Neyfa Khalisha Amaluna (23030130014)

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:

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

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

Baris Perintah

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

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

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

```
16.7552
```

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

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

```
50.2654824574
100.530964915
```

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

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

```
0.540302305868
```

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

```
0.857553215846
```

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

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

```
1.41666666667
1.41421568627
1.41421356237
```

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

Untuk melipat semua multi-baris, tekan Ctrl+L. Kemudian garis-garis berikutnya hanya akan terlihat, jika salah satu dari mereka memiliki fokus. Untuk melipat satu baris multi-baris, mulai baris pertama dengan

"%+ ".

```
>%+ x=4+5; ...
  // This line will not be visible once the cursor is off the line
```

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

```
81
```

Euler mendukung perulangan dalam baris perintah, selama perulangan tersebut masuk ke dalam satu baris tunggal atau beberapa baris. Dalam program, tentu saja pembatasan ini tidak berlaku. Untuk informasi lebih lanjut, baca pengantar berikut ini.

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

```
1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237
```

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

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

```
1.41421356237
```

Struktur bersyarat juga bisa digunakan.

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

```
Thought so!
```

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

Ketika Anda menggerakkan kursor di sepanjang baris, pasangan tanda kurung atau tanda kurung pembuka dan penutup akan disorot. Juga, perhatikan baris status. Setelah tanda kurung pembuka dari fungsi sqrt(), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol return.

```
>sqrt(sin(10°)/cos(20°))
```

```
0.429875017772
```

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

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

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

```
2.5
```

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

Sintaksis Dasar

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

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

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma menekan output dari perintah. Pada akhir baris perintah, "," diasumsikan, jika ";" tidak ada.

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

```
30.65625
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

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

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

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

```
8.77908249441
```

Untuk menghitung ekspresi yang rumit seperti

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

Anda harus memasukkannya dalam bentuk baris.

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

```
23.2671801626
```

Letakkan tanda kurung di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyorot ekspresi yang diselesaikan oleh tanda kurung penutup. Anda juga harus memasukkan nama “pi” untuk huruf Yunani pi.

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

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

```
0.47619047619
10/21
```

Perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi terbuat dari operator dan fungsi. Jika perlu, ekspresi tersebut harus mengandung tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, mengatur tanda kurung adalah ide yang bagus. Perhatikan bahwa EMT menampilkan tanda kurung pembuka dan penutup saat mengedit baris perintah.

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

```
14.4978445072
```

Operator numerik Euler meliputi

+ unary atau operator plus

- unary atau operator minus

*, /

. produk matriks

pangkat a^b untuk a positif atau bilangan bulat b (a**b juga bisa digunakan)

n! operator faktorial

dan masih banyak lagi.

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

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

Beberapa perintah memiliki alias, misalnya ln untuk log.

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

```
2
0.5
```

```
>sin(30°)
```

```
0.5
```

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

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

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

Bilangan Real

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

```
>longest 1/3
```

```
        0.3333333333333333
```

Representasi ganda internal membutuhkan 8 byte.

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

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

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

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

String

String dalam Euler didefinisikan dengan "...".

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

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

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

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

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

Fungsi cetak juga mengonversi angka ke string. Fungsi ini dapat mengambil sejumlah digit dan sejumlah tempat (0 untuk output padat), dan secara optimal satu unit.

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

```
Golden Ratio : 1.61803
```

Ada string khusus tidak ada, yang tidak mencetak. Dikembalikan oleh beberapa fungsi, ketika hasilnya tidak penting. (Dikembalikan secara otomatis, jika fungsi tidak memiliki pernyataan pengembalian).

```
>none
```

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

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

```
1234.5
```

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

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

```
affe
charlie
bravo
```

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

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

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

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

String Unicode dapat digabungkan seperti string lainnya.

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

```
α = 45°
```

I

Dalam komentar, entitas yang sama seperti α, β dll. dapat digunakan. Ini bisa menjadi alternatif yang cepat untuk Latex. (Detail lebih lanjut tentang komentar di bawah).

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

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

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

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

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

```
Ü is a German letter
```

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

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

```
We have α=β.
```

Dimungkinkan juga untuk menggunakan entitas numerik.

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

```
Ähnliches
```

Nilai Boolean

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

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

```
0
1
```

“dan” adalah operator ‘&&’ dan ‘atau’ adalah operator ‘||’, seperti dalam bahasa C. (Kata “dan” dan “atau” hanya dapat digunakan dalam kondisi “jika”).

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

```
1
```

Operator Boolean mematuhi aturan bahasa matriks.

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

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

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

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

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

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

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

Format Keluaran

Format output default EMT mencetak 12 digit. Untuk memastikan bahwa kita melihat format default, kita atur ulang formatnya.

```
>defformat; pi
```

```
3.14159265359
```

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

```
>longest pi
```

```
      3.141592653589793
```

Berikut ini adalah representasi heksadesimal internal dari angka ganda.

```
>printhex(pi)
```

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

Format output dapat diubah secara permanen dengan perintah format.

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

```
    0.33333
    3.14159
    0.84147
```

Standarnya adalah format(12).

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

```
0.333333333333
```

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

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

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

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

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

```
3.1416
```

Fungsi “longestformat” juga menetapkan format skalar.

