

EMT untuk Perhitungan Aljabar

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

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

Contoh pertama

Menyederhanakan bentuk aljabar:

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

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

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

Menjabarkan:

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

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

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

49. Sederhanakan

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

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

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

50. Sederhanakan

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

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

$$\frac{q^{80}}{625p^{16}r^{28}}$$

```
>$&factor(x^2*y^2-14*x*y+49)// mencari faktor dari persamaan tersebut
```

$$(x y - 7)^2$$

Faktor dari

$$4t^3 + 108$$

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

$$4 (t + 3) (t^2 - 3 t + 9)$$

Faktor dari

$$z^3 + 3z^2 - 3z - 9$$

```
>$&factor(z^3+3*z^2-3*z-9)
```

$$(z + 3) (z^2 - 3)$$

Baris Perintah

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

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

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

16.7551608191

```
>r:=21; t:= 18; pi*r^2*t
```

24937.96

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

```
>opi*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, bisa mendapatkan nilai baru setiap kali mengeksekusi baris kedua.

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

0.540302305868

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

0.857553215846

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

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

```
1.41666666667  
1.41421568627  
1.41421356237
```

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

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

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; // komentar dimasukkan di sini sebelum ...  
>repeat xnew:=(x+2/x)/2; until xnew~=x; ...  
>  x := xnew; ...  
>end; ...  
>x,
```

```
1.41421356237
```

Struktur bersyarat juga bisa diterapkan.

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

Thought so!

```
>if E`pi>pi^E; then "Aplikasi Komputer", endif;
```

Aplikasi Komputer

Ketika 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 dapat mengklik bagian komentar di atas perintah untuk membuka perintah tersebut.

Ketika 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°))
```

0.43


```
>sqrt(cos(65°)/sin(24°))
```

1.02

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.

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

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

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

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

lateks: $\left(\frac{\frac{17}{18} + \frac{1}{2}}{\frac{13}{12} + \frac{1}{2}}\right)^2 \pi$

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 mempelajari 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
```

```
14.50
```

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))
```

```
2.00  
0.50
```

```
>sin(30°)
```

```
0.50
```

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)
```

```
2417851639229258349412352.00  
4096.00  
2417851639229258349412352.00
```

```
>5^4^2
```

152587890625.00

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
```

0.333333333333333

Representasi ganda internal membutuhkan 8 byte.

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

[illegible]

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

$$5.555555555554 \times 16^{-1}$$

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"()
```

```
1234.50
```

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
```

```
= 45°
```

```
I
```

```
>u"&beta; = " + 78 + u"&deg;"
```

```
= 78°
```

Dalam komentar, entitas yang sama seperti `α`, 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 =.
```

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

```
>34>15, "pepaya">"pisang", "air" < "batu"
```

1
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
```

```
1.00
```

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=3|4:3:50 // N berisi elemen 3 dan bilangan-bilangan selisih 3 dari 4 s.d. 50
```

```
[3, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43,  
46, 49]
```

```
>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 // hasil dalam format terpanjang
```

```
3.141592653589793
```

Berikut ini adalah representasi heksadesimal internal dari angka ganda.

```
>printhe(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
```

Defaultnya adalah `format(12)`.

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

```
0.333333333333
```

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

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

```
0.52  0.51  0.84  0.61  0.37  0.36  0.93  0.85  
0.76  0.17  0.87  0.98  0.091 0.96  0.39  0.34  
0.77  0.86  0.84  0.66  0.74  0.95  0.35  0.048
```

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.

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

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

Tetapi format keluaran EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

```
3.141592653589793
```

```
>longestformat; 23/7,
```

```
3.285714285714286
```

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

```
3.14159
```

Default atau 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

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
```

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

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
```

```
>$&23!
```

```
25852016738884976640000
```


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

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

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

2481256778

```
>$& 23!/(17!*6!) // nilai C(44,6)
```

100947

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

```
>$binomial(23,6) // menghitung C(23,6) menggunakan fungsi binomial ()
```

100947

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.

