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))))
```

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

Soal Latihan Tambahan

1. Sederhanakan bentuk

$$\left(\frac{24a^{10}b^{-8}c^7}{12a^6b^{-3}c^5}\right)^{-5}$$

```
>$&((24*a^10*b^(-8)*c^7) / (12*a^6*b^(-3)*c^5))^(-5)
```

$$\frac{b^{25}}{32\,a^{20}\,c^{10}}$$

2. Sederhanakan

$$\left(\frac{125p^12q^{-14}r^22}{25p^8q^6r^{-15}}\right)^{-4}$$

```
>$&((125*p^12*q^(-14)*r^22) / (25*p^8*q^6*r^(-15)))^(-4)
```

$$\frac{q^{80}}{625\,p^{16}\,r^{148}}$$

3. Hitunglah hasil operasi berikut

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

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

$$-845$$

4. Jabarkan

$$(m^{x-b}\times n^x+b)^x\times(m^b\times n^{-b})^x$$

```
>$&showev('expand((m^(x-b)*n^(x+b))^x*(m^b*n^(-b))^x))
```

$$expand\left(\left(\frac{m^b}{n^b}\right)^x\,\left(m^{x-b}\,n^{x+b}\right)^x\right)=\left(\frac{m^b}{n^b}\right)^x\,\left(m^{x-b}\,n^{x+b}\right)^x$$

5. Hitunglah

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

$$\frac{r^6\,t^6}{x^{3\,(2\,r-4)}\,y^{3\,(8-2\,t)}}$$

Baris Perintah

Baris perintah dalam euler memuat satu atau beberapa perintah euler dan diikuti oleh semicolon ";" atau coma ",". Penggunaan semicolon artinya mencegah mencetak hasil perintah. Sedangkan penggunaan coma artinya setiap hasil dari perintah akan di cetak semuanya.

Baris perintah dibawah hanya akan mencetak hasil dari ekspresi dan tidak akan mencetak hasil dari perintah yang mungkin ada sebelumnya , misalnya perintah penugasan (assigments) atau perintah format (format commands).

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

```
16.7551608191
```

Perintah harus diikuti dengan spasi, tanpa spasi baris perintah tersebut tidak dapat dijalankan. Baris perintah dibawah mencetak semua hasil perintahnya, karena baris perintah tersebut diikuti oleh koma.

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

```
50.2654824574
100.530964915
```

Baris perintah dijalankan dalam urutan setiap kali kita menekan tombol 'enter'. Jika kita menjalankan baris perintah yang hasilnya bergantung pada baris perintah sebelumnya, maka kita akan mendapatkan hasi yang berbeda, tergantung dengan nilai inputnya.

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

```
0.540302305868
```

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

```
0.857553215846
```

Hasil dari cos x tersebut berbeda karena pada saat menjalankan perintah cos x yang pertama, x disini bernilai x (dari x := 1) dan menghasilkan hasil sekitar 0.540302305868. Sedangkan pada saat menjalankan perintah cos x yang kedua, x disini bernilai 0.540302305868 atau hasil dari cos x sebelumnya. Oleh karena itu kita mendapatkan dua hasil yang berbeda.

Dua baris perintah dihubungkan dengan "...", kedua baris perintah tersebut akan selalu dijalankan secara stimultan atau 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
```

Penggunaan "..." merupakan salah satu fitur yang sangat bagus untuk baris perintah yang panjang. Dengan menekan tombol 'Ctrl+Return' atau 'Ctrl+Enter' kita dapat memecah baris perintah menjadi dua atau lebih baris sehingga kita bisa mendapatkan keterbacaan yang lebih baik. 'Ctrl+Backspace' digunakan untuk menggabungkan baris perintah yang tadi telah dipisahkan. Jika ingin menyembunyikan semua multi baris tersebut, kita dapat memanfaatkan 'Ctrl+L'. Untuk menyembunyikan baris tertentu, awali baris perintah dengan '%+'.

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

A line starting with %% will be completely invisible.

```
81
```

Euler mensupport pengulangan dalam baris perintah, selama pengulangan tersebut masih berada dalam satu baris atau multi baris.

```
>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
```

Untuk tampilan baris perintah yang lebih baik, kita dapat memanfaatkan '...', sebagai berikut.

```
>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 (seperti if, else, dll) juga dapat digunakan dalam EMT.

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

```
Thought so!
```

Saat akan mengeksekusi atau menjalankan baris perintah, posisi kursor dapat berada dimana saja asalkan masih dalam baris perintah. Untuk kembali ke baris perintah sebelumnya atau untuk melompat ke baris perintah selanjutnya, kita dapat menggunakan arrow keys. Atau bisa juga mengklik bagian komen diatas baris perintah untuk menuju ke baris perintah.

Saat kita memindahkan posisi kursor di baris perintah, EMT akan dengan otomatis menghighlight bagian awal dan akhir tanda kurung. Selain itu, perhatikan juga baris statusnya. Setelah tanda kurung buka pada fungsi sqrt() atau akar pangkat dua, baris status akan menunjukan teks petunjuk untuk fungsi tersebut. Jalankan baris perintah dengan return key atau enter.

```
>sqrt(sin(10°)/cos(20°))
```

```
0.429875017772
```

```
>sqrt(4)
```

```
2
```

Untuk mendapatkan bantuan atau petunjuk untuk baris perintah terakhir, buka jendela bantuan dengan F1. Disitu, kita bisa memasukkan teks untuk dicari. Pada baris yang kosong, bantuan untuk jendela bantuan akan ditampilkan. Kita dapat menekan escape untuk menghapus baris, atau untuk menutup jendela bantuan.

Kalian dapat menekan dua kali pada perintah manapun untuk membuka bantuan untuk perintah tersebut. Coba untuk menekan dua kali pada perintah exp dibawah ini.

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

```
2.5
```

Kalian juga dapat menggunakan fitur copy dan paste di Euler. Gunakan Ctrl-C dan Ctrl-V untuk copy dan paste. Untuk menandai teks, tarik mouse atau gunakan shift bersamaan dengan kursor. Selain itu, dapat juga menyalin tanda kurung yang di highlight.

Soal Latihan Tambahan Baris Perintah

```
>x := 4;
>x := 1/2*x
```

```
2
```

```
>x := 1/2*x
```

```
1
```

Hasil dari x tersebut berbeda karena pada saat menjalankan perintah x := 1/2*x yang pertama, nilai dari x disini adalah 4. Sedangkan pada saat menjalankan perintah x := 1/2*x yang kedua, nilai x yang dipakai adalah hasil dari perintah x := 1/2*x pertama, yaitu 2. Oleh karena itu hasil akhir yang kita dapatkan adalah 1.

