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))))
```

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

Baris Perintah

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

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

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

```
16.7551608191
```

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

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

```
50.2654824574
100.530964915
```

Garis perintah dilaksanakan dalam urutan pengembalian pengguna. Jadi anda mendapatkan nilai baru setiap kali anda menjalankan baris kedua.

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

```
0.540302305868
```

```
>x := cos(x)//hasil dari cos(cos(1))
```

```
0.857553215846
```

Jika dua baris terhubung dengan "..." maka kedua baris tersebut akan selalu dieksekusi secara bersamaan. Contoh lain:

```
>y=12; ...
>(y^2+6)/3,
```

```
50
```

```
>a=1; b=2; c=3; ...
>((24*(a^10)*(b^(-8))*(c^7))/12*(a^6)*(b^(-3))*c^5)^(-5)
```

```
0
```

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

```
1.41666666667
1.41421568627
1.41421356237
```

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

Untuk melipat semua baris multi, tekan Ctrl+L. Kemudian baris-baris berikutnya hanya akan terlihat jika salah satunya memiliki fokus. Untuk melipat satu baris multi, mulailah baris pertama dengan "%+".

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

Sebuah baris yang dimulai dengan %% akan sepenuhnya tidak terlihat.

```
81
```

Contoh lain

```
0
```

Euler mendukung perulangan dalam baris perintah, selama perulangan tersebut cukup untuk satu baris atau beberapa baris. Di dalam program, pembatasan ini tidak berlaku, tentu saja. Untuk informasi lebih lanjut, konsultasikan pengenalan berikut ini.

```
>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
```

Boleh menggunakan beberapa baris. Pastikan baris berakhir dengan "...".

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

```
1.41421356237
```

Struktur kondisional juga berfungsi.

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

```
Thought so!
```

Contoh lain

```
>if (((2^6)*(2^(-3)))/((2^10)/2^(-8)))<1; then "yes", endif;
```

```
yes
```

Ketika Anda menjalankan suatu perintah, kursor dapat berada pada posisi apa pun dalam baris perintah. Anda dapat kembali ke perintah sebelumnya atau melompat ke perintah berikutnya dengan tombol panah. Atau Anda dapat mengklik ke dalam bagian komentar di atas perintah untuk menuju ke perintah tersebut.

Ketika Anda memindahkan kursor sepanjang baris, pasangan tanda kurung buka dan tutup akan disorot. Juga, perhatikan baris status. Setelah tanda kurung buka dari fungsi sqrt(), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol enter.

```
>sqrt(sin(10°)/cos(20°))
```

```
0.429875017772
```

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

Anda dapat mengklik dua kali pada setiap perintah untuk membuka bantuan untuk perintah tersebut. Cobalah mengklik dua kali perintah 'exp' di bawah ini dalam 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 tombol shift bersama dengan tombol kursor mana pun. Selain itu, Anda dapat menyalin tanda kurung yang disorot.

Syntax Dasar

"Euler mengenal fungsi matematika yang umum digunakan. Seperti yang Anda lihat di atas, fungsi trigonometri berfungsi dalam radian atau derajat. Untuk mengkonversi ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilai tersebut, atau gunakan fungsi rad(x). Fungsi akar kuadrat disebut sqrt dalam Euler. Tentu saja, xˆ(1/2) juga mungkin.

Untuk mengatur variabel, gunakan baik "=" atau ":=". Untuk kejelasan, pengantar ini menggunakan bentuk terakhir. Spasi tidak masalah. Tetapi ada harapan adanya spasi antara perintah-perintah.

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Semicolon akan menghilangkan output dari perintah tersebut. Di akhir baris perintah, tanda koma "," diasumsikan jika tanda titik koma ";" tidak ada.

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

```
30.65625
```

Contoh lain

```
>x:=2; y:=5; (6*x*y^3)*(9*(x^4)*y^2),
```

```
5400000
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

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

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

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

```
8.77908249441
```

Contoh lain

```
>a=5; b=10; n=2; ((a*n)+(b^n))/((a^n)-(b^n)),
```

```
-1.46666666667
```

Untuk menghitung ekspresi yang rumit seperti

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

anda perlu memasukkannya dalam bentuk baris.

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

```
23.2671801626
```

Contoh lain

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

```
-5
```

Letakkan tanda kurung dengan hati-hati di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT akan membantu Anda dengan cara menyoroti ekspresi yang ditutup oleh tanda kurung penutup. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi.

Hasil dari perhitungan ini adalah bilangan pecahan. Secara default, bilangan ini dicetak dengan sekitar 12 digit akurasi. Dalam baris perintah berikutnya, kita juga akan mempelajari bagaimana kita dapat merujuk ke hasil sebelumnya dalam baris yang sama

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

```
0.47619047619
10/21
```

Contoh lain

```
>1/5+1/10-1/2, fraction %
```

```
-0.2
-1/5
```

Sebuah perintah Euler dapat berupa ekspresi atau perintah primitif. Sebuah ekspresi terdiri dari operator dan fungsi. Jika diperlukan, ekspresi harus mengandung tanda kurung untuk memaksakan urutan eksekusi yang benar. Dalam keraguan, menetapkan tanda kurung adalah ide yang baik. Perlu diingat bahwa EMT menampilkan tanda kurung pembuka dan penutup saat mengedit baris perintah.

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

```
14.4978445072
```

Operator numerik dalam Euler termasuk

operator plus uner atau unary
operator minus uner atau unary
*, /
. produk matriks
a^b pangkat untuk a positif atau b integer (a**b juga berfungsi)
n! operator faktorial
dan banyak lagi.

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

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

Beberapa perintah memiliki alias, misalnya ln untuk log.

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

```
2
0.5
```

```
>sin(30°)
```

```
0.5
```

Contoh lain

$$log_3(81*27)$$

```
>logbase(81*27,3)
```

```
7
```

Contoh lain

$$\ln 5/2$$

```
>ln(5/2)
```

```
0.916290731874
```

Pastikan untuk menggunakan tanda kurung (tanda kurung bulat) setiap kali ada keraguan tentang urutan eksekusi! Yang berikut ini tidak sama dengan (2^3)^4, yang adalah default untuk 2^3^4 dalam EMT (beberapa sistem numerik melakukannya sebaliknya).

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

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

Bilangan Real

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

Catatan tambahan:
Bilangan real meliputi bilangan rasional, seperti bilangan bulat 42 dan pecahan -23/129, dan bilangan irasional, seperti

$$\sqrt{2} dan \pi.$$

Bilangan real juga dapat dilambangkan sebagai salah satu titik dalam garis bilangan.

```
>longest 1/3
```

```
0.3333333333333333
```

Contoh lain

```
>longest 13/17
```

```
0.7647058823529411
```

```
>shortest 13/17
```

```
0.76
```

```
>longest 10/3
```

```
3.333333333333333
```

Representasi ganda internal menggunakan 8 byte.

```
>printdual(1/3)//Mencetak bilangan real x dengan mantisa ganda.
```

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

```
>printhex(1/3)//Mencetak bilangan real x dengan mantisa heksadesimal.
```

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

Contoh lain

```
>printdual(13/17)
```

```
1.1000011110000111100001111000011110000111100001111000*2^-1
```

```
>printhex(10/3)
```

```
3.5555555555556*16^0
```

Dalam Euler, string didefinisikan dengan "..."

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

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

Contoh lain

```
>"Nama saya Wahyu Rananda Westri."
```

```
Nama saya Wahyu Rananda Westri.
```

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

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

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

Contoh lain

```
>"Nama saya Wahyu Rananda Westri. " + "Saya sekarang berumur" +" " + 20 + " " + "tahun."
```

```
Nama saya Wahyu Rananda Westri. Saya sekarang berumur 20 tahun.
```

Fungsi print juga mengkonversi angka menjadi string. Ini dapat mengambil sejumlah digit dan sejumlah tempat (0 untuk output yang padat), dan jika memungkinkan satuan.

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

```
Golden Ratio : 1.61803
```

Contoh lain

```
>"Rata-rata tinggi badan mahasiswa adalah"+ print((156+170+180+160+165)/5)
```

```
Rata-rata tinggi badan mahasiswa adalah    166.20
```

Ada sebuah string khusus yang disebut 'none', yang tidak dicetak. Ini dikembalikan oleh beberapa fungsi ketika hasilnya tidak penting. (Ini dikembalikan secara otomatis jika fungsi tersebut tidak memiliki pernyataan pengembalian.)

```
>none
```

Untuk mengkonversi sebuah 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
```

Contoh lain

```
>ternak:=["ayam","kambing","sapi","bebek"]
```

```
ayam
kambing
sapi
bebek
```

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

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

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

Contoh lain

```
>z=[none]; z|ternak|v
```

```
ayam
kambing
sapi
bebek
affe
charlie
bravo
```

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

String Unicode dapat digabungkan seperti string lainnya.

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

```
 = 45°
```

I

Dalam komentar, entitas yang sama seperti a, ß dll. dapat digunakan. Ini mungkin merupakan alternatif cepat untuk Latex. (Lebih banyak detail tentang 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")//strtochar adalah fungsi untuk melakukan konversi dari st
```

```
[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 sebuah string dengan entitas menjadi string Unicode dalam sebuah variabel.

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

```
We have =.
```

Juga memungkinkan untuk menggunakan entitas numerik.

