

Farah Nisrina Qurratu'aini (24030130022)

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:

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

The Command Line (Baris Perintah)

Baris perintah di EMT sangat penting karena menjadi cara utama untuk berinteraksi dengan program. Setiap baris perintah dapat berisi satu atau lebih instruksi, dan tanda seperti koma atau titik koma digunakan untuk mengatur apakah hasil ditampilkan atau tidak. Ini membantu pengguna bekerja lebih efisien saat menulis banyak perintah.

A command line of Euler consists of one or several Euler commands followed by a semicolon ";" or a comma ",". The semicolon prevents the printing of the result. The comma after the last command can be omitted.

The following command line will only print the result of the expression, not the assignments or the format commands.

Artinya :

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

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

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

```
16.7551608191
```

Commands must be separated with a blank. The following command line prints its two results.

Artinya:

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

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

```
50.2654824574
100.530964915
```

Command lines are executed in the order the user presses return. So you get a new value each time you execute the second line.

Artinya:

Perintah baris perintah dieksekusi sesuai urutan pengguna menekan tombol Enter. Jadi, Anda akan mendapatkan nilai baru setiap kali Anda mengeksekusi baris kedua.

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

```
0.540302305868
```

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

```
0.857553215846
```

If two lines are connected with "...", both lines will always execute simultaneously.

Artinya:

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

```

This is also a good way to spread a long command over two or more lines. You can press Ctrl+Return to split a line in two at the current cursor position, or Ctrl+Back to join the lines.

To fold all multi-lines press Ctrl+L. Then the subsequent lines will only be visible, if one of them has the focus. To fold a single multi-line start the first line with "%+ ".

Artinya:

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

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

```

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

```

A line starting with %% will be completely invisible.

Artinya:

Baris yang dimulai dengan %% akan sepenuhnya tidak terlihat.

81

Euler supports loops in command lines, as long as they fit into one single line or a multi-line. In programs, this restrictions does not hold, of course. For more information consult the following introduction.

Artinya:

Euler mendukung penggunaan loop dalam baris perintah, asalkan loop tersebut dapat dimasukkan ke dalam satu baris tunggal atau baris multi-baris. Dalam program, batasan ini tidak berlaku, tentu saja. Untuk informasi lebih lanjut, silakan merujuk ke pengenalan 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

```

It is okay to use a multi-line. Make sure the line ends with " ...".

Artinya:

Tidak masalah menggunakan baris multi-baris. Pastikan baris berakhir dengan " ...".

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

1.41421356237

Conditional structures do also work.

Artinya:

Struktur kondisional juga berfungsi.

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

Thought so!

When you execute a command, the cursor can be at any position in the command line. You can go back to a previous command or skip to the next command with the arrow keys. Or you can click into the comment section above the command to go to the command.

When you move the cursor along the line the opening and closing pairs of brackets or parentheses will highlight. Also, watch the status line. After the opening bracket of the `sqrt()` function, the status line will display a help text for the function. Execute the command with the return key.

Artinya:

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

Saat Anda memindahkan kursor sepanjang baris, pasangan kurung buka dan tutup atau kurung siku akan ditandai. Perhatikan juga baris status. Setelah kurung buka fungsi `sqrt()`, baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol Enter.

```
>sqrt(sin(10°)/cos(20°))
```

0.429875017772

To see help for the most recent command, open the help window with F1. There, you can enter text to search for. On an empty line, the help for the help window will be displayed. You can press escape to clear the line, or to close the help window.

You can double click on any command to open the help for this command. Try double clicking the `exp` command below in the command line.

Artinya:

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

Anda dapat mengklik ganda pada perintah apa pun untuk membuka bantuan untuk perintah tersebut. Coba klik ganda perintah `exp` di bawah ini di baris perintah.

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

2.5

You can copy and paste in Euler too. Use Ctrl-C and Ctrl-V for this. To mark a text, drag the mouse or use shift together with any cursor key. Moreover, you can copy the highlighted brackets.

Artinya:

Anda juga dapat menyalin dan menempelkan teks di Euler. Gunakan Ctrl-C dan Ctrl-V untuk melakukan hal ini. Untuk menandai teks, seret mouse atau gunakan tombol Shift bersama dengan tombol panah mana pun. Selain itu, Anda dapat menyalin kurung yang telah ditandai.

Basic Syntax (Sintaks Dasar)

Sintaks dasar EMT perlu dipahami agar tidak terjadi error saat menulis perintah. Misalnya, penggunaan operator, tanda kurung, maupun pemisah memiliki aturan khusus. Dengan memahami aturan sintaks ini, pengguna bisa menulis perhitungan matematis yang kompleks tanpa kesalahan format.

Euler knows the usual mathematical functions. As you have seen above, trigonometric functions work in radian or degree. To convert to degrees, append the degree symbol (with the F7 key) to the value, or use the function `rad(x)`. The square root function is called `sqrt` in Euler. Of course, $x^{(1/2)}$ is also possible.

To set variables, use either `"="` or `":="`. For the sake of clarity, this introduction uses the latter form. Spaces do not matter. But a space between commands is expected.

Multiple commands in one line are separated with `","` or `";"`. The semicolon suppresses the output of the command. At the end of the command line a `","` is assumed, if `";"` is missing.

Artinya:

Euler mengenal fungsi matematika yang umum. Seperti yang telah Anda lihat di atas, fungsi trigonometri bekerja dalam satuan 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` di Euler. Tentu saja, $x^{(1/2)}$ juga dimungkinkan.

Untuk menetapkan variabel, gunakan `"="` atau `":="`. Untuk kejelasan, pengenalan ini menggunakan bentuk terakhir. Spasi tidak berpengaruh. Namun, spasi di antara perintah diharapkan.

Perintah ganda dalam satu baris dipisahkan dengan `","` atau `";"`. Tanda titik koma `(;)` menonaktifkan output perintah. Di akhir baris perintah, tanda koma `(,)` dianggap ada jika tanda titik koma `(;)` tidak ada.

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

30.65625

EMT uses a programming syntax for expressions. To enter

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

you have to set the correct brackets and use / for fractions. Watch the highlighted brackets for assistance. Note that the Euler constant e is named E in EMT.

Artinya:

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

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

Anda harus mengatur kurung yang benar dan menggunakan / untuk pecahan. Perhatikan kurung yang ditandai untuk bantuan. Perhatikan bahwa konstanta Euler e disebut E di EMT.

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

```
8.77908249441
```

To compute a complicate expression like

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

you need to enter it in line form.

Artinya:

Untuk menghitung ekspresi yang rumit seperti

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

Anda perlu memasukkannya dalam bentuk baris.

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

```
23.2671801626
```

Carefully put brackets around sub-expressions that need to be computed first. EMT assists you by highlighting the expression that the closing bracket finishes. You will also have to enter the name "pi" for the Greek letter pi.

The result of this computation is a floating point number. It is by default printed with about 12 digits accuracy. In the following command line, we also learn how we can refer to the previous result within the same line.

Artinya:

Letakkan kurung secara hati-hati di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyoroti ekspresi yang ditutup oleh kurung penutup. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi.

Hasil perhitungan ini adalah bilangan floating point. Secara default, hasilnya dicetak dengan akurasi sekitar 12 digit. Dalam perintah baris berikut, kita juga belajar cara merujuk ke hasil sebelumnya dalam baris yang sama.

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

```
0.47619047619
10/21
```

An Euler command can be an expression or a primitive command. An expression is made of operators and functions. If necessary, it must contain brackets to force the correct order of execution. In doubt, setting a bracket is a good idea. Note that EMT shows opening and closing brackets while editing the command line.

Artinya:

Perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi terdiri dari operator dan fungsi. Jika diperlukan, ekspresi harus mengandung kurung untuk memastikan urutan eksekusi yang benar. Jika ragu, menggunakan kurung adalah ide yang baik. Perhatikan bahwa EMT menampilkan kurung pembuka dan penutup saat mengedit baris perintah.

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

```
14.4978445072
```

The numerical operators of Euler include

```
+ 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
```

and many more.

Here are some of the functions you might need. There are many more.

```
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
```

Some commands have aliases, e.g. ln for log.

Artinya:

Operator numerik Euler meliputi

```
+ operator unary atau tambah
- operator unary atau kurang
*, /
. perkalian matriks
```

a^b pangkat untuk a positif atau b bilangan bulat ($a**b$ juga berfungsi)
 $n!$ operator faktorial dan banyak lagi.

Berikut adalah beberapa fungsi yang mungkin Anda butuhkan. Ada banyak lagi.

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

Beberapa perintah memiliki alias, misalnya `ln` untuk `log`.

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

```
2
0.5
```

```
>sin(30°)
```

```
0.5
```

Make sure to use parentheses (round brackets), whenever there is doubt about the order of execution! The following is not the same as $(2^3)^4$, which is the default for 2^3^4 in EMT (some numerical systems do it the other way).

Artinya:

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

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

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

Real Numbers (Bilangan Asli)

Bilangan real dalam EMT mencakup bilangan positif, negatif, pecahan, maupun desimal. EMT mampu menampilkan hasil perhitungan real dengan presisi tinggi, bahkan mendekati angka π atau e . Dengan begitu, kita bisa mengeksplorasi operasi aritmetika maupun fungsi real tanpa terbatas pada kalkulasi manual.

The primary data type in Euler is the real number. Reals are represented in IEEE format with about 16 decimal digits of accuracy.

Artinya:

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

0.3333333333333333

The internal dual representation takes 8 bytes.

Artinya:

Representasi ganda internal membutuhkan 8 byte.

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

[illegible]

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

$$5.555555555555554 \times 16^{-1}$$

Strings (String)

String atau teks digunakan untuk menyimpan informasi berupa kata, kalimat, atau simbol non-numerik. Dalam EMT, string sering dipakai untuk memberi keterangan, label, atau menampilkan pesan. Kemampuan ini membuat notebook lebih interaktif karena hasil hitungan dapat disertai keterangan jelas.

A string in Euler is defined with "...".

Artinya:

Sebuah string dalam Euler didefinisikan dengan "...".

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

A string can contain anything.

Strings can be concatenated with `|` or with `+`. This also works with numbers, which are converted to strings in that case.

Artinya:

String dapat digabungkan menggunakan | atau +. Hal ini juga berlaku untuk angka, yang akan dikonversi menjadi string dalam hal ini.

```
>"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 cm².

The print function does also convert a number to a string. It can take a number of digits and a number of places (0 for dense output), and optionally a unit.

Artinya:

Fungsi cetak juga dapat mengubah angka menjadi string. Fungsi ini dapat menerima jumlah digit dan jumlah posisi (0 untuk output padat), serta secara optimal unit.

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

```
Golden Ratio : 1.61803
```

There is a special string `none`, which does not print. It is returned by some functions, when the result does not matter. (It is returned automatically, if the function does not have a return statement.)

Artinya:

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

```
>none
```

To convert a string to a number simply evaluate it. This works for expressions too (see below).

Artinya:

Untuk mengubah string menjadi angka, cukup evaluasi string tersebut. Hal ini juga berlaku untuk ekspresi (lihat di bawah).

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

```
1234.5
```

To define a string vector, use the vector `[...]` notation.

Artinya:

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

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

```
affe
charlie
bravo
```

The empty string vector is denoted by `[none]`. String vectors can be concatenated.

Artinya:

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

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

```
affe
charlie
bravo
```

affe
charlie
bravo

Strings can contain Unicode characters. Internally, these strings contain UTF-8 code. To generate such a string, use `u"..."` and one of the HTML entities.

Unicode strings can be concatenated like other strings.

Artinya:

String dapat berisi karakter Unicode. Secara internal, string-string ini berisi kode UTF-8. Untuk menghasilkan string semacam 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
```

$\alpha = 45^\circ$

I

In comments, the same entities like α , β etc. can be used. This may be a quick alternative to Latex. (More details on comments below).

