Nama: Ardi Budi Setiawan
Kelas: Matematika-B
NIM: 22305141017

EMT untuk Perhitungan Aljabar

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

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

Contoh pertama

Menyederhanakan bentuk aljabar:

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

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

Menjabarkan:

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

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

Baris Perintah

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

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

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

```
  16.7551608191
```

Perintah harus dipisahkan dengan yang 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
```

Baris perintah dieksekusi dalam urutan yang ditekan pengguna kembali. Jadi anda mendapatkan nilai baru setiap kali anda menjalankan baris kedua.

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

```
  0.540302305868
```

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

```
  0.857553215846
```

Jika dua garis terhubung dengan "..." kedua garis akan selalu dieksekusi secara bersamaan.

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

```
  1.41666666667
  1.41421568627
  1.41421356237
```

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

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

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

Sebuah garis yang dimulai dengan %% tidak akan terlihat sama sekali.

```
  81
```

Euler mendukung pengulangan pada baris perintah, selama mereka masuk ke dalam satu baris atau multi baris. Dalam program, pembatasan tentu saja tidak berlaku. Untuk informasi lebih lanjut, lihat pengantar berikut.

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

```
  1.5
  1.41666666667
  1.41421568627
  1.41421356237
  1.41421356237
```

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

```
>x := 1.5; // Komentar ada di sini sebelum ...
>repeat xnew:=(x+2/x)/2; until xnew~=x; ...
>   x := xnew; ...
>end; ...
>x,
```

```
  1.41421356237
```

Struktur bersyarat juga berfungsi.

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

```
  Thought so!
```

Saat anda menjalankan sebuah perintah,kursor dapat berada di posisi manapun di baris perintah. Kamu bisa kembali ke perintah sebelumnya atau melompat ke perintah berikutnya dengan tombol panah. Atau anda bisa mengklik ke bagian komentar di atas perintah untuk menuju ke perintah.

Ketika kamu menggerakkan kursor di sepanjang garis, pasangan tanda kurung atau tanda kurung buka dan tutup akan disorot. Juga, perhatikan baris status. Setelah kurung buka fungsi sqrt(), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan tombol kembali.

```
>sqrt(sin(10°)/cos(20°))
```

```
  0.429875017772
```

Untuk melihat bantuan guna perintah baru, buka jendela help dengan F1. Di sana, anda dapat memasukkan teks untuk dicari. Pada baris kosong, bantuan untuk jendela help akan ditampilkan. Kamu bisa menekan escape untuk menghapus baris atau untuk menutup jendela help.

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

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

```
  2.5
```

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

Dasar Sintaks

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

Untuk menyetel variabel, gunakan "=" atau ":=". Demi kejelasan, pengantar ini menggunakan bentuk yang terakhir. Spasi tidak masalah. Tapi diharapkan ada ruang di antara perintah.

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma menekan output dari perintah. Di akhir baris perintah sebuah "," diasumsikan, jika ";" hilang.

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

```
  30.65625
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

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

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

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

```
  8.77908249441
```

Untuk menghitung ekspresi rumit seperti

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

Anda harus memasukkannya dalam bentuk baris.

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

```
  23.2671801626
```

Letakkan tanda kurung dengan hati-hati di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu anda dengan menyorot ekspresi bahwa tanda kurung tutup selesai. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi

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

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

```
  0.47619047619
  10/21
```

Sebuah perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi dibuat dari operator dan fungsi. Jika perlu, fungsi harus mengandung tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, memasang tanda kurung adalah ide yang bagus. Perhatikan bahwa EMT menunjukkan tanda kurung buka dan kurung tutup saat mengedit baris perintah.

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

```
  14.4978445072
```

Operator numerik Euler termasuk

```
+ unary atau operator pertambahan
- unary atau operator pengurangan
* operator perkalian
/ operator pembagian
. produk matriks
a^b pangkat untuk a positif atau bilangan bulat b
(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
```

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

```
  2
  0.5
```

```
>sin(30°)
```

```
  0.5
```

Pastikan untuk menggunakan tanda kurung (kurung bulat), setuap 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
```

Bilangan Riil

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

```
>longest 1/3
```

```
       0.3333333333333333
```

Representasi ganda internal membutuhkan 8 byte.

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

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

```
>printdual(129/29)
```

```
  1.0001110010110000100011010011110111001011000010001101*2^2
```

```
>printhex(99999/99)
```

```
  3.F21745D1745D2*16^2
```

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

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

String

Sebuah string dalam Euler didefinisikan dengan "..."

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

```
  Sebuah string dapat berisi apa saja.
```

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

```
>"Luas lingkaran dengan jari-jari "+ 2 + " cm adalah " + pi*4 +  " cm^2."
```

```
  Luas lingkaran dengan jari-jari 2 cm adalah 12.5663706144 cm^2.
```

Fungsi print juga mengonversi angka menjadi string. Ini dapat mengambil sejumlah digit dan sejumlah tempat (0 untuk keluaran padat), dan secara optimal satu unit.

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

```
  Golden Ratio : 1.61803
```

Ada string khusus tidak ada yang tidak dicetak. Itu dikembalikan oleh beberapa fungsi, ketika hasilnya tidak masalah. (Ini dikembalikan secara otomatis, jika fungsi tidak memiliki beberapa pernyataan kembali).