```
>u"&#196;hnliches"//maksud u"&#196; merujuk pada karakter "Ä" (A dengan umlaut)
```

```
Ähnliches
```

Nilai Boolean

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

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

```
0
1
```

Catatan tambahan :
"apel"<"banana" karena dalam urutan leksikografi, "apel" berada sebelum "banana" karena "a" lebih awal dalam alfabet daripada "b". Jadi, perbandingan ini menghasilkan nilai True, yang menunjukkan bahwa "apel" kurang dari "banana" dalam urutan leksikografi.

Contoh lain:

```
>1/7>2/19
```

```
1
```

```
>"ayam">"jerapah"
```

```
0
```

Operator 'and' adalah '&&' dan 'or' adalah '||', seperti dalam bahasa C. (Kata-kata 'and' dan 'or' hanya dapat digunakan dalam kondisi 'if'.)

```
>2<E && E<3//Nilai "E" sekitar 2.71828
```

```
1
```

Contoh lain :

```
>"ayam"<"jerapah" && 1/7<2/19
```

```
0
```

```
>"ayam"<"jerapah" || 1/7<2/19
```

```
1
```

Operator boolean mengikuti aturan bahasa matriks.

```
>(1:10)>5, nonzeros(%)//nonzeroes(%) menghasilkan daftar yang berisi semua elemen dari hasil matriks
```

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

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

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

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

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

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

Contoh lain

```
>M=3:3:100
```

```
[3,  6,  9,  12,  15,  18,  21,  24,  27,  30,  33,  36,  39,  42,  45,
48,  51,  54,  57,  60,  63,  66,  69,  72,  75,  78,  81,  84,  87,
90,  93,  96,  99]
```

Format keluaran default dari EMT mencetak 12 digit. Untuk memastikan bahwa kita melihat format 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 yang penuh, gunakan perintah ”longestformat”, atau gunakan operator ”longest” untuk menampilkan hasil dalam format terpanjang.

```
>longest pi
```

```
3.141592653589793
```

Contoh lain

```
>defformat; 127/17
```

```
7.47058823529
```

```
>longest 127/17
```

```
      7.470588235294118
```

Berikut adalah representasi heksadesimal internal dari angka ganda.

```
>printhex(pi)
```

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

Contoh lain

```
>printhex(127/17)
```

```
7.7878787878788*16^0
```

Format keluaran dapat diubah secara permanen dengan perintah format.

```
>format(12,5); 1/3, pi, sin(1)//artinya totalada 12 digit angka dan 5 diantaranya berada setelah tan
```

```
0.33333
3.14159
0.84147
```

Contoh lain

```
>format(12,5); 123456789/17
```

```
7262164.05882
```

Format default adalah format(12).

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

```
0.333333333333
```

Fungsi-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
```

Contoh lain

```
>setscalarformat(3); 10/7
```

```
1.43
```

Fungsi "longestformat" juga mengatur format skalar.

```
>longestformat; pi
```

```
3.141592653589793
```

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

shortestformat
shortformat
longformat
longestformat
format(length,digits)
goodformat(length)
fracformat(length)
defformat
Akurasi internal EMT adalah sekitar 16 tempat desimal, sesuai dengan 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
```

Format default adalah defformat().

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

Ada operator-operator singkat yang hanya mencetak satu nilai. Operator "longest" akan mencetak semua digit valid dari sebuah angka.

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

```
4.934802200544679
```

Juga ada operator singkat untuk mencetak hasil dalam format pecahan. Kami telah menggunakannya di atas.

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

```
25/12
```

Karena format internal menggunakan cara biner untuk menyimpan angka, nilai 0.1 tidak akan direpresentasikan secara tepat. Kesalahan tersebut akan terakumulasi 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. Untuk kenyamanan, hasil keluaran dari angka yang sangat kecil adalah 0.

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

```
0
```

String atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang dapat dievaluasi oleh EMT. Untuk ini, gunakan tanda kurung setelah ekspresi. Jika Anda bermaksud menggunakan string sebagai ekspresi, gunakan konvensi untuk menamainya ”fx” atau ”fxy” dll. Ekspresi memiliki prioritas lebih tinggi dibandingkan fungsi.

Variabel global dapat digunakan dalam evaluasi.

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

```
12.56637061435917
```

Contoh lain

```
>b=3; fx ="b^2 + E"; longest fx()
```

```
11.71828182845904
```

Parameter diberikan kepada x, y, dan z sesuai urutan tersebut. Parameter tambahan dapat ditambahkan menggunakan parameter-parameter yang telah diberikan sebelumnya.

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

```
-0.919535764538
```

Contoh lain

```
>fx="c^2-cos(x)^2"; fx(3,b=1)
```

```
8.01991485667
```

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

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

```
36
```

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

```
>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 seperti berikut.

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

```
45
```

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

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

```
12
```

Contoh lain

```
>f &=2*x;
>function f(x) := 3*x;
>f(-1)
```

```
-3
```

Sebagai konvensi, ekspresi simbolik atau numerik sebaiknya diberi nama fx, fxy, dll. Skema penamaan ini sebaiknya tidak digunakan untuk fungsi.

```
>fx &= diff(x^x,x); $&fx//diff digunakan untuk menghitung turunan
```

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

Contoh lain

```
>fx &= diff(x^3+2,x); $&fx
```

$$3\,x^2$$

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

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

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

Contoh lain

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

```
@(a,b) a^b + b^a
145
```

Ini memungkinkan untuk memanipulasi ekspresi dalam variabel lain untuk fungsi-fungsi EMT yang membutuhkan ekspresi dalam "x".

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

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

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

```
-0.475
```

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

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

```
-0.425
```

Sebuah ekspresi tidak perlu bersifat simbolik. Ini diperlukan jika ekspresi tersebut mengandung fungsi-fungsi yang hanya dikenal dalam kernel numerik, bukan dalam Maxima.

Matematika simbolik dalam EMT dilakukan dengan bantuan Maxima. Untuk detailnya, mulai dengan tutorial berikut ini, atau telusuri referensi Maxima. Para ahli dalam Maxima harus mencatat bahwa ada perbedaan dalam sintaksis antara sintaksis asli Maxima dan sintaksis default dalam ekspresi simbolik di EMT.

Matematika simbolik terintegrasi dengan lancar ke dalam Euler dengan tanda &. Setiap ekspresi yang dimulai dengan & adalah ekspresi simbolik. Itu dievaluasi dan dicetak oleh Maxima.

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

```
>$&44!
```

$$2658271574788448768043625811014615890319638528000000000$$

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

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

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

$$2481256778$$

Contoh lain
Mari kita hitung

$$C(57,24)=\frac{57!}{33!\cdot 24!}$$

```
>$& 57!/(33!*24!)
```

$$7522327487513475$$

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

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

$$2481256778$$

Contoh lain

```
>$binomial(57,24)
```

$$7522327487513475$$

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik ganda pada fungsi tersebut. Misalnya, cobalah klik ganda pada "&binomial" dalam baris perintah sebelumnya. Ini akan membuka dokumentasi Maxima yang disediakan oleh para pengembang program tersebut.

Anda akan mengetahui bahwa yang berikut ini juga berfungsi.

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

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

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

Contoh lain
Kita akan menghitung

$$C(a,4)=\frac{a!}{(a-3)!\cdot 3!}=\frac{a(a-1)(a-2)(a-3)(a-4)!}{(a-4)!\cdot 24}=\frac{a(a-1)(a-2)(a-3)}{24}$$

```
>$binomial(a,4)
```

$$\frac{(a-3)\,(a-2)\,(a-1)\,a}{24}$$

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

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

Contoh lain

```
>$&binomial(a,5) with a=20
```

$$15504$$

Dengan cara ini, Anda dapat menggunakan solusi dari suatu persamaan dalam persamaan lainnya.

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

Seperti yang Anda lihat dalam contoh-contoh sebelumnya dan berikutnya, jika Anda memiliki LaTeX terinstal, Anda dapat mencetak ekspresi simbolik dengan LaTeX. Jika tidak, perintah berikut akan menghasilkan pesan kesalahan.