Artinya:

Dalam komentar, entitas yang sama seperti α , β , dan sebagainya dapat digunakan. Ini mungkin menjadi alternatif cepat untuk Latex. (Rincian lebih lanjut tentang komentar di bawah ini).

There are some functions to create or analyze unicode strings. The function `strtochar()` will recognize Unicode strings, and translate them correctly.

```
>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]
```

The result is a vector of Unicode numbers. The converse function is `chartoutf()`.

Artinya:

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

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

Ü is a German letter

The function `utf()` can translate a string with entities in a variable into a Unicode string.

Artinya:

Fungsi `utf()` dapat mengubah string yang berisi entitas dalam suatu variabel menjadi string Unicode.

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

We have $\alpha=\beta$.

It is also possible to use numerical entities.

Artinya:

Juga dimungkinkan untuk menggunakan entitas numerik.

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

Ähnliches

Boolean Values (Nilai Boolean)

Nilai boolean adalah nilai logika yang hanya memiliki dua kemungkinan: benar (true) atau salah (false). EMT menggunakan boolean dalam perbandingan, kondisi, maupun percabangan perintah. Hal ini memungkinkan analisis matematis menjadi lebih dinamis dan logis.

Boolean values are represented with 1=true or 0=false in Euler. Strings can be compared, just like numbers.

Artinya:

Nilai Boolean diwakili dengan 1=benar atau 0=salah di Euler. String dapat dibandingkan, sama seperti angka. Juga dimungkinkan untuk menggunakan entitas numerik.

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

0
1

"and" is the operator "&&" and "or" is the operator "||", as in the C language. (The words "and" and "or" can only be used in conditions for "if".)

Artinya:

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

Perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi terdiri dari operator dan fungsi. Jika diperlukan, ekspresi harus mengandung kurung untuk memastikan urutan eksekusi yang benar. Jika ragu, menggunakan kurung adalah ide yang baik. Perhatikan bahwa EMT menampilkan kurung pembuka dan penutup saat mengedit baris perintah.

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

1

Boolean operators obey the rules of the matrix language.

Artinya:

Operator Boolean mengikuti aturan bahasa matriks.

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

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

You can use the function `nonzeros()` to extract specific elements from a vector. In the example, we use the conditional `isprime(n)`.

Artinya:

Anda dapat menggunakan fungsi `nonzeros()` untuk mengekstrak elemen-elemen tertentu dari sebuah vektor. Dalam contoh ini, kita menggunakan kondisi `isprime(n)`.

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

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

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

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

Output Formats (Format Keluaran)

Format keluaran di EMT bisa diatur sesuai kebutuhan, misalnya hasil ditampilkan dalam bentuk desimal, pecahan, atau simbolik. Fitur ini sangat bermanfaat dalam pembelajaran aljabar karena siswa dapat melihat perbedaan antara hasil eksak dan hasil aproksimasi.

The default output format of EMT prints 12 digits. To make sure that we see the default, we reset the format.

Artinya:

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

```
>defformat; pi
```

```
3.14159265359
```

Internally, EMT uses the IEEE standard for double numbers with about 16 decimal digits. To see the full number of digits, use the command "longestformat", or we use the operator "longest" to display the result in the longest format.

Artinya:

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

```
>longest pi
```

```
3.141592653589793
```

Here is the internal hexadecimal representation of a double number.

Artinya:

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhex(pi)
```

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

The output format can be changed permanently with a format command.

Artinya:

Format keluaran dapat diubah secara permanen dengan perintah format.

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

```
0.33333
3.14159
0.84147
```

The default is format(12).

Artinya:

Format default adalah format(12).

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

```
0.333333333333
```

Functions like "shortestformat", "shortformat", "longformat" work for vectors in the following way.

Artinya:

Fungsi seperti "shortestformat", 'shortformat', dan "longformat" bekerja pada 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

The default format for scalars is `format(12)`. But this can be changed.

Artinya:

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

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

3.1416

The function "`longestformat`" set the scalar format too.

Artinya:

Fungsi "`longestformat`" juga mengatur format skalar.

```
>longestformat; pi
```

3.141592653589793

For reference, here is a list of the most important output formats.

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

The internal accuracy of EMT is about 16 decimal places, which is the IEEE standard. Numbers are stored in this internal format.

But the output format of EMT can be set in a flexible way.

Artinya:

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

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

Ketepatan internal EMT sekitar 16 digit desimal, yang sesuai dengan standar IEEE. Angka-angka disimpan dalam format internal ini.

Namun, format output EMT dapat diatur dengan fleksibel.

```
>longestformat; pi,
```

3.141592653589793

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

3.14159

The default is `defformat()`.

Artinya:

Defaultnya adalah `defformat()`.

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

There are short operators which print only one value. The operator "longest" will print all valid digits of a number.

Artinya:

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

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

4.934802200544679

There is also a short operator for printing a result in fractional format. We have already used it above.

Artinya:

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

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

25/12

Since the internal format uses a binary way to store numbers, the value 0.1 will not be represented exactly. The error adds up a bit, as you see in the following computation.

Artinya:

Karena format internal menggunakan cara biner untuk menyimpan angka, nilai 0,1 tidak akan direpresentasikan secara tepat. Kesalahan ini sedikit bertambah, seperti yang dapat 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

But with the default "longformat" you will not notice this. For convenience, the output of very small numbers is 0.

Artinya:

Namun, dengan pengaturan default "longformat", Anda tidak akan menyadarinya. Untuk kemudahan, output dari 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

Expressions (Ekspresi) -1-

Ekspresi matematika dalam EMT bisa ditulis langsung seperti pada kertas, lalu hasilnya akan dihitung otomatis. EMT memungkinkan manipulasi ekspresi, misalnya menyederhanakan, mengembangkan, atau mengevaluasi dengan cepat. Hal ini membuat pemahaman konsep aljabar lebih praktis.

Strings or names can be used to store mathematical expressions, which can be evaluated by EMT. For this, use parentheses after the expression. If you intend to use a string as an expression, use the convention to name it "fx" or "fxy" etc. Expressions take precedence over functions.

Global variables can be used in the evaluation.

Artinya:

String atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang dapat dievaluasi oleh EMT. Untuk ini, gunakan kurung setelah ekspresi. Jika Anda bermaksud menggunakan string sebagai ekspresi, gunakan konvensi untuk menamainya "fx" atau "fxy" dan seterusnya. Ekspresi memiliki prioritas lebih tinggi daripada fungsi.

Variabel global dapat digunakan dalam evaluasi.

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

12.56637061435917

Parameters are assigned to x, y, and z in that order. Additional parameters can be added using assigned parameters.

Artinya:

Parameter-parameter tersebut ditugaskan ke x, y, dan z dalam urutan tersebut. Parameter tambahan dapat ditambahkan menggunakan parameter yang telah ditugaskan.

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

-0.919535764538

Note that expression will always use global variables, even if there is a variable in a function with the same name. (Otherwise the evaluation of expressions in functions could have very confusing results for the user that called the function.)

Artinya:

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

If you want to use another value for "at" than the global value you need to add "at=value".

Artinya:

Jika Anda ingin menggunakan nilai lain untuk "at" daripada nilai global, Anda perlu menambahkan "at=nilai".

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

45

For reference, we remark that call collections (discussed elsewhere) can contain expressions. So we can make the above example as follows.

Artinya:

Sebagai referensi, kami mencatat bahwa kumpulan panggilan (yang dibahas di tempat lain) dapat berisi ekspresi. Oleh karena itu, kami dapat membuat contoh di atas sebagai berikut.

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

45

Expressions in x are often used just like functions.

Note that defining a function with the same name like a global symbolic expression deletes this variable to avoid confusion between symbolic expressions and functions.

Artinya:

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

By way of convention, symbolic or numerical expressions should be named fx, fxy etc. This naming scheme should not be used for functions.

Artinya:

Secara konvensional, ekspresi simbolik atau numerik sebaiknya diberi nama fx, fxy, dan

seterusnya. Skema penamaan ini tidak boleh digunakan untuk fungsi.

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

A special form of an expression allows any variable as an unnamed parameter to the evaluation of the expression, not just "x", "y" etc. For this, start the expression with "@(variables) ...".

Artinya:

Sebuah bentuk khusus dari ekspresi memungkinkan penggunaan variabel apa pun sebagai parameter tanpa nama dalam evaluasi ekspresi, bukan hanya "x", 'y', dan sebagainya. Untuk ini, mulailah ekspresi dengan "@(variables) ...".

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

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

This allows to manipulate expressions in other variables for functions of EMT which need an expression in "x".

The most elementary way to define a simple function is to store its formula in a symbolic or numerical expression. If the main variable is x, the expression can be evaluated just like a function.

As you see in the following example, global variables are visible during the evaluation.

Artinya:

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

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

All other variables in the expression can be specified in the evaluation using an assigned parameter.

Artinya:

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

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

```
-0.425
```

An expression needs not be symbolic. This is necessary, if the expression contains functions, which are only known in the numerical kernel, not in Maxima.

Artinya:

Sebuah ekspresi tidak harus bersifat simbolik. Hal ini diperlukan jika ekspresi tersebut mengandung fungsi yang hanya dikenal di kernel numerik, bukan di Maxima.

Symbolic Mathematics

Matematika simbolik adalah kemampuan EMT untuk memproses ekspresi tanpa harus menghitung nilai numeriknya. Misalnya, EMT bisa menjabarkan persamaan, memfaktorkan polinomial, atau menyelesaikan bentuk umum suatu ekspresi. Ini sangat membantu dalam memahami struktur aljabar.

EMT does symbolic math with the help of Maxima. For details, start with the following tutorial, or browse the reference for Maxima. Experts in Maxima should note that there are differences in the syntax between the original syntax of Maxima and the default syntax of symbolic expressions in EMT.

Symbolic math is integrated seamlessly into Euler with &. Any expression starting with & is a symbolic expression. It is evaluated and printed by Maxima.

First of all, Maxima has an "infinite" arithmetic which can handle very large numbers.

Artinya:

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

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

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

```
>$&44!
```

This way, you can compute large results exactly. Let us compute

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

Artinya:

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)
```

Of course, Maxima has a more efficient function for this (as does the numerical part of EMT).

