld sa tas vic wa 44.5 57.33 55.5 53.6 55 60.5 56 52.25

Array

EMT hanya memiliki dua dimensi untuk array. Tipe data ini disebut matriks. Akan mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau pustaka C untuk ini.

R memiliki lebih dari dua dimensi. Dalam R, array adalah vektor dengan bidang dimensi.

Dalam EMT, vektor adalah matriks dengan satu baris. Vektor dapat dibuat menjadi matriks dengan redim().

shortformat; X=redim(1:20,4,5)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Ekstraksi baris dan kolom, atau sub-matriks, sangat mirip di R.

X[,2:3]

2 3 7 8 12 13 17 18

Namun, dalam R dimungkinkan untuk menetapkan daftar indeks vektor tertentu ke suatu nilai. Hal yang sama dimungkinkan dalam EMT hanya dengan loop.

```
function setmatrixvalue (M, i, j, v) ...
loop 1 to max(length(i),length(j),length(v)) M[i,j] = v; end; endfunction
i/prez Kami mendemonstrasikan ini untuk menunjukkan bahwa matriks dilewatkan dengan referensi dalam EMT. Jika Anda tidak ingin mengubah matriks asli M, Anda perlu menyalinnya dalam fungsi tersebut.
```

```
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 dalam EMT hanya dapat dilakukan antara vektor. Hal ini dilakukan secara otomatis karena bahasa matriks. Satu vektor harus berupa vektor kolom dan yang lainnya berupa vektor baris.

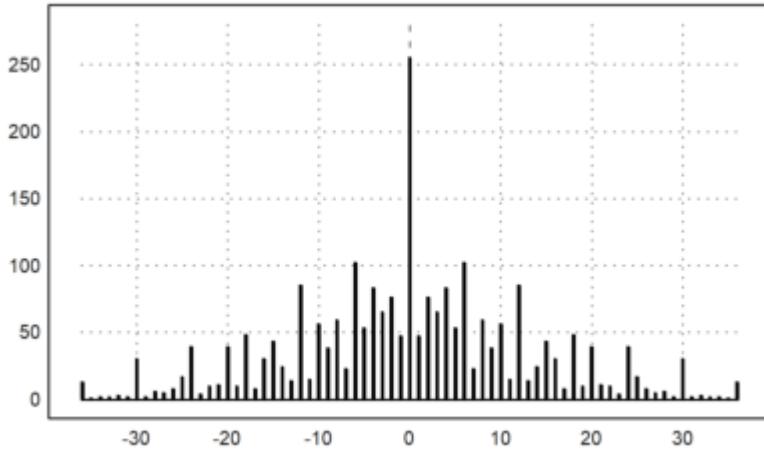
```
(1:5)(1:5)'
1 2 3 4 5 2 4 6 8 10 3 6 9 12 15 4 8 12 16 20 5 10 15 20 25
```

Dalam pengantar PDF untuk R terdapat sebuah contoh, yang menghitung distribusi ab-cd untuk a,b,c,d yang dipilih secara acak dari 0 hingga n. Solusi dalam R adalah membentuk matriks 4 dimensi dan menjalankan table() di atasnya.

Tentu saja, ini dapat dicapai dengan loop. Namun, loop tidak efektif dalam EMT atau R. Dalam EMT, kita dapat menulis loop dalam C dan itu akan menjadi solusi tercepat.

Namun, kita ingin meniru perilaku R. Untuk ini, kita perlu meratakan perkalian ab dan membuat matriks ab-cd.

```
a=0:6; b=a'; p=flatten(ab); q=flatten(p-p'); ... u=sort(unique(q)); f=getmultiplicities(u,q);
... statplot(u,f,"h"):
```

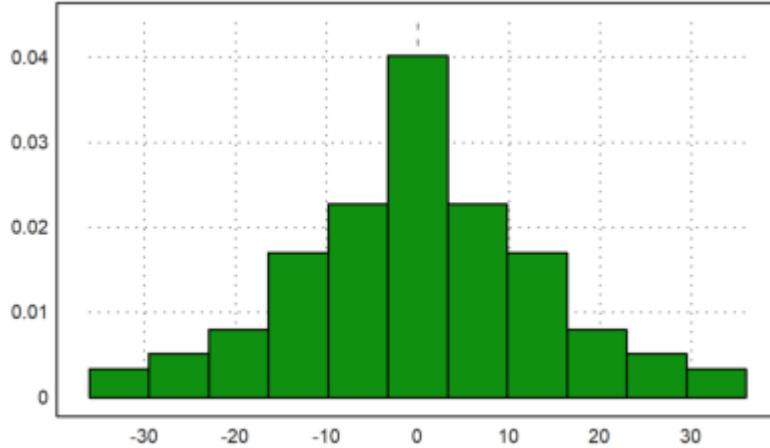


Selain multiplisitas yang tepat, EMT dapat menghitung frekuensi dalam vektor.

