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

Sebuah 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 dari ekspresi, bukan penugasan atau perintah pemformatan.

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

```
16.7551608191
```

Perintah harus dipisahkan dengan spasi kosong. Baris perintah berikut mencentak kedua hasilnya.

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

```
50.2654824574
100.530964915
```

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

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

```
0.540302305868
```

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

```
0.857553215846
```

Jika 2 baris dihubungkan dengan "..." kedua baris tersebut akan selalu dieksekusi secara bersamaan.

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

```
1.41666666667
1.41421568627
1.41421356237
```

Ini adalah jalan terbaik untuk menyebarkan perintah panjang ke-2 atau lebih baris. Anda dapat menekan Ctrl+Return untuk memisahkan baris menjadi 2 pada posisi kursor saat ini, atau Ctrl+Back untuk menggabungkan baris.

Untuk melipat semua multi-baris tekan Ctrl+L. Kemudian baris-baris berikutnya hanya akan terlihat jika salah satunya memiliki fokus. Untuk melipat 1 multi-baris, mulai baris pertama dengan "%+".

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

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

```
81
```

Euler medukung perulangan di baris perintah, selama mereka muat dalam 1 baris tunggal atau multi-baris. Dalam program, batasan ini tidak berlaku, tentu saja. Untuk informasi lebih lanjut, lihat pengantar berikut.

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

```
1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237
```

Tidak apa untuk menggunakan multi-baris. Pastikan diakhiri dengan "...".

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

```
1.41421356237
```

Struktur kondisional juga berfungsi

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

```
Thought so!
```

Ketika Anda mengeksekusi perintah, kursor dapat berada di posisi mana saja dalam baris perintah. Anda dapat kembali ke perintah sebelumnya atau melompat ke perintah berikutnya dengan tombol panah. Atau Anda dapat mengklik ke bagian komentar di atas perintah untuk pergi ke perintah.
Ketika Anda menggerakkan kursor di sepanjang baris, pasangan kurung buka dan tutup atau tanda kurung akan disorot. Juga, perhatikan baris status. Setelah kurung buka dari fungsi sqrt(), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Eksekusi perintah dengan tombol return.

```
>sqrt(sin(10°)/cos(20°))
```

```
0.429875017772
```

Untuk melihat bantuan untuk perintah terakhir, buka jendela bantuan dengan F1. Disana anda dapat memasukan teks untuk dicari. Pada baris kosong, bantuan untuk jendela bantuan akan ditampilkan. Anda dapat menekan escape (esc) untuk menghapus baris, atau untuk menutup jendela bantuan.
Anda dapat klik 2 kali di perintah apapun untuk membuka bantuan pada perintah ini. Coba klik 2 kali perintah exp dalam baris perintah.

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

```
2.5
```

Anda juga dapat menyalin dan menempel di Euler. Menggunakan Ctrl+C dan Ctrl+v untuk ini. Untuk menandai teks, seret mouse atau memakai shift bersamaan dengan tombol kursor apapun. Selain itu, anda dapat menyalin kurung yang disorot.

Sintaks Dasar

Euler mengetahui fungsi matematika yang biasa digunakan. Seperti anda lihat sebelumnya, fungsi trigonometri bekerja di radian atau derajat. Untuk menkonversi ke derajat, tambahkan simbol derajat (dengan tombol F7) ke nilainya, atau menggunakan fungsi rad (x). Fungsi akar kuadrat disebut sqrt di Euler. Tentu saja, xˆ(1/2) juga dimungkinkan.

Untuk mengatur variabel, gunakan ”=” atau ”:=”. Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir. Spasi tidak masalah. Tetapi spasi antar perintah diharapkan.

Beberapa perintah dalam 1 baris dipisahkan dengan ”,” atau ”;”. Titik koma menekan keluaran perintah. Di akhir baris perintah, ”,” diasumsikan jika ”;” hilang.

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

```
30.65625
```

EMT menggunakan sintaks program untuk ekspresi. Untuk memasukan

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

anda harus mengatur tanda kurung yang benar dan gunakan / untuk pecahan. Perhatikan tanda kurung yang disorot untuk bantuan.Perhatikan bahwa kontanta Euler e dinamakan E di EMT.

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

```
8.77908249441
```

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 memasukannya dalam bentuk baris.

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

```
23.2671801626
```

Letakan kurung dengan hati-hati disekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu anda untuk menyorot ekspresi yang diakhiri oleh kurung tutup. Anda juga harus memasukan nama "pi" untuk huruf p Yunani.

Hasil dari perhitungan ini adalah bilangan titik pengambang. Secara default, ini dicetak dengan akurasi sekitar 12 digit. Dalam baris perintah berikut, kami juga mempelajari cara merujuk ke hasil sebelumnya dalam baris yang sama.

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

```
0.47619047619
10/21
```

Perintah Euler dapat berubah ekspresi atau perintah primitif. Sebuah ekspresi terdiri dari operator dan fungsi. Jika perlu, harus berisi tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, memasang tanda kurung adalah ide yang baik. Perhatikan bahwa EMT menunjukan tanda kurung buka dan tutup saat mengedit baris perintah.

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

```
14.4978445072
```

Operator numerik Euler meliputi:

```
+ unari atau operator plus
- unari atau operator minus
*, /
. produk matriks
a^b pangkat untuk a positif atau b bilangan bulat (a**b juga
```

berfungsi)

```
n! operator faktorial
```

dan masih banyak lagi.

Berikut adalah beberapa fungsi yang mungkin anda perlukan. Ada banyak lagi.

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

Beberapa perintah memiliki alias, misalnya ln untuk log.

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

```
2
0.5
```

```
>sin(30°)
```

```
0.5
```

Pastikan untuk menggunakan tanda kurung (kurung bulat), setiap kali ada keraguan tentang urutan eksekusi! Berikut ini tidak sama dengan (2^3)^4, yang merupakan default untuk 2^3^4 di EMT (beberapa sistem numerik melakukannya dengan cara lain).

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

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

Bilangan Real

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

```
>longest 1/3
```

```
      0.3333333333333333
```

Representasi dual internal mengambil 8 byte.

```
>printdual(1/3)
```

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

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

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

Kalimat

Sebuah kalimat di Euler didefinisikan dengan "...".

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

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

Kalimat dapat di konkatenasikan dengan | atau dengan +. Ini juga berfungsi dengan angka yang dikonversi menjadi string dalam hal 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.
```

Fungsi print juga mampu mengkonversikan angka menjadi sebuah kalimat. Fungsi ini dapat menerima sejumlah digit dan sejumlah tempat (0 untuk keluaran padat), dan secara optimal sebuah unit.

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

```
Golden Ratio : 1.61803
```

Terdapat string khusus none yang tidak dicetak. Ini dikembalikan oleh beberapa fungsi, ketika hasilnya tidak penting. (Ini dikembalikan secara otomatis, jika fungsi tidak memiliki pernyataan return).