Artinya:

Tentu saja, Maxima memiliki fungsi yang lebih efisien untuk hal ini (demikian pula bagian numerik dari EMT).

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

To learn more about a specific function double click on it. E.g., try double clicking on "&binomial" in the previous command line. This opens the documentation of Maxima as provided by the authors of that program.

You will learn that the following works too.

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

Artinya:

Untuk mempelajari lebih lanjut tentang suatu fungsi, klik ganda pada fungsi tersebut. Misalnya, coba klik ganda pada "&binomial" di baris perintah sebelumnya. Ini akan membuka dokumentasi Maxima yang disediakan oleh pengembang program tersebut.

Anda akan mengetahui 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)
```

If you want to replace x with any specific value use "with".

Artinya:

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

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

That way you can use a solution of an equation in another equation.

Symbolic expressions are printed by Maxima in 2D form. The reason for this is a special symbolic flag in the string.

As you will have seen in previous and following examples, if you have LaTeX installed, you can print a symbolic expression with Latex. If not, the following command will issue an error message.

To print a symbolic expression with LaTeX, use \$ in front of & (or you may omit &) before the command. Do not run the Maxima commands with \$, if you don't have LaTeX installed.

Artinya:

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

Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Alasan di balik ini adalah bendera simbolik khusus dalam string.

Seperti yang Anda lihat dalam contoh sebelumnya dan berikutnya, 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 memiliki LaTeX terinstal.

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

Symbolic expressions are parsed by Euler. If you need a complex syntax in one expression, you can enclose the expression in "...". To use more than a simple expression is possible, but strongly discouraged.

Artinya:

Ekspresi simbolik diparsing oleh Euler. Jika Anda memerlukan sintaks yang kompleks dalam satu ekspresi, Anda dapat mengapit ekspresi tersebut dengan tanda kutip ganda ("..."). Penggunaan ekspresi yang lebih kompleks dari ekspresi sederhana dimungkinkan, tetapi sangat tidak disarankan.

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

25

For completeness, we remark that symbolic expressions can be used in programs, but need to be enclosed in quotes. Moreover, it is much more effective to call Maxima at compile time if possible.

Artinya:

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

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

Again, % refers to the previous result.

To make things easier we save the solution to a symbolic variable. Symbolic variables are defined with "&=".

Artinya:

Sekali lagi, % merujuk pada hasil sebelumnya.

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

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

Symbolic expressions can be used in other symbolic expressions.

Artinya:

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

A direct input of Maxima commands is available too. Start the command line with "::. The syntax of Maxima is adapted to the syntax of EMT (called the "compatibility mode").

Artinya:

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

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

$$2432902008176640000$$

```
>::: factor(10!)
```

$$\begin{matrix} 8 & 4 & 2 \\ 2 & 3 & 5 & 7 \end{matrix}$$

```
>::: factor(20!)
```

$$\begin{matrix} 18 & 8 & 4 & 2 \\ 2 & 3 & 5 & 7 & 11 & 13 & 17 & 19 \end{matrix}$$

If you are an expert in Maxima, you may wish to use the original syntax of Maxima. You can do this with ":::".

Artinya:

Jika Anda ahli dalam Maxima, Anda mungkin ingin menggunakan sintaks asli Maxima. Anda dapat melakukannya dengan ":::".

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

$$\begin{matrix} 2 \\ g \end{matrix}$$

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

$$\begin{matrix} 3 & x \\ x & E \end{matrix}$$

Such variables can be used in other symbolic expressions. Note, that in the following command the right hand side of &= is evaluated before the assignment to Fx.

Artinya:

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

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

$$125 \cdot 10^5$$

18551.64488782208

```
>fx(5)
```

18551.6448878

For the evaluation of an expression with specific values of the variables, you can use the "with" operator.

The following command line also demonstrates that Maxima can evaluate an expression numerically with float().

Artinya:

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

Perintah berikut juga menunjukkan bahwa Maxima dapat mengevaluasi suatu ekspresi secara numerik menggunakan float().

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

$$1000 \cdot 10^5 - 125 \cdot 10^5$$

2.20079141499189e+7

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

To get the Latex code for an expression, you can use the tex command.

Artinya:

Untuk mendapatkan kode LaTeX untuk suatu ekspresi, Anda dapat menggunakan perintah tex.

```
>tex(fx)
```

$x^3 \cdot e^x$

Symbolic expressions can be evaluated just like numerical expressions.

Artinya:

Ekspresi simbolik dapat dievaluasi sama seperti ekspresi numerik.

```
>fx(0.5)
```

0.206090158838

In symbolic expressions, this does not work, since Maxima does not support it. Instead, use the "with" syntax (a nicer form of the `at(...)` command of Maxima).

Artinya:

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

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

The assignment can also be symbolic.

Artinya:

Tugas tersebut juga dapat bersifat simbolis.

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

The command `solve` solves symbolic expressions for a variable in Maxima. The result is a vector of solutions.

Artinya:

Perintah ``solve`` digunakan untuk menyelesaikan ekspresi simbolik untuk suatu variabel dalam Maxima. Hasilnya adalah vektor solusi.

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

Compare with the numerical "solve" command in Euler, which needs a start value, and optionally a target value.

Artinya:

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

The numerical values of the symbolic solution can be computed by evaluation of the symbolic result. Euler will read over the assignments `x=` etc. If you do not need the numerical results for further computations you can also let Maxima find the numerical values.

Artinya:

Nilai numerik dari solusi simbolik dapat dihitung dengan mengevaluasi hasil simbolik. Euler akan memeriksa penugasan `x=` dan seterusnya. 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]
```

To get a specific symbolic solution, one can use "with" and an index.

Artinya:

Untuk mendapatkan solusi simbolik tertentu, dapat digunakan "with" dan indeks.

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

To solve a system of equations, use a vector of equations. The result is a vector of solutions.

Artinya:

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]
```

Symbolic expressions can have flags, which indicate a special treatment in Maxima. Some flags can be used as commands too, others can't. Flags are appended with "|" (a nicer form of "ev(...,flags)")

Artinya:

Ekspresi simbolik dapat memiliki bendera, yang menandakan perlakuan khusus dalam Maxima. Beberapa bendera dapat digunakan sebagai perintah, sementara yang lain tidak. Bendera 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(%)
```

Functions (Fungsi) -2-

Fungsi dalam EMT memungkinkan kita mendefinisikan relasi matematis antara input dan output. Dengan menuliskan fungsi, kita dapat memanggilnya kembali berkali-kali tanpa menulis ulang perintah. Hal ini meningkatkan efisiensi serta memberi pengalaman mirip dengan pemrograman.

In EMT, functions are programs defined with the command "function". It can be a one-line function or multiline function.

A one-line function can be numerical or symbolic. A numerical one-line function is defined by ":=".

Artinya:

Dalam EMT, fungsi adalah program yang didefinisikan dengan perintah "function". Fungsi dapat berupa fungsi satu baris atau fungsi multi-baris.

Fungsi satu baris dapat bersifat numerik atau simbolik. Fungsi numerik satu baris didefinisikan dengan ":=".

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

For an overview, we show all possible definitions for one-line functions. A function can be evaluated just like any built-in Euler function.

Artinya:

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

```
>f(2)
```

```
4.472135955
```

This function will work for vectors too, obeying the matrix language of Euler, since the expressions used in the function are vectorized.

Artinya:

Fungsi ini juga akan berfungsi untuk vektor, mengikuti bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi tersebut telah di-vectorize.

```
>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]
```

Functions can be plotted. Instead of expressions, we need only provide the function name.

In contrast to symbolic or numerical expressions, the function name must be provided in a string.

Artinya:

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

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

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

```
0.786151377757
```

By default, if you need to overwrite a built-in function, you must add the keyword "overwrite". Overwriting built-in functions is dangerous and can cause problems for other functions depending on them.

You can still call the built-in function as "_...", if it is function in the Euler core.

Artinya:

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

Anda masih dapat memanggil fungsi bawaan sebagai "_...", jika fungsi tersebut berada di inti Euler.

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

```
0.707106781187
```

We better remove this redefinition of sin.

Artinya:

Kita sebaiknya menghapus redefinisi dosa ini.

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

```
0.707106781187
```

Default Parameters (Parameter Default)

Parameter default adalah nilai awal yang otomatis digunakan jika pengguna tidak memberikan input. Fitur ini memudahkan pengguna karena tidak selalu harus mendefinisikan semua nilai. Namun, parameter tersebut tetap bisa diubah sesuai kebutuhan perhitungan.

Numerical function can have default parameters.

Artinya:

Fungsi numerik dapat memiliki parameter default.

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

Omitting this parameter uses the default value.

Artinya:

Jika parameter ini diabaikan, nilai default akan digunakan.

```
>f(4)
```

```
16
```

Setting it overwrites the default value.

Artinya:

Menyetelnya akan mengganti nilai default.

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

```
80
```

An assigned parameter overwrite it too. This is used by many Euler functions like plot2d, plot3d.

Artinya:

Parameter yang ditugaskan juga akan menggantikan parameter tersebut. Hal ini digunakan oleh banyak fungsi Euler seperti plot2d, plot3d.

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

16

If a variable is not a parameter, it must be global. One-line functions can see global variables.

Artinya:

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

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

24

But an assigned parameter overrides the global value.

If the argument is not in the list of pre-defined parameters, it must be declared with ":=":

Artinya:

Tetapi parameter yang ditentukan secara khusus akan menggantikan nilai global.

Jika argumen tidak terdapat dalam daftar parameter yang telah didefinisikan sebelumnya, argumen tersebut harus dideklarasikan dengan ":=":

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

20

Symbolic functions are defined with "&=". They are defined in Euler and Maxima, and work in both worlds. The defining expression is run through Maxima before the definition.

Artinya:

Fungsi simbolik didefinisikan dengan "&=". Fungsi ini didefinisikan dalam Euler dan Maxima, dan berfungsi di kedua lingkungan tersebut. Ekspresi definisi dijalankan melalui Maxima sebelum definisi diterapkan.

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

Symbolic functions can be used in symbolic expressions.

Artinya:

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

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

They can also be used in numerical expressions. Of course, this will only work if EMT can interpret everything inside the function.

Artinya:

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

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

```
178.635099908
```

They can be used to define other symbolic functions or expressions.

Artinya:

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik lainnya.

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

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

```
0.703467422498
```

The following works too, since Euler uses the symbolic expression in the function g, if it does not find a symbolic variable g, and if there is a symbolic function g.