```
getfrequencies(q,-50:10:50)
```

[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]

Cara termudah untuk memplot ini sebagai distribusi adalah sebagai berikut.  
plot2d(q,distribution=11):



Namun, Anda juga dapat menghitung terlebih dahulu jumlah dalam interval yang dipilih. Tentu saja, berikut ini menggunakan getfrequencies() secara internal.

Karena fungsi histo() mengembalikan frekuensi, kita perlu menskalakannya sehingga integral di bawah grafik batang adalah 1.

```
x,y=histo(q,v=-55:10:55); y=y/sum(y)/differences(x); ... plot2d(x,y,bar,style="/");
![images/23030630044_Alfia
```

Daftar

EMT memiliki dua jenis daftar. Satu adalah daftar global yang dapat diubah, dan yang lainnya adalah jenis daftar yang tidak dapat diubah. Kami tidak peduli dengan daftar global di sini.

Jenis daftar yang tidak dapat diubah disebut koleksi dalam EMT. Ia berperilaku seperti struktur dalam C, tetapi elemennya hanya diberi nomor dan tidak diberi nama.

```
L="Fred","Flintstone",40,[1990,1992]
```

```
Fred Flintstone 40 [1990, 1992]
```

Saat ini unsur-unsur tersebut tidak memiliki nama, meskipun nama dapat ditetapkan untuk tujuan khusus. Unsur-unsur tersebut diakses dengan angka.

```
(L[4])[2]
```

```
1992
```

Input dan Output File (Membaca dan Menulis Data)

Anda sering kali ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberi tahu Anda tentang berbagai cara untuk mencapainya. Fungsi sederhana adalah writematrix() dan readmatrix().

Mari kita tunjukkan cara membaca dan menulis vektor bilangan real ke dalam file.

```
a=random(1,100); mean(a), dev(a),
0.52013 0.29327
```

Untuk menulis data ke dalam berkas, kami menggunakan fungsi writematrix().

Karena pengantar ini kemungkinan besar berada di dalam direktori, tempat pengguna tidak memiliki akses tulis, kami menulis data ke direktori beranda pengguna. Untuk buku catatan sendiri, ini tidak diperlukan, karena berkas data akan ditulis ke dalam direktori yang sama.

```
filename="test.dat";
```

Sekarang kita tulis vektor kolom a' ke dalam berkas. Ini menghasilkan satu angka di setiap baris berkas.

```
writematrix(a',filename);
```

Untuk membaca data, kita menggunakan readmatrix().

```
a=readmatrix(filename');
```

Dan hapus berkasnya.

```
fileremove(filename);
```

```
mean(a), dev(a),
```

```
0.52013 0.29327
```

Fungsi writematrix() atau writetable() dapat dikonfigurasi untuk bahasa lain.

Misalnya, jika Anda memiliki sistem bahasa Indonesia (titik desimal dengan koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (nilai default dipisahkan dengan koma). File berikut "test.csv" akan muncul di folder Anda saat ini.

```
filename="test.csv"; ... writematrix(random(5,3),file=filename,separator=",");
```

Anda sekarang dapat membuka berkas ini langsung dengan Excel Indonesia.

```
fileremove(filename);
```

Terkadang kita memiliki string dengan token seperti berikut.

```
s1:="f m m f m m m f f f m m f"; ... s2:="f f f m m f f";
```

Untuk menokenisasi ini, kami mendefinisikan vektor token.

```
tok:=[“f”, “m”]
```

```
f m
```

Lalu kita dapat menghitung berapa kali setiap token muncul dalam string, dan memasukkan hasilnya ke dalam tabel.

```
M:=getmultiplicities(tok,strtokens(s1)); ... getmultiplicities(tok,strtokens(s2));
```

Tulis tabel dengan tajuk token.

```
writetable(M,labc=tok,labr=1:2,wc=8)
```

```
f m 1 6 7 2 5 2
```

Untuk statika, EMT dapat membaca dan menulis tabel.