```
>longestformat; pi
```

```
3.141592653589793
```

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

```
format terpendek format terpendek format panjang, format terpanjang
format (panjang, digit) format bagus (panjang)
format pecahan (panjang)
defformat
```

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

Tetapi format keluaran EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

```
3.141592653589793
```

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

```
  3.14159
```

Standarnya adalah defformat().

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

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

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

```
4.934802200544679
```

Ada juga operator singkat untuk mencetak hasil dalam format pecahan. Kami sudah menggunakannya di atas.

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

```
25/12
```

Karena format internal menggunakan cara biner untuk menyimpan angka, maka nilai 0,1 tidak akan terwakili dengan tepat. Kesalahan bertambah sedikit, seperti yang Anda 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
```

Tetapi, dengan “longformat” default, Anda tidak akan melihat 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
```

Ekspresi

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

Variabel global dapat digunakan dalam evaluasi.

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

```
12.56637061435917
```

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

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

```
-0.919535764538
```

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

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

```
36
```

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

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

```
45
```

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

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

```
45
```

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

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

```
12
```

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

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

Bentuk khusus dari sebuah ekspresi memungkinkan variabel apa pun sebagai parameter tanpa nama untuk evaluasi ekspresi, bukan hanya “x”, “y”, dll. Untuk ini, mulailah ekspresi dengan “@(variabel)...”.

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

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

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

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

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

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

```
-0.475
```

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

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

```
-0.425
```

Sebuah ekspresi tidak perlu berbentuk simbolik. Hal ini diperlukan, jika ekspresi mengandung fungsi-fungsi, yang hanya dikenal di kernel numerik, bukan di Maxima.

Matematika Simbolik

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

Matematika simbolik diintegrasikan secara mulus ke dalam Euler dengan &. Ekspresi apapun yang dimulai dengan & adalah sebuah 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 cara ini, Anda dapat menghitung hasil yang besar secara tepat. Mari kita hitung

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

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

$$2481256778$$

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

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

$$2481256778$$

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik dua kali pada fungsi tersebut. Sebagai contoh, coba klik dua kali pada “&binomial” di baris perintah sebelumnya. Ini akan membuka dokumentasi Maxima yang disediakan oleh pembuat program tersebut.

Anda akan mengetahui bahwa perintah-perintah berikut ini juga dapat digunakan.

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

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

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

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

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

```
Maxima said:
incorrect syntax: with is not an infix operator
binomial(x,3) with
                  ^

Error in:
```

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

Dengan begitu, Anda dapat menggunakan solusi dari sebuah persamaan dalam persamaan lain.

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

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

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

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

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

Ekspresi simbolik diuraikan oleh Euler. Jika Anda membutuhkan sintaks yang kompleks dalam satu ekspresi, Anda dapat mengapit ekspresi dalam "...". Menggunakan lebih dari satu ekspresi sederhana dimungkinkan, tetapi sangat tidak disarankan.

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

                                  25
```

Untuk kelengkapan, kami menyatakan bahwa ekspresi simbolik dapat digunakan dalam program, tetapi harus diapit dengan 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
```

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

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

Sekali lagi, % mengacu pada hasil sebelumnya.

Untuk mempermudah, kita menyimpan solusi ke dalam sebuah variabel simbolik. Variabel simbolik didefinisikan dengan "&=".

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

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

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

```
>&factor(20!)

                          2432902008176640000


>::: factor(10!)

                               8  4  2
                              2  3  5  7


>:: factor(20!)

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

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

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

                                   2
                                  g
```

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

```
 3  x
x  E
```

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

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

```
     5
125 E
```

```
18551.64488782208
```

```
>fx(5)
```

```
18551.6448878
```

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

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

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

```
      10         5
1000 E    - 125 E
```

```
2.20079141499189e+7
```

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

$$\frac{4\,\left(x-1\right)\,x^2\,\left(3\,x^4+8\,x^3+8\,x^2+8\,x+3\right)}{\left(x^4+1\right)^3}$$

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

```
>tex(fx)
```

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

Ekspresi simbolik dapat dievaluasi seperti halnya ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

Dalam ekspresi simbolik, hal ini tidak dapat dilakukan, karena Maxima tidak mendukungnya. Sebagai gantinya, gunakan sintaks “with” (bentuk yang lebih baik dari perintah at(...) pada Maxima).

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

```
Maxima said:
incorrect syntax: with is not an infix operator
fx with
      ^

Error in:
 $&fx with x=1/2 ...
                ^
```

Penugasan ini juga bisa bersifat simbolis.

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

```
Maxima said:
incorrect syntax: with is not an infix operator
fx with
      ^

Error in:
 $&fx with x=1+t ...
                ^
```

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

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

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

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

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

```
Function solve not found.
Try list ... to find functions!
Error in:
solve("x^2+x",1,y=4) ...
                    ^
```

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

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

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

Untuk mendapatkan solusi simbolik yang spesifik, seseorang dapat menggunakan "dengan" dan indeks.

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

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

```
Maxima said:
incorrect syntax: with is not an infix operator
x2:x with
        ^

Error in:
 $&solve(x^2+x=1,x), x2 &= x with %[2]; $&x2 ...
                                    ^
```

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

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

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

```
Maxima said:
incorrect syntax: with is not an infix operator
x*y with
       ^

