

## 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)
```

Menjabarkan:

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

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

---

## 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

Perintah harus dipisahkan dengan spasi. 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 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 baris-baris tersebut.

Untuk melipat semua garis yang ada tekan Ctrl+L. Maka garis-garis berikutnya hanya akan terlihat jika salah satu dari garis-garis tersebut menjadi fokus. Untuk melipat satu baris ganda, mulailah baris pertama dengan "%+".

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

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

Euler mendukung perulangan dalam baris perintah, asalkan dapat dimasukkan ke dalam satu baris atau beberapa baris. Dalam program, pembatasan ini tentu saja tidak berlaku. Untuk informasi lebih lanjut, lihat pengantar berikut.

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

```
1.5  
1.41666666667  
1.41421568627  
1.41421356237  
1.41421356237
```

Tidak apa-apa menggunakan beberapa baris. Pastikan baris diakhiri dengan "...".

```
>x := 1.5; // komentar pergi ke sini sebelum ...  
>repeat xnew:=(x+2/x)/2; until xnew~=x; ...  
>  x := xnew; ...  
>end; ...  
>x,
```

```
1.41421356237
```

Struktur kondisional juga berfungsi.

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

```
Thought so!
```

Saat menjalankan perintah, kursor dapat berada di posisi mana pun di 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.

Saat menggerakkan kursor di sepanjang baris, pasangan tanda kurung buka dan tutup akan disorot. Perhatikan juga baris status. Setelah tanda kurung buka 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 ingin dicari. Pada baris kosong, bantuan untuk jendela bantuan akan ditampilkan. Anda dapat menekan escape untuk menghapus baris, atau untuk menutup jendela bantuan. Anda dapat mengklik dua kali pada perintah apa pun untuk membuka bantuan untuk perintah ini. Coba klik dua kali perintah `expand` di bawah 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 apa pun. Selain itu, Anda dapat menyalin tanda kurung yang disorot.

Euler mengetahui fungsi matematika yang umum. 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 memungkinkan.

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

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Tanda titik koma menghilangkan keluaran perintah. Di akhir baris perintah, tanda ";" 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 menetapkan 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 rumit seperti

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

Anda perlu memasukkannya dalam formulir 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 dengan hati-hati. EMT membantu Anda dengan menyorot ekspresi yang diakhiri tanda kurung tutup. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi.

Hasil perhitungan ini adalah angka floating point. Angka ini dicetak dengan akurasi sekitar 12 digit secara default.

Pada baris perintah berikut, kita juga mempelajari cara 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 terdiri dari operator dan fungsi. Jika perlu, ekspresi harus berisi tanda kurung untuk memaksakan urutan eksekusi yang benar. Jika ragu, sebaiknya gunakan tanda kurung. Perhatikan bahwa EMT menampilkan tanda kurung buka dan tutup saat mengedit baris perintah

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

```
14.4978445072
```

Operator numerik Euler meliputi

```
+ unary or operator plus
- unary or operator minus
*, /
. the matrix product
a^b power for positive a or integer b (a**b works too)
n! the factorial operator
```

dan masih banyak lagi.

Berikut ini 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 (kurung bundar), jika ada keraguan tentang urutan eksekusi! Berikut ini tidak sama dengan  $(2^3)^4$ , yang merupakan default untuk  $2^3^4$  dalam EMT (beberapa sistem numerik melakukannya dengan cara lain).

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

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

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

```
>longest 1/3
```

0.3333333333333333

Representasi ganda internal membutuhkan 8 byte.

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

[illegible]

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

$$5.55555555555554 \times 16^{-1}$$

## String

Suatu string dalam Euler didefinisikan dengan "...".

```
>"Suatu string dapat berisi apa saja."
```

```
Suatu string dapat berisi apa saja.
```

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

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

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

Fungsi cetak juga mengonversi angka menjadi string. Fungsi ini dapat memuat sejumlah digit dan sejumlah tempat (0 untuk keluaran padat), dan optimalnya satu unit.

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

```
Golden Ratio : 1.61803
```

Ada string khusus none, yang tidak dicetak. String ini dikembalikan oleh beberapa fungsi, ketika hasilnya tidak penting. (Dikembalikan secara otomatis, jika fungsi tersebut tidak memiliki pernyataan return.)

```
>none
```

Untuk mengubah 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 dirangkai.

```
>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 dirangkai seperti string lainnya.

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

```
= 45°
```

I

Dalam komentar, entitas yang sama seperti alpha, beta, dll. dapat digunakan. Ini mungkin merupakan alternatif cepat untuk Lateks. (Rincian lebih lanjut pada 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 vektor angka Unicode. Fungsi kebalikannya adalah `chartoutf()`.