```
>none
```

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

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

```
  1234.5
```

```
>p = "1234.5"()
```

```
  1234.5
```

```
>p*2 /Terbukti bahwa evaluasi di atas dapat merubah string menjadi angka
```

```
  Variable Terbukti not found!
  Error in:
  p*2 /Terbukti bahwa evaluasi di atas dapat merubah string menj ...
               ^
```

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

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

```
  affe
  charlie
  bravo
```

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

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

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

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

String Unicode dapat digabungkan seperti string lainnya.

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

```
   = 45°
```

I

Dalam komentar, entitas sama seperti , , dll dapat digunakan. Ini mungkin alternatif cepat untuk Latex. (Lebih detail di komentar bawah).

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

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

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

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

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

```
  Ü is a German letter
```

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

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

```
  We have =.
```

Dimungkinkan juga untuk menggunakan entitas numerik.

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

```
  Ähnliches
```

Nilai Boolean

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

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

```
  0
  1
```

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

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

```
  1
```

Operator Boolean mematuhi aturan bahasa matriks.

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

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

Anda dapat menggunakan fungsi nonzeros() untuk mengekstrak elemen tertentu dari vektor. Dalam contoh, kita menggunakan syarat isprime(n).

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

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

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

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

Format Output

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

```
>defformat; pi
```

```
  3.14159265359
```

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

```
>longest pi
```

```
        3.141592653589793
```

Berikut adalah representasi internal heksadesimal dari bilangan ganda.

```
>printhex(pi)
```

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

Format output dapat diubah secara permanen dengan perintah format.

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

```
      0.33333
      3.14159
      0.84147
```

Standarnya adalah format(12).

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

```
  0.333333333333
```

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

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

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

Format standar untuk skalar adalah format(12). Tapi ini bisa diubah.

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

```
  3.1416
```

Fungsi "longestformat" mengatur format skalar juga.

```
>longestformat; pi
```

```
  3.141592653589793
```

Untuk referensi, berikut adalah daftar format output yang paling penting

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

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

Tetapi format output EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

```
  3.141592653589793
```

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

```
    3.14159
```

Standarnya adalah defformat().

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

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

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

```
        4.934802200544679
```

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

```
>fraction pi^2/2
```

```
  366868/74343
```

```
>fraction pi
```

```
  312689/99532
```

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

```
  25/12
```

Karena format internal menggunakan cara biner untuk menyimpan angka, nilai 0.1 tidak akan direpresentasikan dengan tepat. Kesalahan bertambah sedikit, seperti yang anda lihat dalam perhitungan berikut.

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

```
   -1.110223024625157e-16
```

Tetapi dengan standar "longformat" anda tidak akan melihat ini. Untuk kemudahan, output dari bilangan yang sangat kecil adalah 0.

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

```
  0
```

Ekspresi

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

Variabel global dapat digunakan dalam evaluasi.

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

```
        12.56637061435917
```

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

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

```
  -0.919535764538
```

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

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

```
  36
```

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

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

```
  45
```

Untuk referensi, kita berkomentar bahwa kumpulan panggilan (dibahas di tempat lain) dapat berisi ekspresi. Jadi kita bisa membuat contoh di atas sebagai berikut.

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

```
  45
```

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

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

```
  12
```

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

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

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

Bentuk khusus dari ekspresi memungkinkan variabel apapun sebagai parameter tanpa nama untuk mengevaluasi ekspresi, bukan hanya "x", "y", dll. Untuk ini, mulai ekspresi dengan "@(variables) ...".

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

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

Ini memungkinkan untuk memanipulasi ekspresi dalam variabel lain untuk fungsi EMT yang membutuhkan ekspresi dalam "x".
Cara paling dasar untuk mendefinisikan fungsi sederhana adalah dengan menyimpan rumusnya dalam ekspresi simbolik atau numerik. Jika variabel utama adalah x, ekspresi dapat dievaluasi seperti fungsi.
Seperti yang Anda lihat dalam contoh berikut, variabel global terlihat selama evaluasi.

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

```
  -0.475
```

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

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

```
  -0.425
```

Sebuah ekspresi tidak perlu menjadi simbolik. Ini diperlukan, jika ekspresi berisi fungsi yang hanya diketahui oleh kernel numerik, bukan di Maxima.

Matematika Simbolik

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

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

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

```
>$&44!
```

$$2658271574788448768043625811014615890319638528000000000$$

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

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

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

$$2481256778$$

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

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

$$2481256778$$

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik dua kali di atasnya. Misalnya, coba klik dua kali pada "&binomial" di baris perintah sebelumnya. Ini membuka dokumentasi Maxima seperti yang disediakan oleh program itu.

Anda akan belajar bahwa yang berikut ini juga berfungsi.

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

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

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

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

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

$$120$$

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

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

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

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

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

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

Ekspresi simbolik diuraikan oleh Euler. Jika anda membutuhkan sintaks yang kompleks dalam satu ekspresi, Anda dapat menyertakan ekspresi dalam "...". Untuk menggunakan lebih dari ekspresi sederhana adalah mungkin, tetapi sangat tidak disarankan.

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

$$25$$

Untuk kelengkapan, Kita menyatakan bahwa ekspresi simbolik dapat digunakan dalam program, tetapi perlu diapit dalam tanda kutip. Selain itu, jauh lebih efektif untuk memanggil Maxima pada waktu kompilasi jika memungkinkan.

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

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

Sekali lagi, % mengacu pada hasil sebelumnya.
Untuk mempermudah, Kita simpan solusi ke variabel simbolik. Variabel simbolik didefinisikan dengan "&=".

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

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

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

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

```
2432902008176640000
```

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

```
  8  4  2
2  3  5  7
```

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

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

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

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

```
 2
g
```

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

$$x^3 \, E^x$$

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

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

$$125 \, E^5$$

$$18551.64488782208$$

```
>fx(5)
```

```
  18551.6448878
```

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

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

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

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

$$2.20079141499189e+7$$

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

$$0$$

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

```
>tex(fx)
```

```
  Variable or function fx not found.
  Error in:
  tex(fx) ...
        ^
```

Ekspresi simbolik dapat dievaluasi seperti ekspresi numerik.