```
>none
```

Untuk mengkonversi sebuah kalimat menjadi angka, cukup evaluasi saja. Ini juga berfungsi untuk ekspresi (lihat dibawah).

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

```
1234.5
```

Untuk menentukan vekttor string, gunakan notasi vektor [...].

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

```
affe
charlie
bravo
```

Vektor string kosong dilambangkan dengan [none]. Vektor string dapat dikonkatenasikan.

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

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

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

String Unicode dapat dikonkatenasikan seperti string lainnya.

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

```
 = 45°
```

I

Dalam komentar, entitas yang sama seperti , etc. dapat digunakan. Ini mungkin merupakan alternatif cepat untuk lateks. (detail lebih lanjut tentang komentar dibawah).

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

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

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

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

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

```
Ü is a German letter
```

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

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

```
We have =.
```

Ini juga mungkin untuk menggunakan entitas numerik.

```
>u"&#196;hnliches"
```

```
Ähnliches
```

Nilai Boolean

Nilai boolean direpresentasikan dengan 1 =benar atau 0=salah dalam Euler.String dapat dibandingkan sama dengan halnya angka.

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

```
0
1
```

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

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

```
1
```

Operasi boolean mematuhi aturan bahasa matriks.

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

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

Anda dapat menggunakan fungsi nonzeros() untuk mengekstrak elemen spesifik dari vektor. Seperti contoh, kami menggunakan kondisi isprime(n).

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

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

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

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

Format keluaran default dari EMT adalah mencetak 12 digit. Untuk memastikan bahwa kita melihat defaultnya, kami mereset formatnya.

```
>defformat; pi
```

```
3.14159265359
```

Secara internal, EMT menggunakan standar IEEE untuk bilangan double dengan sekitar 16 digit desimal. Untuk melihat jumlah digit penuh, gunakan perintah "longestformat", atau kita gunakan operator "longest" untuk menampilkan hasilnya dalam format terpanjang.

```
>longest pi
```

```
      3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari bilangan double.

```
>printhex(pi)
```

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

Format keluaran dapat diubah secara permanen dengan perintah format.

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

```
     0.33333
     3.14159
     0.84147
```

Defaultnya adalah format(12).

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

```
 0.333333333333
```

Fungsi seperti ”shortesformat”, ”shortformat”, ”longformat” bekerja untuk vektor yang diikuti.

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

```
     0.6     0.59   0.032   0.053     0.6    0.56    0.84    0.18
     0.4     0.84   0.026    0.66    0.63    0.77    0.67    0.83
    0.98     0.54    0.18     0.7    0.71    0.93    0.49    0.02
```

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

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

```
3.1416
```

Fungsi ”longestformat” juga mengatur format skalar.

```
>longestformat; pi
```

```
3.141592653589793
```

Sebagai referensi, berikut adalah daftar format keluaran yang paling penting

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

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

Tetapi format keluaran EMT dapat diatur secara fleksibel.

```
>longestformat; pi,
```

```
3.141592653589793
```

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

```
3.14159
```

Defaultnya adalah defformat().

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

Terdapat operator pendek yang hanya mencetak 1 nilai. Operator "longest" akan mencetak semua digit valit dari suatu angka.

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

```
4.934802200544679
```

Ada juga operator pendek untuk mencetak hasil dalam format pecahan. Kami telah menggunakannya diatas.

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

```
25/12
```

Karena format internal menggunakan cara biner untuk menyimpan angka, nilai 0.1 tidak akan diwakili secara tepat. Kesalahan bertambah sedikit, seperti yang Anda lihat dalam perhitungan berikut.

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

```
 -1.110223024625157e-16
```

Tetapi dengan default "longformat", Anda tidak akan melihat ini. Untuk kenyamanan, keluaran 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
```

Kalimat atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang dapat dievaluasi oleh EMT. Untuk ini, gunakan tanda kurung setelah ekspresi. Jika Anda ingin menggunakan string sebagai ekspresi, gunakan konversi untuk menamainya "fx" atau "fxy" dll. Ekspresi memiliki prioritas lebih tinggi daripada fungsi.

Variabel global dapat digunakan dalam evaluasi.

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

```
      12.56637061435917
```

Parameter ditetapkan ke x, y, dan z dalam urutan itu. Parameter tambahan dapat ditambah dengan menggunakan parameter yang telah ditetapkan.

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

```
-0.919535764538
```

Perhatikan bahwa ekspresi akan selalu menggunakan variabel global, bahkan jika ada variabel dalam fungsi dengan nama yang sama. (Jika tidak, evaluasi ekspresi dalam fungsi dapat memiliki 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=value”.

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

```
45
```

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

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

```
45
```

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

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

```
12
```

Sebagai konvensi, ekspresi simbolik atau numerik harus dinamai dengan fx, fxy, dsb. Skema penamaan ini tidak boleh digunakan untuk fungsi.

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

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

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

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

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

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

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

Untuk semua variabel lain dalam ekspresi dapat dispesifikasikan dalam evaluasi menggunakan parameter yang ditetapkan.

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

```
-0.425
```

Sebuah ekspresi tidak perlu bersifat simbolik. Hal ini diperlukan, jika ekspresi mengandung sebuah fungsi, yang hanya diketahui dalam kernel numerik, bukan di Maxima.

Simbol Matematika

EMT melakukan matematika simbolik dengan bantuan Maxima. Untuk detailnya, mulai dengan tuttorial berikut, atau jelajahi referensi untuk Maxima. Para ahli di Maxima harus memperhatikan bahwa ada perbedaan dalam sintaks antara sintaks asli Maxima dan sintaks default ekspresi simbolik di EMT.

Simbol matematika diintegrasikan secara mulus ke dalam Euler dengan &. Setiap ekspresi yang dimulai dengan & adalah ekspresi simbolik. Ini dievaluasi dan dicetak oleh Maxima.

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

```
>$&44!
```

$$2658271574788448768043625811014615890319638528000000000$$

Pada cara ini, Anda bisa memasukan hasil yang besar secara tepat. Mari kita hitung

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

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

$$2481256778$$

Tentu saja Maxima memliki lebih banyak fungsi efisien untuk ini (seperti halnya bagian numerik dari EMT).

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

$$2481256778$$

Untuk mempelajari lebih lanjut mengenai fungsi tertentu, klik 2 kali diatasnya. Sebagai contoh, mencoba klik 2 kali pada "&binomial" dalam baris perintah sebelumnya. Hal ini membuka dokumentasi Maxima sebagaimana yang disediakan oleh penulis program tersebut.

Anda akan mempelajari bahwa hal berikut juga berfungsi.

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

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

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

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

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

$$120$$

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

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

Seperti yang akan Anda lihat dalam contoh sebelumnya dan berikut, jika Anda memiliki LaTeX yang diinstal, Anda dapat mencetak ekspresi simbolik dengan Latex. Jika tidak, perintah berikut akan mengeluarkan pesan kesalahan.

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

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

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

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

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


Untuk kelengkapan, kami menyatakan bahwa ekspresi simbolik dapat digunakan dalam program, tetapi perlu dilampirkan dalam tanda kutip. Selain itu, jauh lebih efektif untuk memanggil Maxima pada waktu 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, % mengacu pada hasil sebelumnya.

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

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

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

Ekspresi simbolik dapat digunakan pada eskpresi simbolik lainnya.

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

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

Masukan 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
```

Jika Anda seorang ahli di Maxima, Anda mungkin ingin menggunakan sintaks asli Maxima. Anda bisa menggunakan dengan ":::".