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

```
Ü is a German letter
```

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

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

```
We have =.
```

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

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

```
>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. Dalam contoh ini, kami menggunakan kondisional `isprime(n)`.

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

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

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

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

## Format Keluaran

---

Format keluaran default EMT mencetak 12 digit. Untuk memastikan bahwa kita melihat default, kita mengatur ulang formatnya.

```
>defformat; pi
```

```
3.14159265359
```

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

```
>longest pi
```

```
3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari angka ganda.

```
>printhex(pi)
```

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

Format keluaran dapat diubah secara permanen dengan perintah format.

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

```
0.33333  
3.14159  
0.84147
```

Format defaultnya adalah (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). Namun, ini dapat diubah.

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

3.1416

Fungsi "longestformat" juga mengatur format skalar.

```
>longestformat; pi
```

3.141592653589793

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

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

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

Namun format keluaran EMT dapat diatur secara 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 pendek untuk mencetak hasil dalam format pecahan. Kami telah menggunakannya di atas.

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

25/12

Karena format internal menggunakan cara biner untuk menyimpan angka, nilai 0,1 tidak akan terwakili secara tepat. Kesalahannya bertambah sedikit, seperti yang Anda lihat dalam perhitungan berikut.

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

-1.110223024625157e-16

Namun dengan "longformat" default, Anda tidak akan melihat hal ini. Demi 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 ingin menggunakan string sebagai ekspresi, gunakan konvensi untuk menamainya "fx" atau "fxy", dst. 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 menggunakan parameter yang ditetapkan.

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

```
-0.919535764538
```



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

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

36

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 mencatat 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 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

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

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

Bentuk khusus dari suatu ekspresi memperbolehkan variabel apa pun sebagai parameter tanpa nama untuk evaluasi ekspresi, bukan hanya "x", "y", dst. Untuk ini, awali 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 memerlukan ekspresi dalam "x".

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

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

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

-0.475

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

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

-0.425

Suatu ekspresi tidak harus bersifat simbolis. Hal ini diperlukan jika ekspresi tersebut mengandung fungsi-fungsi yang hanya diketahui dalam kernel numerik, bukan dalam Maxima.

## Matematika Simbolik

---

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

Matematika simbolik terintegrasi dengan lancar ke dalam Euler dengan &. Ekspresi apa pun yang dimulai dengan & adalah ekspresi simbolik. Ekspresi ini dievaluasi dan dicetak oleh Maxima.

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

```
>$&44!
```

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)
```

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

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

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik dua kali pada fungsi tersebut. Misalnya, coba klik dua kali pada "&binomial" pada baris perintah sebelumnya. Ini akan membuka dokumentasi Maxima sebagaimana disediakan oleh penulis program tersebut.

Anda akan mempelajari bahwa hal berikut juga berfungsi

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

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

Jika Anda ingin mengganti x dengan nilai tertentu gunakan "with".

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

Dengan cara itu Anda dapat menggunakan solusi suatu persamaan dalam persamaan lainnya.

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

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

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

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

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

Untuk kelengkapan, kami mencatat bahwa ekspresi simbolik dapat digunakan dalam program, tetapi harus disertakan dalam tanda kutip. Selain itu, akan jauh lebih efektif untuk memanggil Maxima pada waktu kompilasi jika memungkinkan.

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

Sekali lagi, % mengacu pada hasil sebelumnya.

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

Input langsung perintah Maxima juga tersedia. Awali 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 ahli dalam Maxima, Anda mungkin ingin menggunakan sintaksis asli Maxima. Anda dapat melakukannya dengan "::".

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

$$g^2$$

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

$$x^3 E$$

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

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

$$\frac{5}{125} E$$

18551.64488782208

```
>fx(5)
```

18551.6448878

Untuk mengevaluasi suatu ekspresi dengan nilai variabel tertentu, Anda dapat menggunakan operator "with".

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

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

$$\frac{10}{1000} E - \frac{5}{125} E$$

2.20079141499189e+7

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

Untuk mendapatkan kode Latex untuk suatu 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, ini tidak berfungsi, karena Maxima tidak mendukungnya. Sebagai gantinya, gunakan sintaks "with" (bentuk yang lebih baik dari perintah `at(...)` dari Maxima).

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

Penugasan tersebut juga dapat bersifat simbolis.

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

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

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

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

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

```
1.56155281281
```

Nilai numerik dari solusi simbolik dapat dihitung dengan mengevaluasi hasil simbolik. Euler akan membaca ulang penugasan  $x = \text{dst}$ . Jika Anda tidak memerlukan hasil numerik untuk perhitungan lebih lanjut, Anda juga dapat membiarkan Maxima menemukan nilai numeriknya.

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

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

Untuk mendapatkan solusi simbolis yang spesifik, seseorang dapat menggunakan "with" dan indeks.