Error in:
... x+y=3,x^2+y^2=5],[x,y]); $&sol, $&x*y with sol[1] ...
                                                     ^
```

Ekspresi simbolik dapat memiliki bendera, yang menunjukkan perlakuan khusus dalam Maxima. Beberapa bendera dapat digunakan sebagai perintah juga, sementara yang lainnya tidak bisa. Bendera ditambahkan dengan "|" (sebuah bentuk yang lebih baik dari "ev(...,bendera)")

```
>Symbolic expressions can have flags, which indicate a special treatment in Maxima. Some flags can be used as
```

```
Variable Symbolic not found!
Error in:
Symbolic expressions can have flags, which indicate a special  ...
        ^
```

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

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

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

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

Fungsi

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

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

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

```
>f(2)
```

```
4.472135955
```

Fungsi ini juga akan bekerja untuk vektor, mengikuti bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi tersebut telah tervektorisasi.

```
>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 digambarkan. Alih-alih ekspresi, kita hanya perlu memberikan nama fungsi.

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

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

```
0.786151377757
```

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

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

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

```
0.707106781187
```

Kita sebaiknya menghapus redefinisi dosa ini.

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

```
0.707106781187
```

Parameter Default

Fungsi numerik dapat memiliki parameter default.

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

Mengabaikan parameter ini akan menggunakan nilai default.

```
>f(4)
```

```
16
```

Mengaturnya akan menimpa nilai default.

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

```
80
```

Sebuah parameter yang ditugaskan 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 mengakses variabel global.

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

```
24
```

Namun, parameter yang ditetapkan akan mengesampingkan nilai global.

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

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

```
20
```

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

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

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

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

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

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

```
Maxima said:
incorrect syntax: with is not an infix operator
x-%e^-x+3*x^2) with
                 ^

Error in:
 $&diff(g(x),x), $&% with x=4/3 ...
                            ^
```

Mereka juga dapat digunakan dalam ungkapan numerik. Tentu saja, ini hanya akan berhasil jika EMT dapat menginterpretasikan segala sesuatu di dalam fungsi tersebut.

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

```
178.635099908
```

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik lainnya.

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

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

```
0.703467422498
```

Pekerjaan berikut juga, 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(%)
```

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

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

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

```
625
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$x^3-x$$

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

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

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

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

yang tidak dapat dievaluasi secara simbolis.

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

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

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

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

```
2
6.7
```

Selain itu, adalah mungkin untuk menggunakan parameter yang ditetapkan.

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

```
2
```

Seringkali, kita ingin menggunakan fungsi untuk vektor di satu tempat, dan untuk elemen individu di tempat lain. Ini mungkin dilakukan dengan parameter vektor.

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

Fungsi simbolis seperti itu dapat digunakan untuk variabel simbolis.

Namun, fungsi ini juga dapat digunakan untuk vektor numerik.

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

```
17
```

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

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

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

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

$$(y\,I+x)^4$$

$$lapl\left(\left(y\,I+x\right)^4,x,y\right)$$

Tetapi tentu saja, mereka dapat digunakan dalam ungkapan simbolis atau dalam definisi fungsi simbolis.

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

$$lapl\left(\left(y^2+x\right)^5,x,y\right)$$

Untuk merangkum

- &= mendefinisikan fungsi simbolik,
- := mendefinisikan fungsi numerik,
- &&= mendefinisikan fungsi yang sepenuhnya simbolik.

Menyelesaikan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolik.

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

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

```
1.41421356237
```

Ini juga berlaku untuk ekspresi simbolis. Ambil fungsi berikut.

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

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

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

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

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

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

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

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

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

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

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

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

```
Function solve not found.
Try list ... to find functions!
Error in:
solve(px,1,y=2), px(%) ...
                ^
```

Menyelesaikan ekspresi simbolik dalam bentuk simbolik mengembalikan daftar solusi. Kami menggunakan penyelesai 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()
```

```
Variable longest not found!
Error in:
longest sol() ...
         ^
```

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

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

```
Maxima said:
incorrect syntax: with is not an infix operator
x^2 with
       ^

Error in:
 $&x^2 with sol[1], $&expand(x^2-x-1 with sol[2]) ...
                 ^
```

Menyelesaikan sistem persamaan secara simbolis dapat dilakukan dengan vektor persamaan dan penyelesai simbolis solve(). Jawabannya adalah daftar dari 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 mengakses variabel global. Tapi seringkali kita ingin menggunakan parameter lokal.

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

dengan a=3.

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

Salah satu cara untuk meneruskan parameter tambahan ke f() adalah dengan menggunakan daftar yang berisi nama fungsi dan parameter-parameter tersebut (the other way are semicolon parameters).

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

```
2.54116291558
```

Ini juga berlaku untuk ungkapan. Tapi kemudian, elemen daftar yang bernama harus digunakan. (Selebihnya dalah daftar di tutorial tentang sintaks EMT).

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

```
2.54116291558
```

Menyelesaikan Pertidaksamaan

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

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

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

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

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

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

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

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

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

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

$$\mathit{fourier\_elim}\left([x \neq 6], [x]\right)$$

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

$$\mathit{fourier\_elim}\left([x < 1, x > 1], [x]\right)$$

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

$$\mathit{fourier\_elim}\left([-\infty < x, x < \infty], [x]\right)$$

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

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