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

```
 2
g
```

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

```
 3  x
x  E
```

$$x^3 e^x$$

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

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

```
     5
125 E
```

$$125\,e^5$$

```
18551.64488782208
```

```
>fx(5)
```

```
18551.6448878
```

Untuk evaluasi ekspresi dengan nilai spesifik dari variabel, Anda dapat menggunakan operator "with".

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

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

```
            10         5
      1000 E   - 125 E


      2.20079141499189e+7
```

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

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

Untuk mendapatkan kode Lateks pada sebuah ekspresi, Anda bisa menggunakan perintah tex.

```
>tex(fx)
```

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

Ekspresi simbolik dapat dievaluasi seperti ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

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

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

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

Penugasan juga bersifat simbolik.

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

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

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

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

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

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

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

```
1.56155281281
```

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

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

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

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

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

Untuk mendapatkan solusi simbolik yang spesifik, Anda dapat menggunakan "with" dan indeks.

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

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

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

Untuk menyelesaikan sistem persaman, gunakan persamaan vektor. Hasilnya akan berupa solusi vektor.

```
>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 di Maxima. Beberapa flag dapat digunakan sebagai perintah juga, yang lain tidak. Flag ditambahkan dengan "|" (bentuk yang lebih baik dari "ev(...,flags)")

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

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

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

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

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

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

Fungsi

Dalam EMT, fungsi merupakan program yang didefinisikan dengan perintah "function". Ini bisa berupa fungsi satu baris atau multibaris.
Fungsi satu baris dapat bersifat numerik atau simbolik. Fungsi satu baris numerik didefinisikan dengan ":=".

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

Untuk gambaran umum, kami menunjukan semua kemungkinan definisi untuk fungsi satu baris. Sebuah fungsi dapat dievaluasi seperti halnya fungsi bawaan Euler lainnya.

```
>f(2)
```

```
4.472135955
```

Fungsi ini akan berfungsi pada vektor, mematuhi bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi tersebut 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 pada fungsinya.

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

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

```
0.786151377757
```

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

Anda dapat memanggil fungsi dibawah ini sebagai "_...", jika itu adalah sebuah fungsi dalam inti Euler.

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

```
0.707106781187
```

Sebaiknya kita menghapus definisi ulang dari sin.

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

```
0.707106781187
```

Fungsi numerik dapat memiliki parameter default.

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

Melewatkan parameter ini menggunakan nilai default.

```
>f(4)
```

```
16
```

Mengaturnya menimpa nilai default.

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

```
80
```

Parameter yang ditetapkan juga menimpa parameter default. Hal ini digunakan oleh banyak fungsi Euler seperti plot2d, plot3d.

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

```
16
```

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

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

```
24
```

Tetapi parameter yang ditetapkan menimpa nilai global.

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

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

```
20
```

Fungsi simbolik didefinisikan dengan "&=". Mereka didefinisikan dalam Euler dan Maxima, dan bekerja di kedua dunia. Ekspresi definisi 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 lainnya.

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

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

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

Mereka juga dapat digunakan dalam ekspresi numerik. Tentu saja, ini hanya akan berfungsi jika EMT dapat menginterpretasikan semuanya dalam suatu fungsi.

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

```
178.635099908
```

Mereka dapat digunakan untuk mendefinisikan simbol fungsi atau ekspresi lainnya.

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

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

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

```
0.703467422498
```

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

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

```
0.703467422498
```

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

$$\left(2\,x-1\right)^{n}$$

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

$$(x+2)^n$$

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

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

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

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

625

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$x^3 - x$$

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

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

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

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

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

yang tidak dapat dievaluasi secara simbolik.

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

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

Sekarang fungsi tersebut dapat dipanggil dengan atau tanpa parameter ”base”.

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

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

```
2
```

Seringkali, kita ingin menggunakan fungsi dari vektor di satu tempat, dan untuk elemen individu di tempat lain. Hal ini mungkin dengan parameter vektor.

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

$$b^2 - a\,b + b + a^2$$

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

Fungsi simbolik seperti itu dapat digunakan untuk variabel simbolik.

Tetapi fungsi tersebut juga dapat digunakan untuk vektor numerik.

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

```
17
```

Ada juga fungsi-fungsi simbolik murni yang tidak bisa digunakan secara numerik.

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

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

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

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

$$0$$

Tetapi, tentu saja, mereka dapat digunakan dalam ekspresi simbolik atau pada definisi dari fungsi simbolik.

```
>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 meringkas

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

Eskpresi dapat diselesaikan secara numerik dan simbolik.

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

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

```
1.41421356237
```

Ini juga bekerja pada ekspresi simbolik. Ambil fungsi berikut.

```
>$&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])
```

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

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

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

Sekarang kita mencari titik dimana nilai polinomialnya 2. Pada solve (), nilai target default y=0 dapat diubah dengan variabel yang ditetapkan.
Kami menggunakan y=2 dan memeriksa dengan evaluasi polinomial pada hasil sebelumnya.

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

```
0.966715594851
2
```

Menyelesaikan ekspresi simbolik dalam bentuk simbolik mengembalikan daftar solusi. Kami menggunakan solver simbolik solve() yang disediakan oleh Maxima.

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

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

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

```
>longest sol()
```

```
-0.6180339887498949       1.618033988749895
```

Untuk menggunakan solusi simbolik di ekspresi lain, cara termudahnya adalah dengan "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 simbolik dapat dilakukan dengan persamaan vektor dan solver simbolik solve(). Jawabannya adalah 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;
```

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

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

```
2.54116291558
```

Hal ini juga berfungsi dengan ekspresi. Tetapi kemudian, elemen daftar bernama harus digunakan. (Lebih lanjut tentang daftar dalam tutorial tentang sintaks EMT).

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

```
2.54116291558
```

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])
```

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

```
>$&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])
```

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

```
>&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])
```

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

Bahasa Matriks

Dokumentasi inti EMT ini berisi diskusi terperinci tentang bahasa matriks Euler.

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

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

```
            1             2
            3             4
```

Produk matriks dinotasikan dengan titik.

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

```
            3
            4
```

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

```
[3,  4]
```

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

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

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

```
             11
             25
```

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

```
              1             0
              0             1
```

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

```
>A.A
```

```
              7            10
             15            22
```

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

```
         1           4
         9          16
```

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

```
        37          54
        81         118
```

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

```
        37          54
        81         118
```

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

```
         1           1
         1           1
```

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

```
0.333333      0.666667
    0.75             1
```

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

```
      -2
     2.5
```

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

```
      -2
     2.5
```

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

```
       1             0
       0             1
```

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

```
              1               0
              0               1
```

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

```
              1               4
              9              16
```

Ini bukanlah produk matriks, tetapi perkalian elemen dengan elemen. Hal yang sama berlaku untuk vektor.