```
>$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]
```

Ekspresi simbolik dapat memiliki tanda, yang menunjukkan perlakuan khusus di Maxima. Beberapa tanda dapat digunakan sebagai perintah juga, yang lainnya tidak. Tanda ditambahkan dengan "|" (bentuk yang lebih baik dari "ev(...,flags)")

```
>$ diff((x^3-1)/(x+1),x) //turunan bentuk pecahan  
>$ diff((x^3-1)/(x+1),x) | ratsimp //menyederhanakan pecahan  
>$factor(%)
```

## 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°) // redine sine in degrees  
>sin(45)
```

0.707106781187

Sebaiknya kita hilangkan definisi ulang tentang dosa ini.

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

0.707106781187

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

16

Menetapkannya akan menimpa nilai default.

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

80



Parameter yang ditetapkan juga 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)
```

Fungsi simbolis dapat digunakan dalam ekspresi simbolis.

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

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  
>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; $$P(x,n)  
>function Q(x,n) &= (x+2)^n; $$Q(x,n)  
>$$P(x,4), $$expand(%)  
>P(3,4)
```

625

```

>$P(x,4)+ Q(x,3), $expand(%)
>$P(x,4)-Q(x,3), $expand(%), $factor(%)
>$P(x,4)*Q(x,3), $expand(%), $factor(%)
>$P(x,4)/Q(x,1), $expand(%), $factor(%)
>function f(x) &= x^3-x; $f(x)

```

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

```

>$integrate(f(x),x)

```

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

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

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)
```

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

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

17

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

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

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

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

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

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

Untuk meringkas

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

## 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.

```
>$&solve(x^2=2,x)
>$&solve(x^2-2,x)
>$&solve(a*x^2+b*x+c=0,x)
>$&solve([a*x+b*y=c,d*x+e*y=f],[x,y])
>px &= 4*x^8+x^7-x^4-x; $&px
```

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

ami 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
```

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])
```

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])
```

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

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

dengan  $a = 3$ .

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

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

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

2.54116291558

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

```
>$fourier_elim([x^2 - 1>0],[x]) // x^2-1 > 0
>$fourier_elim([x^2 - 1<0],[x]) // x^2-1 < 0
>$fourier_elim([x^2 - 1 # 0],[x]) // x^2-1 <> 0
>$fourier_elim([x # 6],[x])
>$fourier_elim([x < 1, x > 1],[x]) // tidak memiliki penyelesaian
>$fourier_elim([minf < x, x < inf],[x]) // solusinya R
>$fourier_elim([x^3 - 1 > 0],[x])
>$fourier_elim([cos(x) < 1/2],[x]) // ??? gagal
>$fourier_elim([y-x < 5, x - y < 7, 10 < y],[x,y]) // sistem pertidaksamaan
>$fourier_elim([y-x < 5, x - y < 7, 10 < y],[y,x])
>$fourier_elim((x + y < 5) and (x - y > 8),[x,y])
>$fourier_elim(((x + y < 5) and x < 1) or (x - y > 8),[x,y])
>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])
```

## 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

Hasil kali matriks dilambangkan dengan sebuah titik.

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

3
4

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

[3, 4]
--------

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

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

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

$$\begin{pmatrix} 11 \\ 25 \end{pmatrix}$$

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

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

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

```
>A.A
```

$$\begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$$

```
>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 hasil kali matriks, tetapi perkalian elemen demi elemen. Hal yang sama berlaku untuk vektor.

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

9
16

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

```
>2*A
```

2	4
6	8

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

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

1	4
3	8

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

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

2	6
6	12

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

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

1	2
1	2

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

1	4
3	8

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

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

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

Sekali lagi, 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

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, ada `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))
```

Mari kita uji.

```
>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
```

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

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

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

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

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

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

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

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

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

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

1	2	3
4	5	6
7	8	9

Sekarang kita bisa mengekstrak diagonal.

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

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

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

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

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

## Vektorisasi

---

Hampir semua fungsi di Euler juga dapat digunakan untuk input matriks dan vektor, jika hal ini masuk akal.

Sebagai contoh, fungsi `sqrt()` menghitung akar kuadrat dari semua elemen vektor atau matriks.

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

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

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

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

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

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

```
s = t^3-t
```

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

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

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

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

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

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

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

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

Secara default, vektor baris dicetak dalam format ringkas.

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

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

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

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

```
5  
25
```

Untuk mentransposisikan matriks, kita menggunakan apostrof.

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

```
1  
2  
3  
4
```

Jadi kita dapat menghitung matriks A dikali vektor b.

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

```
30  
70
```

Perhatikan bahwa v masih merupakan vektor baris. Jadi  $v'.v$  berbeda dengan  $v.v'$ .

```
>v'.v
```

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

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

```
>v.v'
```



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

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

30

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ini adalah ringkasan aturannya.