```
file="test.dat"; open(file,"w"); ... writeln("A,B,C"); writematrix(random(3,3));
... close();
```

Berkasnya tampak seperti ini.

```
printfile(file)
```

```
A,B,C 0.7199659096994914,0.9543023869002717,0.5452423982068915 0.8758936037546752,0.31644768821910
0.7031089918521533,0.7963205243867099,0.5570648612224093
```

Fungsi `readtable()` dalam bentuk yang paling sederhana dapat membacanya dan mengembalikan kumpulan nilai dan baris judul.

```
L=readtable(file, list);
```

Koleksi ini dapat dicetak dengan `writetable()` ke buku catatan, atau ke berkas.

```
writetable(L,wc=10,dc=5)
```

```
A B C 0.71997 0.9543 0.54524 0.87589 0.31645 0.42475 0.70311 0.79632
0.55706
```

Matriks nilai adalah elemen pertama L. Perhatikan bahwa `mean()` dalam EMT menghitung nilai rata-rata baris matriks.

```
mean(L[1])
```

```
0.73984 0.53903 0.6855
```

Berkas CSV

Pertama, mari kita tulis matriks ke dalam berkas. Untuk output, kita buat berkas di direktori kerja saat ini.

```
file="test.csv"; ... M=random(3,3); writematrix(M,file);
```

Berikut ini isi berkas tersebut.

```
printfile(file)
```

```
0.396060285468178,0.5217040592547716,0.07945744123024982 0.5777005134637205,0.6537971103437032,0.6
0.8326731141550535,0.9143360255838195,0.6760036570300865
```

CSV ini dapat dibuka pada sistem bahasa Inggris ke Excel dengan mengklik dua kali. Jika Anda mendapatkan berkas tersebut pada sistem bahasa Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal.

Namun, titik desimal juga merupakan format default untuk EMT. Anda dapat membaca matriks dari berkas dengan `readmatrix()`.

```
readmatrix(file)
```

```
0.39606 0.5217 0.079457 0.5777 0.6538 0.69742 0.83267 0.91434 0.676
```

Dimungkinkan untuk menulis beberapa matriks ke dalam satu berkas. Perintah `open()` dapat membuka berkas untuk ditulis dengan parameter "w". Nilai default untuk membaca adalah "r".

```
open(file,"w"); writematrix(M); writematrix(M'); close();
```

Matriks dipisahkan oleh baris kosong. Untuk membaca matriks, buka berkas dan panggil `readmatrix()` beberapa kali.

```
open(file); A=readmatrix(); B=readmatrix(); A==B, close();
```

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

Di Excel atau lembar kerja serupa, Anda dapat mengeksport matriks sebagai CSV (nilai yang dipisahkan koma). Di Excel 2007, gunakan "simpan sebagai" dan "format lain", lalu pilih "CSV". Pastikan, tabel saat ini hanya berisi data yang ingin Anda ekspor.

Berikut ini contohnya.

```
printfile("excel-data.csv")
```

```
0;1000;1000 1;1051,271096;1072,508181 2;1105,170918;1150,273799 3;1161,834243;1233,67806
4;1221,402758;1323,129812 5;1284,025417;1419,067549 6;1349,858808;1521,961556
7;1419,067549;1632,31622 8;1491,824698;1750,6725 9;1568,312185;1877,610579 10;1648,721271;2013,752707
```

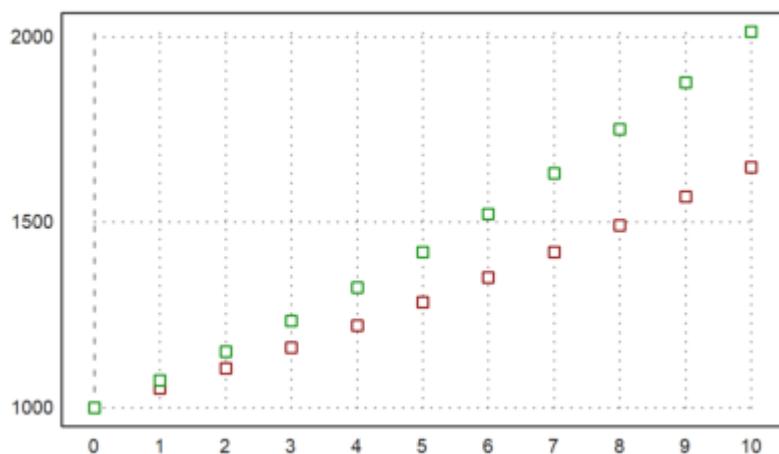
Seperti yang Anda lihat, sistem Jerman saya menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubahnya di pengaturan sistem atau di Excel, tetapi tidak diperlukan untuk membaca matriks ke EMT.

Cara termudah untuk membaca ini ke Euler adalah readmatrix(). Semua koma diganti dengan titik dengan parameter gt;comma. Untuk CSV bahasa Inggris, cukup abaikan parameter ini.