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

```
              9
             16
```

JIka salah satu operand adalah suatu vektor atau skalar, maka akan diperluas secara alami.

```
>2*A
```

```
            2             4
            6             8
```

Sebagai contoh, jika operandnya adalah vektor kolom, elemen-elemennya diterapkan ke semua baris dari A.

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

```
            1             4
            3             8
```

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

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

```
            2             6
            6            12
```

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

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

```
            1             2
            1             2
```

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

```
            1             4
            3             8
```

Hal ini juga berlaku untuk 2 vektor, dimana salah satunya adalah vektor baris dan yang lainnya adalah vektor kolom. Kami menghitung i*j untuk i,j dari 1 hingga 5. Triknya adalah 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, ingat bahwa ini bukanlah produk matriks!

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

```
55
```

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

```
55
```

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

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

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

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

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

```
5
```

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

Kita mendapatkan vektor 0 dan 1, dimana 1 berarti benar.

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

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

Dari vektor semacam itu, "nonzeros" memilih elemen non-nol.

Dalam hal 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 bisa menggunakan vektor indeks ini untuk mendapatkan nilai-nilai yang sesuai dalam t.

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

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

Sebagai contoh, mari kita mencari semua kuadrat dari angka 1 hingga 1000, yang merupakan 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. Ini menggunakan floating point presisi ganda secara internal. Namun, seringkali sangat berguna.

Kita bisa memeriksa keprimaan. 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 pada vektor. Untuk matriks, terdapat mnonzeros().

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

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

Hal ini mengembalikan indeks-indeks dari elemen-elemen yang bukan nol dalam matriks.

```
>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 menjadi 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 pada indeks menjadi entri dari matriks lain.

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

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

Dan ini sangat mungkin untuk mendapatkan elemen dalam vektor.

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

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

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

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

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

Kita dapat menggunakannya untuk mengekstrak nilai maksimal di setiap baris.

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

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

Ini, tentu saja, ini sama seperti fungsi max().

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

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

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

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

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

Untuk membentuk matriks, kita dapat menumpuk 1 matriks diatas yang lain. JIka keduanya tidak memiliki jumlah kolom yang sama, yang lebih pendek akan terisi dengan 0.

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

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

Demikian pula, kita dapat menempelkan matriks ke sisi yang lain secara berdampingan, jika keduanya memiliki jumlah baris yang sama.

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

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

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

Ada pengecualian untuk aturan ini. Bilangan real yang dilampirkan pada matriks akan digunakan sebagai kolom yang diisi dengan angka 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
```

Ini memungkinkan untuk membuat matriks dari vektor baris dan kolom.

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

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

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

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

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

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

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

Untuk mendapatkan ukuran A, kita dapat mengikuti fungsi berikut.

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

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

Untuk vektor, terdapat length().

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

```
9
```

Terdapat banyak lagi fungsi lain, untuk menghasilkan matriks.

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

```
              1              1
              1              1
```

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

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

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

Juga matriks angka acak dapat dihasilkan dengan random (distribusi seragam) atau normal (distribusi Gauß).

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

```
        0.66566      0.831835
          0.977      0.544258
```

Berikut adalah fungsi lain yang berguna, yang menata ulang elemen matriks menjadi matriks lain.

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

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

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

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

Mari kita coba.

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

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

Fungsi multdup() menduplikasi elemen dari sebuah vektor.

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

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

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

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

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

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

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

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

Fungsi khusus drip (v,i) menghapus elemen-elemen dengan 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 elemen dalam v, bukan nilai dari elemen tersebut. Jika Anda ingin menghapus elemen, pertama-tama Anda butuh untuk mencari elemen. Fungsi indexof(v,x) dapat digunakan untuk mencari elemen x dalam vektor v yang terurut.

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

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

Seperti yang anda lihat, tidak ada salahnya memasukan indeks diluar 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 atur diagonal terendah (-1) ke 1:4.

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

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

Perhatikan bahwa kami tidak mengubah nilai matriks A. Kami mendapatkan matriks baru sebagai hasil setdiag().

Berikut adalah 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 matriks juga dapat diekstrak dari matriks. Untuk mendemonstrasikan hal ini, kita menata ulang vektor 1:9 menjadi matriks 3x3.

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

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

Sekarang kita bisa mengekstrak diagonal.

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

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

Sebagai contoh kita dapat membagi maktriks dengan diagonalnya. Bahasa matriks menangani agar vektor kolom d ditetapkan pada matriks baris demi baris.

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

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

Hampir semua fungsi di Euler bekerja untuk matriks dan vektor, kapanpun al ini tidak 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 bisa dengan mudah membuat tabel nilai. Ini adalah salah satu cara untuk mem-plot fungsi (alternatifnya menggunakan Ekspresi).

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

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

Dalam contoh berikut, kami menghasilkan vektor nilai t[i] dengan jarak 0,1 dari -1 hingga 1. Kemudian kami menghasilkan vektor nilai 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.

Sebagai contoh, vektor kolom dikalikan dengan vektor baris akan menghasilkan matrks, jika operator ditetapkan. Dalam hal berikut, v' adalah vektor yang ditranspos (vektor kolom).

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

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

Perhatikan bahwa ini sangat berbeda dari produk matriks. Produk matriks dilambangkan dengan titik "." dalam EMT.

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

```
55
```

Secara default, vektor baris dicetak dalam format ringkas.

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

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

Untuk matriks, operator khusus "." menandakan perkalian matriks, dan "A" menandakan transposisi. Matriks 1x1 dapat digunakan seperti angka real.

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

```
5
25
```

Untuk mentranspose matriks, kita dapat menggunakan tanda apostrof.

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

```
        1
        2
        3
        4
```

Sehingga kita dapat menghitung matriks A dikali vektor b.

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

```
       30
       70
```

Perhatikan bahwa v tetap vektor baris. Jadi v'.v berbeda dengan v.v'.

```
>v'.v
```

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

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

```
>v.v'
```

```
30
```

Terdapat juga fungsi norm (bersama dengan banyaknya fungsi aljabar linear lainnya).

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

```
30
```

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut adalah ringaksan aturannya.

- Suatu fungsi yang diterapkan pada vektor atau matriks diterapkan pada setiap elemennya.

- Sebuah operator yang beroperasi pada 2 matriks berukuran sama diterapkan secara berpasangan pada elemen-elemen matriks tersebut.

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

Sebagai contoh, nilai skalar kali vektor mengalikan nilai dengan setiap elemen vektor. Atau matriks kali vektor (dengan *, bukan .) memperluas vektor ke ukuran matriks dengan menduplikatnya.

Berikut adalah kasus sederhana dengan operator ^.

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

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

Berikut adalah kasus yang lebih rumit. Sebuah vektor baris kali sebuah vektor kolom memperluas keduanya dengan duplikasi.

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

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

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan *!

```
>v.v'
```

```
14
```

Ada banyak fungsi untuk matriks. Kami memberikan daftar singkat. Anda harus berkonsultasi dengan dokumentasi untuk informasi lebih lanjut tentang perintah-perintah ini.