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

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

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

[1, 4, 9]

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

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

1	2	3
2	4	6
3	6	9

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

```
>v.v'
```

14

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

sum,prod menghitung jumlah dan hasil kali dari baris-baris

cumsum, cumprod melakukan hal yang sama secara 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, opsional dengan ukuran langkah.

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

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

Untuk menggabungkan matriks dan vektor, terdapat operator “|” dan “\_”.

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

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

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

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

6

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

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

6  
[7, 8, 9]

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

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

```
[2, 4]
```

```
2  
5  
8
```

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

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

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

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

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

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

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

Sekarang kita menambahkan kolom ke matriks.

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

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

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

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

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

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

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

Kata kunci “map” membuat vektor fungsi. Fungsi ini sekarang akan bekerja 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
5	7	8	9

Kita dapat mengakses baris lengkap dari sebuah matriks.

```
>A[2]
```

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

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-baris yang sesuai dari matriks. Di sini kita menginginkan baris pertama dan kedua dari A.

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

1	2	3
4	5	6

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

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

7	8	9
4	5	6
1	2	3

Trik indeks juga dapat digunakan pada kolom.

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

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

2	3
5	6
8	9

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

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

3
6
9

Sebagai alternatif, biarkan indeks pertama kosong.

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

2	3
5	6
8	9

Kita juga bisa mendapatkan baris terakhir A.

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

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

Sekarang mari kita ubah elemen-elemen dari A dengan memberikan sebuah submatriks dari A kesuatu nilai.

Hal ini sebenarnya mengubah matriks A yang tersimpan.

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

4	2	3
4	5	6
7	8	9

Kita juga dapat menetapkan nilai pada deretan A.

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

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

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

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

5	6	-1
7	8	6
7	8	9

Selain itu, beberapa jalan pintas diperbolehkan.

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

0	0	-1
0	0	6
7	8	9

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

```
>A[4]
```

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

## Menyortir dan Mengacak

---

Fungsi mengurutkan () mengurutkan vektor baris.

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

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

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

Mari kita acak sebuah vektor.

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

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

Indeks berisi urutan v yang tepat.

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

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

Hal ini juga berlaku untuk vektor string.

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

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

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

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

Seperti yang anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

Fungsi unique mengembalikan daftar terurut dari elemen unik vektor.

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

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

Hal ini juga berlaku untuk vektor string.

```
>unique(s)
```

```
a  
aa  
d  
e
```



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

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

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

```
-4  
4.5
```

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

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

```
8.790745908981989e-13
```

Jika sistem tidak memiliki solusi, pencocokan 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 dari matris ini adalah 0.

```
>det(A)
```

0

## Matriks Simbolik

---

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

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

Untuk mengekstrak vektor eigen tertentu, diperlukan pengindeksan yang cermat.

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

[1, - 1]

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

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

1	4
5	2

Dalam ekspresi simbolik, gunakan dengan.

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

Akses ke baris matriks simbolik bekerja seperti halnya matriks numerik.

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

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

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

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

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

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

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

```
>$\text{invert}(B)()
```

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

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

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

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

$$\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}$$

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 simbolik. Lihat tutorial tentang Maxima untuk mengetahui detailnya.

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

## Nilai Numerik dalam Ekspresi simbolik

---

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

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

1	3.14159
4	5

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

```
>$&A
```

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

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



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

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

Ketepatan angka floating point yang besar dapat diubah.

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

```
3.14159265358979323846264338327950288419716939937510582097494\  
4592307816406286208998628034825342117068b0
```

Variabel numerik dapat digunakan dalam ekspresi simbolik apapun dengan menggunakan "@var".

Perhatikan bahwa hal ini hanya diperlukan, jika variabel telah didefinisikan dengan ":"=" atau "=" sebuah variabel numerik.

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

## Demo - Suku Bunga

---

Di bawah ini, kami menggunakan Euler Math Toolbox (EMT) untuk menghitung suku bunga. Kami melakukan secara numerik dan simbolik untuk menunjukkan kepada anda bagaimana euler dapat digunakan untuk memecahkan masalah di kehidupan nyata.

Asumsikan anda memiliki modal awal sebesar modal awal sebesar 5000(katakan dalam dolar).

```
>K=5000
```

```
5000
```

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

```
>K*1.03
```

```
5150
```

Euler juga akan memahami sintaks berikut ini.

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

```
5150
```

Tetapi akan lebih mudah untuk menggunakan faktor.

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

```
1.03  
5150
```

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

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

```
6719.58189672
```

Untuk tujuan kita, kita bisa menetapkan format ke 2 digit setelah titik desimal.

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

```
6719.58
```

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

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

Starting from 5000\$ you get 6719.58\$.

