

## EMT UNTUK PERHITUNGAN ALJABAR

---

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

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

### Contoh pertama

---

Menyederhanakan bentuk aljabar:

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

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

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

Menjabarkan:

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

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

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

### Baris Perintah

---

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

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

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

16.7551608191

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

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

50.2654824574  
100.530964915

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

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

0.540302305868

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

0.857553215846

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

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

```
1.41666666667
1.41421568627
1.41421356237
```

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

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

```
>%+ x=4+5; ...
// garis ini tidak akan terlihat setelah kursor keluar dari garisThis line will not be visible once the cursor is
```

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

81

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

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

```
1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237
```

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

```
>x := 1.5; // komentar pergi ke sini sebelum ...
repeat xnew:=(x+2/x)/2; until xnew~≈x; ...
x := xnew; ...
end; ...
x,
```

1.41421356237

Struktur kondisional juga berfungsi.

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

Thought so!

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

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

```
>sqrt(sin(10°)/cos(20°))
```

0.429875017772

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

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

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

2.5

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

### Sintaksis Dasar

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

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

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

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

30.65625

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

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

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

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

8.77908249441

Untuk menghitung ekspresi rumit seperti

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

Anda perlu memasukkannya dalam formulir baris.

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

23.2671801626

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

Hasil perhitungan ini adalah angka floating point. Angka ini dicetak dengan akurasi sekitar 12 digit secara default.

Pada baris perintah berikut, kita juga mempelajari cara merujuk ke hasil sebelumnya dalam baris yang sama.

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

0.47619047619  
10/21

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

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

14.4978445072

Operator numerik Euler meliputi

- + unary or operator plus
- unary or operator minus
- \*, /
- . the matrix product

`a^b` power for positive `a` or integer `b` (`a**b` works too)  
`n!` the factorial operator

dan masih banyak lagi.

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

```
sin,cos,tan,atan,asin,acos,rad,deg  
log,exp,log10,sqrt,logbase  
bin,logbin,logfac,mod,floor,ceil,round,abs,sign  
conj,re,im,arg,conj,real,complex  
beta,betai,gamma,complexgamma,ellrf,ellf,ellrd,elle  
bitand,bitor,bitxor,bitnot
```

Beberapa perintah memiliki alias, misalnya ln untuk log.

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

2  
0.5

$>\sin(30^\circ)$

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).

$$>2^3 \cdot 4, \quad (2^3)^4, \quad 2^{(3^4)}$$

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

## Bilangan Riil

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)
```

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

5 55555555555554\*16^-1

### String

Suatu string dalam Euler didefinisikan dengan " ".

>"Suatu string dapat berisi apa saja."

Suatu string dapat berisi apa saja

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

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

The area of the circle with radius 2 cm is 12.5663706144 cm<sup>2</sup>

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

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

Golden Ratio : 1.61803

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

```
>none
```

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

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

1234.5

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

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

affe  
charlie  
bravo

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

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

affe  
charlie  
bravo  
affe  
charlie  
bravo

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

String Unicode dapat dirangkai seperti string lainnya.

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

$\alpha = 45^\circ$

|

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

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

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

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

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

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

Ü is a German letter

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

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

We have  $\alpha=\beta$ .

Dimungkinkan juga untuk menggunakan entitas numerik.

```
>u"hnliches"
```

Ähnliches

### Nilai Boolean

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

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

```
0  
1
```

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

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

```
1
```

Operator Boolean mematuhi aturan bahasa matriks.

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

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

Anda dapat menggunakan fungsi `nonzero()` 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[nonzero(isprime(N))] //pilih anggota2 N yang prima
```

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

### Format Keluaran

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

```
>defformat; pi
```

```
3.14159265359
```

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

```
>longest pi
```

```
3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari angka ganda.

```
>printhex(pi)
```

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

Format keluaran dapat diubah secara permanen dengan perintah format.

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

```
0.33333  
3.14159  
0.84147
```

Format defaulnya adalah (12).

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

```
0.333333333333
```

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

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

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

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

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

```
3.1416
```

Fungsi "longestformat" juga mengatur format skalar.

```
>longestformat; pi
```

```
3.141592653589793
```

Sebagai referensi, berikut adalah daftar format keluaran yang paling penting.

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

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

Namun format keluaran EMT dapat diatur secara fleksibel.

```
>longestformat; pi,
```

```
3.141592653589793
```

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

```
3.14159
```

Standarnya adalah deformat().

```
>deformat; // default
```

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

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

```
4.934802200544679
```

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

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

25/12

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

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

-1.110223024625157e-16

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

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

0

## Ekspresi

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

Variabel global dapat digunakan dalam evaluasi.

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

12.56637061435917

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

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

-0.919535764538

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

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

36

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

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

45

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

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

45

Ekspresi dalam x sering digunakan seperti fungsi.

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

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

&gt;fx (2)

12

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

&gt;fx &amp;= diff(x^x,x); \$&amp;fx

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

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

&gt;"@(a,b) a^2+b^2", % (4,5)

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

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

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

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

&gt;fx &amp;= x^3-a\*x; ...
a=1.2; fx(0.5)

```
-0.475
```

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

&gt;fx(0.5,a=1.1)

```
-0.425
```

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

## Matematika Simbolik

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

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

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

&gt;\$&amp;44!

```
26582715747884487680436258110146158903196385280000000000
```

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

$$C(44, 10) = \frac{44!}{34! \cdot 10!}$$

&gt;\$&amp; 44!/(34!\*10!) // nilai C(44,10)

```
2481256778
```

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

&gt;\$binomial(44,10) //menghitung C(44,10) menggunakan fungsi binomial()

```
2481256778
```

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

Anda akan mempelajari bahwa hal berikut juga berfungsi

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

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

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

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

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

120

Dengan cara itu Anda dapat menggunakan solusi suatu persamaan dalam persamaan lainnya.

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

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

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

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

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

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

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

25

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

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

$$\begin{aligned} &x^4 + 4x^3 + 6x^2 + 4x + 1 \\ &4(x+1)^3 \end{aligned}$$

Sekali lagi, % mengacu pada hasil sebelumnya.

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

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

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

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

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

2432902008176640000

&gt;::: factor(10!)

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

&gt;::: factor(20!)

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

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

&gt;::: av:g\$ av^2;

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

&gt;fx &amp;= x^3\*exp(x), \$fx

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

$$x^3 e^x$$

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

&gt;&amp;(fx with x=5), \$%, &amp;float(%)

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

$$125 e^5$$

18551.64488782208

&gt;fx(5)

18551.6448878

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

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

&gt;&amp;(fx with x=10)-(fx with x=5), &amp;float(%)

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

2.20079141499189e+7

&gt;\$factor(diff(fx,x,2))

$$x (x^2 + 6x + 6) e^x$$

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

```
>tex(fx)
```

$x^3 \cdot e^x$

Ekspresi simbolik dapat dievaluasi seperti halnya ekspresi numerik.

```
>fx(0.5)
```

0.206090158838

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

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

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

Penugasan tersebut juga dapat bersifat simbolis.

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

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

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

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

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

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

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

1.56155281281

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

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

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

[-3.23607, 1.23607]

$$[x = -3.2360797749979, x = 1.2360797749979]$$

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

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

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

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

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

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

$$[[x = 2, y = 1], [x = 1, y = 2]]$$

Ekspresi simbolik dapat memiliki tanda, yang menunjukkan perlakuan khusus di Maxima. Beberapa tanda dapat digunakan sebagai perintah juga, yang lainnya tidak. Tanda ditambahkan dengan "|"

(bentuk yang lebih baik dari "ev(...,flags)")

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

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

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

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

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

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

## Fungsi

Dalam EMT, fungsi adalah program yang ditentukan dengan perintah “function”. Fungsi dapat berupa fungsi satu baris atau fungsi multibaris. Fungsi satu baris dapat berupa numerik atau simbolik. Fungsi satu baris numerik didefinisikan dengan “:=”.

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

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

```
> f(2)
```

4.472135955

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

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

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

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

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

0.786151377757

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

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

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

0.707106781187

Sebaiknya kita hilangkan definisi ulang tentang dosa ini.

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

0.707106781187

## Parameter Default

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

16

Menetapkannya akan menimpa nilai default.

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

80

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

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

16

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

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

24

Tetapi parameter yang ditetapkan akan menggantikan nilai global.

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

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

20

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

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

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

Fungsi simbolis dapat digunakan dalam ekspresi simbolis.

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

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

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

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

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

178.635099908

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

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

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

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

0.703467422498

Hal berikut ini juga dapat digunakan, karena Euler menggunakan ekspresi simbolik dalam fungsi g, jika tidak menemukan variabel simbolik g, dan jika ada fungsi simbolik g.

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

0.703467422498

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

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

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

$$(x + 2)^n$$

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

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

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

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

625

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$x^3 - x$$

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

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

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

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

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

yang tidak dapat dievaluasi secara simbolik.

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

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

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

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

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

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

```
2
```

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

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

$$b^2 - a b + b + a^2$$

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

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

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

```
17
```

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

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

$$\text{diff(expr, y, 2) + diff(expr, x, 2)}$$

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

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

```
0
```

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

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

$$10 (y^2 + x)^3 (9 y^2 + x + 2)$$

Untuk meringkas

- **&= mendefinisikan fungsi simbolik,**
- **:= mendefinisikan fungsi numerik,**
- **&&= mendefinisikan fungsi simbolik murni.**

## Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolik.

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

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

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

1.41421356237

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

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

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

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

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

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

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

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

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

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

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

Sekarang kita mencari titik, di mana polinomialnya adalah 2. Dalam solve(), nilai target default y=0

dapat diubah dengan variabel yang ditetapkan.

ami menggunakan y=2 dan mengeceknya dengan mengevaluasi polinomial pada hasil sebelumnya.

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

0.966715594851  
2

Memecahkan sebuah ekspresi simbolik dalam bentuk simbolik mengembalikan sebuah daftar solusi.

Kami menggunakan pemecah simbolik solve() yang disediakan oleh Maxima.

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

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

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

```
>longest sol()
```

-0.6180339887498949      1.618033988749895

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

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

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

0

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

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

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

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

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

dengan  $a = 3$ .

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

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

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

2.54116291558

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

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

2.54116291558

## Menyelesaikan Pertidaksamaan

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

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

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

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

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

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

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

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

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

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

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

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

*emptyset*

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

*universalset*

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned} & [6 < x, x < 8, y < -11] \text{ or } [8 < x, y < -11] \\ & \text{or } [x < 8, 13 < y] \text{ or } [x = y, 13 < y] \text{ or } [8 < x, x < y, 13 < y] \\ & \text{or } [y < x, 13 < y] \end{aligned}$$

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

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

## Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.

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

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

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

Hasil kali matriks dilambangkan dengan sebuah titik.

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

$$\begin{matrix} 3 \\ 4 \end{matrix}$$

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

$$[3, 4]$$

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

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

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

$$\begin{matrix} 11 \\ 25 \end{matrix}$$

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

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

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

```
> A.A
```

|    |    |
|----|----|
| 7  | 10 |
| 15 | 22 |

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

|   |    |
|---|----|
| 1 | 4  |
| 9 | 16 |

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

|    |     |
|----|-----|
| 37 | 54  |
| 81 | 118 |

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

|    |     |
|----|-----|
| 37 | 54  |
| 81 | 118 |

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

|   |   |
|---|---|
| 1 | 1 |
| 1 | 1 |

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

|          |          |
|----------|----------|
| 0.333333 | 0.666667 |
| 0.75     | 1        |

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

|     |
|-----|
| -2  |
| 2.5 |

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

|     |
|-----|
| -2  |
| 2.5 |

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

|   |   |
|---|---|
| 1 | 0 |
| 0 | 1 |

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

|   |   |
|---|---|
| 1 | 0 |
| 0 | 1 |

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

|   |    |
|---|----|
| 1 | 4  |
| 9 | 16 |

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

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

|    |
|----|
| 9  |
| 16 |

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

&gt;2\*A

|   |   |
|---|---|
| 2 | 4 |
| 6 | 8 |

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

&gt;[1,2]\*A

|   |   |
|---|---|
| 1 | 4 |
| 3 | 8 |

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

&gt;A\*[2,3]

|   |    |
|---|----|
| 2 | 6  |
| 6 | 12 |

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

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

|   |   |
|---|---|
| 1 | 2 |
| 1 | 2 |

&gt;A\*dup([1,2],2)

|   |   |
|---|---|
| 1 | 4 |
| 3 | 8 |

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

&gt;(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!

&gt;(1:5).(1:5)' // hasilkali vektor baris dan vektor kolom

55

&gt;sum((1:5)\*(1:5)) // sama hasilnya

55

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

&gt;(1:10)&lt;6 // menguji elemen-elemen yang kurang dari 6

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

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

&gt;sum((1:10)&lt;6) // banyak elemen yang kurang dari 6

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
112
```

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

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

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

Ini mengembalikan indeks elemen, yang bukan nol.

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

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

Indeks ini dapat digunakan untuk menetapkan elemen ke suatu nilai.

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

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

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

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

```

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

```

Dan dimungkinkan untuk mendapatkan elemen-elemen dalam vektor.

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

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

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

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

```

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

```

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

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

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

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

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

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

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

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

```

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

```

## Fungsi Matriks Lainnya (Membangun Matriks)

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

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

```

1      2      3
1      2      3

```

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

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

```

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

```

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

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

```
>A|1
```

```

0.032444      0.0534171      0.595713      0.564454      1
0.83916       0.175552       0.396988      0.83514       1

```

|           |          |          |          |
|-----------|----------|----------|----------|
| 0.0257573 | 0.658585 | 0.629832 | 0.770895 |
|-----------|----------|----------|----------|

1

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

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

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 1 | 2 | 3 |

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

|   |   |   |
|---|---|---|
| 1 | 1 | 3 |
| 2 | 2 | 3 |

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

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

|   |   |   |
|---|---|---|
| 1 | 1 | 9 |
| 2 | 4 | 9 |

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

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

|     |    |   |
|-----|----|---|
| 2   | 1  | 4 |
| [2, | 4] | 4 |

Untuk vektor, ada `length()`.

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

|   |   |   |
|---|---|---|
| 9 | 1 | 4 |
|---|---|---|

Ada banyak fungsi lain yang menghasilkan matriks.

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

|   |   |   |
|---|---|---|
| 1 | 1 | 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, |
|-----|----|----|

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

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

|         |          |          |
|---------|----------|----------|
| 0.66566 | 0.831835 | 0.544258 |
| 0.977   | 0.544258 | 0.544258 |

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

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

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

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

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

Mari kita uji.

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

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

Fungsi multdup() menduplikasi elemen-elemen sebuah vektor.

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

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

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

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

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

Untuk rotasi, Euler memiliki rotoleft() dan rottright().

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

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

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

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

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

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

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

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

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

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

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

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

Kita mulai dengan matriks identitas.

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

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |

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

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

```

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

```

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

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

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

```

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

```

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

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

```

1          2          3
4          5          6
7          8          9

```

Sekarang kita bisa mengekstrak diagonal.

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

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

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

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

```

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

```

## Vektorisasi

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

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

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

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

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

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

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

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

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

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

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

```
-0.357, -0.288, -0.171, 0]
```

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

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

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

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

Perhatikan, bahwa ini sangat berbeda dengan hasil kali matriks. Hasil kali matriks dilambangkan dengan sebuah titik “.” dalam EMT.

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

55

Secara default, vektor baris dicetak dalam format ringkas.

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

[1, 2, 3, 4]

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

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

5  
25

Untuk mentransposisikan matriks, kita menggunakan apostrof.

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

1  
2  
3  
4

Jadi kita dapat menghitung matriks A dikali vektor b.

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

30  
70

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

```
>v'.v
```

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

$v.v'$  menghitung norma v kuadrat untuk vektor baris  $v$ . Hasilnya adalah vektor  $1 \times 1$ , yang berfungsi seperti bilangan real.

```
>v.v'
```

30

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

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

30

Operator dan fungsi mematuhi bahasa matriks Euler.  
Berikut ini adalah ringkasan aturannya.

- Sebuah fungsi yang diterapkan pada vektor atau matriks diterapkan pada setiap elemen.
- Operator yang beroperasi pada dua matriks dengan ukuran yang sama diterapkan secara berpasangan pada elemen-elemen matriks.
- Jika dua matriks memiliki dimensi yang berbeda, keduanya diperluas dengan cara yang masuk akal, sehingga memiliki ukuran yang sama.

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

Berikut ini adalah kasus sederhana dengan operator ^.

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

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

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

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

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 4 | 6 |
| 3 | 6 | 9 |

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

```
>v.v'
```

```
14
```

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

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

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

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

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

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

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

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

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

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

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

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

6

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

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

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

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

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

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

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

4

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

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

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

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

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

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

Sekarang kita menambahkan kolom ke matriks.

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

|     | 0        | 1         | 0         |
|-----|----------|-----------|-----------|
| 45  | 0.785398 | 0.707107  | 0.707107  |
| 90  | 1.5708   | 0         | 1         |
| 135 | 2.35619  | -0.707107 | 0.707107  |
| 180 | 3.14159  | -1        | 0         |
| 225 | 3.92699  | -0.707107 | -0.707107 |
| 270 | 4.71239  | 0         | -1        |

|     |         |          |           |
|-----|---------|----------|-----------|
| 315 | 5.49779 | 0.707107 | -0.707107 |
| 360 | 6.28319 | 1        | 0         |

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

Pada contoh berikut, kita menghitung  $t_j^i$  untuk  $i$  dari 1 hingga  $n$ . Kita mendapatkan sebuah matriks, di mana setiap baris adalah tabel  $t^i$  untuk satu  $i$ . Dengan kata lain, matriks tersebut memiliki elemen-elemen

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

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

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

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

Kata kunci "map" membuat vektor fungsi. Fungsi ini sekarang akan bekerja untuk vektor angka.

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

[0, 2.05045, 13.7251, 113.336, 1241.03]

## Sub-Matriks dan Elemen Matriks

---

Untuk mengakses elemen matriks, gunakan notasi kurung.

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

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

Kita dapat mengakses baris lengkap dari sebuah matriks.

```
>A[2]
```

[4, 5, 6]

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

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

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-baris yang sesuai dari matriks.

Di sini kita menginginkan baris pertama dan kedua dari A.

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

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |

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

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

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

Trik indeks juga dapat digunakan pada kolom.

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

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

|   |   |   |
|---|---|---|
| 2 | 3 | 9 |
| 5 | 6 |   |
| 8 |   | 9 |

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

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

|   |   |   |
|---|---|---|
| 3 | 6 | 9 |
|   |   |   |

Sebagai artenatif, biarkan indeks pertama kosong.

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

|   |   |   |
|---|---|---|
| 2 | 3 | 9 |
| 5 | 6 |   |
| 8 |   | 9 |

Kita juga bisa mendapatkan baris terakhir A.

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

|   |   |   |
|---|---|---|
| 7 | 8 | 9 |
|---|---|---|

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

Hal ini sebenarnya mengubah matriks A yang tersimpan.

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

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

Kita juga dapat menetapkan nilai pada deretan A.

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

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

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

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

|   |   |    |
|---|---|----|
| 5 | 6 | -1 |
| 7 | 8 | 6  |
| 7 | 8 | 9  |

Selain itu, beberapa jalan pintas diperbolehkan.

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

|   |   |    |
|---|---|----|
| 0 | 0 | -1 |
| 0 | 0 | 6  |
| 7 | 8 | 9  |

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

```
>A[3]
```

[7, 8, 9]

## Menyortir dan Mengacak

Fungsi mengurutkan () mengurutkan vektor baris.

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

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

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

Mari kita acak sebuah vektor.

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

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

Indeks berisi urutan v yang tepat.

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

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

Hal ini juga berlaku untuk vektor string.

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

a  
d  
e  
a  
aa  
e

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

a  
a  
aa  
d  
e  
e

Seperti yang anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

Fungsi unique mengembalikan daftar terurut dari elemen unik vektor.

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

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

Hal ini juga berlaku untuk vektor string.

```
>unique(s)
```

```
a
aa
d
e
```

## Aljabar Linier

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

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

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

```
-4
4.5
```

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

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

```
8.790745908981989e-13
```

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

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

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

Determinan dari matriis ini adalah 0.

```
>det(A)
```

```
0
```

## Matriks Simbolik

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

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

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

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

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

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

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

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

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

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

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

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

$$(1 - x) (2 - x) - ab$$

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

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

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

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

Untuk mengekstrak vektor eigen tertentu, diperlukan pengindeksan yang cermat.

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

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

$$[1, -1]$$

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

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

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

Dalam ekspresi simbolik, gunakan dengan.

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

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

Akses ke baris matriks simbolik bekerja seperti halnya matriks numerik.

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

$$[1, a]$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

|   |   |
|---|---|
| 1 | 0 |
| 0 | 1 |

Misalnya, nilai eigen dari `A` dapat dihitung secara numerik.

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

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

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

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

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

## Nilai Numerik dalam Ekspresi simbolik

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

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

|   |         |
|---|---------|
| 1 | 3.14159 |
| 4 | 5       |

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

```
>${&A}
```

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

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

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

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

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

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

```
1.4142135623730950488016887242097B × 100
```

```
1.414213562373095
```

Ketepatan angka floating point yang besar dapat diubah.

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

```
3.14159265358979323846264338327950288419716939937510582097494 \
4592307816406286208998628034825342117068b0
```

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

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

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

## Demo - Suku Bunga

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

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

```
>K=5000
```

5000

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

```
>K*1.03
```

5150

Euler juga akan memahami sintaks berikut ini.

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

5150

Tetapi akan lebih mudah untuk menggunakan faktor.

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

1.03  
5150

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

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

6719.58189672

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

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

6719.58

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

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

Starting from 5000\$ you get 6719.58\$.

Bagaimana jika kita ingin mengetahui hasil antara dari tahun ke-1 hingga tahun ke-9? Untuk hal ini, bahasa matriks euler sangat membantu. Anda tidak perlu menulis perulangan, tetapi cukup memasukkan.

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

Real 1 x 11 matrix

5000.00 5150.00 5304.50 5463.64 ...

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

```
>short 0:10
```

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

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

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

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

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

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

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

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

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

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

```
1271.61
1271.6071
```

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

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

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

```
Real 1 x 11 matrix
5000.00      5150.00      5304.50      5463.64      ...

```

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

```
>VKr'
```

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

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

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

```
5150.00
5000.00      5150.00      5304.50
```

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

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

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

```
6719.60
6719.58
```

Perbedaannya sangat kecil.

## Menyelesaikan Persamaan

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

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

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

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

```
Real 1 x 11 matrix
5000.00    5350.00    5710.50    6081.82    ...

```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

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

```
Real 1 x 11 matrix
5000.00    4950.00    4898.50    4845.45    ...

```

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

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

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

```
Real 1 x 51 matrix
5000.00    4950.00    4898.50    4845.45    ...

```

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

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

48.00

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

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

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

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

```
-19.83
47.00
```

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

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

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

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

Perulangannya sama seperti di atas

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

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

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

Mari kita coba sebuah tes.

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

-19.83

Sekarang kita bisa menyelesaikan masalah kita.

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

3.15

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

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

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

-336.08

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

### Solusi Simbolik untuk Masalah Suku Bunga

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

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

$R + qK$

Sekarang kita bisa mengulangi hal ini.

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

$$\begin{aligned} q(q(q(R + qK) + R) + R) + R \\ q^3R + q^2R + qR + R + q^4K \end{aligned}$$

Kita melihat sebuah pola. Setelah n periode, kita memiliki

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

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

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

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

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

Mari kita buat sebuah fungsi untuk ini.

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

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

Fungsi ini melakukan hal yang sama seperti fungsi f kita sebelumnya. Tetapi fungsi ini lebih efektif.

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

-19.82504734650985  
-19.82504734652684

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

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

20.51

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

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

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

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

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

Biasanya rumus ini diberikan dalam bentuk

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

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

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

Kita dapat menyelesaikan laju  $R$  secara simbolis.

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

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

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

Tentu saja, kita memiliki batas berikut.

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

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

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

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

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

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

## Contoh Penyelesaian Soal-Soal Aljabar

### R.2 Exercise Set

Write an equivalent expression without negative exponents

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

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

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

Calculate.

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

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

## Syntesis

Saving plan. The formula

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

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

```
>function savingPlan(deposit, years, interestRate) ...
  return deposit * ((1 + (years / 12) ^ (12 * years)) - 1 / (interestRate / 12))
endfunction
```

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

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

319945404605897893588266627102000547270066125660294774116001581846226212624189363132318609976546568702425419606

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

round(result)

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

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

958953910649344256369046316836877451804413086859055960464735040618989731647696327342244760610867590961650965696

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

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

191790782129868831478141415755308114186322613668077106819445738748445595153148886793963402351078305863078346801

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

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

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

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

**R.3 Exercise Set**

Determine the terms and the degree of the polynomials

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

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

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

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

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

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

```
>
```

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

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

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

```
>
```

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

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

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

```
>
```

### Synthesis

Multiply. Assume that all exponents are natural numbers.

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

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

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

## R.4 Exercise Set

---

Factor the difference of squares

$$47.z^2 - 81$$

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

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

Factor the square of a binomial.

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

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

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

Factor the sum or difference of cubes.

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

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

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

Factor completely.

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

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

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

### Synthesis

Factor.

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

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

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

Factor. Assume that variables in exponents represent natural numbers.

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

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

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

**R.5 Exercise Set**

$$31). 7(3x + 6) = 11 - (x + 2)$$

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

$$\left[ x = -\frac{3}{2} \right]$$

$$36). y^2 - 4y - 45 = 0$$

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

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

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

for: h (Area of a trapezoid)

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

$$\left[ \begin{array}{l} \frac{2-(b_2+b_1)h}{2} \\ \frac{8-(b_2+b_1)h}{2} \end{array} \right] = 0$$

$$71). Ax + By = C$$

for A

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

□

Synthesis  
Solve

$$81). 3[5 - 3(4 - t)] - 2 = 5[3(5t - 4) + 8] - 26$$

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

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

$$87). 3x^3 + 6x^2 - 27x - 54 = 0$$

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

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

**R.6 Exercise Set**

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

Penyelesaian :  
mencari faktor pembilang

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

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

mencari faktor penyebut

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

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

jadi tugas kita menyelesaikan :

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

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

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

atau dengan perintah EMT untuk menyelesaikan soal ini adalah

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

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

17. Multiply or divide and, if possible, simplify

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

Penyelesaian:

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

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

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

$$\begin{aligned} & \frac{(s+r)(r^2 - s^2)}{r-s} \\ & s^2 + 2rs + r^2 \\ & 28. \frac{(c^3 + 8)}{(c^2 - 4)} : \frac{(c^2 - 2c + 4)}{(c^2 - 4c + 4)} \end{aligned}$$

Penyelesaian:

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

$$\begin{aligned} & \frac{c^2 - 4c + 4}{c - 2} \\ & c - 2 \end{aligned}$$

40. add or subtract and, if possible, simplify

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

Penyelesaian:

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

$$\begin{aligned} & 6y - \frac{5}{y} + \frac{6}{y^2} + 12 \\ & \frac{6y^3 + 12y^2 - 5y + 6}{y^2} \end{aligned}$$

62. Simplify

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

Penyelesaian:

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

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

**R.7 Exercise Set**

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

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

Penyelesaian:

```
>$&expand(2*(3)^1/2 + (5)^1/2)*((3)^1/2 - 3*(5)^1/2)
```

$$-33$$

**Review Exercise**

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

Penyelesaian :

Rumus angsuran per bulan:

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

dengan M : angsuran per bulan

P : jumlah pinjaman

r : suku bunga bulanan

n : jumlah total pembayaran (dalam bulan)

diketahui:

Harga rumah : \$98,000

Uang muka : \$16,000

suku bunga tahunan: 6 1/2%

6 1/2% diubah menjadi 6.5%

lama angsur : 25 tahun

ditanya: angsuran per bulan berapa

Variabel yang dibutuhkan pada rumus adalah :

```
>P = 98000-16000
```

82000.00

```
>r = 6.5/100
```

0.07

```
>n = 25*12
```

300.00

Substitusi semua variabel yang telah diperoleh ke rumus M

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

553.67

Jadi, biaya angsuran per bulan selama 25 tahun dengan suku bunga sebesar 6.5% persen per tahun adalah \$553,669872305

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

Penyelesaian:

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

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

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

Penyelesaian:

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

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

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

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

$$\begin{aligned} \text{expand}\left((a^n - b^n)^3\right) &= -b^{3n} + 3a^n b^{2n} - 3a^{2n} b^n + a^{3n} \\ &\quad -b^{3n} + 3a^n b^{2n} - 3a^{2n} b^n + a^{3n} \end{aligned}$$

### 2.3 Exercise Set

Given that

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

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

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

Find each of the following

Menyimpan semua nilai fx, gx, dan hx.

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

1. Mencari nilai

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

Penyelesaian:

fungsi komposisi di atas juga dapat ditulis sebagai :

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

Mencari nilai g(-1)

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

-3.00

Lalu mencari nilai f(-3)

```
>fx(-3)
```

-8.00

Perintah fungsi komposisi di EMT

Akan menghasilkan hasil yang sama dengan cara manual di atas. Cara ini lebih efisien.

```
>fx(gx(-1)) //cara cepat
```

-8.00

4. Mencari nilai

$$(g \circ h)\left(\frac{1}{2}\right)$$

Penyelesaian:

Fungsi komposisi di atas juga dapat ditulis

$$g(h\left(\frac{1}{2}\right))$$

```
>gx(hx(1/2))
```

-6.23

## 7. Mencari nilai

$$(f \circ h)(-3)$$

Penyelesaian:

Fungsi komposisi di atas bisa ditulis

$$f(g(-3))$$

&gt;fx(gx(-3))

28.00

## 11. Mencari nilai

$$(h \circ h)(2)$$

Penyelesaian

Fungsi komposisi di atas dapat ditulis

$$(h(h(2)))$$

&gt;hx(hx(2))

512.00

## 13. Mencari nilai

$$(f \circ f)(-4)$$

Penyelesaian:

Fungsi komposisi di atas dapat ditulis

$$f(f(-4))$$

&gt;fx(fx(-4))

-32.00

**3.1 Exercise Set**11. Simplify. Where answer in the form  $a + bi$ , where  $a$  and  $b$  are real numbers.

$$(-5 + 3i) + (7 + 8i)$$

Penyelesaian:

&gt;bilangan1 = -5 + 3\*I, bilangan2 = 7 + 8\*I, bilangan1 + bilangan2

```
-5.00+3.00i
7.00+8.00i
2.00+11.00i
```

$$21.(10 + 7i) - (5 + 3i)$$

Penyelesaian:

&gt;bilangan1 = 10 + 7\*I, bilangan2 = 5 + 3\*I, bilangan1 - bilangan2

```
10.00+7.00i
5.00+3.00i
5.00+4.00i
```

$$35)7i \cdot (2 - 5i)$$

Penyelesaian:

```
>bilangan1 = 7*I, bilangan2 = 2-5*I, bilangan1*bilangan2
```

0.00+7.00i  
2.00-5.00i  
35.00+14.00i

$$36) 3i \cdot (6 + 4i)$$

Penyelesaian:

```
>bilangan1 = 3*I, bilangan2 = 6 + 4*I, bilangan1*bilangan2
```

0.00+3.00i  
6.00+4.00i  
-12.00+18.00i

$$37) -2i \cdot (-8 + 3i)$$

Penyelesaian:

```
>bilangan1 = -2*I, bilangan2 = -8 + 3*I, bilangan1*bilangan2
```

-0.00-2.00i  
-8.00+3.00i  
6.00+16.00i

### 3.4 Exercise Set

---

$$1) \frac{1}{4} + \frac{1}{5} = \frac{1}{t}$$

```
> $&solve((1/4)+(1/5)=(1/t))
```

$$\left[ t = \frac{20}{9} \right]$$

$$3) \frac{x+2}{4} - \frac{x-1}{5} = 15$$

```
> $&solve(((x+2)/4)-((x-1)/5)=15)
```

$$[x = 286]$$

$$4) \frac{t+1}{3} - \frac{t-1}{2} = 1$$

```
> $&solve(((t+1)/3)-((t-1)/2))
```

$$[t = 5]$$

$$7) \frac{5}{3x+2} = \frac{3}{2x}$$

```
> $&solve((5/(3*x+2))=(3/2*x))
```

$$\left[ x = \frac{-\sqrt{11}-1}{3}, x = \frac{\sqrt{11}-1}{3} \right]$$

$$29). \sqrt{3x-4} = 1$$

```
> $&solve((3*x-4)^1/2 = 1)
```

$$[x = 2]$$

### 3.5 Exercise Set

---

$$23). |x+3| - 2 = 8$$

```
> $&solve(abs(x+3)-2=8, x)
```

$$[|x+3| = 10]$$

```
>$&solve(abs(x+3))
```

$$[x = -3]$$

```
>$&solve(abs(x+3)-2=8)
```

$$[|x + 3| = 10]$$

```
>$&solve((abs(x+3))=10)
```

$$[|x + 3| = 10]$$

```
>$&expand(abs(x+3)=10)
```

$$|x + 3| = 10$$

```
>$&solve((x+3)=10)
```

$$[x = 7]$$

```
>$&solve((x+3)=-10)
```

$$[x = -13]$$

$$28). |5x + 4| + 2 = 5$$

```
>$&solve(abs(5*x+4)+2=5, x)
```

$$[|5x + 4| = 3]$$

```
>$&solve((5*x+4)=3)
```

$$\left[ x = -\frac{1}{5} \right]$$

```
>$&solve((5*x+4)=-3)
```

$$\left[ x = -\frac{7}{5} \right]$$

$$32). 5 - |4x + 3| = 2$$

```
>$&solve(5-(abs(4*x+3)=2))
```

$$[|4x + 3| = 2]$$

```
>$&solve((4*x+3)=2)
```

$$\left[ x = -\frac{1}{4} \right]$$

```
>$&solve((4*x+3)=-2)
```

$$\left[ x = -\frac{5}{4} \right]$$

$$43). |2x| \geq 6$$

```
>&load(fourier_elim)
```

```
C:/Program Files/Euler x64/maxima/share/maxima/5.35.1/share/f\
ourier_elim/fourier_elim.lisp
```

```
>$&fourier_elim([abs(2*x)] >= 6, [x])
```

$$[[2(|x| - 3)] = 0] \vee [[2(|x| - 3)] > 0]$$

### Chapter 3 test

$$7). x + 5\sqrt{x} - 36 = 0$$

```
> $&solve (x+5*(x)^1/2-36=0)
```

$$\left[ x = \frac{72}{7} \right]$$

$$10).\sqrt{x+4}-\sqrt{x-4}=2$$

```
> $&solve ((x+4)^1/2 - (x-4)^1/2=2)
```

$$[x = 8]$$

$$9).\sqrt{x+4}-2=1$$

```
> $&solve ((x+4)^1/2-2=1)
```

$$[x = 2]$$

$$13).|x+3|\leq 4$$

```
> $&fourier_elim([abs(x+3)<=4], [x])
```

$$[x = 1] \vee [x = -7] \vee [-7 < x, x < 1]$$

$$14).|2x-1|<5$$

```
> $&fourier_elim([abs(2*x-1)<5], [x])
```

$$[-2 < x, x < 3]$$

#### 4.1 Exercise Set

$$36).f(x) = (x^2 - 5x + 6)^2$$

```
> $&solve ((x^2-5*x+6)^2)
```

$$[x = 3, x = 2]$$

$$37).f(x) = x^4 - 4x^2 + 3$$

```
> $&solve (x^4-4*x^2+3)
```

$$[x = -1, x = 1, x = -\sqrt{3}, x = \sqrt{3}]$$

$$40).f(x) = x^3 - x^2 - 2x + 2$$

```
> $&solve (x^3-x^2-2*x+2)
```

$$[x = -\sqrt{2}, x = \sqrt{2}, x = 1]$$

$$61).2y - 3 \geq 1 - y + 5$$

```
> $&fourier_elim([2*y-3>=1-y+5], [x])
```

$$[y - 3] \vee [y - 3 > 0]$$

$$64).\left|x + \frac{1}{4}\right| \leq \frac{2}{3}$$

```
> $&fourier_elim([x+(1/4)<=(2/3)], [x])
```

$$\left[ x = \frac{5}{12} \right] \vee \left[ x < \frac{5}{12} \right]$$

#### Mid-Chapter Mixed Review

$$18).g(x) = x^3 - 9x^2 + 4x - 10;$$

find g(-5)

```
> function g(x) := (x^3-9x^2+4x-10)
> g(-5)
```

-380.00

$$20).f(x) = 5x^4 + x^3 - x;$$

find

$$f(-\sqrt{-2})$$

```
>a = - sqrt(2)
```

-1.41

```
>fx &= 5*x^4 + x^3 - x; fx(a)
```

18.59

$$17).\frac{x^5 - 5}{x + 1}$$

```
>$&expand((x^5-5)/(x+1))
```

$$\frac{x^5}{x + 1} - \frac{5}{x + 1}$$

$$19).f(x) = 20x^2 - 40x;$$

find

$$f(\frac{1}{2})$$

```
>function f(x) :=(20*x^2-40*x)
>f(1/2)
```

-15.00

$$23).h(x) = x^3 - 2x^2 - 55x + 56$$

Penyelesaian :

```
>$&factor(x^3-2*x^2-55*x+56)
```

$$(x - 8) (x - 1) (x + 7)$$

```
>$&solve(x^3-2*x^2-55*x+56)
```

$$[x = -7, x = 8, x = 1]$$

$$24).g(x) = x^4 - 2x^3 - 13x^2 + 14x + 24$$

Penyelesaian :

```
>$&factor(x^4-2*x^3-13*x^2+14*x+24)
```

$$(x - 4) (x - 2) (x + 1) (x + 3)$$

```
>$&solve(x^4-2*x^3-13*x^2+14*x+24)
```

$$[x = -3, x = -1, x = 2, x = 4]$$

#### 4.3 Exercise Set

$$f(x) = x^2 + 4x^2 + x - 6$$

penyelesaian :

```
>$&solve(x^3+4*x^2+x-6)
```

$$[x = -3, x = -2, x = 1]$$

$$50).f(x) = x^4 + x^3 - 3x^2 - 5x - 2$$

Penyelesaian

```
> $& solve(x^4+x^3-3*x^2-5*x-2)
```

$$[x = 2, x = -1]$$

51).  $f(x) = x^3 - 7x + 6$

Penyelesaian

```
> $& solve(x^3-7*x+6)
```

$$[x = 1, x = 2, x = -3]$$

52).  $f(x) = x^3 - 12x + 16$

Penyelesaian :

```
> $& solve(x^3-12*x+16)
```

$$[x = -4, x = 2]$$

53).  $f(x) = -x^3 + 3x^2 + 6x - 8$

```
> $& solve(-x^3+3*x^2+6*x-8)
```

$$[x = -2, x = 4, x = 1]$$

## Menggambar Grafik 2D dengan EMT

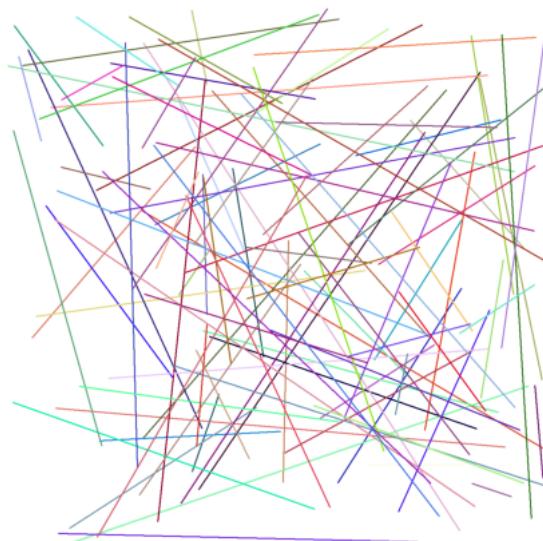
Notebook ini menjelaskan tentang cara menggambar berbagai kurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi plot2d() untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

### Plot Dasar

Ada 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 ini, kita hanya menggambar beberapa garis acak dengan berbagai warna. Untuk detail tentang fungsi-fungsi ini, pelajari fungsi-fungsi inti EMT.

```
> clg; // clear screen
> window(0,0,1024,1024); // use all of the window
> setplot(0,1,0,1); // set plot coordinates
> hold on; // start overwrite mode
> n=100; X=random(n,2); Y=random(n,2); // get random points
> colors=rgb(random(n),random(n),random(n)); // get random colors
> loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
> hold off; // end overwrite mode
> insimg; // insert to notebook
```



```
>reset;
```

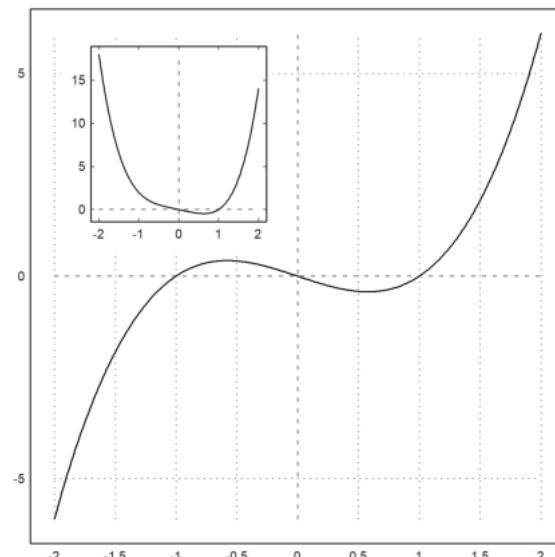
Perlu untuk menahan grafik, karena perintah plot() akan menghapus jendela plot.

Untuk menghapus semua yang telah kita lakukan, kita gunakan reset().

Untuk menampilkan gambar hasil plot di layar notebook, perintah plot2d() dapat diakhiri dengan titik dua (:). Cara lain adalah perintah plot2d() diakhiri dengan titik koma (:), kemudian menggunakan perintah insimg() untuk menampilkan gambar hasil plot.