```
>fx(0.5)
```

```
  0.206090158838
```

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

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

$$fx$$

Assignment juga bisa bersifat simbolik.

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

$$fx$$

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

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

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

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

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

```
  1.56155281281
```

Nilai numerik dari solusi simbolik dapat dihitung dengan evaluasi hasil simbolik. Euler akan membaca assignment x= dll. Jika anda tidak memerlukan hasil numerik untuk perhitungan lebih lanjut, Anda juga dapat membiarkan Maxima menemukan nilai numerik.

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

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

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

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

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

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

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

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

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

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

2

Ekspresi simbolik dapat memiliki flag, yang menunjukkan perlakuan khusus di Maxima. Beberapa flag dapat digunakan sebagai perintah juga, yang lain tidak. Bendera ditambahkan dengan "|" (bentuk yang lebih bagus dari "ev(...,flags)")

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

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

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

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

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

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

Fungsi

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

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

Untuk gambaran umum, kita tunjukkan semua kemungkinan definisi untuk fungsi satu baris. Suatu fungsi dapat dievaluasi sama seperti fungsi Euler bawaan lainnya.

```
>f(2)
```

```
  4.472135955
```

Fungsi ini akan bekerja untuk vektor juga, dengan mematuhi bahasa matriks Euler, karena ekspresi yang digunakan dalam fungsi divektorkan.

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

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

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

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

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

```
  0.786151377757
```

Secara default, jika Anda perlu menimpa fungsi bawaan, Anda harus menambahkan kata kunci "overwrite". Menimpa fungsi bawaan berbahaya dan dapat menyebabkan masalah untuk fungsi lain tergantung pada fungsi tersebut.

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

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

```
  0.707106781187
```

Lebih baik kita menghapus redefinisi sin ini.

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

```
  0.707106781187
```

Parameter Default

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

```
>f(4)
```

```
  16
```

Menyetelnya akan menimpa nilai default.

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

```
  80
```

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

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

```
  16
```

Jika suatu variabel bukan parameter, variabel harus global. Fungsi satu baris dapat mengetahui variabel global.

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

```
  24
```

Tetapi parameter yang ditetapkan menimpa nilai global.

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

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

```
  20
```

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

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

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

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

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

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

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

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

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

```
  178.635099908
```

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik.

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

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

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

```
  0.703467422498
```

Berikut ini juga berfungsi, karena Euler menggunakan ekspresi simbolik dalam fungsi g, jika tidak menemukan variabel simblok g, dan jika tidak ada fungsi simbolik g.

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

```
  0.703467422498
```

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

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

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

$$(x+2)^n$$

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

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

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

```
  625
```

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

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

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

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

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

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

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

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

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

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

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

$$x^3 - x$$

Dengan &= fungsinya simbolik, dan dapat digunakan dalam ekspresi simbolik lainnya.

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

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

Dengan := fungsinya numerik. Contoh yang baik adalah integral tentu seperti

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

Yang tidak dapat dievaluasi secara simbolik.
Jika kita mendefinisikan kembali fungsi dengan kata kunci "map", fungsi dapat digunakan untuk vektor x. Secara internal, fungsi dipanggil untuk semua nilai x satu kali, dan hasilnya disimpan dalam sebuah vektor.

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

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

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

```
  2
  6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

```
  2
```

Seringkali, kita ingin menggunakan fungsi untuk vektor di satu tempat, dan untuk elemen individual di tempat lain. Ini dimungkinkan dengan parameter vektor.

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

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

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

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

```
  17
```

Ada juga fungsi simbolik murni, 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)
```

$$0$$

Tetapi 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)
```

$$10\,\left(y^2+x\right)^3\,\left(9\,y^2+x+2\right)$$

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

Menyelesaikan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolik.

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

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

```
  1.41421356237
```

Ini juga berfungsi untuk ekspresi simbolis. Ambil fungsi berikut.

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

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

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

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

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

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

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

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

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

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

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

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

```
  0.966715594851
  2
```

Menyelesaikan sebuah ekspresi simbolik dalam bentuk simbolik mengembalikan daftar solusi. Kita menggunakan penyelesaian simbolik solve() yang disediakan oleh Maxima.

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

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

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

```
>longest sol()
```

```
      -0.6180339887498949       1.618033988749895
```

Untuk menggunakan solusi secara simbolik dalam ekspresi lain, cara termudah adalah "with".