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

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

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

```
>$binomial(y,5) // C(y,5)
```

$$\frac{(y-4)(y-3)(y-2)(y-1)y}{120}$$

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

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

120

```
> $&binomial (y,5) with y=8 // mensubstitusikan y=8 ke kombinasi C(y,5)
```

56

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
```

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

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

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))
```

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

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

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

```
>::: 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
```

$$x^3 e^x$$

$$x^3 e^x$$

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(%)
```

$$125 e^5$$

$$125 e^5$$

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(%)
```

$$1000 E^{10} - 125 E^5$$

2.20079141499189e+7

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

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

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

```
>tex(fx)
```

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

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
```

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

Penugasan ini juga dapat bersifat simbolis.

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

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

35. Faktor dari

$$32x^4 - 40xy^3$$

```
>$&factor(32*x^4-40*x*y^3)
```

$$-8x (5y^3 - 4x^3)$$

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)
```

1.56155281281

Nilai numerik dari solusi simbolik dapat dihitung dengan evaluasi hasil simbolik. Euler akan membaca penugasan $x = \text{dst}$. 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)
```

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

[-3.23607, 1.23607]

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

Untuk mendapatkan solusi simbolik yang spesifik, seseorang dapat menggunakan “with” dan indeks.

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

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

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

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

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

Ekspresi simbolik dapat memiliki bendera, yang menunjukkan perlakuan khusus di Maxima. Beberapa flag dapat digunakan sebagai perintah juga, namun ada juga yang tidak. Bendera ditambahkan dengan “|” (bentuk yang lebih baik dari “ev(...,flags)”)

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

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

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

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

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

$$-8x(5y^3 - 4x^3)$$

31. Solve

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

$$21x + 42 = 11 - (x + 2)$$

```
>%solve(21*x+42 = 11-(x+2),x)
```

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

32. Solve

$$9(2x + 8) = 20 - (x + 5)$$

$$18x + 72 = 20 - (x + 5)$$

```
>%solve(18*x+72=20-(x+5),x)
```

$$[x = -3]$$

35. Solve

$$x^2 + 3x - 28 = 0$$

```
>%solve(x^2+3*x-28=0,x)
```

$$[x = 4, x = -7]$$

39. Solve

$$y^2 + 6y + 9 = 0$$

```
>$&solve(y^2+6*y+9=0,y)
```

$$[y = -3]$$

87. Solve

$$3x^3 + 6x^2 - 27x - 54 = 0$$

```
>$&solve(3*x^3+6*x^2-27*x-54=0,x)
```

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

Fungsi

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

Fungsi satu baris dapat berupa numerik atau simbolik. Fungsi satu baris numerik didefinisikan dengan “:=”.

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

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

```
>f(2)
```

```
4.472135955
```

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

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

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


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

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

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

```
0.786151377757
```

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

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

```
>function overwrite sin (x) := _sin(x°) //tentukan kembali sinus dalam derajat  
>sin(45)
```

```
0.707106781187
```

Sebaiknya kita hilangkan definisi ulang tentang sin ini.

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

Temukan nilai $x=2$ jika

$$f(x) = x^4 - 4x^2 + 3$$

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

3

Parameter Default

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

Menetapkannya akan menimpa nilai default.

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

80

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

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

16

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

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

24

Tetapi parameter yang ditetapkan akan menggantikan nilai global.

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

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

20

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

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

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

Fungsi simbolis dapat digunakan dalam ekspresi simbolis.

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

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

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

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

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

```
178.635099908
```

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

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

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

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

```
0.703467422498
```

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

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

0.703467422498

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

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

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

$$(x + 2)^n$$

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

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

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

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

625

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

$$(2x - 1)^4 + (x + 2)^3$$
$$16x^4 - 31x^3 + 30x^2 + 4x + 9$$

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

$$(2x - 1)^4 - (x + 2)^3$$
$$16x^4 - 33x^3 + 18x^2 - 20x - 7$$
$$16x^4 - 33x^3 + 18x^2 - 20x - 7$$

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

$$(x + 2)^3 (2x - 1)^4$$
$$16x^7 + 64x^6 + 24x^5 - 120x^4 - 15x^3 + 102x^2 - 52x + 8$$

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

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

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

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

$$x^3 - x$$

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

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

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

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

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 memiliki nilai default untuk parameter.

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

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

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

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

2

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

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

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

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

Fungsi simbolik seperti itu dapat digunakan untuk variabel simbolik.

Tetapi fungsi ini juga dapat digunakan untuk vektor numerik.

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

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

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

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

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

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

$$0$$

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

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

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

23. Find $f(1)$, $f(-2)$, and $f(3)$.

$$f(x) = x^3 - 6x^2 + 11x - 6$$