```
>sqrt(15)
```

```
3.87298334621
```

```
>r := 5; pi*r^2
```

```
78.5398163397
```

```
>sqrt(cos(15°))
```

```
0.982815255421
```

```
>r := 156; h := 15; pi*r^2*h
```

```
1146806.98227
```

Basic Syntax

Euler tahu fungsi matematika yang biasa kita pakai. Seperti yang sudah duperlihatkan di atas, fungsi trigonometri bekerja pada radian atau derajat. Untuk mengubah ke derajat, tambahkan simbol derajat (dengan menggunakan tombol F7) pada nilai yang kita tentukan, atau gunakan fungsi rad(x). Fungsi akar pada Euler disebut dengan sqrt. Penulisan x^(1/2) juga bisa digunakan.

Untuk menyimpan variable, gunakan antara '=' ata ':='. untuk mengklarifikasi, pengenalan pada notebook EMT ini menggunakan yang terakhir, yaitu ':='. Spasi dalam penyimpanan variabel tidak berpengaruh. Kita dapat menuliskan x=1 atau x = 1, keduanya memiliki arti yang sama. Namun, spasi dalam baris perintah perlu diperhatikan karena hal itu dioerlukan.

Perintah yang lebih daru satu dipisahkan dengan ',' atau ';'. Tanda pemisah semicolon menyembunyikan output dari perintah. Di akhir baris perintah, diasumsikan terdapat tanda ',' jika tidak ada tanda semicolon.

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

```
30.65625
```

EMT menggunakan sintaks programming untuk menulis ekspresi matematika. Untuk menulis

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

kita harus menggunakan tanda kurung dengan benar dan menggunakan tanda '/' untuk bentuk pecahan. Perhatikan highlight tanda kurung. Ingat bahwa konstanta Euler e pada EMT disimbolkan dengan E.

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

```
8.77908249441
```

Untuk bisa mengoperasikan ekspresi matematika yang kompleks, kita harus menuliskan ekspresti tersebut dengan format yang sesuai. Contohnya seperti:

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

Ingat bahwa untuk menuliskan pangkat negatif, menggunakan tanda kurung {}

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

```
23.2671801626
```

Selalu ingat untuk menambahkan tanda kurung dalam sub ekspresi yang perlu dioperasikan terlebih dahulu. EMT mempermudah kita dengan menghighlight bagian ekspresi yang termasuk ke dalam tanda kurung tertentu. Untuk menuliskan simbol pi, kita harus menulisnya dengan 'pi' karena dalam EMT tidak tersedia simbol pi.

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

```
0.47619047619
10/21
```

Perintah Euler dapat dalam bentuk ekspresi atau primitive command. Primitive command atau perintah dasar adalah perintah yang tidak dibangun dari perintah lainnya tetapi langsung dieksekusi oleh sistem. Suatu ekspresi dibentuk dari operasi atau fungsi. Jika diperlukan, kita harus menambahkan tanda kurung supaya mempermudah selama perintah di eksekusi. Selalu ingat bahwa EMT menghighlight tanda kurung selama proses editing baris perintah.

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

```
14.4978445072
```

Operasi numerik dalam Euler meliputi

+ untuk operator penambahan
- untuk operator pengurangan
*,/
. untuk operator matriks produk
a^b untuk menuliskan a pangkat b (atau dapat juga ditulis dengan a**b)
n! untuk operator faktorial
dan masih banyak laagi

Berikut adalah beberapa fungsi yang mungkin kalian butuhkan.

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 mempunyai nama lain, misalnya seperti ln, mempunyai makna yang sama dengan log.

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

```
2
0.5
```

```
>sin(30°)
```

```
0.5
```

Pastikan untuk memakai tanda kurung supaya memudahkan perhitungan. 2^3^4 tidak sama dengan (2^3)^4.

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

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

Soal Latihan Tambahan Basic Sytanx

```
>((2/9)+4+6/17)*E^2
```

```
33.8061390147
```

```
>ln(E^(-3))
```

```
-3
```

```
>tan(156°)
```

```
-0.445228685309
```

```
>sin(pi/5)-tan(2*pi)
```

```
0.587785252292
```

```
>(4^11)^3
```

```
7.37869762948e+19
```

Bilangan Asli

Tipe data dalam Euler adalah bilangan asli. Dalam format IEEE, bilangan asli memiliki akurasi sekitar 16 desimal.

```
>longest 1/3
```

```
     0.3333333333333333
```

Representasi ganda internal membutuhkan 8 byte.

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

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

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

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

Soal Latihan Tambahan Bilangan Real

```
>longest 19/15
```

```
      1.266666666666667
```

```
>printdual(19/25)
```

```
1.1000010100011110101110000101000111101011100001010010*2^-1
```

```
>longest 11/147
```

```
    0.07482993197278912
```

```
>printdual(11/147)
```

```
1.0011001010000000110111101110100101011100010011001010*2^-4
```

```
>printhex(11/147)
```

```
 1.3280DEE95C4CA*16^-1
```

Strings

String dalam Euler didefinisikan dengan "..."

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

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

String dapat dihubungkan dengan | atau dengan +. Aturan ini juga berlaku untuk angka, dimana dalam kasus tersebut, angkat tersebut diubah ke dalama bentuk string.

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

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

Fungsi print juga dapat mengonversi angka ke string. Fungsi ini dapat mengambil sejumlah digit dan sejumlah sejumllah tempat desimal dalam satu unit.

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

```
 Golden Ratio : 1.61803
```

Dalam Python, ada string khusus yang tidak dicetak, yaitu string none. None bukan merupakan string melainkan sebuah objek khusus untuk mewakili tidak adanya nilai atau nilai kosong. Jika sebuah fungsi dijalankakn dan tidak ada nilai yang dikembalikan, maka hasil yang akan didapatkan adalah none.

```
>none
```

Untuk mengonversi string ke angka, kalian cukup mengevaluasikan saja. Cara ini bekerja untuk ekspresi berikut.

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

```
 1234.5
```

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

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

```
 affe
 charlie
 bravo
```

String kosong dilambangkan dengan [none]. Vektor string dapat digabungkan.

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

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

String dapat memuat karakter Unicode, yaitu sistem pengkodean karakter yang dirancang untuk mencakup semua karakter dari berbagai bahsa san sistem oenulisan di seluruh dunia. String ini memuat kode UTF-8. Untuk menghasilkan string yang seperti itu, gunakan u"..." dan satu dari entitas HTML.

Unicode string juga dapat digabungkan sama dengan jenis string yang lainnya.

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

```
 = 45°
```

I

Dalam komentar, entitas simbol seperti , dll dapat digunakan. Ini mungkin adalah alternatif cepat untuk Latex. (Detail yang lebih lengkap ada di kolom komentar).

Ada beberapa fungsi untuk membuat atau menganalisis unicode string. Fungsi strtochar() akan mengenali unicode string dan menerjemahkan unicode string tersebut secara tepat.

```
>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]
```

Hasil dari baris perintah tersebut adalah vektor angka unicode. Fungsi untuk mengkorversi adalah chartoutf().

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

```
Ü is a German letter
```

Fungsi utf() dapat menerjemahkan string yang disertai dengan simbol dalam variabel ke dalam unicode string.

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

```
We have =.
```

Hal ini juga berlaku saat menggunakan simbol numerik.

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

```
Ähnliches
```

Soal Latihan Tambahan String

```
>"Fakultas Matematika dan Ilmu Pengetahuan Alam"
```

```
Fakultas Matematika dan Ilmu Pengetahuan Alam
```

```
>"Luas lingkaran dengan jari-jari " + 5 + " cm adalah " + 78.5398163397 + ""
```

```
Luas lingkaran dengan jari-jari 5 cm adalah 78.5398163397
```

```
>none
>a:=["Aplikasi Komputer","Logika dan Himpunan","Geometri Analitik"]
```

```
Aplikasi Komputer
Logika dan Himpunan
Geometri Analitik
```

```
>b:=[none]; a|b|a
```

```
Aplikasi Komputer
Logika dan Himpunan
Geometri Analitik
Aplikasi Komputer
Logika dan Himpunan
Geometri Analitik
```

Nilai Boolean

Boolean values are represented with 1=true or 0=false in Euler. Strings can be compared, just like numbers.Nilai boolean dilambangkan dengan 1=true atau 0=false di Euler.

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

```
0
1
```

Pada bahasa C, "&&" merupakan operator untuk "and" dan "||" adalah operator untuk "or".(Kata "and" dan "or" hanya dapat digunakan pada kondisi untuk "if").

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

```
1
```

Operator boolean mengikuti aturan pada bahasa matriks.

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

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

Kalian dapat menggunakan fungsi nonzeros() untuk mengekstrak elemen tertentu dari vektor. Sebagai contoh, kita menggunakan kondisi isprime(n).

Baris perintah dibawah digunakan untuk memanggil anggota N, yaitu elemen 2 dan bilangan ganjil dari 3 sampai dengan 99.