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

$$0$$

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

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

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

Fungsi f() dapat mengetahui 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;
```

Salah satu cara untuk meneruskan parameter tambahan ke f() adalah dengan menggunakan daftar dengan nama fungsi dan parameter (sebaliknya adalah parameter titik koma).

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

```
  2.54116291558
```

Ini juga bekerja dengan ekspresi. Tapi kemudian, elemen daftar bernama harus digunakan. (Lebih lanjut tentang daftar di tutorial tentang sintaks EMT).

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

```
  2.54116291558
```

Menyelesaikan Pertidaksamaan

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

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

```
          D:/Documents/TUGAS KULIAH/APLIKASI KOMPUTER/Euler x64/maxima/\
  share/maxima/5.35.1/share/fourier_elim/fourier_elim.lisp
```

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

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

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

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

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

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

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

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

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

$$emptyset$$

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

$$universalset$$

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

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

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

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

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

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

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

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

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

$$\left[y + 8 < x, x < 5 - y, y < -\frac{3}{2}\right]$$

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

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

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

```
        [6 < x, x < 8, y < - 11] or [8 < x, y < - 11]
or [x < 8, 13 < y] or [x = y, 13 < y] or [8 < x, x < y, 13 < y]
or [y < x, 13 < y]
```

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

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

Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.

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

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

```
                1                2
                3                4
```

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*b //perkalian skalar
```

```
                3                6
               12               16
```

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

```
                1                0
                0                1
```

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

```
>A.A
```

```
             7            10
            15            22
```

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

```
             1             4
             9            16
```

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

```
            37            54
            81           118
```

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

```
            37            54
            81           118
```

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

```
             1             1
             1             1
```

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

```
       0.333333      0.666667
           0.75             1
```

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

```
            -2
           2.5
```

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

```
            -2
           2.5
```

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

```
                1                0
                0                1
```

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

```
                1                0
                0                1
```

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

```
                1                4
                9               16
```

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

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

```
                9
               16
```

Jika salah satu operan adalah vektor atau skalar, itu diperluas secara alami.

```
>2*A
```

```
                2                4
                6                8
```

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

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

```
                1                4
                3                8
```

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

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

```
                2                6
                6               12
```

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

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

```
                1              2
                1              2
```

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

```
                1              4
                3              8
```

Hal ini juga berlaku untuk dua vektor di mana satu adalah vektor baris dan yang lainnya adalah vektor kolom. Kita hitung i*j untuk i,j dari 1 hingga 5. Caranya adalah dengan mengalikan 1:5 dengan transposnya. Bahasa matriks Euler secara otomatis menghasilkan tabel nilai.

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

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

Sekali lagi, ingat bahwa ini bukan produk matriks!

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

```
  55
```

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

```
  55
```

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

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

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

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

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

```
  5
```

Euler memiliki operator perbandingan, seperti "==", yang memeriksa kesetaraan.
Kita mendapatkan vektor 0 dan 1, di mana 1 berarti benar.

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

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

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

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

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

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

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

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

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

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

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

EMT tidak sepenuhnya efektif untuk perhitungan bilangan bulat. Ini menggunakan floating point presisi ganda secara internal. Namun, seringkali sangat berguna.
Kita dapat memeriksa keprimaan. Mari kita cari tahu, berapa banyak kuadrat ditambah 1 adalah bilangan prima.

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

```
  112
```

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

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

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

Ini mengembalikan indeks elemen, yang bukan nol.

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

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

Indeks ini dapat digunakan untuk mengatur elemen ke beberapa nilai.

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

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

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

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

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

Dan dimungkinkan untuk mendapatkan elemen dalam vektor.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Fungsi Matriks Lainnya (Matriks Membangun)

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

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

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

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

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

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

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

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

```
>A|1
```

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

Dimungkinkan untuk membuat matriks vektor baris dan vektor kolom.

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

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

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

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

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

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

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

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut.

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

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

Untuk vektor, ada length().

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

```
  9
```

Ada banyak fungsi lain, yang menghasilkan matriks.

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

```
                1              1
                1              1
```

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

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

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

Juga matriks bilangan acak dapat dihasilkan dengan acak (distribusi uniform) atau normal (distribusi Gau).

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

```
        0.66566       0.831835
          0.977       0.544258
```

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

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

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

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

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

Mari kita uji.

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

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

Fungsi multdup() menduplikasi elemen vektor.

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

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

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

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

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

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

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

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

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

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

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

Perhatikan bahwa vektor i di drop(v,i) mengacu pada indeks elemen di v, bukan nilai dari elemen. Jika Anda ingin menghapus elemen, Anda harus menemukan elemennya terlebih dahulu. Fungsi indexof(v,x) dapat digunakan untuk menemukan elemen x dalam vektor yang diurutkan v.

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

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

Seperti yang Anda lihat, tidak ada salahnya untuk memasukkan indeks di luar rentang (seperti 0), indeks ganda, atau indeks yang tidak diurutkan.

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

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

Ada beberapa fungsi khusus untuk mengatur diagonal atau untuk menghasilkan matriks diagonal.
Kita mulai dengan matriks identitas.

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

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

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

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

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

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

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

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

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

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

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

Sekarang kita dapat mengekstrak diagonal.

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

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

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

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

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

Vektorisasi

Hampir semua fungsi di Euler bekerja untuk matriks dan input vektor juga, kapanpun ini masuk akal.
Misalnya, fungsi sqrt() yang menghitung akar kuadrat dari semua elemen vektor atau matriks.

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

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

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

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

Dengan ini dan operator titik dua a:delta:b, vektor nilai fungsi dapat dihasilkan dengan mudah.
Dalam contoh berikut, kita menghasilkan sebuah nilai vektor t[i] dengan jarak 0,1 dari -1 hingga 1. Kemudian kita menghasilkan nilai vektor 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 memperluas operator untuk skalar, vektor, dan matriks dengan cara yang jelas.

Misalnya, vektor kolom kali vektor baris berkembang menjadi matriks, jika operator diterapkan. Berikut ini, v' adalah vektor yang ditranspos (vektor kolom).

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

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

Perhatikan, bahwa ini sangat berbeda dari produk matriks. Produk matriks dilambangkan dengan titik "." di EMT.

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

```
  55
```

Secara default, vektor baris dicetak dalam format yang ringkas.