```
>$&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])
```

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

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

$$fourier\_elim\left(y + x < 5 \wedge x - y > 8, [x, y]\right)$$

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

$$fourier\_elim\left(y + x < 5 \wedge x < 1 \vee x - y > 8, [x, y]\right)$$

```
>&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])
```

$$fourier\_elim\left(\left[\frac{x+6}{x-9} \le 6\right], [x]\right)$$

Bahasa Matriks

Dokumentasi inti EMT berisi pembahasan mendetail tentang bahasa matriks Euler.

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

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

                1             2
                3             4
```

Produk matriks dilambangkan dengan titik.

```
>b=[3;4]

                3
                4
```

```
>b' // transpose b

     [3,   4]
```

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

               -2             1
              1.5          -0.5
```

```
>A.b //perkalian matriks

               11
               25
```

```
>A.inv(A)

                1             0
                0             1
```

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

```
>A.A

                7            10
               15            22
```

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

```
                 1             4
                 9            16
```

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

```
                37            54
                81           118
```

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

```
                37            54
                81           118
```

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

```
                 1             1
                 1             1
```

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

```
          0.333333      0.666667
              0.75             1
```

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

```
                -2
               2.5
```

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

```
                -2
               2.5
```

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

```
                 1             0
                 0             1
```

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

```
                 1             0
                 0             1
```

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

```
                 1             4
                 9            16
```

Ini bukan produk matriks, melainkan perkalian elemen per elemen. Hal yang sama berlaku untuk vektor.

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

```
                 9
                16
```

Jika salah satu operand adalah vektor atau skalar, itu akan diperluas dengan cara yang alami.

```
>2*A
```

```
                 2             4
                 6             8
```

Misalnya, jika operand adalah vektor kolom, elemennya diterapkan ke semua baris A.

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

```
                 1             4
                 3             8
```

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

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

```
                2             6
                6            12
```

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

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

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

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

Sekali lagi, ingatlah bahwa ini bukan produk matriks!

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

```
    55
```

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

```
    55
```

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

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

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

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

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

```
    5
```

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

Kita mendapatkan vektor yang berisi 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-elemen yang tidak nol.

Dalam hal ini, kita mendapatkan indeks dari semua elemen yang lebih besar dari 50.

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

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

Tentu saja, kita bisa menggunakan vektor indeks ini untuk mendapatkan nilai 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 hingga 1000, yang merupakan 5 modulo 11 dan 3 modulo 13.

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

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

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

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

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

```
112
```

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

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

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

Ini mengembalikan indeks dari elemen-elemen yang bukan nol.

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

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

Indeks-indeks ini dapat digunakan untuk menetapkan elemen-elemen ke nilai tertentu.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Fungsi Matriks Lainnya (Membangun Matrix)

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

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

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

Demikian pula, kita dapat melampirkan sebuah matriks di sisi lain, jika keduanya memiliki jumlah baris yang sama.

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

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

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

Ada pengecualian untuk aturan ini. Sebuah bilangan riil yang terlampir pada sebuah matriks akan digunakan sebagai kolom yang diisi dengan bilangan riil tersebut.

```
>A|1
```

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

Ini mungkin untuk membuat matriks dari vektor baris dan vektor kolom.

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

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

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

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

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

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

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

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

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

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

Untuk vektor, ada panjang().

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

```
9
```

Ada banyak fungsi lain yang menghasilkan matriks.

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

```
              1               1
              1               1
```

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

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

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

Juga, sebuah matriks angka acak dapat dihasilkan dengan acak (distribusi seragam) atau normal (distribusi Gauss/Gauß).

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

```
   0.66566     0.831835
     0.977     0.544258
```

Berikut adalah fungsi berguna lainnya, yang merestrukturisasi elemen-elemen dari sebuah matriks menjadi matriks lainnya.

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

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

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

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

Let us test.

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

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

Fungsi multdup() menggandakan elemen-elemen dari sebuah vektor.

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

```
Function multdup not found.
Try list ... to find functions!
Error in:
multdup(1:3,5), multdup(1:3,[2,3,2]) ...
              ^
```

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

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

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

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

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

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

Fungsi khusus adalah drop(v,i), yang menghapus elemen dengan indeks yang ada 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 tersebut. Jika Anda ingin menghapus elemen, Anda perlu menemukan 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 untuk menyertakan indeks yang di luar jangkauan (seperti 0), indeks ganda, atau indeks yang tidak terurut.

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

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

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

Kita mulai dengan matriks identitas.

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

```
                   1             0             0             0             0
                   0             1             0             0             0
                   0             0             1             0             0
                   0             0             0             1             0
                   0             0             0             0             1
```

Kemudian kita atur diagonal bawah (-1) menjadi 1:4.

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

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

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

Berikut adalah sebuah fungsi yang mengembalikan matriks tri-diagonal.

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

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

Diagonal dari sebuah matriks juga dapat diambil dari matriks tersebut. Untuk menunjukkan hal ini, kami merestrukturisasi vektor 1:9 menjadi matriks 3x3.

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

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

Sekarang kita bisa mengekstrak diagonal.

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

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

Misalnya, kita dapat membagi matriks dengan diagonalnya. Bahasa matriks memastikan bahwa vektor kolom d diterapkan pada matriks baris per baris.

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

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

Vektorisasi

Hampir semua fungsi di Euler juga bekerja untuk input matriks dan vektor, kapan pun itu masuk akal.

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

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

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

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

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

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

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

$$s = t^3 - t$$

```
>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 dengan vektor baris akan menghasilkan matriks, jika sebuah operator diterapkan. Dalam hal ini, v' adalah vektor yang ditransposisikan (sebuah 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 cukup berbeda dari produk matriks. Produk matriks dilambangkan dengan titik "." dalam EMT.