```
>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]
```

Baris perintah dibawah digunakan untuk memanggil bilangan prima yang ada dalam vektor N tadi.

```
>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]
```

Soal Latihan Tambahan Nilai Boolean

```
>pi<E
```

```
0
```

```
>5>E && E>1
```

```
1
```

```
>(17:25)>20
```

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

```
>A=1:5:50
```

```
[1,   6,   11,   16,   21,   26,   31,   36,   41,   46]
```

```
>A[nonzeros(isprime(A))]
```

```
[11,   31,   41]
```

Output Formats

Hasil default yang dicetak dalam EMT ada 12 digit. Untuk memastikan kita melihat hasil default nya, kita perlu mereset format tersebut.

```
>defformat; pi
```

```
3.14159265359
```

EMT menggunakan standar IEEE untuk mencetak angka double, yaitu hingga 16 digit desimal. Untuk melihat versi lengkap dari digit angka, gunakan perintah "longestformat" atau juga bisa gunakan operator "longest" untuk memperlihatkan hasil dalam versi yang paling panjang.

```
>longest pi
```

```
      3.141592653589793
```

Di bawah ini merupakan representasi dari angka double dalam format heksadesimal (basis 16).

```
>printhex(pi)
```

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

Format hasil dapat diubah secara permanen dengan perintah format.

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

```
    0.33333
    3.14159
    0.84147
```

Baris perintah di atas memerintahkan untuk mencetak hasil desimal dari 1/3, pi, dan sin 1 dalam format 5 digit desimal.
Bentuk defaultnya ada 12 digit.

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

```
0.333333333333
```

Fungsi seperti "shortestformat", "shortformat", "longformat" dapat bekerja pada vektor dengan cara sebagai 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), tetapi kita dapat mengubahnya berbeda dengan format default.

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

```
  3.1416
```

Fungsi "longestformat" juga mengatur format skalar.

```
>longestformat; pi
```

```
  3.141592653589793
```

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

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

Akurasi internal EMT adalah sekitar 16 digit desimal, yang merupakan standar IEEE. Angka disimpan dalam format internal ini.
Akan tetapi, fortmat output EMT dapat diatur dengan cara yang lebih fleksibel.

```
>longestformat; pi,
```

```
  3.141592653589793
```

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

```
    3.14159
```

Defaultnya adalah defformat().

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

Terdapat operator pendek yang hanya akan mencetak satu nilai. Operator "longest" akan mencetak semua digit angka yang valid.

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

```
        4.934802200544679
```

Selain itu, ada juga operator pendek yang digunakan untuk mencetak hasil dalam format pecahan. Kita telah menggunakan operator tersebut diatas.

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

```
  25/12
```

Karena format internal menggunakan cara biner untuk menyimpan angka, maka nilai 0.1 tidal akan terwakili dengan tepat. Kesalahan bertambah sedikit, seperti yang kalian lihat dalam perhitungan berikut ini.

```
>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
```

Tapi dengan default "longformat", kalian tidak akan menyadari hal ini. Untuk kenyamanan, output angka yang sangat kecil adalah 0.

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

```
  0
```

Soal Latihan Tambahan Output Format

```
>defformat; E
```

```
  2.71828182846
```

```
>longest E
```

```
        2.718281828459045
```

```
>short E
```

```
  2.7183
```

```
>printhex(E)
```

```
  2.B7E151628AED2*16^0
```

```
>format(12,4); E
```

```
2.7183
```

Ekspresi

String atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang mana dapat dioperasikan dengan EMT. Untuk ini, gunakan tanda kurung setelah ekspresi. Jika kalian bermaksud untuk menggunakan string sebagai ekspresi, gunakan konvensi untuk menamainya "fx" atau "fxy" dsb. Ekspresi lebih diutamakan atau didahulukan daripada fungsi.

Variable global dapat digunakan dalam evaluasi.

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

```
12.56637061435917
```

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

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

```
-0.9195
```

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

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

```
36.0000
```

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

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

```
45.0000
```

Untuk referensi, kami menyatakan bahwa koleksi panggilan (dibahas di bagian lain) dapat berisi ekspresi. Jadi kita akan membuat contoh di atas sebagai berikut.

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

```
45.0000
```

Ekspresi di x tidak sering digunakan seperti fungsi. Ingat bahwa mendefinisikan suatu fungsi dengan nama yang sama seperti ekspresi simbolik global akan menghaous variabel ini untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

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

```
      12.0000
```

Untuk memudahkan, ekspresi simbolik atau numerik harus diberi nama dengan fx, fxy, dsb. Skema penamaan ini harus tidak digunakan untuk fungsi.

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

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

Bentuk khusus dari suatu ekspresi memungkinkan variabel manapun sebagai parameter tanpa nama untuk mengevaluasi ekspresi, tidak hanya "x" \, "yy", dst. Untuk itu, mulailah sebuah ekspresi dengan "@(variables)...".

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

```
  @(a,b) a^2+b^2
      41.0000
```

Hal ini memungkinkan untuk memanipulasi ekspresi pada variavel lain untuk fungsi EMT yang mana membutuhkan ekspresi di "x".

Cara paling dasar untuk mendefinisikan suatu fungsi sederhana adalah dengan cara menyimpan formula dalam ekspresi simbolik atau numerik. Jika variabel utama adalah x, ekspresi dapat dievaluasi seperti halnya fungsi.

Seperti yang terlihat pada contoh di bawah, variabel global dapat terlihat selama proses evaluasi.

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

```
      -0.4750
```

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

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

```
      -0.4250
```

Suatu ekspresi tidak harus berbentuk simbolik. Hal ini diperlukan, jika ekspresi memuat fungsi, yang mana hanya dikenal di kernel numerik, bukan Maxim.

Soal Latihan Tambahan Ekspresi

```
>b:=3; function f(expr,x,b) := expr(x);
>f("b*x^2",4,6)
```

```
48.0000
```

```
>f &= x^3;
>function f(x) := x^4;
>f(2)
```

```
16.0000
```

```
>fx &= x^5-a^2;
>a=7; f(3)
```

```
81.0000
```

Matematika Simbolik

EMT mengerjakan matematika simbolik dengan bantuan Maxima. Untuk lebih lengkap, mulai dengan tutorial berikut, atau search referensi untuk Maxima di internet. Expert di Maxima harus mengingat bahwa teedapat perbedaan pada sintaks, antara sintaks asli Maxima dan sintaks bawaan dari ekspresi simbolik di EMT.

Matematika simbolik diintegrasikan dengan mulus ke dalam Euler dengan &. Setiap ekspresi yang dimulai dengan & adalah ekspresi simbolik. Ekspresi ini dievaluasi dan dicetak oleh Maxima.

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

```
>$&44!
```

$$2658271574788448768043625811014615890319638528000000000$$

Dengan ini, kalian 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)
```

$$2481256778$$

Tentu saja Maxima memiliki fungsi yang lebih efektif untuk masalah ini.