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

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

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

Contoh lain

```
>$ (7-sqrt(-16))+(2+sqrt(-9))
```

$$9 - i$$

```
>$ ((4-2*k)/(1+k))+((2-5*k)/(1+k))
```

$$\frac{4-2\,k}{k+1}+\frac{2-5\,k}{k+1}$$

Ekspresi simbolik dianalisis oleh Euler. Jika Anda membutuhkan sintaks yang kompleks dalam satu ekspresi, Anda dapat melampirkan ekspresi tersebut dalam "...". Menggunakan lebih dari satu ekspresi sederhana memungkinkan, tetapi sangat tidak disarankan.

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

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

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

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

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

Sekali lagi, % merujuk pada hasil sebelumnya.

Untuk memudahkan, kita simpan solusi ke dalam variabel simbolik. Variabel simbolik didefinisikan dengan "&=".

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

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

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

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

$$2432902008176640000$$

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

```
 8  4  2
2  3  5  7
```

```
>:: factor(20!)
```

```
 18   8   4   2
2    3   5   7   11 13 17 19
```

Contoh lain

```
>:: factor(12)
```

```
 2
2  3
```

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

```
>:: ::: av:g$ av^2;
>fx &= x^3*exp(x), $fx
```

```
 3  x
x  E
```

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

Variabel-variabel semacam itu dapat digunakan dalam ekspresi simbolik lainnya. Perhatikan bahwa dalam perintah berikut, sisi kanan dari &= dievaluasi sebelum penugasan ke Fx.

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

```
                            5
                       125 E
```

$$125\,e^5$$

```
                   18551.64488782208
```

```
>fx(5)
```

```
18551.6448878
```

Untuk evaluasi suatu ekspresi dengan nilai-nilai tertentu dari variabel, Anda dapat menggunakan operator "with".

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

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

```
                     10          5
               1000 E   - 125 E


               2.20079141499189e+7
```

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

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

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

```
>tex(fx)
```

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

Ekspresi simbolik dapat dievaluasi seperti ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

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

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

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

Penugasan juga dapat bersifat simbolik.

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

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

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

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

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

Contoh lain

```
>$solve(x^2-2*x=15,x)
```

$$[x = -3, x = 5]$$

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

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

```
1.56155281281
```

Contoh lain

```
>solve("5*x^2+3*x",2,y=15)
```

```
1.45783958312
```

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

$$[x=-3.23606797749979,x=1.23606797749979]$$

Untuk mendapatkan solusi simbolik tertentu, Anda dapat menggunakan "with" dan sebuah 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 sebuah 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]
```

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

$$2$$

Ekspresi simbolik dapat memiliki flag, yang menunjukkan perlakuan khusus dalam Maxima. Beberapa flag dapat digunakan sebagai perintah juga, sementara yang lain tidak dapat. Flag-flag ini ditempatkan setelah "|" (sebuah bentuk yang lebih baik dari "ev(...,flags)").

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

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

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

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

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

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

Functions

Dalam EMT, fungsi adalah program yang didefinisikan dengan perintah "function". Ini bisa menjadi fungsi satu baris atau fungsi multi-baris. Fungsi satu baris dapat berupa fungsi numerik atau simbolik. Fungsi satu baris numerik didefinisikan dengan ":=".

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

Untuk gambaran umum, kami menampilkan semua definisi yang mungkin untuk fungsi satu baris. Sebuah fungsi dapat dievaluasi seperti fungsi Euler bawaan lainnya.

```
>f(2)
```

```
4.472135955
```

Fungsi ini akan berfungsi untuk vektor juga, mengikuti bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi tersebut di-vektorisasi.

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

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

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

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

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

```
0.786151377757
```

Secara default, jika Anda perlu menggantikan fungsi bawaan, Anda harus menambahkan kata kunci "overwrite". Menggantikan fungsi bawaan ini berbahaya dan dapat menyebabkan masalah bagi fungsi lain yang bergantung pada mereka.

Anda masih dapat memanggil fungsi bawaan sebagai "_...", jika itu adalah fungsi inti Euler.

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

```
0.707106781187
```

Lebih baik kita menghapus penggantian definisi sin ini.

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

```
0.707106781187
```

Contoh lain

```
>function f(x):= 2*x-(9*sqrt(x))+4
>f(4)
```

```
-6
```

```
>f(7)
```

```
-5.81176179958
```

```
>function f(x):=(2*x-3)^2-5*(2*x-3)+6
>f(2)
```

```
2
```

```
>function overwrite cos (x) := _cos(x°)
>cos(90)
```

```
0
```

```
>forget cos; cos(pi/2)
```

```
0
```

Parameter Default

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini akan menggunakan nilai default.

```
>f(4)
```

```
16
```

Mengaturnya akan mengganti nilai default.

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

```
80
```

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

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

```
16
```

Contoh lain

```
>function f(x, a=3) := x^(1/2)-a*x^(1/4)+2
>f(2)
```

```
-0.153407782635
```

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

```
-1
```

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

```
2.58578643763
```

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
```

Namun, parameter yang ditetapkan akan mengesampingkan nilai global.

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

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

```
20
```

Contoh lain

```
>function f(x) := x^4 + a*x^2-5
>a=1; f(2)
```

```
15
```

```
>f(2, a:= 10)
```

```
51
```

Fungsi simbolik didefinisikan dengan "&=". Mereka didefinisikan dalam Euler dan Maxima, dan berfungsi di kedua dunia tersebut. Ekspresi yang digunakan untuk mendefinisikan dijalankan melalui Maxima sebelum definisi.

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

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

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

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

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

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

Contoh lain

```
>function f(x) &= x^4-6*x^3+8*x^2+6*x-9; $&f(x)
```

$$x^4-6\,x^3+8\,x^2+6\,x-9$$

```
>$&diff(f(x),x), $&% with x=2
```

$$4\,x^3-18\,x^2+16\,x+6$$

$$-2$$

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

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

```
178.635099908
```

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik lainnya.

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

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

Contoh lain

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

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

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

```
0.703467422498
```

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

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

```
0.703467422498
```

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

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

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

$$(x+2)^n$$

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

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

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

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

625

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$x^3-x$$

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

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

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

Dengan :=, fungsi tersebut bersifat numerik. Contoh yang baik adalah integral definitif seperti

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

Jika kita mendefinisikan ulang fungsi dengan kata kunci "map," itu dapat digunakan untuk vektor x. Secara internal, fungsi tersebut dipanggil untuk semua nilai x sekali, 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-parameternya.

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

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

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

```
2
6.7
```

Selain itu, memungkinkan untuk menggunakan parameter yang sudah diassign.

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

```
2
```

Seringkali, kita ingin menggunakan fungsi untuk vektor di satu tempat, dan untuk elemen-elemen individual di tempat lain. Hal ini memungkinkan dengan menggunakan 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 simbolis seperti itu dapat digunakan untuk variabel-variabel simbolis.

Namun, fungsi tersebut juga dapat digunakan untuk vektor numerik.

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

```
17
```

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

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

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

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

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

$$0$$

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

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

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

Untuk merangkum:

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

Contoh soal lain

```
>function A(x,n) &= (x*4-2)^(2*n); $&A(x,n)
```

$$(4\,x-2)^{2\,n}$$

```
>function B(x,n) &= (x-1)^n; $&B(x,n)
```

$$(x-1)^n$$

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

$$(4\,x-2)^8$$

$$65536\,x^8-262144\,x^7+458752\,x^6-458752\,x^5+286720\,x^4-114688\,x^3+28672\,x^2-4096\,x+256$$

```
>P(2,4)
```

```
81
```

```
>$&A(x,2)+ B(x,1), $&expand(%)
```

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

$$256\,x^4-512\,x^3+384\,x^2-127\,x+15$$

```
>$&A(x,2)-B(x,1), $&expand(%), $&factor(%)
```

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

$$256\,x^4-512\,x^3+384\,x^2-129\,x+17$$

$$256\,x^4-512\,x^3+384\,x^2-129\,x+17$$

```
>$&A(x,2)*B(x,1), $&expand(%), $&factor(%)
```

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

$$256\,x^5-768\,x^4+896\,x^3-512\,x^2+144\,x-16$$

$$16\,\left(x-1\right)\,\left(2\,x-1\right)^4$$

```
>function f(x) &= x^2+2*x
```

```
  2
 x  + 2 x
```

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

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

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

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

```
>function mylog (x,base=10) := ln(x)/ln(base);
>mylog(100), mylog(2^6.7,2)
```

```
2
6.7
```

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

```
[13.7251,  113.336]
```

Ekspresi dapat dipecahkan secara numerik dan simbolis.