Untuk contoh lain, kita menggambar plot sebagai inset di plot lain. Ini dilakukan dengan menentukan 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("x^3-x");
>xw=200; yw=100; ww=300; hw=300;
>ow>window();
>window(xw,yw,xw+ww,yw+hw);
>hold on;
>barclear(xw-50,yw-10,ww+60,ww+60);
>plot2d("x^4-x",grid=6);
```



```
>hold off;
>>window(ow);
```

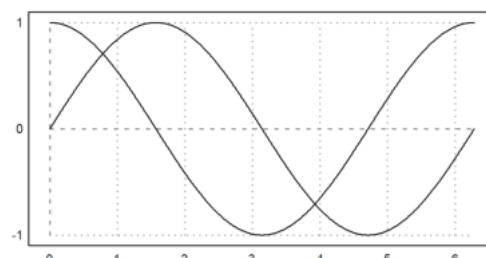
Plot dengan beberapa gambar dibuat dengan cara yang sama. Ada fungsi utilitas figure() untuk ini.

### Aspek Plot

Plot default menggunakan jendela plot persegi. Anda dapat mengubahnya dengan fungsi aspect(). Jangan lupa untuk mengatur ulang aspect nanti. Anda juga dapat mengubah default ini di menu dengan "Set Aspect" ke rasio aspek tertentu atau ke ukuran jendela grafik saat ini.

Namun, 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 pengaturan plot default termasuk rasio aspek.

Plot 2D dalam Euler

EMT Math Toolbox memiliki plot dalam 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d. Fungsi ini dapat memplot fungsi dan data.

Dimungkinkan untuk membuat plot di Maxima menggunakan Gnuplot atau di Python menggunakan Math Plot Lib.

Euler dapat memplot plot 2D dari

- ekspresi
- fungsi, variabel, atau kurva berparameter,
- vektor nilai xy,
- titik-titik pada bidang,
- kurva implisit dengan level atau daerah level.
- Fungsi kompleks

Gaya plot mencakup berbagai gaya untuk garis dan titik, plot batang, dan plot berbayangan.

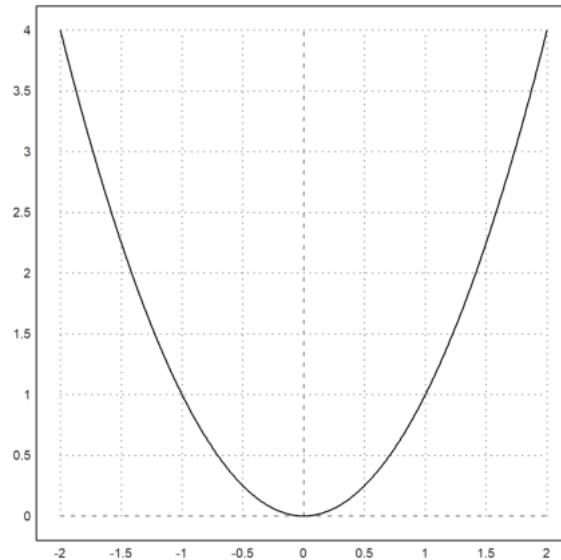
## Plot Ekspresi atau Variabel

Ekspresi tunggal dalam "x" (misalnya "4\*x^2") atau nama suatu fungsi (misalnya "f") menghasilkan grafik fungsi tersebut.

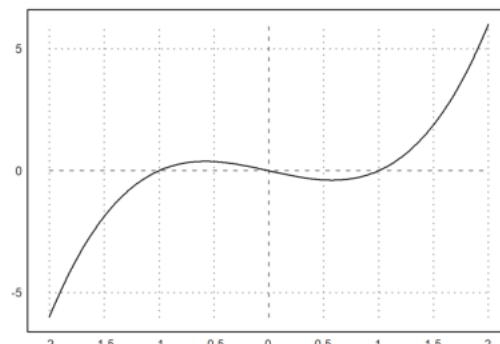
Berikut adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsi.

Catatan: Jika Anda mengakhiri baris perintah dengan titik dua ":", plot akan dimasukkan ke dalam jendela teks. Jika tidak, tekan TAB untuk melihat plot jika jendela plot tertutup.

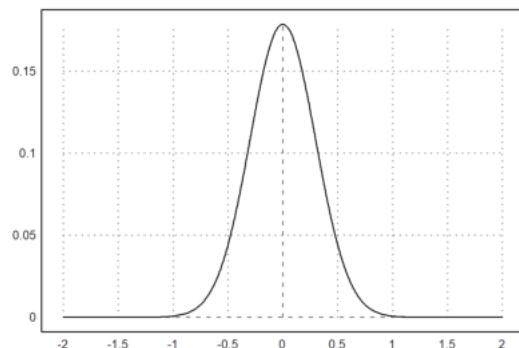
```
>plot2d("x^2") :
```



```
>aspect(1.5); plot2d("x^3-x") :
```



```
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25 baris
```

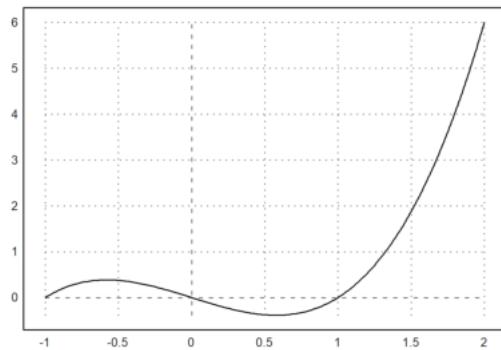


Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

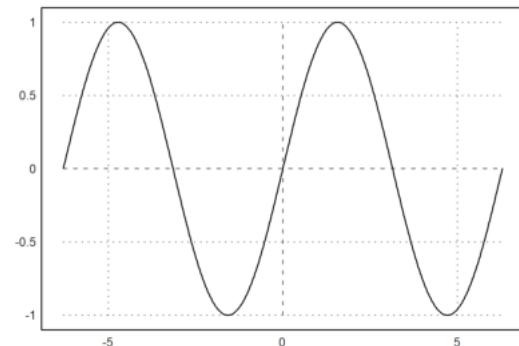
Rentang plot ditetapkan dengan parameter yang ditetapkan berikut

- 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)

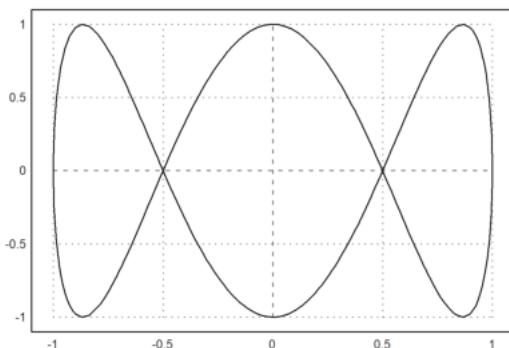
```
>plot2d("x^3-x",-1,2):
```



```
>plot2d("sin(x)",-2*pi,2*pi): // plot sin(x) pada interval [-2pi, 2pi]
```



```
>plot2d("cos(x)","sin(3*x)",xmin=0,xmax=2*pi):
```



Alternatif untuk titik dua adalah perintah insimg(baris), yang menyiapkan plot yang menempati sejumlah baris teks tertentu.

Dalam opsi, plot dapat diatur agar muncul

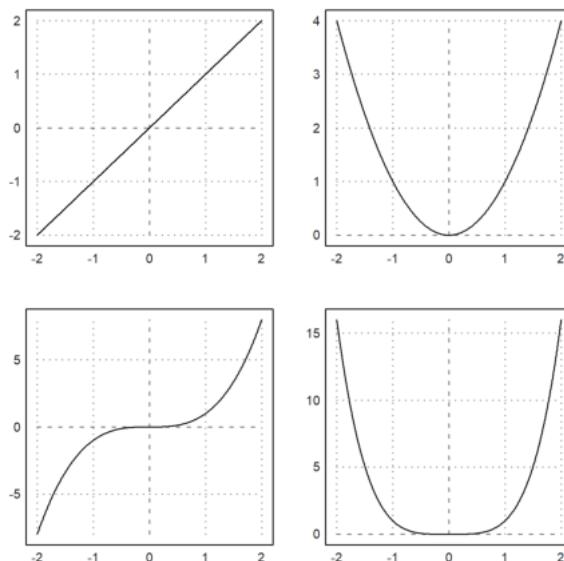
- di jendela terpisah yang dapat diubah ukurannya,
- di jendela catatan.

Lebih banyak gaya dapat dicapai dengan perintah plot yang spesifik.

Bagaimanapun, tekan tombol tabulator untuk melihat plot, jika tersembunyi.

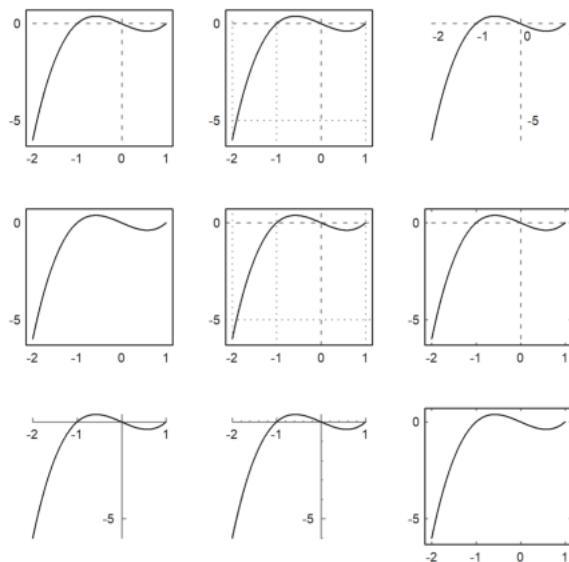
Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Dalam contoh ini, kita memplot  $x^1$  hingga  $x^4$  ke dalam 4 bagian jendela. figure(0) akan mengatur ulang jendela default.

```
>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):
```

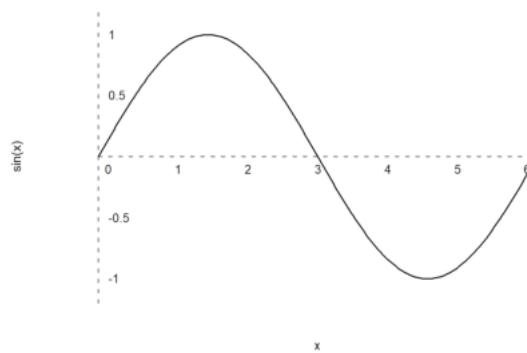


Jika argumen untuk `plot2d()` adalah ekspresi yang diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot.

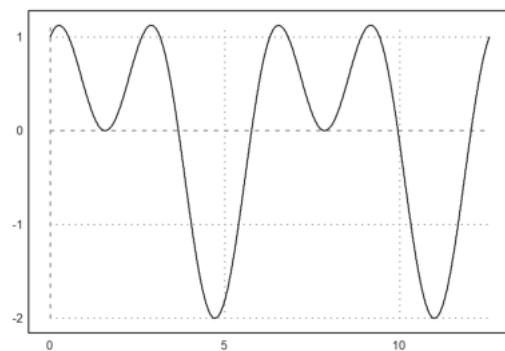
Sebagai alternatif, a, b, c, d dapat ditetapkan sebagai parameter yang ditetapkan sebagai a=... dst.

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("sin(x)+cos(2*x)",0,4pi):
```

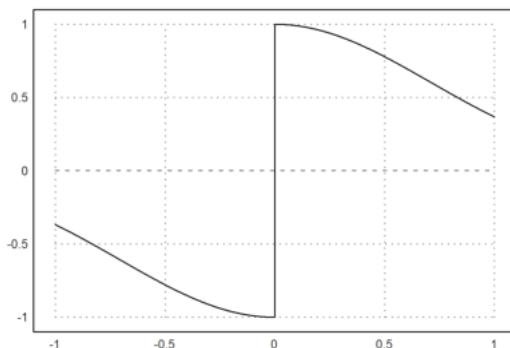


Gambar yang dihasilkan dengan memasukkan plot ke dalam jendela teks disimpan dalam direktori yang sama dengan buku catatan, secara default dalam subdirektori bernama "images". Gambar tersebut juga digunakan oleh eksport HTML.

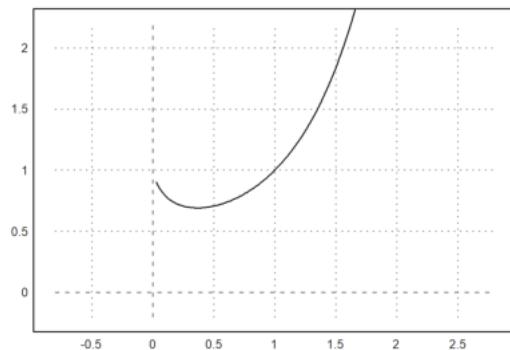
Anda cukup menandai gambar apa saja dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengekspor grafik saat ini dengan fungsi-fungsi di menu File.

Fungsi atau ekspresi dalam `plot2d` dievaluasi secara adaptif. Untuk kecepatan lebih tinggi, matikan plot adaptif dengan `<adaptive` dan tentukan jumlah subinterval dengan `n=...` Ini hanya diperlukan dalam kasus yang jarang terjadi.

```
>plot2d("sign(x)*exp(-x^2)",-1,1,<adaptive,n=10000):
```

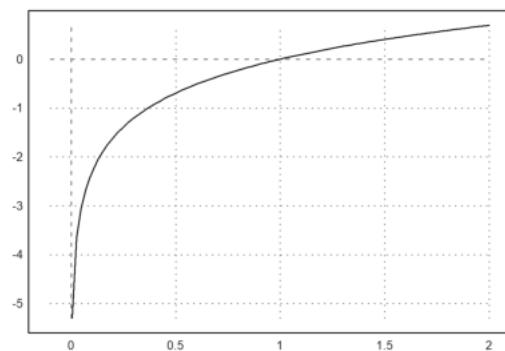


```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```



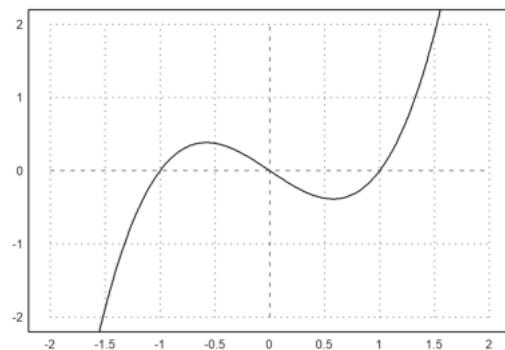
Perhatikan bahwa  $x^x$  tidak didefinisikan untuk  $x \leq 0$ . Fungsi plot2d menangkap kesalahan ini, dan mulai memplot segera setelah fungsi didefinisikan. Ini berfungsi untuk semua fungsi yang mengembalikan NAN di luar rentang definisinya.

```
>plot2d("log(x)",-0.1,2):
```

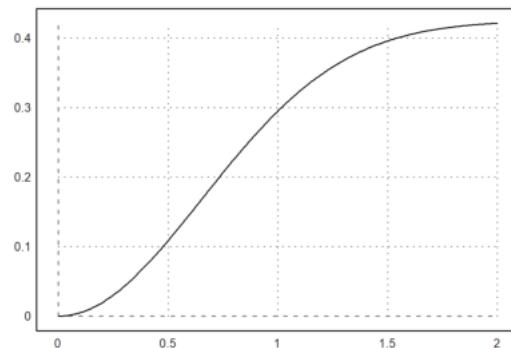


Parameter square=true (atau >square) memilih rentang y secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

```
>plot2d("x^3-x",>square):
```

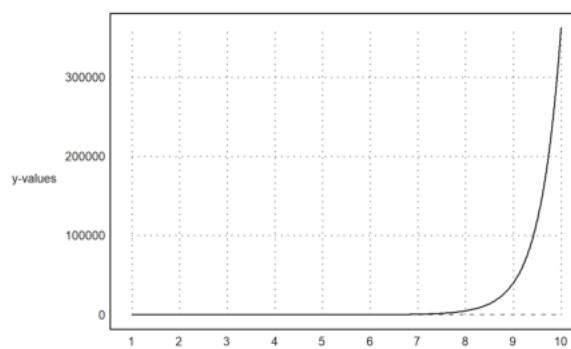


```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2); // plot integral
```



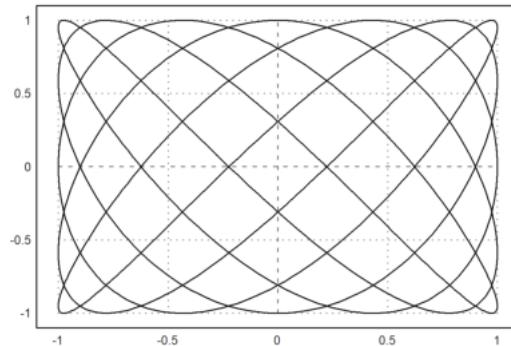
Jika Anda memerlukan lebih banyak ruang untuk label-y, panggil shrinkwindow() dengan parameter yang lebih kecil, atau tetapkan nilai positif untuk "lebih kecil" di plot2d().

```
>plot2d("gamma(x)",1,10,yl="y-values",smaller=6,<vertical):
```

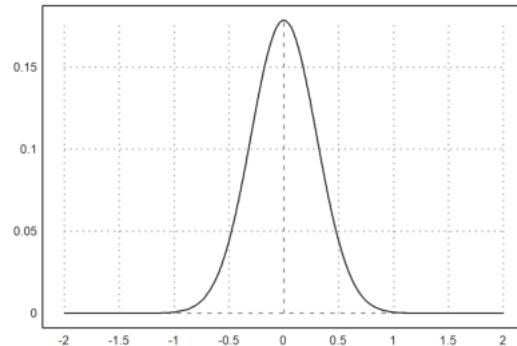


Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

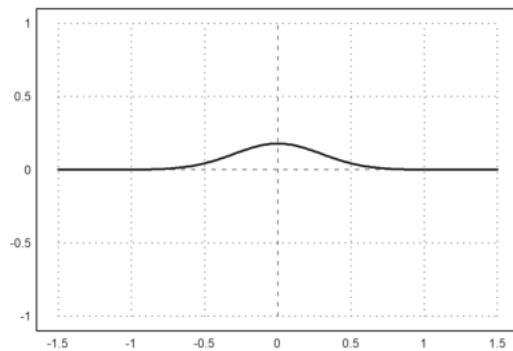
```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```



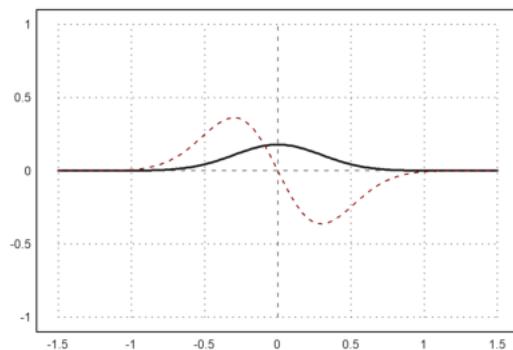
```
>a:=5.6; expr &= exp(-a*x^2)/a; // mendefinisikan ekspresi
>plot2d(expr,-2,2); // plot dari -2 hingga 2
```



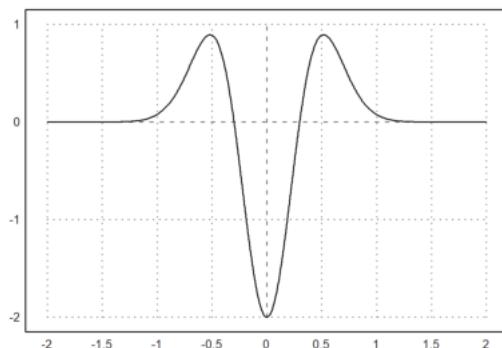
```
>plot2d(expr,r=1,thickness=2): // plot dalam kotak di sekitar (0,0)
```



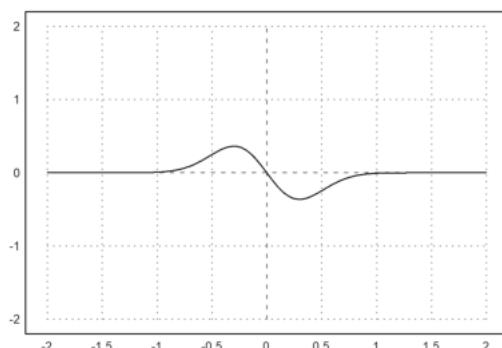
```
>plot2d(&diff(expr,x),>add,style="--",color=red): // menambahkan plot lain
```



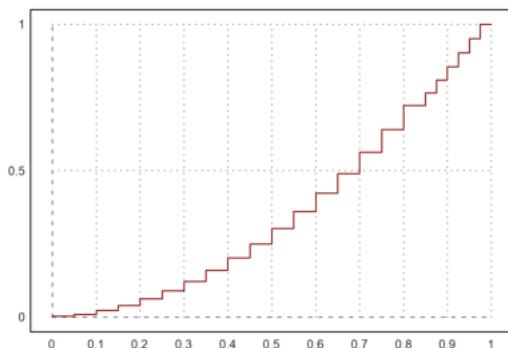
```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot dalam persegi panjang
```



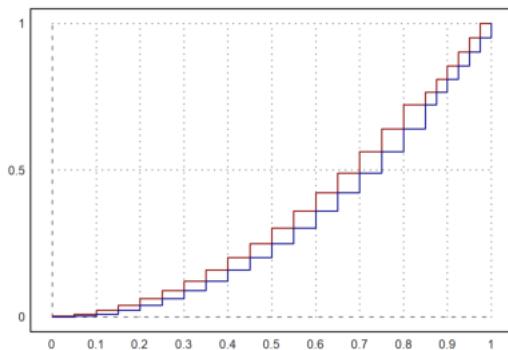
```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // pertahankan plot tetap persegi
```



```
>plot2d("x^2",0,1,steps=1,color=red,n=10):
```



```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```

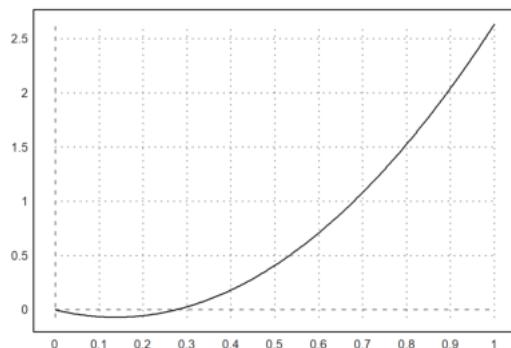


## Fungsi dalam Satu Parameter

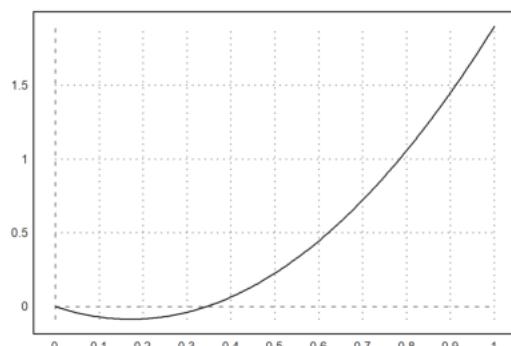
Fungsi plotting yang paling penting untuk plot planar adalah `plot2d()`. Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat di awal program.

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 kumpulan panggilan.

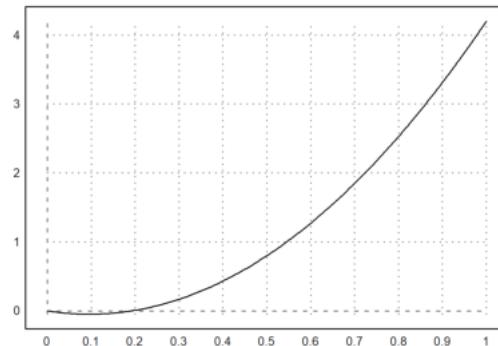
```
>function f(x,a) := x^2/a+a*x^2-x; // mendefinisikan sebuah fungsi
>a=0.3; plot2d("f",0,1;a); // plot dengan a=0.3
```



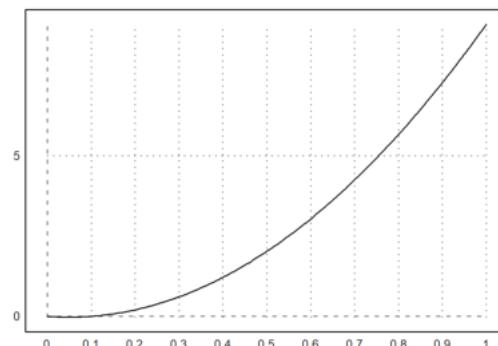
```
>plot2d("f",0,1;0.4); // plot dengan a=0.4
```



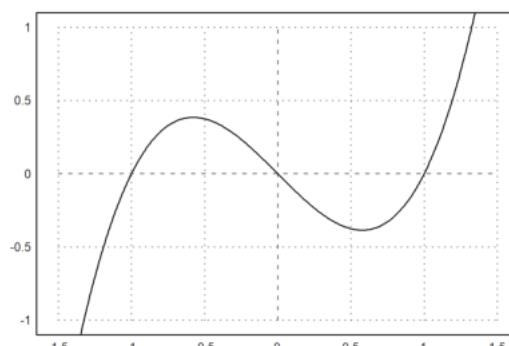
```
>plot2d({{"f",0.2}},0,1); // plot dengan a=0.2
```



```
>plot2d({{"f(x,b)",b=0.1}},0,1); // plot dengan 0.1
```



```
>function f(x) := x^3-x; ...
plot2d("f",x=1);
```



Berikut ini adalah ringkasan fungsi yang diterima

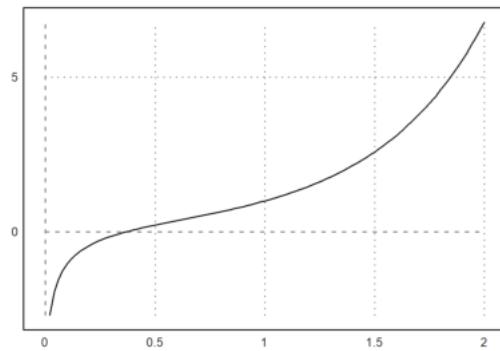
- ekspresi atau ekspresi simbolik dalam x
- fungsi atau fungsi simbolik berdasarkan nama seperti "f"
- fungsi simbolik hanya berdasarkan nama f

Fungsi plot2d() juga menerima fungsi simbolik. Untuk fungsi simbolik, hanya nama yang berfungsi.

```
>function f(x) &= diff(x^x,x)
```

$$\frac{d}{dx} x^x = x^x (\log(x) + 1)$$

```
>plot2d(f,0,2);
```

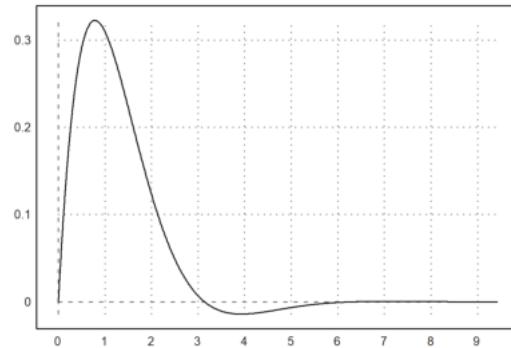


Tentu saja, untuk ekspresi atau ungkapan simbolik, nama variabel sudah cukup untuk memplotnya.

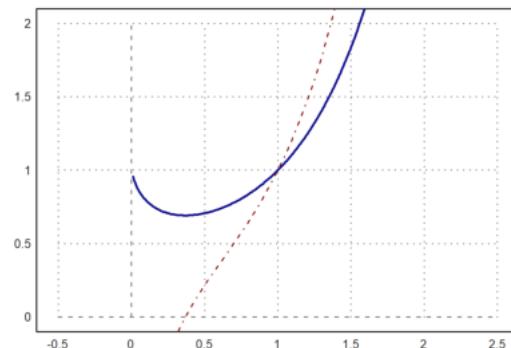
```
>expr &= sin(x)*exp(-x)
```

$$\frac{-x}{e} \sin(x)$$

```
>plot2d(expr,0,3pi):
```



```
>function f(x) &= x^x;
>plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
>plot2d(&diff(f(x),x),>add,color=red,style="--"):
```



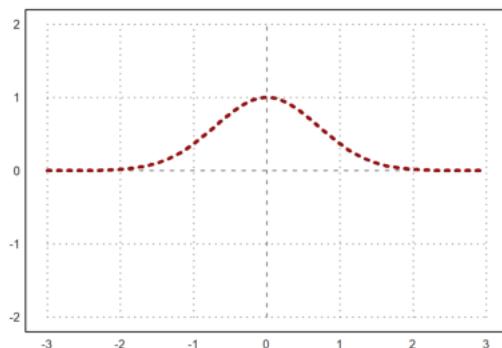
Untuk garis gaya, ada berbagai pilihan.

- `style="..."`. Pilih dari `"-", "--", "-.", ".-", "-.-"`.
- warna: Lihat dibawah untuk warna.
- ketebalan: Defaultnya adalah 1.

Warna dapat dipilih sebagai salah satu warna default, atau sebagai warna RGB.

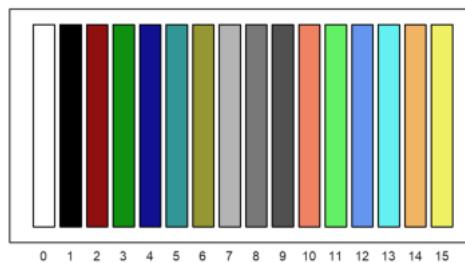
- 0..15: indeks warna default.
- warna konstan: putih, hitam, merah, hijau, biru, biru kehijauan, hijau kekuningan, abu-abu muda, abu-abu tua, oranye, hijau muda, toska, biru muda, oranye muda, kuning
- `rgb(merah,hijau,biru)`: parameteranya adalah bilangan riil dalam [0,1].

```
>plot2d("exp(-x^2)",r=2,color=red,thickness=3,style="--"):
```



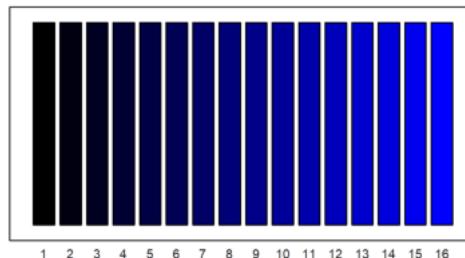
Berikut ini tampilan warna EMT yang telah ditetapkan sebelumnya.

```
>aspect(2); columnsplot(ones(1,16),lab=0:15,grid=0,color=0:15):
```



Namun anda dapat menggunakan warna apapun.

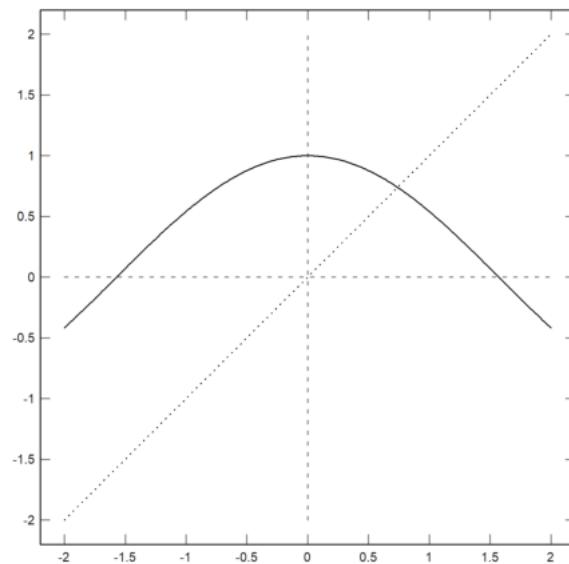
```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```



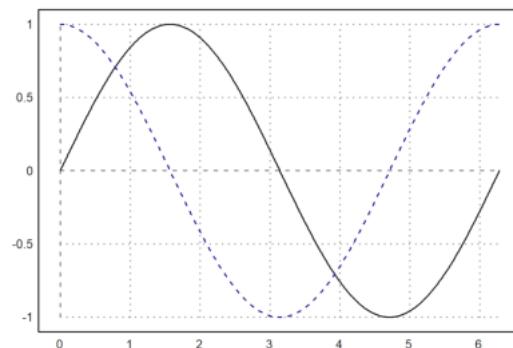
### Menggambar Beberapa Kurva pada bidang koordinat yang sama

Plotting lebih dari satu fungsi (multifungsi) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu metodenya adalah menggunakan >add untuk beberapa panggilan ke plot2d secara keseluruhan, kecuali panggilan pertama. Kami telah menggunakan fitur ini pada contoh di atas.

```
>aspect(); plot2d("cos(x)",x=2,grid=6); plot2d("x",style=".",>add):
```

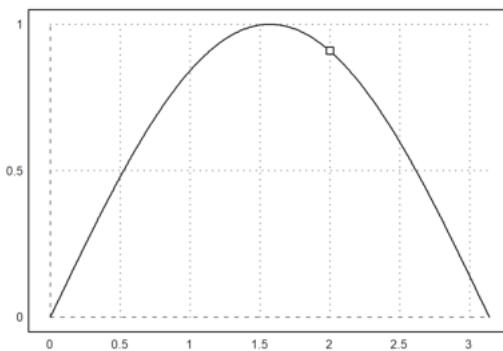


```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```



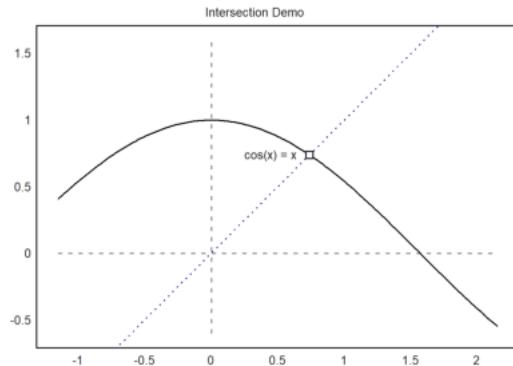
Salah satu kegunaan `>add` adalah untuk menambahkan titik pada kurva.

```
>plot2d("sin(x)",0,pi); plot2d(2,sin(2),>points,>add):
```



Kami menambahkan titik persimpangan dengan label (pada posisi "cl" untuk 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):
```



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.

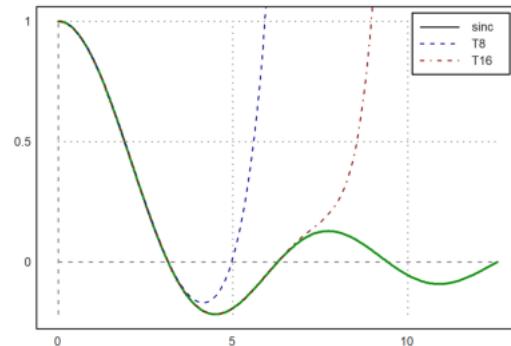
Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke `plot2d()`. Yang kedua dan ketiga memiliki tanda `>add` yang ditetapkan, yang membuat plot menggunakan rentang sebelumnya.

Kami menambahkan kotak label yang menjelaskan fungsinya.

```
>$taylor(sin(x)/x,x,0,4)
```

$$\frac{x^4}{120} - \frac{x^2}{6} + 1$$

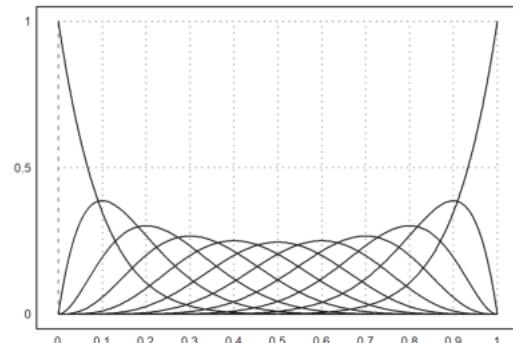
```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-."); ...
labelbox(["sinc","T8","T16"],styles=["-","--","-."], ...
colors=[black,blue,red]):
```



Dalam contoh berikut, kami menghasilkan Polinomial Bernstein.

$$B_i(x) = \binom{n}{i} x^i (1-x)^{n-i}$$

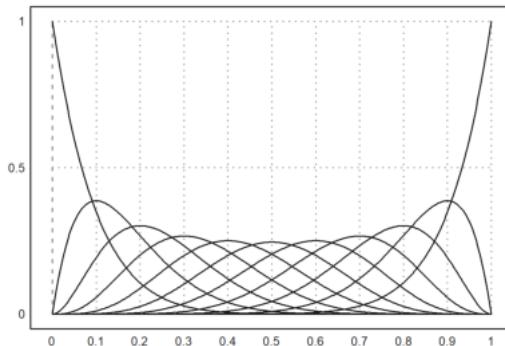
```
>plot2d("(1-x)^10",0,1); // plot fungsi pertama
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```



Metode kedua menggunakan sepasang matriks nilai-x dan matriks nilai-y yang berukuran sama.

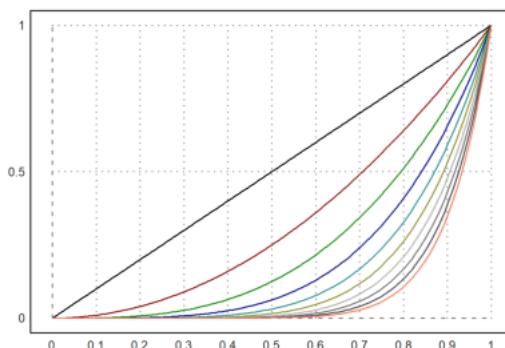
Kami membuat matriks nilai dengan satu Bernstein-Polinomial di setiap baris. Untuk ini, kami cukup menggunakan vektor kolom  $i$ . Lihat pengantar tentang bahasa matriks untuk mempelajari lebih detail.

```
>x=linspace(0,1,500);
>n=10; k=(0:n)'; // n adalah vektor baris, k adalah vektor kolom
>y=bin(n,k)*x^k*(1-x)^(n-k); // y adalah sebuah matriks, maka
>plot2d(x,y):
```



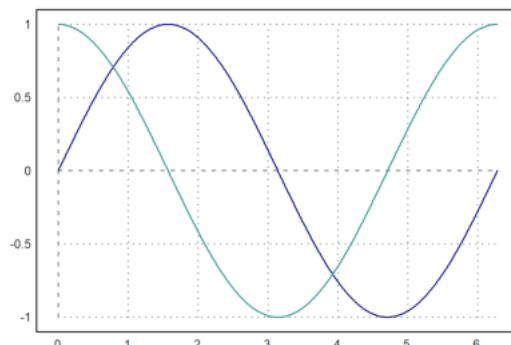
Perhatikan bahwa parameter warna dapat berupa vektor. Maka setiap warna digunakan untuk setiap baris matriks.

```
>x=linspace(0,1,200); y=x^(1:10)'; plot2d(x,y,color=1:10):
```

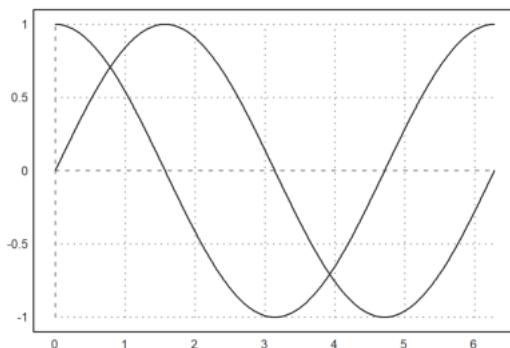


Metode lain adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan array warna, array gaya, dan array ketebalan dengan panjang yang sama.

```
>plot2d(["sin(x)","cos(x")],0,2pi,color=4:5):
```



```
>plot2d(["sin(x)","cos(x")],0,2pi): // plot vektor ekspresi
```



Kita bisa mendapatkan vektor tersebut dari Maxima menggunakan makelist() dan mxm2str().

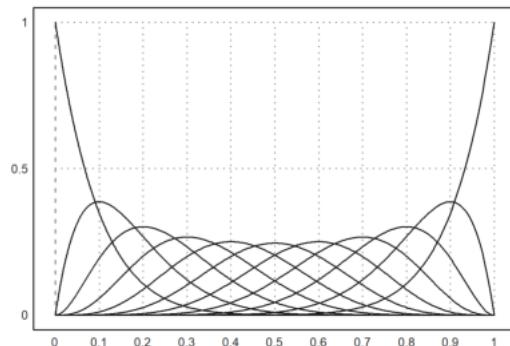
```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // membuat daftar
```

```
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          9  10
45 (1 - x) x , 10 (1 - x) x , x ]
```

```
>mxm2str(v) // mendapatkan vektor string dari vektor simbolik
```

```
(1-x)^10
10*(1-x)^9*x
45*(1-x)^8*x^2
120*(1-x)^7*x^3
210*(1-x)^6*x^4
252*(1-x)^5*x^5
210*(1-x)^4*x^6
120*(1-x)^3*x^7
45*(1-x)^2*x^8
10*(1-x)*x^9
x^10
```

```
>plot2d(mxm2str(v),0,1); // fungsi plot
```

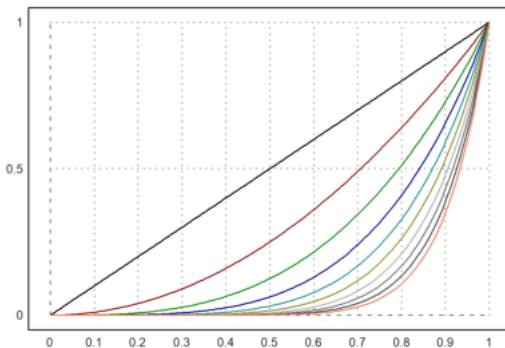


Alternatif lain adalah menggunakan bahasa matriks Euler.

Jika suatu ekspresi menghasilkan matriks fungsi, dengan satu fungsi pada 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);
```

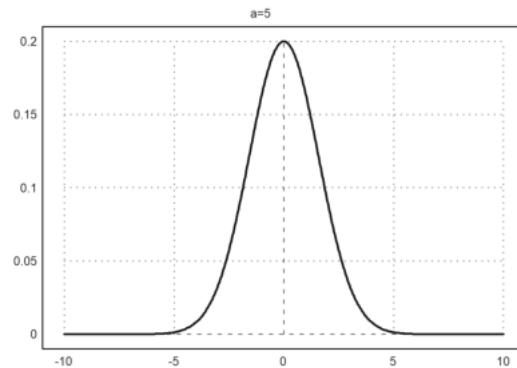


Ekspresi dan fungsi satu baris dapat melihat variabel global.

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, kita meneruskan a=5 ke fungsi f, yang kita plot dari -10 hingga 10.

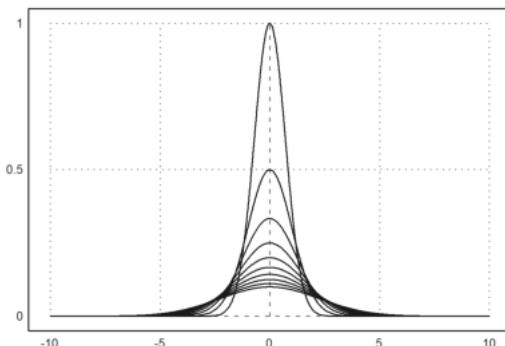
```
>function f(x,a) := 1/a*exp(-x^2/a); ...
plot2d("f",-10,10,5,thickness=2,title="a=5");
```



Atau, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut koleksi panggilan, dan itu adalah cara yang lebih disukai untuk meneruskan argumen ke 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).

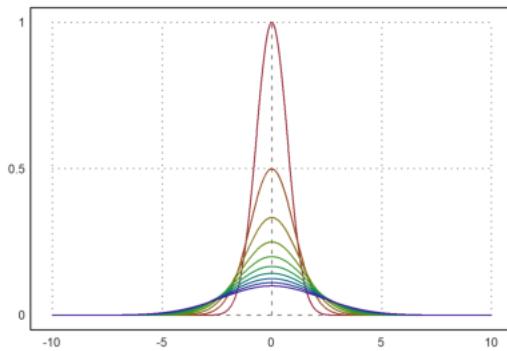
```
>plot2d({{"f",1}},-10,10); ...
for a=2:10; plot2d({{"f",a}},>add); end;
```



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));
```



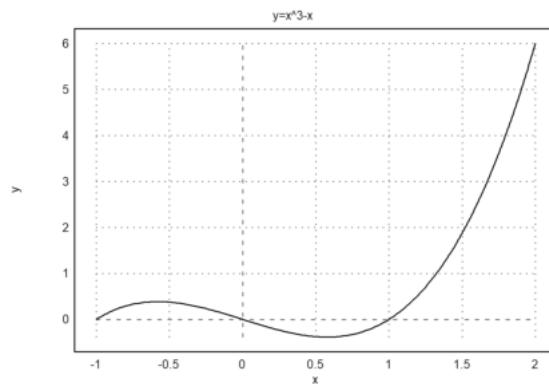
### Label Teks

Dekorasi sederhana bisa berupa

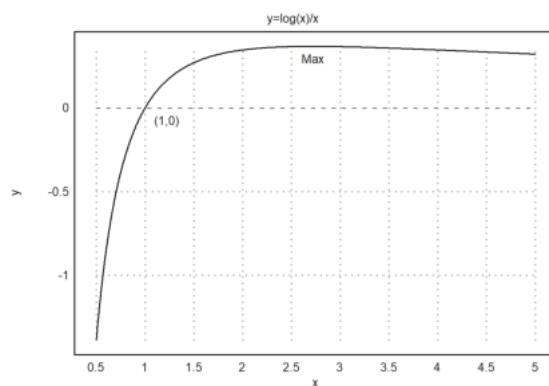
- judul dengan title="..."
- label x dan y dengan xl="...", yl="..."
- label teks dengan label("...",x,y)

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") :
```

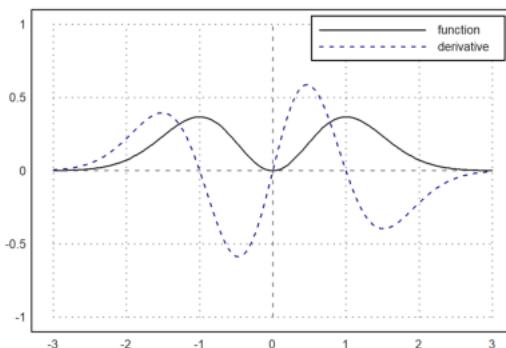


```
>expr := "log(x)/x"; ...
plot2d(expr, 0.5, 5, title="y="+expr, xl="x", yl="y"); ...
label("(1,0)", 1, 0); label("Max", E, expr(E), pos="lc"):
```



Ada juga fungsi labelbox(), yang dapat menampilkan fungsi dan teks. Fungsi ini mengambil vektor string dan warna, satu item untuk setiap fungsi.

```
>function f(x) &= x^2*exp(-x^2); ...
plot2d(&f(x), a=-3, b=3, c=-1, d=1); ...
plot2d(&diff(f(x),x), >add, color=blue, style="--"); ...
labelbox(["function", "derivative"], styles=["-", "--"], ...
colors=[black, blue], w=0.4):
```

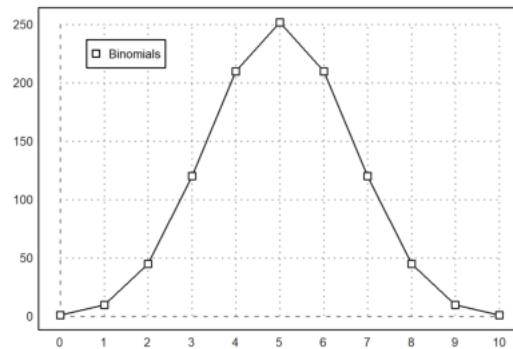


Kotak tersebut ditambatkan di kanan atas secara default, tetapi `>left` menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat mana pun yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angka-angkanya adalah pecahan dari ukuran jendela grafis. Lebarnya otomatis.