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

$$2481256778$$

Untuk belajar lebih mengenai fungsi spesifik, klik dua kali pada fungsi tersebut.

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

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

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

Jika kalian ingin mengganti x dengan nilai spesifik tertentu, gunakan "with:.

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

$$120$$

Dengan cara tersebut, kalian dapat menggunakan solusi dari persamaan dalam persamaan lain.

Ekspresi simbolik dicetak dengan Maxima pada bentuk 2D. Alasannya adalah karena adanya bendera simbolis khusus pada string.

Seperti yang akan kalian lihat di contoh sebelum dan selanjutnya, jika kalian sudah menginstall LaTex, kalian dapat mencetak ekspresi simbolik dengan LaTex. Jika tidak, perintah selanjutnya akan error.

Untuk mencetak ekspresi simbolik dengan LaTex, gunakan $ di depan & (atau kalian bisa abaikan &) sebelum perintah. Jangan jalankan perintah Maxima dengan $, jika kalian belum meningtall LaTex.

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

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

Ekspresi simbolik diuraikan oleh Euler. Jika kalian membutuhkan sintaks yang kompeks dalam satu ekspresi, kalian dapat mengapit ekspresi dengan "...". Menggunakan lebih dari satu ekspresi sederhana memungkinkan, tetapi sangat tidak disarankan.

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

```
25
```

Ekspresi simbolik diuraikan oleh Euler. Jika kalian membutuhkan kompleksitas untuk kelengkapan, kami menyatakan bahwa ekspresi simbolik dapat digunakan dalam program, tetapi harus diapit oleh tanda kutip. Selain itu, akan jauh lebih efektif untuk memanggil Maxima pada saat kompilasi jika memungkinkan.

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

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

Sekali lagi, % merujuk pada hasil sebelumnya.

Untuk membuatnya menjadi lebih mudah, kami menyimpan solusi untuk variabel simbolik. Variavel 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 langsnug dari perintah Maxima juga tersedia. Mulai dari baris perintah dengan "::". Sintaks Maxima disesuaikan dengan sintaks EMT (disebut "mode kompabilitas").

```
>&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 kalian sudah mahir menggunakan Maxima, kalian mungkin ingin menggunakan sintaks asli Maxima. Kalian dapat melakukan ini dengan ":::".

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

```
 2
g
```

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

```
 3  x
x  E
```

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

Variabel seperti itu dapat digunakan dalam ekspresi simbolik lain. Ingat bahwa pada perintah berikut ini, 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 mengevaluasi ekspresi dengan nilai tertentu dari variabel, kalian dapat menggunakan operator "with" .

Perintah berikut ini juga mendemostrasikan bahwa Maxima dapat mengecaluasi ekspresi 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 LaTex kode untuk ekspresi, kalian dapat menggunakan perintah tex.

```
>tex(fx)
```

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

Ekspresi simbolik dapat dievaluasikan sama seperti ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

Di ekspresi simbolik, hal ini tidak berlaku, karena Maxima tidak mensupportnya. Sebaliknya, gunakan sintaks "with" (bentuk yang lebih baik dari perintah at(...) Maxima).

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

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

Penugasan juga dapat berupa simbolik.

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

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

Perintah "solve" akan menyelesaikan ekspresi simbolik untuk variabel dalam Maxima. Hasilnya adalah solusi dalam bentuk vektor.

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

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

Bandingkan dengan perintah numerik "solve" dalam Euler, yang membutuhkan nilai aawal, dan pilihan target tujuan.

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

```
1.56155281281
```

Nilai numerik dari solusi simbolik dapat dihitung dengan evaluasi hasil simbolik. Euler akan membaca penugasan x= dst. Jika kalian tidak membutuhkan hasil numerik untuk perhitungan lebih lanjut, kalian juga bisa 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 tertentu, dapat menggunakan "with" dan index.

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

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

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

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

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

$$2$$

Ekspresi simbolik dapat memiliki bendera, yang bermakna perlakuan khusus di Maxima. Bebebrapa bendera dapat digunakan sebagai perintah, yang lainnya tidak. Bendera ditambahkan dengan "|" (bentuk yang lebih baik dari "ev(...,flags)").

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

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

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

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

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

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

Soal Latihan Tambahan Matematika Simbolik

```
>$& 35!/(12!*23!)
```

$$834451800$$

```
>$binomial(35,12)
```

$$834451800$$

```
>$binomial(x,8)
```

$$\frac{\left(x-7\right)\,\left(x-6\right)\,\left(x-5\right)\,\left(x-4\right)\,\left(x-3\right)\,\left(x-2\right)\,\left(x-1\right)\,x}{40320}$$

```
>$solve(x-7)/(2+x)
```

$$\left[\frac{x}{x+2}=\frac{7}{x+2} \right]$$

```
>$&expand((a-b)^4)
```

$$b^4-4\,a\,b^3+6\,a^2\,b^2-4\,a^3\,b+a^4$$

Fungsi

Di EMT, fungsi adalah program yang didefinikan dengan perintah "function". Fungsi dapat berupa fungsi satu baris atau fungsi multibaris. Fungsi satu baris dapat berupa numerik atau simbolik. Fungsi satu baris numerik didefinisikan dengan ":=".

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

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

```
>f(2)
```

```
4.472135955
```

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

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

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

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

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

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

```
0.786151377757
```

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

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

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

```
0.7071
```

Sebaiknya kita menghapus redefinisi dari sin.

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

```
0.7071
```

Soal Latihan Tambahan Fungsi

```
>function f(x) := sqrt(8*x^5*9*x)
>f(3)
```

```
229.1026
```

```
>f(1:2:10)
```

```
  8.4853    229.1026    1060.6602    2910.4515    6185.7701
```

Default Parameter

Fungsi numerik dapat mempunyai default parameter.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

```
16
```

Menetapkannya akan menimpa nilai default.

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

```
80
```

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

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

```
16
```

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

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

```
24
```

Tetapi parameter yang ditetapkan akan menggantikan nilai global.
Jika argumen tidak ada dalam daftar parameter yang telah ditentukan sebelumnya, argumen tersebut harus dideklarasikan dengan ":="!

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

```
20
```

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

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

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

Fungsi simbolik dapat digunakan di epsresi simbolik

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

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

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

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

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

```
178.635099908
```

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

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

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

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

```
0.703467422498
```

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

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

```
0.703467422498
```

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

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

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

$$(x+2)^{n}$$

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

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

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

```
625
```

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

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

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

$$P\left(x , 4\right)-Q\left(x , 3\right)$$

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

$$Q\left(x , 3\right)\,P\left(x , 4\right)$$

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

$$\frac{P\left(x , 4\right)}{Q\left(x , 1\right)}$$

$$\frac{P\left(x , 4\right)}{Q\left(x , 1\right)}$$

$$\frac{P\left(x , 4\right)}{Q\left(x , 1\right)}$$

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

$$x^3-x$$

dengan &= fungsi ini bersifat simbolik, dan dapat digunakan dalam ekspresi simbolik lainnya.

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

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

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

lateks: f(x) = \int_1^x t^t \, dt,

yang tidak dapat dievaluasi secara simbolik.

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

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

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

Fungsi dapat mempunyai 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, dimungkingkan untuk menggunakan parameter yang ditetapkan.

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