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

```
  55
```

Secara default, vektor baris dicetak dalam format yang ringkas.

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

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

Secara default, vektor baris dicetak dalam format yang ringkas.

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

```
  5
  25
```

Untuk mentransposisikan sebuah matriks, kita menggunakan tanda apostrof.

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

```
                  1
                  2
                  3
                  4
```

Jadi kita bisa menghitung matriks A dikalikan vektor b.

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

```
                 30
                 70
```

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

```
>v'.v
```

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

v.v' menghitung norma dari v kuadrat untuk vektor baris v. Hasilnya adalah vektor 1x1, yang berfungsi sama seperti angka nyata.

```
>v.v'
```

```
  30
```

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

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

```
  30
```

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut adalah ringkasan dari aturan-aturannya.

- Sebuah fungsi yang diterapkan pada vektor atau matriks diterapkan pada setiap elemen.
- Sebuah operator yang bekerja pada dua matriks dengan ukuran yang sama diterapkan secara berpasangan pada elemen-elemen matriks tersebut.

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

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

Berikut adalah kasus sederhana dengan operator ^.

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

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

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

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

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

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan *!

```
>v.v'
```

```
14
```

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

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

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

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

Operator : menghasilkan vektor baris yang terdistribusi secara merata, dengan opsi 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 seluruh baris ke-i dari matriks tersebut.

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

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

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

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

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

```
5
8
```

Bentuk singkat untuk: adalah menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

```
4
```

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

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

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

Untuk menggunakan matriks untuk tabel, mari kita reset ke format default, dan hitung 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 dalam sebuah matriks.

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

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

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

Dalam contoh berikut, kita menghitung t[j]^i untuk i dari 1 hingga n. Kita mendapatkan sebuah matriks, di mana setiap baris adalah tabel dari t^i untuk satu i. Yaitu, matriks memiliki elemen latex: a_{i,j} = t_j^i, \quad 1 \le j \le 101, \quad 1 \le i \le n

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

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

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

Kata kunci "map" memetakan fungsi. Fungsi sekarang akan berfungsi untuk vektor angka.

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

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

Sub-Matriks dan Elemen Matriks

Untuk mengakses elemen matriks, gunakan notasi kurung.

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

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

```
      7        8        9
5
```

Kita dapat mengakses satu baris lengkap dari sebuah matriks.

```
>A[2]
```

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

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

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

```
2
```

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

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

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

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

Di sini kami ingin baris pertama dan kedua dari A.

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

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

Kita bahkan bisa mengurutkan A menggunakan vektor indeks. Untuk lebih tepatnya, kami tidak mengubah A di sini, tetapi menghitung versi A yang diurutkan ulang.

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

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

Trik indeks juga berlaku untuk kolom.

Contoh ini memilih semua baris dari A dan kolom kedua serta 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]
```

Sekarang mari kita ubah elemen-elemen A dengan menetapkan sebuah submatriks dari A ke suatu nilai. Ini memang mengubah matriks A yang disimpan.

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

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

Kita juga dapat memberikan nilai pada satu baris dari A.

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

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

Kita 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. Defaultnya adalah pesan kesalahan. Namun, ingatlah bahwa indeks negatif dapat digunakan untuk mengakses elemen-elemen dari sebuah matriks dengan menghitung dari akhir.

```
>A[4]
```

```
A is not a variable!
Error in:
A[4] ...
    ^
```

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 dari vektor yang terurut dalam vektor asli. Ini dapat digunakan untuk mengurutkan vektor lain dengan cara yang sama.

Mari kita acak sebuah vektor.

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

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

Indeks-indeks tersebut berisi urutan yang tepat dari v.

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

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

Ini juga berlaku untuk vektor string.

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

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

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

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

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

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

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

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

Ini juga berlaku untuk vektor string.

```
>unique(s)
```

```
a
aa
d
e
```

Aljabar Linier

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

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

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

```
           -4
          4.5
```

Sebagai contoh lain, kita menghasilkan matriks 200x200 dan jumlah dari baris-barisnya. Kemudian kita menyelesaikan Ax=b menggunakan matriks invers. Kami mengukur kesalahan sebagai deviasi maksimal dari semua elemen terhadap 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, pemodelan linier meminimalkan norma dari kesalahan Ax-b.

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

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

Determinan dari matriks ini adalah 0.