Untuk menyelesaikan ekspresi sederhana dengan satu variabel, kita dapat menggunakan fungsi solve(). Ini memerlukan nilai awal untuk memulai pencarian. Secara internal, solve() menggunakan metode sekant.

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

```
1.41421356237
```

Ini juga berlaku untuk ekspresi simbolis. Ambil fungsi berikut sebagai contoh.

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

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

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

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

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

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

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

$$\left[\left[x=-\frac{c\,e}{b\,\left(d-2\right)-a\,e} , y=\frac{c\,\left(d-2\right)}{b\,\left(d-2\right)-a\,e}\right]\right]$$

Contoh lain

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

```
1.25992104989
```

```
>$&solve(x^3=2,x)
```

$$\left[x=\frac{2^{\frac{1}{3}}\,\sqrt{3}\,i-2^{\frac{1}{3}}}{2} , x=\frac{-2^{\frac{1}{3}}\,\sqrt{3}\,i-2^{\frac{1}{3}}}{2} , x=2^{\frac{1}{3}}\right]$$

```
>$&solve(x^3-2,x)
```

$$\left[x=\frac{2^{\frac{1}{3}}\,\sqrt{3}\,i-2^{\frac{1}{3}}}{2},x=\frac{-2^{\frac{1}{3}}\,\sqrt{3}\,i-2^{\frac{1}{3}}}{2},x=2^{\frac{1}{3}}\right]$$

```
>$&solve(x-(12/x)-a=0,x)
```

$$\left[x=\frac{a-\sqrt{a^2+48}}{2},x=\frac{\sqrt{a^2+48}+a}{2}\right]$$

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

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

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

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

```
0.966715594851
2
```

Menyelesaikan ekspresi simbolis dalam bentuk simbolis akan mengembalikan daftar solusi. Kami menggunakan pemecah masalah simbolis 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 tercepat untuk mendapatkan nilai numerik adalah dengan mengevaluasi solusinya secara numerik seperti ekspresi biasa.

```
>longest sol()
```

```
     -0.6180339887498949       1.618033988749895
```

Untuk menggunakan solusi secara simbolis dalam ekspresi lain, cara yang paling mudah adalah dengan menggunakan "with".

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

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

$$0$$

Menyelesaikan sistem persamaan secara simbolis dapat dilakukan dengan menggunakan vektor-vektor persamaan dan pemecah masalah simbolis solve(). Hasilnya adalah daftar dari daftar-daftar persamaan.

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

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

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

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

dengan a=3.

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

Salah satu cara untuk meneruskan parameter tambahan ke f() adalah dengan menggunakan sebuah daftar yang berisi nama fungsi dan parameter-parameternya (cara lainnya adalah menggunakan parameter semikolon).

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

```
2.54116291558
```

Ini juga berfungsi dengan ekspresi. Namun, dalam hal ini, harus digunakan elemen daftar yang diberi nama. (Lebih lanjut tentang daftar dapat ditemukan dalam tutorial tentang sintaksis EMT).

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

```
2.54116291558
```

Contoh lain

```
>qx &= 2*x^4+x^3-x^2-x; $&qx
```

$$2\,x^4+x^3-x^2-x$$

```
>solve(qx,2,y=2), qx(%)
```

```
1.10455409197
2
```

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

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

Menyelesaikan Pertidaksamaan

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

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

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

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

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

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

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

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

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

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

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

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

$$emptyset$$

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

$$universalset$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Contoh lain

```
>$&fourier_elim([x^3 -8>0],[x])
```

$$\left[2<x , x^2+2\,x+4>0\right] \lor \left[x<2 , -x^2-2\,x-4>0\right]$$

```
>$&fourier_elim([x # 1],[x])
```

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

```
>$&fourier_elim([x < 4, x > 4],[x])//tidak punya penyelesaian
```

$$emptyset$$

```
>$&fourier_elim([2*y-3*x < 7, 2*x - y < 10, 12 < y],[x,y])
```

$$\left[\frac{2\,y}{3}-\frac{7}{3}<x, x<\frac{y}{2}+5, 12<y, y<44\right]$$

```
>$&fourier_elim(((3*x + y < 6) and x < 3) or  (x - 3*y >10),[x,y])
```

$$[3\,y+10<x] \vee \left[x<min\left(3, 2-\frac{y}{3}\right)\right]$$

```
>$&fourier_elim([x^3 - 1\2 # 0],[x])
```

$$[12-x^3>0] \vee [x^3-12>0]$$

Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci mengenai bahasa matriks Euler.

Vektor dan matriks dimasukkan dengan tanda kurung siku, elemen-elemennya dipisahkan oleh koma, dan barisnya dipisahkan oleh titik koma.

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

```
             1              2
             3              4
```

Hasil perkalian matriks dilambangkan dengan titik (dot).

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

```
             3
             4
```

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

```
  [3,  4]
```

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

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

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

```
             11
             25
```

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

```
              1             0
              0             1
```

Poin utama dari bahasa matriks adalah bahwa semua fungsi dan operator bekerja pada setiap elemen secara individu.

```
>A.A
```

```
              7            10
             15            22
```

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

1	4
9	16

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

37	54
81	118

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

37	54
81	118

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

1	1
1	1

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

```
    0.333333      0.666667
        0.75             1
```

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

```
          -2
         2.5
```

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

```
          -2
         2.5
```

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

```
           1             0
           0             1
```

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

```
              1               0
              0               1
```

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

```
              1               4
              9              16
```

Ini bukanlah perkalian matriks, melainkan perkalian elemen demi elemen. Hal yang sama berlaku untuk vektor.

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

```
              9
             16
```

Jika salah satu operand adalah vektor atau skalar, maka operand tersebut diperluas dengan cara yang alami.

```
>2*A
```

```
            2             4
            6             8
```

Contohnya, jika operandnya adalah vektor kolom, elemennya diterapkan pada semua baris A.

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

```
            1             4
            3             8
```

Jika itu adalah vektor baris, maka vektor tersebut diterapkan pada semua kolom A.

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

```
            2             6
            6            12
```

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

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

```
              1             2
              1             2
```

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

```
              1             4
              3             8
```

Contoh lain

```
>C=[2,4,6,8;3,6,9,12]
```

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

```
>D=[1,2;2,1]
```

```
         1            2
         2            1
```

```
>C'
```

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

```
>inv(D)
```

```
 -0.333333     0.666667
  0.666667    -0.333333
```

```
>E=[3,2;4,3]
```

```
         3            2
         4            3
```

```
>D.E
```

```
        11            8
        10            7
```

```
>D.inv(D)
```

1	0
0	1

```
>E.E
```

17	12
24	17

```
>C^2
```

4	16	36	64
9	36	81	144

```
>power(D,4)
```

41	40
40	41

```
>D/E
```

```
 0.333333         1
      0.5  0.333333
```

```
>E\D
```

```
       -1         4
        2        -5
```

```
>C*C
```

```
        4        16        36        64
        9        36        81       144
```

```
>2*C
```

```
        4         8        12        16
        6        12        18        24
```

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

```
        2         6
        4         3
```

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

```
        3             4
        4             6
```

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

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

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

Sekali lagi, ingatlah bahwa ini bukanlah perkalian matriks!

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

```
        55
```

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

```
55
```

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

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

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

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

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

```
5
```

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

Kita mendapatkan vektor berisi 0 dan 1, di mana 1 mengindikasikan nilai benar (true).

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

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

Dari vektor seperti itu, "nonzeros" memilih elemen-elemen yang bukan nol.

Dalam kasus ini, kita mendapatkan indeks dari 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 ini untuk mengambil nilai-nilai yang sesuai dalam t.

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

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

Sebagai contoh, mari temukan semua kuadrat dari angka 1 hingga 1000 yang memiliki sisa 5 modulo 11 dan 3 modulo 13.

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

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

EMT tidak sepenuhnya efektif untuk perhitungan bilangan bulat. Ia menggunakan titik koma presisi ganda secara internal. Namun, seringkali sangat berguna.

Kita dapat memeriksa apakah sebuah bilangan adalah 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 berfungsi untuk vektor. Untuk matriks, ada mnonzeros().

```
>seed(2); A=random(3,4)//seed untuk menetapkan angka-angka acak
```

```
    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 dari elemen-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-indeks ini dapat digunakan untuk mengatur elemen-elemen ke suatu nilai tertentu.

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

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

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

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

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

Dan memungkinkan untuk mendapatkan elemen-elemen dalam bentuk vektor.