```
M=readmatrix("excel-data.csv", comma)
0 1000 1000 1 1051.3 1072.5 2 1105.2 1150.3 3 1161.8 1233.7 4 1221.4 1323.1
5 1284 1419.1 6 1349.9 1522 7 1419.1 1632.3 8 1491.8 1750.7 9 1568.3 1877.6 10
1648.7 2013.8
```

Mari kita plot ini.

```
plot2d(M'[1],M'[2:3], points,color=[red,green]):
```



Ada cara yang lebih mendasar untuk membaca data dari sebuah berkas. Anda dapat membuka berkas dan membaca angka baris demi baris. Fungsi getvectorline() akan membaca angka dari sebaris data. Secara default, fungsi ini mengharapkan titik desimal. Namun, fungsi ini juga dapat menggunakan koma desimal, jika Anda memanggil setdecimaldot(“,”) sebelum menggunakan fungsi ini.

Fungsi berikut adalah contohnya. Fungsi ini akan berhenti di akhir berkas atau baris kosong.

```
function myload (file) ...
open(file); M=[]; repeat until eof(); v=getvectorline(3); if length(v)>0 then
M=M_v; else break; end; end; return M; close(file); endfunction </pre>
myload(file)
0.39606 0.5217 0.079457 0.5777 0.6538 0.69742 0.83267 0.91434 0.676
Semua angka dalam berkas itu juga dapat dibaca dengan getvector().
open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
0.39606 0.5217 0.079457 0.5777 0.6538 0.69742 0.83267 0.91434 0.676
Jadi sangat mudah untuk menyimpan vektor nilai, satu nilai di setiap baris
dan membaca kembali vektor ini.
```

```

v=random(1000); mean(v)
0.51242
writematrix(v',file); mean(readmatrix(file)')
0.51242

```

### Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Misalkan, kita menulis tabel dengan tajuk baris dan kolom ke dalam sebuah berkas.

```

file="test.tab"; M=random(3,3); ... open(file,"w"); ... writetable(M,separator=",",labc=["one","two",""])
... close(); ... printfile(file)
one,two,three 0.92, 0.87, 0.47 0.51, 0.63, 0.98 0.11, 0.53, 0.32

```

Ini dapat diimpor ke Excel.

Untuk membaca berkas di EMT, kami menggunakan readtable().

```

M,headings=readtable(file, clabs); ... writetable(M,labc=headings)
one two three 0.92 0.87 0.47 0.51 0.63 0.98 0.11 0.53 0.32

```

### Menganalisis Garis

Anda bahkan dapat mengevaluasi setiap garis secara manual. Misalkan, kita memiliki garis dengan format berikut.

```
line="2020-11-03,Tue,1'114.05"
```

```
2020-11-03,Tue,1'114.05
```

Pertama, kita dapat membuat token pada baris tersebut.

```
vt=strtoks(line)
```

```
2020-11-03 Tue 1'114.05
```

Kemudian kita dapat mengevaluasi setiap elemen garis menggunakan evaluasi yang tepat.

```

day(vt[1]), ... indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])),
... strrepl(vt[3],"","","")()
7.3816e+05 2 1114

```

Dengan menggunakan ekspresi reguler, dimungkinkan untuk mengekstrak hampir semua informasi dari sebaris data.

Asumsikan kita memiliki baris berikut sebagai dokumen HTML.

```
line="<tr ><td 1145.45</td ><td 5.6</td ><td -4.5</td ><tr >
lt;trgt;lt;tdgt;1145.45lt;/tdgt;lt;tdgt;5.6lt;/tdgt;lt;tdgt;-4.5lt;/tdgt;lt;trgt;
```

Untuk mengekstraknya, kami menggunakan ekspresi reguler, yang mencari \* tanda kurung tutup gt;,

\* string apa pun yang tidak mengandung tanda kurung dengan \* subkecocokan "(...)”,

\* tanda kurung buka dan tutup menggunakan solusi terpendek,

\* lagi-lagi string apa pun yang tidak mengandung tanda kurung,

\* dan tanda kurung buka lt;.

Ekspresi reguler agak sulit dipelajari tetapi sangat ampuh.