```
>det(A)
```

```
0
```

Matriks Simbolik

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

```
>A &= [a,1,1;1,a,1;1,1,a]; $A
>$&det(A), $&factor(%)
>$&invert(A) with a=0
```

```
Maxima said:
incorrect syntax: with is not an infix operator
invert(A) with
             ^

Error in:
 $&invert(A) with a=0 ...
                    ^
```

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

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

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

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 sebuah eigenvektor tertentu diperlukan pengindeksan yang hati-hati.

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

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

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

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

```
1          4
5          2
```

Dalam ungkapan simbolis, gunakan with.

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

```
Maxima said:
incorrect syntax: with is not an infix operator
A with
     ^

Error in:
 $&A with [a=4,b=5] ...
                 ^
```

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

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

$$[1,a]$$

Sebuah ekspresi simbolik dapat mengandung sebuah penugasan. Dan itu mengubah matriks A.

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

Ada fungsi simbolik di Maxima untuk membuat vektor dan matriks. Untuk ini, silakan merujuk pada 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)
```

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

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

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

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

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

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

```
1                    0
0                    1
```

Misalnya, nilai eigen dari A dapat dihitung secara numerik.

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

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

Atau secara simbolis. Lihat tutorial tentang Maxima untuk rincian lebih lanjut mengenai hal ini.

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

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

Nilai Numerik dalam Ekspresi Simbolik

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

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

```
                1       3.14159
                4             5
```

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

```
>$&A
```

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

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

```
Function mxmset not found.
Try list ... to find functions!
Error in:
mxmset(A); $&A ...
         ^
```

Maxima juga dapat melakukan perhitungan dengan angka floating point, bahkan dengan angka floating besar yang memiliki 32 digit. Namun, evaluasinya jauh lebih lambat.

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

$$1.414213562373095_B \times 10^0$$

$$1.414213562373095$$

Presisi dari angka titik mengapung besar dapat diubah.

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

```
pi
```

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)
```

Demo - Suku Bunga

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

Anggaplah Anda memiliki modal awal sebesar 5000 (dalam dolar).

```
>K=5000
```

```
5000
```

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

```
>K*1.03
```

```
5150
```

Euler juga akan memahami sintaks berikut ini.

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

```
5150
```

Tapi lebih mudah menggunakan faktor tersebut.

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

```
1.03
5150
```

Selama 10 tahun, kita bisa dengan mudah mengalikan faktor-faktor dan mendapatkan nilai akhir dengan suku bunga majemuk.

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

```
6719.58189672
```

Untuk keperluan kita, kita bisa mengatur formatnya menjadi 2 digit setelah titik desimal.

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

```
    6719.58
```

Mari kita cetak angka tersebut 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 sementara dari tahun 1 hingga tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menulis sebuah loop, cukup masukkan saja.

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

```
Real 1 x 11 matrix

    5000.00     5150.00     5304.50     5463.64     ...
```

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

```
>short 0:10
```

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

Maka semua operator dan fungsi dalam Euler dapat diterapkan pada vektor secara 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 dari q^0 hingga q^10. Ini dikalikan dengan K, dan kita mendapatkan vektor nilai.

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

Tentu saja, cara 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, 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 mengulangi tahun-tahun tersebut. Euler memberikan banyak solusi untuk ini.

Cara yang paling mudah adalah fungsi iterate, yang mengulangi fungsi tertentu sejumlah kali.

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

```
Real 1 x 11 matrix

    5000.00     5150.00     5304.50     5463.64     ...
```

Kita bisa mencetak itu dengan cara yang ramah, menggunakan format kita dengan tempat desimal tetap.

```
>VKr'
```

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

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

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

```
        5150.00
        5000.00      5150.00      5304.50
```

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

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

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

```
        6719.60
        6719.58
```

Perbedaannya sangat kecil.

Menyelesaikan Persamaan

Sekarang kita mengambil fungsi yang lebih maju, yang menambahkan sejumlah uang tertentu setiap tahun.

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

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

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

```
    Real 1 x 11 matrix

        5000.00      5350.00      5710.50      6081.82      ...
```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

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

```
    Real 1 x 11 matrix

        5000.00      4950.00      4898.50      4845.45      ...
```

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

Bagaimana kita dapat menentukan berapa lama uang tersebut akan bertahan? Kita harus menulis sebuah loop untuk ini. Cara termudah adalah dengan mengulangi cukup lama.

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

```
    Real 1 x 51 matrix

        5000.00      4950.00      4898.50      4845.45      ...
```

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

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

```
          48.00
```

Alasan untuk ini adalah bahwa nonzeros(VKR<0) mengembalikan vektor indeks i, di mana VKR[i]<0, dan min menghitung indeks minimal.

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

Fungsi iterate() memiliki satu trik lagi. Ini dapat mengambil kondisi akhir sebagai argumen. Kemudian 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. Berapa suku bunga yang akan diterapkan?

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

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

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

Iterasi ini sama seperti di atas.

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

Tapi kami tidak lagi menggunakan nilai global R dalam ungkapan kami. Fungsi seperti iterate() memiliki trik khusus dalam Euler. Anda dapat meneruskan nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R.

Selain itu, kami 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. Kami mengambil nilai awal sebesar 3% untuk algoritma. Fungsi solve() selalu memerlukan nilai awal.

Kita dapat menggunakan fungsi yang sama untuk menyelesaikan pertanyaan berikut: Berapa banyak yang bisa kita ambil per tahun agar modal awal habis setelah 20 tahun dengan asumsi suku bunga 3% per tahun.

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

```
-336.08
```

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

Solusi Simbolis untuk Masalah Tingkat Bunga

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

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

Kita sekarang bisa mengulangi ini.

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

Kami melihat sebuah pola. Setelah n periode kita memiliki

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

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

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

Ini agak sulit. Jumlah dievaluasi dengan bendera "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. Tapi itu lebih efektif.

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