```
>mget(A,k)//mendapatkan elemen-elemen dari A dengan indeks k
```

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

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

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

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

Kita dapat menggunakan ini untuk mengekstrak nilai maksimal 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]
```

Tapi dengan mget(), kita bisa mengambil indeks dan menggunakan informasi ini untuk mengambil elemen-elemen pada posisi yang sama dari matriks lain.

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

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

Contoh lain

```
>(2:5)*(2:5)'
```

```
             4             6             8            10
             6             9            12            15
             8            12            16            20
            10            15            20            25
```

```
>(2:5).(2:5)'
```

54

```
>sum((2:5)*(2:5))
```

54

```
>(1:12)>5/2
```

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

```
>sum((1:12)>5/2)
```

10

```
>k=(2:20); k==11
```

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

```
>nonzeros(k>11)
```

```
[11,  12,  13,  14,  15,  16,  17,  18,  19]
```

```
>k[nonzeros(k<11)]
```

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

```
>r=1:200; nonzeros(mod(r,2)==1)//mencari daftar bilangan ganjil dari 1 sampai 200
```

```
[1,  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,  101,  103,  105,  107,  109,  111,
113,  115,  117,  119,  121,  123,  125,  127,  129,  131,  133,  135,
137,  139,  141,  143,  145,  147,  149,  151,  153,  155,  157,  159,
161,  163,  165,  167,  169,  171,  173,  175,  177,  179,  181,  183,
185,  187,  189,  191,  193,  195,  197,  199]
```

```
>g=mnonzeros(D<4)
```

```
            1             1
            1             2
            2             1
            2             2
```

```
>mget(E,g)
```

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

```
>ex=extrema(C)
```

2	1	8	4
3	1	12	4

Fungsi Matriks Lainnya (Membangun Matriks)

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

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

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

Demikian pula, kita dapat melampirkan sebuah matriks ke samping yang lain, 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 akan diisi dengan 0.

Ada pengecualian untuk aturan ini. Sebuah bilangan real yang dilampirkan ke 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
```

Mungkin membuat matriks dari vektor baris dan kolom.

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

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

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

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

Tujuan utamanya adalah untuk menginterpretasikan sebuah vektor dari ekspresi sebagai vektor kolom.

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

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

Untuk mendapatkan ukuran matriks A, kita dapat menggunakan fungsi-fungsi berikut.

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

```
            0             0             0             0
            0             0             0             0
2
4
[2,  4]
4
```

Untuk vektor, ada fungsi length().

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

```
9
```

Ada banyak fungsi lain yang menghasilkan matriks.

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

```
              1             1
              1             1
```

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

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

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

Juga, matriks dari angka-angka acak dapat dihasilkan dengan random (distribusi seragam) atau normal (distribusi Gaussian).

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

```
        0.66566      0.831835
          0.977      0.544258
```

Berikut adalah fungsi lain yang berguna, yang mengubah struktur 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 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() menggandakan elemen-elemen dari 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 matriks. Dengan kata lain, fungsi flipx() melakukan pembalikan horizontal.

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

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

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

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

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

Fungsi khusus adalah drop(v, i), yang menghapus elemen-elemen dengan indeks-indeks dalam 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) mengacu pada indeks-indeks elemen-elemen dalam v, bukan nilai-nilai elemen. Jika Anda ingin menghapus elemen-elemen, Anda perlu menemukan elemen-elemen tersebut terlebih dahulu. Fungsi indexof(v, x) dapat digunakan untuk menemukan elemen-elemen x dalam vektor yang telah diurutkan.

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

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

Catatan tambahan:
-indexof digunakan untuk menemukan kemunculan pertama dari x dalam vektor v
-drop digunakan untuk menghapus elemen-elemen i dari vektor baris v

Seperti yang Anda lihat, tidak masalah jika menyertakan indeks-indeks di luar jangkauan (seperti 0), indeks ganda, atau indeks yang tidak terurut.

```
>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 kita mengatur diagonal bawah (-1) menjadi 1:4.

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

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

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

Berikut adalah sebuah fungsi yang mengembalikan matriks tri-diagonal.

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

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

Diagonal dari sebuah matriks juga dapat diekstrak dari matriks itu sendiri. Untuk mendemonstrasikannya, kita merestrukturisasi vektor 1:9 menjadi matriks 3x3.

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

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

Sekarang kita dapat mengekstrak diagonalnya.

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

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

Contohnya, kita dapat membagi matriks dengan diagonalnya. Bahasa matriks akan mengurus agar vektor kolom d diterapkan pada matriks baris demi baris.

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

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

Contoh lain

```
>w=4:6; w_w
```

```
           4             5             6
           4             5             6
```

```
>W=random(3,4); W|w'
```

```
    0.208566      0.220144      0.855399     0.0288546             4
    0.259286      0.181379      0.293642      0.791497             5
   0.0155055      0.312754      0.381387      0.875381             6
```

```
>[w; w]
```

```
           4             5             6
           4             5             6
```

```
>[w',w']
```

```
           4             4
           5             5
           6             6
```

```
>"[x,x^3]"(w')
```

4	64
5	125
6	216

```
>M=zeros(3,7); rows(M), cols(M), size(M), length(M)
```

```
3
7
[3,  7]
7
```

```
>redim(1:4,2,2)
```

1	2
3	4

```
>rep(2:4, 7)
```

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

```
>multdup(2:4,7), multdup(2:4, [2])
```

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

```
>flipx(999:1005)
```

```
[1005,  1004,  1003,  1002,  1001,  1000,  999]
```

```
>rotleft(999:1005)
```

```
[1000,  1001,  1002,  1003,  1004,  1005,  999]
```

```
>drop(20:30,3)
```

```
[20,  21,  23,  24,  25,  26,  27,  28,  29,  30]
```

```
>K=id(3)
```

1	0	0
0	1	0
0	0	1

```
>setdiag(K,-1,1:4)
```

1	0	0
1	1	0
0	2	1

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

1	2	3
4	5	6
7	8	9

```
>m=getdiag(M,0)
```

[1,   5,   9]

```
>fraction M/m'
```

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

Vektorisasi

Hampir semua fungsi dalam Euler berfungsi juga untuk input matriks dan vektor, bila 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 membuat grafik fungsi (alternatifnya menggunakan ungkapan).

```
>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 dari fungsi dapat dibuat dengan mudah.

Pada contoh berikut, kami menghasilkan vektor nilai t[i] dengan selang 0,1 dari -1 hingga 1. Kemudian kami menghasilkan vektor nilai dari fungsi

$$s = t^3 - t$$

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

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

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

Contohnya, perkalian antara vektor kolom dan vektor baris akan diperluas menjadi matriks jika operator diterapkan. Dalam contoh berikut, v' adalah vektor transpos (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
```

Perlu diperhatikan bahwa ini sangat berbeda dari perkalian matriks. Perkalian matriks ditandai dengan titik "." dalam EMT.

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

```
55
```

Secara default, vektor baris akan dicetak dalam format yang ringkas.

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

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

Untuk matriks, operator khusus "." menunjukkan perkalian matriks, dan A' menunjukkan transpose. Matriks 1x1 dapat digunakan sama seperti angka riil.

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

```
5
25
```

Untuk melakukan transpose pada sebuah matriks, kita menggunakan tanda apostrof (').

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

```
             1
             2
             3
             4
```

Jadi, kita dapat menghitung perkalian matriks A dengan vektor b.

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

```
            30
            70
```

Perlu diingat bahwa v tetap merupakan vektor baris. Jadi, v'.v berbeda dari v.v'.

```
>v'.v
```

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

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

```
>v.v'
```

```
30
```

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

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

```
30
```

Operator dan fungsi mengikuti bahasa matriks Euler.

Berikut adalah ringkasan aturan-aturannya:

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

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

- Jika dua matriks memiliki dimensi yang berbeda, keduanya diperluas dengan cara yang masuk akal, sehingga memiliki ukuran yang sama.

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

Berikut adalah contoh sederhana dengan operator ^.

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

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

Berikut adalah kasus yang lebih rumit. Sebuah vektor baris dikali dengan vektor kolom akan memperluas keduanya dengan menggandakannya.

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

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

Perlu diperhatikan bahwa produk skalar menggunakan perkalian matriks, bukan *!

```
>v.v'
```

```
14
```

Ada banyak fungsi untuk matriks. Berikut adalah daftar singkat. Anda sebaiknya merujuk ke dokumentasi untuk informasi lebih lanjut tentang perintah-perintah ini.

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

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

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

Operator titik dua ":" menghasilkan vektor baris dengan jarak 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 dirujuk dengan "A[i,j]".

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

```
6
```

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

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

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

Indeks juga bisa berupa vektor baris dari indeks. ":" menunjukkan semua indeks.