```
pos,s,vt=strcmp(line,"(<)+ < .+? (<)+ < ");
```

Hasilnya adalah posisi kecocokan, string yang cocok, dan vektor string untuk sub-kecocokan.

```
for k=1:length(vt); vt[k](), end;
```

```
1145.5 5.6
```

Berikut adalah fungsi yang membaca semua item numerik antara lt;tdgt; dan lt;/tdgt;;

```
function readtd (line) ...
v=[]; cp=0; repeat pos,s,vt=strxfind(line,"<td.*?>(.+?)</td>",cp); until pos==0;
if length(vt)>0 then v=v—vt[1]; endif; cp=pos+strlen(s); end; return v; end-
function i/pre;>_ readtd(line+>_<td non-numerical>_</td>")
```

1145.45 5.6 -4.5 non-numerical

Membaca dari Web

Situs web atau berkas dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris.

Dalam contoh ini, kami membaca versi terkini dari situs EMT. Kami menggunakan ekspresi reguler untuk memindai "Versi ..." dalam judul.

```
function readversion () ...
urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html"); re-
peat 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 i/pre;>_ readversion
Version 2024-01-12
```

Input dan Output Variabel

Anda dapat menulis variabel dalam bentuk definisi Euler ke dalam file atau ke baris perintah.

```
writevar(pi,"mypi");
mypi = 3.141592653589793;
```

Untuk pengujian, kami membuat file Euler di direktori kerja EMT.

```
file="test.e"; ... writevar(random(2,2),"M",file); ... printfile(file,3)
M = [.. 0.3731518880081878, 0.5100363690187648; 0.8536630359298354,
0.5969300413220439];
```

Sekarang kita dapat memuat berkas tersebut. Berkas tersebut akan mendefinisikan matriks M.

```
load(file); show M,
M = 0.37315 0.51004 0.85366 0.59693
```

Ngomong-ngomong, jika writevar() digunakan pada suatu variabel, ia akan mencetak definisi variabel dengan nama variabel ini.

```
writevar(M); writevar(inch)
M = [.. 0.3731518880081878, 0.5100363690187648; 0.8536630359298354,
0.5969300413220439]; inch= 0.0254;
```

Kita juga dapat membuka berkas baru atau menambahkannya ke berkas yang sudah ada. Dalam contoh ini, kita menambahkannya ke berkas yang dibuat sebelumnya.

```
open(file,"a"); ... writevar(random(2,2),"M1"); ... writevar(random(3,1),"M2");
... close();
load(file); show M1; show M2;
M1 = 0.79972 0.80464 0.30558 0.16514 M2 = 0.075582 0.25656 0.11703
Untuk menghapus file apa pun gunakan fileremove().
fileremove(file);
```

Vektor baris dalam sebuah berkas tidak memerlukan koma, jika setiap angka berada di baris baru. Mari kita buat berkas seperti itu, tulis setiap baris satu per satu dengan writeln().

```
open(file,"w"); writeln("M = ["); ... for i=1 to 5; writeln("'" +random());
end; ... writeln("]"); close(); ... printfile(file)
M = [0.947939143784 0.834390864672 0.283694840089 0.583171970476 0.425722158135
];
```

```
load(file); M
[0.94794, 0.83439, 0.28369, 0.58317, 0.42572]
```

catatan : ketika mengenter perintah-perintah diatas ternyata hasil yang didapatkan berbeda-beda

Latihan soal

Nomor 1 Carilah rata-rata dan standar deviasi beserta plot dari data berikut

X = 1000,1500,1700,2500,3500,4000

```
X=[500,1000,1500,2000,3500,4000]; ... mean(X), dev(X),
```

2083.3 1393.4

```
aspect(1.5); boxplot(X);
```

```
![images/23030630044_Alfia
```

Nomor 2

Misalkan diberikan data skor hasil statistika dari 14 orang mahasiswa sebagai berikut:

69,65,58,95,75,79,85,91,75,79,75,95

Tentukan rata-rata dari data tersebut!

```
X=[69,65,58,95,75,79,85,91,75,79,75,95]
```

```
[69, 65, 58, 95, 75, 79, 85, 91, 75, 79, 75, 95]
```

```
mean(X)
```

78.417