```
-19.82504734650985
-19.82504734652684
```

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

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

```
20.51
```

Jawaban ini mengatakan bahwa itu akan menjadi negatif setelah 21 tahun.

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

Misalkan kita mendapatkan pinjaman sebesar K, dan melakukan n pembayaran sebesar R (dimulai setelah tahun pertama) yang meninggalkan utang sisa sebesar Kn. (di waktu pembayaran terakhir). Rumus untuk ini jelas adalah

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

$$fs\left(K, R, P, n\right) = Kn$$

Biasanya rumus ini diberikan dalam bentuk

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

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

```
Maxima said:
incorrect syntax: with is not an infix operator
equ:(equ with
             ^

Error in:
equ &= (equ with P=100*i); $&equ ...
                         ^
```

Kita dapat menyelesaikan untuk laju R secara simbolis.

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

$$[fs\left(K, R, P, n\right) = Kn]$$

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

Tentu saja, kita memiliki batasan berikut.

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

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

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

Persamaan tersebut juga dapat diselesaikan untuk n. Tampaknya lebih menarik jika kita menerapkan beberapa penyederhanaan padanya.

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

$$[fs\left(K, R, P, n\right) = Kn]$$

Latihan Soal

Operasi Aljabar

1. Selesaikan operasi aljabar berikut.

$$x^2 + 4*x + 3*x^2 - x + 1$$

```
>$& (x^2+4*x+3*x^2-x+1)
```

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

2. Selesaikan operasi aljabar berikut.

$$\left(\frac{2x^2+5x+3}{x+1}\right)$$

```
>$& factor (2*x^2+5*x+3)/(x+1)
```

$$2\,x+3$$

3. Selesaikan operasi aljabar berikut.

$$(a+2b)(a-3)$$

```
>$& expand((a+2*b)*(a-3))
```

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

4. Selesaikan operasi aljabar berikut.

$$(p-3)^2$$

```
>$& (expand((p-3)^2))
```

$$p^2-6\,p+9$$

5. Selesaikan operasi aljabar berikut.

$$(5x+2y+3)(5x+2y-3)$$

```
>$& expand((5*x+2*y+3)*(5*x+2*y-3))
```

$$4\,y^2+20\,x\,y+25\,x^2-9$$

Operasi dan Fungsi Matematika

1. Carilah nilai dari

$$(x^y \times x^{-y})^3$$

```
>$& (x^y*x^(-y))^3
```

$$1$$

2. Carilah nilai dari

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

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

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

3. Sederhanakanlah.

$$\frac{9cos^2\alpha - 25}{2cos\alpha - 2} \times \frac{cos^2\alpha - 1}{6cos\alpha - 10}$$

```
>$& factor ((9*cos(\alpha)^2-25)/(2*cos(\alpha)^2-2))*((cos(\alpha)^2-1)/(6*cos(\alpha)-10))
```

$$\frac{(3\,\cos\alpha - 5)\,(3\,\cos\alpha + 5)\,\left(\cos^2\alpha - 1\right)}{2\,(\cos\alpha - 1)\,(\cos\alpha + 1)\,(6\,\cos\alpha - 10)}$$

4. Berapakah nilai dari sin(60)?

```
>sin(60)
```

```
-0.304811
```

5. Berapakah nilai dari

$$sin(45) + tan(60)$$

```
>sin(45)+tan(60)
```

```
1.17094
```

Bilangan Kompleks

Sederhanakanlah.

1.

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

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

$$4\,i + 5$$

2.

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

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

$$9\,i + 5$$

3.

$$(12 + 3i) + (-8 + 5i)$$

```
>$& (12+3*i)+(-8+5*i)
```

$$8\,i + 4$$

4.

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

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

$$-12$$

5.

$$-2i\,(-8 + 3i)$$

```
>$& (-2*i(-8+3*i))
```

$$-2\,i\,(3\,i - 8)$$

Fungsi-fungsi Buatan Sendiri

1. Tentukan nilai f(4) jika diberikan fungsi

$$f(x) = 2x + 3$$

```
>function f(x):=2*x+3
>f(4)
```

```
11
```

2. Tentukan nilai g(-2) jika diberikan fungsi

$$g(x) = x^2 - 4x + 1$$

```
>function g(x):=x^2-4*x+1
>g(-2)
```

```
13
```

3. Diketahui dua buah fungsi

$$f(x) = 2x + 1$$

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

tentukan

$$(f \circ h)(2)$$

```
>function h(x):=x^2-3
>h(2)
```

```
1
```

```
>function f(x):=2x+1
>f(h(2))
```

```
3
```

4. Tentukan nilai k(3) jika diketahui fungsi

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

```
>function k(x):=(3*x+2)/(x-1)
>k(3)
```

```
5.5
```

5. Tentukan nilai m(3) jika diberikan fungsi

$$m(x) = 2^x$$

```
>function m(x):=2^x
>m(3)
```

```
8
```

Persamaan dan Sistem Pertidaksamaan

Selesaikanlah persamaan-persamaan berikut.

1.

$$3x - 7 = 11$$

```
>$ solve([3*x-7=11],[x])
```

$$[x = 6]$$

2.

$$x^2 - 5x + 6 = 0$$

```
>$ solve([x^2-5*x+6],[x])
```

$$[x = 3, x = 2]$$

3.

$$\begin{cases} 2x + 3y = 7 \\ x - y = 2 \end{cases}$$