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

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

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

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

```
  5
  25
```

Untuk mentranspos matriks kita menggunakan apostrophe.

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

```
              1
              2
              3
              4
```

Jadi kita dapat menghitung matriks A kali vektor b.

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

```
             30
             70
```

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

```
>v'.v
```

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

v.v' menghitung norm v kuadrat untuk vektor baris v. Hasilnya adalah vektor 1x1, yang bekerja seperti bilangan riil.

```
>v.v'
```

```
  30
```

Ada juga fungsi norm (bersama dengan banyak fungsi lain dari Aljabar Linier).

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

```
  30
```

Operator dan fungsi memathui bahasa matriks Euler.

Berikut ringkasan aturannya.

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

- Sebuah operator yang beroperasi pada dua matriks dengan ukuran yang sama diterapkan berpasangan ke elemen matriks.

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

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

Berikut adalah kasus sederhana dengan operator ^.

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

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

Berikut adalah kasus yang lebih rumit. Vektor baris dikalikan dengan vektor kolom diekspansi ke keduanya dengan menduplikasi.

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

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

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan *!

```
>v.v'
```

```
  14
```

Ada banyak fungsi matriks. Kita berikan daftar singkat. Anda harus berkonsultasi dengan dokumentasi untuk informasi lebih lanjut tentang perintah ini.

```
  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
```

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

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

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

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

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

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

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

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

Unsur-unsur matriks disebut dengan "A[i,j]".

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

```
  6
```

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

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

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

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

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

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

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

```
  4
```

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

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

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

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

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

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

Sekarang kita menambahkan kolom ke matriks.

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

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

Dengan menggunakan bahasa matriks, kita dapat menghasilkan beberapa tabel dari beberapa fungsi sekaligus.
Dalam contoh berikut, kita menghitung t[j]^i untuk i dari 1 hingga n. Kita dapatkan matriks, di mana setiap baris adalah t^i untuk satu i. Yaitu, matriks memiliki elemen

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

Sebuah fungsi yang tidak bekerja untuk input vektor harus "divektorisasi". Ini dapat dicapai dengan kata kunci "map" dalam definisi fungsi. Kemudian fungsi tersebut akan dievaluasi untuk setiap elemen dari parameter vektor.
Integrasi numerik integrate() hanya berfungsi untuk batas interval skalar. Jadi kita perlu membuat vektor.

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

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

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

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

Sub-Matriks dan Matriks-Elemen

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
                7             8             9
  5
```

Kita dapat mengakses satu baris matriks yang lengkap.

```
>A[2]
```

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

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

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

```
  2
```

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

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

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

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

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

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

Kita bahkan dapat mengurutkan ulang A menggunakan vektor indeks. Tepatnya, kita tidak mengubah A di sini, tetapi menghitung versi A yang disusun ulang.

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

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

Trik index juga berfungsi dengan kolom.

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

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

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

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

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

```
                3
                6
                9
```

Atau, biarkan indeks pertama kosong.

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

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

Kita juga bisa mendapatkan baris terakhir A.

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

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

Sekarang mari kita ubah elemen A dengan menetapkan submatriks A ke beberapa nilai. Ini sebenarnya mengubah matriks A yang tersimpan.

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

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

Kita juga dapat menetapkan nilai ke baris A.

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

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

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

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

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

Selain itu, beberapa pintasan diperbolehkan.

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

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

Peringatan : Indeks di luar batas mengembalikan matriks kosong, atau pesan kesalahan, tergantung pengurutan sistem. Standarnya adalah pesan kesalahan. Ingat, bagaimanapun, bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks yang dihitung dari akhir.

```
>A[4]
```

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

Mengurutkan dan Mengacak

Fungsi sort() mengurutkan vektor baris.

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

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

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

Mari kita mengacak vektor.

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

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

Indeks berisi urutan yang tepat dari v.

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

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

Ini bekerja untuk vektor string juga.

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

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

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

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

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

Fungsi tunggal mengembalikan daftar elemen unik vektor yang diurutkan.

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

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

Ini bekerja untuk vektor string juga.

```
>unique(s)
```

```
  a
  aa
  d
  e
```

Aljabar Linier

EMT memiliki banyak fungsi untuk menyelesaikan sistem linier, sistem sparse, atau masalah regresi.
Untuk sistem linier Ax=b, Anda dapat menggunakan algoritma Gauss, matriks invers, atau kesesuaian linier. Operator A/b menggunakan versi algoritma Gauss.

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

```
            -4
           4.5
```

Untuk contoh lain, kita buat matriks 200x200 dan jumlah barisnya. Kemudian kita selesaikan Ax=b menggunakan matriks invers. Kita ukur kesalahan sebagai deviasi maksimal semua elemen dari 1, yang tentu saja merupakan solusi yang benar.

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

```
    8.790745908981989e-13
```

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

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

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

Determinan dari matriks ini adalah 0.