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

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

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

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

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

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

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

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

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

Indeks juga dapat berupa vektor baris indeks. : menunjukan 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 tujuan vektorisasi, elemen-elemen sebuah matriks dapat diakses seolah-olah mereka adalah vektor.

```
>A{4}
```

```
4
```

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

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

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

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

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

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

Sekarang kita menambahkan kolom ke matriks.

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

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

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

Dalam contoh berikut, kami menghitung t[j]ˆi untuk i dari 1 hingga n. Kami mendapatkan matriks, di mana setiap baris adalah tabel tˆi untuk satu i. Artinya, matriks memiliki elemen

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

Fungsi yang tidak berfungsi untuk input vektor harus "divektorisasi". Hal ini dapat dicapai dengan kata kunci "peta" dalam definisi fungsi. Kemudian fungsi akan dievaluasi untuk setiap elemen parameter vektor.

Integrasi numerik integrate() hanya berfungsi untuk batas interval skalar. Jadi kita perlu memvektorisasinya

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

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

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

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

Untuk mengakses elemen matriks, gunakan notasi kurung siku.

```
>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 matriks.

```
>A[2]
```

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

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

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

```
2
```

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

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

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

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

Disini kita menginginkan baris pertama dan kedua dari A.

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

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

Kita bahkan dapat mengubah urutan A menggunakan vektor indeks. Secara tepat, kita tidak mengubah A disini, tetapi menghitung versi A yang diurutkan ulang.

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

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

Trik indeks juga berfungsi dengan kolom.

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

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

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

Sebagai singkatan, ":" melambangkan 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 bisa mendapatkan baris terakhir pada A.

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

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

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

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

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

Kita juga dapat menetapkan nilai ke baris A.

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

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

Kita bahkan bisa menetapkan nilai ke 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 diizinkan.

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

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

Perhatian: Indeks yang berada di luar batas akan mengembalikan matriks kosong, atau pesan kesalahan, tergantung pada pengaturan sistem. Secara default, akan muncul pesan kesalahan. Namun, ingatlah bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks dengan menghitung dari akhir.

```
>A[4]
```

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

Fungsi sort() mengurutkan vektor baris

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

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

Seringkali perlu untuk mengetahui indeks dari vektor yang diurutkan dalam vektor aslinya. Ini bisa digunakan untuk mengurutkan vektor lain dengan cara yang sama.

Mari kita acak vektor.

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

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

Indeks ditampung berisi urutan yang tepat dari v.

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

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

Ini juga berlaku 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
```

Dapat anda lihat, posisi entri ganda cukup acak.

```
>ind
```

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

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

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

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

Ini juga bekerja pada vektor string.

```
>unique(s)
```

```
a
aa
d
e
```

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

Untuk sistem linear Axb, anda dapat menggunakan algoritma Gauss, matriks invers atau kecocokan linear. Operasi A\b menggunakan versi dari algoritma Gauss.

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

```
             -4
            4.5
```

Untuk contoh lain, kita menghasilkan matriks 200x200 dan jumlah baris-barisnya. Kemudian kita menyelesaikan Ax=b menggunakan matriks invers. Kita mengukur kesalahan sebagai deviasi maksimum dari semua elemen dari 1, yang tentu saja adalah solusi yang benar.

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

```
    1.177724584522366e-12
```

Jika sistem tidak memiliki solusi, kecocokan linier 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
```

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

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

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

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

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

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

```
>$&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 simbolik, matriks ini dapat digunakan dalam ekspresi simbolik 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 eigen juga dapat dihitung secara otomatis. Hasilnya adalah vektor dengan dua vektor nilai eigen dan kelipatannya.

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

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

Untuk mengekstrak vektor eigen tertentu, diperlukan pemilihan indeks yang hati-hati.

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

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

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

Matriks simbolik dapat dievaluasi secara numerik di Euler, sama seperti ekspresi simbolik lainnya.

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

```
              1             4
              5             2
```

Pada ekspresi simbolik, gunakan with.

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

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

Akses ke baris matriks simbolik bekerja sama seperti pada matriks numerik.

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

$$[1, a]$$

Sebuah ekspresi simbolik dapat berisi penugasan. Dan hal itu mengubah matriks A.

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

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

Terdapat fungsi simbolik di Maxima yang bertujuan untuk membentuk 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 di Euler. Untuk infomasi lebih lanjut tentang Maxima, lihat pada pengantar tentang Maxima.

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

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

Euler juga memiliki fungsi kuat xinv(), yang dapat melakukan usaha lebih besar dan mendapatkan hasil yang lebih akurat.

Perhatikan bahwa dengan &:= , matriks B didefinisikan secara simbolik pada ekspresi simbolik dan numerik pada ekspresi numerik. Jadi kita dapat menggunakannya disini.

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

```
                      1                       0
                      0                       1
```

Sebagai contoh pada nilai eigen A dapat dihitung secara numerik.

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

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

Atau simbolik. Lihat tutorial tentang Maxima untuk detail hal ini.

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

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

Nilai Numerik dalam Ekspresi Simbolik

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

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

```
            1         3.14159
            4               5
```

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

```
>$&A
```

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

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

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

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

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

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

$$1.4142135623730950488016887242097_B \times 10^0$$

$$1.414213562373095$$

Presisi angka floating poin besar dapat diubah.

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

```
        3.14159265358979323846264338327950288419716939937510582097494\
4592307816406286208998628034825342117068b0
```

Variabel numerik dapat digunakan pada ekspresi simbolik manapun menggunakan "@var".

Perhatikan bahwa ini hanya diperlukan jika variabel telah didefinisikan dengan := atau = sebagai variabel numerik.

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

$$-5.424777960769379$$

Dibawah ini, kita menggunakan Euler Math Toolbox (EMT) untuk menghitung tingkat bunga. Kami melakukannya secara numerik dan simbolik untuk menunjukan pada Anda bagaimana Euler dapat bekerja dalam menangani masalah kehidupan nyata.

Misalkan Anda memiliki modal awal sebesar 5000(katakan dalam dolar).

```
>K=5000
```

```
5000
```

Sekarang kita asumsikan tingkat bunga sebesar 3% pertahun. Mari kita tambahkan satu tingkat bunga sederhana dan hitung hasilnya.

```
>K*1.03
```

```
5150
```

Euler juga akan memahami sintaks berikut.

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

```
5150
```

Namun, lebih mudah menggunakan faktor.

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

```
1.03
5150
```

Untuk 10 tahun, kita bisa cukup mengalikan faktor-faktor tersebut dan mendapatkan nilai akhir dengan tingkat bunga majemuk.

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

```
6719.58189672
```

Untuk keperluan kita, kita bisa mengatur format ke 2 digit setelah titik desimal.