```
>function f(x) := x^3-6*x^2+11*x-6  
>f(1), f(-2), f(3)
```

```
0  
-60  
0
```

26. Find $f(-10)$, $f(2)$, and $f(1)$.

$$2x^4 + x^2 - 10x + 1$$

```
>function f(x):= 2*x^4+x^2-10*x+1  
>f(-10), f(2), f(1)
```

```
20201  
17  
-6
```

28. Find $f(-10)$ and $f(5)$.

$$f(x) = x^5 - 10x^4 + 20x^3 - 5x - 100$$

```
>function f(x) := x^5-10*x^4+20*x^3-5*x-100  
>f(-10), f(5)
```

```
-220050  
-750
```

29. Find $f(2)$, $f(-2)$, $f(3)$, $f(1-\sqrt{2})$

$$f(x) = x^4 - 16$$

```
>function f(x) := x^4-16  
>f(2), f(-2), f(3), f(1-sqrt(2))
```

```
0  
0  
65  
-15.97056274847714
```

30. Find $f(2)$, $f(-2)$, $f(3)$, and $f(2+3i)$

$$f(x) = x^5 + 32$$

```
>function f(x) := x^5+32  
>f(2), f(-2), f(3), f(2+3i)
```

```
64  
0  
275  
154-597i
```

Untuk meringkas

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

Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolik.

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

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

```
1.41421356237
```

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

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

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

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

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

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

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

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

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

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

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

Sekarang kita mencari titik, di mana polinomialnya adalah 2. Dalam solve(), nilai target default y=0 dapat diubah dengan variabel yang ditetapkan.

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 dalam ekspresi lain, cara termudah adalah “with”.

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

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

0

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

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

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

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

lateks: $a^x - x^a = 0.1$

with $a = 3$.

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

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

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

2.54116291558

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

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

2.541162915581603

7. Solve

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

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

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

9. Solve

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

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

$$[x = 5]$$

Menyelesaikan Pertidaksamaan

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

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

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

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

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

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

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

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

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

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

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

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

emptyset

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

universalset

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

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

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

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

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

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

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

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

```
>$fourier_elim((x + y < 5) 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])
```

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

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

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

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

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

62. Solve

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

$$(x-2)(x+5) > (x^2 - 3x)$$

```
>$&fourier_elim([(x^2+3*x-10) > (x^2-3*x)], [x])
```

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

61. Solve

$$2y - 3 \geq 1 - y + 5$$

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

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

37. Solve

$$x^2 + 12 < 4x$$

```
>$fourier_elim([x^2+12<4*x],[x])
```

$$[-x^2 + 4x - 12 > 0]$$

49. Solve

$$x^3 + 5x^2 - 25x \leq 125$$

```
>%fourier_elim([x^3+5*x^2-25*x<=125],[x])
```

$$[x = -5] \vee [x = 5] \vee [x < -5] \vee [-5 < x, x < 5]$$

51. Solve

$$0.1x^3 - 0.6x^2 - 0.1x + 2 < 0$$

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

$$[-0.1x^3 + 0.6x^2 + 0.1x - 2 > 0]$$

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

```
>E=[5,6;7,8]
```

5	6
7	8

Hasil kali matriks dilambangkan dengan sebuah titik.

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

3
4

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

```
[3, 4]
```

```
>E' // transpose matriks E
```

```
5 7  
6 8
```

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

```
-2 1  
1.5 -0.4999999999999999
```

```
>inv(E) // invers matriks E
```

```
-4.0000000000000002 3.0000000000000001  
3.5000000000000002 -2.5000000000000001
```

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

```
11
25
```

```
>E.b // perkalian matriks E dengan b
```

```
39
53
```

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

```
0.9999999999999998 1.110223024625157e-16
0 1
```

```
>E.inv(E)
```

```
1.0000000000000004 0
0 1
```

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

```
>A.A
```

7	10
15	22

```
>E.E
```

67	78
91	106

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

1	4
9	16

```
>E^3 // perpangkatan elemen 3 E
```

125	216
343	512

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

37	54
81	118