```
>$ solve([2*x+3*y=7,x-y=2],[x,y])
```

$$\left[\left[x=\frac{13}{5},y=\frac{3}{5}\right]\right]$$

4.

$$\begin{cases} x+y+z=6 \\ 2x-y+3z=14 \\ -x+4y+2z=-2 \end{cases}$$

```
>$ solve([x+y+z=6,2*x-y+3*z=14,-x+4*y+2*z=-2],[x,y,z])
```

$$\left[\left[x=\frac{32}{7},y=-\frac{1}{7},z=\frac{11}{7}\right]\right]$$

5.

$$2x-9\sqrt{x}+4=0$$

```
>$ solve([2*x-9*sqrt(x)+4=0],[x])
```

$$\left[x=\frac{9\sqrt{x}-4}{2}\right]$$

Pertidaksamaan dan Sistem Pertidaksamaan

Selesaikanlah pertidaksamaan-pertidaksamaan berikut.

1.

$$|2x|\geq 6$$

```
>$ solve([abs(2*x)>=6],[x])
```

2.

$$\left|x+\frac{1}{4}\right|<\frac{1}{2}$$

3.

$$|3x-1|>5x-2$$

4.

$$|p-4|+|p+4|<8$$

5.

$$\left|\frac{2x+1}{3}\right|>5$$

Matriks dan Vektor

Diketahui matriks A dan B sebagai berikut.

```
>A=[3,9;7,1]
```

```
            3             9
            7             1
```

```
>B=[2,2;5,1]
```

```
            2             2
            5             1
```

1. Tentukan

$$A+B,(2B)-A,A\times B,\frac{B}{A}$$

```
>A+B
```

```
            5            11
           12             2
```

```
>2B-A
```

```
            1            -5
            3             1
```

```
>A*B
```

```
            6            18
```

```
              35              1
```

```
>B/A
```

```
        0.666667      0.222222
        0.714286             1
```

2. Diketahui vektor kolom sebagai berikut.

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

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

tentukan g^3.

```
>g^3
```

```
             729
             512
             343
             216
             125
              64
              27
               8
               1
```

3. Dari soal nomor 2, tentukan kuadrat dari tranpose vektor kolom g.

```
>g'
```

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

```
>(g')^2
```

```
[81,  64,  49,  36,  25,  16,  9,  4,  1]
```

4. Diberikan vektor baris sebagai berikut.

```
>N=[27,31,30,31,7,37,25,27,29,25,7,9,54]
```

```
[27,  31,  30,  31,  7,  37,  25,  27,  29,  25,  7,  9,  54]
```

Urutkanlah dari angka terkecil ke terbesar.

```
>sort (N)
```

```
[7,  7,  9,  25,  25,  27,  27,  29,  30,  31,  31,  37,  54]
```

5. Buatlah dalam bentuk vektor baris deret angka dari -75 hingga 95 dengan beda konstan yaitu 5.

```
>r=[-75:5:95]
```

```
[-75,  -70,  -65,  -60,  -55,  -50,  -45,  -40,  -35,  -30,  -25,  -20,
-15,  -10,  -5,  0,  5,  10,  15,  20,  25,  30,  35,  40,  45,  50,
55,  60,  65,  70,  75,  80,  85,  90,  95]
```

Penyelesaian dalam Kehidupan Sehari-hari

Ada sebuah toko alat tulis yang menjual berbagai kebutuhan sekolah. Sasha ingin membeli pensil, pena, dan penggaris untuk sekolahnya di semester baru. Toko tersebut menjual pensil seharga Rp.3000/pcs, pena seharga Rp.5000/pcs, dan penggaris seharga Rp.2000/pcs.

Jawablah pertanyaan nomor 1-3 berikut.

1. Jika Sasha membeli 2 pensil, 1 pena, dan 1 penggaris, berapakah jumlah yang harus Sasha bayar?

x=3000, x=harga jual pensil per-pcs-nya
y=5000, y=harga jual pena per-pcs-nya
z=2000, z=harga jual penggaris per-pcs-nya

```
>function f(x,y,z):=2x+y+z
```

```
>f(3000,5000,2000)
```

```
13000
```

2. Jika Sasha memberi uang kepada penjual sebesar Rp.20000, berapakah uang kembalian yang Sasha terima?

```
>20000-13000
```

```
7000
```

3. Jika penjual memberikan diskon 2% untuk setiap pembelian 2 pensil, berapakah total harga yang seharusnya Sasha bayar?

```
>(2*3000*2/100)+5000+2000
```

```
7120
```

4. Daniel memiliki 15 potong kue dan ingin membagi kue tersebut secara merata kepada 5 temannya. Berapa potong kue yang diterima masing-masing teman Daniel?

misal m=kue yang diterima setiap teman Daniel

```
>$ solve([5*m=15],[m])
```

$$[m = 3]$$

5. Jihan membeli 3 apel dan 2 jeruk dengan total biaya Rp.14000. Jika harga satu apel adalah Rp.3000, berapakah harga satu jeruk?

misalkan P=harga satu jeruk

```
>$ solve([3*3000+2*P=14000],[P])
```

$$[P = 2500]$$