Untuk plot titik, kotak label juga berfungsi. Tambahkan parameter `>points`, atau vektor bendera, satu untuk setiap label.

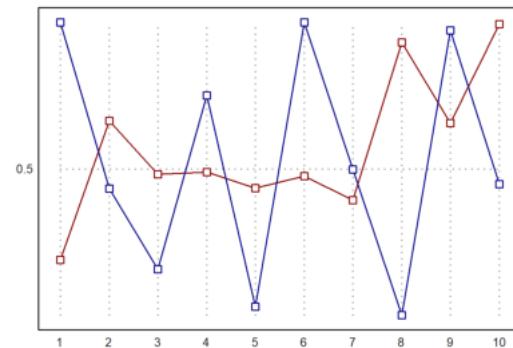
Dalam contoh berikut, hanya ada satu fungsi. Jadi, kita dapat menggunakan string, bukan vektor string. Kita tetapkan warna teks menjadi hitam untuk contoh ini.

```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
tcolor=black,>left):
```



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).

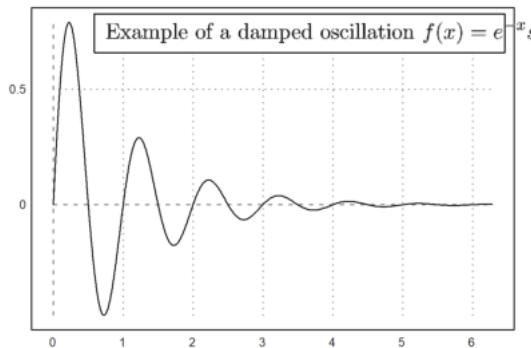
```
>statplot(1:10,random(2,10),color=[red,blue]):
```



Fitur serupa adalah fungsi `textbox()`.

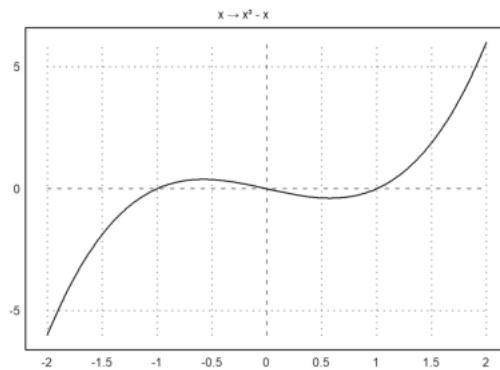
Lebarnya secara default adalah lebar maksimal baris teks. Namun, lebarnya juga dapat diatur oleh pengguna.

```
>function f(x) &= exp(-x)*sin(2*pi*x); ...
plot2d("f(x)",0,2pi); ...
textbox(latex("\text{Example of a damped oscillation}\backslash f(x)=e^{-x}\sin(2\pi x)"),w=0.85):
```



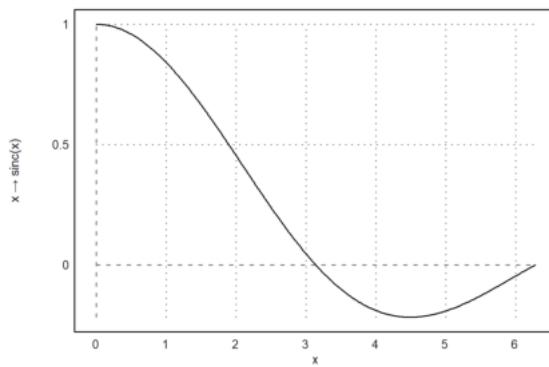
Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaksis EMT untuk informasi lebih lanjut tentang string Unicode).

```
>plot2d("x^3-x",title=u"x &rarr; x³ - x"):
```



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):
```



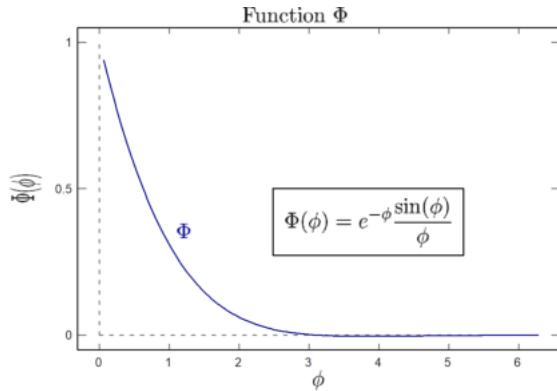
## LaTeX

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 opsi menu.

Perlu dicatat, bahwa penguraian LaTeX berlangsung lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil `latex()` sebelum loop sekali dan menggunakan hasilnya (gambar dalam matriks RGB).

Pada plot berikut, kami menggunakan LaTeX untuk label x dan y, label, kotak label, dan judul plot.

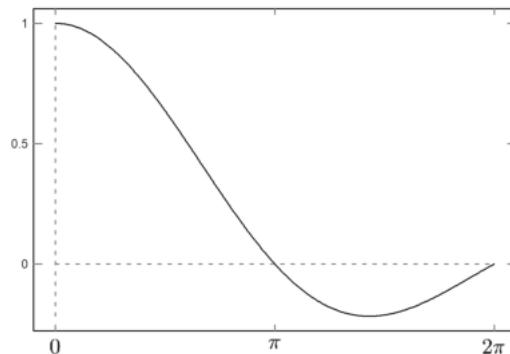
```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
    title=latex("\text{Function } \$\Phi\$"), ...
    xl=latex("\phi"),yl=latex("\Phi(\phi)"); ...
    textbox( ...
        latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
    label(latex("\Phi",color=blue),1,0.4);
```



Seringkali, 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 dengan membuat plot kosong dengan bingkai menggunakan grid=4, lalu menambahkan grid dengan ygrid() dan xgrid(). Dalam contoh berikut, kami menggunakan tiga string LaTeX untuk label pada sumbu x dengan xtick().

```
>plot2d("sinc(x)",0,2pi,grid=4,<ticks); ...
ygrid(-2:0.5:2,grid=6); ...
xgrid([0:2]*pi,<ticks,grid=6); ...
xtick([0,pi,2pi],["0","\pi","2\pi"],>latex);
```



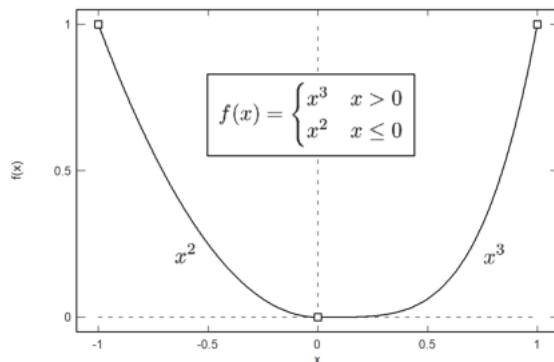
Tentu saja, fungsi juga dapat digunakan.

```
>function map f(x) ...
if x>0 then return x^4
else return x^2
endif
endfunction
```

Parameter "map" membantu penggunaan fungsi untuk vektor. Untuk plot, parameter ini tidak diperlukan. Namun, untuk menunjukkan bahwa vektorisasi berguna, kami menambahkan beberapa poin penting ke plot pada x=-1, x=0, dan x=1.

Pada plot berikut, kami juga memasukkan beberapa kode LaTeX. Kami menggunakanannya untuk dua label dan satu kotak teks. Tentu saja, Anda hanya dapat menggunakan LaTeX jika Anda telah menginstal LaTeX dengan benar.

```
>plot2d("f",-1,1,xl="x",yl="f(x)",grid=6); ...
plot2d([-1,0,1],f([-1,0,1]),>points,>add); ...
label(latex("x^3"),0.72,f(0.72)); ...
label(latex("x^2"),-0.52,f(-0.52),pos="ll"); ...
textbox( ...
    latex("f(x)=\begin{cases} x^3 & x>0 \\ x^2 & x \leq 0 \end{cases}"), ...
    x=0.7,y=0.2);
```



### Interaksi User

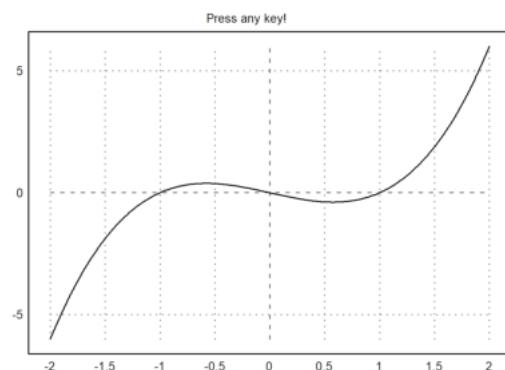
Saat memplot fungsi atau ekspresi, parameter `>user` memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol cursor atau mouse. Pengguna dapat

- zoom dengan + or -
- memindahkan plot dengan tombol cursor
- memilih jendela plot dengan mouse
- mengatur ulang tampilan dengan spasi
- keluar dengan return

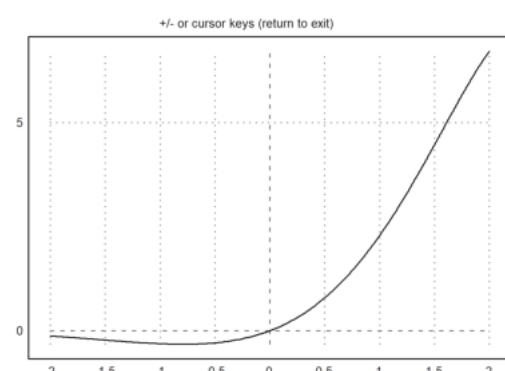
Tombol spasi akan mengatur ulang plot ke jendela plot asli.

Saat memetakan data, tanda `>user` akan menunggu penekanan tombol.

```
>plot2d({{"x^3-a*x",a=1}},>user,title="Press any key!":
```



```
>plot2d("exp(x)*sin(x)",user=true, ...
        title="+/- or cursor keys (return to exit)":
```



Berikut ini menunjukkan cara interaksi pengguna yang canggih (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi bawaan `mousedown()` menunggu kejadian mouse atau keyboard. Fungsi ini melaporkan gerakan mouse ke atas dan penekanan tombol. Fungsi `dragpoints()` memanfaatkan fungsi ini, dan memungkinkan pengguna menyeret titik m pun dalam plot.

Pertama-tama, kita perlu fungsi `plotf`. Misalnya, kita melakukan interpolasi pada 5 titik dengan polinomial. Fungsi tersebut harus diplot ke dalam area plot yang tetap.

```
>function plotf(xp,yp,select) ...
  d=interp(xp,yp);
  plot2d("interpval(xp,d,x)";d,xp,r=2);
```

```

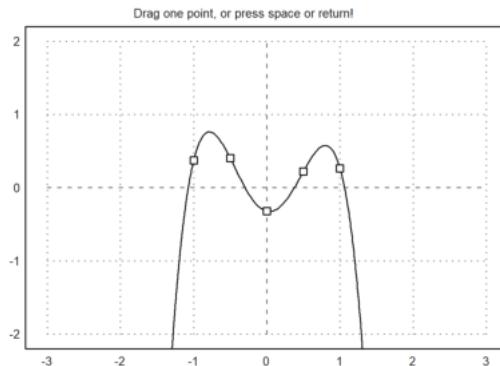
plot2d(xp,yp,>points,>add);
if select>0 then
  plot2d(xp[select],yp[select],color=red,>points,>add);
endif;
title("Drag one point, or press space or return!");
endfunction

```

Perhatikan parameter titik koma di plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, yang 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);
```



Ada juga fungsi yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

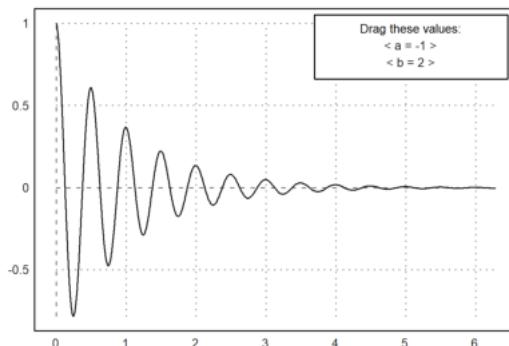
Pertama kita butuh fungsi plot.

```
>function plotf([a,b]) := plot2d("exp(a*x)*cos(2pi*b*x)",0,2pi;a,b);
```

Kemudian kita perlu nama untuk parameter, nilai awal dan matriks rentang nx2, secara opsional baris judul.

Terdapat slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakannya.

```
>dragvalues("plotf",["a","b"],[-1,2],[[-2,2];[1,10]], ...
heading="Drag these values:",hcolor=black);
```

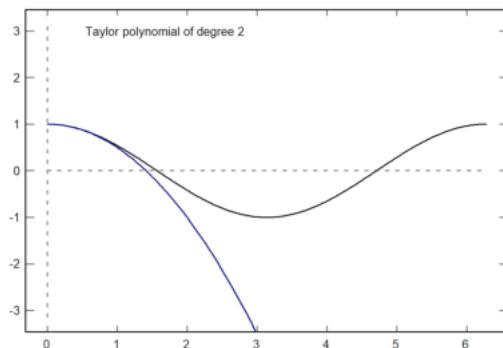


Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Misalnya, kita menulis fungsi plot, yang memplot polinomial Taylor berderajat n ke fungsi kosinus.

```
>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
```

Sekarang kita mengizinkan 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.

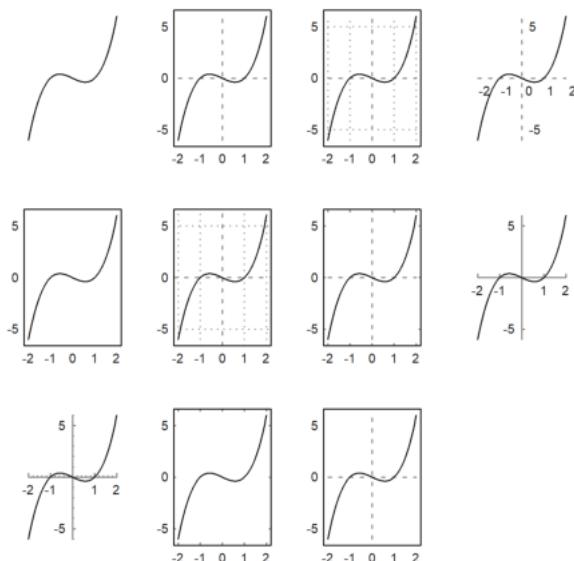
```
>function dragtest ...
plot2d(null,r=1,title="Drag with the mouse, or press any key!");
start=0;
repeat
{flag,m,time}=mousedrag();
if flag==0 then return; endif;
if flag==2 then
    hold on; mark(m[1],m[2]); hold off;
endif;
end
endfunction
```

```
>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().

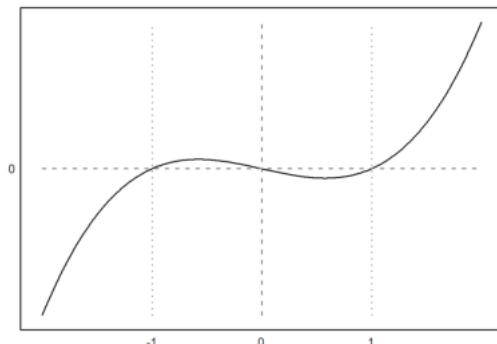
```
>aspect();
>figure(3,4); ...
figure(1); plot2d("x^3-x",grid=0); ... // tidak ada grid, bingkai atau sumbu
> figure(2); plot2d("x^3-x",grid=1); ... // sumbu x-y
> figure(3); plot2d("x^3-x",grid=2); ... // tanda centang default
> figure(4); plot2d("x^3-x",grid=3); ... // sumbu x-y dengan label di dalamnya
> figure(5); plot2d("x^3-x",grid=4); ... // tidak ada tanda centang, hanya label
> figure(6); plot2d("x^3-x",grid=5); ... // default, tetapi tidak ada margin
> figure(7); plot2d("x^3-x",grid=6); ... // sumbu saja
> figure(8); plot2d("x^3-x",grid=7); ... // sumbu saja, centang pada sumbu
> figure(9); plot2d("x^3-x",grid=8); ... // sumbu saja, ticks lebih halus pada sumbu
> figure(10); plot2d("x^3-x",grid=9); ... // default, small ticks inside
> figure(11); plot2d("x^3-x",grid=10); ....// no ticks, hanya sumbu
> figure(0):
```



Parameter `<frame` mematikan frame, dan `framecolor=blue` mengatur frame menjadi warna biru.

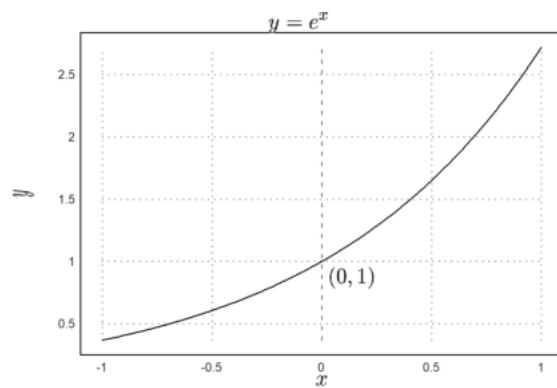
Jika Anda menginginkan tanda centang Anda sendiri, Anda dapat menggunakan `style=0`, dan menambahkan semuanya nanti.

```
>aspect(1.5);
>plot2d("x^3-x",grid=0); // plot
>frame; xgrid([-1,0,1]); ygrid(0); // menambahkan frame dan grid
```



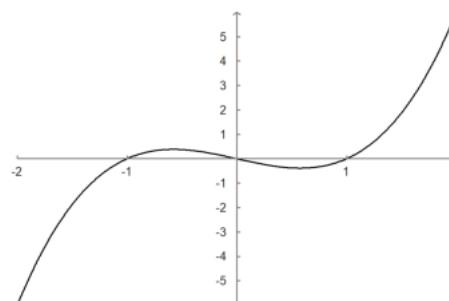
Untuk judul plot dan label sumbu, lihat contoh berikut.

```
>plot2d("exp(x)",-1,1);
>textcolor(black); // mengatur warna teks menjadi hitam
>title(latex("y=e^x")); // judul di atas plot
>xlabel(latex("x")); // "x" untuk sumbu x
>ylabel(latex("y"),>vertical); // vertical "y" untuk sumbu y
>label(latex("(0,1)),0,1,color=blue): // memberi label sebuah titik
```



Sumbu dapat digambar secara terpisah dengan `xaxis()` dan `yaxis()`.

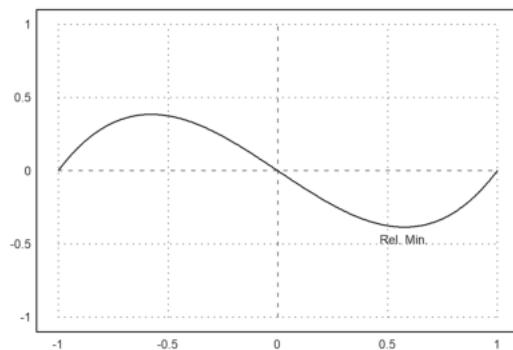
```
>plot2d("x^3-x",<grid,<frame);
>xaxis(0,xx=-2:1,style="->"); yaxis(0,yy=-5:5,style="->");
```



Teks pada plot dapat diatur dengan `label()`. Dalam contoh berikut, "lc" berarti tengah bawah. Ini mengatur posisi label relatif terhadap koordinat plot.

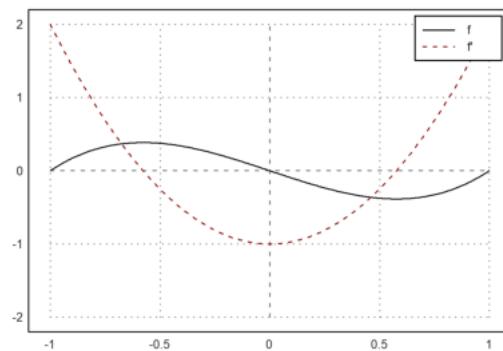
```
>function f(x) &= x^3-x
```

```
>plot2d(f,-1,1,>square);
>x0=fmin(f,0,1); // menghitung titik minimum
>label("Rel. Min.",x0,f(x0),pos="lc"); // tambahkan label disana
```

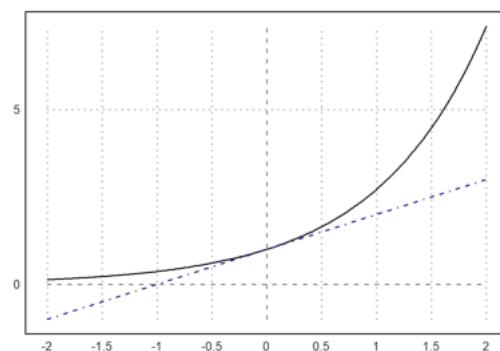


Ada juga kotak teks.

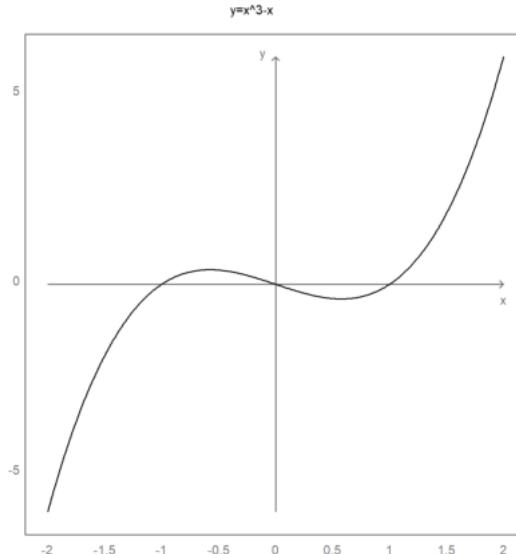
```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // turunan
>labelbox(["f","f'"],["-", "--"],[black,red]): // label box
```



```
>plot2d(["exp(x)","1+x"],color=[black,blue],style=[ "-", "-.-"]):
```



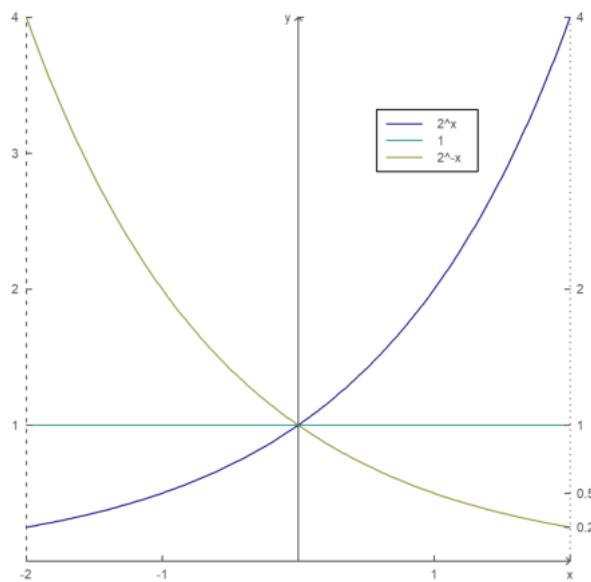
```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray); ...
plot2d("x^3-x",grid=1); ...
settitle("y=x^3-x",color=black); ...
label("x",2,0,pos="bc",color=gray); ...
label("y",0,6,pos="cl",color=gray); ...
reset();
```



Untuk kontrol lebih, sumbu x dan sumbu y dapat dilakukan secara manual.

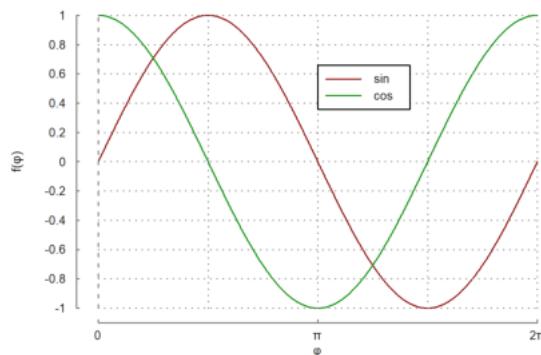
Perintah fullwindow() memperluas jendela plot karena kita tidak lagi memerlukan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk mengatur ulang ke default.

```
>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;
```



Berikut contoh lain, dimana 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", "u"2&#9675;"], style="-", >ticks, >zero); ...
xgrid((0:0.5:2)*pi, <ticks); ...
yaxis(-0.1*pi, -1:0.2:1, style="-", >zero, >grid); ...
labelbox(["sin", "cos"], colors=[red, green], x=0.5, y=0.2, >left); ...
xlabel(u"\u03c6"); ylabel(u"f(\u03c6)");
```



## Memplot 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, `>square` dapat diatur untuk mempertahankan rasio aspek persegi.

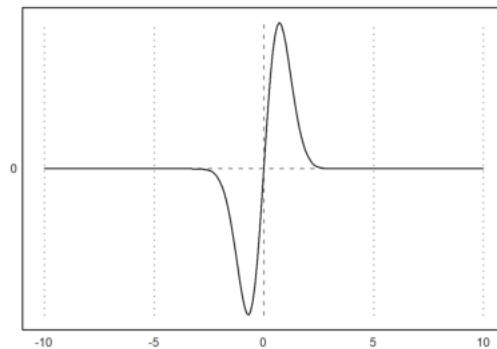
Memplot ekspresi hanyalah singkatan dari plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai- $x$ , dan satu atau beberapa baris nilai- $y$ . Dari rentang dan nilai- $x$ , fungsi `plot2d` akan menghitung data yang akan diplot, secara default dengan evaluasi adaptif fungsi tersebut. Untuk plot titik gunakan "`>points`", untuk garis dan titik campuran gunakan "`>addpoints`".

Namun Anda dapat memasukkan data secara langsung.

- Gunakan vektor baris untuk  $x$  dan  $y$  untuk satu fungsi.
- Matriks untuk  $x$  dan  $y$  diplot baris demi baris.

Berikut adalah contoh dengan satu baris untuk  $x$  dan  $y$ .

```
>x=-10:0.1:10; y=exp(-x^2)*x; plot2d(x,y);
```



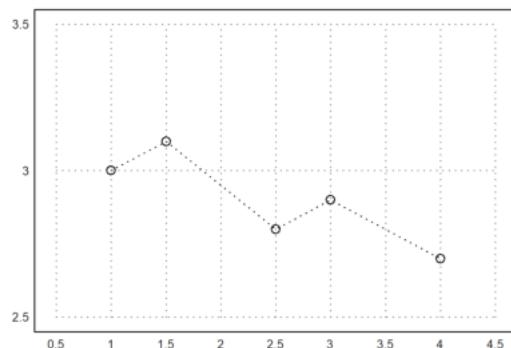
Data juga dapat diplot sebagai titik. Gunakan `points=true` untuk ini. Plot bekerja seperti poligon, tetapi hanya menggambar sudut.

- `style="..."`: Pilih dari "[]", "<>", "o", ".", "..", "+", "\*\*", "[]#", "<>#", "o#", ".#", "#", "|".

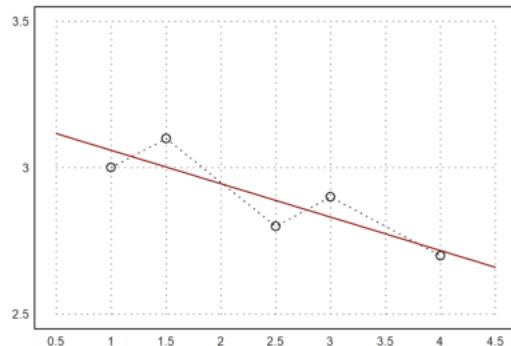
Untuk memplot kumpulan titik, gunakan `>points`. Jika warnanya adalah vektor warna, setiap titik akan mendapatkan warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna berlaku untuk baris matriks.

Parameter `>addpoints` menambahkan titik ke segmen garis untuk plot data.

```
>xdata=[1,1.5,2.5,3,4]; ydata=[3,3.1,2.8,2.9,2.7]; // data
>plot2d(xdata,ydata,a=0.5,b=4.5,c=2.5,d=3.5,style="."); // garis
>plot2d(xdata,ydata,>points,>add,style="o"); // tambahkan poin
```



```
>p=polyfit(xdata,ydata,1); // mendapatkan garis regresi
>plot2d("polyval(p,x)",>add,color=red); // menambahkan plot garis
```



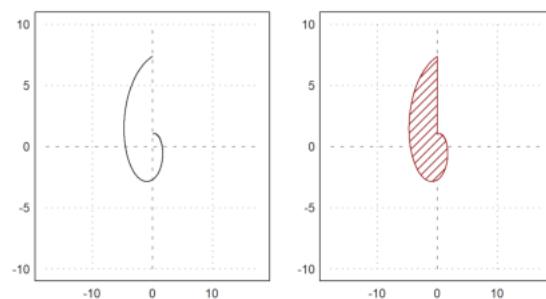
## Menggambar Daerah Yang Dibatasi Kurva

Plot data sebenarnya adalah poligon. Kita juga dapat memplot kurva atau kurva terisi.

- `filled=true` mengisi plot.
- `style="...":` Pilih dari "#", "/", "\", "/".
- `fillcolor:` Lihat di atas untuk warna yang tersedia.

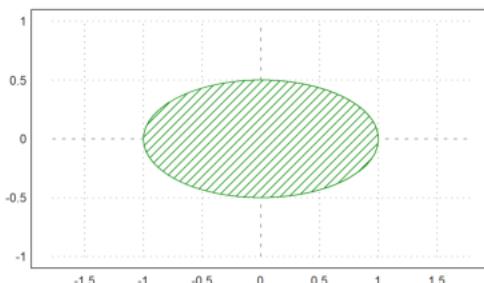
Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional mencegah penggambaran batas untuk semua gaya kecuali yang default.

```
>t=linspace(0,2pi,1000); // parameter untuk kurva
>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 kurva
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // mengisi kurva
>figure(0):
```

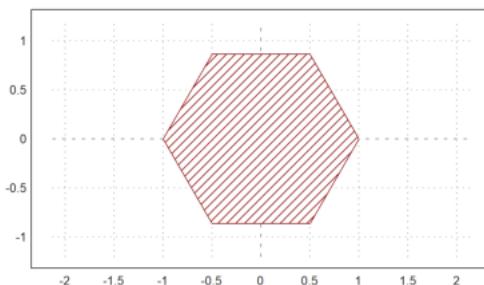


Pada contoh berikut ini kami memplot elips yang terisi dan dua segi enam yang terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian yang berbeda.

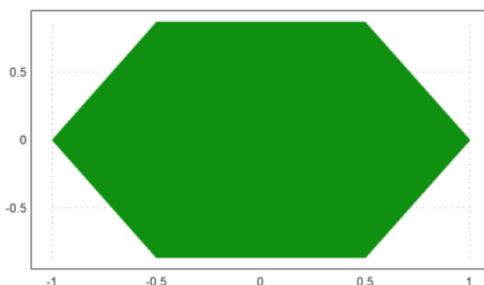
```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/") :
```



```
>t=linspace(0,2pi,6); ...
plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.2):
```

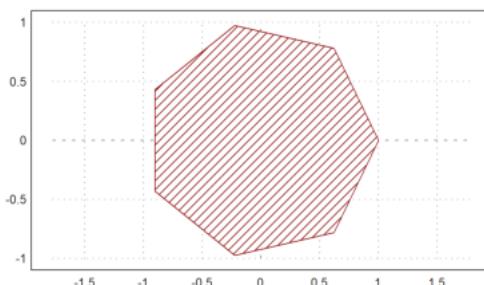


```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```



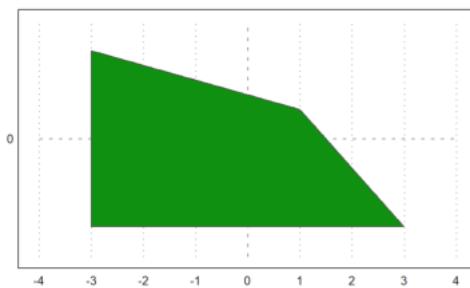
Contoh lain adalah septagon, yang kita buat dengan 7 titik pada lingkaran satuan.

```
>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  $A[k].v \leq 3$  untuk semua baris A. Untuk mendapatkan sudut yang bagus, kita menggunakan n yang relatif besar.

```
>A=[2,1;1,2;-1,0;0,-1];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=4,level=[0;3],color=green,n=111):
```

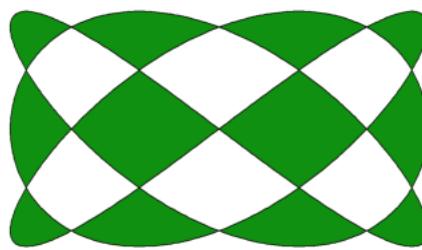


Poin utama dari bahasa matriks adalah memungkinkan pembuatan tabel fungsi dengan mudah.

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);
```

Sekarang kita memiliki vektor nilai x dan y. `plot2d()` dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plot dapat diisi. Dalam kasus ini, hasilnya akan bagus karena aturan lilitan, yang digunakan untuk mengisi.

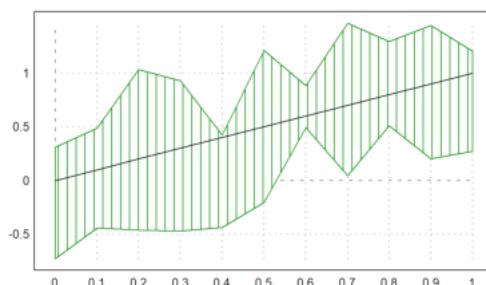
```
>plot2d(x,y,<grid,<frame,>filled):
```



Vektor interval diplot terhadap nilai x sebagai daerah terisi antara nilai interval bawah dan atas.

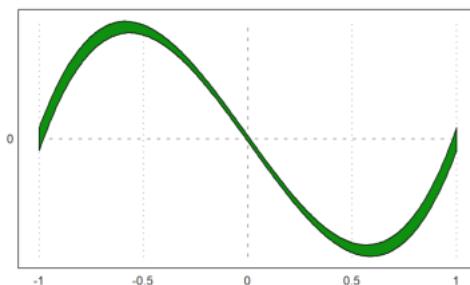
Hal ini dapat berguna untuk memetakan kesalahan perhitungan. Namun, hal ini juga dapat digunakan untuk memetakan kesalahan statistik.

```
>t=0:0.1:1; ...
plot2d(t,interval(t-random(size(t)),t+random(size(t))),style="|");
plot2d(t,t,add=true):
```



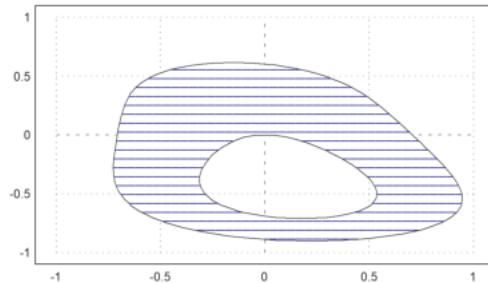
Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka `plot2d` akan memplot rentang interval yang terisi pada bidang, gaya isian sama dengan gaya poligon.

```
>t=-1:0.01:1; x=~t-0.01,t+0.01~;
>plot2d(t,y):
```



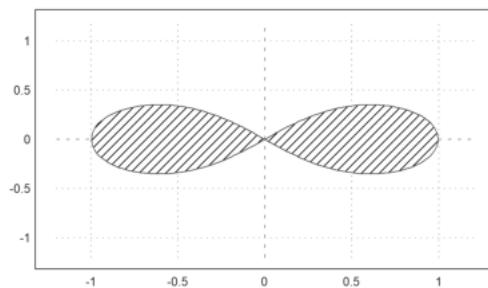
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

```
>expr := "2*x^2+x*y+3*y^4+y"; // mendefinisikan sebuah ekspresi f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue); // 0 <= f(x,y) <= 1
```

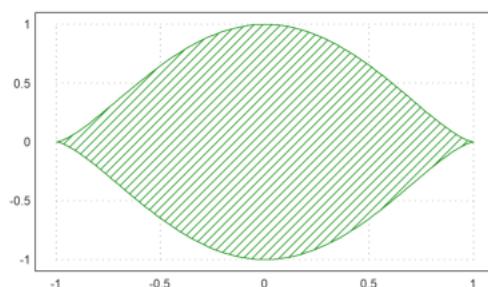


Kita juga dapat mengisi rentang nilai seperti

```
>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/"):
```



```
>plot2d("cos(x)","sin(x)^3",xmin=0,xmax=2pi,>filled,style="/"):
```



## Grafik Fungsi Parametrik

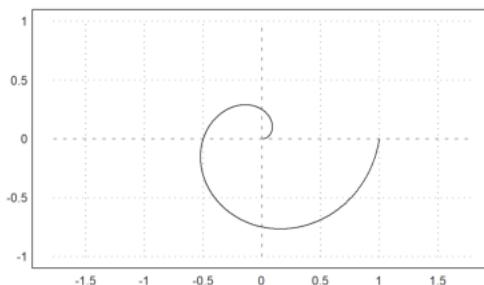
Nilai x tidak perlu diurutkan. (x,y) hanya menggambarkan sebuah kurva. Jika x diurutkan, kurva tersebut adalah grafik fungsi.

Pada contoh berikut, kita memplot spiral

$$\gamma(t) = t \cdot (\cos(2\pi t), \sin(2\pi t))$$

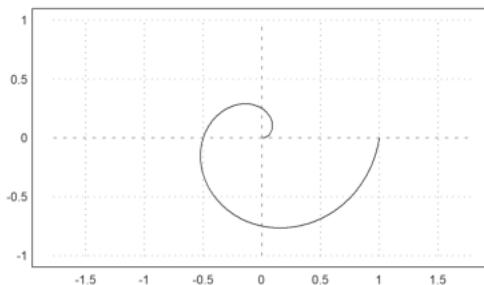
Kita mungkin perlu menggunakan sangat banyak titik untuk tampilan yang halus atau fungsi adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk lebih jelasnya).

```
>t=linspace(0,1,1000); ...
plot2d(t*cos(2*pi*t),t*sin(2*pi*t),r=1):
```

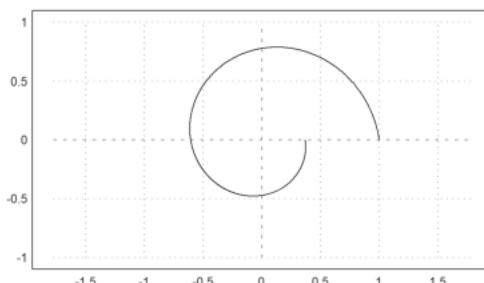


Sebagai alternatif, Anda dapat menggunakan dua ekspresi untuk kurva. Berikut ini memplot kurva yang sama seperti di atas.

```
>plot2d("x*cos(2*pi*x)", "x*sin(2*pi*x)", xmin=0, xmax=1, r=1):
```



```
>t=linspace(0,1,1000); r=exp(-t); x=r*cos(2pi*t); y=r*sin(2pi*t);
>plot2d(x,y,r=1):
```



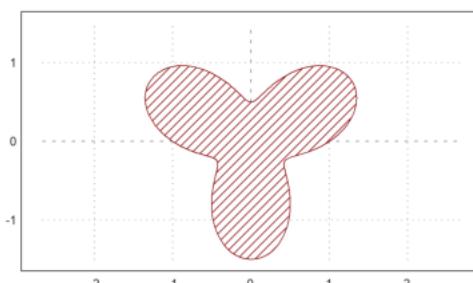
Pada contoh berikutnya, kita memplot kurva

$$\gamma(t) = (r(t) \cos(t), r(t) \sin(t))$$

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}.$$

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5):
```



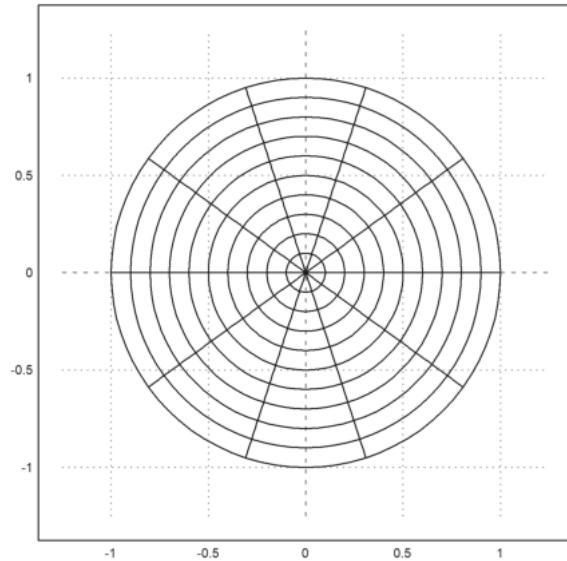
## Menggambar Grafik Bilangan Kompleks

Sebuah deretan bilangan kompleks juga dapat diplot. Kemudian titik-titik kisi akan dihubungkan. Jika sejumlah garis kisi ditentukan (atau vektor 1x2 garis kisi) pada argumen cgrid, hanya garis-garis kisi tersebut yang akan terlihat.

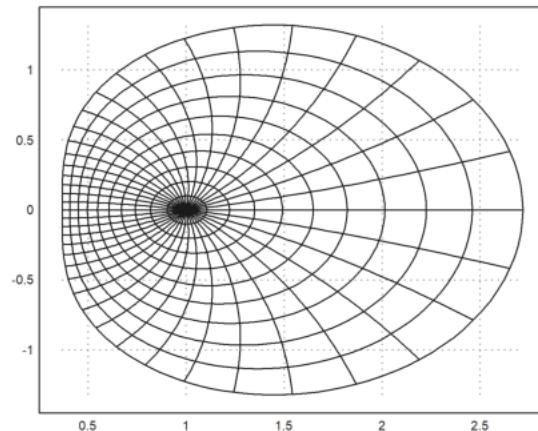
Matriks bilangan kompleks akan secara otomatis diplot sebagai sebuah grid pada bidang kompleks.