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

```
     6719.58
```

Mari kita cetak hasil tersebut untuk 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 antara tahun pertama hingga tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu untuk menuliskan loop, tetapi cukup memasukan

```
>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 di Euler dapat diterapkan ke elemen-elemen vektor satu per satu. 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 sebuah vektor dari faktor q^0 ke q^10. Ini dikalikan dengan K, dan kita mendapatkan vektor nilai-nilai.

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

Tentu saja, cara realistik untuk menghitung tingkat bunga ini adalah dengan membulatkan ke sen terdekat setelah tiap tahun. Mari kita menambahkan fungsi untuk ini.

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

Mari kita membandingkan hasil keduanya, dengan dan tanpa pembulatan.

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

```
      1271.61
    1271.6071
```

Sekarang tidak ada formula sederhana untuk tahun n, dan kita harus melakukan loop selama tahun-tahun tersebut. Euler menyediakan banyak solusi untuk ini.

Cara termudah yaitu dengan fungsi iterate, yang mengulangi fungsi tertentu 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 kita dengan tempat desimal tetap.

```
>VKr'
```

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

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

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

```
5150.00
5000.00     5150.00     5304.50
```

Secara mengejutkan, kita juga dapat menggunakan vektor indeks. Ingat bahwa 1:3 membentuk vektor [1,2,3].

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

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

```
6719.60
6719.58
```

Perbedaannya sangat kecil.

Sekarang kta menggunakan fungsi yang lebih maju, yang menambahkan sejumlah uang tertentu setiap tahun.

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

Kita tidak perlu menentukan q atau R untuk mendefinisikan suatu fungsi. Hanya saat menjalankan perintah, kita harus mendefinisikan nilai-nnilai ini. Kami pilih R=200.

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

```
Real 1 x 11 matrix

    5000.00     5350.00     5710.50     6081.82     ...
```

Bagaimana jika kita mengurangi 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 lihat bahwa uang berkurang. Jelas, karena kita hanya mendapatkan 150 dari bunga pada tahun pertama, tetapi mengurangi 200, kita kehilangan uang setiap tahun.

Bagaimana kita bisa menentukan berapa lama uang akan bertahan? Kita harus menulis loop untuk ini. Cara termudah adalah dengan melakukan iterasi cukup lama.

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

```
 Real 1 x 51 matrix

     5000.00     4950.00     4898.50     4845.45     ...
```

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

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

```
        48.00
```

Alasannya adalah bahwa nonzeros(VKR<0) mengembalikan vektor indeks i, dimana VKR[i]<0, dan min menghitung indeks minimal.

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

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

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

```
-19.83
 47.00
```

Mari kita coba untuk menjawab pertanyaan yang lebih ambigu. Misalkan kita tahu bahwa nilai uang adalah 0 setelah 50 tahun. Berapa tingkat bunga yang diperlukan?

Pada pertanyaan ini, kita hanya dapat menjawabnya dengan numerik. Dibawah, kita akan menurunkan rumus yang diperlukan. Kemudan Anda akan melihat bahwa tidak ada rumus mudah untuk tingkat bunga. Tetapi untuk sekarang, kita bertujuan untuk solusi numerik.

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

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

Iterasi adalah seperti dibawah ini

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

Tetapi kita tidak lagi menggunakan nilai flobal R pada ekspresi ini. Fungsi seperti iterate() memiliki trik khusus di Euler. Anda dapat melewatkan nilai variabel pada ekspresi sebagai parameter titik koma. Pada kasus ini P dan R.

Selain itu, kita hanya tertarik pada nilai akhir. Sehingga kita mengambil indeks [-1].

Mari kita lakukan uji coba.

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

```
-19.83
```

Sekarang kita dapat menyelesaikan permasalahan kita

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

```
3.15
```

Rutin solve menyelesaikan expression=0 untuk variabel x. Jawabannya adalah 3.15% per tahun. Kita mengambil nilai awal 3% untuk algoritma. Fungsi solve() selalu dibutuhkan untuk nilai awal.

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

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

```
-336.08
```

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

Solusi Simbolik untuk Masalah Tingkat Bunga

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

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

$$R + q\,K$$

Kita bisa melakukan iterasi pada fungsi ini.

```
>$&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 agak rumit. Jumlahnya dievaluasi dengan flag "simpsum" untuk menyederhanakannya menjadi bentuk pecahan.

Mari kita buat fungsi untuk ini.

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

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

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

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

```
-19.82504734650985
-19.82504734652684
```

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

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

```
20.51
```

Jawaban ini mengatahakan bahwa ini akan menjadi negatif setelah 21 tahun.

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

Misalkan kita mendapatkan pinjaman sebesar K, dan membayar n pembayaran sebesar R (memulai setelah 1 tahun) meninggalkan utang sisa Kn (pada saat pembayaran terakhir). Rumus untuk ini jelas adalah

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

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

Biasanya rumus ini diberikan dalam bentuk

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

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

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

Kita dapat mneyelesaikan untuk besaran simbolik R.

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

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

Seperti yang anda lihat pada rumus, fungsi ini menghasilkan error floating point untuk i=0. Meskipun demikian, Euler masih memplotnya.

Tentu saja, kita memiliki batas 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 10 cicilan sebesar 500.

Persamaan ini juga dapat diselesaikan untuk n. Hasilnya terlihat lebih baik jika kita menerapkan beberapa penyederhanaan pada persamaan tersebut.

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

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

LATIHAN SOAL

R.2

1) Hitunglah nilai dari

$$2^6 \cdot 2^{-3} \div 2^{10} \div 2^{-8}$$

```
>$&expand(2^6*2^-3/2^10/2^-8)
```

Hasil yang didapat dari perhitungan tersebut yaitu 2, sama seperti apabila dihitung secara manual

2) Hitunglah nilai dari

$$\frac{4(8-6)^2-4\cdot 3+2\cdot 8}{3^1+19^0}$$

```
>$&expand(4*((8-6)^2)-(4*3)+(2*8))/(3^1+19^0)
```

$$5$$

Hasil yang didapat dari perhitungan tersebut yaitu 5, sama seperti apabila dihitung secara manual

3) Sederhanakanlah

$$\left[\frac{(3x^a y^b)^3}{(-3x^a y^b)^2}\right]^2$$

```
>$&expand((3*x^a*y^b)^3/(-3*x^a*y^b)^2)^2
```

$$9\,x^{2\,a}\,y^{2\,b}$$

Hasil yang didapat dari pernyederhanaan tersebut yaitu 9xˆ(2a)yˆ(2b), sama seperti apabila dihitung secara manual

4) Hitunglah nilai dari

$$\frac{[4(8-6)^2+4](3-2\cdot 8)}{2^2(2^3+5)}$$

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

$$-845$$

Hasil yang didapat dari perhitungan tersebut yaitu -845, sama seperti apabila dihitung secara manual.