```
>E.E.E
```

881	1026
1197	1394

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

37	54
81	118

```
>power(E,4)
```

11587	13494
15743	18334

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

1	1
1	1

```
>E/E // Pembagian elemen
```

1	1
1	1

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

0.3333333333333333	0.6666666666666666
0.75	1

```
>A\b // hasilkali invers A dan b,  $A^{-1}b$ 
```

-2
2.5

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

-1.9999999999999999
2.4999999999999999

```
>inv(E).b
```

```
0  
0.5
```

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

```
1 0  
0 1
```

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

```
1 4.440892098500626e-16  
0 0.9999999999999998
```

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

```
1 4  
9 16
```



```
>E*E
```

25	36
49	64

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

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

9
16

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

```
>2*A
```

2	4
6	8

```
>2*E
```

10	12
14	16

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

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

1	4
3	8

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

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

2	6
6	12

```
> E*[4,5]
```

20	30
28	40

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

```
>E*dup([3,4],2)
```

15	24
21	32

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, 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]
```

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

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

5

```
>sum((2:20)>4) // banyak elemen yang lebih dari 4
```

16

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, ada `mnonzeros()`.

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

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

Ini mengembalikan indeks elemen, yang bukan nol.

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

1	4
2	1
2	2
3	2

Indeks ini dapat digunakan untuk menetapkan elemen ke suatu nilai.

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

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

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

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

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

Dan dimungkinkan untuk mendapatkan elemen-elemen dalam vektor.

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

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


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

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

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

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

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

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

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

```
>max(A)
```

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

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

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

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

Fungsi Matriks Lainnya (Membangun Matriks)

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

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

1	2	3
1	2	3

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

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

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

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

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

```
>A|1
```

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

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

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

1	2	3
1	2	3

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

1	1
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 A, kita dapat menggunakan fungsi berikut ini.

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

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

Untuk vektor, terdapat `length()`.

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

Ada banyak fungsi lain yang menghasilkan matriks.

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

```
1 1
1 1
```

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

Let us test.

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

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

Fungsi multdup() menduplikasi elemen-elemen sebuah vektor.

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

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

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

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

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

Untuk rotasi, Euler memiliki `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 `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
```

Real 5 x 5 matrix

1	...
0	...
0	...
0	...
0	...

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

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

Real 5 x 5 matrix

1	...
1	...
0	...
0	...
0	...

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)
```

Real 5 x 5 matrix

2	...
1	...
0	...
0	...
0	...

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)
```

Real 3 x 3 matrix

1	...
4	...
7	...

Sekarang, kita bisa mengekstrak diagonalnya.

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

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

Misalnya, kita dapat membagi matriks berdasarkan diagonalnya. Bahasa matriks memastikan bahwa vektor kolom d diterapkan ke matriks baris demi baris.

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

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

```
>A=[3,2;5,3] // Pendefinisian matriks A
```

3	2
5	3

```
>inv(A) // Mencari nilai invers A
```

```

-2.999999999999997      1.999999999999999
4.999999999999996      -2.999999999999997

```

```
>A.inv(A) // Mencari nilai matriks A dikali invers A
```

```

0.9999999999999982      1.77635683940025e-15
0                        1

```

```
> A=[1,2,3,4;0,1,3,-5;0,0,1,-2;0,0,0,-1]
```

Real 4 x 4 matrix

```

1      ...
0      ...
0      ...
0      ...

```

```
>inv(A)
```

Real 4 x 4 matrix

```

1      ...
0      ...
0      ...
0      ...

```

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

Temukan nilai determinan dari

```
>B=[5,3;-2,-4]
```

5	3
-2	-4

```
>det(B)  
>C=[-8,6;-1,2]
```

-8	6
-1	2

```
>det(C)
```

```
-9.999999999999998
```

```
>D=[4,-7;-2,3]
```

4
-2

-7
3

```
>det(D)
```

-2.0000000000000001

```
>F=[-2,-sqrt(5);-sqrt(5),3]
```

-2
-2.23606797749979

-2.23606797749979
3

```
>det(F)
```

-11

```
>G=[sqrt(5),-3;4,2]
```

2.23606797749979
4

-3
2

```
>det(G)
```

```
16.47213595499958
```


Vektorisasi

Hampir semua fungsi di Euler juga berfungsi untuk masukan matriks dan vektor, jika ini masuk akal. Misalnya, fungsi `sqrt()` menghitung akar kuadrat dari semua elemen vektor atau matriks.

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