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

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

Bentuk singkat untuk ":" adalah dengan menghilangkan indeks sepenuhnya.

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

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

Untuk keperluan vektorisasi, elemen-elemen dari sebuah matriks dapat diakses seolah-olah mereka adalah vektor.

```
>A{4}
```

```
4
```

Sebuah matriks juga dapat ”diratakan” (flattened), menggunakan fungsi redim(). Ini diimplementasikan dalam fungsi flatten().

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

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

Untuk menggunakan matriks untuk tabel, mari kembalikan ke format default dan hitung tabel nilai-nilai sinus dan kosinus. Perlu diingat bahwa sudutnya dalam radian secara default.

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

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

Sekarang kita akan menambahkan kolom-kolom ke dalam sebuah matriks.

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

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

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

Pada contoh berikut, kita menghitung t[j]ˆi untuk i mulai dari 1 hingga n. Kita mendapatkan sebuah matriks, di mana setiap baris merupakan tabel dari tˆi untuk satu nilai i. Artinya, matriks tersebut memiliki elemen-elemen latex: a_{i,j} = t_jˆi, \quad 1 \le j \le 101, \quad 1 \le i \le n

Sebuah fungsi yang tidak berfungsi untuk masukan vektor harus "divektorisasi". Ini dapat dicapai dengan kata kunci "map" dalam definisi fungsi. Kemudian fungsi akan dievaluasi untuk setiap elemen dari parameter vektor.

Pengintegrasi numerik integrate() hanya berfungsi untuk batas interval skalar. Jadi kita perlu melakukan vektorisasi terhadapnya.

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

Kata kunci ”map” melakukan vektorisasi pada fungsi tersebut. Fungsi ini sekarang akan berfungsi untuk vektor-vektor angka.

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

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

Contoh lain

```
>k=2:5
```

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

```
>norm(k)^2
```

```
54
```

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

```
[4,  9,  16,  25]
54
[4,  13,  29,  54]
```

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

```
           1             2
           4             5
           7             8
```

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

```
[1,  2,  4,  5,  7,  8]
[1,  2,  4,  5,  7,  8]
```

```
>defformat; w=0°:15°:180°; w=w'; deg(w)
```

```
           0
          15
          30
          45
          60
          75
          90
         105
         120
         135
         150
         165
         180
```

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

```
      0          0          1          0
     15   0.261799   0.965926   0.258819
     30   0.523599   0.866025        0.5
     45   0.785398   0.707107   0.707107
     60     1.0472        0.5   0.866025
     75      1.309   0.258819   0.965926
     90     1.5708          0          1
    105     1.8326  -0.258819   0.965926
    120     2.0944       -0.5   0.866025
    135    2.35619  -0.707107   0.707107
    150    2.61799  -0.866025        0.5
    165    2.87979  -0.965926   0.258819
    180    3.14159         -1          0
```

```
>f([2:7])
```

```
[2.05045,  13.7251,  113.336,  1241.03,  17128.1,  284713]
```

Untuk mengakses elemen matriks, gunakan notasi tanda kurung.

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

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

Kita dapat mengakses seluruh baris dari sebuah matriks.

```
>A[2]
```

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

Dalam kasus vektor baris atau vektor kolom, ini mengembalikan sebuah elemen dari vektor tersebut.

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

```
 2
```

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

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

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

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

Di sini kita ingin mendapatkan baris pertama dan kedua dari A.

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

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

Kita bahkan dapat mengurutkan ulang A menggunakan vektor indeks. Untuk menjadi lebih tepat, kita tidak mengubah A di sini, tetapi menghitung versi A yang diurutkan ulang.

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

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

Trik indeks ini juga berfungsi dengan kolom-kolom.

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

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

2	3
5	6
8	9

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

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

```
3
6
9
```

Sebagai alternatif, biarkan indeks pertama kosong.

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

2	3
5	6
8	9

Kita juga dapat mendapatkan baris terakhir dari A.

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

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

Sekarang mari kita ubah elemen-elemen A dengan memberikan sebuah submatriks dari A ke beberapa nilai. Ini sebenarnya mengubah matriks A yang tersimpan.

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

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

Kita juga dapat memberikan sebuah nilai kepada sebuah baris dari A.

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

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

Kita bahkan dapat memberikan nilai kepada sebuah submatriks jika ukurannya sesuai.

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

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

Selain itu, beberapa pintasan juga diperbolehkan.

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

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

Peringatan: Indeks yang keluar dari batas akan mengembalikan matriks kosong atau pesan kesalahan, tergantung pada pengaturan sistem. Secara default, pesan kesalahan akan ditampilkan. Namun, perlu diingat bahwa indeks negatif dapat digunakan untuk mengakses elemen-elemen matriks dengan menghitung dari akhir.

```
>A[3]
```

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

Contoh lain

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

```
             4             4             5
             5             4             6
             7             8             9
  9
```

```
>B[2]
```

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

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

```
             7             8             9
             5             4             6
             4             4             5
```

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

```
             4             5
             4             6
             8             9
```

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

0	0	5
0	0	6
7	8	9

```
>A[2]
```

```
[0,  0,  6]
```

Fungsi sort() mengurutkan vektor baris.

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

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

Seringkali penting untuk mengetahui indeks dari vektor yang sudah diurutkan dalam vektor aslinya. Ini dapat digunakan untuk mengurutkan kembali vektor lain dengan cara yang sama.

Mari kita mengacak sebuah vektor.

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

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

Indeks mengandung urutan yang tepat dari v.

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

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

Ini juga berfungsi 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 dari entri ganda agak acak.

```
>ind
```

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

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

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

```
[5,  8,  5,  2,  7,  10,  4,  4,  2,  1]
[1,  2,  4,  5,  7,  8,  10]
```

Ini juga berfungsi untuk vektor string.

```
>unique(s)
```

```
a
aa
d
e
```

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

Untuk sistem linear Ax=b, Anda dapat menggunakan algoritma Gauss, matriks invers, atau regresi linear. Operator A\b menggunakan versi algoritma Gauss.

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

```
           -4
          4.5
```

Sebagai contoh lain, kita menghasilkan sebuah matriks berukuran 200x200 dan menjumlahkan semua barisnya. Kemudian kita menyelesaikan Ax=b menggunakan matriks invers. Kita mengukur kesalahan sebagai deviasi maksimum dari semua elemen dari nilai 1, yang tentu saja adalah solusi yang benar.

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

```
    8.810729923425242e-13
```

Jika sistem tersebut tidak memiliki solusi, regresi linear akan meminimalkan norma dari kesalahan Ax-b.

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

1	2	3
4	5	6
7	8	9

Determinan matriks ini adalah 0.

```
>det(A)
```

```
0
```

Matriks Simbolis

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Untuk mengambil sebuah vektor eigen tertentu, perlu perhatian khusus pada indeksnya.

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

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

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

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

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

```
            1             4
            5             2
```

Dalam ekspresi simbolis, gunakan dengan (”with”).

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

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

Akses ke baris dari matriks simbolis berfungsi sama seperti pada matriks numerik.

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

$$[1, a]$$

Ekspresi simbolis dapat berisi sebuah penugasan, dan itu mengubah matriks A.

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

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

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

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

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

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

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

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

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

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

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

Euler juga memiliki fungsi yang kuat yaitu xinv(), yang melakukan upaya lebih besar dan memberikan hasil yang lebih akurat.

Perhatikan bahwa dengan &:= matriks B telah didefinisikan sebagai simbolik dalam ekspresi simbolik dan sebagai numerik dalam ekspresi numerik. Jadi kita dapat menggunakannya di sini.

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

```
                      1                       0
                      0                       1
```

Contohnya, nilai-nilai eigen dari A dapat dihitung secara numerik.

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

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

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

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

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

Nilai Numerik dalam Ekspresi Simbolis

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

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

```
            1        3.14159
            4              5
```

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

```
>$&A
```

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

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

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

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

Maxima juga dapat melakukan perhitungan dengan angka desimal, bahkan dengan angka desimal besar dengan 32 digit. Namun, evaluasinya akan lebih lambat.

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

$$1.4142135623730950488016887242097_B \times 10^0$$

$$1.414213562373095$$

Presisi angka desimal besar dapat diubah.

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

```
        3.14159265358979323846264338327950288419716939937510582097494\
459230781640628620899862803482534211706 8b0
```

Variabel numerik dapat digunakan dalam ekspresi simbolis menggunakan "@var".

Perlu diingat bahwa ini hanya diperlukan jika variabel tersebut telah didefinisikan dengan ":=" atau "=" sebagai variabel numerik.