5)Sederhanakanlah

$$\left[(\frac{x^r}{y^t})^2(\frac{x^{2r}}{y^{4t}})^{-2}\right]^{-3}$$

```
>$&expand ([(x^r/y^t)^2*(x^(2*r)/y^(4*t))^-2]^(-3))
```

$$\left[\frac{x^{6\,r}}{y^{18\,t}}\right]$$

Hasil yang didapat dari pernyederhanaan tersebut yaitu xˆ(6r)/yˆ(18t), sama seperti apabila dihitung secara manual

R.3

---

1)Lakukan operasi berikut

$$(2x + 3y + z - 7) + (4x - 2y - z + 8) + (-3x + y - 2z - 4)$$

```
>$&expand (2*x+3*y+z-7)+(4*x-2*y-z+8)+(-3*x+y-2*z-4)
```

$$-2\,z + 2\,y + 3\,x - 3$$

Dari operasi tersebut akan didapat hasil -2z+2y+3x-3

2) Sederhanakan operasi berikut

$$(3a^2)(-7a^4)$$

```
>$&expand((3*a^2)*(-7^4))
```

$$-7203\,a^2$$

Dari operasi tersebut akan didapat hasil -7203aˆ2

3) Hitung penyederhanaan dari

$$(2x+3y+4)(2x+3y-4)$$

```
>$&expand((2*x+3*y+4)*(2*x+3*y-4))
```

$$9\,y^2+12\,x\,y+4\,x^2-16$$

Dari operasi tersebut akan didapat hasil 9yˆ2+12xy+4xˆ2-16

4) Hitung penyelesaian dari

$$(m+1)(m-1)$$

```
>$&expand((m+1)*(m-1))
```

$$m^2-1$$

Dari operasi tersebut akan didapat hasil mˆ2-1

5) Hitung penyelesaian dari

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

```
>$&expand((x+1)*(x-1)*(x^2+1))
```

$$x^4-1$$

Dari operasi tersebut akan didapat hasil xˆ4-1

1) Hitunglah faktor dari

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

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

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

Dari operasi tersebut akan didapat faktor berupa t=-3 atau t=-5

2) Tentukan faktor dari

$$z^2 - 81$$

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

$$[z = -9, z = 9]$$

Dari operasi tersebut akan didapat faktor berupa z=-9 atau z=9

3) Tentukan faktor kuadrat dari

$$25ab^4 - 25az^4$$

```
>$&solve(25*a*b^4-25*a*z^4)
```

```
Maxima said:
solve: more unknowns than equations.
Unknowns given :
[z,b,a]
Equations given:
[25*a*b^4-25*a*z^4]
 -- an error. To debug this try: debugmode(true);

Error in:
$&solve(25*a*b^4-25*a*z^4) ...
                          ^
```

Dari operasi tersebut tidak akan didapat faktor karena perbedaan dalam sukunya.

4) Tentukan faktor dari

$$4t^3 + 108$$

```
>$&solve(4*t^3+108)
```

$$\left[t=\frac{3-3^{\frac{3}{2}}\,i}{2}, t=\frac{3^{\frac{3}{2}}\,i+3}{2}, t=-3\right]$$

Dari operasi tersebut akan didapat 3 faktor karena merupakan perpangkatan 3

5) Tentukan faktor dari

$$x^2-5x+\frac{25}{4}$$

```
>$&solve(x^2+5*x+25/4)
```

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

Dari operasi tersebut hanya akan mendapat 1 faktor.

R.5

1) Selesaikan persamaan berikut

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

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

$$\left[x = -\frac{3}{2}\right]$$

Dari operasi tersebut akan didapat penyederhanaan berupa x=-3/2

2) Selesaikan persamaan berikut

$$x^2 + 100 = 20x$$

```
>$&solve(x^2+100=20*x)
```

$$[x = 10]$$

Dari operasi tersebut akan didapat penyederhanaan berupa x=10

3) Selesaikan

$$x^2 - 36 = 0$$

```
>$&solve(x^2-36=0)
```

$$[x = -6, x = 6]$$

Dari operasi tersebut akan didapat penyederhanaan berupa x=-6 dan x=6 karena 36 merupakan kuadrat dari -6 dan 6

4) Selesaikan

$$st = t - 4$$

```
>$&solve(st=t-4,t)
```

$$[t = st + 4]$$

Maka akan didapat t= karena yang dicari adalah nilai t

5) Selesaikan

$$3[5-3(4-t)]-2=5[3(5t-4)+8]-26$$

```
>$&solve(3*(5-3*(4-t))-2=5*(3*(5*t-4)+8)-26)
```

$$\left[t=\frac{23}{66}\right]$$

Dari operasi tersebut akan didapat penyederhanaan berupa t=23/66

R.6

1) Sederhanakan

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

```
>$&solve((x^2-4)/(x^2-4*x+4))
```

$$[x=-2]$$

Dari operasi tersebut akan didapat penyederhanaan berupa x=-2

2) Sederhanakan

$$\frac{6-x}{x^2-36}$$

```
>$&solve((6-x)/x^2-36)
```

$$\left[x=\frac{-\sqrt{865}-1}{72}, x=\frac{\sqrt{865}-1}{72}\right]$$

Dari operasi tersebut akan didapat penyederhanaan berupa akar

3) Kalikan atau bagi, dan jika mungkin, sederhanakan

$$\frac{r-s}{r+s} \cdot \frac{r^2-s^2}{(r-s)^2}$$

```
>$&expand(((r-s)/(r+s))*((r^2-s^2)/(r-s)^2))
```

$$\frac{r^2}{r^2-s^2} - \frac{s^2}{r^2-s^2}$$

```
>$&solve((r^2/(r^2-s^2))-(s^2/(r^2-s^2)))
```

```
Maxima said:
solve: more unknowns than equations.
Unknowns given :
[s,r]
Equations given:
[r^2/(r^2-s^2)-s^2/(r^2-s^2)]
 -- an error. To debug this try: debugmode(true);

Error in:
$&solve((r^2/(r^2-s^2))-(s^2/(r^2-s^2))) ...
                                        ^
```

Dari operasi tersebut tidak didapat penyederhanaannya

4) Tambah atau kurang, dan jika mungkin sederhanakan

$$\frac{7}{5x}+\frac{3}{5x}$$

```
>$&expand((7/5*x)+(3/5*x))
```

$$2\,x$$

Dari operasi tersebut akan didapat penyederhanaan berupa 2x

5) Sederhanakan

$$\frac{y^5-5y^4+4y^3}{y^3-6y^2+8y}$$

```
>$&expand((y^5-5*y^4+4*y^3)/(y^3-6*y^2+8*y))
```

$$\frac{y^5}{y^3-6\,y^2+8\,y}-\frac{5\,y^4}{y^3-6\,y^2+8\,y}+\frac{4\,y^3}{y^3-6\,y^2+8\,y}$$

```
>$&solve((y^5-5*y^4+4*y^3)/(y^3-6*y^2+8*y))
```

$$[y=0, y=1]$$

1) Kalikan dan sederhanakan:

$$\frac{x^2+x-6}{x^2+8x+15} \cdot \frac{x^2-25}{x^2-4x+4}$$

```
>$&expand(((x^2+x-6)/(x^2+8*x+15))*((x^2-25)/(x^2-4*x+4)))
```

$$\frac{x^4}{x^4+4\,x^3-13\,x^2-28\,x+60}+\frac{x^3}{x^4+4\,x^3-13\,x^2-28\,x+60}-\frac{31\,x^2}{x^4+4\,x^3-13\,x^2-28\,x+60}-\frac{25\,x}{x^4+4\,x^3-13\,x^2-28\,x+6}$$

```
>$&solve(((x^2+x-6)/(x^2+8*x+15))*((x^2-25)/(x^2-4*x+4)))
```