```
2
```

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

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

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

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

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

```
17
```

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

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

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

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

$$\mathit{lapl}\left(y^4 - 6\,x^2\,y^2 + x^4, x, y\right)$$

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

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

Untuk meringkas
- &= mendefinisikan fungsi simbolik,

```
 := mendefinisikan fungsi numerik,


 &&= mendefinisikan fungsi simbolik murni.
```

Soal Latihan Tambahan Deafult Parameter

```
>function f(x,a=5) := x^(-1/3)*2*a
>f(3)
```

```
       6.9336
```

```
>f(3,2)
```

```
       2.7734
```

```
>function a(x) &= (2*x^7)+3*exp(-x); $&a(x)
```

$$3\,e^{-x}+2\,x^7$$

```
>$&diff(a(x),x), $&% with x=1/3
```

$$\frac{14}{729}-3\,e^{-\frac{1}{3}}$$

$$\frac{14}{729}-3\,e^{-\frac{1}{3}}$$

```
>function A(x) &= diff(a(x),x); $&A(c)
```

$$14\,c^6-3\,e^{-c}$$

```
>solve(&a(x),0.3)
```

```
-1.0911
```

Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolik.

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

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

```
1.41421356237
```

Hal ini juga bekerja untuk ekspresi simbolik. Perhatikan 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])
```

$$\left[\left[x=\left(-\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)\,\left(\frac{\sqrt{\frac{4\,a^3\,e^3+\left(27\,b\,c^2-12\,a^2\,b\,d\right)\,e^2+12\,a\,b^2\,d^2\,e-4\,b^3\,d^3}{b}}}{2\,3^{\frac{3}{2}}\,b}+\frac{c\,e}{2\,b}\right)^{\frac{1}{3}}-\frac{\left(\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)\,\left(a\,e-b\,d\right)}{3\,b\,\left(\frac{\sqrt{\frac{4\,a^3\,e^3+\left(27\,b\,c^2-12\,a^2\,b\,d\right)\,e^2+12\,a\,b^2\,d^2\,e-4\,b^3\,d^3}{b}}}{2\,3^{\frac{3}{2}}\,b}\right.}\right.\right.$$

```
>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 adalah 2. Dalam solve(), nilai target default y=0 dapat diubah dengan variabel yang ditetapkan.

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

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

```
0.966715594851
2
```

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

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

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

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

```
>longest sol()
```

```
    -0.6180339887498949       1.618033988749895
```

Untuk menggunakan solusi secara simbolis di ekspresi yang lain, cara termudah adalah dengan "with".

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

$$0$$

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

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

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

Fungsi f() dapat melihat variabel global. Akan tetapi, 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 mengoper parameter tambahan ke f() adalah dengan menggunakan sebuah daftar yang berisi nama fungsi dan parameternya (cara lainnya adalah dengan menggunakan parameter titik koma).

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

```
  2.54116291558
```

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

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

```
        2.5412
```

Soal Latihan Tambahan Memecahkan Ekspresi

```
>px &= solve((y-2)*(y+2)*(y^2+4),y); $&px
```

$$[y=-2\,i,y=2\,i,y=-2,y=2]$$

```
>ax &= (2*x-7)*(2*x+7); $&ax
```

$$(2\,x-7)\,(2\,x+7)$$

```
>solve(ax,1,y=3), ax(%)
```

```
        3.6056
        3.0000
```

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:/Aplikasi Komputer/Euler x64/maxima/share/maxima/5.35.1/sha\
re/fourier_elim/fourier_elim.lisp
```

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

$$fourier\_\_elim\left(\left[x^2-1>0\right],[x]\right)$$

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

$$fourier\_\_elim\left(\left[x^2-1<0\right],[x]\right)$$

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

$$fourier\_\_elim\left(\left[x^2-1\neq 0\right],[x]\right)$$

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

$$fourier\_\_elim\left([x\neq 6],[x]\right)$$

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

$$fourier\_\_elim\left([x<1,x>1],[x]\right)$$

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

$$fourier\_\_elim\left([-\infty<x,x<\infty],[x]\right)$$

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

$$fourier\_\_elim\left(\left[x^3-1>0\right],[x]\right)$$

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

$$fourier\_\_elim\left(\left[\cos x < \frac{1}{2}\right], [x]\right)$$

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

$$fourier\_\_elim\left([y - x < 5, x - y < 7, 10 < y], [x, y]\right)$$

```
>$&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])
```

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

```
>&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])
```

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

Soal Latihan Tambahan Menyelesaikan Pertidaksamaan

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

```
        C:/Aplikasi Komputer/Euler x64/maxima/share/maxima/5.35.1/sha\
  re/fourier_elim/fourier_elim.lisp
```

```
>$&fourier_elim([2*y-3>=1-y+5],[y])
```

$$[y = 3] \vee [3 < y]$$

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

$$[1 < x] \vee [x < -2]$$

```
>$&fourier_elim([0.1*x^3-0.6*x^2-0.1*x+2<0],[x])
```

$$\mathit{fourier\_elim}\left(\left[0.1\,x^3 - 0.6\,x^2 - 0.1\,x + 2 < 0\right], [x]\right)$$

```
>$&fourier_elim([(x+6)/(x-2)>(x-8)/(x-5)],[x])
```

$$[5 < x] \vee \left[2 < x, x < \frac{46}{11}\right]$$

Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.

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

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

```
              1                   2
              3                   4
```

Matriks produk atau hasil kali matriks dilambangkan dengan sebuah titik.

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

```
              3
              4
```

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

```
  [3,  4]
```

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

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

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

```
            11
            25
```

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

```
             1             0
             0             1
```

Poin terpenting dalam bahasa matriks adalah semua fungsi dan operator bekerja atau berlaku untuk setiap 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
```

Perkalian dari matriks dengan inverse matriks itu sendiri akan menghasilkan matriks identitas.

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

```
              1              0
              0              1
```

Perkalian matriks dan perpangkatan matriks.

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

```
              1              4
              9             16
```

b^2 tidak sama dengan b*b. b^2 artinya kalian memangkatkan masing masing elemen matriks dengan 2. Sedangkan b*b atrinya kalian mengalikan sebuah matriks (dengan aturan perkalian matriks biasa).

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

```
              9
             16
```

Jika dalam operasi matriks tersebut terdapat perkalian vektor atau skalar, maka operasi tersebut dieksekusi dengan cara perkalian biasa antara vektor atau skalar dengan masing masing elemen matriks.

```
>2*A
```

```
              2             4
              6             8
```

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

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

```
              1             4
              3             8
```

Jika itu adalah vektor baris, maka berlaku untuk semua kolom

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

```
              2             6
              6            12
```

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

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

```
>(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 bukan matriks produk!