```
[1,  1.414213562373095,  1.732050807568877]
```

```
>sqrt(15:21)
```

```
[3.872983346207417,  4,  4.123105625617661,  4.242640687119285,  
4.358898943540674,  4.47213595499958,  4.58257569495584]
```

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

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

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

Dalam contoh berikut, kita buat vektor nilai $t[i]$ dengan spasi 0,1 dari -1 hingga 1. Kemudian kita buat vektor nilai fungsi

$$s = t^3 - t$$

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

```
[0, 0.17099999999999999, 0.28799999999999999, 0.357, 0.384, 0.375,  
0.33600000000000001, 0.27300000000000001, 0.19200000000000001,  
0.099000000000000014, 1.387778780781446e-16, -0.09899999999999987,  
-0.19199999999999999, -0.27299999999999999, -0.336, -0.375, -0.384,  
-0.35700000000000001, -0.28800000000000001, -0.17100000000000003,  
-4.440892098500626e-16]
```

```
>v=0:2:20; w=v^3-2
```

```
[-2, 6, 62, 214, 510, 998, 1726, 2742, 4094, 5830, 7998]
```

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

Misalnya, vektor kolom dikalikan vektor baris akan mengembang 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

```
>shortest (2:10)*(4:12)'
```

8	12	16	20	24	28	32	36	40
10	15	20	25	30	35	40	45	50
12	18	24	30	36	42	48	54	60
14	21	28	35	42	49	56	63	70
16	24	32	40	48	56	64	72	80
18	27	36	45	54	63	72	81	90
20	30	40	50	60	70	80	90	1e+02
22	33	44	55	66	77	88	99	1.1e+02
24	36	48	60	72	84	96	1.1e+02	1.2e+02

Perhatikan bahwa ini sangat berbeda dari perkalian matriks. Perkalian matriks dilambangkan dengan titik "." dalam EMT.

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

55

```
>(2:10).(4:12)'
```

Secara default, vektor baris dicetak dalam format ringkas.

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

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

Untuk matriks, operator khusus `.` menunjukkan perkalian matriks, dan `A'` menunjukkan transposisi. Matriks 1x1 dapat digunakan seperti bilangan riil.

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

```
5  
25
```

Untuk mentranspos suatu matriks, kita menggunakan tanda apostrof.

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

```
1  
2  
3  
4
```

```
>w=(2:5); w'
```

```
2  
3  
4  
5
```

Jadi kita dapat menghitung matriks A dikali vektor b.

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

```
30  
70
```

```
>H=[2,3,5,8;7,9,10,3]; H.w'
```

```
73  
96
```

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 v kuadrat untuk vektor baris v . Hasilnya adalah vektor 1×1 , yang bekerja seperti bilangan riil.

```
>v.v'
```

```
30
```

```
>w.w'
```

54

```
>w'.w
```

Real 4 x 4 matrix

```
4    ...  
6    ...  
8    ...  
10   ...
```

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

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

30

```
>norm(w)^3
```

396.8173383308749

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ini ringkasan aturannya.

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

Misalnya, nilai skalar dikalikan vektor mengalikan nilai dengan setiap elemen vektor. Atau matriks dikalikan vektor (with *, not .) mengekspansi vektor ke ukuran matriks dengan menduplikasinya.

Berikut ini adalah kasus sederhana dengan operator ^.

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

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

```
> [10,23,54]^3
```

```
[1000, 12167, 157464]
```

```
> [21,56,71]^3
```

```
[9261, 175616, 357911]
```


Berikut ini adalah kasus yang lebih rumit. Vektor baris dikalikan vektor kolom, keduanya diekspansi dengan cara 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'
```

Ada banyak fungsi untuk matriks. Kami memberikan daftar singkatnya. Anda harus merujuk ke dokumentasi untuk informasi lebih lanjut tentang perintah-perintah ini.

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

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

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

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

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

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

Untuk menggabungkan matriks dan vektor ada operator "|" dan "_".

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

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

Elemen-elemen suatu 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. Untuk matriks, ini mengembalikan baris ke-i lengkap dari matriks.