Artinya:

Hal ini juga berlaku, karena Euler menggunakan ekspresi simbolik dalam fungsi g, jika tidak menemukan variabel simbolik g, dan jika terdapat 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)
```

With &= the function is symbolic, and can be used in other symbolic expressions.

Artinya:

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

simbolik lainnya.

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

With := the function is numerical. A good example is a definite integral like

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

which can not be evaluated symbolically.

If we redefine the function with the keyword "map" it can be used for vectors x. Internally, the function is called for all values of x once, and the results are stored in a vector.

Artinya:

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

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

yang tidak dapat dievaluasi secara simbolis.

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

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

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

Functions can have default values for parameters.

Artinya:

Fungsi dapat memiliki nilai default untuk parameter.

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

Now the function can be called with or without a parameter "base".

Artinya:

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

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

```
2
6.7
```

Moreover, it is possible to use assigned parameters.

Artinya:

Selain itu, dimungkinkan untuk menggunakan parameter yang telah ditentukan.

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

2

Often, we want to use functions for vectors at one place, and for individual elements at other places. This is possible with vector parameters.

Artinya:

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

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

Such a symbolic function can be used for symbolic variables.

But the function can also be used for a numerical vector.

Artinya:

Fungsi simbolik semacam itu dapat digunakan untuk variabel simbolik.

Namun, fungsi tersebut juga dapat digunakan untuk vektor numerik.

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

17

There are also purely symbolic functions, which cannot be used numerically.

Artinya:

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

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

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

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

But of course, they can be used in symbolic expressions or in the definition of symbolic functions.

Artinya:

Tentu saja, mereka dapat digunakan dalam ekspresi simbolik atau dalam definisi fungsi simbolik.

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

To summarize

- &= defines symbolic functions,

- `:=` defines numerical functions,
- `&:=` defines purely symbolic functions.

Artinya:

Ringkasan

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

Solving Expressions -3-

EMT dapat digunakan untuk menyelesaikan ekspresi dengan cara mengganti variabel tertentu dengan nilai yang diinginkan. Proses ini membuat ekspresi yang kompleks menjadi sederhana. Hal ini bermanfaat dalam memahami bagaimana perubahan nilai memengaruhi hasil.

Expressions can be solved numerically and symbolically.

To solve a simple expression of one variable, we can use the `solve()` function. It needs a start value to start the search. Internally, `solve()` uses the secant method.

Artinya:

Ekspresi dapat dipecahkan secara numerik dan simbolik.

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

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

```
1.41421356237
```

This works for symbolic expression too. Take the following function.

Artinya:

Ini juga berlaku untuk ekspresi simbolik. Pertimbangkan fungsi berikut.

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

Now we search the point, where the polynomial is 2. In `solve()`, the default target value `y=0` can be changed with an assigned variable.

We use `y=2` and check by evaluating the polynomial at the previous result.

Artinya:

Sekarang kita mencari titik di mana polinomial bernilai 2. Dalam fungsi `solve()`, nilai target default `y=0` dapat diubah dengan variabel yang ditentukan.

Kita menggunakan $y=2$ dan memeriksanya dengan mengevaluasi polinomial pada hasil sebelumnya.

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

```
0.966715594851
2
```

Solving a symbolic expression in symbolic form returns a list of solutions. We use the symbolic solver `solve()` provided by Maxima.

Artinya:

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

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

The easiest way to get the numerical values is to evaluate the solution numerically just like an expression.

Artinya:

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

```
>longest sol()
```

```
-0.6180339887498949      1.618033988749895
```

To use the solutions symbolically in other expressions, the easiest way is "with".

Artinya:

Untuk menggunakan simbol solusi dalam ekspresi lain, cara termudah adalah "dengan".

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

Solving systems of equations symbolically can be done with vectors of equations and the symbolic solver `solve()`. The answer is a list of lists of equations.

Artinya:

Penyelesaian sistem persamaan secara simbolis dapat dilakukan dengan menggunakan vektor persamaan dan pemecah simbolis `solve()`. Hasilnya adalah daftar dari daftar persamaan.

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

The function `f()` can see global variables. But often we want to use local parameters.

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

with $a=3$.

Artinya:

Fungsi `f()` dapat mengakses variabel global. Namun, seringkali kita ingin menggunakan parameter lokal.

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

dengan $a = 3$.

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

One way to pass the additional parameter to $f()$ is to use a list with the function name and the parameters (the other way are semicolon parameters).

Artinya:

Salah satu cara untuk meneruskan parameter tambahan ke $f()$ adalah dengan menggunakan daftar yang berisi nama fungsi dan parameter-parameternya (cara lain adalah menggunakan parameter titik koma).

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

2.54116291558

This does also work with expressions. But then, a named list element has to be used. (More on lists in the tutorial about the syntax of EMT).

Artinya:

Ini juga berlaku untuk ekspresi. Namun, dalam hal ini, elemen daftar yang diberi nama harus digunakan. (Lebih lanjut tentang daftar dalam tutorial tentang sintaksis EMT).

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

2.54116291558

Menyelesaikan Pertidaksamaan -4-

Selain persamaan, EMT juga mampu menyelesaikan pertidaksamaan. Hasilnya bisa berupa himpunan nilai yang memenuhi kondisi tertentu. Ini membantu memvisualisasikan solusi matematika yang lebih luas daripada sekadar angka tunggal.

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])

```

The Matrix Language (Bahasa Matriks) -5-

Bahasa matriks di EMT memungkinkan kita melakukan operasi matriks seperti penjumlahan, perkalian, invers, maupun transpose. Karena matriks sering muncul di banyak cabang matematika, fitur ini membuat EMT menjadi alat belajar yang sangat kuat.

The documentation of the EMT core contains a detailed discussion on the matrix language of Euler.

Vectors and matrices are entered with square brackets, elements separated by commas, rows separated by semicolons.

Artinya:

Dokumentasi inti EMT berisi pembahasan rinci mengenai bahasa matriks Euler.

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

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

1	2
3	4

The matrix product is denoted by a dot.

Artinya:

Produk matriks dilambangkan dengan titik.

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

3
4

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

```
[3, 4]
```

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

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

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

```
11
25
```

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

```
    1      0
    0      1
```

The main point of a matrix language is that all functions and operators work element for element.

Artinya:

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

```
>A.A
```

```
    7      10
15      22
```

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

```
    1      4
    9      16
```

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

```
    37      54
    81     118
```

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

37	54
81	118

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

1	1
1	1

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

0.333333	0.666667
0.75	1

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

-2
2.5

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

-2
2.5

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

1	0
0	1

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

1	0
0	1

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

1	4
9	16

This is not the matrix product, but a multiplication element by element. The same works for vectors.

Artinya:

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

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

```
9
16
```

If one of the operands is a vector or a scalar it is expanded in the natural way.

Artinya:

Jika salah satu operand adalah vektor atau skalar, maka operand tersebut diperluas secara alami.

```
>2*A
```

```
2      4
6      8
```

E.g., if the operand is a column vector its elements are applied to all rows of A.

Artinya:

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

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

```
1      4
3      8
```

If it is a row vector it is applied to all columns of A.

Artinya:

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

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

```
2      6
6     12
```

One can imagine this multiplication as if the row vector v had been duplicated to form a matrix of the same size as A.

Artinya:

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

This does also apply for two vectors where one is a row vector and the other is a column vector. We compute $i*j$ for i,j from 1 to 5. The trick is to multiply 1:5 with its transpose. The matrix language of Euler automatically generates a table of values.

Artinya:

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

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

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

Again, remember that this is not the matrix product!

Artinya:

Sekali lagi, ingatlah bahwa ini bukan hasil kali matriks!

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

55

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

55

Even operators like $<$ or $==$ work in the same way.

Artinya:

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

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

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

E.g., we can count the number of elements satisfying a certain condition with the function `sum()`.

Artinya:

Misalnya, kita dapat menghitung jumlah elemen yang memenuhi kondisi tertentu

menggunakan fungsi `sum()`.

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

5

Euler has comparison operators, like "=", which checks for equality.

We get a vector of 0 and 1, where 1 stands for true.

Artinya:

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

Kita mendapatkan vektor yang berisi angka 0 dan 1, di mana 1 mewakili nilai 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]
```

From such a vector, "nonzeros" selects the non-zero elements.

In this case, we get the indices of all elements greater than 50.

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

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

Of course, we can use this vector of indices to get the corresponding values in t.

Artinya:

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

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

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

As an example, let us find all squares of the numbers 1 to 1000, which are 5 modulo 11 and 3 modulo 13.

Artinya:

Sebagai contoh, mari kita cari semua kuadrat dari angka 1 hingga 1000 yang merupakan kelipatan 5 modulo 11 dan kelipatan 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 is not completely effective for integer computations. It uses double precision floating point internally. However, it is often very useful.

We can check for primality. Let us find out, how many squares plus 1 are primes.

Artinya:

EMT tidak sepenuhnya efektif untuk perhitungan bilangan bulat. Secara internal, ia menggunakan bilangan floating point presisi ganda. Namun, ia seringkali sangat berguna.

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

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

112

The function `nonzeros()` works only for vectors. For matrices, there is `mnonzeros()`.

Artinya:

Fungsi `nonzeros()` hanya berfungsi untuk vektor. Untuk matriks, terdapat fungsi `mnonzeros()`.

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

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

It returns the indices of the elements, which are not zeros.

Artinya:

Fungsi ini mengembalikan indeks dari elemen-elemen yang nilainya bukan nol.

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

1	4
2	1
2	2
3	2

These indices can be used to set the elements to some value.

Artinya:

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

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

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

The function `mset()` can also set the elements at the indices to the entries of some other matrix.

Artinya:

Fungsi `mset()` juga dapat mengatur elemen pada indeks tertentu menjadi entri dari matriks lain.

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

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

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

Artinya:

Dan dimungkinkan untuk mendapatkan elemen-elemen dalam sebuah vektor.

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

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

Another useful function is `extrema`, which returns the minimal and maximal values in each row of the matrix and their positions.

Artinya:

Fungsi lain yang berguna adalah `extrema`, yang mengembalikan nilai minimum dan maksimum dalam setiap baris matriks beserta posisinya.

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

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

We can use this to extract the maximal values in each row.

Artinya:

Kita dapat menggunakan ini untuk mengekstrak nilai maksimum di setiap baris.

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

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

This, of course, is the same as the function `max()`.

Artinya:

Tentu saja, ini sama dengan fungsi `max()`.

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

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

But with `mget()`, we can extract the indices and use this information to extract the elements at the same positions from another matrix.

Artinya:

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

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

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

Other Matrix Functions (Building Matrix) -7-

* Fungsi Matriks Lainnya (Membuat Matriks)

Selain operasi dasar, EMT menyediakan fungsi khusus untuk membangun matriks dengan pola tertentu, seperti matriks identitas atau nol. Dengan ini, siswa dapat memahami konsep matriks secara lebih konkret dan mudah diimplementasikan.

To build a matrix, we can stack one matrix on top of another. If both do not have the same number of columns, the shorter one will be filled with 0.

Artinya:

Untuk membuat 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
```

Likewise, we can attach a matrix to another side by side, if both have the same number of rows.