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

```
55
```

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

```
55
```

Bahkan operasi 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 dengn 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, di mana 1 berarti benar.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
 112
```

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

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

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

Ini mengembalikan indeks elemen, yang bukan nol.

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

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

Indeks ini dapat digunakan untuk menetapkan elemen ke suatu nilai.

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

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

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

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

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

And it is possible to get the elements in a vector.

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

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

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

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

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

Kita dapat menggunakan ini untuk mengekstrak nilai maksimal di tiap baris.

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

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

Ini tentu saja sama dengan fungsi max().

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

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

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

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

```
  ex is not a variable!
  Error in:
  j=(1:rows(A))'|ex[,4], mget(-A,j) ...
                       ^
```

Soal Latihan Tambahan Bahasa Matriks

```
>A=[1,2;4,3]; B=[-3,5;2,-1]; C=[1,-1;-1,1]; D=[1,1;1,1];
>A-B
```

```
      4.0000     -3.0000
      2.0000      4.0000
```

```
>A.B
```

```
      1.0000      3.0000
     -6.0000     17.0000
```

```
>C.D
```

```
      0.0000        0.0000
      0.0000        0.0000
```

Cari hasil kali invers matriks koefisien dengan matriks konstanta berikut.

$$x + 2y - 3z = 9$$

$$2x - y + 2z = -8$$

$$3x - y - 4z = 3$$

```
>A=[1,2,-3;2,-1,2;3,-1,-4]; B=[9;-8;3];
>X=A\B
```

```
     -1.0000
      2.0000
     -2.0000
```

```
>(-1)*D
```

```
     -1.0000       -1.0000
     -1.0000       -1.0000
```

```
>E=[-1,0,7;3,-5,2]; F=[6;-4;1];
>E.F
```

```
      1.0000
     40.0000
```

Fungsi Matriks Lainnya (Membangun Matriks)

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

(Untuk membangun sebuah matriks, kita dapat menumpuk satu matriks di atas matriks lainnya. Jika keduanya tidak memiliki jumlah kolom yang sama, kolom yang lebih pendek akan diisi dengan 0.)
Translated with DeepL.com (free version)

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

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

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

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

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

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

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

```
>A|1
```

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

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

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

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

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

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

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

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

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

Untuk mendapatkan ukuran yang sama dengan A, kita dapat menggunakan fungsi berikut.

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

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

Untuk vektor, terdapat fungsi lenght().

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

```
  9
```

Ada banyak fungsi lain yang juga menghahsilkan matriks.

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

```
                1              1
                1              1
```

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

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

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

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

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

```
          0.66566       0.831835
            0.977       0.544258
```

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

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

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

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

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

Mari kita coba.

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

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

Fungsi multdup() menduplikasi elemen vektor.

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

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

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

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

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

Untuk merotasi, Euler mempunyai fungsi rotleft() dan rotright.

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

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

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

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

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

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

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

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

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

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

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

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

Kita mulai dengan matriks identitas.

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

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

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

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

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

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

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

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

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

Diagonal sebuah matriks juga dapat diekstrak dari matriks. Untuk mendemonstrasikan hal ini, kami 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 diagonal.

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

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

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

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

```
Variable d not found!
Error in:
fraction A/d' ...
             ^
```

```
>v=random(1,3)
```

```
      0.2623      0.8666      0.5361
```

```
>A=ones(3,3)
```

```
      1.0000      1.0000      1.0000
      1.0000      1.0000      1.0000
      1.0000      1.0000      1.0000
```

```
>B=A|v
```

```
  Real 3 x 6 matrix

      1.0000      1.0000      1.0000      0.2623     ...
      1.0000      1.0000      1.0000      0.0000     ...
      1.0000      1.0000      1.0000      0.0000     ...
```

```
>flipx(B)
```

```
  Real 3 x 6 matrix

      0.5361      0.8666      0.2623      1.0000     ...
      0.0000      0.0000      0.0000      1.0000     ...
      0.0000      0.0000      0.0000      1.0000     ...
```

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

```
      1.0000      1.0000      1.0000
```

```
>fraction d/v
```

```
  [278845/73128,  94046/81499,  828523/444202]
```

Vektorisasi

Hampir semua fungsi di Euler juga dapat digunakan untuk input matriks dan vektor, jika hal ini masuk akal. Sebagai contoh, fungsi sqrt() menghitung akar kuadrat dari semua elemen vektor atau matriks.

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

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

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

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

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

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

lateks: s = t^3-t

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

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

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

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

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

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

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

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

```
55
```

Secara default, vektor baris dicetak dalam format ringkas.

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

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

Untuk matriks, operator khusus . menyatakan perkalian matriks, dan A' menyatakan transposisi. Matriks 1x1 dapat digunakan seperti halnya bilangan real.

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

```
5
25
```

Untuk mentranspose matriks kita dapat gunakan apostrophe.

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

```
              1
              2
              3
              4
```

Jadi kita dapat menghitung matriks A dikali vektor b.

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

```
             30
             70
```

Ingat bahwa v masih 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 berfungsi seperti bilangan real.

```
>v.v'
```

```
 30
```

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

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

```
 30
```

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ini adalah ringkasan aturannya.

- Sebuah fungsi yang diterapkan pada vektor atau matriks diterapkan pada setiap elemen.

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

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

Misalnya, nilai skalar dikalikan vektor mengalikan nilai dengan setiap elemen vektor. Atau matriks dikali vektor (dengan *, bukan .) memperluas vektor ke ukuran matriks dengan menduplikasinya.
Berikut ini adalah kasus sederhana dengan operator ^.

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

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

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

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

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

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

```
>v.v'
```

```
14
```

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

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

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

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

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

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

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

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

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

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

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

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

```
6
```

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

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

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

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

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

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

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

```
4
```

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

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

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

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

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

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

Sekarang kita menambahkan kolom ke matriks.

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

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

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

Pada contoh berikut, kita menghitung t[j]^i untuk i dari 1 hingga n. Kita mendapatkan sebuah matriks, di mana setiap baris adalah tabel t^i untuk satu i. Dengan kata lain, matriks tersebut memiliki elemen-elemen lateks: a_{i, j} = t_j^i, \qad 1 \le j \le 101, \qad 1 \le i \le n

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

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

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

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

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

```
      0.0000       2.0504      13.7251     113.3356    1241.0333
```

Soal Latihan Tambahan Vektorisasi

```
>A=redim(5:20,4,4)
```

```
       5.0000        6.0000        7.0000        8.0000
       9.0000       10.0000       11.0000       12.0000
      13.0000       14.0000       15.0000       16.0000
      17.0000       18.0000       19.0000       20.0000
```

```
>B=sqrt(rotright(A))
```

```
       2.8284        2.2361        2.4495        2.6458
       3.4641        3.0000        3.1623        3.3166
       4.0000        3.6056        3.7417        3.8730
       4.4721        4.1231        4.2426        4.3589
```

```
>shortest A*A'
```

```
      25      54      91 1.4e+02
      54   1e+02 1.5e+02 2.2e+02
      91 1.5e+02 2.3e+02   3e+02
  1.4e+02 2.2e+02   3e+02   4e+02
```

```
>A.A'
```

```
     174.0000      278.0000      382.0000      486.0000
     278.0000      446.0000      614.0000      782.0000
     382.0000      614.0000      846.0000     1078.0000
     486.0000      782.0000     1078.0000     1374.0000
```

```
>B[2,2]
```

```
       3.0000
```

```
>C=A|A'
```

```
  Real 4 x 8 matrix

       5.0000        6.0000        7.0000        8.0000     ...
       9.0000       10.0000       11.0000       12.0000     ...
      13.0000       14.0000       15.0000       16.0000     ...
      17.0000       18.0000       19.0000       20.0000     ...
```

Sub-Matriks dan Elemen Matriks

Untuk mengakses elemen matriks, gunakan notasi kurung.

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

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

Kita dapat mengakses baris lengkap dari sebuah matriks.