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

$$-5.424777960769379$$

Demo - Tingkat Bunga

Di bawah ini, kita menggunakan Euler Math Toolbox (EMT) untuk menghitung tingkat bunga. Kita melakukannya secara numerik dan simbolis untuk menunjukkan bagaimana Euler dapat digunakan untuk menyelesaikan masalah dunia nyata.

Misalkan Anda memiliki modal awal sebesar 5000 (katakanlah dalam dolar).

```
>K=5000
```

```
5000
```

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

```
>K*1.03
```

```
5150
```

Euler akan memahami sintaks berikut juga.

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

```
5150
```

Namun, lebih mudah menggunakan faktor.

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

```
1.03
5150
```

Selama 10 tahun, kita dapat dengan mudah mengalikan faktor-faktor tersebut dan mendapatkan nilai akhir dengan tingkat bunga majemuk.

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

```
6719.58189672
```

Untuk keperluan kita, kita dapat mengatur formatnya menjadi 2 digit setelah tanda desimal.

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

```
    6719.58
```

Mari cetak itu dibulatkan menjadi 2 digit dalam sebuah 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-hasil perantara dari tahun 1 hingga tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menulis sebuah perulangan, tetapi cukup masukkan:

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

```
Real 1 x 11 matrix

    5000.00     5150.00     5304.50     5463.64     ...
```

Bagaimana cara kerja keajaiban ini? Pertama, ekspresi 0:10 mengembalikan vektor bilangan bulat.

```
>short 0:10
```

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

Kemudian semua operator dan fungsi dalam Euler dapat diterapkan pada elemen-elemen vektor secara berurutan. Jadi,

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

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

adalah vektor faktor dari q^0 hingga q^10. Ini dikalikan dengan K, dan kita mendapatkan vektor nilai.

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

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

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

Mari membandingkan dua hasil, dengan dan tanpa pembulatan.

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

```
        1271.61
      1271.6071
```

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

Cara termudah adalah fungsi iterate, yang mengulangi suatu fungsi yang diberikan sejumlah kali.

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

```
Real 1 x 11 matrix

    5000.00     5150.00     5304.50     5463.64     ...
```

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

```
>VKr'
```

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

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

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

```
5150.00
5000.00     5150.00     5304.50
```

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

Mari bandingkan elemen terakhir dari nilai-nilai yang dibulatkan dengan nilai-nilai lengkap.

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

```
6719.60
6719.58
```

Perbedaannya sangat kecil.

Sekarang kita akan menggunakan fungsi yang lebih canggih, yang menambahkan jumlah uang tertentu setiap tahunnya.

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

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

Kita bisa melihat bahwa jumlah uangnya berkurang. Tentu saja, jika kita hanya mendapatkan 150 dari bunga pada tahun pertama, tetapi mengambil 200, kita akan kehilangan uang setiap tahun.

Bagaimana kita bisa menentukan berapa tahun uangnya akan habis? Kita harus menulis perulangan untuk ini. Cara termudah adalah dengan mengulangi cukup lama.

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

```
Real 1 x 51 matrix

        5000.00      4950.00      4898.50      4845.45      ...
```

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

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

```
48.00
```

Alasan untuk ini adalah bahwa nonzeros(VKR<0) mengembalikan vektor indeks i, di mana VKR[i]<0, dan min menghitung indeks minimal.

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

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

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

```
-19.83
 47.00
```

Mari mencoba menjawab pertanyaan yang lebih ambigu. Misalkan kita tahu bahwa nilai adalah 0 setelah 50 tahun. Berapa tingkat bunga yang akan diterapkan?

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

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

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

Iterasinya sama seperti di atas

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

Tetapi kita tidak lagi menggunakan nilai global dari R dalam ekspresi kita. Fungsi-fungsi seperti iterate() memiliki trik khusus di Euler. Anda dapat melewati nilai-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 uji coba.

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

```
-19.83
```

Sekarang kita dapat menyelesaikan masalah kita.

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

```
3.15
```

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

Kita dapat menggunakan fungsi yang sama untuk menyelesaikan pertanyaan berikut: Berapa banyak yang dapat kita ambil setiap 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 untuk jumlah tahun, karena fungsi kami mengasumsikan n sebagai nilai bulat.

Solusi Simbolik untuk Masalah Tingkat Bunga

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

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

$$R + q\,K$$

Sekarang kita dapat mengulanginya.

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

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

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

Kita melihat sebuah pola. Setelah n periode kita memiliki

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

Rumus ini adalah rumus untuk jumlah geometri, yang dikenal oleh 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 pengaturan "simpsum" untuk menguranginya menjadi pecahan.

Mari 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 dengan fungsi f sebelumnya. Tetapi lebih efektif.

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

```
-19.82504734650985
-19.82504734652684
```

Sekarang kita bisa menggunakannya untuk menanyakan waktu n. Kapan modal kita habis? Tebakan awal kita adalah 30 tahun.

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

```
20.51
```

Jawaban ini mengatakan bahwa uang akan habis setelah 21 tahun.

Kita juga bisa menggunakan sisi simbolik Euler untuk menghitung rumus-rumus pembayaran.

Misalkan kita mendapatkan pinjaman sebesar K, dan membayar n pembayaran sebesar R (dimulai setelah tahun pertama) dengan sisa utang sebesar Kn (pada saat pembayaran terakhir). Rumus ini jelas:

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

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

Biasanya rumus ini dinyatakan dalam bentuk

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

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

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

Kita dapat menyelesaikan untuk tingkat R secara simbolis.

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

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

Seperti yang dapat Anda lihat dari rumusnya, fungsi ini menghasilkan kesalahan titik desimal untuk i=0. Namun, Euler tetap memplotnya.

Tentu saja, kita memiliki batasan berikut.

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

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

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

Persamaan ini juga dapat diselesaikan untuk n. Terlihat lebih bagus jika kita melakukan beberapa penyederhanaan padanya.

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

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

Sederhanakan!
Nomor 1

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

jawaban manual:

$$(\frac{24a^{10}b^{-8}c^{7}}{12a^{6}b^{-3}c^{5}})^{-5} = (\frac{2a^{4}c^{2}}{b^{5}})^{-5} =$$
$$(\frac{b^{5}}{2a^{4}c^{2}})^{5} = \frac{b^{25}}{32a^{20}c^{10}}$$

jawaban emt:

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

Nomor 2

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

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

Nomor 3
Kalkulasikan!

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

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

Sederhanakan!
Nomor 4

$$(m^{x-b}\cdot n^{x+b})^x(m^bn^{-b})^x$$

```
>$ ((((m^(x-b))*n^(x+b))^x)*((m^b)*(n^(-b))^x))
```

Nomor 5

$$[\frac{(3x^ay^b)^3}{(-3x^ay^b)^2}]^2$$

```
>$ (((3*(x^a)*(y^b))^3)/(-3*(x^a)*(y^b))^2)^2
```

R.3

Lakukan operasi yang ditunjukkan!

Nomor 1

$$(3a^2)(-7a^4)$$

jawaban manual:

$$(3a^2)(-7a^4) = (3)(-7)(a^{2+4}) = -21a^6$$

jawaban emt:

```
>$ (3*a^2)*(-7*a^4)
```

Nomor 2

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

```
>$ (2*x+3*y)^2
```

Nomor 3

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

```
>$ (x+1)*(x-1)*(x^2+1)
```

Nomor 4

$$(z+4)(z-2)$$

```
>$ (z+4)*(z-2)
```

Nomor 5

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

```
>$ (a^n+b^n)^2
```

R.4

Faktorkan!
Nomor 1

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

jawaban manual:

$$t^2 + 8t + 15 = (t + 5)(t + 3)$$

jawaban emt:

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

Nomor 2

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

```
>$&factor(y^2+12*y+27)
```

Nomor 3

$$5m^4 - 20$$

```
>$&factor(5*(m^4)-20)
```

Nomor 4

$$6x^2 - 6$$

```
>$&factor(6*x^2-6)
```

Nomor 5

$$4t^3 + 108$$

```
>$&factor(4*t^3+108)
```

R.5

Hitung!
Nomor 1

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

jawaban manual:

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

$$21x+42=11-x-2$$

$$22x=33$$

$$x=\frac{-}{3}2$$

jawaban emt:

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

Nomor 2

$$x^2+5x$$

```
>$&solve(x^2+5*x, x)
```

Nomor 3

$$y^2 + 6y + 9 = 0$$

```
>$&solve(y^2+6*y+9=0, y)
```

Nomor 4

$$n^2 + 4n + 4 = 0$$

```
>$&solve(n^2+4*n+4=0, n)
```

Nomor 5

$$6x^2 - 7x = 10$$

```
>$&solve(6*x^2-7*x=10, x)
```

Kali atau bagi, dan sederhanakan!
Nomor 1

$$\frac{r-s}{r+s} \cdot \frac{r^2-s^2}{(r-s)^2}$$

```
>$&(((r-s)/(r+s))*((r^2-s^2)/(r-s)^2)); $&factor(%)
```