Artinya:

Demikian pula, kita dapat menggabungkan dua matriks 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
```

If they do not have the same number of rows the shorter matrix is filled with 0.

There is an exception to this rule. A real number attached to a matrix will be used as a column filled with that real number.

Artinya:

Jika jumlah barisnya tidak sama, matriks yang lebih pendek akan diisi dengan 0.

Ada pengecualian untuk aturan ini. Bilangan real yang terkait dengan 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

It is possible to make a matrix of row and column vectors.

Artinya:

Dimungkinkan untuk membuat matriks dari vektor baris dan vektor kolom.

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

1	2	3
1	2	3

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

1	1
2	2
3	3

The main purpose of this is to interpret a vector of expressions for column vectors.

Artinya:

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

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

1	1
2	4
3	9

To get the size of A, we can use the following functions.

Artinya:

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

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

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

For vectors, there is `length()`.

Artinya:

Untuk vektor, terdapat fungsi `length()`.

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

```
9
```

There are many other functions, which generate matrices.

Artinya:

Ada banyak fungsi lain yang menghasilkan matriks.

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

```
1 1
1 1
```

This can also be used with one parameter. To get a vector with another number than 1, use the following.

Artinya:

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]
```

Also a matrix of random numbers can be generated with `random` (uniform distribution) or `normal` (Gauß distribution).

Artinya:

Selain itu, matriks angka acak juga dapat dihasilkan dengan distribusi acak (distribusi seragam) atau distribusi normal (distribusi Gauss).

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

```
0.66566 0.831835
0.977 0.544258
```

Here is another useful function, which restructures the elements of a matrix into another matrix.

Artinya:

Berikut ini adalah fungsi berguna lainnya, yang merestrukturisasi elemen-elemen suatu 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

With the following function, we can use this and the dup function to write a rep() function, which repeats a vector n times.

Artinya:

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

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

Let us test.

Artinya:

Mari kita uji

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

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

The function multdup() duplicates elements of a vector.

Artinya:

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

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

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

The functions flipx() and flipy() revert the order of the rows or columns of a matrix. I.e., the function flipx() flips horizontally.

Artinya:

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

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

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

For rotations, Euler has rotright() and rotright().

Artinya:

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

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

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

A special function is `drop(v,i)`, which removes the elements with the indices in `i` from the vector `v`.

Artinya:

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

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

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

Note that the vector `i` in `drop(v,i)` refers to indices of elements in `v`, not the values of the elements. If you want to remove elements, you need to find the elements first. The function `indexof(v,x)` can be used to find elements `x` in a sorted vector `v`.

Artinya:

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

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

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

As you see, it does not harm to include indices out of range (like 0), double indices, or unsorted indices.

Artinya:

Seperti yang Anda lihat, tidak ada masalah jika menyertakan indeks di luar rentang (seperti 0), indeks ganda, atau indeks yang tidak terurut.

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

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

There are some special functions to set diagonals or to generate a diagonal matrix.

We start with the identity matrix.

Artinya:

Ada beberapa fungsi khusus untuk menetapkan 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

Then we set the lower diagonal (-1) to 1:4.

Artinya:

Kemudian kami mengatur diagonal bawah (-1) menjadi 1:4.

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

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

Note that we did not change the matrix A. We get a new matrix as result of setdiag().

Here is a function, which returns a tri-diagonal matrix.

Artinya:

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

Berikut adalah fungsi yang mengembalikan matriks tri-diagonal.

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

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

The diagonal of a matrix can also be extracted from the matrix. To demonstrate this, we restructure the vector 1:9 to a 3x3 matrix.

Artinya:

Diagonal dari suatu matriks juga dapat diekstraksi dari matriks tersebut. Untuk memperlihatkan hal ini, kita merestrukturisasi vektor 1:9 menjadi matriks 3x3.

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

1	2	3
4	5	6
7	8	9

Now we can extract the diagonal.

Artinya:

Sekarang kita dapat mengekstrak diagonal.

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

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

E.g. We can divide the matrix by its diagonal. The matrix language takes care that the column vector d is applied to the matrix row by row.

Artinya:

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

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

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

Vectorization (Vektorisasi) -8-

Vektorisasi memungkinkan perhitungan dilakukan sekaligus pada banyak data tanpa harus menggunakan perulangan. Dalam EMT, hal ini mempercepat perhitungan dan membuat kode lebih singkat. Teknik ini sangat berguna untuk analisis data dan operasi matematis besar.

Almost all functions in Euler work for matrix and vector input too, whenever this makes sense.

E.g., the `sqrt()` function computes the square root of all elements of the vector or matrix.

Artinya:

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

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

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

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

So you can easily create a table of values. This is one way to plot a function (the alternative uses an expression).

Artinya:

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

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

With this and the colon operator `a:delta:b`, vectors of values of functions can be generated easily.

In the following example, we generate a vector of values `t[i]` with spacing 0.1 from -1 to 1. Then we generate a vector of values of the function

$$s = t^3 - t$$

Artinya:

Dengan operator kolom `a:delta:b`, vektor nilai fungsi dapat dihasilkan dengan mudah.

Dalam contoh berikut, kita menghasilkan vektor nilai `t[i]` dengan selisih 0.1 dari -1 hingga 1. Kemudian kita menghasilkan vektor nilai fungsi

$$s = t^3 - t$$

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

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

EMT expands operators for scalars, vectors, and matrices in the obvious way.

E.g., a column vector times a row vector expands to matrix, if an operator is applied. In the following, `v'` is the transposed vector (a column vector).

Artinya:

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

Misalnya, perkalian vektor kolom dengan vektor baris akan diperluas menjadi matriks jika operator diterapkan. Dalam contoh berikut, `v'` adalah vektor transpos (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

Note, that this is quite different from the matrix product. The matrix product is denoted with a dot `"."` in EMT.

Artinya:

Perhatikan, bahwa hal ini cukup berbeda dengan hasil kali matriks. Hasil kali matriks dilambangkan dengan titik `"."` dalam EMT.

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

```
55
```

By default, row vectors are printed in a compact format.

Artinya:

Secara default, vektor baris dicetak dalam format kompak.

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

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

For matrices the special operator `.` denotes matrix multiplication, and `A'` denotes transposing. A 1x1 matrix can be used just like a real number.

Artinya:

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

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

```
5
25
```

To transpose a matrix we use the apostrophe.

Artinya:

Untuk mentransposisi sebuah matriks, kita menggunakan tanda apostrof.

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

```
1
2
3
4
```

So we can compute matrix A times vector b.

Artinya:

Jadi, kita dapat menghitung perkalian matriks A dengan vektor b.

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

```
30
70
```

Note that `v` is still a row vector. So `v'.v` is different from `v.v'`.

Artinya:

Perhatikan bahwa v masih merupakan vektor baris. Oleh karena itu, $v'.v$ berbeda dengan $v.v'$.

```
>v'.v
```

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

$v.v'$ computes the norm of v squared for row vectors v . The result is a 1×1 vector, which works just like a real number.

Artinya:

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

```
>v.v'
```

30

There is also the function `norm` (along with many other function of Linear Algebra).

Artinya:

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

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

30

Operators and functions obey the matrix language of Euler.

Here is a summary of the rules.

- A function applied to a vector or matrix is applied to each element.
- An operator operating on two matrices of same size is applied pairwise to the elements of the matrices.
- If the two matrices have different dimensions, both are expanded in a sensible way, so that they have the same size.

E.g., a scalar value times a vector multiplies the value with each element of the vector. Or a matrix times a vector (with `*`, not `.`) expands the vector to the size of the matrix by duplicating it.

The following is a simple case with the operator $\wedge$.

Artinya:

Operator dan fungsi mengikuti bahasa matriks Euler.

Berikut adalah ringkasan aturan-aturannya.

- Fungsi yang diterapkan pada vektor atau matriks diterapkan pada setiap elemen.
- Operator yang bekerja pada dua matriks dengan ukuran yang sama diterapkan secara berpasangan pada elemen-elemen matriks tersebut.
- Jika dua matriks memiliki dimensi yang berbeda, keduanya diperluas secara wajar sehingga memiliki ukuran yang sama.

Contoh: Nilai skalar dikalikan dengan vektor mengalikan nilai tersebut dengan setiap elemen vektor. Atau matriks dikalikan dengan vektor (dengan *, bukan .) memperluas vektor menjadi ukuran matriks dengan menduplikasinya.

Berikut adalah contoh sederhana dengan operator $\wedge$.

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

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

Here is a more complicated case. A row vector times a column vector expands both by duplicating.

Artinya:

Ini adalah kasus yang lebih rumit. Perkalian vektor baris dengan vektor kolom memperluas keduanya dengan cara menduplikasi.

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

1	2	3
2	4	6
3	6	9

Note that the scalar product uses the matrix product, not the *!

Artinya:

Perhatikan bahwa hasil kali skalar menggunakan hasil kali matriks, bukan tanda bintang (*).

```
>v.v'
```

```
14
```

There are numerous functions for matrices. We give a short list. You should to consult the documentation for more information on these commands.

```
sum,prod computes the sum and products of the rows
cumsum,cumprod does the same cumulatively
computes the extremal values of each row
extrema returns a vector with the extremal information
diag(A,i) returns the i-th diagonal
setdiag(A,i,v) sets the i-th diagonal
id(n) the identity matrix
det(A) the determinant
```

```
charpoly(A) the characteristic polynomial
eigenvalues(A) the eigenvalues
```

Artinya:

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

```
sum,prod menghitung jumlah dan hasil kali 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) mengatur diagonal ke-i
id(n) matriks identitas
det(A) determinan
charpoly(A) polinomial karakteristik
eigenvalues(A) nilai eigen
```

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

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

The : operator generates an equally spaced row vector, optionally with a step size.

Artinya:

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]
```

To concatenate matrices and vectors there are the operators "|" and "_".

Artinya:

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

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

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

The elements of a matrix are referred with "A[i,j]".

Artinya:

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

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

6

For row or column vectors, $v[i]$ is the i -th element of the vector. For matrices, this returns the complete i -th row of the matrix.

Artinya:

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

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

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

The indices can also be row vectors of indices. $:$ denotes all indices.

Artinya:

Indeks juga dapat berupa vektor baris dari indeks. $:$ menandakan semua indeks.

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

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

A short form for $:$ is omitting the index completely.

Artinya:

Bentuk singkat untuk $:$ adalah menghilangkan indeks sepenuhnya.

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

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

For purposes of vectorization, the elements of a matrix can be accessed as if they were vectors.

Artinya:

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

```
>A{4}
```

4

A matrix can also be flattened, using the `redim()` function. This is implemented in the function `flatten()`.

Artinya:

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]
```

To use matrices for tables, let us reset to the default format, and compute a table of sine and cosine values. Note that angles are in radians by default.

Artinya:

Untuk menggunakan matriks dalam tabel, mari kita kembalikan ke format default, dan hitung tabel nilai sinus dan kosinus. Perhatikan bahwa sudut secara default menggunakan satuan radian.

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

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

Now we append columns to a matrix.