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

6
[7, 8, 9]

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

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

```
[2, 4]
```

```
2  
5  
8
```

Bentuk singkat dari : 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}
```

Matriks juga dapat diratakan, 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 pada tabel, mari kita atur ulang 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 tambahkan 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.

Dalam contoh berikut, kita menghitung $t[j]^i$ untuk i dari 1 hingga n . Kita memperoleh matriks, yang setiap barisnya merupakan tabel t^i untuk satu i . Yaitu, matriks tersebut memiliki elemen
 lateks: $a_{i,j} = t_j^i, \quad 1 \leq j \leq 101, \quad 1 \leq i \leq n$

Fungsi yang tidak berfungsi untuk input vektor harus ”divektorkan”. Ini dapat dicapai dengan kata kunci ”map” dalam definisi fungsi. Kemudian fungsi tersebut akan dievaluasi untuk setiap elemen parameter vektor.

Integrasi numerik `integrate()` hanya berfungsi untuk batas interval skalar. Jadi, kita perlu memvektorkannya.

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

Kata kunci "map" akan memvektorkan fungsi tersebut. Fungsi tersebut 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 tanda kurung.

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

	1	2	3
	4	5	6
	7	8	9
5			

Kita dapat mengakses baris matriks yang lengkap.

```
>A[2]
```

```
[5, 6, 7, 8]
```

```
>A[1]
```

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


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

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

2

Untuk memastikan, Anda mendapatkan baris pertama untuk matriks $1 \times n$ dan $m \times n$, tentukan semua kolom menggunakan indeks kedua yang kosong.

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

[4, 5, 6]

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

Di sini kita menginginkan baris pertama dan kedua dari A.

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

Real 2 x 4 matrix

1	...
5	...

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

Real 2 x 4 matrix

5	...
1	...

Kita bahkan dapat menyusun ulang A menggunakan vektor indeks. Untuk lebih tepatnya, kita tidak mengubah A di sini, tetapi menghitung versi A yang telah disusun ulang.

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

7	8	9
4	5	6
1	2	3

Trik indeks juga berfungsi dengan kolom.

Contoh ini memilih semua baris 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
```

Atau, biarkan indeks pertama kosong.

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

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

Kita juga bisa mendapatkan baris terakhir A.

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

```
[5, 6, 7, 8]
```

```
>A[-2]
```

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

Sekarang mari kita ubah elemen A dengan menetapkan submatriks A ke suatu nilai. Hal ini pada kenyataannya mengubah matriks A yang tersimpan.

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

4	2	3
4	5	6
7	8	9

Kita juga dapat menetapkan nilai ke baris A.

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

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

Kita bahkan dapat menetapkan ke submatriks jika ukurannya 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 yang tidak sesuai batas akan mengembalikan matriks kosong, atau pesan kesalahan, tergantung pada pengaturan sistem. Pesan kesalahan adalah standar. Namun, perlu diingat bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks yang dihitung dari akhir.

```
>A[4]
```

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

Sortir dan Acak

Fungsi `sort()` mengurutkan vektor baris.

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

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

```
>sort([23,67,98,46,13,2,5])
```

```
[2, 5, 13, 23, 46, 67, 98]
```

Seringkali perlu untuk mengetahui indeks vektor yang diurutkan dalam vektor asli. Ini dapat digunakan untuk menyusun ulang vektor lain dengan cara yang sama.

Mari kita acak sebuah vektor.

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

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

```
>w=shuffle(10:30)
```

```
[14, 29, 22, 15, 11, 19, 10, 30, 24, 20, 13, 16, 12, 21,  
27, 25, 26, 18, 17, 28, 23]
```

Indeks berisi urutan v yang tepat.

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

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

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

```
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,  
24, 25, 26, 27, 28, 29, 30]
```

Ini juga berlaku untuk vektor string.