```
>A[2]
```

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

Untuk vektor baris atau kolom, ini mengembalikan elemen vektor.

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

```
  2
```

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

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

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

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

Di sini kita menginginkan baris pertama dan kedua dari A.

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

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

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

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

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

Trik indeks juga bekerja dengan kolom.

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

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

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

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

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

```
                    3
                    6
                    9
```

Sebagai alternatif, biarkan indeks pertama kosong.

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

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

Kita juga bisa mendapatkan baris terakhir dari A.

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

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

Now let us change elements of A by assigning a submatrix of A to some value. This does in fact change the stored matrix A.

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

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

Kita juga dapat menetapkan nilai pada baris di A.

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

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

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

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

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

Selain itu, beberapa jalan pintas diperbolehkan.

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

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

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

```
>A[4]
```

Soal Latihan Tambahan Sub Matriks dan Elemen Matriks

```
>N=random(4,5)
```

```
      0.4935      0.6013      0.6595      0.9675      0.1932
      0.9359      0.0729      0.9890      0.0104      0.3566
      0.5214      0.4289      0.1681      0.1827      0.2880
      0.7500      0.4729      0.3244      0.3404      0.1955
```

```
>N[3,4]
```

```
      0.1827
```

```
>N[[2,3]]
```

```
      0.9359      0.0729      0.9890      0.0104      0.3566
      0.5214      0.4289      0.1681      0.1827      0.2880
```

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

```
      0.9359      0.0729      0.9890
      0.5214      0.4289      0.1681
      0.7500      0.4729      0.3244
```

```
>N[:,5]
```

```
      0.1932
      0.3566
      0.2880
      0.1955
```

Mengurutkan dan Mengacak

Fungsi sort() mengurutkan vektor baris.

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

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

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

Mari kita mengacak sebuah vektor.

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

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

Indeks berisi urutan v yang tepat.

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

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

Hal ini juga berlaku untuk vektor string.

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

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

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

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

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

```
Real 1 x 6 matrix

     4.0000      1.0000      5.0000      2.0000     ...
```

Fungsi unique mengembalikan daftar terurut dari elemen unik sebuah vektor.

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

```
Real 1 x 10 matrix

     5.0000       3.0000       6.0000       8.0000      ...
Real 1 x 7 matrix

     3.0000       4.0000       5.0000       6.0000      ...
```

Hal ini juga berlaku untuk vektor string.

```
>unique(s)
```

```
a
aa
d
e
```

Soal Latihan Tambahan Mengurutkan dan Mengacak

```
>M=sort([99,104,32,17,11,56,29])
```

```
Real 1 x 7 matrix

    11.0000      17.0000      29.0000      32.0000      ...
```

```
>shuffle(M)
```

```
Real 1 x 7 matrix

    17.0000      29.0000      32.0000      99.0000      ...
```

```
>v=["1","4","7","4","0","12","1","5"]
```

```
1
4
7
4
0
12
1
5
```

```
>unique(v)
```

```
0
1
12
4
5
7
```

Aljabar Linier

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

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

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

```
                  -4
                 4.5
```

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

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

```
    8.790745908981989e-13
```

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

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

```
      1.0000      2.0000      3.0000
      4.0000      5.0000      6.0000
      7.0000      8.0000      9.0000
```

Determinan dari matriks ini adalah 0.

```
>det(A)
```

```
      0.0000
```

Soal Latihan Tambahan Aljabar Linier

```
>A=[4,2;3,-1]; b=[11;2];
>det(A)
```

```
    -10.0000
```

```
>A\b
```

```
1.5000
2.5000
```

```
>inv(A)
```

```
0.1000      0.2000
0.3000     -0.4000
```

Matriks Simbolik

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Untuk mengekstrak vektor eigen tertentu, diperlukan pengindeksan yang cermat.

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

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

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

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

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

```
        1.0000        4.0000
        5.0000        2.0000
```

Dalam ekspresi simbolik, gunakan with.

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

Akses ke baris matriks simbolik bekerja seperti halnya matriks numerik.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
                      1                        0
                      0                        1
```

Sebagai contoh, nilai eigen dari A dapat diperasikan secara numerik.

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

```
      16.1168     -1.1168     -0.0000
```

Atau secara simbolik. Lihat tutorial mengenai Maxima untuk lebih detail menegnai hal ini.

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

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

Soal Latihan Tambahan Matriks Simbolik

```
>A = [-1,2,-3,1;-1,1,1,-1;1,1,1,1;-1,1,-1,-1]; $A
```

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

```
>invert(A)
```

```
    -0.5000     -1.0000      0.7500      1.2500
     0.0000      0.0000      0.5000      0.5000
     0.0000      0.5000      0.0000     -0.5000
     0.5000      0.5000     -0.2500     -1.2500
```

```
>B &= [6,13,a;8,b,19;c,11,2]; $B
```

$$\begin{pmatrix}6 & 13 & a\\ 8 & b & 19\\ c & 11 & 2\end{pmatrix}$$

```
>$&B with [a=10,b=6,c=1]
```

$$\begin{pmatrix}6 & 13 & 10\\ 8 & 6 & 19\\ 1 & 11 & 2\end{pmatrix}$$

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

$$\begin{pmatrix}\frac{2b-209}{a(88-bc)+13(19c-16)+6(2b-209)} & \frac{11a-26}{a(88-bc)+13(19c-16)+6(2b-209)} & \frac{247-ab}{a(88-bc)+13(19c-16)+6(2b-209)}\\ \frac{19c-16}{a(88-bc)+13(19c-16)+6(2b-209)} & \frac{12-ac}{a(88-bc)+13(19c-16)+6(2b-209)} & \frac{8a-114}{a(88-bc)+13(19c-16)+6(2b-209)}\\ \frac{88-bc}{a(88-bc)+13(19c-16)+6(2b-209)} & \frac{13c-66}{a(88-bc)+13(19c-16)+6(2b-209)} & \frac{6b-104}{a(88-bc)+13(19c-16)+6(2b-209)}\end{pmatrix}$$

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

$$\left[\left[-\frac{23\left(\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)}{9\left(\frac{4\sqrt{29}}{3^{\frac{3}{2}}}+\frac{19}{27}\right)^{\frac{1}{3}}}+\left(\frac{4\sqrt{29}}{3^{\frac{3}{2}}}+\frac{19}{27}\right)^{\frac{1}{3}}\left(-\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)-\frac{2}{3},\left(\frac{4\sqrt{29}}{3^{\frac{3}{2}}}+\frac{19}{27}\right)^{\frac{1}{3}}\left(\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)-\frac{23\left(-\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)}{9\left(\frac{4\sqrt{29}}{3^{\frac{3}{2}}}+\frac{19}{27}\right)^{\frac{1}{3}}}-\frac{2}{3},\left(4\sqrt{}\right.\right.\right.$$

Nilai Numerik dalam Ekspresi simbolik

Ekspresi simbolik hanyalah sebuah string yang berisi ekspresi. Jika kita ingin mendefinisikan nilai baik untuk ekspresi simbolik maupun ekspresi numerik, kita harus menggunakan “&:=”.