Artinya:

Sekarang kita menambahkan kolom ke dalam 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
```

Using the matrix language, we can generate several tables of several functions at once.

In the following example, we compute $t[j]^i$ for i from 1 to n . We get a matrix, where each row is a table of t^i for one i . I.e., the matrix has the elements

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

A function which does not work for vector input should be "vectorized". This can be achieved by the "map" keyword in the function definition. Then the function will be evaluated for each element of a vector parameter.

The numerical integration `integrate()` works only for scalar interval bounds. So we need to vectorize it.

Artinya:

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

Dalam contoh berikut, kita menghitung $t[j]^i$ untuk i dari 1 hingga n . Kita mendapatkan sebuah matriks, di mana setiap baris adalah tabel t^i untuk satu i . Artinya, matriks tersebut memiliki elemen-elemen

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

Sebuah fungsi yang tidak berfungsi untuk masukan vektor harus "divektorisasi". Hal ini dapat dicapai dengan menggunakan 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. Oleh karena itu, kita perlu divektorisasinya.

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

The "map" keyword vectorizes the function. The function will now work for vectors of numbers.

Artinya:

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

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

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

Sub-Matrices and Matrix-Elements -9-

* (Sub-Matriks dan Elemen Matriks)

EMT memungkinkan kita mengambil bagian tertentu dari sebuah matriks (sub-matriks) atau elemen spesifik di dalamnya. Fitur ini berguna dalam analisis matriks yang besar, karena kita bisa fokus hanya pada bagian yang relevan.

To access a matrix element, use the bracket notation.

Artinya:

Untuk mengakses elemen matriks, gunakan notasi kurung siku.

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

1	2	3
4	5	6

5 7 8 9

We can access a complete line of a matrix.

Artinya:

Kita dapat mengakses baris lengkap dari sebuah matriks.

```
>A[2]
```

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

In case of row or column vectors, this returns an element of the vector.

Artinya:

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

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

```
2
```

To make sure, you get the first row for a 1xn and a mxn matrix, specify all columns using an empty second index.

Artinya:

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

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

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

If the index is a vector of indices, Euler will return the corresponding rows of the matrix.

Here we want the first and second row of A.

Artinya:

Untuk memastikan, Anda mendapatkan baris pertama dari A. Jika indeks adalah vektor indeks, Euler akan mengembalikan baris yang sesuai dari matriks.

Di sini, kita ingin baris pertama dan kedua dari A.xn dan matriks mxn, tentukan semua kolom menggunakan indeks kedua yang kosong.

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

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

We can even reorder A using vectors of indices. To be precise, we do not change A here, but compute a reordered version of A.

Artinya:

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

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

7	8	9
4	5	6
1	2	3

The index trick works with columns too.

This example selects all rows of A and the second and third column.

Artinya:

Trik indeks juga berlaku untuk kolom.

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

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

2	3
5	6
8	9

For abbreviation ":" denotes all row or column indices.

Artinya:

Untuk singkatan ":", menandakan semua indeks baris atau kolom.

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

3
6
9

Alternatively, leave the first index empty.

Artinya:

Sebagai alternatif, biarkan indeks pertama kosong.

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

2	3
5	6
8	9

We can also get the last line of A.

Artinya:

Kami juga dapat mendapatkan baris terakhir dari A.

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

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

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

Artinya:

Sekarang mari kita ubah elemen-elemen dari A dengan mengisi submatriks dari A dengan nilai tertentu. Hal ini memang mengubah matriks A yang disimpan.

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

4	2	3
4	5	6
7	8	9

We can also assign a value to a row of A.

Artinya:

Kita juga dapat memberikan nilai pada baris A.

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

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

We can even assign to a sub-matrix if it has the proper size.

Artinya:

Kita bahkan dapat mengisi sub-matriks jika ukurannya sesuai.

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

5	6	-1
7	8	6
7	8	9

Moreover, some shortcuts are allowed.

Artinya:

Selain itu, beberapa jalan pintas diperbolehkan.

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

0	0	-1
0	0	6
7	8	9

A warning: Indices out of bounds return empty matrices, or an error message, depending on a system setting. The default is an error message. Remember, however, that negative indices may be used to access the elements of a matrix counting from the end.

Artinya:

Peringatan: Indeks yang melebihi batas akan mengembalikan matriks kosong atau pesan kesalahan, tergantung pada pengaturan sistem. Pengaturan default adalah pesan kesalahan. Perlu diingat, bagaimanapun, bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks dengan menghitung dari akhir.

```
>A[4]
```

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

Sorting and Shuffling -10-

* (Pengurutan dan Pengacakan)

Data dalam bentuk vektor atau matriks bisa diurutkan (sorting) maupun diacak (shuffling) di EMT. Hal ini berguna untuk simulasi, pengolahan data, maupun pengacakan sampel dalam eksperimen.

The function `sort()` sorts a row vector.

Artinya:

Fungsi `sort()` mengurutkan vektor baris.

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

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

It is often necessary to know the indices of the sorted vector in the original vector. This can be used to reorder another vector in the same way.

Let us shuffle a vector.

Artinya:

Seringkali diperlukan untuk mengetahui indeks vektor yang telah diurutkan dalam vektor asli. Hal ini dapat digunakan untuk mengurutkan 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]
```

The indices contain the proper order of `v`.

Artinya:

Indeks-indeks tersebut berisi urutan yang benar dari v.

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

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

This works for string vectors too.

Artinya:

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
```

As you see, the position of double entries is somewhat random.

Artinya:

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

The function unique returns a sorted list of unique elements of a vector.

Artinya:

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

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

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

This works for string vectors too.

Artinya:

Ini juga berlaku untuk vektor string.

```
>unique(s)
```

```
a
aa
d
e
```

Linear Algebra (Aljabar Linear) -11-

Aljabar linier adalah salah satu cabang utama matematika yang banyak diterapkan di ilmu komputer, fisika, dan teknik. Dengan EMT, kita bisa menghitung determinan, eigenvalue, maupun sistem persamaan linier dengan mudah.

EMT has lots of functions to solve linear systems, sparse systems, or regression problems.

For linear systems $Ax=b$, you can use the Gauss algorithm, the inverse matrix or a linear fit. The operator $A \backslash b$ uses a version of the Gauss algorithm.

Artinya:

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

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

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

```
-4
4.5
```

For another example, we generate a 200x200 matrix and the sum of its rows. Then we solve $Ax=b$ using the inverse matrix. We measure the error as the maximal deviation of all elements from 1, which of course is the correct solution.

Artinya:

Sebagai contoh lain, kita menghasilkan matriks 200x200 dan menjumlahkan baris-barisnya. Kemudian kita menyelesaikan persamaan $Ax=b$ menggunakan matriks invers. Kita mengukur kesalahan sebagai penyimpangan maksimum dari semua elemen terhadap 1, yang tentu saja merupakan solusi yang benar.

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

```
8.790745908981989e-13
```

If the system does not have a solution, a linear fit minimizes the norm of the error $Ax-b$.

Artinya:

Jika sistem tidak memiliki solusi, regresi 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

The determinant of this matrix is 0.

Artinya:

Determinan matriks ini adalah 0.

```
>det(A)
```

0

Symbolic Matrices (Matriks Simbolik) -12-

Matriks simbolik adalah matriks yang elemennya berupa variabel, bukan angka. EMT dapat memanipulasi matriks simbolik, misalnya mencari determinan atau invers dalam bentuk ekspresi. Hal ini membantu memahami sifat-sifat matriks tanpa harus menggunakan angka tertentu.

Maxima has symbolic matrices. Of course, Maxima can be used for such simple linear algebra problems. We can define the matrix for Euler and Maxima with `&:=`, and then use it in symbolic expressions. The usual `[...]` form to define matrices can be used in Euler to define symbolic matrices.

Artinya:

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

```
>A &= [a,1,1;1,a,1;1,1,a]; $A
>$det(A), $factor(%)
>$invert(A) with a=0
>A &= [1,a;b,2]; $A
```

Like all symbolic variables, these matrices can be used in other symbolic expressions.

Artinya:

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

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

The eigenvalues can also be computed automatically. The result is a vector with two vectors of eigenvalues and multiplicities.

Artinya:

Nilai eigen juga dapat dihitung secara otomatis. Hasilnya adalah vektor yang berisi dua vektor nilai eigen dan multiplisitasnya.

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

To extract a specific eigenvector needs careful indexing.

Artinya:

Untuk mengekstrak vektor eigen tertentu, diperlukan pengindeksan yang cermat.

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

$$[1, -1]$$

Symbolic matrices can be evaluated in Euler numerically just like other symbolic expressions.

Artinya:

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

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

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

In symbolic expressions, use with.

Artinya:

Dalam ekspresi simbolik, gunakan with.

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

Access to rows of symbolic matrices work just like with numerical matrices.

Artinya:

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

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

A symbolic expression can contain an assignment. And that changes the matrix A.

Artinya:

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

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

There are symbolic functions in Maxima to create vectors and matrices. For this, refer to the documentation of Maxima or to the tutorial about Maxima in EMT.

Artinya:

Ada fungsi simbolik di Maxima untuk membuat vektor dan matriks. Untuk hal ini, silakan merujuk ke dokumentasi Maxima atau ke tutorial tentang Maxima di EMT.

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

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

The result can be evaluated numerically in Euler. For more information about Maxima, see the introduction to Maxima.

Artinya:

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

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

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

Euler has also a powerful function `xinv()`, which makes a bigger effort and gets more exact results.

Note, that with `&:=` the matrix B has been defined as symbolic in symbolic expressions and as numerical in numerical expressions. So we can use it here.

Artinya:

Euler juga memiliki fungsi yang kuat, `xinv()`, yang melakukan perhitungan yang lebih intensif dan menghasilkan hasil yang lebih akurat.

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

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

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

E.g. the eigenvalues of A can be computed numerically.

Artinya:

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]
```

Or symbolically. See the tutorial about Maxima for details on this.

Artinya:

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detail lebih lanjut mengenai

hal ini.

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

Numerical Values in symbolic Expressions -13-

* (Nilai Numerik dalam Ekspresi Simbolik)

Terkadang kita perlu mengganti sebagian variabel dalam ekspresi simbolik dengan angka tertentu. EMT mendukung penggabungan simbolik dan numerik ini, sehingga pengguna bisa mengevaluasi hasil dengan lebih fleksibel.

A symbolic expression is just a string containing an expression. If we want to define a value both for symbolic expressions and for numerical expressions, we must use "&:=".

Artinya:

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

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

1	3.14159
4	5

There is still a difference between the numerical and the symbolic form. When transferring the matrix to the symbolic form, fractional approximations for reals will be used.

Artinya:

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

```
>$&A
```

To avoid this, there is the function "mxmset(variable)".

Artinya:

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

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

Maxima can also compute with floating point numbers, and even with big floating numbers with 32 digits. The evaluation is much slower, however.

Artinya:

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

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

The precision of the big floating point numbers can be changed.

Artinya:

Ketepatan bilangan floating point besar dapat diubah.

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

```
3.14159265358979323846264338327950288419716939937510582097494\
4592307816406286208998628034825342117068b0
```

A numerical variable can be used in any symbolic expressions using "@var".

Note that this is only necessary, if the variable has been defined with "==" or "=" as a numerical variable.

Artinya:

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

Perhatikan bahwa hal ini hanya diperlukan jika variabel telah didefinisikan sebagai variabel numerik menggunakan "==" atau "=".

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

Demo - Interest Rates (Suku Bunga) -14-

Demo ini biasanya memperlihatkan bagaimana EMT digunakan untuk menghitung bunga majemuk atau suku bunga. Contoh seperti ini menghubungkan matematika dengan aplikasi nyata dalam bidang keuangan.