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

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

```
>t=["w","g","c","y","z","hi","op"]
```

```
w  
g  
c  
y  
z  
hi  
op
```

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

```
a  
a  
aa
```



```
d  
e  
e
```

```
>{tt,ind}=sort(t); tt
```

```
c  
g  
hi  
op  
w  
y  
z
```

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

Fungsi unik mengembalikan daftar yang diurutkan dari elemen unik suatu 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
```

```
>unique(t)
```

```
c  
g  
hi  
op  
w  
y  
z
```

Aljabar Linier

EMT memiliki banyak fungsi untuk memecahkan sistem linier, sistem renggang, atau masalah regresi.

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

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

```
-3.9999999999999999  
4.4999999999999999
```

```
>B=[5,6;7,8]; c=[8;9]; B\c
```

```
-5.0000000000000002  
5.5000000000000002
```

Untuk contoh lain, kita buat matriks 200x200 dan jumlah barisnya. Kemudian kita selesaikan $Ax=b$ menggunakan matriks invers. Kita ukur kesalahan sebagai deviasi maksimal 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 mempunyai solusi, penyesuaian linier meminimalkan norma kesalahan $Ax=b$.

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

1	2	3
4	5	6
7	8	9

Determinan matriks ini adalah 0.

```
>det(A)
```

0

Matriks Simbolik

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

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

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

```
>B &= [x,1,1,1;0,y,0,0;0,0,z,0;0,0,0,w]; $B
```

$$\begin{pmatrix} x & 1 & 1 & 1 \\ 0 & y & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & w \end{pmatrix}$$

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

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

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

$w\ x\ y\ z$

$w\ x\ y\ z$

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

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

```
>$&invert(B) with x=2, y=3
```

$$\begin{pmatrix} \frac{1}{2} & -\frac{1}{2y} & -\frac{1}{2z} & -\frac{1}{2w} \\ 0 & \frac{1}{y} & 0 & 0 \\ 0 & 0 & \frac{1}{z} & 0 \\ 0 & 0 & 0 & \frac{1}{w} \end{pmatrix}$$

3

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

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

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

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

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

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

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

```
>%eigenvalues([a,1;1,a])
```

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

Untuk mengekstrak vektor eigen tertentu dibutuhkan 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 ekspresi simbolik lainnya.

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

$$\begin{array}{cc} 1 & 4 \\ 5 & 2 \end{array}$$

Dalam ekspresi simbolik, gunakan dengan.

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

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

Akses terhadap baris matriks simbolik bekerja seperti halnya matriks numerik.

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

$$[1, a]$$

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

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

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

Terdapat fungsi simbolik di Maxima untuk membuat vektor dan matriks. Untuk ini, rujuk dokumentasi Maxima atau tutorial tentang Maxima di EMT.

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

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

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

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

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

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

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

Euler juga memiliki fungsi `xinv()` yang hebat, yang melakukan upaya lebih besar dan mendapatkan hasil yang lebih tepat.

Perlu dicatat, 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)
```

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

Misalnya nilai eigen A dapat dihitung secara numerik.

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

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

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

```
>$eigenvalues(A)
```

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

Nilai Numerik dalam Ekspresi Simbolik

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

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

$$\begin{pmatrix} 1 & 3.14159 \\ 4 & 5 \end{pmatrix}$$

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

```
>$&A
```

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

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

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

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

Maxima juga dapat melakukan komputasi dengan angka floating point, dan bahkan dengan angka floating point besar dengan 32 digit. Namun, evaluasinya jauh lebih lambat.

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

1.414213562373095048801688724209698078569671875376948073176679737990732478462107038850387534327641573_B × 10

1.414213562373095

```
>&bfloat(sqrt(7)), &float(sqrt(7))
```

2.645751311064590590501615753639260425710259183082450180368334459201068823230283627760392886474543611_B × 10

2.645751311064591

Ketepatan angka floating point besar dapat diubah.

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

3.14159265358979323846264338327950288419716939937510582097494\
4592307816406286208998628034825342117068b0

Variabel numerik dapat digunakan dalam ekspresi simbolik apa pun menggunakan "@var".

Perlu dicatat bahwa ini hanya diperlukan jika variabel telah didefinisikan dengan "==" atau "=" sebagai variabel numerik.

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

-5.424777960769379

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 (misalnya dalam dolar).

```
>K=5000
```

```
5000
```

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

```
>K*1.03
```

```
5150
```

Euler juga akan memahami sintaksis berikut.