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

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

Real 1 x 11 matrix

5000.00	5150.00	5304.50	5463.64	...
---------	---------	---------	---------	-----

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

```
>short 0:10
```

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

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

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

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

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

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

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

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

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

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

```
1271.61  
1271.6071
```

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

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

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

```
Real 1 x 11 matrix
```

```
5000.00    5150.00    5304.50    5463.64    ...
```

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

```
>VKr'
```

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

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

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

```
5150.00  
5000.00      5150.00      5304.50
```

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

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

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

```
6719.60
```

```
6719.58
```

Perbedaannya sangat kecil.



## Menyelesaikan Persamaan

---

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

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

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

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

Real 1 x 11 matrix

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

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

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

Real 1 x 11 matrix

5000.00	4950.00	4898.50	4845.45	...
---------	---------	---------	---------	-----

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

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

```
>VKR=iterate("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 karena `nonzeros(VKR<0)` mengembalikan vektor dengan indeks `i`, di mana `VKR[i]<0`, dan `min` menghitung indeks minimal.

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

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

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

```
-19.83  
47.00
```

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

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

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

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

Perulangannya sama seperti di atas

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

Tetapi kita tidak lagi menggunakan nilai global R dalam ekspresi kita. Fungsi-fungsi seperti `iterate()` memiliki trik khusus dalam Euler. Anda bisa mengoper nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R.

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

Mari kita coba sebuah tes.

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

-19.83

Sekarang kita bisa menyelesaikan masalah kita.

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

3.15

Rutin penyelesaian menyelesaikan ekspresi  $= 0$  untuk variabel  $x$ . Jawabannya adalah 3,15% per tahun. Kita mengambil nilai awal 3% untuk algoritma ini. Fungsi `solve()` selalu membutuhkan nilai awal.

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

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

-336.08

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

## Solusi Simbolik untuk Masalah Suku Bunga

---

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

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

Sekarang kita bisa mengulangi hal ini.

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

Kita melihat sebuah 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 geometris, yang dikenal dengan Maxima.

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

Ini sedikit rumit. Penjumlahan dievaluasi dengan flag “simpsum” untuk mengurangnya menjadi hasil bagi.

Mari kita buat sebuah fungsi untuk ini.

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

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

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

```
-19.82504734650985
```

```
-19.82504734652684
```

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

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

20.51

Jawaban ini mengatakan bahwa nilai tersebut akan menjadi negatif setelah 21 tahun.

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

Asumsikan kita mendapatkan pinjaman sebesar  $K$ , dan membayar  $n$  kali pembayaran sebesar  $R$  (dimulai setelah tahun pertama) sehingga menyisakan sisa utang sebesar  $Kn$  (pada saat pembayaran terakhir). Rumus untuk hal ini adalah sebagai berikut

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

Biasanya rumus ini diberikan dalam bentuk

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

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

Kita dapat menyelesaikan laju R secara simbolis.

```
>%solve(equ,R)
```

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

Tentu saja, kita memiliki batas berikut.

```
>%limit(R(5000,0,x,10),x,0)
```

Jelasnya, tanpa bunga kita harus membayar kembali 10 suku bunga 500.

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

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



## Contoh Penyelesaian Soal-Soal Aljabar

---

### R.2 Exercise Set

---

Write an equivalent expression without negative exponents

$$49. \left( \frac{24a^{10}b^{-8}c^7}{12a^6b^{-3}c^5} \right)^{-5}$$

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

Calculate.

$$90. (2^6 \times 2^{-3} \div 2^{10} \div 2^{-8})$$

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

Syntesis

Saving plan. The formula

$$S = P \left( \frac{(1 + \frac{r}{12})^{12t} - 1}{\frac{r}{12}} \right)$$

gives the amount  $S$  accumulated in a savings plan when a deposit of  $P$  dollars is made each month for  $t$  years in an account with interest rate  $r$ , compounded monthly.

```
>function savingPlan(deposit, years, interestRate) ...
```

```
    return deposit * ((1 + (years / 12) ^ (12 * years)) - 1 / (interestRate / 12))
endfunction
```

97. James deposits \$250 in a retirement account each month beginning at age 40. If the investment earns 5% interest, compounded monthly, how much will have accumulated in the account when he retires 27 years later?

```
>result := savingPlan(250, 27, 0.05)
```

```
31994540460589789358826662710200054727006612566029477411600158184622621262418936313231860997654656
```

```
>round(result)
```

```
round(result)
```

98. Kayla deposits \$100 in a retirement account each month beginning at age 25. If the investment earns 4% interest, compounded monthly, how much will have accumulated in the account when she retires at age 65?

```
>savingPlan(100, 65-25, 0.04)
```

```
95895391064934425636904631683687745180441308685905596046473504061898973164769632734224476061086759
```

100. Lamont wants to have \$200,000 accumulated in a retirement account by age 70. If he starts making monthly deposits to the plan at age 30 and can count on an interest rate of 4.5%, compounded monthly, how much should he deposit each month in order to accomplish this?