Below, we use Euler Math Toolbox (EMT) for the calculation of interest rates. We do that numerically and symbolically to show you how Euler can be used to solve real life problems.

Assume you have a seed capital of 5000 (say in dollars).

Artinya:

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

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

```
>K=5000
```

```
5000
```

Now we assume an interest rate of 3% per year. Let us add one simple rate and compute the result.

Artinya:

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

```
>K*1.03
```

```
5150
```

Euler would understand the following syntax too.

Artinya:

Euler juga akan memahami sintaks berikut ini.

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

```
5150
```

But it is easier to use the factor

Artinya:

Tetapi lebih mudah menggunakan faktor tersebut.

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

```
1.03  
5150
```

For 10 years, we can simply multiply the factors and get the final value with compound interest rates.

Artinya:

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

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

```
6719.58189672
```

For our purposes, we can set the format to 2 digits after the decimal dot.

Artinya:

Untuk keperluan kita, kita dapat mengatur format menjadi 2 digit di belakang titik desimal.

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

```
6719.58
```

Let us print that rounded to 2 digits in a complete sentence.

Artinya:

Mari kita cetak angka tersebut dibulatkan hingga 2 digit dalam kalimat lengkap.

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

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

What if we want to know the intermediate results from year 1 to year 9? For this, Euler's matrix language is a big help. You do not have to write a loop, but simply enter

Artinya:

Bagaimana jika kita ingin mengetahui hasil sementara dari tahun 1 hingga tahun 9? Untuk hal 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    ...
```

How does this miracle work? First the expression 0:10 returns a vector of integers.

Artinya:

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

```
>short 0:10
```

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

Then all operators and functions in Euler can be applied to vectors element for element. So

Artinya:

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]
```

is a vector of factors q^0 to q^{10} . This is multiplied by K, and we get the vector of values.

Artinya:

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

Artinya:

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)
```

Let us compare the two results, with and without rounding.

Artinya:

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

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

```
1271.61
1271.6071
```

Now there is no simple formula for the n-th year, and we must loop over the years. Euler provides many solutions for this.

The easiest way is the function iterate, which iterates a given function a number of times.

Artinya:

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

Cara termudah adalah fungsi iterate, yang mengulang suatu fungsi sebanyak kali yang ditentukan.

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

```
Real 1 x 11 matrix
```

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

We can print that in a friendly way, using our format with fixed decimal places.

Artinya:

Kami dapat mencetak angka tersebut dengan cara yang ramah, menggunakan format kami dengan jumlah 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

To get a specific element of the vector, we use indices in square brackets.

Artinya:

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

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

5150.00
5000.00 5150.00 5304.50

Surprisingly, we can also use a vector of indices. Remember that 1:3 produced the vector [1,2,3].

Let us compare the last element of the rounded values with the full values.

Artinya:

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

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

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

6719.60
6719.58

The difference is very small.

Artinya:

Perbedaannya sangat kecil.

Solving Equations (Menyelesaikan Persamaan)

-15-

EMT menyediakan perintah khusus untuk menyelesaikan persamaan, baik linear maupun non-linear. Hasilnya bisa berupa solusi eksak atau numerik, tergantung bentuk persamaan. Fitur ini sangat membantu untuk mempelajari metode penyelesaian matematika.

Now we take a more advanced function, which adds a certain rate of money each year.

Artinya:

Sekarang kita menggunakan fungsi yang lebih canggih, yang menambahkan jumlah uang tertentu setiap tahun.

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

We do not have to specify q or R for the definition of the function. Only if we run the command, we have to define these values. We select R=200.

Artinya:

Kami tidak perlu menentukan q atau R untuk definisi fungsi. Hanya jika kami menjalankan perintah, kami perlu 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    ...
```

What if we remove the same amount each year?

Artinya:

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    ...
```

We see that the money decreases. Obviously, if we get only 150 of interest in the first year, but remove 200, we lose money each year.

How can we determine the number of years the money will last? We would have to write a loop for this. The easiest way is to iterate long enough.

Artinya:

Kami melihat bahwa uangnya berkurang. Jelas, jika kami hanya mendapatkan 150 bunga pada tahun pertama, tetapi mengurangi 200, kami akan rugi setiap tahun.

Bagaimana cara menentukan berapa lama uang tersebut akan bertahan? Kami harus menulis loop untuk ini. Cara termudah adalah dengan mengulangi proses tersebut cukup lama.

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

Real 1 x 51 matrix

```
5000.00    4950.00    4898.50    4845.45    ...
```

Using the matrix language, we can determine the first negative value in the following way.

Artinya:

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

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

```
48.00
```

The reason for this is that `nonzeros(VKR<0)` returns a vector of indices i , where $VKR[i] < 0$, and `min` computes the minimal index.

Since vectors always start with index 1, the answer is 47 years.

The function `iterate()` has one more trick. It can take an end condition as an argument. Then it will return the value and the number of iterations.

Artinya:

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

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

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

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

```
-19.83
47.00
```

Let us try to answer a more ambiguous question. Assume we know that the value is 0 after 50 years. What would be the interest rate?

This is a question, which can only be answered numerically. Below, we will derive the necessary formulas. Then you will see that there is no easy formula for the interest rate. But for now, we aim for a numerical solution.

The first step is to define a function which does the iteration n times. We add all parameters to this function.

Artinya:

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

Ini adalah pertanyaan yang hanya dapat dijawab secara numerik. Di bawah ini, kita akan menyusun rumus-rumus yang diperlukan. Anda akan melihat bahwa tidak ada rumus sederhana untuk suku bunga. Namun, untuk saat ini, kita bertujuan untuk solusi numerik.

Langkah pertama adalah mendefinisikan fungsi yang melakukan iterasi 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]
```

The iteration is just as above

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

But we do longer use the global value of R in our expression. Functions like `iterate()` have a special trick in Euler. You can pass the values of variables in the expression as semicolon parameters. In this case P and R.

Moreover, we are only interested in the last value. So we take the index `[-1]`.

Let us try a test.

Artinya:

Iterasi ini sama seperti di atas

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

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

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

Mari kita coba uji coba.

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

```
-19.83
```

Now we can solve our problem.

Artinya:

Sekarang kita bisa menyelesaikan masalah kita.

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

```
3.15
```

The solve routine solves `expression=0` for the variable x. The answer is 3.15% per year. We take the start value of 3% for the algorithm. The `solve()` function always needs a start value.

We can use the same function to solve the following question: How much can we remove per year so that the seed capital is exhausted after 20 years assuming an interest rate of 3% per year.

Artinya:

Rutin penyelesaian menyelesaikan persamaan `expression=0` untuk variabel x. Jawabannya adalah 3,15% per tahun. Kami menggunakan nilai awal 3% untuk algoritma. Fungsi `solve()` selalu memerlukan nilai awal.

Kami dapat menggunakan fungsi yang sama untuk menyelesaikan pertanyaan berikut: Berapa banyak yang dapat kami ambil 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

Note that you cannot solve for the number of years, since our function assumes n to be an integer value.

Artinya:

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

Symbolic Solutions to the Interest Rate Problem

Solusi Simbolik untuk Masalah Suku Bunga

Pada bagian ini, EMT digunakan untuk mencari solusi simbolik dari masalah suku bunga. Hal ini menunjukkan bahwa EMT tidak hanya bisa menghitung angka, tetapi juga memberikan bentuk umum yang berlaku untuk berbagai kondisi.

We can use the symbolic part of Euler to study the problem. First we define our function `onepay()` symbolically.

Artinya:

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

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

We can now iterate this.

Artinya:

Sekarang kita dapat mengulangi ini.

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

We see a pattern. After n periods we have

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

The formula is the formula for the geometric sum, which is known to Maxima.

Artinya:

Kami melihat 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 ini adalah rumus untuk jumlah geometris, yang dikenal oleh Maxima.

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

This is a bit tricky. The sum is evaluated with the flag "simpsum" to reduce it to the quotient.

Let us make a function for this.

Artinya:

Ini agak rumit. Jumlah tersebut dievaluasi dengan bendera "simplsum" untuk menyederhanakannya 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)
```

The function does the same as our function f before. But it is more effective.

Artinya:

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

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

```
-19.82504734650985
```

```
-19.82504734652684
```

We can now use it to ask for the time n. When is our capital exhausted? Our initial guess is 30 years.

Artinya:

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
```

This answer says that it will be negative after 21 years.

We can also use the symbolic side of Euler to compute formulas for the payments.

Assume we get a loan of K, and pay n payments of R (starting after the first year) leaving a residual debt of Kn (at the time of the last payment). The formula for this is clearly

Artinya:

Jawaban ini menyatakan bahwa nilainya akan menjadi negatif setelah 21 tahun.

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

Misalkan kita mendapatkan pinjaman sebesar K, dan melakukan n kali pembayaran sebesar R (mulai setelah tahun pertama), sehingga tersisa utang sebesar Kn (pada saat pembayaran terakhir). Rumus untuk ini jelas adalah

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

Usually this formula is given in terms of

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

Artinya:

Biasanya rumus ini dinyatakan dalam bentuk

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

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

We can solve for the rate R symbolically.

Artinya:

Kita dapat menghitung laju R secara simbolis.

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

As you can see from the formula, this function returns a floating point error for i=0. Euler plots it nevertheless.

Of course, we have the following limit.

Artinya:

Seperti yang dapat Anda lihat dari rumus, fungsi ini mengembalikan kesalahan bilangan floating point untuk i=0. Euler tetap menampilkannya.

Tentu saja, kita memiliki batas berikut.

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

Clearly, without interest we have to pay back 10 rates of 500.

The equation can also be solved for n. It looks nicer, if we apply some simplification to it.

Artinya:

Jelas, tanpa bunga, kita harus membayar kembali 10 kali lipat dari 500.

Persamaan ini juga dapat diselesaikan untuk n. Akan terlihat lebih rapi jika kita melakukan penyederhanaan pada persamaan tersebut.