```
>det(A)
```

```
  0
```

Matriks Simbolik

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Untuk mengekstrak vektor eigen tertentu perlu pengindeksan yang cermat.

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

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

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

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

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

```
                1                4
                5                2
```

Dalam ekspresi simbolik, gunakan with.

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

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

Akses ke baris matriks simbolik bekerja seperti halnya dengan matriks numerik.

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

$$[1, a]$$

Ekspresi simbolik dapat berisi assignment. Dan itu mengubah matriks A.

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

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

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

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

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

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

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

$$\left(\begin{smallmatrix}-2 & 1\\ \frac{3}{2} & -\frac{1}{2}\end{smallmatrix}\right)$$

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

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

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

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

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

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

```
                             1                             0
                             0                             1
```

Misalnya nilai eigen dari A dapat dihitung secara numerik.

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

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

Atau secara simbolik. Lihat tutorial tentang Maxima untuk detailnya.

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

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

Nilai Numerik dalam Ekspresi Simbolik

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

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

```
                1        3.14159
                4              5
```

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

```
>$&A
```

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

Untuk menghindarinya, ada fungsi "mxset(variable)".

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

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

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

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

$$1.414213562373095$$

Ketepatan angka floating point besar juga dapat diubah.

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

```
          3.14159265358979323846264338327950288419716939937510582097494\
 459230781640628620899862803482534211706 8b0
```

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

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

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

$$-5.424777960769379$$

Demo - Suku Bunga

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

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

```
>K=5000
```

```
  5000
```

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

```
>K*1.03
```

```
  5150
```

Euler akan memahami sintaks berikut juga.

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

```
  5150
```

Tapi lebih mudah menggunakan faktornya.

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

```
  1.03
  5150
```

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

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

```
  6719.58189672
```

Untuk tujuan kita, kita dapat mengatur format menjadi 2 digit setelah titik desimal.

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

```
      6719.58
```

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

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

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

Bagaimana jika kita ingin mengetahui hasil antara dari tahun 1 sampai tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menulis loop, cukup masukkan.

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

```
  Real 1 x 11 matrix

       5000.00       5150.00       5304.50       5463.64       ...
```

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

```
>short 0:10
```

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

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

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

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

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

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

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

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

Mari kita bandingkan dua hasil, dengan dan tanpa pembulatan.

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

```
                  1271.61
                1271.6071
```

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

Cara termudah adalah iterasi, yang mengulangi fungsi tertentu beberapa kali.

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

```
  Real 1 x 11 matrix

     5000.00     5150.00     5304.50     5463.64      ...
```

Kita dapat mencetaknya dengan cara yang ramah, menggunakan format kita dengan tempat desimal tetap.

```
>VKr'
```

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

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

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

```
      5150.00
      5000.00     5150.00     5304.50
```

Anehnya, kita juga bisa menggunakan vektor indeks. Ingat bahwa 1:3 menghasilkan vektor [1,2,3].
Mari kita bandingkan elemen terakhir dari nilai yang dibulatkan dengan nilai penuh.

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

```
      6719.60
      6719.58
```

Perbedaannya sangat kecil.

Menyelesaikan persamaan

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

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

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

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

```
  Real 1 x 11 matrix

     5000.00     5350.00     5710.50     6081.82     ...
```

Bagaumana jika kita menghapus jumlah yang sama setiap tahun?

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

```
  Real 1 x 11 matrix

     5000.00     4950.00     4898.50     4845.45     ...
```

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

Bagaimana kita bisa menentukan berapa tahun uang itu akan bertahan? Kita harus menuliskan loop untuk ini. Cara termudah adalah dengan iterasi yang cukup lama.

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

```
  Real 1 x 51 matrix

     5000.00     4950.00     4898.50     4845.45     ...
```

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

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

```
        48.00
```

Alasan untuk ini adalah bukan 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. Itu bisa mengambil kondisi akhir sebagai argumen. Kemudian akan mengembalikan nilai dan jumlah iterasi.

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

```
       -19.83
        47.00
```

Mari kita coba menjawab pertanyaan yang lebih ambigu. Asumsikan kita tahu bahwa nilainya adalah 0 setelah 50 tahun. Apa yang akan menjadi suku bunga?

Ini adalah pertanyaan yang hanya bisa dijawab dengan angka. Di bawah ini, kita akan mendapatkan formula yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada formula yang mudah untuk suku bunga. Tapi untuk saat ini, kita bertujuan untuk solusi numerik.

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

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

Iterasinya sama seperti di atas

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

Tapi kita lebih lama menggunakan nilai global dari R dalam ekspresi kita. Fungsi seperti iterate() mempunyai trik khusus di Euler. Anda dapat meneruskan nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R.

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

Mari kita coba tes.

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

```
       -19.83
```

Sekarang kita bisa menyelesaikan masalah kita.

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

```
         3.15
```

Penyelesaian rutin menyelesaikan ekspresi=0 untuk variabel x. Jawabannya adalah 3.15% per tahun. Kita ambil nilai awal 3% untuk algoritma. Fungsi solve() selalu membutuhkan nilai awal.

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

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

```
      -336.08
```

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

Solusi simbolik untuk Masalah Suku Bunga

Kita dapat menggunakan bagian simbolik dari Euler untuk mempelajari masalah tersebut. Pertama kita definisikan fungsi onepay() kita secara simbolik.

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

$$R + q\,K$$

Kita sekarang dapat mengiterasi ini.

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

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

Kita lihat sebuah pola. Setelah n periode kita dapatkan

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

Rumusnya adalah rumus untuk jumlah geometrik, yang telah diketahui Maxima.

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

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

Ini agak rumit. Jumlahnya dievaluasi dengan flag "simpsum" untuk menguranginya menjadi hasil bagi. Mari kita membuat fungsi untuk ini.

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

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

Fungsi tersebut melakukan hal yang sama seperti fungsi f sebelumnya. Tapi itu lebih efektif.

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

```
       -19.82504734650985
       -19.82504734652684
```

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

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