```
>savingPlan(200000, 70 - 30, 0.045)
```

```
19179078212986883147814141575530811418632261366807710681944573874844559515314888679396340235107830
```

Simplify. Assume that all exponents are integers, all denominators are nonzero, and zero is not raised to a nonpositive power

$$104. ((m^{x-b}n^{x+b})^x(m^bn^{-b})^x)$$

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

### R.3 Exercise Set

---

Determine the terms and the degree of the polynomials

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

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

$$8. (2x^2 + 12xy - 11) + (6x^2 - 2x + 4) + (-x^2 - y - 2)$$

```
>$&factor((2*x^2 + 12*x*y - 11) + (6*x^2 - 2*x + 4) + (-x^2 - y - 2))
>
```

$$9. (3x^2 - 2x - x^3 + 2) - (5x^2 - 8x - x^3 + 4)$$

```
>$&factor((3*x^2 - 2*x - x^3 + 2) - (5*x^2 - 8*x - x^3 + 4))
>
```

$$31. (5x - 3)^2$$

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

Synthesis

Multiply. Assume that all exponents are natural numbers.

$$55. (a + b + c)^2$$

```
>$&expand((a + b + c)^2)
```

Factor the difference of squares

$$47. z^2 - 81$$

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

Factor the square of a binomial.

$$66. (5a^2 - 10ab + 5b^2)$$

```
>$&factor(5*a^2 - 10*a*b + 5*b^2)
```

Factor the sum or difference of cubes.

$$75. (t^6 + 1)$$

```
>$&factor(t^6 + 1)
```

Factor completely.

$$78. (4x^2 + 12xy^2)$$

```
>$&factor(4*x^2 + 12*x*y^2)
```

Synthesis

Factor.

$$130. (x + 0.01)^2 - x^2$$

```
>$&factor((x + 0.01)^2 - x^2)
```

Factor. Assume that variables in exponents represent natural numbers.

$$139. (y - 1)^4 - (y - 1)^2$$

```
>$&factor((y - 1)^4 - (y - 1)^2)
```

## R.5 Exercise Set

---

$$31. 7(3x + 6) = 11 - (x + 2)$$

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

$$36. y^2 - 4y - 45 = 0$$

```
> solve(y^2 - 4*y - 45 = 0, y)
```

$$65. A = \frac{1}{2}h(b_1 + b_2)$$

for:  $h$  (Area of a trapezoid)

```
> solve(A = (1/2)* h * (b1 + b2), h)
```

$$71. Ax + By = C$$



for A

```
>$&solve(A*x + B*y = C, A)
```

Synthesis  
Solve

$$81.3[5 - 3(4 - t)] - 2 = 5[3(5t - 4) + 8] - 26$$

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

$$87.3x^3 + 6x^2 - 27x - 54 = 0$$

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

$$11). \frac{x^3 - 6x^2 + 9x}{x^3 - 3x^2}$$

Penyelesaian :  
mencari faktor pembilang

```
>$&factor(x^3-6*x^2+9*x)
```

mencari faktor penyebut

```
>$&factor(x^3-3*x^2)
```

jadi tugas kita menyelesaikan :

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

```
>$&((x-3)^2 * x)/((x-3) * x^2)
```

atau dengan perintah EMT untuk menyelesaikan soal ini adalah

```
>$ratsimp((x^3-6*x^2+9*x)/(x^3-3*x^2)) ...  
>//menyederhanakan pecahan dengan perintah ratsimp
```

17. Multiply or divide and, if possible, simplify

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

Penyelesaian:

```
>$ratsimp((r-s)/(r+s))*((r^2-s^2)/((r-s)^2)) ...  
>//menyederhanakan pecahan  
>$((r^2-s^2)/(r-s)*(s+r), $ratsimp(%)) ...  
>//menyederhanakan pecahan lalu hasilnya disederhanakan lagi pakai %
```

$$28. \frac{(c^3+8)}{(c^2-4)} : \frac{(c^2-2c+4)}{(c^2-4c+4)}$$

Penyelesaian:

```
>$&ratsimp((c^3+8)/(c^2-4))/((c^2-2*c+4)/(c^2-4*c+4)), $&ratsimp(%) ...  
>//menyederhanakan pecahan lalu hasilnya disederhanakan lagi pakai %
```

40. add or subtract and, if possible, simplify

$$\frac{6}{y^2 + 6y + 9} - \frac{5}{y - 3}$$

Penyelesaian:

```
>$&expand(6/y^2+6*y+9)-(5/y-3), $&ratsimp(%) ...  
>//menyederhanakan pecahan lalu melakukan operasi pada hasilnya
```

62. Simplify

$$\frac{\frac{a^2}{b} + \frac{b^2}{a}}{a^2 - ab + b^2}$$

Penyelesaian:

```
>$&factor((a^2/b)+(b^2/a))/(a^2-a*b+b^2) ...  
>//menyederhanakan bentuk paling sederhana dengan perintah factor
```

## R.7 Exercise Set

---

53. Simplify. Assume that no radicands were formed by raising negative quantities to even powers.

$$(2\sqrt{3} + \sqrt{5}) \cdot (\sqrt{3} - 3\sqrt{5})$$

Penyelesaian:

```
>$&expand(2*(3)^1/2 + (5)^1/2)*((3)^1/2 - 3*(5)^1/2)
```

## Review Exercise

---

66. the cost of a house is \$98,000. The down payment is \$16,000, the interest rate is 6 1/2%, and the loan period is 25 years. What is the monthly mortgage payment?

Penyelesaian :

Rumus angsuran per bulan:

$$M = P \cdot \frac{\frac{r}{12} \cdot (1 + \frac{r}{12})^2}{(1 + \frac{r}{12})^2 - 1}$$

dengan M : angsuran per bulan

P : jumlah pinjaman

r : suku bunga bulanan

n : jumlah total pembayaran (dalam bulan)

diketahui:

Harga rumah : \$98,000

Uang muka : \$16,000

suku bunga tahunan: 6 1/2%

6 1/2% diubah menjadi 6.5%

lama angsur : 25 tahun

ditanya: angsuran per bulan berapa

Variabel yang dibutuhkan pada rumus adalah :

$$>P = 98000 - 16000$$

$$82000$$

$$>r = 6.5/100$$

0.065

```
>n = 25*12
```

300

Substitusi semua variabel yang telah diperoleh ke rumus M

```
>M = P * (r/12 * (1 + r/12)^n)/((1+r/12)^n -1 )
```

553.669872305

Jadi, biaya angsuran per bulan selama 25 tahun dengan suku bunga sebesar 6.5% persen per tahun adalah \$553,669872305

$$70.(x^n + 10)(x^n - 4)$$

Penyelesaian:

```
>$\expand((x^n + 10)*(x^n - 4)) ...  
>//menjabarkan fungsi
```

$$71.(t^a + t^{-a})^2$$

Penyelesaian:

```
>$&expand((t^a + t^(-a))^2) ...  
>//menjabarkan fungsi
```

$$73.(a^n - b^n)^3$$

```
>$&showev('expand((a^n - b^n)^3)), $&expand((a^n - b^n)^3) ...  
>//menjabarkan fungsi dengan showev
```

## 2.3 Exercise Set

---

Given that

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

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

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



Find each of the following

Menyimpan semua nilai fx, gx, dan hx.

```
>fx <= 3*x + 1; gx <= x^2 - 2*x - 6; hx <= x^3;
```

1. Mencari nilai

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

Penyelesaian:

fungsi komposisi di atas juga dapat ditulis sebagai :

$$(f(g(-1)))$$

Mencari nilai  $g(-1)$

```
>gx(-1) //cara manual
```

Lalu mencari nilai  $f(-3)$

```
>fx(-3)
```

-8

Perintah fungsi komposisi di EMT

Akan menghasilkan hasil yang sama dengan cara manual di atas. Cara ini lebih efisien.

```
>fx(gx(-1)) //cara cepat
```

-8

4. Mencari nilai

$$(g \circ h)\left(\frac{1}{2}\right)$$

Penyelesaian:

Fungsi komposisi di atas juga dapat ditulis

$$g\left(h\left(\frac{1}{2}\right)\right)$$

```
>gx(hx(1/2))
```

-6.234375

7. Mencari nilai

$$(f \circ h)(-3)$$

Penyelesaian:

Fungsi komposisi di atas bisa ditulis

$$f(g(-3))$$

```
>fx(gx(-3))
```

28

11. Mencari nilai

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

Penyelesaian

Fungsi komposisi di atas dapat ditulis

$$(h(h(2)))$$

```
>hx(hx(2))
```

512

13. Mencari nilai

$$(f \circ f)(-4)$$

Penyelesaian:

Fungsi komposisi di atas dapat ditulis

$$f(f(-4))$$

```
>fx(fx(-4))
```

-32

### 3.1 Exercise Set

---

11. Simplify. Where answer in the form  $a + bi$ , where  $a$  and  $b$  are real numbers.

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

Penyelesaian:

```
>bilangan1 = -5 + 3*I, bilangan2 = 7 + 8*I, bilangan1 + bilangan2
```

-5+3i

7+8i

2+11i

$$21.(10 + 7i) - (5 + 3i)$$

Penyelesaian:

```
>bilangan1 = 10 + 7*I, bilangan2 = 5 + 3*I, bilangan1 - bilangan2
```

10+7i

5+3i

5+4i

$$35.7i \cdot (2 - 5i)$$

Penyelesaian:

```
>bilangan1 = 7*I, bilangan2 = 2-5*I, bilangan1*bilangan2
```

0+7i

2-5i

35+14i

$$36.3i \cdot (6 + 4i)$$

Penyelesaian:

```
>bilangan1 = 3*I, bilangan2 = 6 + 4*I, bilangan1*bilangan2
```

```
0+3i  
6+4i  
-12+18i
```

$$37. - 2i \cdot (-8 + 3i)$$

Penyelesaian:

```
>bilangan1 = -2*I, bilangan2 = -8 + 3*I, bilangan1*bilangan2
```

```
0-2i  
-8+3i  
6+16i
```

---

### 3.4 Exercise Set

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

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

$$3). \frac{x+2}{4} - \frac{x-1}{5} = 15$$

```
>$&solve(((x+2)/4)-((x-1)/5)=15)
```

$$4). \frac{t+1}{3} - \frac{t-1}{2} = 1$$

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

$$7). \frac{5}{3x+2} = \frac{3}{2x}$$

```
>$&solve((5/(3*x+2))=(3/2*x))
```



29).  $\sqrt{3x-4} = 1$

```
>$&solve((3*x-4)^1/2 = 1)
```

### 3.5 Exercise Set

---

23).  $|x+3| - 2 = 8$

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

$$28). |5x + 4| + 2 = 5$$

```
>$&solve(abs(5*x+4)+2=5, x)
>$&solve((5*x+4)=3)
>$&solve((5*x+4)=-3)
```

$$32). 5 - |4x + 3| = 2$$

```
>$&solve(5-(abs(4*x+3)=2))
>$&solve((4*x+3)=2)
>$&solve((4*x+3)=-2)
```

$$43). |2x| \geq 6$$

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

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

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

## Chapter 3 test

---

$$7).x + 5\sqrt{x} - 36 = 0$$

```
>$solve(x+5*(x)^1/2-36=0)
```

$$10).\sqrt{x+4} - \sqrt{x-4} = 2$$

```
>$solve(((x+4)^1/2)-((x-4)^1/2=2))
```

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

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

13).  $|x + 3| \leq 4$

```
>$&fourier_elim([abs(x+3)<=4],[x])
```

14).  $|2x - 1| < 5$

```
>$&fourier_elim([abs(2*x-1)<5],[x])
```

## 4.1 Exercise Set

---

$$36).f(x) = (x^2 - 5x + 6)^2$$

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

$$37).f(x) = x^4 - 4x^2 + 3$$

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

$$40).f(x) = x^3 - x^2 - 2x + 2$$

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

$$61).2y - 3 \geq 1 - y + 5$$

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

$$64). \left| x + \frac{1}{4} \right| \leq \frac{2}{3}$$

```
>$fourier_elim([x+(1/4)<=(2/3)], [x])
```

## Mid-Chapter Mixed Review

---

$$18). g(x) = x^3 - 9x^2 + 4x - 10;$$

find  $g(-5)$

```
>function g(x) := (x^3-9x^2+4x-10)
>g(-5)
```

-380

$$20). f(x) = 5x^4 + x^3 - x;$$

find

$$f(-\sqrt{-2})$$

```
>a = - sqrt(2)
```

-1.41421356237

```
>fx &= 5*x^4 + x^3 - x; fx(a)
```

18.5857864376

$$17).\frac{x^5-5}{x+1}$$

```
>$\&expand((x^5-5)/(x+1))
```

$$19).f(x) = 20x^2 - 40x;$$

find

$$f\left(\frac{1}{2}\right)$$

```
>function f(x) :=(20*x^2-40*x)
>f(1/2)
```

-15

$$23).h(x) = x^3 - 2x^2 - 55x + 56$$

Penyelesaian :

```
>$&factor(x^3-2*x^2-55*x+56)
>$&solve(x^3-2*x^2-55*x+56)
```

$$24).g(x) = x^4 - 2x^3 - 13x^2 + 14x + 24$$



Penyelesaian :

```
>$&factor(x^4-2*x^3-13*x^2+14*x+24)
>$&solve(x^4-2*x^3-13*x^2+14*x+24)
```

### 4.3 Exercise Set

---

$$f(x) = x^2 + 4x^2 + x - 6$$

penyelesaian :

```
>$&solve(x^3+4*x^2+x-6)
```

$$50).f(x) = x^4 + x^3 - 3x^2 - 5x - 2$$

Penyelesaian

```
>$&solve(x^4+x^3-3*x^2-5*x-2)
```

$$51).f(x) = x^3 - 7x + 6$$

Penyelesaian

```
>$&solve(x^3-7*x+6)
```

$$52).f(x) = x^3 - 12x + 16$$

Penyelesaian :

```
>$&solve(x^3-12*x+16)
```

$$53).f(x) = -x^3 + 3x^2 + 6x - 8$$

```
>$&solve(-x^3+3*x^2+6*x-8)
```