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

Soal-Soal Latihan dan Pembahasannya

R.2 Exercise Set

Soal No. 49

Sederhanakan

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

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

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

Penjelasan:

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

$$(2a^4b^{-5}c^2)^{-5}$$

$$(32a^{-20}b^{25}c^{-10})$$

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

Soal No. 50
Sederhanakan

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

$$>\$a:=((125*p^{12}*q^{-14}*r^{22}) / (25*p^8*q^6*r^{-15})) ^{-4}; \$(a)$$

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

Penjelasan:

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

$$(5p^4q^{-20}r^{37})^{-4}$$

$$(625p^{-16}q^{80}r^{-148})$$

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

Soal No. 90
Hitunglah

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

$$>2^6*2^{-3}/2^{10}/2^{-8}$$

2

Penjelasan:

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

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

$$2^{-7} \div 2^{-8}$$

$$2^1 = 2$$

Soal No. 91

Hitunglah

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

```
> $m:=(4*(8-6)^2-4*3+2*8)/(3^1+19^0); $ ( m )
```

5

Penjelasan:

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

$$\frac{4(2)^2 - 12 + 16}{3 + 1}$$

$$\frac{4 \cdot 4 + 4}{4}$$

$$\frac{16 + 4}{4}$$

5

Soal No. 92

Hitunglah

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

```
> $d:= ([4*(8-6)^2+4]*(3-2*8))/(2^2*(2^3+5)); $ ( d )
```

[-5]

Penjelasan:

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

$$\frac{[4(2)^2 + 4](3 - 16)}{4(8 + 5)}$$

$$\frac{[4(4) + 4](-13)}{4(13)}$$

$$\frac{[16 + 4](-13)}{52}$$

$$\frac{[20](-13)}{52}$$

$$\frac{-260}{52}$$

-5

R.3 Exercise Set

Soal No. 7
Hitunglah

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

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

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

Penjelasan:

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

$$(6x + y + 1) + (-3x + y - 2z - 4)$$

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

Soal No. 21
Hitunglah

$$(x + 3)(x + 6)$$

```
> $expand((x+3)*(x+6))
```

$$x^2 + 9x + 18$$

Penjelasan:

Menggunakan perkalian distributif untuk mengalikannya

Soal No. 27
Hitunglah

$$(x + 3)^2$$

```
> $expand((x+3)^2)
```

$$x^2 + 6x + 9$$

Penjelasan:

$$(x + 3)^2$$

$$(x + 3)(x + 3)$$

Soal No. 32
Hitunglah

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

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

$$9x^2 - 12x + 4$$

Penjelasan:

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

$$(3x - 2)(3x - 2)$$

Soal No. 44
Hitunglah

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

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

$$4y^2 + 20xy + 25x^2 - 9$$

Penjelasan:
Dengan menggunakan perkalian distributif biasa

R.4 Exercise Set

Soal No. 23
Faktorkan

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

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

$$(t + 3)(t + 5)$$

Soal No. 47
Faktorkan

$$z^2 - 81$$

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

$$(z - 9)(z + 9)$$

Soal No. 57
Faktorkan

$$x^2 + 12x + 36$$

```
> $factor(x^2+12*x+36)
```

$$(x + 6)^2$$

Soal No. 67
Faktorkan

$$3a^5 - 24a^2$$

```
> $factor(3*a^5-24*a^2)
```

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

Soal No. 118

Faktorkan

$$x^4 - 37x^2 + 36$$

```
> $factor(x^4-37*x^2+36)
```

$$(x - 6) (x - 1) (x + 1) (x + 6)$$

R.5 Exercise Set

Soal No. 31

Cari nilai x

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

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

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

Penjelasan:

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

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

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

$$22x = -33$$

$$x = \left(\frac{-33}{22} \right)$$

$$x = \left(\frac{-3}{2} \right)$$

Soal No. 33

Cari nilai y

$$4(3y - 1) - 6 = 5(y + 2)$$

```
> $solve(4*(3*y-1)-6=5*(y+2),y)
```

$$\left[y = \frac{20}{7} \right]$$

Penjelasan:

$$4(3y - 1) - 6 = 5(y + 2)$$

$$12y - 4 - 6 = 5y + 10$$

$$12y - 10 = 5y + 10$$

$$7y = 20$$

$$y = \left(\frac{20}{7}\right)$$

Soal No. 37
Cari nilai x

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

```
> $solve((x^2+5*x)=0,x)
```

$$[x = -5, x = 0]$$

Penjelasan:
Menggunakan cara pemfaktoran fungsi

Soal No. 51
Cari nilai x

$$14 = x(x - 5)$$

```
> $solve(14=x*(x-5),x)
```

$$[x = 7, x = -2]$$

Penjelasan:

$$14 = x(x - 5)$$

$$14 = x^2 - 5x$$

$$x^2 - 5x - 14 = 0$$

Kemudian faktorkan fungsi tersebut

Soal No. 56
Cari nilai t

$$t^2 = 25$$

```
> $solve(t^2=25,t)
```

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

Penjelasan:

$$t^2 = 25$$

$$t = \sqrt{25}$$

$$t = -5, t = 5$$

R.6 Exercise Set

Soal No. 9

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

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

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

Penjelasan:

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

$$\frac{(x-2)(x+2)}{(x-2)(x-2)}$$

$$\frac{(x+2)}{(x-2)}$$

Soal No. 16
Sederhankan

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

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

$$-\frac{1}{x+6}$$

Penjelasan:

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

$$\frac{-(x-6)}{(x-6)(x+6)}$$

$$\frac{-1}{x+6}$$

Soal No. 18
Sederhanakan

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

```
> $ratsimp(((x^2-y^2)/(x-y)^2)*(1/(x+y)))
```

$$-\frac{1}{y-x}$$

Penjelasan:

Dapat dilakukan dengan cara memfaktorkan pembilangnya

Soal No. 31
Hitunglah

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

$$> \text{\$ } (7 / (5 * x) + 3 / (5 * x))$$

$$\frac{2}{x}$$

Penjelasan:

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

$$\frac{10}{5x}$$

$$\frac{2}{x}$$

Soal No. 35
Hitunglah

$$\frac{5}{4z} - \frac{3}{8z}$$

$$> \text{\$ } (5 / (4 * z)) - (3 / (8 * z))$$

$$\frac{7}{8z}$$

Penjelasan

$$\frac{5}{4z} - \frac{3}{8z}$$

$$\frac{10}{8z} - \frac{3}{8z}$$

$$\frac{7}{8z}$$

R.7 Exercise Set

Soal No. 70
Hitung dan asumsikan semua eksponen bilangan bulat

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

$$> \text{\$ } \text{expand}((x^n + 10) * (x^n - 4))$$

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

Penjelasan:

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

$$(x^{2n} - 4x^n + 10x^n - 40)$$

$$(x^{2n} + 6x^n - 40)$$

Soal No. 71

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

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

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

Penjelasan:

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

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

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

Soal No. 72

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

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

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

Penjelasan:

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

$$(y^b - z^c)^2$$

$$y^{b(2)} - z^{c(2)}$$

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

Soal No. 73

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

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

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

Penjelasan:

Dapat dilakukan dengan operasi distributif atau teorema pascal 1331

Soal no. 74

Faktorkan

$$y^{2n} + 16y^n + 64$$

```
> $factor(y^(2*n)+16*y^n+64)
```

$$(y^n + 8)^2$$

2.3 Exercise Set

Asumsi

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

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

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

Selesaikan beberapa soal di bawah

Soal No.1

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

$$> \$ (3 * ((-1)^2 - 2 * (-1) - 6) + 1)$$

$$-8$$

Soal No. 3

$$h(f(1))$$

$$> \$ (3 * (1) + 1)^3$$

$$64$$

Soal No. 5

$$g(f(5))$$

$$> \$ ((3 * (5) + 1)^2 - 2 * (3 * (5) + 1) - 6)$$

$$218$$

Soal No. 7

$$f(h(-3))$$

$$> \$ 3 * (-3)^3 + 1$$

$$-80$$

Soal No.11

$$h(h(2))$$

$$> \$ ((2)^3)^3$$

$$512$$

3.1 Exercise Set

Soal No. 11

Sederhanakan

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

> $\$(-5+3*i)+(7+8*i)$

$$11i + 2$$

Penjelasan:

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

$$2 + 11i$$

$$11i + 2$$

Soal No. 18

Sederhanakan

$$(-5 - i) + (6 + 2i)$$

> $\$(-5-i)+(6+2*i)$

$$i + 1$$

Penjelasan:

$$(-5 - i) + (6 + 2i)$$

$$1 + i$$

$$i + 1$$

Soal No. 19

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

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

$$9i + 5$$

Penjelasan:

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

$$(3 + 4i) + (2 + 5i)$$

$$5 + 9i$$

$$9i + 5$$

Soal No. 31

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

> $\$(\text{sqrt}(-4))*\text{sqrt}(-36))$

$$-12$$

Penjelasan:

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

$$2i \cdot 6i$$

$$12(-1)$$

$$-12$$

Soal No. 34

$$\sqrt{-16} \cdot \sqrt{-100}$$

```
> $(sqrt(-16))*(sqrt(-100))
```

$$-40$$

Penjelasan:

$$\sqrt{-16} \cdot \sqrt{-100}$$

$$4i \cdot 10i$$

$$40(-1)$$

$$-40$$

3.2 Exercise Set

Soal No. 37

Sederhanakan

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

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

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

Penjelasan:

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

$$x^2 - 2x - 15 = 0$$

$$(x = -3)(x = 5)$$

Soal No. 54

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

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

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

Penjelasan:
Menggunakan rumus pemfaktoran

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

atau

$$\frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Soal No. 79

$$x^4 - 3x^2 + 2 = 0$$

```
> $solve((x^4-3*x^2+2=0),x)
```

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

Soal No. 93

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

```
> $solve((2*t^2+t)^2-4*(2*t^2+t)+3,t)
```

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

Soal No. 125

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

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

$$[x = 1]$$

3.4 Exercise Set

Soal No. 1

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

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

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

Penjelasan:

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

$$\frac{5}{20} + \frac{4}{20} = \frac{1}{t}$$

$$\frac{9}{20} = \frac{1}{t}$$

$$\frac{20}{9} = t$$

Soal No. 2

$$\frac{1}{3} - \frac{5}{6} = \frac{1}{x}$$

```
> $solve((1/3-5/6=1/x),x)
```

$$[x = -2]$$

Penjelasan:

$$\frac{1}{3} - \frac{5}{6} = \frac{1}{x}$$

$$\frac{2}{6} - \frac{5}{6} = \frac{1}{x}$$

$$\frac{-3}{6} = \frac{1}{x}$$

$$\frac{6}{-3} = x$$

$$-2 = x$$

Soal No. 3

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

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

$$[x = 286]$$

Soal No. 19

$$\frac{1}{5x+20} - \frac{1}{x^2-16} = \frac{3}{x-4}$$

```
> $solve(1/(5*x+20)-1/(x^2-16)=3/(x-4),x)
```

$$\left[x = -\frac{69}{14} \right]$$

Soal No. 29

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

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

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

3.5 Exercise Set

Soal No. 23

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

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

$$[|x+3|=10]$$

Soal No. 25

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

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

$$[|3x+1|=3]$$

Soal No. 27

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

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

$$[|4x-3|=6]$$

Soal No. 29

$$12-|x+6|=5$$

```
> $solve(12-abs(x+6)=5,x)
```

$$[|x+6|=7]$$

Soal No. 31

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

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

$$[|2x-1|=1]$$

4.3 Exercise Set

Membuat grafik dari 5 fungsi dibawah

Soal No. 49

$$f(x) = x^4 - x^3 - 7x^2 + x + 6$$

Soal No. 50

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

Soal No. 51

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

Soal No. 52

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

Soal No. 53

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

```
> plot2d("x^4+x^3-7*x^2+x+6",-5,5 )
> plot2d("x^4+x^3-3*x^2-5*x-2",-1,1)
> plot2d("x^3-7*x+6",-2,3)
> plot2d("x^3-12*x+16",-6,1)
> plot2d("-x^3+3*x^2+6*x-8",-3,4)
```