Pada contoh berikut, kita memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

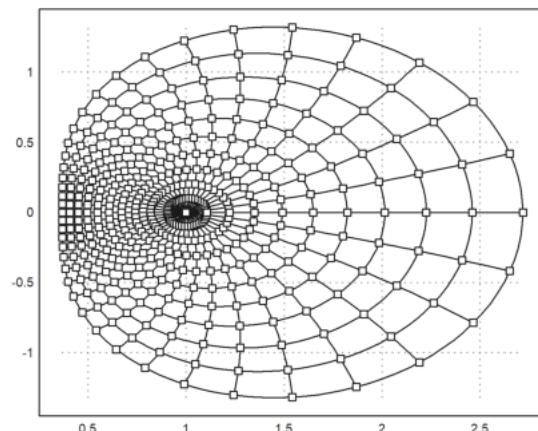
```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```



```
>aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=r*exp(I*a);
>plot2d(exp(z),cgrid=[40,10]):
```



```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```

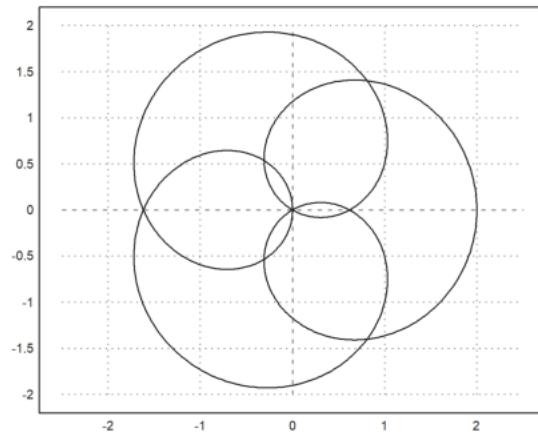


Vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian nyata dan bagian imajiner.

Pada contoh, kami memplot lingkaran satuan dengan

$$\gamma(t) = e^{it}$$

```
>t=linspace(0,2pi,1000); ...
plot2d(exp(I*t)+exp(4*I*t),z=2):
```



## Plot Statistik

Terdapat banyak fungsi yang dikhkusukan untuk plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

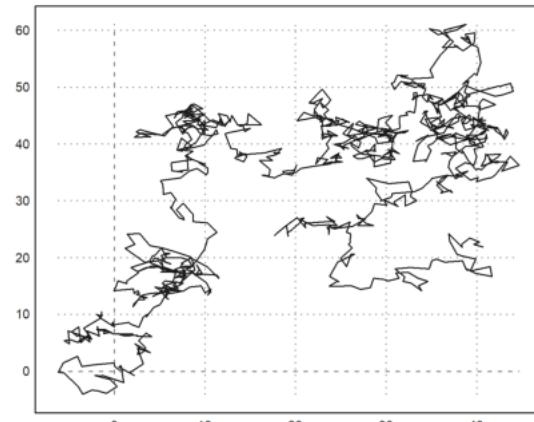
Jumlah kumulatif dari nilai berdistribusi normal 0-1 menghasilkan jalan acak.

```
>plot2d(cumsum(randnormal(1,1000))):
```

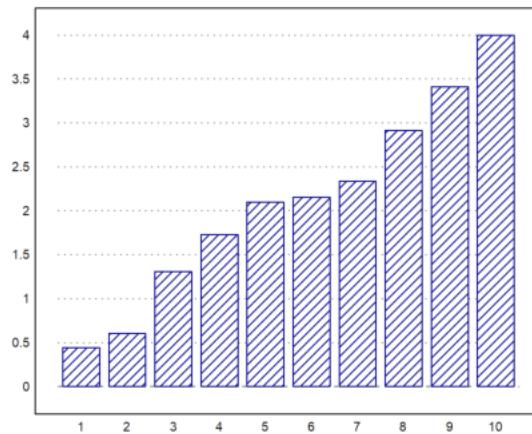


Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

```
>X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
```

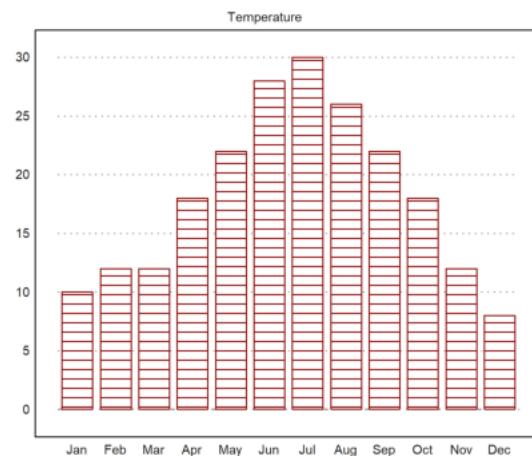


```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

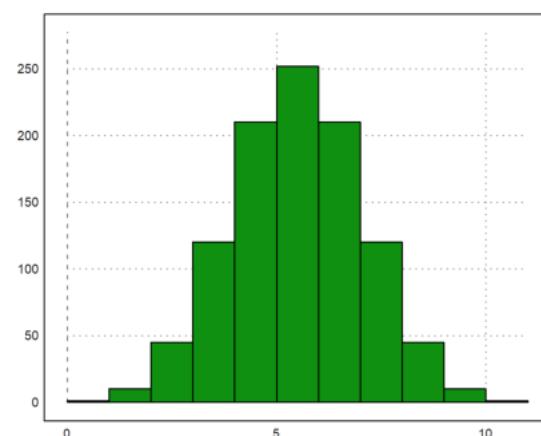


Ini juga dapat menampilkan string sebagai label.

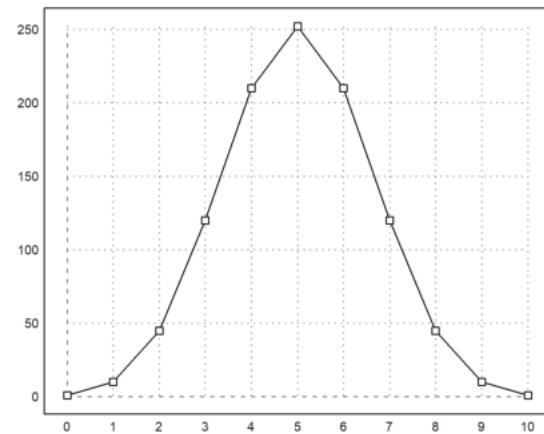
```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
"Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```



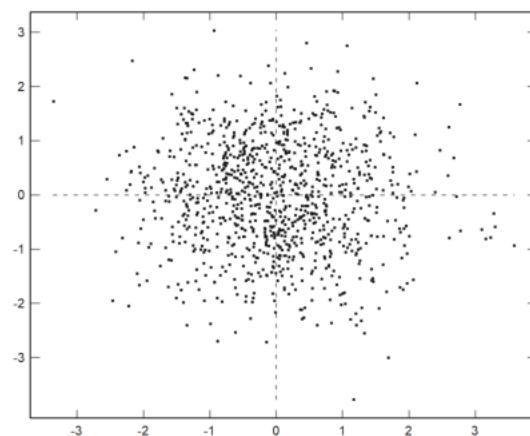
```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```



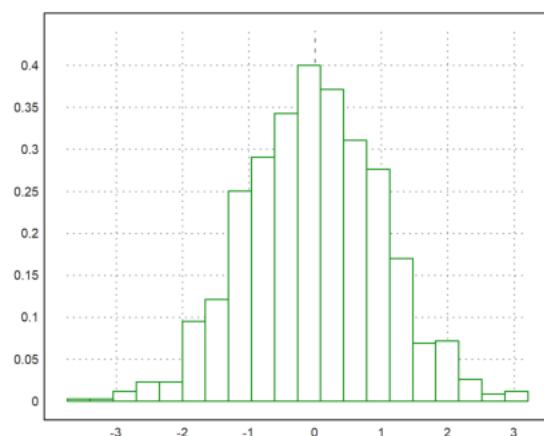
```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```



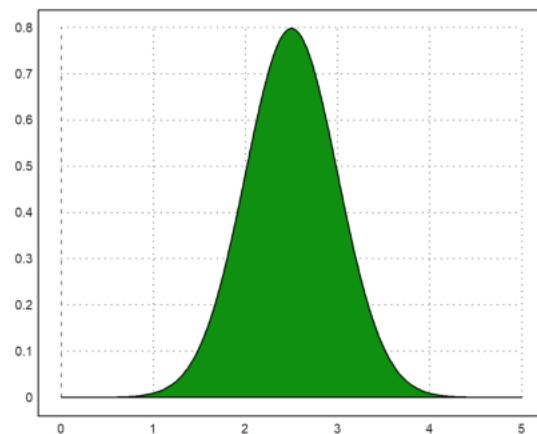
```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```



```
>plot2d(normal(1,1000),>distribution,style="O"):
```

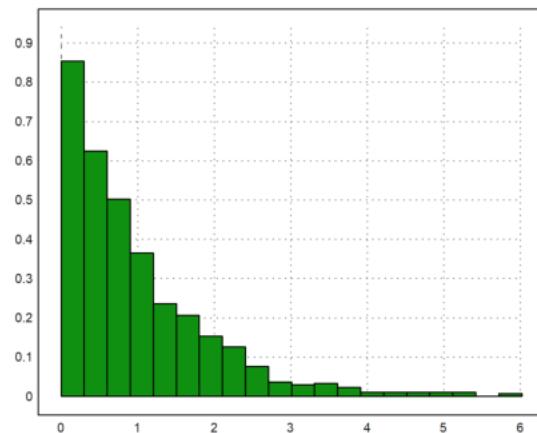


```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```



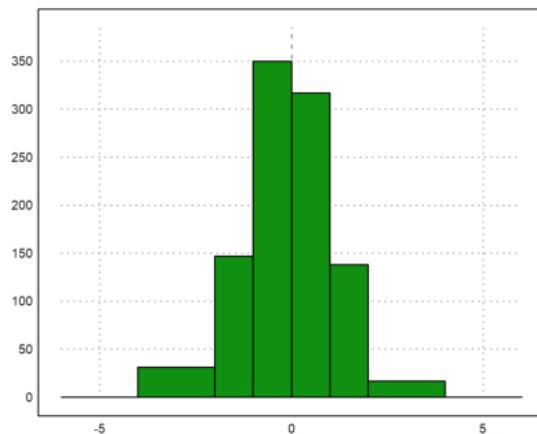
Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan `distribution=n` dengan `plot2d`.

```
>w=randexponential(1,1000); // exponential distribution
>plot2d(w,>distribution): // or distribution=n with n intervals
```



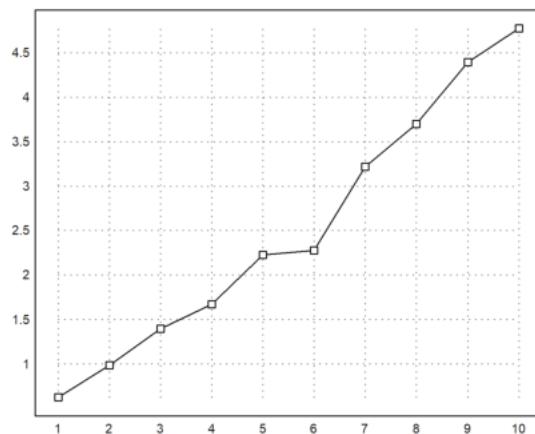
Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan `>bar` di `plot3d`, atau dengan `plot kolom`.

```
>w=normal(1000); // 0-1-normal distribution
>{x,y}=histo(w,10,v=[-6,-4,-2,-1,0,1,2,4,6]); // batas-batas interval v
>plot2d(x,y,>bar):
```

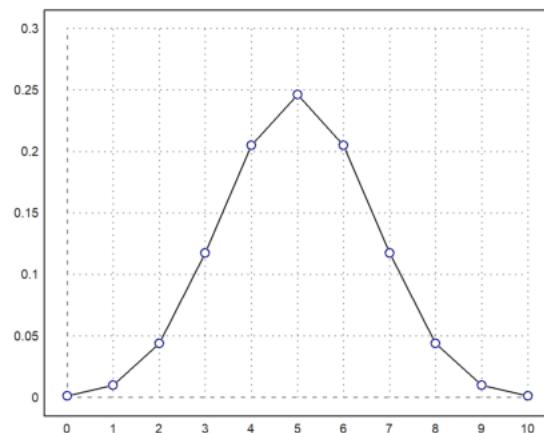


Fungsi `statplot()` mengatur gaya dengan string sederhana.

```
>statplot(1:10,cumsum(random(10)), "b"):
```



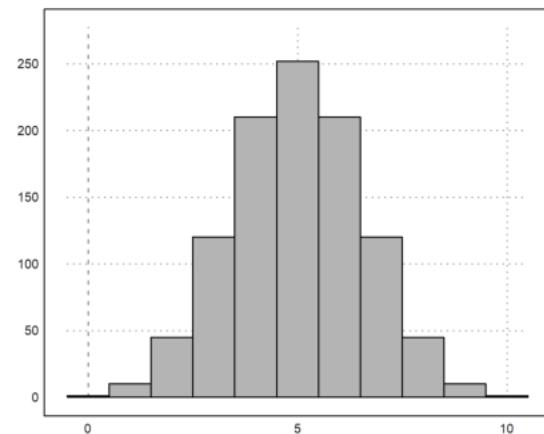
```
>n=10; i=0:n; ...
plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```



Selain itu, data dapat diplot sebagai batang. Dalam hal ini, x harus diurutkan dan satu elemen lebih panjang dari y. Batang akan memanjang dari  $x[i]$  ke  $x[i+1]$  dengan nilai  $y[i]$ . Jika x memiliki ukuran yang sama dengan y, maka x akan diperpanjang satu elemen dengan jarak terakhir.

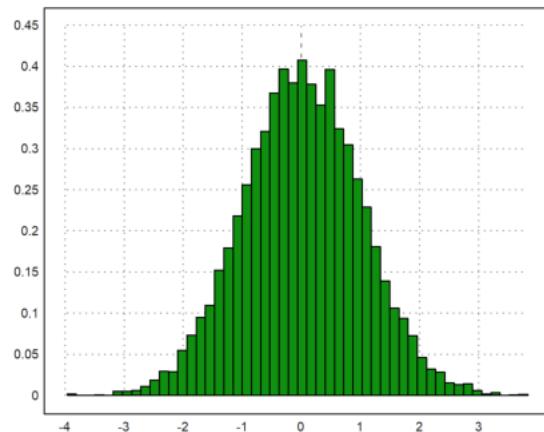
Gaya isian dapat digunakan seperti di atas.

```
>n=10; k=bin(n,0:n); ...
plot2d(-0.5:n+0.5,k,bar=true,fillcolor=lightgray):
```

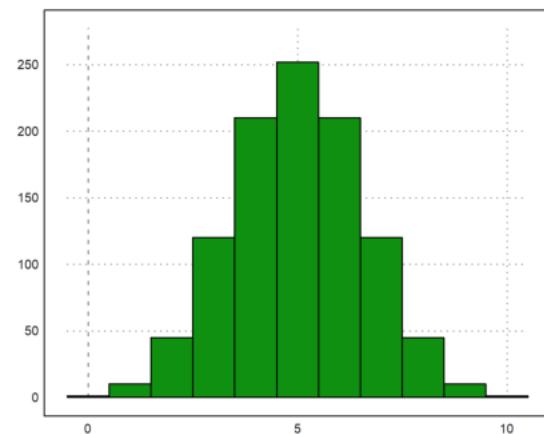


Data untuk plot batang (bar=1) dan histogram (histogram = 1) dapat diberikan secara eksplisit dalam xv dan yy, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribusi (atau distribusi = n). Histogram dari nilai xv akan dihitung secara otomatis dengan >histogram. Jika >even ditentukan, nilai xv akan dihitung dalam interval bilangan bulat.

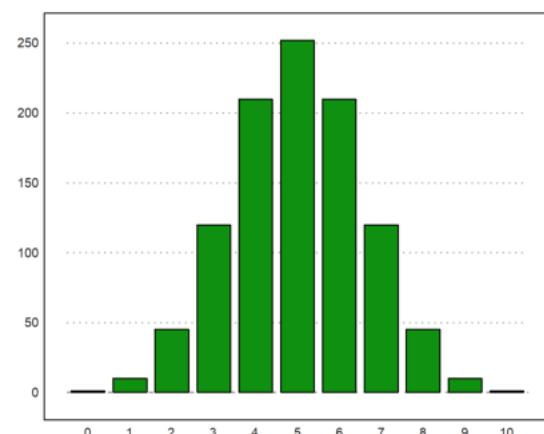
```
>plot2d(normal(10000),distribution=50):
```



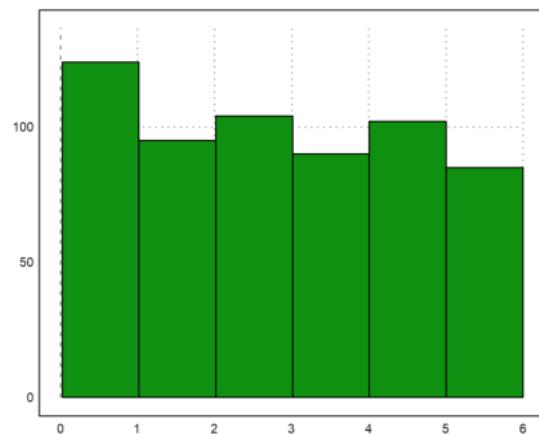
```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```



```
>columnsplot(m,k):
```

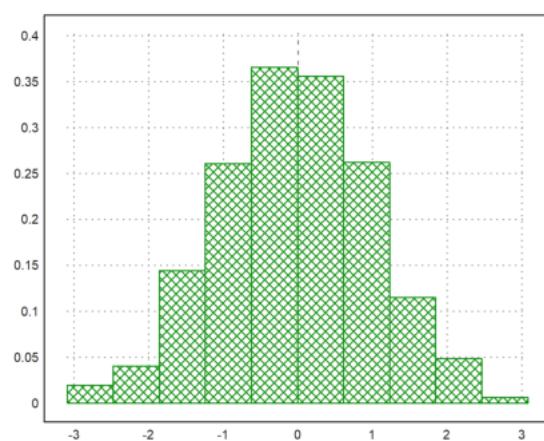


```
>plot2d(random(600)*6,histogram=6):
```



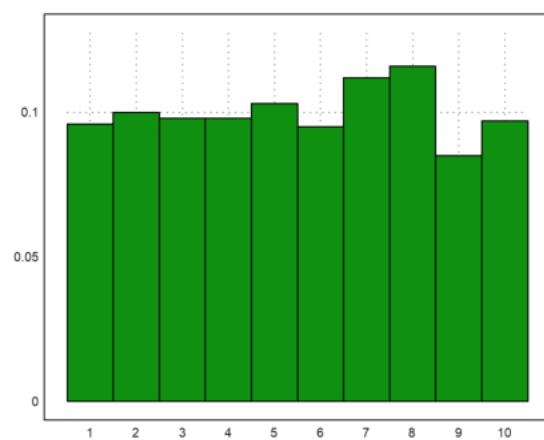
Untuk distribusi, ada parameter `distribution=n`, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan  $n$  sub-interval.

```
>plot2d(normal(1,1000),distribution=10,style="\."):
```



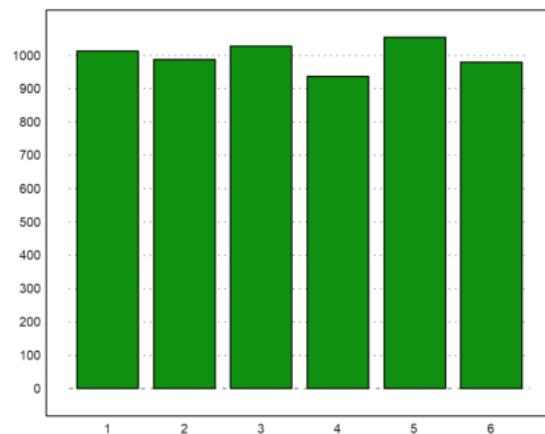
Dengan parameter `even=true`, ini akan menggunakan interval integer.

```
>plot2d(intrandom(1,1000,10),distribution=10,even=true):
```

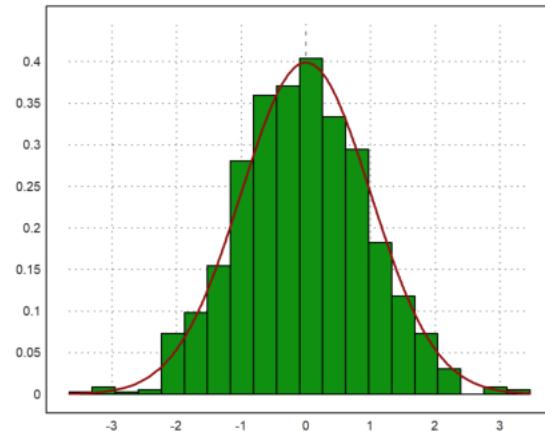


Perhatikan bahwa ada banyak plot statistik yang mungkin berguna. Lihatlah tutorial tentang statistik.

```
>columnsplot(getmultiplicities(1:6,intrandom(1,6000,6)):
```

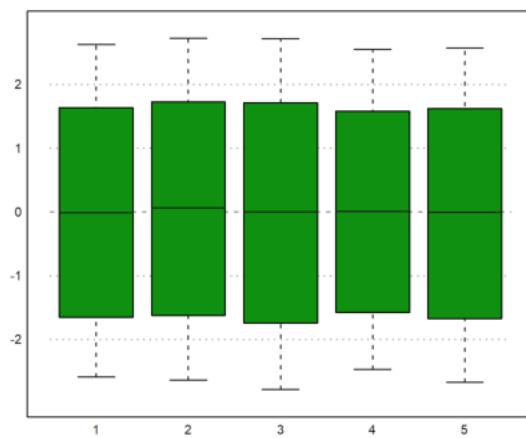


```
>plot2d(normal(1,1000),>distribution); ...
plot2d("qnormal(x)",color=red,thickness=2,>add):
```



Ada juga banyak plot khusus untuk statistik. Boxplot menunjukkan kuartil dari distribusi ini dan banyak pencilan. Menurut definisi, pencilan dalam boxplot adalah data yang melebihi 1,5 kali kisaran 50% tengah plot.

```
>M=normal(5,1000); boxplot(quartiles(M));
```



## Fungsi Implisit

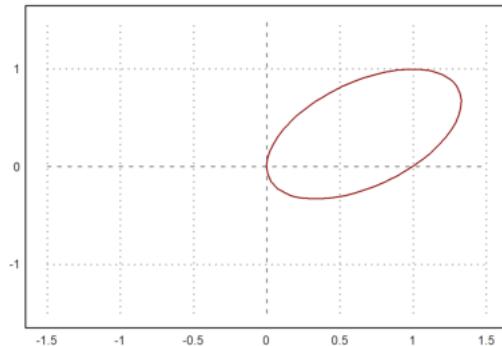
Plot implisit menunjukkan garis level yang menyelesaikan  $f(x,y)=\text{level}$ , di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika level = "auto", akan ada nc garis level, yang akan menyebar di antara nilai minimum dan maksimum fungsi secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan >hue untuk mengindikasikan nilai fungsi. Untuk fungsi implisit, xv haruslah sebuah fungsi atau ekspresi dari parameter x dan y, atau, sebagai alternatif, xv dapat berupa sebuah matriks nilai.

Euler dapat menandai garis level

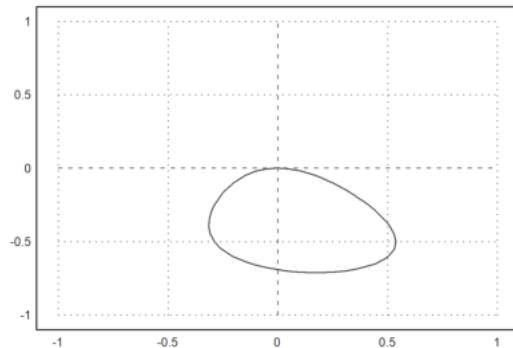
dari fungsi apa pun.

Untuk menggambar himpunan  $f(x,y)=c$  untuk satu atau lebih konstanta  $c$ , Anda dapat menggunakan `plot2d()` dengan plot implisitnya pada bidang. Parameter untuk  $c$  adalah  $\text{level} = c$ , di mana  $c$  dapat berupa vektor garis level. Sebagai tambahan, sebuah skema warna dapat digambar pada latar belakang untuk mengindikasikan nilai fungsi untuk setiap titik pada plot. Parameter "n" menentukan kehalusan plot.

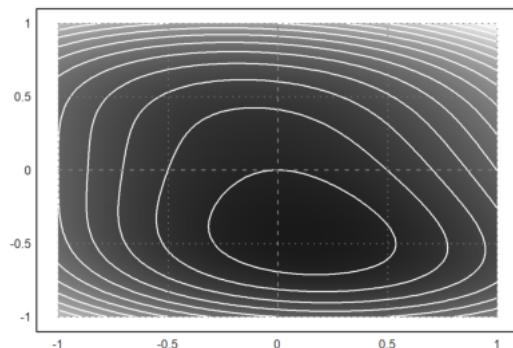
```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```



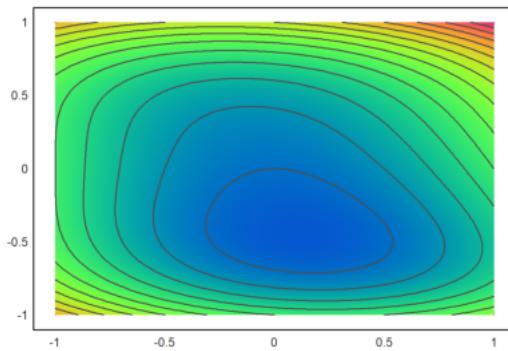
```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0); // Solutions of f(x,y)=0
```



```
>plot2d(expr,level=0:0.5:20,>hue,contourcolor=white,n=200); // nice
```

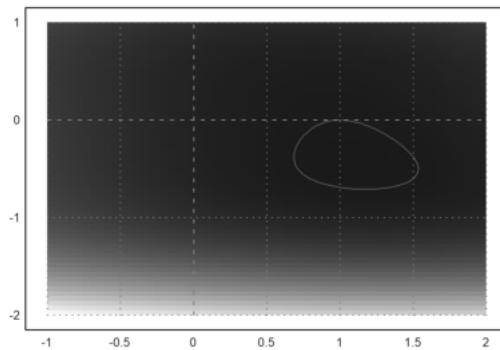


```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4); // nicer
```

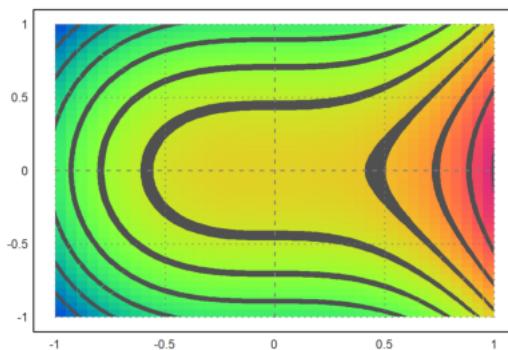


Hal ini juga berlaku untuk plot data. Tetapi Anda harus menentukan rentang untuk label sumbu.

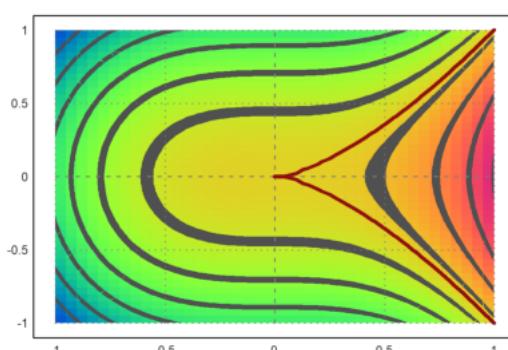
```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```



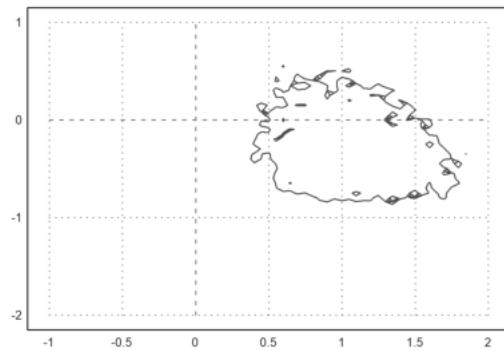
```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```



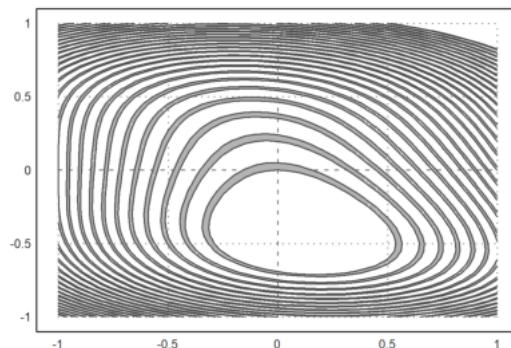
```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```



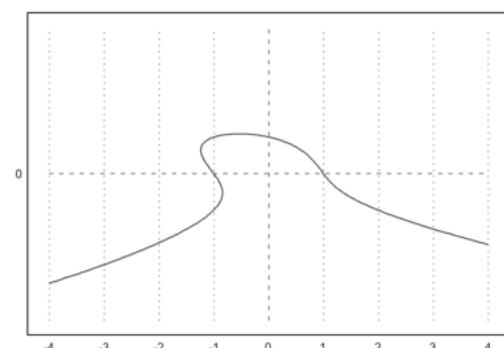
```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```



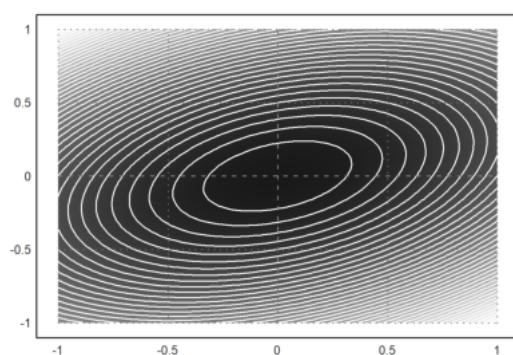
```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```



```
>plot2d("x^2+y^3+x*y",level=1,r=4,n=100):
```



```
>plot2d("x^2+2*y^2-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):
```



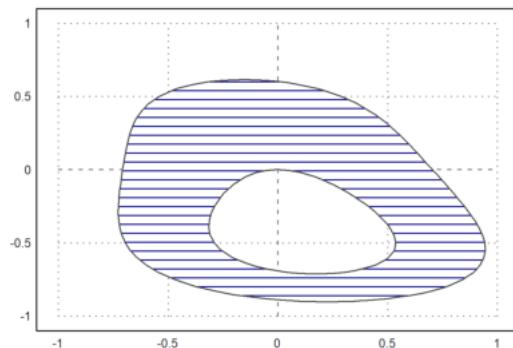
Dimungkinkan juga untuk mengisi set

$$a \leq f(x, y) \leq b$$

dengan rentang level.

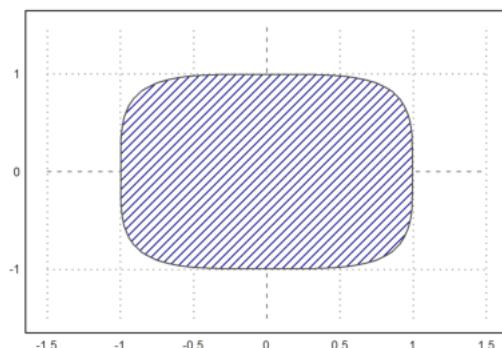
Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

```
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```

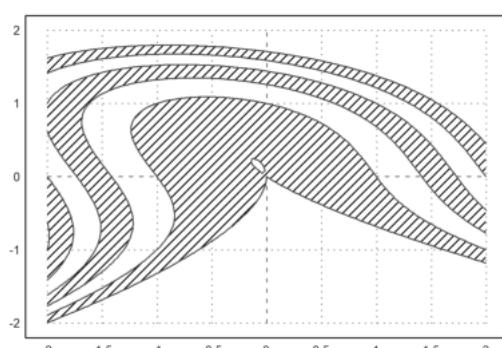


Plot implisit juga dapat menunjukkan rentang level. Maka level harus berupa matriks 2xn interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Sebagai alternatif, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level ke interval.

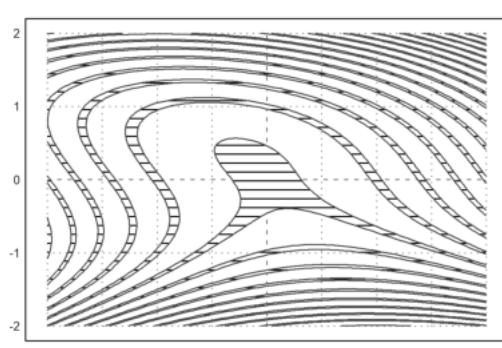
```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```



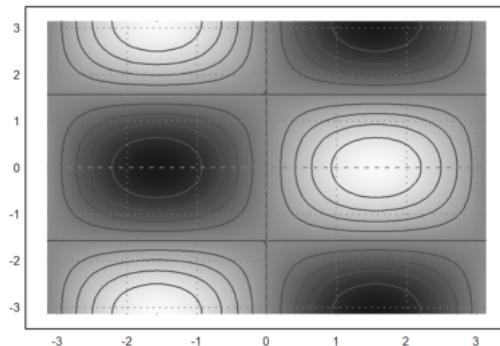
```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```



```
>plot2d("x^2+y^3+x*y",level=-10:20,r=2,style="-",dl=0.1,n=100):
```



```
>plot2d("sin(x)*cos(y)",r=pi,>hue,>levels,n=100):
```

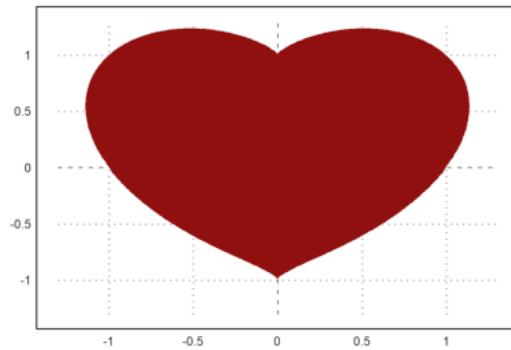


Anda juga dapat menandai suatu wilayah

$$a \leq f(x,y) \leq b.$$

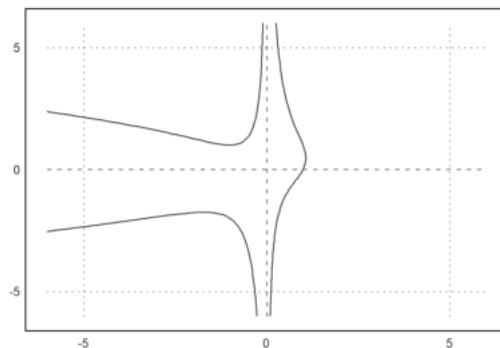
Hal ini dilakukan dengan menambahkan level dengan dua baris.

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
style="#",color=red,<outline, ...
level=[-2;0],n=100):
```



Dimungkinkan untuk menentukan level tertentu. Sebagai contoh, kita dapat memplot solusi dari persamaan seperti

```
>plot2d("x^3-x*y+x^2*y^2",r=6,level=1,n=100):
```



```
>function starplot1 (v, style="/", color=green, lab=none) ...
if !holding() then clg; endif;
w=window(); window(0,0,1024,1024);
h=holding(1);
r=max(abs(v))*1.2;
setplot(-r,r,-r,r);
n=cols(v); t=linspace(0,2pi,n);
v=v|v[1]; c=v*cos(t); s=v*sin(t);
cl=barcolor(color); st=barstyle(style);
loop 1 to n
  polygon([0,c[#],c[#+1]], [0,s[#],s[#+1]],1);
  if lab!=none then
    rlab=v[#]+r*0.1;
    {col,row}=toscreen(cos(t[#])*rlab,sin(t[#])*rlab);
```

```

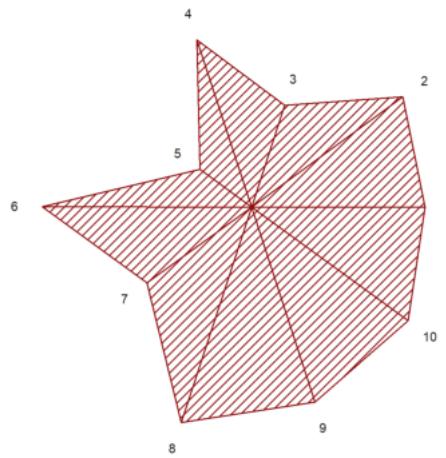
        ctext(""+lab[#],col,row-textheight()/2);
        endif;
end;
barcolor(cl); barstyle(st);
holding(h);
window(w);
endfunction

```

Tidak ada kisi-kisi atau kutu sumbu di sini. Selain itu, kami menggunakan jendela penuh untuk plot.

Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Hal ini tidak perlu dilakukan, 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.

Pada fungsi berikut ini, kita akan membuat plot impuls logaritmik. plot2d dapat membuat plot logaritmik, tetapi tidak untuk batang impuls.

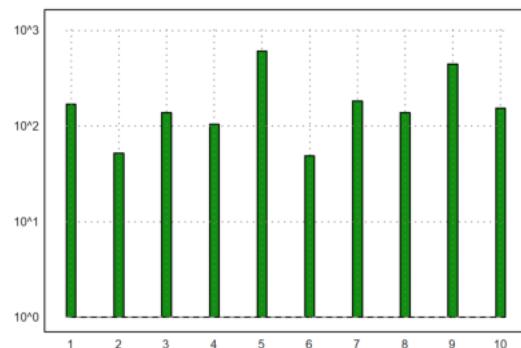
```

>function logimpulseplot1 (x,y) ...
{x0,y0)=makeimpulse(x,log(y)/log(10));
plot2d(x0,y0,>bar,grid=0);
h=holding(1);
frame();
xgrid(ticks(x));
p=plot();
for i=-10 to 10;
  if i<=p[4] and i>=p[3] then
    ygrid(i,yt="10^"+i);
  endif;
end;
holding(h);
endfunction

```

Mari kita uji dengan nilai yang terdistribusi secara eksponensial.

```
>aspect(1.5); x=1:10; y=-log(random(size(x)))*200; ...
logimpulseplot1(x,y):
```



Mari kita menghidupkan kurva 2D dengan menggunakan plot langsung. Perintah plot(x,y) hanya memplot kurva ke dalam jendela plot. setplot(a,b,c,d) mengatur jendela ini.

Fungsi wait(0) memaksa plot untuk muncul pada 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
```

Tekan sembarang tombol untuk menghentikan animasi ini.

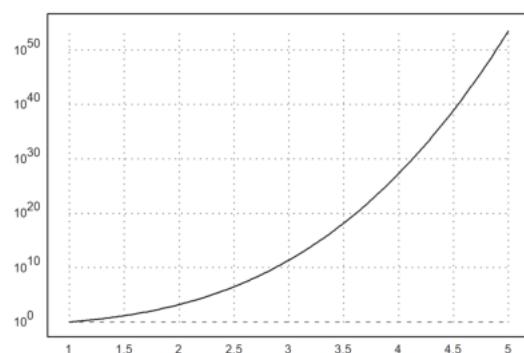
```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

## Logarithmic Plots

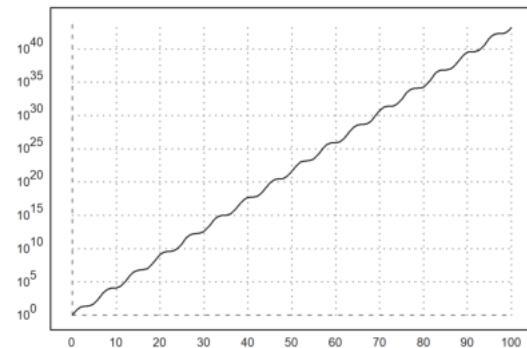
EMT menggunakan parameter "logplot" untuk skala logaritmik. Plot logaritmik dapat diplot menggunakan skala logaritmik dalam y dengan logplot = 1, atau menggunakan skala logaritmik dalam x dan y dengan logplot = 2, atau dalam x dengan logplot = 3.

- logplot=1: y-logarithmic
- logplot=2: x-y-logarithmic
- logplot=3: x-logarithmic

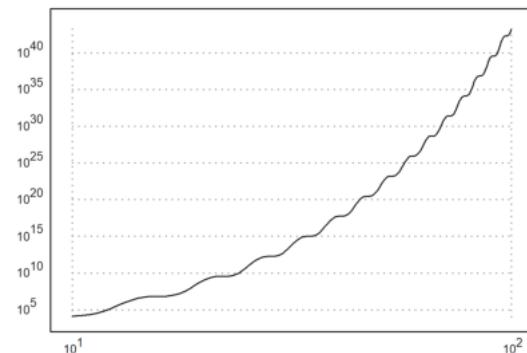
```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```



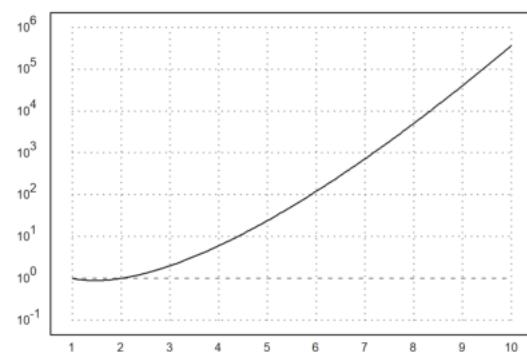
```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```



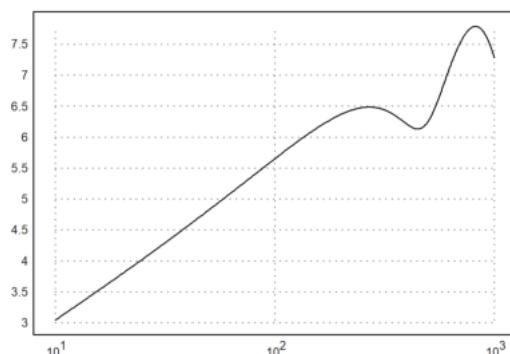
```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```



```
>plot2d("gamma(x)",1,10,logplot=1):
```

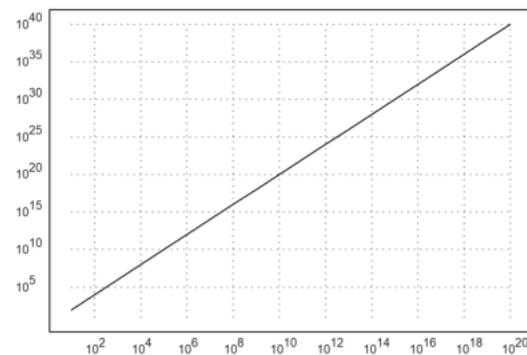


```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```



Hal ini juga bisa dilakukan dengan plot data.

```
>x=10^(1:20); y=x^2-x;
>plot2d(x,y,logplot=2):
```

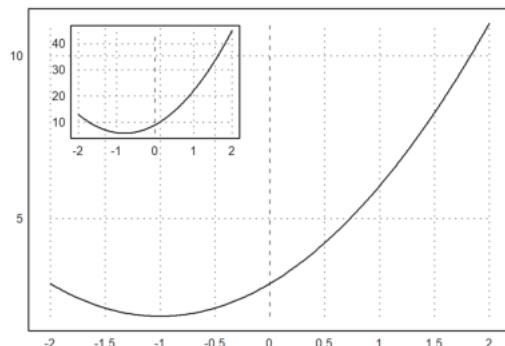
**Contoh Soal dan Pembahasan**

- 1). Gambarkan 2 grafik fungsi dibawah ini dalam 1 jendela plot yang berbeda ukuran

$$y = x^2 + 2x + 3$$

$$y = 5x^2 + 8x + 9$$

```
>plot2d("x^2+2x+3";
>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("5x^2+8x+9":
```

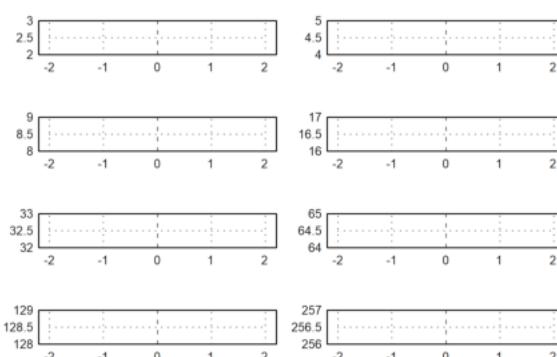


```
>hold off;
>window(ow);
```

- 2).Buat grafik dari

$$2^x, 1 \leq x \leq 9$$

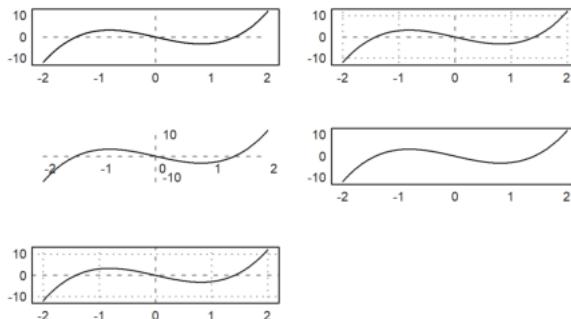
```
>figure(4,2); ...
for x=1 to 8; figure(x); plot2d("2^"+x); end; ...
figure(0):
```



- 3).Buatlah dengan k dari 1 sampai 5, dimana k untuk menunjukkan bentuk grid yang dihasilkan dari fungsi

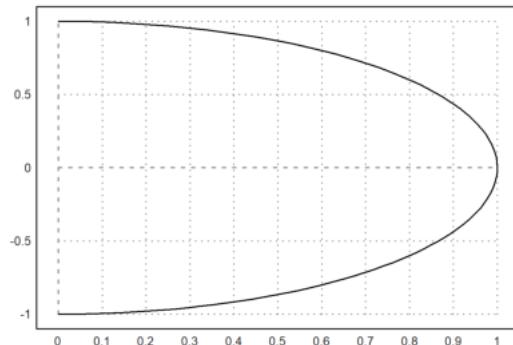
```
>reset; ...
figure(5,2); ...
```

```
for k=1 to 5; figure(k); plot2d("3x^3-6x",grid=k); end; ...
figure(0):
```



4).Buatlah grafik plot 2d dengan fungsi  $\sin(x)$  dan fungsi  $\cos(x)$ , jika  $x_{\min} = 0$  dan  $x_{\max} = \pi$

```
>reset; ...
aspect(1.5); plot2d("sin(x)","cos(x)",xmin=0,xmax=pi):
```



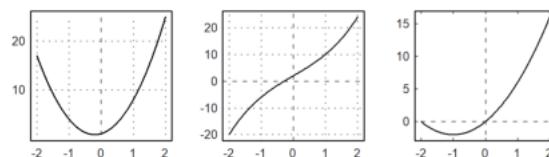
5).Gambarkan grafik berikut:

$$5x^2 + 2x + 1$$

$$x^3 + 7x + 2$$

$$2x^2 + 4x$$

```
>reset; ...
aspect(3,1); ...
figure(1,3); ...
figure(1); plot2d("5x^2+2x+1", grid=2); ...
figure(2); plot2d("x^3+7x+2", grid=5); ...
figure(3); plot2d("2x^2+4x", grid=6); ...
figure(0):
```



## Rujukan Lengkap Fungsi plot2d()

```
function plot2d (xv, yv, btest, a, b, c, d, xmin, xmax, r, n, ...
logplot, grid, frame, framecolor, square, color, thickness, style, ...
auto, add, user, delta, points, addpoints, pointstyle, bar, histogram, ...
distribution, even, steps, own, adaptive, hue, level, contour, ...
nc, filled, fillcolor, outline, title, xl, yl, maps, contourcolor, ...
```

```
contourwidth, ticks, margin, clipping, cx, cy, insimg, spectral, ..
cgrid, vertical, smaller, dl, niveau, levels)
```

Multipurpose plot function for plots in the plane (2D plots). This function can do plots of functions of one variables, data plots, curves in the plane, bar plots, grids of complex numbers, and implicit plots of functions of two variables.