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

```
5150
```

Namun lebih mudah menggunakan faktor

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

```
1.03  
5150
```

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

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

```
6719.58189672
```

Untuk keperluan kita, kita dapat mengatur format 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 antara dari tahun 1 hingga tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menulis loop, 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 menghasilkan vektor bilangan bulat.

```
>short 0:10
```

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

Maka semua operator dan fungsi di Euler dapat diaplikasikan ke 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 memperoleh vektor nilai.

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

Of course, the realistic way to compute these interest rates would be to round to the nearest cent after each year. Let us add a function for this.

```
>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 mengulanginya selama bertahun-tahun. Euler menyediakan banyak solusi untuk ini.

Cara termudah adalah fungsi iterate, 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 dapat mencetaknya dengan cara yang ramah, menggunakan format kami 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 vektor tertentu, kita menggunakan indeks dalam tanda kurung siku.

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

```
5150.00  
5000.00      5150.00      5304.50
```

Takjubnya, 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 ambil fungsi yang lebih maju, yang menambahkan nilai 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 menentukan nilai-nilai ini. Kita pilih $R=200$.

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

Real 1 x 11 matrix

5000.00	5350.00	5710.50	6081.82	...
---------	---------	---------	---------	-----

Bagaimana jika kita menghilangkan 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 uang berkurang. Jelas, jika kita hanya memperoleh bunga sebesar 150 pada tahun pertama, tetapi mengurangi 200, kita akan kehilangan uang setiap tahun.

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

```
>VKR=iterate("oneway",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 `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. Fungsi ini dapat mengambil kondisi akhir sebagai argumen. Kemudian, fungsi ini akan mengembalikan nilai dan jumlah iterasi..

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

```
-19.83  
47.00
```

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

Ini adalah pertanyaan yang hanya dapat dijawab secara numerik. Di bawah ini, kita akan memperoleh rumus yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada rumus yang mudah untuk tingkat bunga. Namun untuk saat ini, kita bertujuan untuk mencari solusi numerik.

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

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

Iterasinya sama seperti di atas

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

Namun, kita tidak lagi menggunakan nilai global R dalam ekspresi kita. Fungsi seperti `iterate()` memiliki trik khusus di Euler. Anda dapat meneruskan nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam kasus ini P dan R.

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

Mari kita coba uji coba.

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

```
-19.83
```

Sekarang kita bisa memecahkan masalah kita.

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

```
3.15
```

Rutin `solve` menyelesaikan ekspresi=0 untuk variabel x. Jawabannya adalah 3,15% per tahun. Kita ambil nilai awal 3% untuk algoritma tersebut. 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 suku bunga 3% per tahun.

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

-336.08

Perhatikan bahwa Anda tidak dapat memecahkan masalah jumlah tahun, karena fungsi kita mengasumsikan n sebagai nilai integer.

Solusi Simbolis untuk Masalah Suku Bunga

Kita dapat menggunakan bagian simbolis Euler untuk mempelajari masalah tersebut. Pertama, kita mendefinisikan fungsi `oneway()` secara simbolis.

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

$$R + q K$$

Sekarang kita dapat mengulanginya.

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

$$q \left(q \left(q \left(R + q K \right) + R \right) + R \right) + R$$

$$q^3 R + q^2 R + q R + R + q^4 K$$

Kita melihat suatu pola. Setelah n periode kita memiliki

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

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

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

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

Ini agak rumit. Jumlahnya dievaluasi dengan tanda "simpsum" untuk mereduksinya menjadi hasil bagi. Mari kita buat fungsi untuk ini.

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

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

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

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

```
-19.82504734650985  
-19.82504734652684
```

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

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

```
20.51
```

Jawaban ini menyatakan bahwa akan negatif setelah 21 tahun.

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

Asumsikan kita mendapatkan pinjaman sebesar K , dan membayar n kali cicilan sebesar R (dimulai setelah tahun pertama) sehingga menyisakan utang residual sebesar K_n (pada saat pembayaran terakhir). Rumus untuk ini jelas

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

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

Biasanya rumus ini diberikan dalam bentuk

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

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

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

Kita dapat mencari laju R secara simbolis.

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

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

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

Tentu saja, kita memiliki limit berikut.

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

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

Dengan jelas, tanpa bunga, kita harus membayar kembali 10 suku bunga sebesar 500.

Persamaan ini juga dapat diselesaikan untuk n . Akan terlihat lebih bagus jika kita menerapkan beberapa penyederhanaan.

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

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