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

```
              1         3.14159
              4               5
```

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

```
>$&A
```

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

Untuk menghindari ini, terdapat fungsi "mxmset(variable)".

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

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

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

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

$$1.414213562373095$$

Ketepatan angka floating point yang besar dapat diubah.

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

```
        3.14159265358979323846264338327950288419716939937510582097494\
  4592307816406286208998628034825342117068b0
```

Variabel numerik dapat digunakan dalam ekspresi simbolik apa pun dengan menggunakan “@var”.

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

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

$$-5.424777960769379$$

Soal Latihan Tambahan Nilai Numerik dalam Ekspresi Simbolik

```
>A &:= [8,12;pi,sqrt(pi)]
```

```
        8.0000      12.0000
        3.1416       1.7725
```

```
>$&A
```

$$\begin{pmatrix} 8 & 12 \\ \frac{1146408}{364913} & \frac{582540}{328663} \end{pmatrix}$$

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

$$\begin{pmatrix} 8 & 12 \\ 3.141592653589793 & 1.772453850905516 \end{pmatrix}$$

```
>$&float(E)
```

$$2.718281828459045$$

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

$$-18.43975113426441$$

Demo - Suku Bunga

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

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

```
>K=5000
```

```
5000.00
```

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

```
>K*1.03
```

```
5150.00
```

Euler juga akan memahami sintaks berikut ini.

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

```
5150.00
```

Tetapi lebih mudah untuk menggunakan faktor

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

```
   1.03
5150.00
```

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

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

```
6719.58
```

Untuk tujuan kita, kita bisa menetapkan formatnya menjadi 2 digit setelah titik desimal.

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

```
6719.58
```

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

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

```
Starting from 5000$ you get 6719.58$.
```

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

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

```
  Real 1 x 11 matrix

      5000.00      5150.00      5304.50      5463.64      ...
```

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

```
>short 0:10
```

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

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

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

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

adalah vektor faktor q^0 hingga q^10. Ini dikalikan dengan K, dan kita mendapatkan vektor nilai.

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

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

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

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

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

```
                  1271.61
                1271.6071
```

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

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

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

```
  Real 1 x 11 matrix

      5000.00      5150.00      5304.50      5463.64      ...
```

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

```
>VKr'
```

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

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

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

```
     5150.00
     5000.00     5150.00     5304.50
```

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

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

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

```
     6719.60
     6719.58
```

Perbedaannya sangat kecil.

Soal Latihan Tambahan Demo-Suku Bunga

1. Sebuah rumah dijual dengan harga $98,000 dengan uang muka sebanyak $16,000. Jika seseorang meminjam uang untuk membeli rumah tersebut dengan lama pinjaman 25 tahun dan suku bunga 6.5%, berapa jumlah yang harus dibayarkan oleh peminjam setiap bulannya.

```
>K=98000-16000, q=1+6.5%, format(12,2); (K*q^25)/(12*25)
```

```
    82000.00
        1.06
     1319.57
```

Jumlah yang harus dibayarkan setiap bulannya adalah $1319.57

2. Sebuah rumah dijual dengan harga $124,000 dengan uang muka sebanyak $20,000. Jika seseorang meminjam uang untuk membeli rumah tersebut dengan lama pinjaman 30 tahun dan suku bunga 5.75%, berapa jumlah yang harus dibayarkan oleh peminjam setiap bulannya.

```
>K=124000-20000, q=1+5.75%, format(12,2); (K*q^30)/(12*30)
```

```
 104000.0000
      1.0575
     1545.76
```

Jumlah yang harus dibayarkan setiap bulannya adalah $1545.76

3. Sebuah rumah dijual dengan harga $135,000 dengan uang muka sebanyak $18,000. Jika seseorang meminjam uang untuk membeli rumah tersebut dengan lama pinjaman 20 tahun dan suku bunga 7.5%, berapa jumlah yang harus dibayarkan oleh peminjam setiap bulannya.

```
>K=135000-18000, q=1+7.5%, format(12,2); (K*q^20)/(12*20)
```

```
   117000.00
        1.07
     2070.83
```

Jumlah yang harus dibayarkan setiap bulannya adalah $2070.83

4. Sebuah rumah dijual dengan harga $151,000 dengan uang muka sebanyak $21,000. Jika seseorang meminjam uang untuk membeli rumah tersebut dengan lama pinjaman 25 tahun dan suku bunga 6.25%, berapa jumlah yang harus dibayarkan oleh peminjam setiap bulannya.

```
>K=151000-21000, q=1+6.25%, format(12,2); (K*q^25)/(12*25)
```

```
   130000.00
        1.06
     1972.63
```

Jumlah yang harus dibayarkan setiap bulannya adalah $1972.63

Menyelesaikan Persamaan

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

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

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

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

```
 Real 1 x 11 matrix

     5000.00     5350.00     5710.50     6081.82     ...
```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?.

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

```
  Real 1 x 11 matrix

      5000.00     4950.00     4898.50     4845.45     ...
```

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

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

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

```
  Real 1 x 51 matrix

      5000.00     4950.00     4898.50     4845.45     ...
```

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

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

```
         48.00
```

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

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

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

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

```
        -19.83
         47.00
```

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

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

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

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

Perulangannya sama seperti di atas

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

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

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

Mari kita coba sebuah tes.

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

```
-19.83
```

Sekarang kita dapat menyelesaikan masalah kita.

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

```
3.15
```

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

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

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

```
-336.08
```

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

Soal Latihan Tambahan Penyelesaiian Persamaan

1. Selesaikan persamaan dibawah ini

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

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

2. Selesaikan persamaan dibawah ini

$$9(2x+8) = 20 - (x+5)$$

```
>$&solve(9*(2*x+8)=20-(x+5))
```

3. Selesaikan persamaan dibawah ini

$$4(3y-1)-6=5(y+2)$$

```
>$&solve(4*(3*y-1)-6=5*(y+2))
```

4. Selesaikan persamaan dibawah ini

$$3y^2+8y+4=0$$

```
>$&solve(3*y^2+8*y+4=0)
```

5. Selesaikan persamaan dibawah ini

$$5x^2-75=0$$

```
>$&solve(5*x^2-75=0)
```

$$\left[x=-\sqrt{15},x=\sqrt{15}\right]$$

Solusi Simbolik untuk Masalah Suku Bunga

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

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

Sekarang kita dapat mengiterasi ini

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

```
Maxima said:
part: argument must be a non-atomic expression; found K
 -- an error. To debug this try: debugmode(true);

Error in:
 $&op(op(op(op(K)))),$&expand(%) ...
                     ^
```

Kita melihat sebuah pola. Setelah n periode, kita memiliki

lateks: K_n = q^n K + R (1+q+\ titik-titik+q^{n-1}) = q^n K + \frac{q^n-1}{q-1} R

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

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

Ini sedikit rumit. Penjumlahan dievaluasi dengan flag "simpsum" untuk menguranginya menjadi hasil bagi. Mari kita buat sebuah fungsi untuk ini.

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

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

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

```
-19.82504734650985
-19.82504734652684
```

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

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

```
20.51
```

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

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

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

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

Biasanya rumus ini diberikan dalam bentuk

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

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

Kita dapat menyelesaikan laju R secara simbolis.

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

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

Tentu saja, kita memiliki batas berikut.

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

Jelasnya, tanpa bunga kita harus membayar kembali 10 suku bunga 500.

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

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