#### Parameters

x,y : equations, functions or data vectors  
a,b,c,d : Plot area (default a=-2,b=2)  
r : if r is set, then a=cx-r, b=cx+r, c=cy-r, d=cy+r  
r can be a vector [rx,ry] or a vector [rx1,rx2,ry1,ry2].  
xmin,xmax : range of the parameter for curves  
auto : Determine y-range automatically (default)  
square : if true, try to keep square x-y-ranges  
n : number of intervals (default is adaptive)  
grid : 0 = no grid and labels,  
1 = axis only,  
2 = normal grid (see below for the number of grid lines)  
3 = inside axis  
4 = no grid  
5 = full grid including margin  
6 = ticks at the frame  
7 = axis only  
8 = axis only, sub-ticks  
frame : 0 = no frame  
framecolor: color of the frame and the grid  
margin : number between 0 and 0.4 for the margin around the plot  
color : Color of curves. If this is a vector of colors,  
it will be used for each row of a matrix of plots. In the case of  
point plots, it should be a column vector. If a row vector or a  
full matrix of colors is used for point plots, it will be used for  
each data point.  
thickness : line thickness for curves  
This value can be smaller than 1 for very thin lines.  
style : Plot style for lines, markers, and fills.  
For points use  
"[]", "<>", ".", "..", "...",  
"\*", "+", "|", "-", "o"  
"[]#", "<>#", "o#" (filled shapes)  
"[]w", "<>w", "ow" (non-transparent)  
For lines use  
"-", "--", "-.", "-.", "-.-", "->"  
For filled polygons or bar plots use  
"#", "#O", "O", "/", "\", "V",  
"+", "|", "-", "t"  
points : plot single points instead of line segments  
addpoints : if true, plots line segments and points  
add : add the plot to the existing plot  
user : enable user interaction for functions  
delta : step size for user interaction  
bar : bar plot (x are the interval bounds, y the interval values)  
histogram : plots the frequencies of x in n subintervals  
distribution=n : plots the distribution of x with n subintervals  
even : use inter values for automatic histograms.  
steps : plots the function as a step function (steps=1,2)  
adaptive : use adaptive plots (n is the minimal number of steps)  
level : plot level lines of an implicit function of two variables  
outline : draws boundary of level ranges.

If the level value is a 2xn matrix, ranges of levels will be drawn  
in the color using the given fill style. If outline is true, it  
will be drawn in the contour color. Using this feature, regions of  
f(x,y) between limits can be marked.

hue : add hue color to the level plot to indicate the function  
value  
contour : Use level plot with automatic levels  
nc : number of automatic level lines  
title : plot title (default "")  
xl, yl : labels for the x- and y-axis  
smaller : if >0, there will be more space to the left for labels.  
vertical :  
Turns vertical labels on or off. This changes the global variable  
verticallabels locally for one plot. The value 1 sets only vertical  
text, the value 2 uses vertical numerical labels on the y axis.  
filled : fill the plot of a curve  
fillcolor : fill color for bar and filled curves

outline : boundary for filled polygons  
 logplot : set logarithmic plots  
 1 = logplot in y,  
 2 = logplot in xy,  
 3 = logplot in x  
 own :  
 A string, which points to an own plot routine. With >user, you get  
 the same user interaction as in plot2d. The range will be set  
 before each call to your function.  
 maps : map expressions (0 is faster), functions are always mapped.  
 contourcolor : color of contour lines  
 contourwidth : width of contour lines  
 clipping : toggles the clipping (default is true)  
 title :  
 This can be used to describe the plot. The title will appear above  
 the plot. Moreover, a label for the x and y axis can be added with  
 xl="string" or yl="string". Other labels can be added with the  
 functions label() or labelbox(). The title can be a unicode  
 string or an image of a Latex formula.  
 cgrid :  
 Determines the number of grid lines for plots of complex grids.  
 Should be a divisor of the the matrix size minus 1 (number of  
 subintervals). cgrid can be a vector [cx,cy].

#### Overview

##### The function can plot

- expressions, call collections or functions of one variable,
- parametric curves,
- x data against y data,
- implicit functions,
- bar plots,
- complex grids,
- polygons.

If a function or expression for xv is given, plot2d() will compute values in the given range using the function or expression. The expression must be an expression in the variable x. The range must be defined in the parameters a and b unless the default range [-2,2] should be used. The y-range will be computed automatically, unless c and d are specified, or a radius r, which yields the range [-r,r] for x and y. For plots of functions, plot2d will use an adaptive evaluation of the function by default. To speed up the plot for complicated functions, switch this off with <adaptive, and optionally decrease the number of intervals n. Moreover, plot2d() will by default use mapping. I.e., it will compute the plot element for element. If your expression or your functions can handle a vector x, you can switch that off with <maps for faster evaluation.

Note that adaptive plots are always computed element for element. If functions or expressions for both xv and for yv are specified, plot2d() will compute a curve with the xv values as x-coordinates and the yv values as y-coordinates. In this case, a range should be defined for the parameter using xmin, xmax. Expressions contained

## Menggambar Plot 3D dengan EMT

---

Ini adalah pengenalan plot 3D di Euler. Kita memerlukan plot 3D untuk memvisualisasikan fungsi dua variabel.

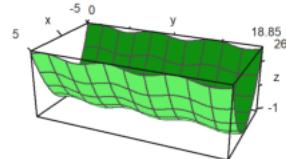
Euler menggambar fungsi tersebut menggunakan algoritma pengurutan untuk menyembunyikan bagian-bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Standarnya adalah dari kuadran x-y positif ke arah titik asal  $x=y=z=0$ , tetapi sudut=0° terlihat dari arah sumbu y. Sudut pandang dan ketinggian dapat diubah.

Euler dapat memetakan

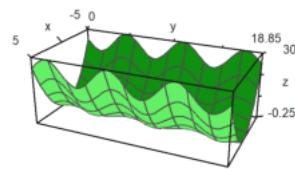
- permukaan dengan bayangan dan garis level atau rentang level,
- awan titik-titik,
- kurva parametrik,
- permukaan implisit.

Plot 3D dari sebuah fungsi menggunakan plot3d. Cara termudah adalah memplot ekspresi dalam x dan y. Parameter r mengatur rentang plot di sekitar (0,0).

```
>aspect(1.5); plot3d("x^2+sin(y)", -5, 5, 0, 6*pi);
```



```
>plot3d("x^2+x*sin(y)", -5, 5, 0, 6*pi):
```



Silakan lakukan modifikasi agar gambar "talang bergelombang" tersebut tidak lurus melainkan melengkung/melingkar, baik melingkar secara mendatar maupun melingkar turun/naik (seperti papan peluncur pada kolam renang). Temukan rumusnya.

## Fungsi dari dua Variabel

Untuk grafik sebuah fungsi, gunakan

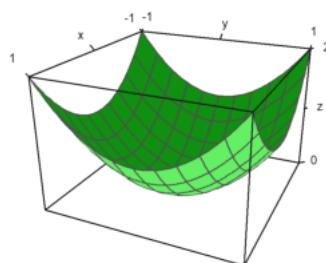
- ekspresi sederhana dalam x dan y,
- nama fungsi dari dua variabel
- atau matriks data.

Standarnya adalah kisi-kisi kawat yang terisi dengan warna yang berbeda di kedua sisi. Perhatikan bahwa jumlah default interval grid adalah 10, namun plot menggunakan jumlah default 40x40 persegi panjang untuk membangun permukaan. Hal ini dapat diubah.

- n=40, n=[40,40]: jumlah garis kisi di setiap arah
- grid=10, grid=[10,10]: jumlah garis grid di setiap arah.

Kami menggunakan default n=40 dan grid=10.

```
>plot3d("x^2+y^2"):
```



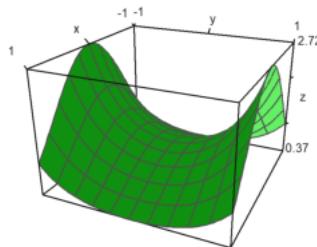
Interaksi pengguna dapat dilakukan dengan parameter >user. Pengguna dapat menekan tombol berikut ini.

- left,right,up,down: memutar sudut pandang
- +,-: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l: berubah memutar sumber cahaya (lihat di bawah)
- space: mengatur ulang ke default

- return: mengakhiri interaksi

```
>plot3d("exp(-x^2+y^2)",>user, ...
title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



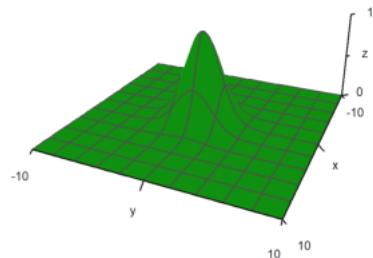
Rentang plot untuk fungsi dapat ditentukan dengan

- a, b: rentang x
- c, d: rentang y
- r: bujur sangkar simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

Terdapat beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: skala untuk nilai fungsi (standarnya adalah <fscale>).  
 scale: angka atau vektor 1x2 untuk menskalakan ke arah x dan y.  
 frame: jenis bingkai (default 1).

```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):
```



Tampilan dapat diubah dengan berbagai cara.

- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut ke sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.

Nilai default dapat diperiksa atau diubah dengan fungsi view(). Fungsi ini mengembalikan parameter dalam urutan di atas.

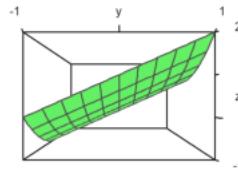
```
>view
```

```
[5, 2.6, 2, 0.4]
```

Jarak yang lebih dekat membutuhkan zoom yang lebih sedikit. Efeknya lebih seperti lensa sudut lebar.

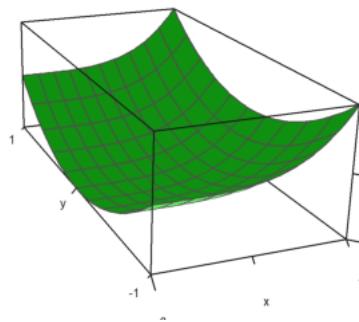
Dalam contoh berikut ini, angle = 0 dan height = 0 terlihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```



Plot terlihat selalu ke bagian tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter center.

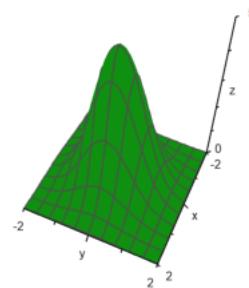
```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
center=[0.4,0,0],zoom=5):
```



Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi, tidak perlu mengubah jarak atau melakukan zoom, tergantung pada ukuran plot. Namun demikian, label mengacu ke ukuran yang sesungguhnya.

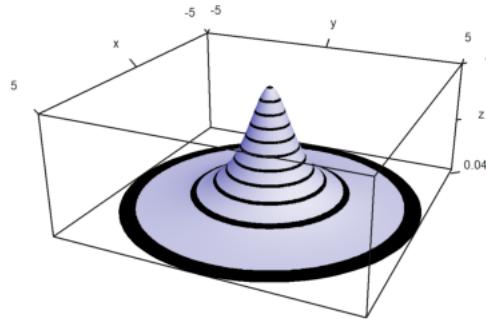
Jika Anda menonaktifkannya dengan scale=false, Anda harus berhati-hati agar plot tetap muat di dalam jendela plotting, dengan mengubah jarak tampilan atau zoom, dan memindahkan bagian tengahnya.

```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
center=[0,0,-2],frame=3):
```

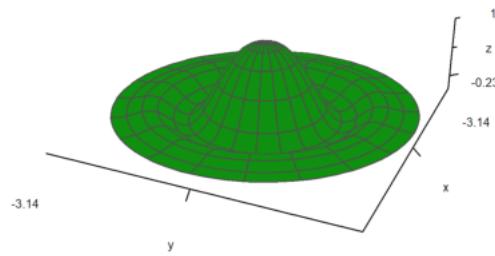


Plot polar juga tersedia. Parameter polar=true menggambar plot polar. Fungsi harus tetap merupakan fungsi dari x dan y. Parameter "fscale" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi akan diskalakan agar sesuai dengan kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
fscale=2,>hue,n=100,zoom=4,>contour,color=blue):
```



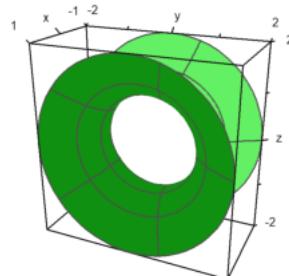
```
>function f(r) := exp(-r/2)*cos(r); ...
plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```



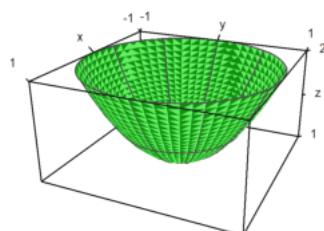
Parameter `rotate` memutar fungsi dalam `x` di sekitar sumbu `x`.

- `rotate = 1`: Menggunakan sumbu `x`
- `rotate = 2`: Menggunakan sumbu `z`

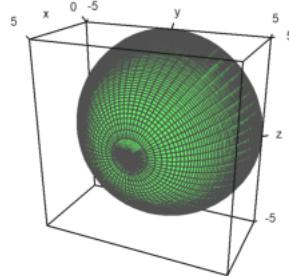
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



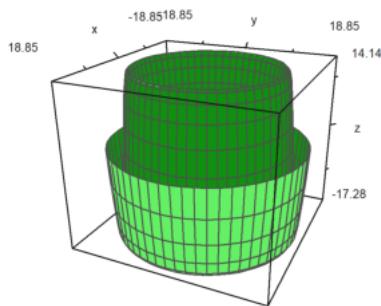
```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```



```
>plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1):
```

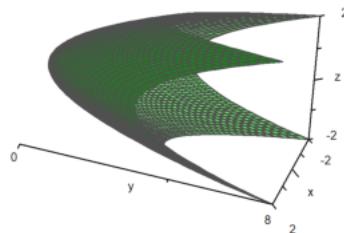


```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```



Berikut ini adalah plot dengan tiga fungsi.

```
>plot3d("x", "x^2+y^2", "y", r=2, zoom=3.5, frame=3):
```



## Plot Kontur

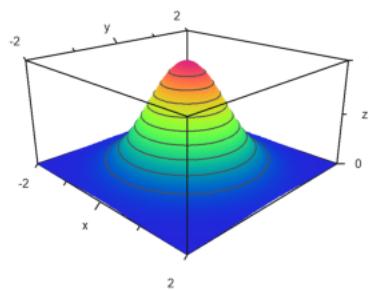
Untuk plot, Euler menambahkan garis kisi-kisi. Sebagai gantinya, dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona berwarna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan bayangan. Pada semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.

- >hue: Mengaktifkan bayangan cahaya, bukan kabel.
- >contour: Memplot garis kontur otomatis pada plot.
- level=... (atau levels): Vektor nilai untuk garis kontur.

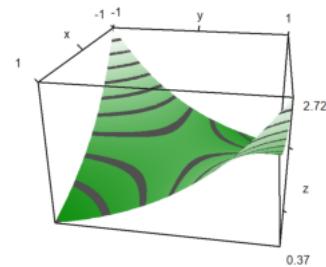
Defaultnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut ini, kami menggunakan grid yang lebih halus untuk titik 100x100, skala fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...
>contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):
```



```
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```

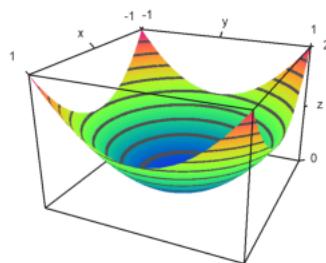


Bayangan default menggunakan warna abu-abu. Tetapi, kisaran warna spektral juga tersedia.

- >spectral: Menggunakan skema spektral default
- color =...: Menggunakan warna khusus atau skema spektral

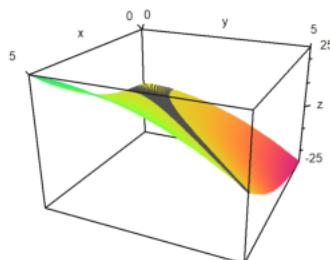
Untuk plot berikut ini, kami menggunakan skema spektral default dan menambah jumlah titik untuk mendapatkan tampilan yang sangat mulus.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```



Alih-alih garis level otomatis, kita juga dapat menetapkan nilai garis level. Hal ini akan menghasilkan garis level yang tipis, alih-alih rentang level.

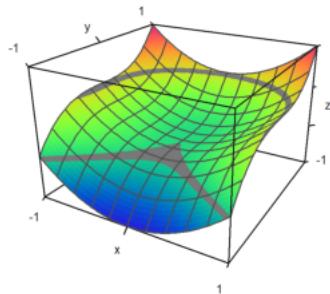
```
>plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=redgreen):
```



Pada plot berikut ini, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas-batas level sebagai kolom.

Selain itu, kami menghamparkan grid dengan 10 interval di setiap arah.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...
>spectral,angle=30°,grid=10,contourcolor=gray):
```

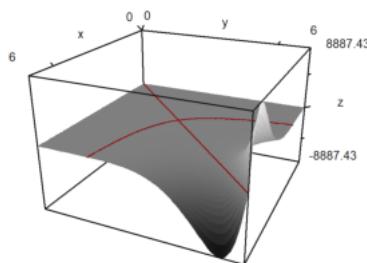


Pada contoh berikut, kami memplot himpunan, di mana

$$f(x,y) = x^y - y^x = 0$$

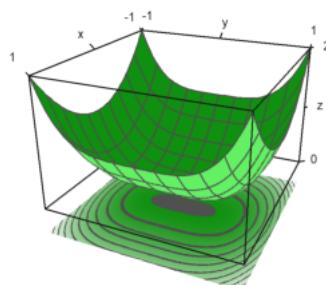
Kita menggunakan satu garis tipis untuk garis level.

```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```



Diumungkinkan untuk menampilkan 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 mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi-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); ...
```

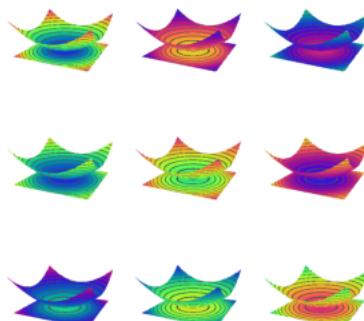
```
>figure(4); ...
  plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...
figure(0):
```



Ada beberapa skema spektral lainnya, yang diberi nomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan `color=value`, di mana `value`

- `spectral`: untuk rentang dari biru ke merah
- `white`: untuk rentang yang lebih redup
- `yellowblue`, `purplegreen`, `blueyellow`, `greenred`
- `blueyellow`, `greenpurple`, `yellowblue`, `redgreen`

```
>figure(3,3); ...
for i=1:9; ...
  figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4); ...
end; ...
figure(0):
```



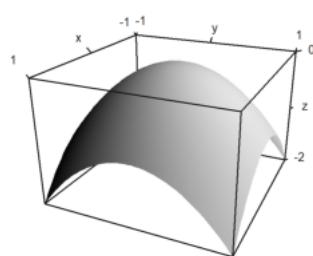
Sumber cahaya dapat diubah dengan `I` dan tombol cursor selama interaksi pengguna. Ini juga dapat ditetapkan dengan parameter.

- `light`: arah cahaya
- `amb`: cahaya sekitar antara 0 dan 1

Perhatikan, bahwa program ini tidak membuat perbedaan di antara sisi-sisi plot. Tidak ada bayangan. Untuk ini Anda akan membutuhkan Povray.

```
>plot3d("-x^2-y^2", ...
hue=true,light=[0,1,1],amb=0,user=true, ...
title="Press l and cursor keys (return to exit)":
```

Press I and cursor keys (return to exit)



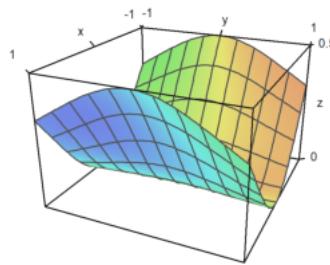
Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01):
```



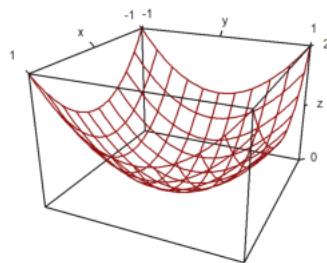
Warna 0 memberikan efek pelangi yang istimewa.

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



Permukaannya juga bisa transparan.

```
>plot3d("x^2+y^2",>transparent,grid=10,wirecolor=red):
```



## Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari sebuah fungsi dalam tiga variabel. olusi dari

$$f(x, y, z) = 0$$

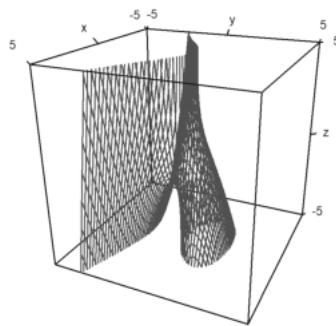
dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, bidang x-z, dan bidang y-z.

- implicit = 1: potong sejajar dengan bidang y-z
- implicit = 2: memotong sejajar dengan bidang x-z
- implicit = 4: memotong sejajar dengan bidang x-y

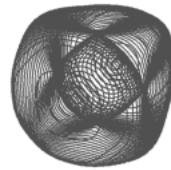
Tambahkan nilai-nilai ini, jika Anda mau. Pada contoh, kami memplot

$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

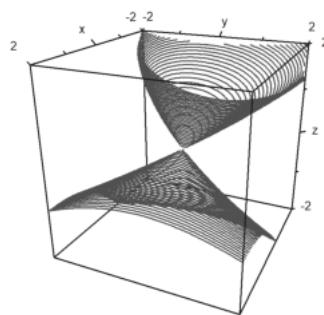
```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```



```
>c=1; d=1;
>plot3d("((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)-d",r=2,<frame,>implicit
```



```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):
```



## Memplot Data 3D

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu menyediakan matriks nilai x, y, dan z, atau tiga fungsi atau ekspresi  $fx(x,y)$ ,  $fy(x,y)$ ,  $fz(x,y)$ .

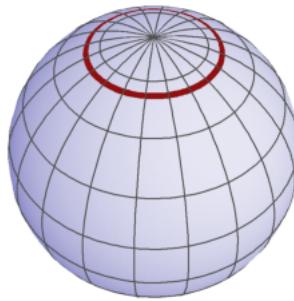
$$\gamma(t,s) = (x(t,s), y(t,s), z(t,s))$$

arena x,y,z adalah matriks, kita mengasumsikan bahwa  $(t,s)$  berjalan melalui kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang dalam ruang.

Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

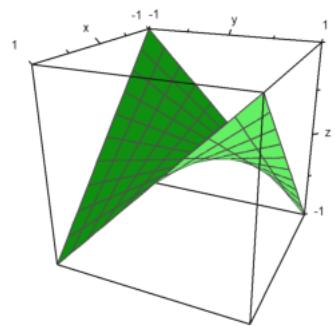
Pada contoh berikut, kita menggunakan vektor nilai  $t$  dan vektor kolom nilai  $s$  untuk memparameterkan permukaan bola. Pada gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)';
x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s);
plot3d(x,y,z,>hue,
color=blue,<frame,grid=[10,20],
values=s,contourcolor=red,level=[90°-24°;90°-22°],
scale=1.4,height=50°):
```



Berikut ini adalah contoh, yang merupakan grafik suatu fungsi.

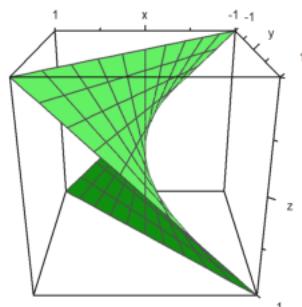
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10);
```



Namun demikian, kita bisa membuat segala macam permukaan. Berikut ini adalah permukaan yang sama dengan suatu fungsi

$$x = yz$$

```
>plot3d(t*s,t,s,angle=180°,grid=10);
```



Dengan lebih banyak upaya, kita bisa menghasilkan banyak permukaan.

Dalam contoh berikut ini, kami membuat tampilan berbayang dari bola yang terdistorsi. Koordinat yang biasa digunakan untuk bola adalah

$$\gamma(t, s) = (\cos(t) \cos(s), \sin(t) \cos(s), \sin(s))$$

dengan

$$0 \leq t \leq 2\pi, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2}.$$

Kami mengubahnya dengan sebuah faktor

$$d(t, s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

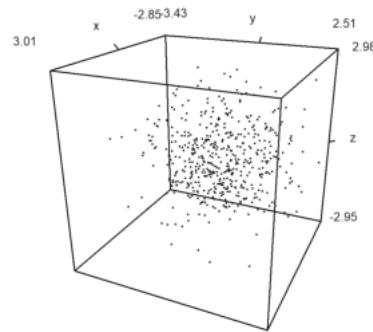
```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
d=1+0.2*(cos(4*t)+cos(8*s)); ...
plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
light=[1,0,1],frame=0,zoom=5);
```



Tentu saja, awan titik juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor untuk koordinat titik.

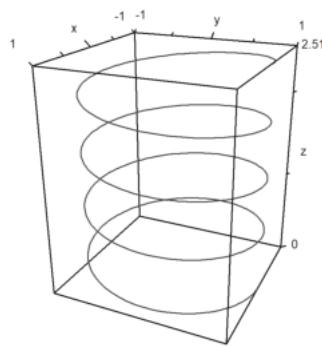
Gaya-gayanya sama seperti pada plot2d dengan points=true;

```
>n=500; ...
plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

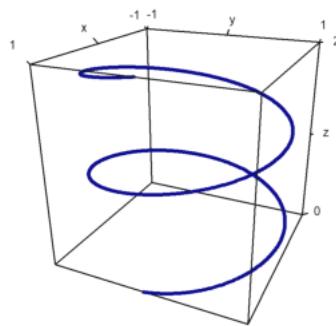


Anda juga dapat memplot kurva dalam bentuk 3D. Dalam hal ini, akan lebih mudah untuk menghitung titik-titik kurva. Untuk kurva pada bidang, kami menggunakan urutan koordinat dan parameter wire = true.

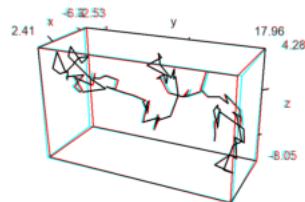
```
>t=linspace(0,8pi,500); ...
plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
linewidth=3, wirecolor=blue):
```

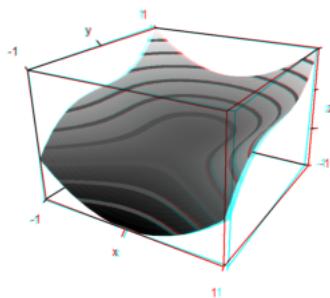


```
>X=cumsum(normal(3,100)); ...
plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



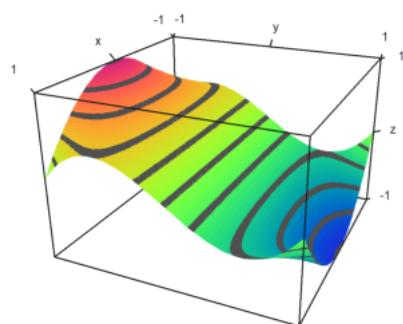
EMT juga dapat membuat plot dalam mode anaglyph. Untuk melihat plot semacam itu, Anda memerlukan red/cyan glasses.

```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```



Sering kali, skema warna spektral digunakan untuk plot. Hal ini menekankan ketinggian fungsi.

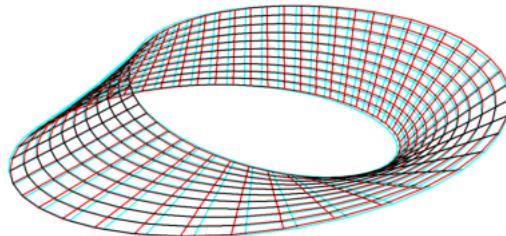
```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```



Euler juga dapat memplot permukaan yang diparameterkan, ketika parameternya adalah nilai x-, y-, dan z- dari gambar kisi-kisi persegi panjang di dalam ruang.

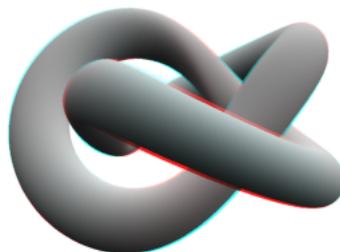
Untuk demo berikut ini, kami menyiapkan parameter u- dan v-, dan menghasilkan koordinat ruang dari parameter ini.

```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...
X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```



Berikut ini contoh yang lebih rumit, yang tampak megah dengan red/cyan glasses.

```
>u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)'; ...
x:=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...
y:=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...
z=sin(u)+2*cos(3*v); ...
plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```



## Plot Statistik

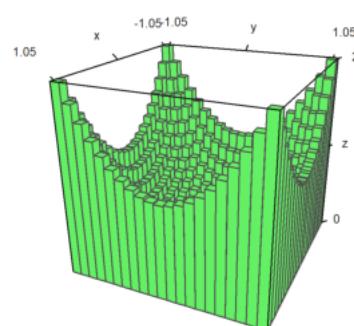
Plot batang juga dapat digunakan. Untuk ini, kita harus menyediakan

- x: vektor baris dengan n+1 elemen
- y: vektor kolom dengan n+1 elemen
- z: matriks nilai berukuran nxn.

z dapat lebih besar, tetapi hanya nilai nxn yang akan digunakan.

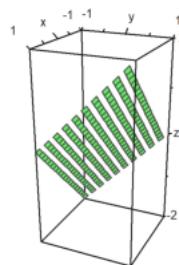
Pada contoh, pertama-tama kita menghitung nilainya. Kemudian kita sesuaikan x dan y, sehingga vektor berada di tengah-tengah nilai yang digunakan.

```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...
xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...
plot3d(xa,ya,z,bar=true):
```



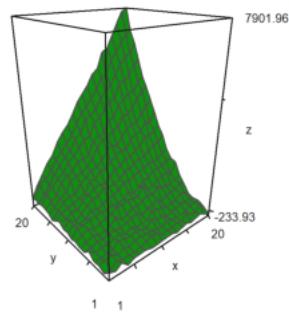
Dimungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
plot3d(x,y,z,disconnect=2:2:20);
```

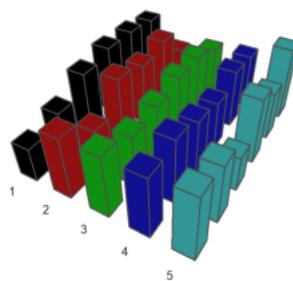


Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan scale(M), atau menskalakan matriks dengan >zscale. Hal ini dapat dikombinasikan dengan faktor penskalaan individual yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...
plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8);
```

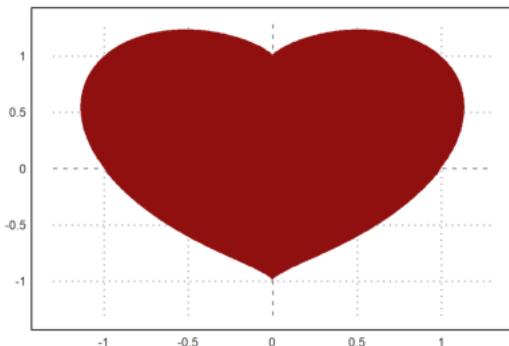


```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
columnsplot3d(v',scols=1:5,ccols=[1:5]);
```



## Permukaan Benda Putar

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1..3, ...
style="#",color=red,<outline, ...
level=[-2;0],n=100);
```



```
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

$$(y^2 + x^2 - 1)^3 - x^2 y^3$$

Kami ingin memutar kurva jantung di sekitar sumbu y. Berikut ini adalah ekspresi yang mendefinisikan heart:

$$f(x,y) = (x^2 + y^2 - 1)^3 - x^2 \cdot y^3.$$

Selanjutnya kita atur

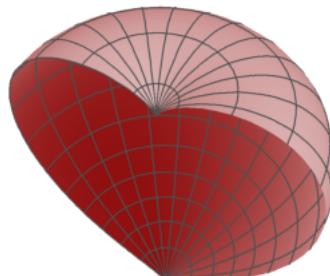
$$x = r \cos(a), \quad y = r \sin(a).$$

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2 \sin a) r^5}{16}$$

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang menyelesaikan untuk r, jika a diberikan. Dengan fungsi tersebut kita dapat memplotkan jantung yang diputar sebagai permukaan parametrik.

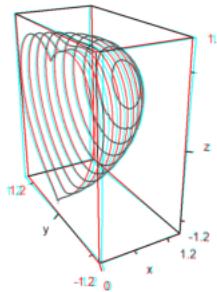
```
>function map f(a) := bisect("fr",0,2;a); ...
t=linspace(-pi/2,pi/2,100); r=f(t); ...
s=linspace(pi,2pi,100)'; ...
plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```



Berikut ini adalah plot 3D dari gambar di atas yang diputar mengelilingi sumbu-z. Kami mendefinisikan fungsi, yang menggambarkan objek.

```
>function f(x,y,z) ...
r=x^2+y^2;
return (r+z^2-1)^3-r*z^3;
endfunction
```

```
>plot3d("f(x,y,z)", ...
xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...
implicit=1,angle=-30°,zoom=2.5,n=[10,100,60],>anaglyph):
```



## Plot 3D Khusus

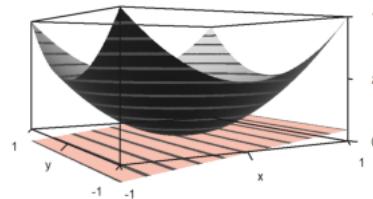
Fungsi plot3d memang bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda juga bisa mendapatkan plot berbingkai dari objek apa pun yang Anda suka.

Meskipun Euler bukan program 3D, namun dapat menggabungkan beberapa objek dasar. Kami mencoba memvisualisasikan parabola dan garis singgungnya.

```
>function myplot ...
y=-1:0.01:1; x=(-1:0.01:1)';
plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ...
    hues=0.5,>contour,color=orange);
h=holding(1);
plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
holding(h);
endfunction
```

Sekarang framedplot() menyediakan frame, dan mengatur tampilan.

```
>framedplot("myplot",[-1,1,-1,1,0,1],height=0,angle=-30°, ...
center=[0,0,-0.7],zoom=3);
```

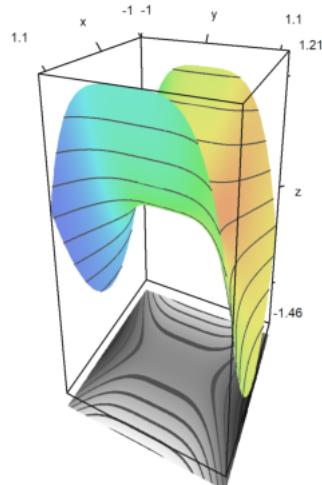


Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d() mengatur jendela ke fullwindow() secara default, namun plotcontourplane() menggunakannya.

```
>x=-1:0.02:1.1; y=x'; z=x^2-y^4;
>function myplot (x,y,z) ...
```

```
zoom(2);
wi=fullwindow();
plotcontourplane(x,y,z,level="auto",<scale>;
plot3d(x,y,z,>hue,<scale,>add,color=white,level="thin");
window(wi);
reset();
endfunction
```

```
>myplot(x,y,z):
```



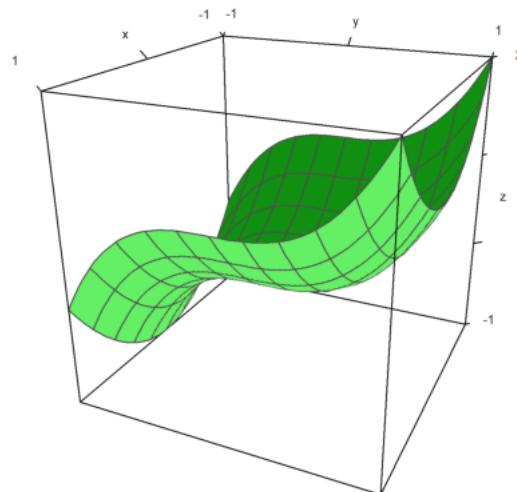
## Animasi

Euler dapat menggunakan frame untuk melakukan pra-komputasi animasi.

Salah satu fungsi yang memanfaatkan teknik ini adalah `rotate`. Fungsi ini dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi ini memanggil `addpage()` untuk setiap plot baru. Akhirnya fungsi ini menganimasikan plot tersebut.

Silakan pelajari sumber dari `rotate` untuk melihat lebih detail.

```
>function testplot () := plot3d("x^2+y^3"); ...
rotate("testplot"); testplot();
```



## Menggambar Povray

Dengan bantuan file Euler povray.e, Euler dapat menghasilkan file Povray. Hasilnya sangat bagus untuk dilihat.

Anda perlu menginstal Povray (32bit atau 64bit) dari <http://www.povray.org/>, dan meletakkan sub-direktori "bin" dari Povray ke dalam jalur lingkungan, atau mengatur variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "povengine.exe".

Antarmuka Povray dari Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah `current.pov`, dan direktori defaultnya adalah `eulerhome()`, biasanya `c:\Users\Username\Euler`. Povray menghasilkan sebuah file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan berkas-berkas ini, gunakan `povclear()`.

Fungsi `pov3d` memiliki semangat yang sama dengan `plot3d`. Fungsi ini dapat menghasilkan grafik dari sebuah fungsi  $f(x,y)$ , atau sebuah permukaan dengan koordinat X,Y,Z dalam bentuk matriks, termasuk garis-garis level yang bersifat opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat adegan ke dalam notebook Euler.

Selain pov3d(), ada banyak fungsi yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string, yang berisi kode Povray untuk objek. Untuk menggunakan fungsi-fungsi ini, mulai file Povray dengan povstart(). Kemudian gunakan writeln(...) untuk menulis objek ke file scene. Terakhir, akhiri file dengan povend(). Secara default, raytracer akan dimulai, dan PNG akan dimasukkan ke dalam buku catatan Euler.

Fungsi objek memiliki parameter yang disebut "look", yang membutuhkan string dengan kode povray untuk tekstur dan hasil akhir objek. Fungsi povlook() dapat digunakan untuk menghasilkan string ini. Fungsi ini memiliki parameter untuk warna, transparansi, Phong Shading, dll.

Perhatikan bahwa Povray universe memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat tetap berpikir dalam sistem koordinat Euler dengan z yang mengarah vertikal ke atas, dan sumbu x, y, z di tangan kanan. Anda perlu memuat file povray.

```
>load povray;
```

Pastikan direktori bin povray berada di dalam path. Jika tidak, edit variabel berikut sehingga berisi jalur ke povray yang dapat dieksekusi.

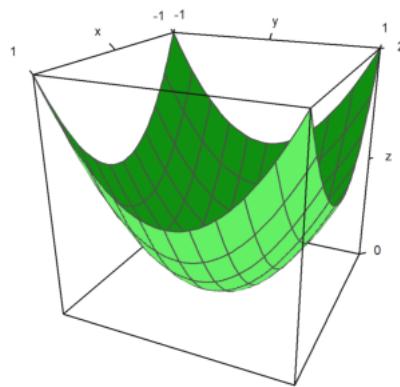
```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe

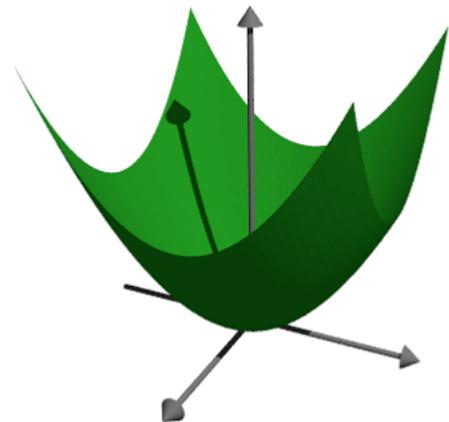
Untuk kesan pertama, kita plot sebuah fungsi sederhana. Perintah berikut ini menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk melacak sinar pada file ini.