```
        20.51
```

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

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

Asumsikan kita mendapatkan pinjaman sebesar K, dan membayar n pembayaran sebesar R (dimulai setelah tahun pertama) meninggalkan sisa hutang sebesar Kn (pada saat pembayaran terakhir). Rumus ini dengan jelas

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

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

Biasanya rumus ini diberikan dalam bentuk

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

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

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

Kita dapat menyelesaikan laju R secara simbolis.

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

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

Seperti yang Anda lihat dari rumus, fungsi ini mengembalikan kesalahan titik mengambang untuk i=0. Euler tetap merencanakannya.

Tentu saja, kami memiliki limit berikut.

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

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

Jelas, tanpa bunga kita harus membayar kembali 10 suku dari 500.

Persamaan juga dapat diselesaikan untuk n. Kelihatannya lebih bagus, jika kita menerapkan beberapa penyederhanaan untuk itu.

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

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

Nama: Ardi Budi Setiawan
Kelas: Matematika-B
NIM: 22305141017

contoh penyelesaian soal-soal Aljabar

1. Tulis persamaan yang setara tanpa eksponen negatif (Sederhanakan)

a.

$$\frac{m^{-1}n^{-12}}{t^{-6}}$$

b.

$$m^{12} \cdot m^{-25}$$

c.

$$\frac{x^2y^2}{x^{-1}y}$$

d.

$$(8ab^7)(-7a^{-5}b^2)$$

e.

$$n^9 \cdot n^{-9}$$

```
>"a. Hasilnya adalah ", $&(m^(-1)*n^(-12))/t^-6
```

```
  a. Hasilnya adalah
```

$$\frac{t^6}{m\,n^{12}}$$

Seperti yang kita ketahui bahwa:

$$x^{-1}$$

akan sama dengan

$$\frac{1}{x}$$

Maka, dari soal yang (a) dapat kita rubah penyebut dan pembilangnya sesuai dengan aturan eksponen tersebut, agar pangkat negatif bisa hilang.

```
>"b. Hasilnya adalah ", $&(m^(12-25))
```

```
  b. Hasilnya adalah
```

$$\frac{1}{m^{13}}$$

```
>"c. Hasilnya adalah ", $&((x^2*y^2)/(x^-1*y))
```

```
  c. Hasilnya adalah
```

$$x^3\,y$$

```
>"d. Hasilnya adalah ", $&((8*a*b^7)/(-7*a^-5*b^2))
```

```
  d. Hasilnya adalah
```

$$-\frac{8\,a^6\,b^5}{7}$$

```
>"e. Hasilnya adalah ", $&(n^(9-9))
```

```
  e. Hasilnya adalah
```

$$1$$

Untuk soal (b), (c), (d), (e):
Seperti yang kita ketahui bahwa:

$$x^2 : x^1 = x^{2-1}$$

sehingga, akan berlaku juga pada pecahan:
sebagai contoh

$$\frac{x^5}{x^3}$$

$$= x^{5-3}$$

Maka, hasil soal-soal diatas adalah benar.

---

2. Faktorkan persamaan berikut ini:
a.

$$2x^2 + 13x + 15 = 0$$

b.

$$2x^3 + 9x^2 + 12x + 4 = 0$$

c.

$$5a^3 - 10a^2 + 25a - 50 = 0$$

```
>"a. Menulis persamaan kuadrat: ", ...
>$& (2*x^2)+13*x+15=0, $factor(%), "Itu adalah Memfaktorkan persamaannya "
```

a. Menulis persamaan kuadrat:

$$2\,x^2 + 13\,x + 15 = 0$$

$$(x + 5)\,(2\,x + 3) = 0$$

Itu adalah Memfaktorkan persamaannya

```
>"b. Menulis persamaan kuadrat: ", ...
>$& (2*x^3)+(9*x^2)+12*x+4=0, $factor(%), "Itu adalah Memfaktorkan persamaannya "
```

b. Menulis persamaan kuadrat:

$$2\,x^3 + 9\,x^2 + 12\,x + 4 = 0$$

$$(x + 2)^2\,(2\,x + 1) = 0$$

Itu adalah Memfaktorkan persamaannya

```
>"c. Menulis persamaan kuadrat: ", ...
>$& (5*a^3)-(10*a^2)+25*a-50=0, $factor(%), "Itu adalah Memfaktorkan persamaannya "
```

c. Menulis persamaan kuadrat:

$$5\,a^3 - 10\,a^2 + 25\,a - 50 = 0$$

$$5\,(a-2)\,\left(a^2+5\right) = 0$$

Itu adalah Memfaktorkan persamaannya

Pastikan koefisien a tidak nol. Pastikan koefisien a dalam persamaan kuadrat ax^2 + bx + c = 0 tidak sama dengan nol. Cari faktor-faktor a dan c. Faktorkan a dan c menjadi faktor-faktor yang mengalikan keduanya. Cari dua bilangan yang menjumlahkan b. Temukan dua bilangan yang dapat menjumlahkan b dan mengalikan a dan c.

Tulis persamaan dalam bentuk faktorisasi. Tulis persamaan dalam bentuk (ax - p)(x - q) = 0 dengan menggunakan bilangan-bilangan yang ditemukan pada langkah sebelumnya. Atasi faktorisasi lebih lanjut jika mungkin. Faktorkan lebih lanjut masing-masing faktor jika memungkinkan.

---

3. Temukan solusi dari persamaan berikut:

a.

$$2(x+7) = 5x + 14$$

b.

$$9y^2 + 15y + 4 = 0$$

c.

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

d.

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