$$[x=5]$$

2) Kalikan

$$(x^n+10)(x^n-4)$$

```
>$&expand((x^n+10)*(x^n-4))
```

$$x^{2\,n}+6\,x^n-40$$

3) Kalikan

$$(a^n - b^n)^3$$

```
>$&expand((a^n-b^n)^3)
```

$$-b^{3\,n} + 3\,a^n\,b^{2\,n} - 3\,a^{2\,n}\,b^n + a^{3\,n}$$

4) Kalikan

$$(y^b - z^c)(y^b + z^c)$$

```
>$&expand((y^b-z^c)*(y^b+z^c))
```

$$y^{2\,b} - z^{2\,c}$$

5) Kalikan

$$(t^a + t^{-a})^2$$

```
>$&expand((t^a+t^(-a))^2)
```

$$t^{2\,a} + \frac{1}{t^{2\,a}} + 2$$

2.3

1) Diberikan

$$f(x) = 3x + 1$$

$$g(x) = x^2 - 2x - 6$$

$$h(x) = x^3$$

a. Tentukan nilai

$$(f \circ g)(-1)$$

```
>function f(x):= 3*x+1
>function g(x):= x^2-2*x-6
>function h(x):= x^3
```

3.1

1) Sederhanakan

$$(-5+3i)+(7+8i)$$

```
>$&expand((-5+3*i)+(7+8*i))
```

$$11\,i+2$$

2) Sederhanakan

$$(10+7i)-(5+3i)$$

```
>$&expand((10+7*i)-(5+3*i))
```

$$4\,i+5$$

3) Sederhanakan

$$\sqrt{-4}\cdot\sqrt{-36}$$

```
>$&expand(sqrt(-4)*sqrt(-36))
```

$$-12$$

4) Sederhanakan

$$(3+\sqrt{-16})+(2+\sqrt{-25})$$

```
>$&expand((3+sqrt(-16))+(2+sqrt(-25)))
```

$$9\,i+5$$

5) Sederhanakan

$$\frac{3+2i}{1-i}+\frac{6+2i}{1-i}$$

```
>$&expand (((3+2*i)/(1-i))+((6+2*i)/(1-i)))
```

$$\frac{4\,i}{1-i}+\frac{9}{1-i}$$

3.2

1) Selesaikan

$$(2t^2 + t)^2 - 4(2t^2 + t) + 3 = 0$$

```
>$&expand((2*t^2+t)^2-4*(2*t^2+t)+3=0)
```

$$4\,t^4 + 4\,t^3 - 7\,t^2 - 4\,t + 3 = 0$$

```
>$&solve((2*t^2+t)^2-4*(2*t^2+t)+3=0)
```

$$\left[t = \frac{1}{2}, t = -1, t = 1, t = -\frac{3}{2}\right]$$

2) Selesaikan

$$x^4 - 8x^2 = 9$$

```
>$&solve(x^4-8*x^2=9,x)
```

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

3) Selesaikan

$$(x-2)^3 = x^3 - 2$$

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

$$[x = 1]$$

4) Carilah solusi dari

$$x^2 - 2x = 15$$

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

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

5) Carilah solusi dari

$$4x^2 + 3 = x$$

```
>$&solve(4*x^2+3=x)
```

$$\left[x = \frac{1 - \sqrt{47}\, i}{8}, x = \frac{\sqrt{47}\, i + 1}{8}\right]$$

3.4

1) Selesaikan

$$\frac{1}{4} + \frac{1}{5} = \frac{1}{t}$$

```
>$&solve((1/4)+(1/5)=(1/r))
```

$$\left[r = \frac{20}{9}\right]$$

2) Selesaikan

$$\frac{t+1}{3} - \frac{t-1}{2} = 1$$

```
>$&solve(((t+1)/3)-((t-1)/2)=1)
```

$$[t = -1]$$

3) Selesaikan

$$\sqrt{2x+5} = x - 5$$

```
>$&solve(sqrt(2*x+5)=x-5)
```

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

4) Selesaikan

$$\sqrt{y+7}+\sqrt{y+16}=9$$

```
>$&solve(sqrt(y+7)+sqrt(y+16)=9)
```

$$\left[\sqrt{y+16}=9-\sqrt{y+7}\right]$$

5) Selesaikan

$$\sqrt{3x-4}=1$$

```
>$&solve(sqrt(3*x-4)=1)
```

$$\left[x=\frac{5}{3}\right]$$

3.5

1) Selesaikan

$$|x+3|-2=8$$

```
>$solve(abs(x+3)-2=8,x); $solve((x+3)=10)
```

$$[x=7]$$

2) Selesaikan

$$|4x-3|+1=7$$

```
>$&solve(abs(4*x-3)+1=7,x); $solve(4*x-3=6)
```

$$\left[x=\frac{9}{4}\right]$$

3) Selesaikan

$$7-|2x-1|=6$$

```
>$&solve(7-abs(2*x-1)=6); $&solve(2*x-1=1)
```

$$[x=1]$$

4) Selesaikan

$$|2x-4|<-5$$

BAB 3

Selesaikan
a.

$$x+5\sqrt{x}-36=0$$

```
>$&solve(x+5*sqrt(x)-36=0)
```

$$\left[x=36-5\,\sqrt{x}\right]$$

b.

$$\sqrt{x+4}-2=1$$

```
>$&solve(sqrt(x+4)-2=1)
```

$$[x=5]$$

c.

$$R = \sqrt{3np}$$

```
>$&solve(R=sqrt(3*n*p),n)
```

```
Answering "Is R positive, negative or zero?" with "positive"
```

$$\left[n = \frac{R^2}{3\,p}\right]$$

d.

$$|x + 4| = 7$$

```
>$&solve(x+4=7,x)
```

$$[x = 3]$$

e.

$$|x+5|>2$$

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

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

```
>$&fourier_elim[abs(x+5)>2],x)
```

$$fourier\_\_elim_{|x+5|>2}$$

```
Space between commands expected!
Found: x) (character 120)
You can disable this in the Options menu.
Error in:
$&fourier_elim[abs(x+5)>2],x) ...
                             ^
```

4.1

1) Cari 0 pada polinomial fungsi dan kalikan tiap persaman berikut
a.

$$f(x) = (x^2 - 5x + 6)^2$$

```
>$&expand((x^2-5*x+6)^2)
```

$$x^4 - 10\,x^3 + 37\,x^2 - 60\,x + 36$$

```
>px &= x^4-10*x^3+37*x^2-60*x+36; $&px
```

$$x^4 - 10\,x^3 + 37\,x^2 - 60\,x + 36$$

```
>solve(px,1,y=0), px(%)
```

```
Floating point error!
secant:
    x2=x1-y1*(x1-x0)/(y1-y0);
Try "trace errors" to inspect local variables after errors.
solve:
    if eps then return secant(f$,a,b,y;args(),eps=eps);
```