Jika Anda mulai perintah berikut, GUI Povray akan terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya, apakah Anda ingin mengizinkan file exe dijalankan. Anda dapat menekan cancel untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK pada jendela Povray untuk mengetahui dialog awal Povray.

```
>plot3d("x^2+y^2", zoom=2) :
```

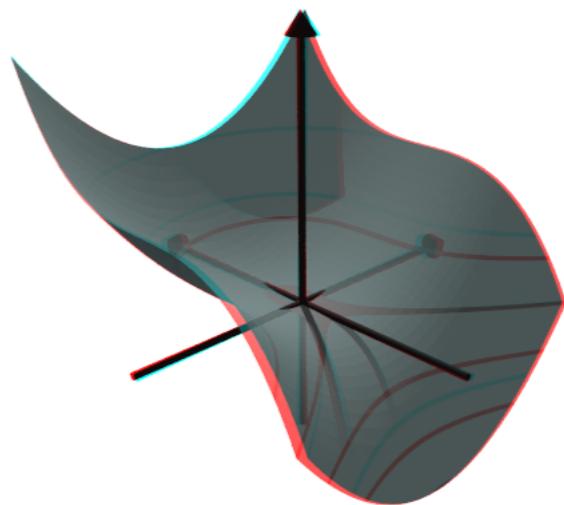


```
>pov3d("x^2+y^2", zoom=3);
```



Kita dapat membuat fungsi menjadi transparan dan menambahkan hasil akhir lainnya. Kita juga dapat menambahkan garis level ke plot fungsi.

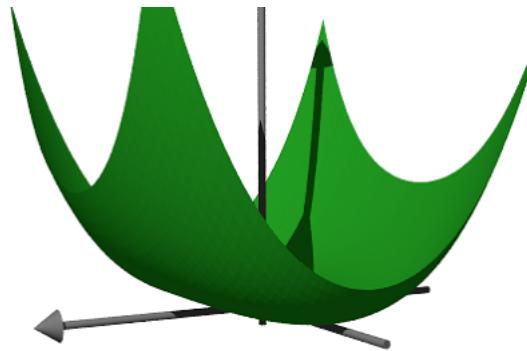
```
>pov3d("x^2+y^3",axiscolor=red,angle=-45°,>anaglyph, ...
look=povlook(cyan,0.2),level=-1:0.5:1,zoom=3.8);
```



Kadang-kadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.

Kami memplot kumpulan titik pada bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=2, ...
angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=10°,n=50, ...
<fscale,zoom=3.8);
```

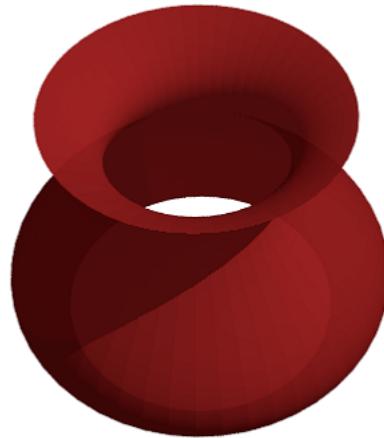


## Merencanakan dengan Koordinat

Sebagai pengganti fungsi, kita dapat membuat plot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objek.

Pada contoh, kita memutar sebuah fungsi pada sumbu z.

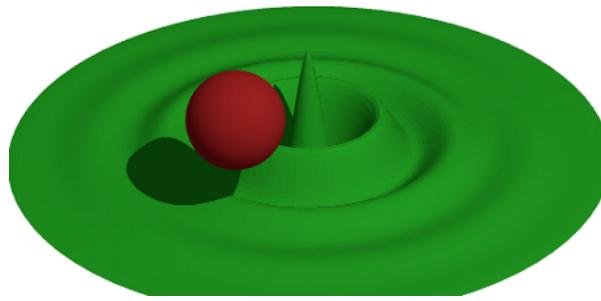
```
>function f(x) := x^3-x+1; ...
x=-1:0.01:1; t=linspace(0,2pi,50)';
Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light=[10,5,15]);
```



Pada contoh berikut, kita memplot gelombang teredam. Kami menghasilkan gelombang dengan bahasa matriks Euler.

Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut. Perhatikan bahwa plot3d menskalakan plot, sehingga sesuai dengan kubus satuan.

```
>r=linspace(0,1,80); phi=linspace(0,2pi,80)';
x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3; ...
pov3d(x,y,z,zoom=6,axis=0,height=30°,add=povsphere([0.5,0,0.25],0.15,povlook(red)), ...
w=500,h=300);
```



Dengan metode bayangan canggih Povray, hanya sedikit titik yang bisa menghasilkan permukaan yang sangat halus. Hanya pada batas-batas dan bayangan, trik ini bisa terlihat jelas.

Untuk itu, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>Z &= x^2*y^3
```

$$\begin{matrix} 2 & 3 \\ x & y \end{matrix}$$

Persamaan permukaannya adalah  $[x,y,Z]$ . Kami menghitung dua turunan terhadap  $x$  dan  $y$  dari persamaan ini dan mengambil hasil perkalian silang sebagai normal.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

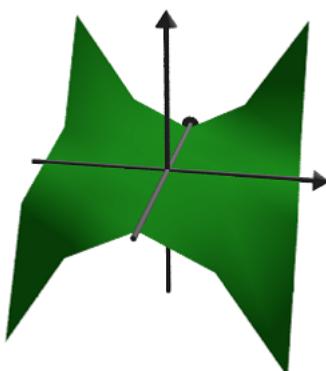
Kami mendefinisikan normal sebagai hasil kali silang dari turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

$$\begin{matrix} 3 & 2 & 2 \\ [-2x^y, -3x^y, 1] \end{matrix}$$

Kami hanya menggunakan 25 titik.

```
>x=-1:0.5:1; y=x';
>pov3d(x,y,Z(x,y),angle=10°, ...
xv=NX(x,y),yv=NY(x,y),zv=NZ(x,y),<shadow>;
```



Berikut ini adalah simpul Trefoil yang dibuat oleh A. Busser di Povray. Ada versi yang lebih baik dari ini dalam contoh.

#### Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kita tambahkan vektor normal di sini. Kita menggunakan Maxima untuk menghitung normal untuk kita. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolik.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunan terhadap x dan y.

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang yang normal, yang merupakan produk silang dari dua turunan.

```
>dn &= crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

```
>x:=linspace(-pi,pi,40); y:=linspace(-pi,pi,100)';
```

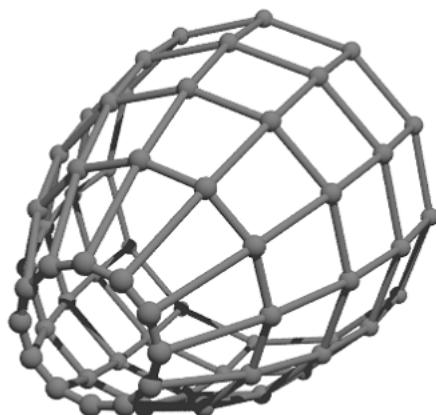
Vektor normal adalah evaluasi dari ekspresi simbolik dn[i] untuk i=1,2,3. Sintaks untuk ini adalah &“expression”(parameter). Ini adalah sebuah alternatif dari metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),>anaglyph,axis=0,zoom=5,w=450,h=350, ...
<shadow,look=povlook(blue), ...
xv=&"dn[1]"(x,y), yv=&"dn[2]"(x,y), zv=&"dn[3]"(x,y));
```



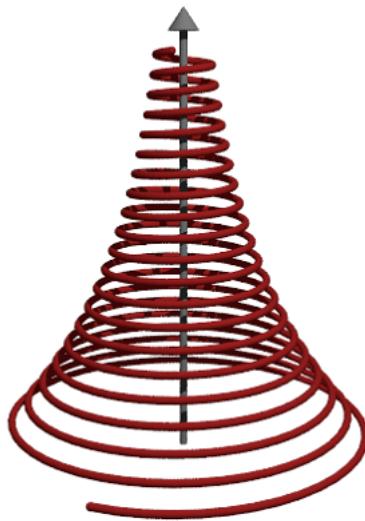
Kita juga dapat menghasilkan kisi-kisi dalam bentuk 3D

```
>povstart(zoom=4); ...
x=-1:0.5:1; r=1-(x+1)^2/6; ...
t=(0°:30°:360°)'; y=r*cos(t); z=r*sin(t); ...
writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...
povend();
```



Dengan povgrid(), kurva dimungkinkan.

```
>povstart(center=[0,0,1],zoom=3.6); ...
t=linspace(0,2,1000); r=exp(-t); ...
x=cos(2*pi*10*t)*r; y=sin(2*pi*10*t)*r; z=t; ...
writeln(povgrid(x,y,z,povlook(red))); ...
writeAxis(0,2,axis=3); ...
povend();
```



## 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 dalam file Povray.

Kita memulai output dengan povstart()

```
>povstart(zoom=4);
```

Pertama, kita mendefinisikan tiga silinder, dan menyimpannya dalam bentuk string di Euler.

Fungsi povx() dll. hanya mengembalikan vektor [1,0,0], yang dapat digunakan sebagai gantinya.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...
c2=povcylinder(-povy,povy,1,povlook(yellow)); ...
c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

```
>c2
```

```
cylinder { <0,0,-1>, <0,0,1>, 1
    texture { pigment { color rgb <0.941176,0.941176,0.392157> } }
    finish { ambient 0.2 }
}
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.

Hal ini dilakukan dengan povlook(), yang mengembalikan sebuah string dengan kode Povray yang relevan. Kita dapat menggunakan warna default Euler, atau menentukan warna kita sendiri. Kita juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

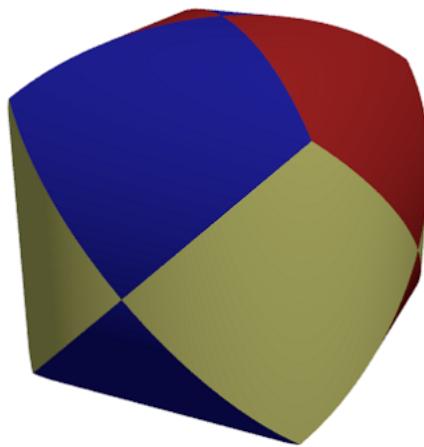
```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> } }
finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek perpotongan, dan menulis hasilnya ke file.

```
>writeln(povintersection([c1,c2,c3]));
```

Perpotongan tiga silinder sulit untuk divisualisasikan, jika Anda belum pernah melihatnya.

```
>povend;
```



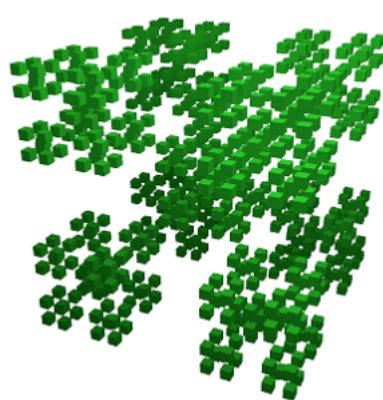
Fungsi-fungsi berikut ini menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana. Fungsi povbox() mengembalikan sebuah string, yang berisi koordinat kotak, tekstur dan hasil akhir.

```
>function onebox(x,y,z,d) := povbox([x,y,z],[x+d,y+d,z+d],povlook());
>function fractal (x,y,z,h,n) ...
```

```
if n==1 then writeln(onebox(x,y,z,h));
else
  h=h/3;
  fractal(x,y,z,h,n-1);
  fractal(x+2*h,y,z,h,n-1);
  fractal(x,y+2*h,z,h,n-1);
  fractal(x,y,z+2*h,h,n-1);
  fractal(x+2*h,y+2*h,z,h,n-1);
  fractal(x+2*h,y,z+2*h,h,n-1);
  fractal(x,y+2*h,z+2*h,h,n-1);
  fractal(x+2*h,y+2*h,z+2*h,h,n-1);
  fractal(x+h,y+h,z+h,h,n-1);
endif;
endfunction
```

```
>povstart(fade=10,<shadow>;
>fractal(-1,-1,-1,2,4);
>povend();
```



Perbedaan memungkinkan pemotongan satu objek dari objek lainnya. Seperti persimpangan, ada bagian dari objek CSG Povray.

```
>povstart(light=[5,-5,5], fade=10);
```

Untuk demonstrasi ini, kita mendefinisikan sebuah objek di Povray, alih-alih menggunakan string di Euler. Definisi akan langsung dituliskan ke file.

Koordinat kotak -1 berarti [-1,-1,-1].

```
>povdefine("mycube",povbox(-1,1));
```

Kita dapat menggunakan objek ini dalam povobject(), yang mengembalikan sebuah string seperti biasa.

```
>c1=povobject("mycube",povlook(red));
```

Kami menghasilkan kubus kedua, dan memutar serta menskalakannya sedikit.

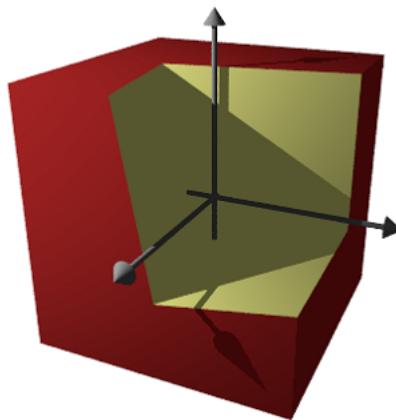
```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...
rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita ambil selisih dari kedua objek tersebut.

```
>writeln(povdifference(c1,c2));
```

Sekarang tambahkan tiga sumbu.

```
>writeAxis(-1.2,1.2,axis=1); ...
writeAxis(-1.2,1.2,axis=2); ...
writeAxis(-1.2,1.2,axis=4); ...
povend();
```



## Fungsi Implisit

Povray dapat memplot himpunan di mana  $f(x,y,z)=0$ , seperti parameter implisit di plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsi-fungsi ini sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

$$((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2) = d$$

```
>povstart(angle=70°,height=50°,zoom=4);
>c=0.1; d=0.1; ...
writeln(povsurface("pow(pow(x,2)+pow(y,2)-pow(c,2),2)+pow(pow(z,2)-1,2))*pow(pow(y,2)+pow(z,2)-pow(c,2),2)+pow(p
povend();
```

Error : Povray error!

Error generated by error() command

povray:

```

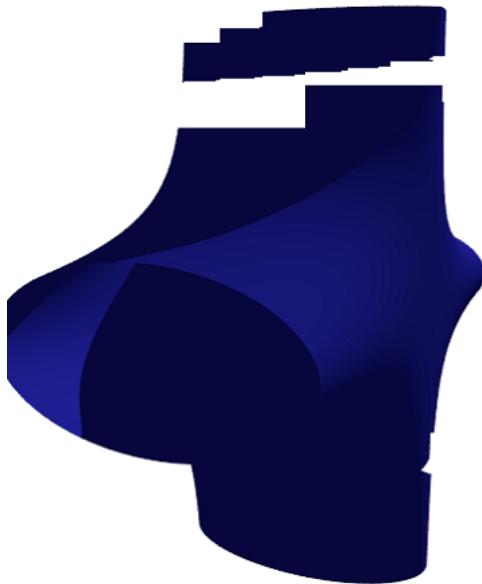
error("Povray error!");
Try "trace errors" to inspect local variables after errors.
povend();
povray(file,w,h,aspect,exit);

```

```

>povstart(angle=25°,height=10°);
>writeln(povsurface("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,2,"")));
>povend();

```



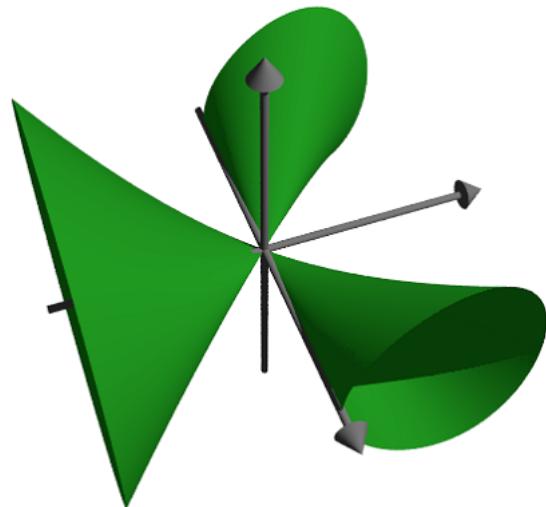
```
>povstart(angle=70°,height=50°,zoom=4);
```

Membuat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi.

```

>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...
writeAxes(); ...
povend();

```



## Objek Jaring

Pada contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.

Kami ingin memaksimalkan xy di bawah kondisi  $x+y = 1$  dan mendemonstrasikan sentuhan tangensial dari garis level.

```
>povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kita tidak dapat menyimpan objek dalam sebuah string seperti sebelumnya, karena ukurannya terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan #declare. Fungsi povtriangle()

melakukan hal ini secara otomatis. Fungsi ini dapat menerima vektor normal seperti halnya pov3d().

Berikut ini mendefinisikan objek mesh, dan menuliskannya langsung ke dalam file.

```
>x=0:0.02:1; y=x'; z=x*y; vx=-y; vy=-x; vz=1;
>mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kita tentukan dua cakram, yang akan berpotongan dengan permukaan.

```
>c1=povdisc([0.5,0.5,0],[1,1,0],2); ...
l1=povdisc([0,0,1/4],[0,0,1],2);
```

Tuliskan permukaan dikurangi kedua cakram.

```
>writeln(povdifference(mesh,povunion([c1,l1]),povlook(green)));
```

Tuliskan kedua perpotongan tersebut.

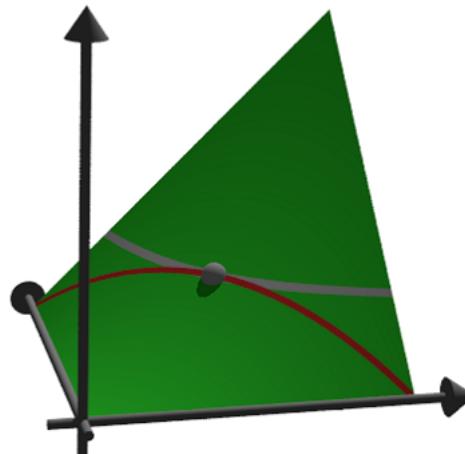
```
>writeln(povintersection([mesh,c1],povlook(red))); ...
writeln(povintersection([mesh,l1],povlook(gray)));
```

Tulislah satu titik secara maksimal.

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsiz));
```

Tambahkan sumbu dan selesaikan.

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...
povend();
```



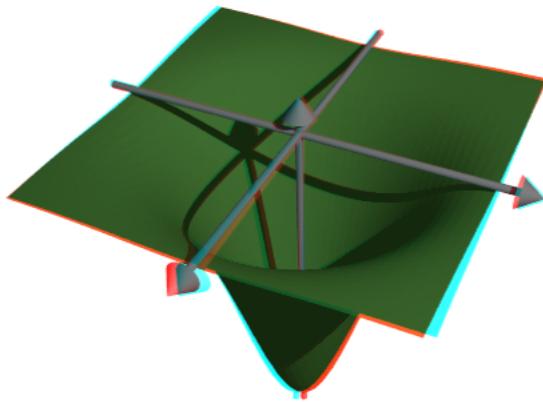
## Anaglyph dalam Povray

Untuk menghasilkan anaglyph untuk kacamata merah/cyan, Povray harus dijalankan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph().

Tentu saja, Anda membutuhkan kacamata merah/cyan untuk melihat contoh berikut dengan benar.

Fungsi pov3d() memiliki tombol sederhana untuk menghasilkan anaglyph.

```
>pov3d("-exp(-x^2-y^2)/2",r=2,height=45°,>anaglyph, ...
center=[0,0,0.5],zoom=3.5);
```

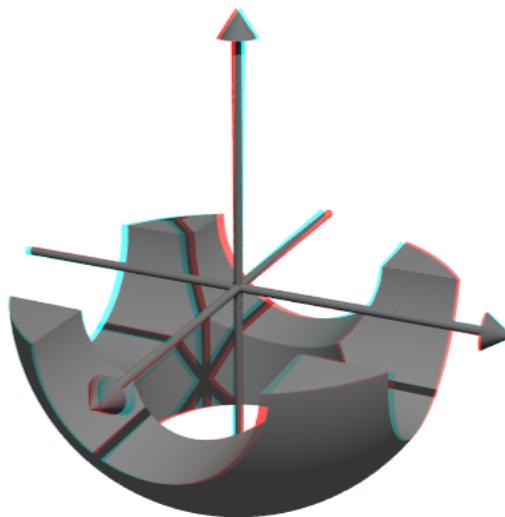


Jika Anda membuat scene dengan objek, Anda harus menempatkan pembuatan scene ke dalam suatu fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

```
>function myscene ...
  s=povsphere(povc,1);
  cl=povcylinder(-povz,povz,0.5);
  clx=povobject(cl,rotate=xrotate(90°));
  cly=povobject(cl,rotate=yrotate(90°));
  c=povbox([-1,-1,0],1);
  un=povunion([cl,clx,cly,c]);
  obj=povdifference(s,un,povlook(red));
  writeln(obj);
  writeAxes();
endfunction
```

Fungsi povanaglyph() melakukan semua ini. Parameter-parameternya seperti pada povstart() dan povend() yang digabungkan.

```
>povanaglyph ("myscene", zoom=4.5);
```



## Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak objek. Namun Anda tidak dibatasi pada objek-objek tersebut. Anda dapat membuat objek sendiri, yang menggabungkan objek-objek lain, atau objek yang benar-benar baru.

Kami mendemonstrasikan sebuah torus. Perintah Povray untuk ini adalah "torus". Jadi kita mengembalikan sebuah string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat pada titik asal.

```
>function povdonat (r1,r2,look "") ...
  return "torus {" + r1 + "," + r2 + look + "}";
endfunction
```

Inilah torus pertama kami.

```
>t1=povdonat(0.8,0.2)
```

```
torus {0.8,0.2}
```

Mari kita gunakan objek ini untuk membuat torus kedua, ditranslasikan dan diputar.

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
object { torus {0.8,0.2}
  rotate 90 *x
  translate <0.8,0,0>
}
```

Sekarang, kita tempatkan semua benda ini ke dalam suatu pemandangan. Untuk tampilannya, kami menggunakan Phong Shading.

```
>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...
writeln(povobject(t1,povlook(green,phong=1))); ...
writeln(povobject(t2,povlook(green,phong=1))); ...
```

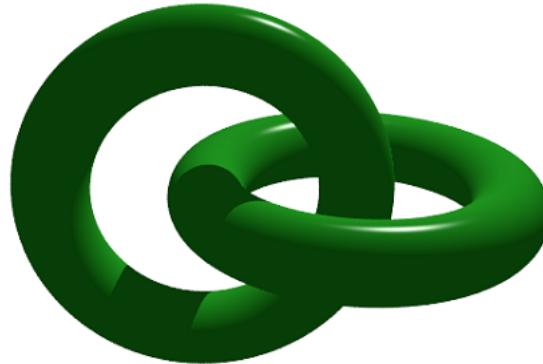
```
>povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, program ini tidak menampilkan kesalahan. Oleh karena itu, Anda harus menggunakan

```
>povend(<exit>);
```

jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray tetap terbuka.

```
>povend(h=320,w=480);
```



Berikut adalah contoh yang lebih rumit. Kami menyelesaikan

$$Ax \leq b, \quad x \geq 0, \quad c \cdot x \rightarrow \text{Max.}$$

dan menunjukkan titik-titik yang layak dan optimal dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
>b=[10,10,10,10]';
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini memiliki solusi atau tidak.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0, 1, 0.5]
```

Ya, benar.

Selanjutnya kita mendefinisikan dua objek. Yang pertama adalah pesawat

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look "") ...
  return povplane(a,b,look)
endfunction
```

Kemudian kita tentukan perpotongan semua setengah ruang dan kubus.

```
>function adm (A, b, r, look "") ...
  ol=[];
  loop 1 to rows(A); ol=ol|oneplane(A[#],b[#]); end;
  ol=ol|povbox([0,0,0],[r,r,r]);
  return povintersection(ol,look);
endfunction
```

Sekarang, kita bisa merencanakan adegan tersebut.

```
>povstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); ...
writeln(adm(A,b,2,povlook(green,0.4))); ...
writeAxes(0,1.3,0,1.6,0,1.5); ...
```

Berikut ini adalah lingkaran di sekeliling optimal.

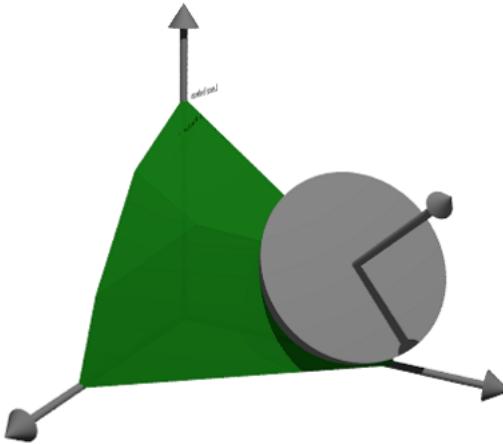
```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
povlook(red,0.9)));
```

Dan kesalahan pada arah yang optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah sebuah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

```
>writeln(povtext("Linear Problem", [0,0.2,1.3], size=0.05, rotate=5°)); ...
povend();
```



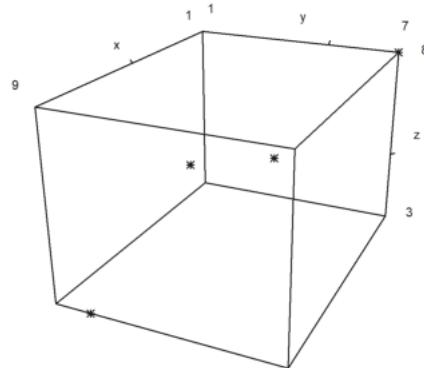
## Contoh Lainnya

Anda dapat menemukan beberapa contoh lain untuk Povray di Euler dalam file-file berikut.

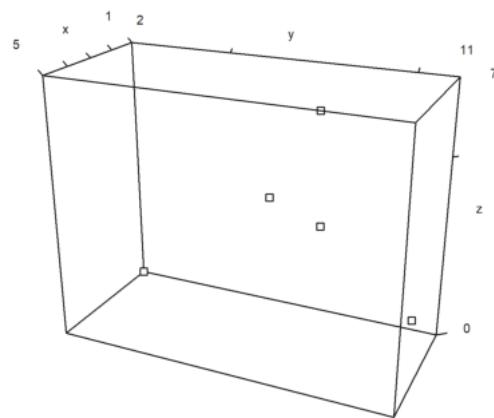
- [Examples/Dandelin Spheres](#)
- [Examples/Donut Math](#)
- [Examples/Trefoil Knot](#)
- [Examples/Optimization by Affine Scaling](#)

## Contoh Soal dan Penyelesaian

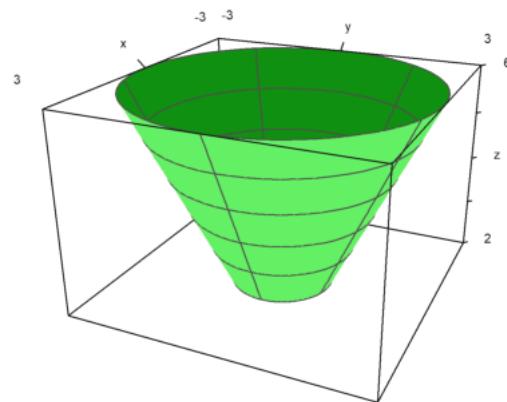
```
>x=[5,1,2,9]; y=[5,7,1,2]; z=[6,8,4,3];
>plot3d(x,y,z,points=true,style="*");
```



```
>x=[1,2,3,4,5]; y=[2,8,7,11,9]; z=[0,3,4,2,7];
>plot3d(x,y,z,points=true,zoom=3,style="/"):
```



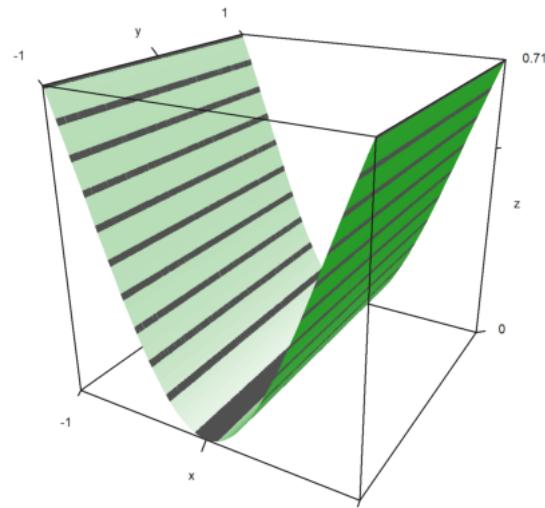
```
>plot3d("2*x",a=1,b=3,rotate=2,grid=5):
```



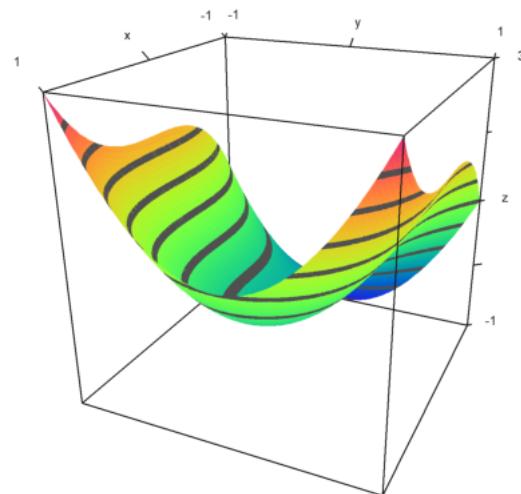
Gambarkan

$\sin(x)^2$ 

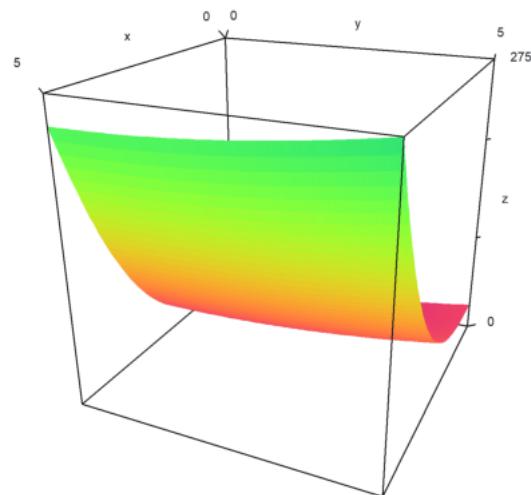
```
>plot3d("sin(x)^2",angle=pi/6,>contour,color=green,>hue):
```



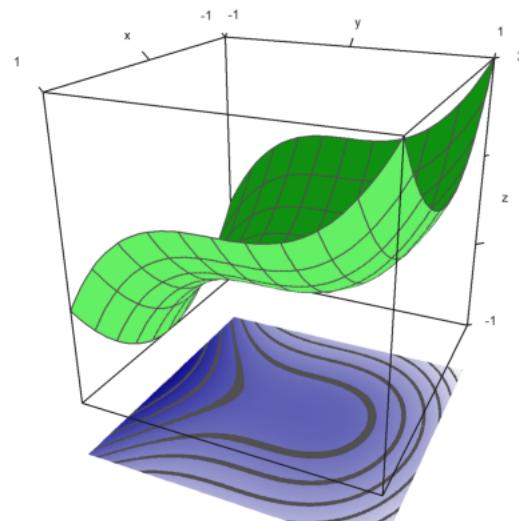
```
>plot3d("x^3+2*y^2",>spectral,>contour,n=100):
```



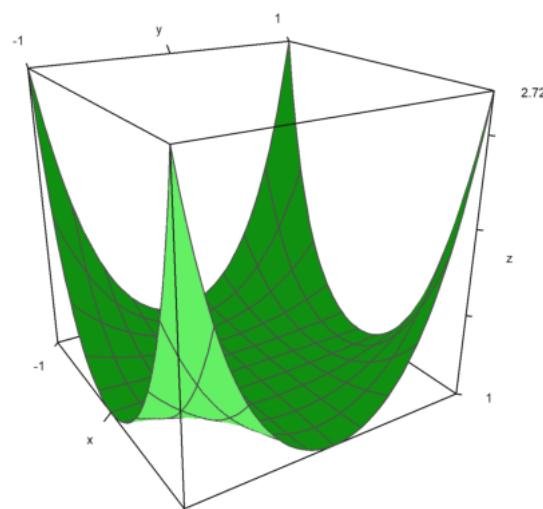
```
>plot3d("2*x^3+y^2",0,5,0,5,level=-6:0.3:1,color=redgreen):
```



```
>plot3d("x^2+y^3",>cp,cpcolor=blue,cpdelta=0.2):
```



```
>plot3d("exp (x^2*y^2)",angle=45,<contour,color=blue):
```



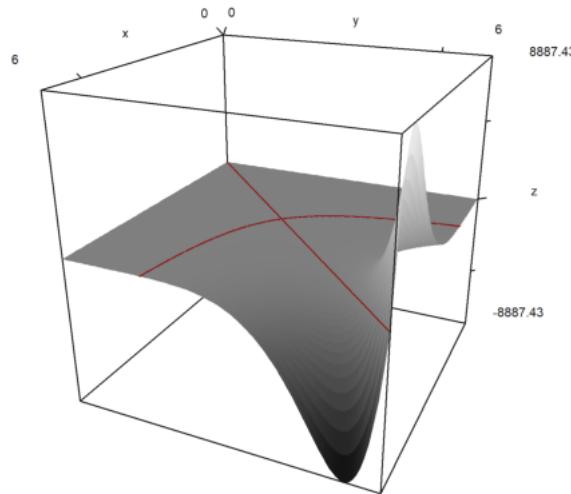
```
>spectral, angle=30
```

```
-2
30
```

Memplot himpunan dimana:

$$f(x, y) = x^y - y^x = 0$$

```
>plot3d("x^y-y^x", level=0, a=0, b=6, c=0, d=6, contourcolor=red, n=100):
```



## Kalkulus dengan EMT

Materi Kalkulus mencakup di antaranya:

- Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi)
- Limit Fungsi,
- Turunan Fungsi,
- Integral Tak Tentu,
- Integral Tentu dan Aplikasinya,
- Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

### Mendefinisikan Fungsi

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:

- Menggunakan format `nama_fungsi := rumus fungsi` (untuk fungsi numerik),
- Menggunakan format `nama_fungsi &= rumus fungsi` (untuk fungsi simbolik, namun dapat dihitung secara numerik),
- Menggunakan format `nama_fungsi &&= rumus fungsi` (untuk fungsi simbolik murni, tidak dapat dihitung langsung),
- Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah `function` (bukan sebagai ekspresi).

Berikut adalah beberapa contoh cara mendefinisikan fungsi:

$$f(x) = 2x^2 + e^{\sin(x)}.$$

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
```

```
1
4.31977682472
20.7392088022
```

```
>f(a) // tidak dapat dihitung nilainya
```

Real 41 x 1 matrix

```
1
1.21868
1.55948
2.01872
```

```

2.58957
3.26182
4.02223
4.85563
5.74672
6.68221
7.65308
8.65613
9.69456
10.7774
11.9179
13.1314
14.4331
15.8362
17.3508
18.984
...

```

Silakan Anda plot kurva fungsi di atas!

Berikutnya kita definisikan fungsi:

$$g(x) = \frac{\sqrt{x^2 - 3x}}{x + 1}.$$

```

>function g(x) := sqrt(x^2-3*x)/(x+1)
>g(3)

```

0

```
>g(0)
```

0

```
>g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik
```

```

Floating point error!
Error in sqrt
Try "trace errors" to inspect local variables after errors.
g:
  useglobal; return sqrt(x^2-3*x)/(x+1)
Error in:
g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik ...
^

```

Silakan Anda plot kurva fungsi di atas!

```
>f(g(5)) // komposisi fungsi
```

2.20920171961

```
>g(f(5))
```

0.950898070639

```

>function h(x) := f(g(x)) // definisi komposisi fungsi
>h(5) // sama dengan f(g(5))

```

2.20920171961

Silakan Anda plot kurva fungsi komposisi fungsi f dan g:

$$h(x) = f(g(x))$$

dan

$$u(x) = g(f(x))$$

bersama-sama kurva fungsi f dan g dalam satu bidang koordinat.

```
>f(0:10) // nilai-nilai f(0), f(1), f(2), ..., f(10)
```

```
[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,
99.929, 130.69, 163.51, 200.58]
```

```
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
```

```
[1, 4.31978, 10.4826, 19.1516, 32.4692, 50.3833, 72.7562,
99.929, 130.69, 163.51, 200.58]
```

```
>gmap(200:210)
```

```
[0.987534, 0.987596, 0.987657, 0.987718, 0.987778, 0.987837,
0.987896, 0.987954, 0.988012, 0.988069, 0.988126]
```

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0. \end{cases}$$

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format :=, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.

```
>function map f(x) ...
  if x>0 then return x^3
  else return x^2
  endif;
endfunction
```

```
>f(1)
```

1

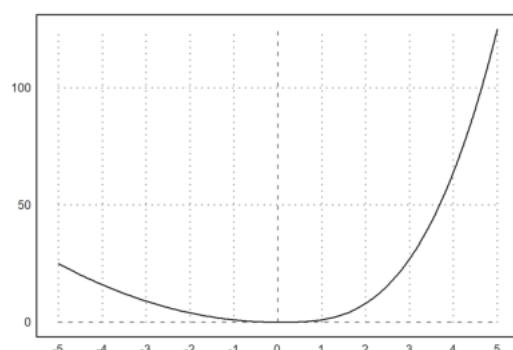
```
>f(-2)
```

4

```
>f(-5:5)
```

```
[25, 16, 9, 4, 1, 0, 1, 8, 27, 64, 125]
```

```
>aspect(1.5); plot2d("f(x)", -5, 5):
```



```
>function f(x) &= 2*E^x // fungsi simbolik
```

x  
2 E

```
>$f(a) // nilai fungsi secara simbolik
```

$2e^a$

```
>f(E) // nilai fungsi berupa bilangan desimal
```

30.308524483

```
>$f(E), $float(%)
```

$2e^e$

30.30852448295852

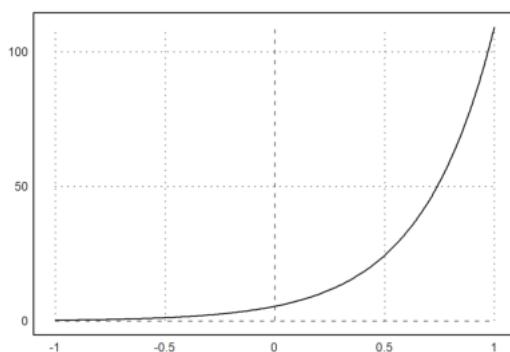
```
>function g(x) &= 3*x+1
```

$3x + 1$

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

$\frac{3}{2}e^x + 1$

```
>plot2d("h(x)", -1, 1);
```



## Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan fungsi-fungsi tersebut dan komposisinya di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik fungsi-fungsi tersebut dan komposisi-komposisi 2 fungsi.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

```
>function f(x) := x^2-3*x+2; // fungsi aljabar
>f(-4)
```

30

```
>f(6)
```

20

```
>plot2d("f(x)", -4, 6, grid=1); ...
title("Grafik f(x) = x^2-3x+2"); xlabel("X"); ylabel("Y");
```

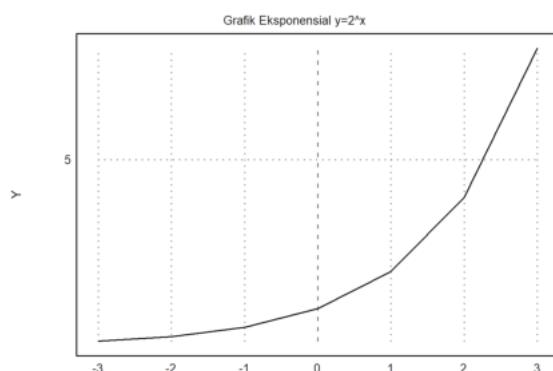


Buatlah grafik eksponensial

$$y = 2^x$$

dengan  $x = [-3, -2, -1, 0, 1, 2, 3]$

```
>x=[-3,-2,-1,0,1,2,3]; y=2^x; plot2d(x,y,"0-"); ...
ylabel("Y"); title("Grafik Eksponensial y=2^x"); ...
grid := true;
```



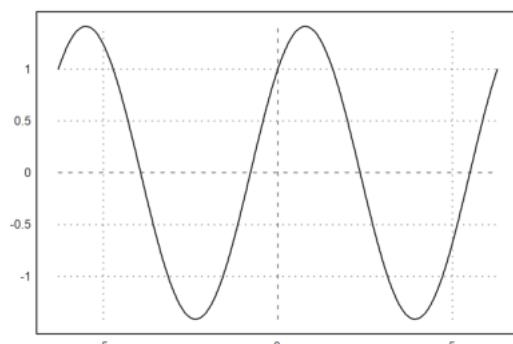
```
>function f(x) := sin(x)+cos(x) // fungsi trigonometri
>f(0)
```

1

```
>f(pi/2)
```

1

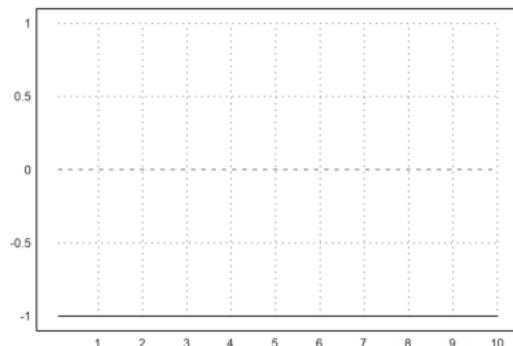
```
>plot2d("sin(x)+cos(x)", -2*pi, 2*pi);
```



```
>function f(x) := log10(x) // fungsi logaritma
>f(2)
```

0.301029995664

```
>plot2d(["log10(x)"], 0.1, 10):
```



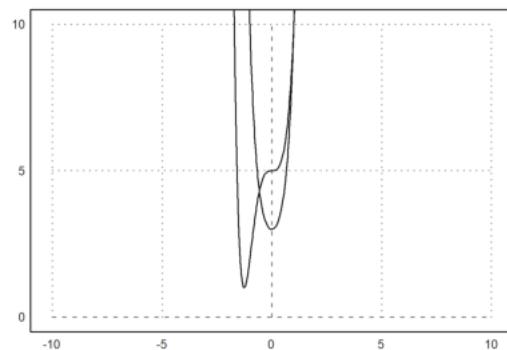
```
>function f(x) := x^2+1
>function g(x) := x^3+2
>f(g(4)) // komposisi fungsi
```

4357

```
>g(f(4))
```

4915

```
>function h(x) := f(g(x))
>function u(x) := g(f(x))
>plot2d(["h(x)", "u(x)"], -10, 10, 0, 10):
```



## Menghitung Limit

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi "plus" atau "minus". Hasil limit dapat berupa nilai, "und" (tak definisi), "ind" (tak tentu namun terbatas), "infinity" (kompleks tak hingga).

Perhatikan beberapa contoh berikut. Perhatikan cara menampilkan perhitungan secara lengkap, tidak hanya menampilkan hasilnya saja.

```
>$showev('limit(sqrt(x^2-3*x)/(x+1),x,inf))
```

$$\lim_{x \rightarrow \infty} \frac{\sqrt{x^2 - 3x}}{x + 1} = 1$$

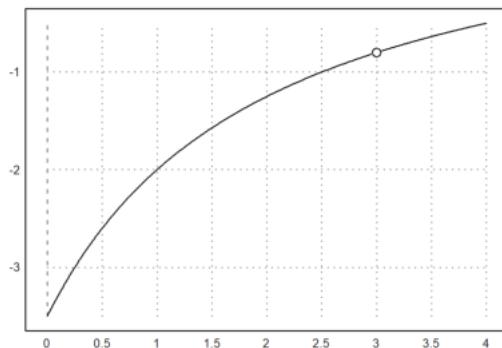
```
>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

$$-\frac{4}{5}$$

$$\lim_{x \rightarrow 3} \frac{x^3 - 13x^2 + 51x - 63}{x^3 - 4x^2 - 3x + 18} = -\frac{4}{5}$$

Fungsi tersebut diskontinu di titik  $x=3$ . Berikut adalah grafik fungsinya.