```
>"a.", fx &=(2*(x+7)=5*x+14) ; sol &= solve(2*(x+7)=5*x+14,x); ...
>"Menulis persamaan kuadrat", $& fx, ...
>"Ditemukan solusi untuk x yakni", $&sol
```

a.
Menulis persamaan kuadrat

$$2\,(x+7) = 5\,x + 14$$

Ditemukan solusi untuk x yakni

$$[x = 0]$$

```
>"b.", fy &=(9*y^2+15*y+4=0) ; sol &= solve(9*y^2+15*y+4=0,y); ...
>"Menulis persamaan kuadrat", $& fy, ...
>"Ditemukan solusi untuk y yakni", $&sol
```

b.
Menulis persamaan kuadrat

$$9\,y^2 + 15\,y + 4 = 0$$

Ditemukan solusi untuk y yakni

$$\left[y = -\frac{4}{3}, y = -\frac{1}{3}\right]$$

```
>"c.", fx &=(3*x^3+6*x^2-27*x-54=0) ; sol &= solve(3*x^3+6*x^2-27*x-54=0,x); ...
>"Menulis persamaan kuadrat", $& fx, ...
>"Ditemukan solusi untuk x yakni", $&sol
```

c.
Menulis persamaan kuadrat

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

Ditemukan solusi untuk x yakni

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

```
>"d.", fx &=(sqrt(x+4)-2=1) ; sol &= solve(sqrt(x+4)-2=1,x); ...
>"Menulis persamaan kuadrat", $& fx, ...
>"Ditemukan solusi untuk x yakni", $&sol
```

d.
Menulis persamaan kuadrat

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

Ditemukan solusi untuk x yakni

$$[x = 5]$$

Hitung Nilai Diskriminan

$$(D = b^2 - 4ac)$$

untuk menentukan jenis solusi:
- Jika D > 0, maka terdapat dua akar berbeda (solusi riil).
- Jika D = 0, maka terdapat satu akar ganda (solusi riil).
- Jika D < 0, maka terdapat dua akar kompleks konjugat (solusi kompleks).

Lalu Hitung Nilai x1 dan x2 dengan menggunakan rumus kuadrat

$$x = \frac{-b \pm \sqrt{(b^2 - 4ac)}}{2a}$$

untuk menghitung nilai x1 dan x2 berdasarkan diskriminan dan koefisien a, b, dan c.

---

4. Tentukan solusi dari pertidaksamaan tersebut

a.

$$|x + 8| < 9$$

b.

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

c.

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

```
>"a. Solusinya adalah ", $&fourier_elim([abs(x+8)<9],[x])
```

a. Solusinya adalah

$$[-17 < x, x < 1]$$

```
>"b. Solusinya adalah ", $&fourier_elim([abs(p-4)+abs(p+4)<8],[p])
```

b. Solusinya adalah

$$emptyset$$

```
>"c. Solusinya adalah ", $&fourier_elim([abs((2*x+1)/3)>5],[x])
```

c. Solusinya adalah

$$[7 < x] \vee [x < -8]$$

Pertidaksamaan Nilai Mutlak ($|ax + b| < c$ atau $|ax + b| > c$):
Bagi pertidaksamaan menjadi dua kasus: ketika nilai dalam nilai mutlak positif (ax + b) dan ketika nilai dalam nilai mutlak negatif (-(ax + b)). Selesaikan kedua kasus tersebut secara terpisah dan gabungkan solusi-solusi dari kedua kasus. Maka, akan ditemukan solusinya seperti jawaban diatas.

---

5. Diberikan matriks sebagai berikut

a.

$$\begin{pmatrix} -2 & 5 & 3 \\ 4 & -1 & 3 \\ 7 & -2 & 5 \end{pmatrix}$$

b.

$$\begin{pmatrix} 10 & 20 & -30 & 15 \\ 3 & -7 & 14 & -8 \\ -7 & -2 & -1 & 2 \\ 4 & 4 & -3 & 1 \end{pmatrix}$$

Tentukan determinan dari matriks tersebut.

```
>"Membuat matriks", A&=[-2,5,3;4,-1,3;7,-2,5]; $ A,  ...
>"Determinannya adalah", $ det(A)
```

Membuat matriks

$$\begin{pmatrix} -2 & 5 & 3 \\ 4 & -1 & 3 \\ 7 & -2 & 5 \end{pmatrix}$$

Determinannya adalah

$$0$$

```
>"Membuat matriks", B&=[10,20,-30,15;3,-7,14,-8;-7,-2,-1,2;4,4,-3,1]; $ B,  ...
>"Determinannya adalah", $ det(B)
```

Membuat matriks

$$\begin{pmatrix} 10 & 20 & -30 & 15 \\ 3 & -7 & 14 & -8 \\ -7 & -2 & -1 & 2 \\ 4 & 4 & -3 & 1 \end{pmatrix}$$

Determinannya adalah

$$125$$

Cara menentukan determinan matriks
Gunakan Metode Ekspansi Kofaktor:
Untuk matriks 4x4, Anda dapat menggunakan metode ekspansi kofaktor, yang melibatkan langkah-langkah berikut:

```
a. Pilih salah satu baris atau kolom (biasanya yang memiliki elemen
```

nol adalah pilihan yang baik).

```
b. Untuk setiap elemen di baris atau kolom yang Anda pilih, hitung
```

kofaktornya (produk dari matriks 3x3 yang dihasilkan dengan menghilangkan baris dan kolom yang berisi elemen tersebut) dan kalikan dengan elemen tersebut.

```
c. Jumlahkan semua produk yang Anda hitung dalam langkah sebelumnya.
```

Ini adalah rumus umum untuk mencari determinan matriks 4x4 dengan metode ekspansi kofaktor.