Nomor 2

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

```
>$&((x^2-4)/(x^2-4*x+4)); $&factor(%)
```

Nomor 3

$$\frac{4-x}{x^2+4x-32}$$

```
>$&((4-x)/(x^2+4*x-32)); $&factor(%)
```

Nomor 4

$$\frac{7}{5x}+\frac{3}{5x}$$

jawaban manual:

$$\frac{7}{5x}+\frac{3}{5x}=\frac{7+3}{5x}=\frac{10}{5x}=\frac{2}{x}$$

jawaban emt:

```
>$&((7/(5*x))+(3/(5*x))); $&factor(%)
```

Nomor 5

$$\frac{5}{4z} - \frac{3}{8z}$$

```
>$&((5/4*z)-(3/8*z)); $&(%)
```

REVIEW EXERCISES

Nomor 1

$$(x^n + 10)(x^n - 4)$$

jawaban manual:

$$(x^n + 10)(x^n - 4) = x^{2n} - 4x^n + 10x^n - 40 = x^{2n} + 6x^n - 40$$

jawaban emt:

```
>$&expand((x^n+10)*(x^n-4))
```

Nomor 2

$$(t^a + t^{-a})^2$$

```
>$&expand((t^a+t^(-a))^2)
```

Nomor 3

$$(a^n - b^n)^3$$

```
>$&expand((a^n-b^n)^3)
```

Nomor 4

$$y^{2n} + 16y^n + 64$$

```
>$& factor(y^(2*n)+16*y^n+64)
```

Nomor 5

$$m^{6n} - m^{3n}$$

```
>$&factor((m^(6*n))-(m^(3*n)))
```

2.3 Exercise Set

Diberikan bahwa

$$f(x) = 3x + 1, g(x) = x^2 - 2x - 6, h(x) = x^3,$$

temukan masing-masing dari yang berikut ini.

Nomor 1

$$(f \circ g)(-1)$$

jawaban manual :

$$(f \circ g)(-1) = 3((-1)^2 - 2(-1) - 6) + 1 = 3(1 + 2 - 6) + 1 = 3(-3) + 1 = -8$$

jawaban emt:

```
>function f(x):= 3*x+1;
>function g(x):= x^2-2*x-6;
>function h(x):= x^3
>f(g(-1))
```

```
-8.00
```

Nomor 2

$$(h \circ f)(1)$$

```
>h(f(1))
```

```
64.00
```

Nomor 3

$$(f \circ h)(-3)$$

```
>f(h(-3))
```

```
-80.00
```

Nomor 4

$$(f \circ f)(-4)$$

```
>f(f(-4))
```

```
-32.00
```

Nomor 5

$$(f \circ g)(1/3)$$

```
>f(g(1/3))
```

```
-18.67
```

3.1 Exercise Set

Sederhanakan. Tulis jawaban dalam bentuk a+bi, di mana a dan b adalah bilangan real.

Nomor 1

$$(-5+3i)+(7+8i)$$

jawaban manual:

$$(-5+3i)+(7+8i) = -5+7+3i+8i = 2+11i$$

jawaban emt:

```
>$ ((-5+3*i)+(7+8*i))
```

Nomor 2

$$(12+3i)+(-8+5i)$$

```
>$((12+3*i)+(-8+5*i))
```

Nomor 3

$$7i(2-5i)$$

```
>$&expand((7*i)*(2-5*i))
```

Nomor 4

$$-2i(-8+3i)$$

```
>$&expand(2*i*(-8+3*i))
```

Nomor 5

$$(10-4i)-(8+2i)$$

```
>$((10-4*i)-(8+2*i))
```

3.4 Exercise Set

Cari solusinya

Nomor 1

$$\frac{1}{4}+\frac{1}{5}=\frac{1}{t}$$

jawaban manual:

$$\frac{1}{4}+\frac{1}{5}=\frac{1}{t}$$
$$\frac{5+4}{20}=1t$$
$$9t=20$$
$$t=\frac{20}{9}$$

jawaban emt:

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

Nomor 2

$$x+\frac{6}{x}=5$$

```
>$&solve(x+(6/x)=5, x)
```

Nomor 3

$$\sqrt{3x-4}=1$$

```
>$&solve(sqrt(3*x-4)=1, x)
```

Nomor 4

$$\sqrt(4)x^2-1=1$$

```
>$&solve(sqrt(4)*x^2-1=1, x)
```

Nomor 5

$$\sqrt{y+4}-\sqrt{y-1}=1$$

```
>$&solve(sqrt(y+4)-sqrt(y-1)=1, y)
```

Hitung!
Nomor 1

$$|2x| \geq 6$$

```
>&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])
```

Nomor 2

$$|x+8| < 9$$

```
>$&fourier_elim(abs(x+8)< 9, [x])
```

Nomor 3

$$|x-5|>0.1$$

```
>$&fourier_elim(abs(x-5)> 0.1, [x])
```

Nomor 4

$$|4x|>20$$

```
>$&fourier_elim(abs(4*x)> 20, [x])
```

Nomor 5

$$|x+6|>10$$

```
>$&fourier_elim(abs(x+6)> 10, [x])
```

Cari solusinya
Nomor 1

$$x + 5\sqrt{x} - 36 = 0$$

```
>$&solve(x+5*sqrt(x)-36=0, x)
```

Nomor 2

$$\frac{3}{3x+4} + \frac{2}{x-1} = 2$$

```
>$&solve((3/(3*x+4))+(2/(x-1))=2, x)
```

Nomor 3

$$\sqrt{x+4} - 2 = 1$$

```
>$&solve(sqrt(x+4)-2=1, x)
```

Nomor 4

$$\sqrt{x+4} - \sqrt{x-4} = 2$$

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

4.1 Exercise Set

Nomor 1
Gunakan substitusi untuk menentukan apakah 4, 5, dan -2 adalah akar dari

$$f(x) = x^3 - 9x^2 + 14x + 24$$

jawaban manual:

$$4^3 - 9(4)^2 + 14(4) - 24 = 0$$

$$5^3 - 9(5)^2 + 14(5) - 24 \neq 0$$

$$(-2)^3 - 9(-2)^2 + 14(-2) - 24 \neq 0$$

Jadi, 4 adalah akar dari fungsi tersebut, sedangkan 5 dan -2 bukan.
jawaban emt:

```
>function f(x):=x^3-9*x^2+14*x+24
>f(4)
```

```
0.00
```

```
>f(5)
```

```
-6.00
```

```
>f(-2)
```

```
-48.00
```

Nomor 2
Gunakan substitusi untuk menentukan apakah 2, 3, dan -1 adalah akar dari

$$f(x) = 2x^3 - 3x^2 + x + 6.$$

```
>function f(x) :=2*x^3-3*x^2+x+6
>f(2)
```

```
12.00
```

```
>f(3)
```

```
36.00
```

```
>f(-1)
```

```
0.00
```

Jadi, -1 adalah akar dari fungsi tersebut, sedangkan 2 dan 3 bukan.

Cari akar-akarnya
Nomor 3

$$f(x) = (x^2 - 5x + 6)^2$$

```
>$& solve((x^2-5*x+6)^2=0, x)
```

Nomor 4

$$f(x) = x^4 - 4x^2 + 3$$

```
>$& solve(x^4-4*x^2+3=0, x)
```

Nomor 5

$$f(x) = x^3 - x^2 - 8x + 4$$

```
>$& solve(x^3-x^2-8*x+4=0, x)
```

Faktorkan fungsi polinomial, kemudian selesaikan persamaannya f(x)=0.
Nomor 1

$$f(x) = x^3 + 4x^2 + x - 6$$

```
>$&factor(x^3+4*x^2+x-6), $&solve(%)
```

Nomor 2

$$f(x) = x^3 + 2x^2 - 13x + 10$$

```
>$&factor(x^3+2*x^2-13*x+10), $&solve(%)
```

Nomor 3

$$x^3 - 3x^2 - 10x + 24$$

```
>$&factor(x^3-3*x^2-10*x+24), $&solve(%)
```

Nomor 4

$$x^4 - x^3 - 19x^2 + 49x - 30$$

```
>$&factor(x^4-x^3-19*x^2+49*x-30), $&solve(%)
```

Nomor 5

$$f(x) = x^4 + 11x^3 + 41x^2 + 61x + 30$$

```
>$&factor(x^4+11*x^3+41*x^2+61*x+30), $&solve(%)
```