```
>aspect(1.5); plot2d("(x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18)",0,4); plot2d(3,-4/5,>points,style="ow",>add):
```

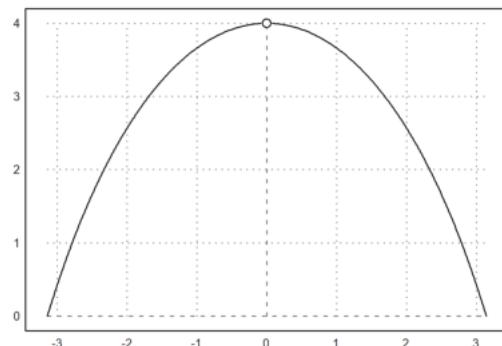


```
>$limit(2*x*sin(x)/(1-cos(x)),x,0)
```

$$2 \left( \lim_{x \rightarrow 0} \frac{x \sin x}{1 - \cos x} \right) = 4$$

Fungsi tersebut diskontinu di titik  $x=0$ . Berikut adalah grafik fungsinya.

```
>plot2d("2*x*sin(x)/(1-cos(x))",-pi,pi); plot2d(0,4,>points,style="ow",>add):
```



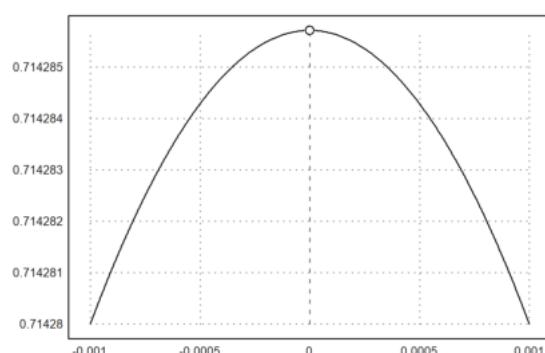
```
>$limit(cot(7*h)/cot(5*h),h,0)
```

$$\frac{5}{7}$$

$$\lim_{h \rightarrow 0} \frac{\cot(7h)}{\cot(5h)} = \frac{5}{7}$$

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di  $x=0$ . Berikut adalah grafiknya.

```
>plot2d("cot(7*x)/cot(5*x)",-0.001,0.001); plot2d(0,5/7,>points,style="ow",>add):
```



```
>$showev('limit(((x/8)^(1/3)-1)/(x-8),x,8))
```

$$\lim_{x \rightarrow 8} \frac{\frac{x^{1/3}}{2} - 1}{x - 8} = \frac{1}{24}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(1/(2*x-1),x,0))
```

$$\lim_{x \rightarrow 0} \frac{1}{2x-1} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

$$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(sqrt(x^2+x)-x,x,inf))
```

$$\lim_{x \rightarrow \infty} \sqrt{x^2 + x} - x = \frac{1}{2}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(abs(x-1)/(x-1),x,1,minus))
```

$$\lim_{x \downarrow 1} \frac{|x-1|}{x-1} = -1$$

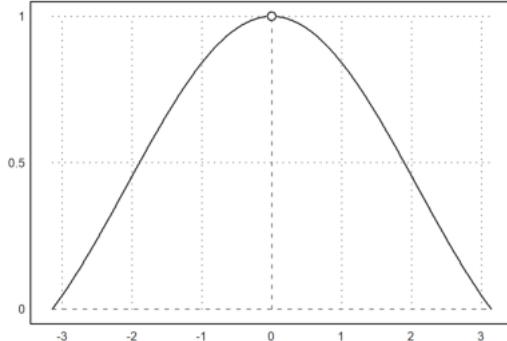
Hitung limit di atas untuk x menuju 1 dari kanan.

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(sin(x)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

```
>plot2d("sin(x)/x",-pi,pi); plot2d(0,1,>points,style="ow",>add):
```



```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit(log(x), x, minf))
```

$$\lim_{x \rightarrow -\infty} \log x = \text{infinity}$$

```
>$showev('limit((-2)^x,x, inf))
```

$$\lim_{x \rightarrow \infty} (-2)^x = \text{infinity}$$

```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t \uparrow 2} t - \sqrt{2-t} = 2$$

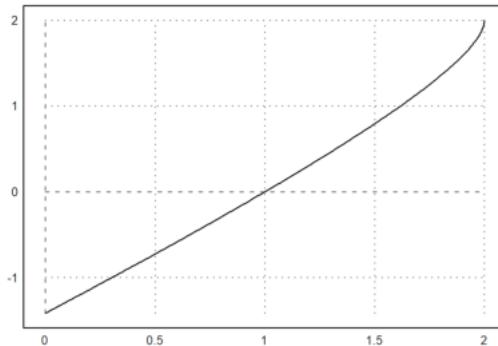
```
>$showev('limit(t-sqrt(2-t),t,2,plus))
```

$$\lim_{t \downarrow 2} t - \sqrt{2-t} = 2$$

```
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
```

$$\lim_{t \downarrow 5} t - \sqrt{2-t} = 5 - \sqrt{3}i$$

```
>plot2d("x-sqrt(2-x)", 0, 2):
```



```
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
```

$$\lim_{x \rightarrow 3} \frac{x^2 - 9}{2x^2 - 5x - 3} = \frac{6}{7}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

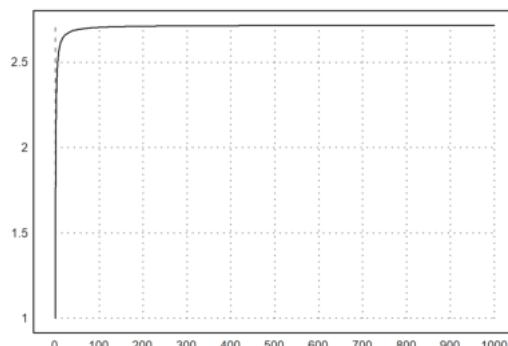
$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$showev('limit((1+1/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{1}{x} + 1 \right)^x = e$$

```
>plot2d("(1+1/x)^x", 0, 1000):
```



```
>$showev('limit((1+k/x)^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{k}{x} + 1 \right)^x = e^k$$

```
>$showev('limit((1+x)^(1/x),x,0))
```

$$\lim_{x \rightarrow 0} (x+1)^{\frac{1}{x}} = e$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contohnya.

```
>$showev('limit((x/(x+k))^x,x,inf))
```

$$\lim_{x \rightarrow \infty} \left( \frac{x}{x+k} \right)^x = e^{-k}$$

```
>$showev('limit((E^x-E^2)/(x-2),x,2))
```

$$\lim_{x \rightarrow 2} \frac{e^x - e^2}{x - 2} = e^2$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contohnya.

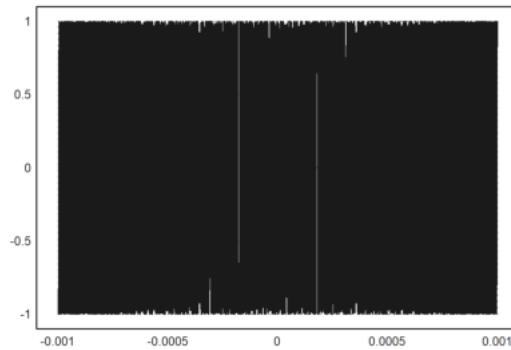
```
>$showev('limit(sin(1/x),x,0))
```

$$\lim_{x \rightarrow 0} \sin\left(\frac{1}{x}\right) = \text{ind}$$

```
>$showev('limit(sin(1/x),x,inf))
```

$$\lim_{x \rightarrow \infty} \sin\left(\frac{1}{x}\right) = 0$$

```
>plot2d("sin(1/x)",-0.001,0.001):
```



## 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). hitunglah nilai limitnya dan buat grafiknya

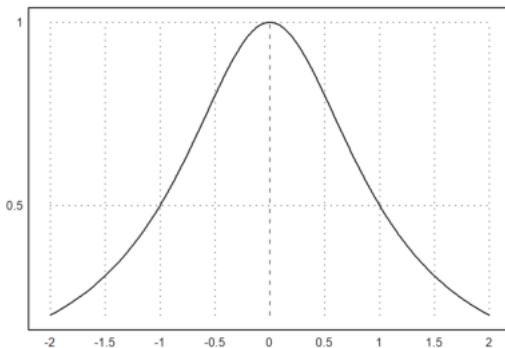
maxima: `limit(1/(x^2+1),x,0)`

$$\lim_{x \rightarrow 0} \frac{1}{x^2 + 1}$$

```
>$showev('limit(1/(x^2+1),x,0))
```

$$\lim_{x \rightarrow 0} \frac{1}{x^2 + 1} = 1$$

```
>plot2d("(1)/(x^2+1)":
```



2). hitung limitnya

maxima: 'limit((3\*x^2+2\*x)/(x-1),x,minf)

$$\lim_{x \rightarrow -\infty} \frac{3x^2 + 2x}{x - 1}$$

```
>$showev('limit((3*x^2+2*x)/(x-1),x,minf))
```

$$\lim_{x \rightarrow -\infty} \frac{3x^2 + 2x}{x - 1} = -\infty$$

3). cari limit kanan dan kiri dari

maxima: 'limit((abs(x-1))/(x-1),x,1)

$$\lim_{x \rightarrow 1} \frac{|x - 1|}{x - 1}$$

```
>$showev('limit(abs(x-1)/(x-1),x,1,minus))
```

$$\lim_{x \uparrow 1} \frac{|x - 1|}{x - 1} = -1$$

```
>$showev('limit(abs(x-1)/(x-1),x,1,plus))
```

$$\lim_{x \downarrow 1} \frac{|x - 1|}{x - 1} = 1$$

3). Hitung limit dari

maxima: 'limit((4\*x+7)/(x-1),x,inf)

$$\lim_{x \rightarrow \infty} \frac{4x + 7}{x - 1}$$

```
>$showev('limit((4*x+7)/(x-1),x,inf))
```

$$\lim_{x \rightarrow \infty} \frac{4x + 7}{x - 1} = 4$$

## Turunan Fungsi

---

Definisi turunan:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>$showev('limit(((x+h)^2-x^2)/h,h,0)) // turunan x^2
```

$$\lim_{h \rightarrow 0} \frac{(x + h)^2 - x^2}{h} = 2x$$

```
>p &= expand((x+h)^2-x^2)|simplify; $p // pembilang dijabarkan dan disederhanakan
```

$$2hx + h^2$$

```
>q &= ratsimp(p/h); $q // ekspresi yang akan dihitung limitnya disederhanakan
```

$$2x + h$$

```
>$limit(q,h,0) // nilai limit sebagai turunan
```

2 x

```
>$showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} = n x^{n-1}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan  $(x+h)^n$  dengan menggunakan teorema binomial.

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

$$\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h} = \cos x$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan  $\sin(x+h)$  dengan menggunakan rumus jumlah dua sudut.

```
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
```

$$\lim_{h \rightarrow 0} \frac{\log(x+h) - \log x}{h} = \frac{1}{x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, gunakan sifat-sifat logaritma dan hasil limit pada bagian sebelumnya di atas.

```
>$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x
```

$$\lim_{h \rightarrow 0} \frac{\frac{1}{x+h} - \frac{1}{x}}{h} = -\frac{1}{x^2}$$

```
>$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x
```

```
Answering "Is x an integer?" with "integer"
Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is x an integer?
```

Use assume!

Error in:

```
$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x ...  
^
```

Maxima bermasalah dengan limit:

$$\lim_{h \rightarrow 0} \frac{e^{x+h} - e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{e^h - 1}{h} = 1$$

```
>$factor(E^(x+h)-E^x)
```

$$(e^h - 1) e^x$$

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

$$\left( \lim_{h \rightarrow 0} \frac{e^h - 1}{h} \right) e^x = e^x$$

```
>function f(x) &= x^x
```

x  
x

```
>$showev('limit(f(x),x,0))
```

$$\lim_{x \rightarrow 0} x^x = 1$$

Silakan Anda gambar kurva

$$y = x^x.$$

```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = infinity$$

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h}.$$

Dalam hal ini diperlukan asumsi nilai x.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = x^x (\log x + 1)$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

$$[x > 0]$$

```
>&forget(x<0)
```

$$[x < 0]$$

```
>&facts()
```

```
[kind(sinh, one_to_one), kind(log, one_to_one),
 kind(tanh, one_to_one), kind(log, increasing)]
```

```
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
```

$$\lim_{h \rightarrow 0} \frac{\arcsin(x+h) - \arcsin x}{h} = \frac{1}{\sqrt{1-x^2}}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

$$\lim_{h \rightarrow 0} \frac{\tan(x+h) - \tan x}{h} = \frac{1}{\cos^2 x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

$$\sinh(x)$$

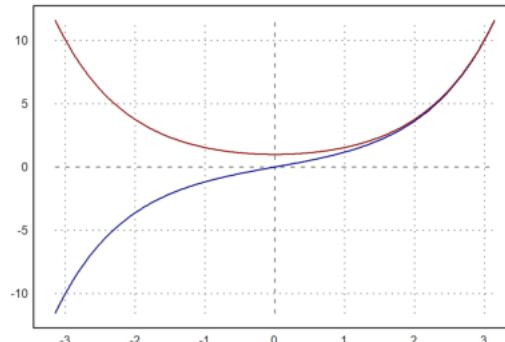
```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

$$\frac{e^{-x} (e^{2x} + 1)}{2}$$

Hasilnya adalah  $\cosh(x)$ , karena

$$\frac{e^x + e^{-x}}{2} = \cosh(x).$$

```
>plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]):
```



```
>function f(x) &= sin(3*x^5+7)^2
```

$$\sin^2(3x^5 + 7)$$

```
>diff(f, 3), diffc(f, 3)
```

```
1198.32948904  
1198.72863721
```

Apakah perbedaan `diff` dan `diffc`?

```
>$showev('diff(f(x), x)')
```

$$\frac{d}{dx} \sin^2(3x^5 + 7) = 30x^4 \cos(3x^5 + 7) \sin(3x^5 + 7)$$

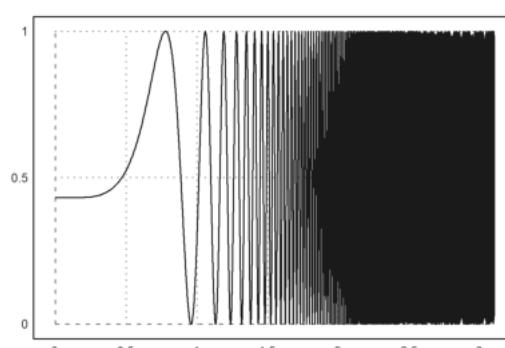
```
>$% with x=3
```

$$\%at\left(\frac{d}{dx} \sin^2(3x^5 + 7), x = 3\right) = 2430 \cos 736 \sin 736$$

```
>$float(%)
```

$$\%at\left(\frac{d^{1.0}}{dx^{1.0}} \sin^2(3.0x^5 + 7.0), x = 3.0\right) = 1198.728637211748$$

```
>plot2d(f, 0, 3.1):
```



```
>function f(x) &= 5*cos(2*x)-2*x*sin(2*x) // mendefinisikan fungsi f
```

```
5 cos(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)
```

$$f(1) = 5 \cos 2 - 2 \sin 2$$

$$-3.899329036387075$$

$$f(2) = 5 \cos 4 - 4 \sin 4$$

$$-0.2410081230863468$$

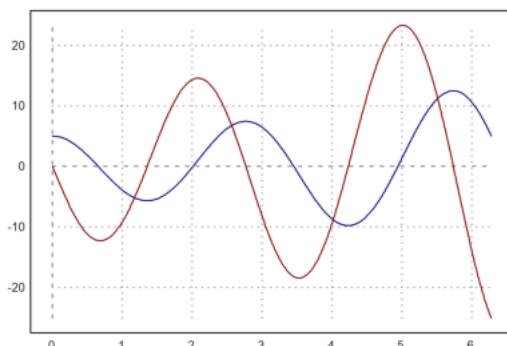
```
>xp=solve("df(x)",1,2,0) // solusi f'(x)=0 pada interval [1, 2]
```

$$1.35822987384$$

```
>df(xp), f(xp) // cek bahwa f'(xp)=0 dan nilai ekstrim di titik tersebut
```

$$\begin{array}{l} 0 \\ -5.67530133759 \end{array}$$

```
>plot2d(["f(x)","df(x")],0,2*pi,color=[blue,red]): //grafik fungsi dan turunannya
```



Perhatikan titik-titik "puncak" grafik  $y=f(x)$  dan nilai turunan pada saat grafik fungsinya mencapai titik "puncak" tersebut.

## Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), menggunakan perintah diff, dan secara manual (langkah demi langkah yang dihitung dengan Maxima) seperti contoh-contoh di atas. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

1). Hitung turunan berikut:

$$f(x) = x^2$$

```
>function f(x) := x^2
>$showev('limit(((x+h)^2-x^2)/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2}{h} = 2x$$

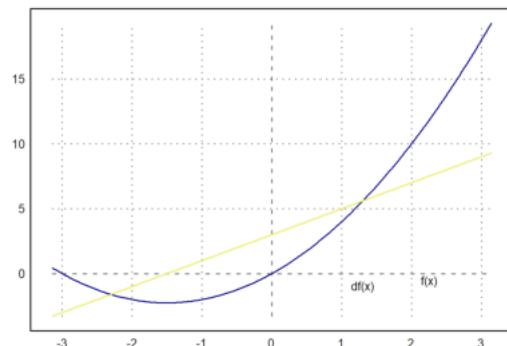
2). Hitung turunan dan gambar grafiknya

$$f(x) = x^2 + 3x$$

```
>function f(x) := x^2+3x
>$showev('limit(((x+h)^2+3*(x+h)-(x^2+3*x))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2 + 3(x+h) - 3x}{h} = 2x + 3$$

```
>function df(x) := 2*x+3
>plot2d(["f(x)","df(x")],-pi,pi,color=[blue,yellow]), label("f(x)",2,0.4), label("df(x)",1,0):
```



3). Hitunglah turunan eksponensial berikut dan gambar grafiknya:

$$e^{3x} + 4xe^x$$

```
>function f(x) &= E^(3*x)+4*x*E^x
```

$$E^{3x} + 4x E^x$$

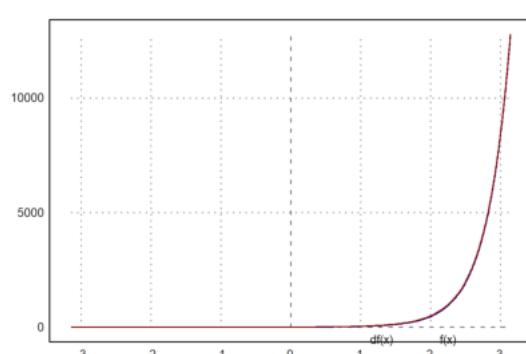
```
>$showev('limit(factor((E^(3*x+h)+4*(x+h)*E^(x+h)-E^(3*x)-4*x*E^x)/h),h,0))
```

$$e^x \left( \lim_{h \rightarrow 0} \frac{e^{2x+h} - e^{2x} + 4e^h x - 4x + 4h e^h}{h} \right) = e^x (e^{2x} + 4x + 4)$$

```
>function df(x) &= E^x*(E^(2*x)+4*x+4)
```

$$E^x (E^{2x} + 4x + 4)$$

```
>plot2d(["f(x)","df(x")],-pi,pi,color=[blue,red]), label("f(x)",2,0.6), label("df(x)",1,-0.5):
```



4). Hitunglah turunan berikut dan gambarkan grafiknya

$$\log(3x + 1)$$

```
>function f(x) &= log(3*x+1)
```

$$\log(3x + 1)$$

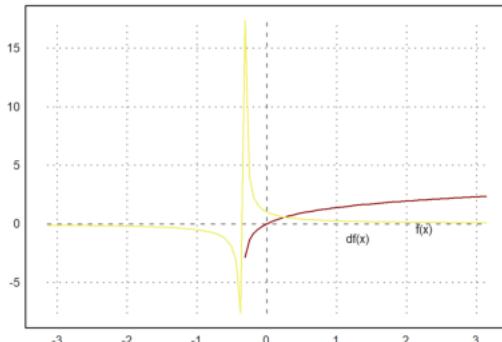
```
>$showev('limit((log(3*x+h+1)-log(3*x+1))/h,h,0))
```

$$\lim_{h \rightarrow 0} \frac{\log(3x + h + 1) - \log(3x + 1)}{h} = \frac{1}{3x + 1}$$

```
>function df(x) &= limit((log(3*x+h+1)-log(3*x+1))/h,h,0); $df(x)// df(x) = f'(x)
```

$$\frac{1}{3x+1}$$

```
>plot2d(["f(x)","df(x")],-pi,pi,color=[red,yellow]), label("f(x)",2,0.6), label("df(x)",1,-0.2):
```



## Integral

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).

Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

$$\int_a^b f(x) dx = F(b) - F(a), \quad \text{dengan } F'(x) = f(x).$$

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak integrabel, kita tidak akan menggunakan metode ini).

```
>$showev('integrate(x^n,x)')
```

Answering "Is n equal to -1?" with "no"

$$\int x^n dx = \frac{x^{n+1}}{n+1}$$

```
>$showev('integrate(1/(1+x),x)')
```

$$\int \frac{1}{x+1} dx = \log(x+1)$$

```
>$showev('integrate(1/(1+x^2),x)')
```

$$\int \frac{1}{x^2+1} dx = \arctan x$$

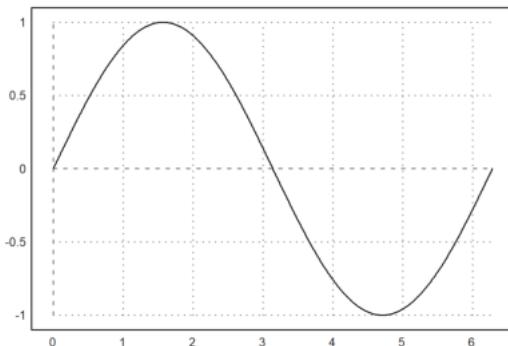
```
>$showev('integrate(1/sqrt(1-x^2),x)')
```

$$\int \frac{1}{\sqrt{1-x^2}} dx = \arcsin x$$

```
>$showev('integrate(sin(x),x,0,pi))
```

$$\int_0^\pi \sin x dx = 2$$

```
>plot2d("sin(x)",0,2*pi):
```



```
>$showev('integrate(sin(x),x,a,b))
```

$$\int_a^b \sin x \, dx = \cos a - \cos b$$

```
>$showev('integrate(x^n,x,a,b))
```

Answering "Is n positive, negative or zero?" with "positive"

$$\int_a^b x^n \, dx = \frac{b^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
```

$$\int x^2 \sqrt{2x+1} \, dx = \frac{(2x+1)^{\frac{7}{2}}}{28} - \frac{(2x+1)^{\frac{5}{2}}}{10} + \frac{(2x+1)^{\frac{3}{2}}}{12}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
```

$$\int_0^2 x^2 \sqrt{2x+1} \, dx = \frac{25^{\frac{5}{2}}}{21} - \frac{2}{105}$$

```
>$ratsimp(%)
```

$$\int_0^2 x^2 \sqrt{2x+1} \, dx = \frac{25^{\frac{7}{2}} - 2}{105}$$

```
>$showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x}+a) e^{\sqrt{x}}}{\sqrt{x}} \, dx = (-e^\pi - 1) \sin a + (e^\pi + 1) \cos a$$

```
>$factor(%)
```

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x}+a) e^{\sqrt{x}}}{\sqrt{x}} \, dx = (-e^\pi - 1) (\sin a - \cos a)$$

```
>function map f(x) &= E^(-x^2)
```

$$\frac{2}{E} - x$$

```
>$showev('integrate(f(x),x))
```

$$\int e^{-x^2} \, dx = \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

$$\operatorname{erf}(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}} \, dx.$$

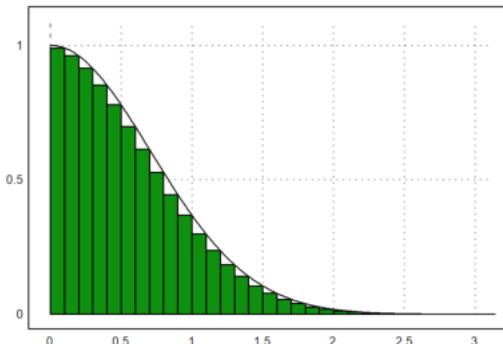
Kita tidak dapat menggunakan teorema Dasar Kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus

kuadratur).

Misalkan kita akan menghitung:

$$\int_0^{\pi} e^{-x^2} dx$$

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```



Integral tentu

$$\int_0^{\pi} e^{-x^2} dx$$

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:

$$\int_0^{\pi} e^{-x^2} dx = 0.8362196102528469$$

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai  $f(x)$  untuk  $x = 0.1, 0.2, 0.3, \dots, 3.2$ .

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

0.836219610253

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

Tunjukkan kebenaran hasil di atas!

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

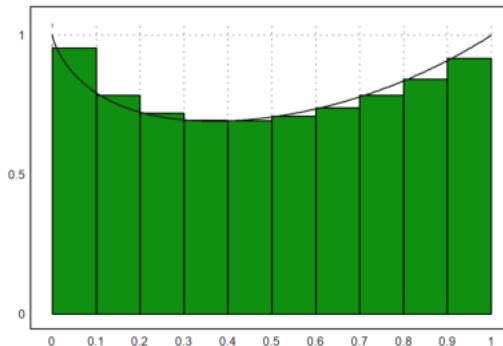
```
>function f(x) &= x^x
```

x  
x

```
>$showev('integrate(f(x),x,0,1))
```

$$\int_0^1 x^x dx = \int_0^1 x^x dx$$

```
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add);
```



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));
```

$$\int_0^1 x^x \, dx = 0.7834935879025506$$

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
>function f(x) &= sin(3*x^5+7)^2
```

$$\sin(3x + 7)$$

```
>integrate(f,0,1)
```

0 542581176074

```
>&showev('integrate(f(x),x,0,1))
```

```

1
/
[      2      5                               1
I sin (3 x + 7) dx = (((I gamma_incomplete(-, 6 I)
]                                         5
/
0
1                               2
- I gamma_incomplete(-, - 6 I)) sin (7)
5
1                               1
+ (- 2 gamma_incomplete(-, 6 I) - 2 gamma_incomplete(-, - 6 I)
5                                         5
1                               1
+ 4 gamma(-) cos(7) sin(7) + (I gamma_incomplete(-, - 6 I)
5                                         5
1                               2           pi      1/5      2
- I gamma_incomplete(-, 6 I)) cos (7) sin(--)) sin(--) + 10 6      sin (7)
5                                         10
1/5      2           1/5
+ 10 6      cos (7))/(20 6      )

```

>&float(%)

```

1.0
/
[      2      5
I      sin (3.0 x + 7.0) dx =
]
/
0.0

```

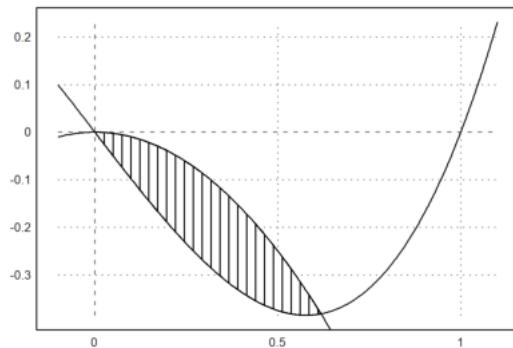
```
0.03494135593857896 (0.3090169943749474
(0.4316313908960832 (I gamma_incomplete(0.2, 6.0 I)
- 1.0 I gamma_incomplete(0.2, - 6.0 I))
+ 0.5683686091039167 (I gamma_incomplete(0.2, - 6.0 I)
- 1.0 I gamma_incomplete(0.2, 6.0 I))
+ 0.4953036778474351 (- 2.0 gamma_incomplete(0.2, 6.0 I)
- 2.0 gamma_incomplete(0.2, - 6.0 I) + 18.36337484799522)
+ 14.30969081105255)
```

```
>$showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)
```

$$\int_0^1 x e^{-x} dx = 1 - 2e^{-1}$$

## Aplikasi Integral Tentu

```
>plot2d("x^3-x",-0.1,1.1); plot2d("-x^2",>add); ...
b=solve("x^3-x+x^2",0.5); x=linspace(0,b,200); xi=flipx(x); ...
plot2d(x|xi,x^3-x|-xi^2,>filled,style="|",fillcolor=1,>add); // Plot daerah antara 2 kurva
```



```
>a=solve("x^3-x+x^2",0), b=solve("x^3-x+x^2",1) // absis titik-titik potong kedua kurva
```

```
0
0.61803398875
```

```
>integrate("(-x^2)-(x^3-x)",a,b) // luas daerah yang diarsir
```

```
0.0758191713542
```

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```
>a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik potong kedua kurva secara eksak
```

$$\left[ x = \frac{-\sqrt{5} - 1}{2}, x = \frac{\sqrt{5} - 1}{2}, x = 0 \right]$$

```
>$showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara eksak
```

$$\int_0^{\frac{\sqrt{5}-1}{2}} -x^3 - x^2 + x \, dx = \frac{13 - 5^{\frac{3}{2}}}{24}$$

```
>$float(%)
```

$$\int_{0.0}^{0.6180339887498949} -1.0 x^3 - 1.0 x^2 + x \, dx = 0.07581917135421037$$

## Panjang Kurva

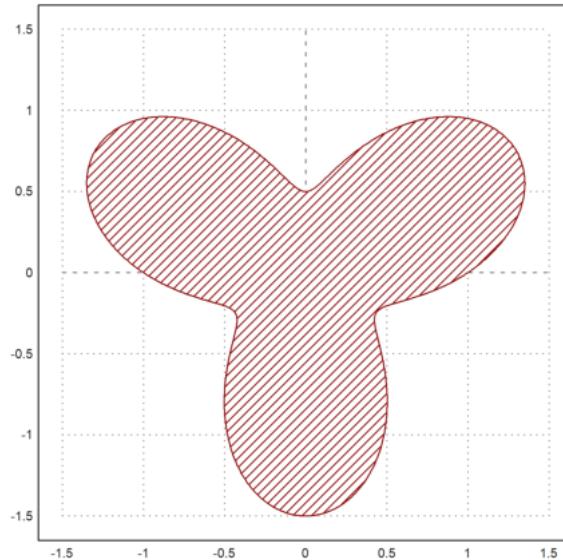
Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

$$\gamma(t) = (r(t) \cos(t), r(t) \sin(t))$$

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}, \quad 0 \leq t \leq 2\pi.$$

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
plot2d(x,y,>filled,fillcolor=red,style="/" ,r=1.5); // Kita gambar kurvanya terlebih dahulu
```



```
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
```

$$r(t) = 1 + \frac{\sin(3t)}{2}$$

```
>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
```

$$fx(t) = \cos t \left( 1 + \frac{\sin(3t)}{2} \right)$$

```
>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
```

$$fy(t) = \sin t \left( 1 + \frac{\sin(3t)}{2} \right)$$

```
>function ds(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t)
```

$$ds(t) = \frac{\sqrt{9 + 4 \sin(3t) + 4 \cos(6t)}}{2}$$

```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
```

$$\frac{\int_0^{2\pi} \sqrt{9 + 4 \sin(3x) + 4 \cos(6x)} dx}{2}$$

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara numerik dengan perintah EMT.

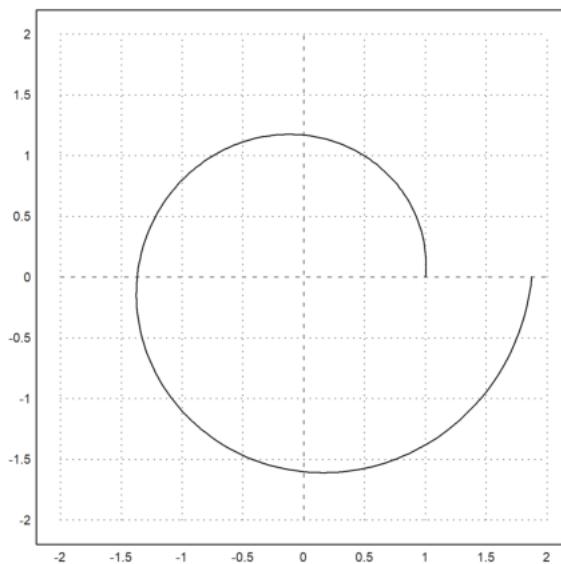
```
>integrate("ds(x)",0,2*pi)
```

9.0749467823

### Spiral Logaritmik

$$x = e^{ax} \cos x, \quad y = e^{ax} \sin x.$$

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi):
```



```
>&kill(a) // hapus expresi a
```

done

```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
```

$$fx(t) = e^{at} \cos t$$

```
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
```

$$fy(t) = e^{at} \sin t$$

```
>function df(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t)
```

$$df(t) = \sqrt{1 + a^2} e^{at}$$

```
>S &=integrate(df(t),t,0,2*pi); $S // panjang kurva (spiral)
```

$$\sqrt{1 + a^2} \left( -\frac{1}{a} + \frac{e^{2\pi a}}{a} \right)$$

```
>S(a=0.1) // Panjang kurva untuk a=0.1
```

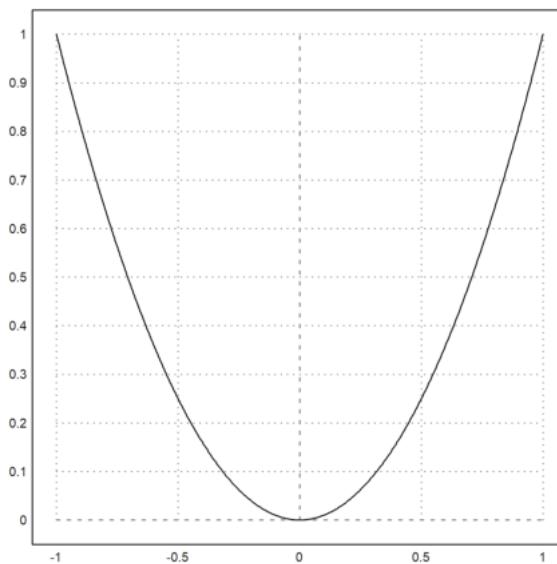
8.78817491636

Soal:

Tunjukkan bahwa keliling lingkaran dengan jari-jari r adalah  $K=2\pi r$ .

Berikut adalah contoh menghitung panjang parabola.

```
>plot2d("x^2",xmin=-1,xmax=1):
```



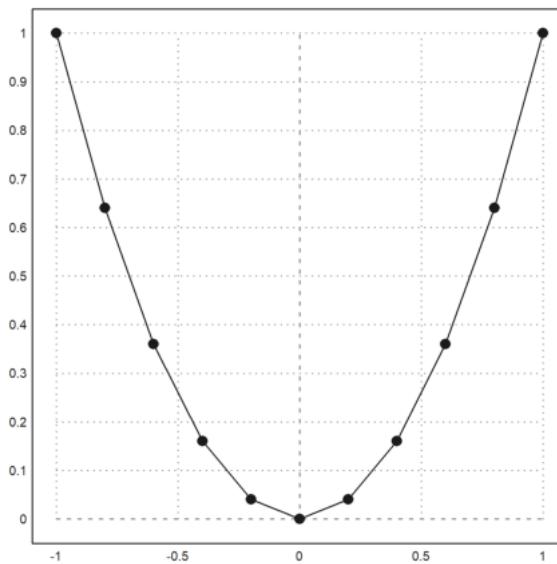
```
>$showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))
```

$$\int_{-1}^1 \sqrt{1 + 4x^2} dx = \frac{2\sqrt{5} + \operatorname{asinh} 2}{2}$$

```
>$float(%)
```

$$\int_{-1.0}^{1.0} \sqrt{1.0 + 4.0x^2} dx = 2.957885715089195$$

```
>x=-1:0.2:1; y=x^2; plot2d(x,y); ...
plot2d(x,y,points=1,style="o#",add=1):
```



Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```
>i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])^2+(y[i+1]-y[i])^2))
```

2.95191957027

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

#### Koordinat Kartesius

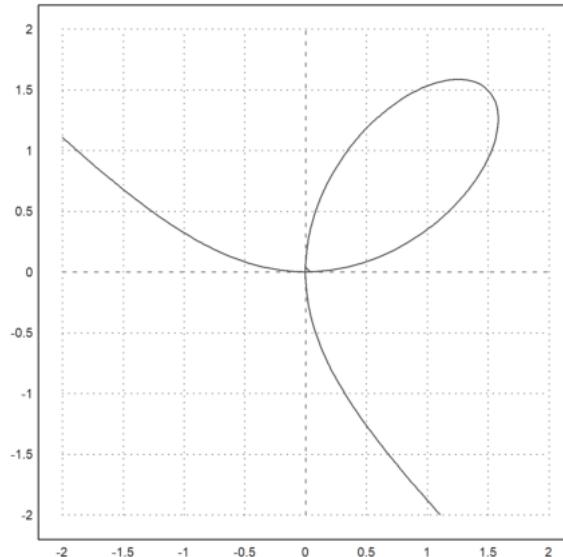
Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

$$x^3 + y^3 - 3xy = 0.$$

```
>z &= x^3+y^3-3*x*y; $z
```

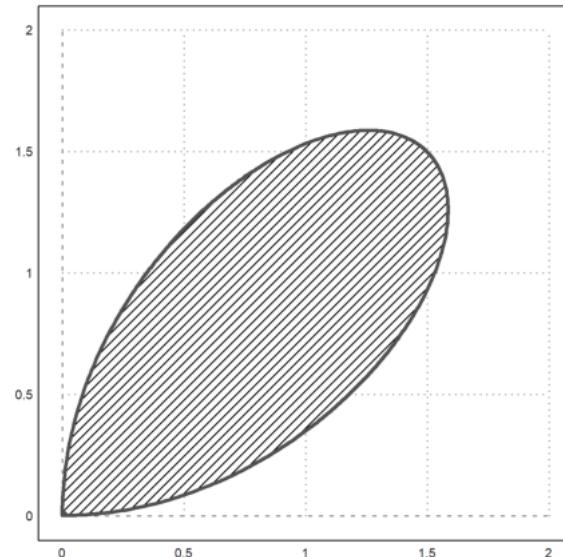
$$x^3 - 3xy + y^3$$

```
>plot2d(z,r=2,level=0,n=100):
```



Kita tertarik pada kurva di kuadran pertama.

```
>plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="/"):
```



Kita selesaikan persamaannya untuk x.

```
>$z with y=l*x, sol &= solve(%,x); $sol
```

$$-3lx^2 + x^3 + l^3x^3$$

$$\left[ x = \frac{3l}{1+l^3}, x = 0 \right]$$

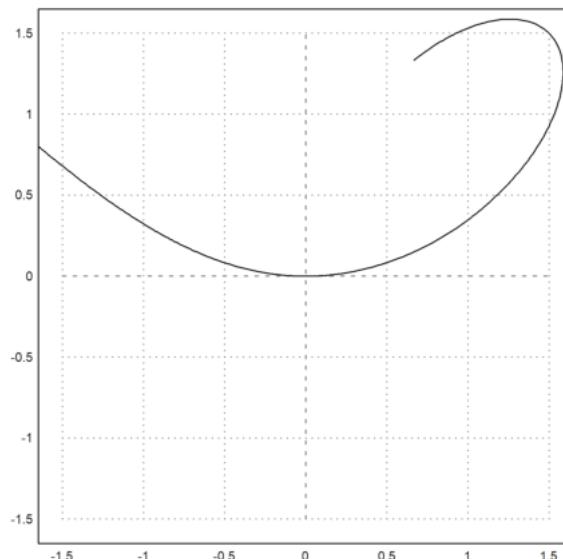
Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); $'f(l)=f(l)
```

$$f(l) = \frac{3l}{1+l^3}$$

Fungsi tersebut juga dapat digunakan untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai y=l\*x, yakni x=f(l) dan y=l\*f(l).

```
>plot2d(&f(x), &x*f(x), xmin=-0.5, xmax=2, a=0, b=2, c=0, d=2, r=1.5) :
```



Elemen panjang kurva adalah:

$$ds = \sqrt{f'(l)^2 + (lf'(l) + f(l))^2}.$$

```
>function ds(l) &= ratsimp(sqrt(diff(f(l), l)^2+diff(l*f(l), l)^2)); $'ds(l)=ds(l)
```

$$ds(l) = \frac{\sqrt{9 + 36l^2 - 36l^3 - 36l^5 + 36l^6 + 9l^8}}{\sqrt{1 + 4l^3 + 6l^6 + 4l^9 + l^{12}}}$$

```
>$integrate(ds(l), l, 0, 1)
```

$$\int_0^1 \frac{\sqrt{9 + 36l^2 - 36l^3 - 36l^5 + 36l^6 + 9l^8}}{\sqrt{1 + 4l^3 + 6l^6 + 4l^9 + l^{12}}} dl$$

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral tersebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
>2*integrate("ds(x)", 0, 1)
```

4.91748872168

```
>2*romberg(&ds(x), 0, 1)// perintah Euler lain untuk menghitung nilai hampiran integral yang sama
```

4.91748872168

Perhitungan di atas dapat dilakukan untuk sebarang fungsi x dan y dengan mendefinisikan fungsi EMT, misalnya kita beri nama panjangkurva. Fungsi ini selalu memanggil Maxima untuk menurunkan fungsi yang diberikan.

```
>function panjangkurva(fx,fy,a,b) ...
  ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");
  return romberg(ds,a,b);
endfunction
```

```
>panjangkurva("x","x^2",-1,1) // cek untuk menghitung panjang kurva parabola sebelumnya
```

2.95788571509

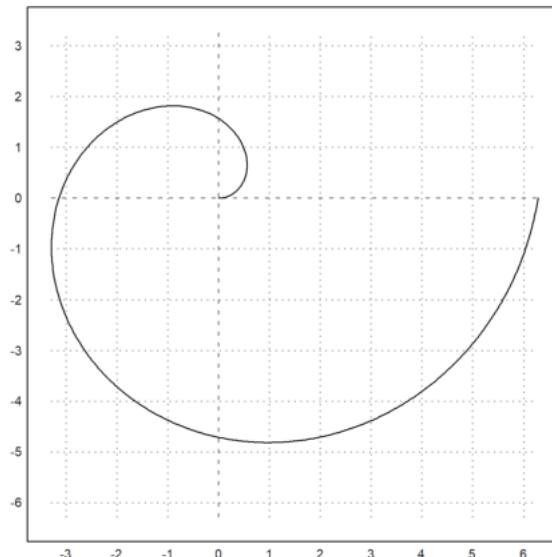
Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)"),mxm("x*f(x)"),0,1) // cek contoh terakhir, bandingkan hasilnya!
```

4.91748872168

Kita hitung panjang spiral Archimedes berikut ini dengan fungsi tersebut.

```
>plot2d("x*cos(x)", "x*sin(x)", xmin=0, xmax=2*pi, square=1):
```



```
>panjangkurva("x*cos(x)", "x*sin(x)", 0, 2*pi)
```

21.2562941482

Berikut kita definisikan fungsi yang sama namun dengan Maxima, untuk perhitungan eksak.

```
>&kill(ds,x,fx,fy)
```

done

```
>function ds(fx,fy) &=& sqrt(diff(fx,x)^2+diff(fy,x)^2)
```

$$\sqrt{(\cos x + x \sin x)^2 + (\sin x - x \cos x)^2}$$

```
>sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk menghitung panjang kurva terakhir di atas
```

$$\sqrt{(x \cos x + \sin x)^2 + (\cos x - x \sin x)^2}$$

```
>$sol | trigreduce | expand, $integrate(% ,x,0,2*pi), %()
```

$$\sqrt{1+x^2}$$

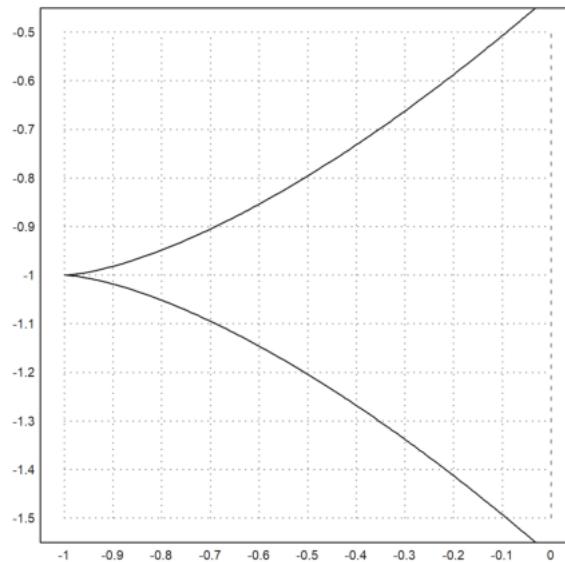
$$\frac{2\pi\sqrt{1+4\pi^2} + \operatorname{asinh}(2\pi)}{2}$$

21.2562941482

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.

Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
>plot2d("3*x^2-1", "3*x^3-1", xmin=-1/sqrt(3), xmax=1/sqrt(3), square=1):
```



```
>sol &= radcan(ds(3*x^2-1,3*x^3-1)); $sol
```

$$3x \sqrt{4 + 9x^2}$$

```
>$showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%); // panjang kurva di atas
```

$$3 \int_0^{\frac{1}{\sqrt{3}}} x \sqrt{4 + 9x^2} dx = 3 \left( -\frac{8}{27} + \frac{7\frac{3}{2}}{27} \right)$$

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{4.0 + 9.0x^2} dx = 2.337835372767141$$

### Sikloid

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar ke kanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah  $r$ . Posisi titik pusat lingkaran pada saat  $t$  adalah:

$$(rt, r).$$

Misalkan posisi titik pada lingkaran tersebut mula-mula  $(0,0)$  dan posisinya pada saat  $t$  adalah:

$$(r(t - \sin(t)), r(1 - \cos(t))).$$

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika  $t=0$ ,  $t=\pi/2$ ,  $t=r*\pi$ .

```
>x &= r*(t-sin(t))
```

$$r (t - \sin(t))$$

```
>y &= r*(1-cos(t))
```

$$r (1 - \cos(t))$$

Berikut kita gambar sikloid untuk  $r=1$ .

```
>ex &= x-sin(x); ey &= 1-cos(x); aspect(1);
>plot2d(ex,ey,xmin=0,xmax=4pi,square=1); ...
plot2d("2+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
plot2d([2,ex(2)], [1,ey(2)],color=red,>add); ...
plot2d(ex(2),ey(2),>points,>add,color=red); ...
plot2d("2pi+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
plot2d([2pi,ex(2pi)], [1,ey(2pi)],color=red,>add); ...
plot2d(ex(2pi),ey(2pi),>points,>add,color=red);
```

Error :  $r*(t-\sin(t))-\sin(r*(t-\sin(t)))$  does not produce a real or column vector

Error generated by error() command

adaptiveeval:

```
error(f$" does not produce a real or column vector");
Try "trace errors" to inspect local variables after errors.
plot2d:
    dw/n,dw/n^2,dw/n;args());
```

Berikut dihitung panjang lintasan untuk 1 putaran penuh. (Jangan salah menduga bahwa panjang lintasan 1 putaran penuh sama dengan keliling lingkaran!)

```
>ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds) // elemen panjang kurva sikloid
```

```
Maxima said:
diff: second argument must be a variable; found r*(t-sin(t))
-- an error. To debug this try: debugmode(true);

Error in:
ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds ...
^
```

```
>ds &= trigsimp(ds); $ds
>$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu putaran penuh
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found r*(t-sin(t))
#0: showev(f='integrate(ds,r*(t-sin(t)),0,2*pi))
-- an error. To debug this try: debugmode(true);

Error in:
$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid sat ...
^
```

```
>integrate(mxm("ds"),0,2*pi) // hitung secara numerik
```

```
Illegal function result in map.
%evalexpression:
if maps then return %mapexpression1(x,f$;args());
gauss:
if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());
adaptivegauss:
t1=gauss(f$,c,c+h;args(),=maps);
Try "trace errors" to inspect local variables after errors.
integrate:
return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
```

```
>romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik
```

Wrong argument!

Cannot combine a symbolic expression here.  
Did you want to create a symbolic expression?  
Then start with &.

Try "trace errors" to inspect local variables after errors.
romberg:
if cols(y)==1 then return y\*(b-a); endif;
Error in:
romberg(mxm("ds"),0,2\*pi) // cara lain hitung secara numerik ...
^

Perhatikan, seperti terlihat pada gambar, panjang sikloid lebih besar daripada keliling lingkarannya, yakni:

### Kurvatur (Kelengkungan) Kurva

image: Osculating.png

Aslinya, kelengkungan kurva diferensial (yakni, kurva mulus yang tidak lancip) di titik P didefinisikan melalui lingkaran oskulasi (yaitu, lingkaran yang melalui titik P dan terbaik memperkirakan, paling banyak menyentuh kurva di sekitar P). Pusat dan radius kelengkungan kurva di P adalah pusat dan radius lingkaran oskulasi. Kelengkungan adalah kebalikan dari radius kelengkungan:

$$\kappa = \frac{1}{R}$$

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar  $2\pi$  sejauh  $2\pi R$ .)

Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekivalen.

#### Definisi Kurvatur dengan Fungsi Parametrik Panjang Kurva

Setiap kurva diferensiabel dapat dinyatakan dengan persamaan parametrik terhadap panjang kurva s:

$$\gamma(s) = (x(s), y(s)),$$

dengan x dan y adalah fungsi riil yang diferensiabel, yang memenuhi:

$$\|\gamma'(s)\| = \sqrt{x'(s)^2 + y'(s)^2} = 1.$$

Ini berarti bahwa vektor singgung

$$\mathbf{T}(s) = (x'(s), y'(s))$$

memiliki norm 1 dan merupakan vektor singgung satuan.

Apabila kurvanya memiliki turunan kedua, artinya turunan kedua x dan y ada, maka  $\mathbf{T}'(s)$  ada. Vektor ini merupakan normal kurva yang arahnya menuju pusat kurvatur, norm-nya merupakan nilai kurvatur (kelengkungan):

$$\begin{aligned}\mathbf{T}(s) &= \gamma'(s), \\ \mathbf{T}^2(s) &= 1 \text{ (konstanta)} \Rightarrow \mathbf{T}'(s) \cdot \mathbf{T}(s) = 0 \\ \kappa(s) &= \|\mathbf{T}'(s)\| = \|\gamma''(s)\| = \sqrt{x''(s)^2 + y''(s)^2}.\end{aligned}$$

Nilai

$$R(s) = \frac{1}{\kappa(s)}$$

disebut jari-jari (radius) kelengkungan kurva.

Bilangan riil

$$k(s) = \pm \kappa(s)$$

disebut nilai kelengkungan bertanda.

Contoh:

Akan ditentukan kurvatur lingkaran

$$x = r \cos t, \quad y = r \sin t.$$

```
>fx &= r*cos(t); fy &= r*sin(t);
>&assume(t>0,r>0); s &=integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s // elemen panjang kurva, panjang busur l
```

r t

```
>&kill(s); fx &= r*cos(s/r); fy &= r*sin(s/r); // definisi ulang persamaan parametrik terhadap s dengan substitusi t
>k &= trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur lingkaran dengan menggunakan definisi di
```

$$\frac{1}{r}$$

Untuk representasi parametrik umum, misalkan

$$x = x(t), \quad y = y(t)$$

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

$$\begin{aligned}\kappa &= \frac{d\phi}{ds} = \frac{\frac{d\phi}{dt}}{\frac{ds}{dt}} \quad (\phi \text{ adalah sudut kemiringan garis singgung dan } s \text{ adalah panjang kurva}) \\ &= \frac{\frac{d\phi}{dt}}{\sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2}} = \frac{\frac{d\phi}{dt}}{\sqrt{x'(t)^2 + y'(t)^2}}.\end{aligned}$$

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

$$\sec^2 \phi \frac{d\phi}{dt} = \frac{d}{dt} (\tan \phi) = \frac{d}{dt} \left( \frac{dy}{dx} \right) = \frac{d}{dt} \left( \frac{dy/dt}{dx/dt} \right) = \frac{d}{dt} \left( \frac{y'(t)}{x'(t)} \right) = \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2}.$$

$$\begin{aligned}\frac{d\phi}{dt} &= \frac{1}{\sec^2 \phi} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1 + \tan^2 \phi} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1 + \left(\frac{y'(t)}{x'(t)}\right)^2} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2 + y'(t)^2}.\end{aligned}$$

Jadi, rumus kurvatur untuk kurva parametrik

$$x = x(t), \quad y = y(t)$$

adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}.$$

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kutub

$$x = r(\theta) \cos \theta, \quad y = r(\theta) \sin \theta,$$

maka rumus kurvurnya adalah

$$\kappa(\theta) = \frac{r(\theta)^2 + 2r'(\theta)^2 - r(\theta)r''(\theta)}{(r'(\theta)^2 + r''(\theta)^2)^{3/2}}.$$

(Silakan Anda turunkan rumus tersebut!)

Contoh:

Lingkaran dengan pusat (0,0) dan jari-jari r dapat dinyatakan dengan persamaan parametrik

$$x = r \cos t, \quad y = r \sin t.$$

Nilai kelengkungan lingkaran tersebut adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}} = \frac{r^2}{r^3} = \frac{1}{r}.$$

Hasil cocok dengan definisi kurvatur suatu kelengkungan.

Kurva

$$y = f(x)$$

dapat dinyatakan ke dalam persamaan parametrik

$$x = t, \quad y = f(t), \quad \text{dengan } x'(t) = 1, \quad x''(t) = 0,$$

sehingga kurvurnya adalah

$$\kappa(t) = \frac{y''(t)}{(1 + y'(t)^2)^{3/2}}.$$

Contoh:

Akan ditentukan kurvatur parabola

$$y = ax^2 + bx + c.$$

```
>function f(x) &= a*x^2+b*x+c; $y=f(x)
```

$$r (1 - \cos t) = c + a r^2 (t - \sin t)^2 + b r (t - \sin t)$$

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) // kelengkungan parabola
```

Maxima said:

```
diff: second argument must be a variable; found r*(t-sin(t))
-- an error. To debug this try: debugmode(true);
```

Error in:

```
... (x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) ...
```

```
>function f(x) &= x^2+x+1; $y=f(x) // akan kita plot kelengkungan parabola untuk a=b=c=1
```

$$r (1 - \cos t) = 1 + r^2 (t - \sin t)^2 + r (t - \sin t)$$

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) // kelengkungan parabola
```

```
Maxima said:
diff: second argument must be a variable; found r*(t-sin(t))
-- an error. To debug this try: debugmode(true);

Error in:
... (x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) ...
^
```

Berikut kita gambar parabola tersebut beserta kurva kelengkungan, kurva jari-jari kelengkungan dan salah satu lingkaran oskulasi di titik puncak parabola. Perhatikan, puncak parabola dan jari-jari lingkaran oskulasi di puncak parabola adalah

$$(-1/2, 3/4), \quad 1/k(2) = 1/2,$$

sehingga pusat lingkaran oskulasi adalah  $(-1/2, 5/4)$ .

```
>plot2d(["f(x)", "k(x)", -2, 1, color=[blue, red]); plot2d("1/k(x)", -1.5, 1, color=green, >add); ...
plot2d("-1/2+1/k(-1/2)*cos(x)", "5/4+1/k(-1/2)*sin(x)", xmin=0, xmax=2pi, >add, color=blue):
```

```
Error : f(x) does not produce a real or column vector
```

```
Error generated by error() command
```

```
%ploteval:
error(f$|" does not produce a real or column vector");
adaptiveevalone:
s=%ploteval(g$,t;args());
Try "trace errors" to inspect local variables after errors.
plot2d:
dw/n,dw/n^2,dw/n,auto;args());
```

Untuk kurva yang dinyatakan dengan fungsi implisit

$$F(x, y) = 0$$

dengan turunan-turunan parsial

$$F_x = \frac{\partial F}{\partial x}, \quad F_y = \frac{\partial F}{\partial y}, \quad F_{xy} = \frac{\partial}{\partial y} \left( \frac{\partial F}{\partial x} \right), \quad F_{xx} = \frac{\partial}{\partial x} \left( \frac{\partial F}{\partial x} \right), \quad F_{yy} = \frac{\partial}{\partial y} \left( \frac{\partial F}{\partial y} \right),$$

berlaku

$$F_x dx + F_y dy = 0 \text{ atau } \frac{dy}{dx} = -\frac{F_x}{F_y},$$

sehingga kurvturnya adalah

$$\kappa = \frac{F_y^2 F_{xx} - 2F_x F_y F_{xy} + F_x^2 F_{yy}}{(F_x^2 + F_y^2)^{3/2}}.$$

(Silakan Anda turunkan sendiri!)

Contoh 1:  
Parabola

$$y = ax^2 + bx + c$$

dapat dinyatakan ke dalam persamaan implisit

$$ax^2 + bx + c - y = 0.$$

```
>function F(x,y) &=a*x^2+b*x+c-y; $F(x,y)
```

$$c - r (1 - \cos t) + ar^2 (t - \sin t)^2 + br (t - \sin t)$$

```
>Fx &= diff(F(x,y),x), Fxx &= diff(F(x,y),x,2), Fy &= diff(F(x,y),y), Fxy &= diff(diff(F(x,y),x),y), Fyy &= diff(F(x,y),y,2)
```

```
Maxima said:
diff: second argument must be a variable; found r*(t-sin(t))
-- an error. To debug this try: debugmode(true);
```

```
Error in:
Fx &= diff(F(x,y),x), Fxx &= diff(F(x,y),x,2), Fy &= diff(F(x,y) ...
^
```

```
>function k(x) &= (Fy^2*Fxx-2*Fx*Fy*Fxy+Fx^2*Fyy)/(Fx^2+Fy^2)^(3/2); $'k(x)=k(x) // kurvatur parabola tersebut
```

$$k(r(t - \sin t)) = \frac{-2Fx Fxy Fy + Fxx Fy^2 + Fx^2 Fyy}{(Fx^2 + Fy^2)^{\frac{3}{2}}}$$

Hasilnya sama dengan sebelumnya yang menggunakan persamaan parabola biasa.

## Latihan

---

- Pilih buku Kalkulus.
- Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi).
- Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah integral tentu dengan batas-batas yang menarik (Anda tentukan sendiri), seperti contoh-contoh tersebut.
- Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat diintegralkan (cari sedikitnya 3 fungsi).
- Gambar grafik fungsi dan daerah integrasinya pada sumbu koordinat yang sama.
- Gunakan integral tentu untuk mencari luas daerah yang dibatasi oleh dua kurva yang berpotongan di dua titik. (Cari dan gambar kedua kurva dan arsir (warnai) daerah yang dibatasi oleh keduanya.)
- Gunakan integral tentu untuk menghitung volume benda putar kurva  $y=f(x)$  yang diputar mengelilingi sumbu  $x$  dari  $x=a$  sampai  $x=b$ , yakni

$$V = \int_a^b \pi(f(x))^2 dx.$$

(Pilih fungsinya dan gambar kurva dan benda putar yang dihasilkan. Anda dapat mencari contoh-contoh bagaimana cara menggambar benda hasil perputaran suatu kurva.)

- Gunakan integral tentu untuk menghitung panjang kurva  $y=f(x)$  dari  $x=a$  sampai  $x=b$  dengan menggunakan rumus:

$$S = \int_a^b \sqrt{1 + (f'(x))^2} dx.$$

(Pilih fungsi dan gambar kurvanya.)

- Apabila fungsi dinyatakan dalam koordinat kutub  $x=f(r,t)$ ,  $y=g(r,t)$ ,  $r=h(t)$ ,  $x=a$  bersesuaian dengan  $t=t_0$  dan  $x=b$  bersesuaian dengan  $t=t_1$ , maka rumus di atas akan menjadi:

$$S = \int_{t_0}^{t_1} \sqrt{x'(t)^2 + y'(t)^2} dt.$$

- Pilih beberapa kurva menarik (selain lingkaran dan parabola) dari buku kalkulus.  
Nyatakan setiap kurva tersebut dalam bentuk: a. koordinat Kartesius (persamaan  $y=f(x)$ )  
b. koordinat kutub ( $r=r(\theta)$ ) c. persamaan parametrik  $x=x(t)$ ,  $y=y(t)$  d. persamaan implisit  $F(x,y)=0$
- Tentukan kurvatur masing-masing kurva dengan menggunakan keempat representasi tersebut (hasilnya harus sama).
- Gambarlah kurva asli, kurva kurvatur, kurva jari-jari lingkaran oskulasi, dan salah satu lingkaran oskulasinya.

## Barisan dan Deret

---

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan (menggunakan titik dua ":");
- menggunakan perintah "sequence" dan rumus barisan (suku ke  $-n$ );
- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create\_list" atau "makelist" untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturut-turut

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]
```

&gt;1:2:30

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
```

## Iterasi dan Barisan

EMT menyediakan fungsi `iterate("g(x)", x0, n)` untuk melakukan iterasi

$$x_{k+1} = g(x_k), \quad x_0 = x_0, \quad k = 1, 2, 3, \dots, n.$$

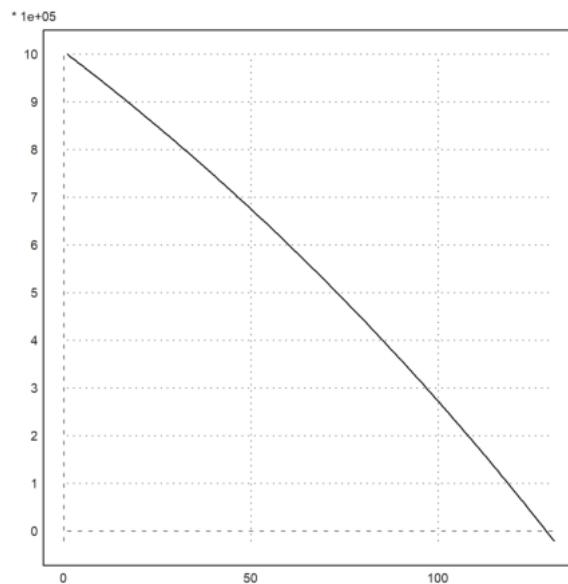
Berikut ini disajikan contoh-contoh penggunaan iterasi dan rekursi dengan EMT. Contoh pertama menunjukkan pertumbuhan dari nilai awal 1000 dengan laju pertambahan 5%, selama 10 periode.

```
>q=1.05; iterate("x*q",1000,n=10)'
```

```
1000
1050
1102.5
1157.63
1215.51
1276.28
1340.1
1407.1
1477.46
1551.33
1628.89
```

Contoh berikutnya memperlihatkan bahaya menabung di bank pada masa sekarang! Dengan bunga tabungan sebesar 6% per tahun atau 0.5% per bulan dipotong pajak 20%, dan biaya administrasi 10000 per bulan, tabungan sebesar 1 juta tanpa diambil selama sekitar 10 tahunan akan habis diambil oleh bank!

```
>r=0.005; plot2d(iterate("(1+0.8*r)*x-10000",1000000,n=130)):
```



Silakan Anda coba-coba, dengan tabungan minimal berapa agar tidak akan habis diambil oleh bank dengan ketentuan bunga dan biaya administrasi seperti di atas.

Berikut adalah perhitungan minimal tabungan agar aman di bank dengan bunga sebesar  $r$  dan biaya administrasi  $a$ , pajak bunga 20%.

```
>$solve(0.8*r*A-a,A), $% with [r=0.005, a=10]
```

$$\left[ A = \frac{5a}{4r} \right]$$

$$[A = 2500.0]$$

Berikut didefinisikan fungsi untuk menghitung saldo tabungan, kemudian dilakukan iterasi.

```
>function saldo(x,r,a) := round((1+0.8*r)*x-a,2);
>iterate({{"saldo",0.005,10}},1000,n=6)
```

```
[1000, 994, 987.98, 981.93, 975.86, 969.76, 963.64]
```

```
>iterate({{"saldo",0.005,10}},2000,n=6)
```

```
[2000, 1998, 1995.99, 1993.97, 1991.95, 1989.92, 1987.88]
```

```
>iterate({{"saldo",0.005,10}},2500,n=6)
```

```
[2500, 2500, 2500, 2500, 2500, 2500, 2500]
```

Tabungan senilai 2,5 juta akan aman dan tidak akan berubah nilai (jika tidak ada penarikan), sedangkan jika tabungan awal kurang dari 2,5 juta, lama kelamaan akan berkurang meskipun tidak pernah dilakukan penarikan uang tabungan.

```
>iterate({{"saldo",0.005,10}},3000,n=6)
```

```
[3000, 3002, 3004.01, 3006.03, 3008.05, 3010.08, 3012.12]
```

Tabungan yang lebih dari 2,5 juta baru akan bertambah jika tidak ada penarikan.

Untuk barisan yang lebih kompleks dapat digunakan fungsi "sequence()". Fungsi ini menghitung nilai-nilai  $x[n]$  dari semua nilai sebelumnya,  $x[1], \dots, x[n-1]$  yang diketahui.

Berikut adalah contoh barisan Fibonacci.

$$x_n = x_{n-1} + x_{n-2}, \quad x_1 = 1, \quad x_2 = 1$$

```
>sequence("x[n-1]+x[n-2]", [1,1],15)
```

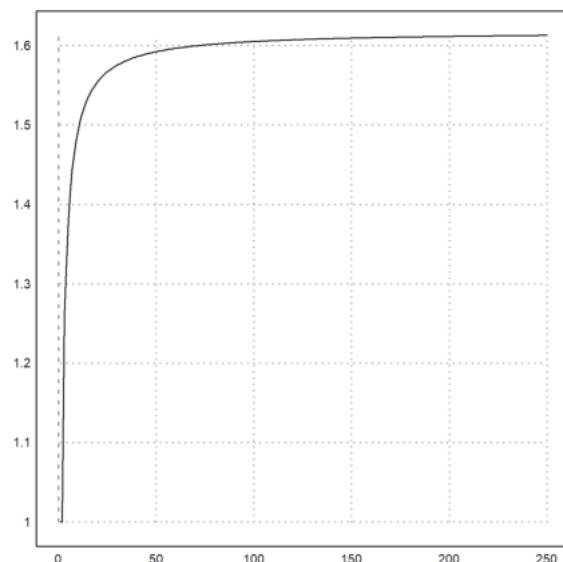
```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]
```

Barisan Fibonacci memiliki banyak sifat menarik, salah satunya adalah akar pangkat ke-n suku ke-n akan konvergen ke pecahan emas:

```
>$'(1+sqrt(5))/2=float((1+sqrt(5))/2)
```

$$\frac{1 + \sqrt{5}}{2} = 1.618033988749895$$

```
>plot2d(sequence("x[n-1]+x[n-2]", [1,1],250)^(1/(1:250))):
```



Barisan yang sama juga dapat dihasilkan dengan menggunakan loop.

```
>x=ones(500); for k=3 to 500; x[k]=x[k-1]+x[k-2]; end;
```

Rekursi dapat dilakukan dengan menggunakan rumus yang tergantung pada semua elemen sebelumnya. Pada contoh berikut, elemen ke-n merupakan jumlah  $(n-1)$  elemen sebelumnya, dimulai dengan 1 (elemen ke-1). Jelas, nilai elemen ke-n adalah  $2^{n-2}$ , 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

$$x_n = A \cdot x_{n-1},$$

dengan A suatu matriks 2x2, dan setiap  $x[n]$  merupakan matriks/vektor 2x1.

```
>A=[1,1;1,2]; function suku(x,n) := A.x[,n-1]
>sequence("suku", [1;1], 6)
```

```
Real 2 x 6 matrix
```

```
1 2 5 13 ...  
1 3 8 21 ...
```

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "matrixpower()". Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak  $\log_2(n)$ .

$$x_n = A \cdot x_{n-1} = A^2 \cdot x_{n-2} = A^3 \cdot x_{n-3} = \dots = A^{n-1} \cdot x_1.$$

```
>sequence("matrixpower(A,n).[1;1]",1,6)
```

```
Real 2 x 6 matrix
```

```
1 5 13 34 ...  
1 8 21 55 ...
```

## Spiral Theodorus

image: Spiral\_of\_Theodorus.png

Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

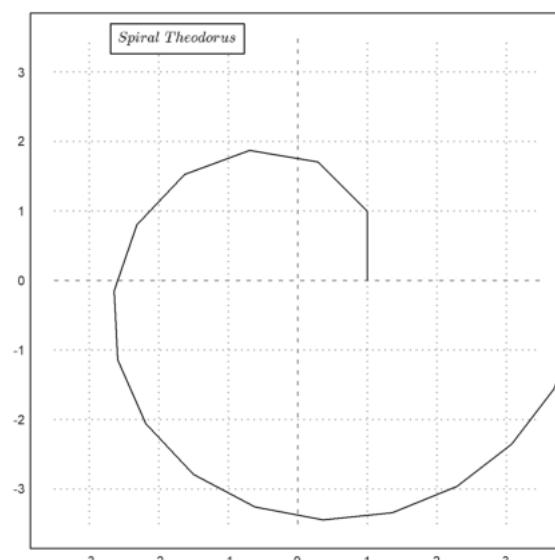
$$x_n = \left(1 + \frac{i}{\sqrt{n-1}}\right) x_{n-1}, \quad x_1 = 1,$$

yang menghasilkan barisan bilangan kompleks.

```
>function g(n) := 1+i/sqrt(n)
```

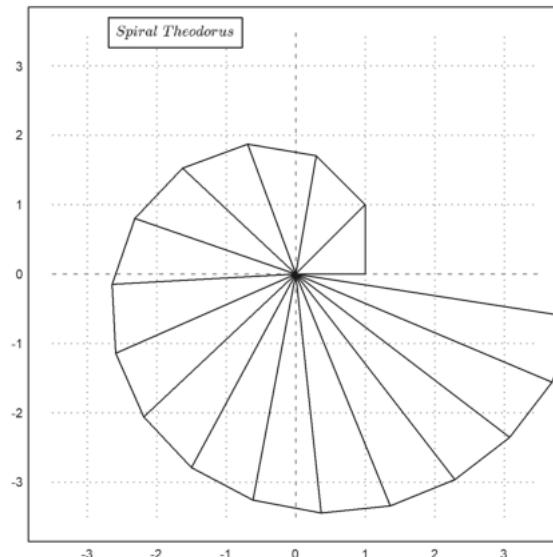
Rekursinya dapat dijalankan sebanyak 17 untuk menghasilkan barisan 17 bilangan kompleks, kemudian digambar bilangan-bilangan kompleksnya.

```
>x=sequence("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox(latex("Spiral\ Theodorus"),0.4):
```



Selanjutnya dihubungkan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end:
```



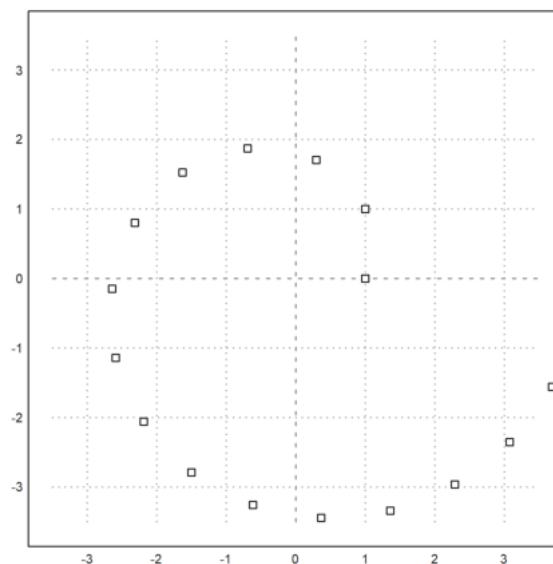
```
>
```

Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memerlukan indeks dan bilangan kompleks. Dalam hal ini digunakan vektor kolom pada bidang.

```
>function gstep (v) ...
w=[-v[2];v[1]];
return v+w/norm(w);
endfunction
```

Jika dilakukan iterasi 16 kali dimulai dari [1;0] akan didapatkan matriks yang memuat vektor-vektor dari setiap iterasi.

```
>x=iterate("gstep", [1;0], 16); plot2d(x[1],x[2],r=3.5,>points):
```



## Kekonvergenan

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```
>iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1.
```

0.739085133216

Iterasi tersebut konvergen ke penyelesaian persamaan

$x = \cos(x)$ .

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.

```
>hasil := iterate("cos(x)",~1,2~) //iterasi x(n+1)=cos(x(n)), dengan interval awal (1, 2)
```

$\sim 0.739085133211, 0.7390851332133\sim$

Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan  $x=\cos(x)$ .

```
>h=expand(hasil,100), cos(h) << h
```

$\sim 0.73908513309, 0.73908513333\sim$   
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:

$$x = \frac{1}{2} \left( x + \frac{2}{x} \right).$$

Satu-satunya solusi adalah

$$x = \sqrt{2}.$$

```
>iterate("s",~1,2~)
```

$\sim 1.41421356236, 1.41421356239\sim$

Fungsi "iterate()" juga dapat bekerja pada vektor. Berikut adalah contoh fungsi vektor, yang menghasilkan rata-rata aritmetika dan rata-rata geometri.

$$(a_{n+1}, b_{n+1}) = \left( \frac{a_n + b_n}{2}, \sqrt{a_n b_n} \right)$$

Iterasi ke-n disimpan pada vektor kolom x[n].

```
>function g(x) := [(x[1]+x[2])/2,sqrt(x[1]*x[2])]
```

Iterasi dengan menggunakan fungsi tersebut akan konvergen ke rata-rata aritmetika dan geometri dari nilai-nilai awal.

```
>iterate("g", [1;5])
```

```
2.60401  
2.60401
```

Hasil tersebut konvergen agak cepat, seperti kita cek sebagai berikut.

```
>iterate("g", [1;5], 4)
```

|   |         |         |         |         |
|---|---------|---------|---------|---------|
| 1 | 3       | 2.61803 | 2.60403 | 2.60401 |
| 5 | 2.23607 | 2.59002 | 2.60399 | 2.60401 |

Iterasi pada interval dapat dilakukan dan stabil, namun tidak menunjukkan bahwa limitnya pada batas-batas yang dihitung.

```
>iterate("g", [~1~;~5~], 4)
```

Interval 2 x 5 matrix

```
~0.9999999999999778, 1.0000000000000022~ ...  
~4.999999999999911, 5.000000000000089~ ...
```

Iterasi berikut konvergen sangat lambat.

$$x_{n+1} = \sqrt{x_n}.$$

```
>iterate("sqrt(x)", 2, 10)
```

```
[2, 1.41421, 1.18921, 1.09051, 1.04427, 1.0219, 1.01089,  
1.00543, 1.00271, 1.00135, 1.00068]
```

Kekonvergenan iterasi tersebut dapat dipercepat dengan percepatan Steffenson:

```
>steffenson("sqrt(x)", 2, 10)
```

```
[1.04888, 1.00028, 1, 1]
```

## Iterasi menggunakan Loop yang ditulis Langsung

Berikut adalah beberapa contoh penggunaan loop untuk melakukan iterasi yang ditulis langsung pada baris perintah.

```
>x=2; repeat x=(x+2/x)/2; until x^2~=2; end; x,
```

```
1.41421356237
```

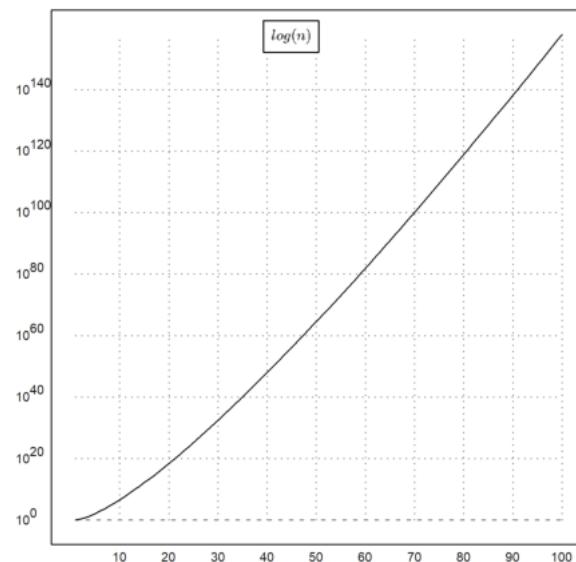
Penggabungan matriks menggunakan tanda "|" dapat digunakan untuk menyimpan semua hasil iterasi.

```
>v=[1]; for i=2 to 8; v=v|(v[i-1]*i); end; v,
```

```
[1, 2, 6, 24, 120, 720, 5040, 40320]
```

hasil iterasi juga dapat disimpan pada vektor yang sudah ada.

```
>v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]*i; end; ...  
plot2d(v,logplot=1); textbox(latex(&log(n)),x=0.5):
```



```
>A =[0.5,0.2;0.7,0.1]; b=[2;2]; ...
x=[1;1]; repeat xnew=A.x-b; until all(xnew~==x); x=xnew; end; ...
x,
```

-7.09677  
-7.74194

## Iterasi di dalam Fungsi

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.

Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

```
>function map hiter(f$,x0) ...
x=x0;
maxiter=0;
repeat
  xnew=f$(x);
  maxiter=maxiter+1;
  until xnew~==x;
  x=xnew;
end;
return maxiter;
endfunction
```

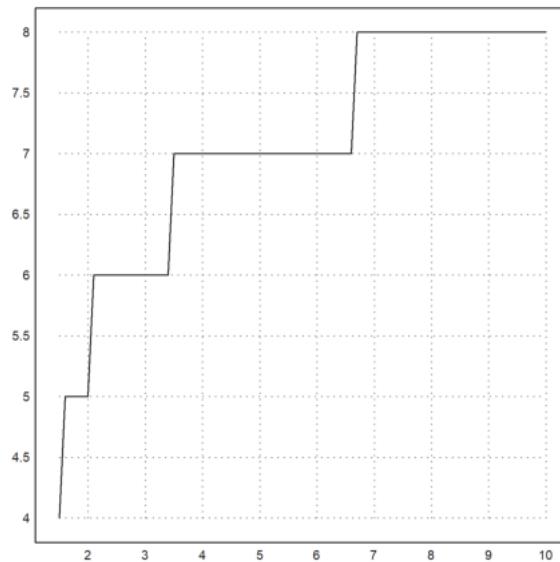
Misalnya, berikut adalah iterasi untuk mendapatkan hampiran akar kuadrat 2, cukup cepat, konvergen pada iterasi ke-5, jika dimulai dari hampiran awal 2.

```
>hiter("(x+2/x)/2",2)
```

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, 5, 6, 6, 6, 6]
```

```
>x[nonzeros(differences(hasil))]
```

```
[1.5, 2, 3.4, 6.6]
```

maksimum iterasi sampai konvergen meningkat pada saat nilai awalnya 1.5, 2, 3.4, dan 6.6.

Contoh berikutnya adalah metode Newton pada polinomial kompleks berderajat 3.

```
>p &= x^3-1; newton &= x-p/diff(p,x); $newton
```

```
Maxima said:  
diff: second argument must be a variable; found r*(t-sin(t))  
-- an error. To debug this try: debugmode(true);  
  
Error in:  
p &= x^3-1; newton &= x-p/diff(p,x); $newton ...  
^
```

Selanjutnya didefinisikan fungsi untuk melakukan iterasi (aslinya 10 kali).

```
>function iterasi(f$,x,n=10) ...  
loop 1 to n; x=f$(x); end;  
return x;  
endfunction
```

Kita mulai dengan menentukan titik-titik grid pada bidang kompleksnya.

```
>r=1.5; x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z);
```

Berikut adalah akar-akar polinomial di atas.

```
>z=&solve(p) ()
```

```
Maxima said:  
solve: more unknowns than equations.  
Unknowns given :  
[t,r]  
Equations given:  
[-1+r^3*(t-sin(t))^3]  
-- an error. To debug this try: debugmode(true);  
  
Error in:
```

```
z=&solve(p) () ...
^
```

Untuk menggambar hasil iterasinya, dihitung jarak dari hasil iterasi ke-10 ke masing-masing akar, kemudian digunakan untuk menghitung warna yang akan digambar, yang menunjukkan limit untuk masing-masing nilai awal.

Fungsi plotrgb() menggunakan jendela gambar terkini untuk menggambar warna RGB sebagai matriks.

```
>C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1)); ...
plot2d(None,-r,r,-r,r); plotrgb(C):
```

```
Wrong argument.
Cannot use a string here.

Error in:
C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1)) ...
```

## Iterasi Simbolik

Seperti sudah dibahas sebelumnya, untuk menghasilkan barisan ekspresi simbolik dengan Maxima dapat digunakan fungsi makelist().

```
>&powerdisp:true // untuk menampilkan deret pangkat mulai dari suku berpangkat terkecil
```

```
true
```

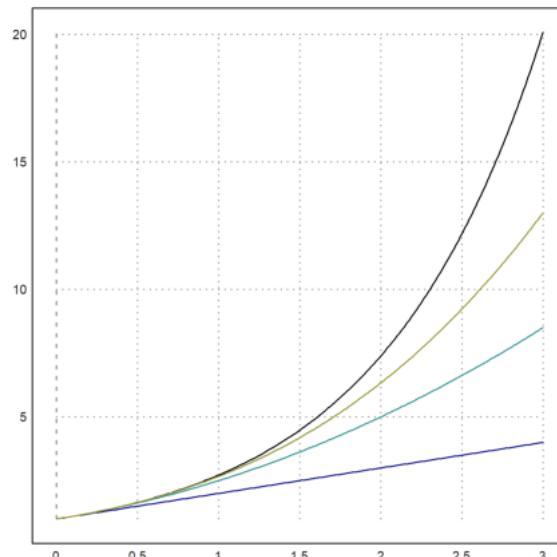
```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Taylor untuk e^x
```

```
Maxima said:
taylor: r*(t-sin(t)) cannot be a variable.
-- an error. To debug this try: debugmode(true);

Error in:
deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // baris ...
^
```

Untuk mengubah barisan deret tersebut menjadi vektor string di EMT digunakan fungsi mxm2str(). Selanjutnya, vektor string/ekspresi hasilnya dapat digambar seperti menggambar vektor ekspresi pada EMT.

```
>plot2d("exp(x)",0,3); // plot fungsi aslinya, e^x
>plot2d(mxm2str("deret"),>add,color=4:6): // plot ketiga deret taylor hampiran fungsi tersebut
```

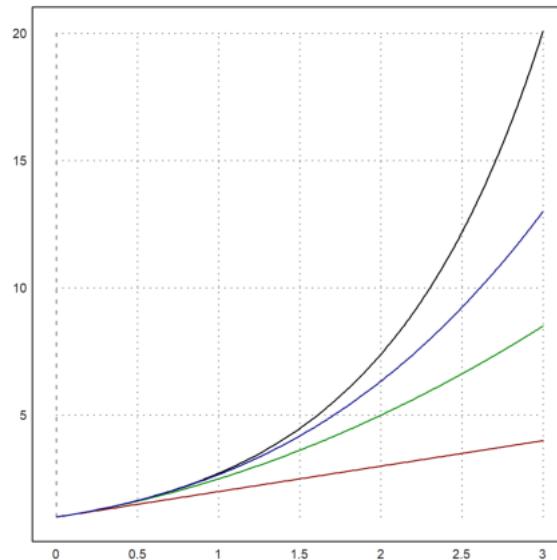


Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

```
>$deret[3]
```

$$1 + \frac{r^2 (t - \sin t)^2}{2} + \frac{r^3 (t - \sin t)^3}{6} + r (t - \sin t)$$

```
>plot2d(["exp(x)", &deret[1], &deret[2], &deret[3]], 0, 3, color=1:4):
```



```
>$sum(sin(k*x)/k, k, 1, 5)
```

$$\sin(r(t - \sin t)) + \frac{\sin(2r(t - \sin t))}{2} + \frac{\sin(3r(t - \sin t))}{3} + \frac{\sin(4r(t - \sin t))}{4} + \frac{\sin(5r(t - \sin t))}{5}$$

Berikut adalah cara menggambar kurva

$$y = \sin(x) + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \dots$$

```
>plot2d(&sum(sin((2*k+1)*x)/(2*k+1), k, 0, 20), 0, 2pi):
```

```
Error : sin(r*(t-sin(t)))+sin(3*r*(t-sin(t)))/3+sin(5*r*(t-sin(t)))/5+sin(7*r*(t-sin(t)))/7+sin(9*r*(t-sin(t)))
```

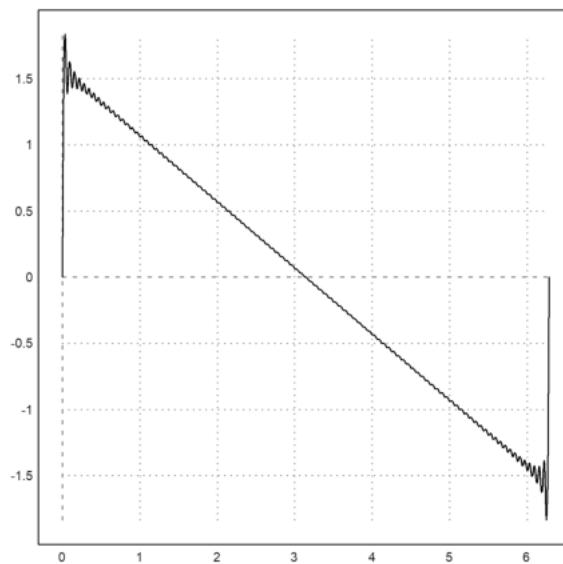
```
Error generated by error() command
```

```
%ploteval:  
error(f$|" does not produce a real or column vector");  
adaptiveevalone:  
s=%ploteval(g$,t;args());  
Try "trace errors" to inspect local variables after errors.  
plot2d:  
dw/n,dw/n^2,dw/n,auto;args());
```

Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan kita akan menggambar kurva

$$y = \sum_{k=1}^{100} \frac{\sin(kx)}{k}, \quad 0 \leq x \leq 2\pi.$$

```
>x=linspace(0,2pi,1000); k=1:100; y=sum(sin(k*x')/k)'; plot2d(x,y):
```



## Tabel Fungsi

Terdapat cara menarik untuk menghasilkan barisan dengan ekspresi Maxima. Perintah mxmtable() berguna untuk menampilkan dan menggambar barisan dan menghasilkan barisan sebagai vektor kolom.

Sebagai contoh berikut adalah barisan turunan ke-n  $x^n$  di  $x=1$ .

```
>mxmtable("diffat(x^n,x=1,n)","n",1,8,frac=1);
```

```
Maxima said:
diff: second argument must be a variable; found r*(t-sin(t))
#0: diffat(expr=r^(r*(t-sin(t)))* (t-sin(t))^(r*(t-sin(t))),x=[r*(t-sin(t)) = 1,1])
-- an error. To debug this try: debugmode(true);

%mxmtable:
return mxm("@expr,@var=@value")();
Try "trace errors" to inspect local variables after errors.
mxmtable:
y[#,1]=%mxmtable(expr,var,x[#]);
```

```
>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:menghitung deret secara simbolik
```

$$\sum_{k=1}^n k = \frac{n(1+n)}{2}$$

```
>$'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))
```

$$\sum_{k=0}^{\infty} \frac{1}{k+3^k} = \sum_{k=0}^{\infty} \frac{1}{k+3^k}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai ekspresi
```

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

```
>$'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^{-1+k}}{k} = - \sum_{x=1}^{\infty} \frac{(-1)^x}{x}$$