EMT untuk Perhitungan Aljabar

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

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

Contoh pertama

Menyederhanakan bentuk aljabar:

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

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

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

Menjabarkan:

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

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

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

Baris Perintah

Satu baris perintah Euler terdiri atas satu atau beberapa perintah Euler yang diakhiri dengan tanda titik koma ";" atau koma ",". Tanda titik koma (";") mencegah hasil perhitungan ditampilkan. Koma setelah perintah terakhir boleh dihilangkan.

Baris perintah berikut hanya akan menampilkan hasil ekspresi, bukan penugasan variabel atau perintah formatnya.

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

```
  16.7551608191
```

Perintah-perintah harus dipisahkan dengan spasi. Baris perintah berikut menampilkan dua hasil:

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

```
  50.2654824574
  100.530964915
```

Baris perintah dieksekusi sesuai urutan pengguna menekan tombol Return. Jadi Anda akan mendapat nilai baru setiap kali menjalankan baris kedua.

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

```
  0.540302305868
```

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

```
  0.857553215846
```

Jika dua baris dihubungkan dengan "...", kedua baris tersebut akan selalu dijalankan 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 ke dalam dua baris atau lebih. Anda dapat menekan Ctrl+Enter untuk memisahkan satu baris menjadi dua pada posisi kursor saat ini, atau Ctrl+Backspace untuk menggabungkan baris-baris tersebut.

Untuk melipat (menyembunyikan) semua baris ganda, tekan Ctrl+L. Kemudian, baris-baris berikutnya hanya akan terlihat jika salah satu dari baris tersebut dalam keadaan fokus. Untuk melipat satu baris ganda, awali baris pertama dengan "%+ ".

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

Baris yang diawali dengan %% akan sepenuhnya tidak terlihat.

```
  81
```

Euler mendukung pengulangan dalam baris perintah, selama muat dalam satu baris tunggal atau multi-baris. Dalam program, batasan ini tentu 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 masalah menggunakan multi-baris. Pastikan baris diakhiri dengan " ...".

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

```
  1.41421356237
```

Struktur kondisional juga dapat digunakan

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

```
  Thought so!
```

Saat kamu mengeksekusi perintah, kursor dapat berada di posisi mana pun dalam baris perintah. Kamu dapat kembali ke perintah sebelumnya atau melompat ke perintah berikutnya dengan tombol panah. Atau kamu dapat mengklik bagian komentar di atas perintah untuk langsung menuju ke perintah tersebut.

Saat kamu menggerakkan kursor sepanjang baris, pasangan tanda kurung buka dan tutup akan disorot. Perhatikan juga baris status. Setelah tanda kurung buka dari fungsi sqrt(), baris status akan menampilkan teks bantu untuk fungsi tersebut. Eksekusi perintah dengan menekan tombol Enter.

```
>sqrt(sin(10°)/cos(20°))
```

```
  0.429875017772
```

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

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

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

```
  2.5
```

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

Sintaks Dasar

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

Untuk menetapkan variabel, gunakan tanda "=" atau ":=". Dalam panduan ini, kita akan menggunakan ":=" untuk mempermudah pemahaman. Penambahan spasi tidak memengaruhi perintah, namun disarankan memberikan spasi antar perintah untuk keterbacaan.

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma (;) digunakan untuk mencegah hasil perintah ditampilkan di layar. Jika di akhir baris tidak diberi tanda pemisah, Euler secara otomatis akan menganggapnya sebagai koma.

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

```
  30.65625
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk memasukkan

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

kamu harus mengatur tanda kurung dengan benar dan menggunakan / untuk pecahan. Perhatikan tanda kurung yang di-highlight untuk bantuan. Perhatikan bahwa konstanta Euler e disebut E dalam EMT.

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

```
  8.77908249441
```

Untuk menghitung ekspresi rumit seperti

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

kamu perlu memasukkan dalam bentuk barisan.

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

```
  23.2671801626
```

Pastikan untuk meletakkan tanda kurung dengan tepat pada setiap sub-ekspresi yang perlu dihitung terlebih dahulu. EMT akan memudahkan Anda dengan cara menyorot bagian ekspresi yang sesuai saat tanda kurung ditutup. Perlu diingat bahwa untuk merepresentasikan huruf Yunani pi, Anda harus mengetikkan "pi".

Hasil dari komputasi tersebut akan berupa bilangan floating point (desimal). Secara bawaan, hasil akan ditampilkan dengan tingkat akurasi sekitar 12 digit. Pada bagian selanjutnya, kita juga akan mempelajari bagaimana cara menggunakan hasil perhitungan sebelumnya dalam baris perintah yang sama.

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

```
  0.47619047619
  10/21
```

Perintah dalam Euler dapat berupa ekspresi atau perintah primitif. Ekspresi dibentuk dari operator dan fungsi. Jika diperlukan, ekspresi harus memuat tanda kurung untuk memastikan urutan eksekusi yang benar. Jika ragu, sebaiknya gunakan tanda kurung. Perhatikan bahwa EMT akan menampilkan tanda kurung buka dan tutup yang sesuai selama proses pengeditan baris perintah.

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

```
  14.4978445072
```

OPerator numerik dalam Euler meliputi:

```
+ plus unary atau operator penjumlahan
- minus unary atau operator pengurangan
*, /
. produk matiks
a^b pemangkatan untuk a positif atau b integer (bisa juga menggunakan
```

a**b)

```
n! operator faktorial
```

dan masih banya lagi

Berikut beberapa fungsi yang mungkin sering digunakan (masih banyak fungsi lainnya):

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

Beberapa perintah memiliki alias, misalnya ln untuk log.

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

```
  2
  0.5
```

```
>sin(30°)
```

```
  0.5
```

Pastikan untuk menggunakan tanda kurung (tanda kurung bulat) ketika ada keraguan tentang urutan eksekusi! Contoh berikut tidak sama dengan (2^3)^4, yang merupakan standar untuk 2^3^4 dalam EMT (beberapa sistem numerik mungkin menerapkan urutan yang berbeda).

```
>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. Bilangan riil direpresentasikan dalam format IEEE dengan akurasi sekitar 16 digit desimal.

```
>longest 1/3
```

```
       0.3333333333333333
```

Representasi internal ganda menggunakan 8 byte.

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

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

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

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

String

String dalam Euler didefinisikan dengan tanda "..." .

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

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

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

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

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

Fungsi print juga dapat mengonversi angka menjadi string. Fungsi ini dapat menerima sejumlah digit dan sejumlah tempat (0 untuk hasil yang ringkas), dan sebaiknya menambahkan satuan (unit) juga.

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

```
  Golden Ratio : 1.61803
```

Terdapat string khusus bernama none, yang tidak akan ditampilkan saat dicetak. String ini dikembalikan oleh beberapa fungsi ketika hasilnya tidak perlu ditampilkan (misalnya, fungsi yang tidak memiliki pernyataan return akan secara otomatis mengembalikan none).

```
>none
```

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

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

```
  1234.5
```

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

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

```
  affe
  charlie
  bravo
```

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

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

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

String dapat mengandung karakter Unicode. Secara internal, string ini menggunakan pengkodean UTF-8. Untuk membuat string seperti ini, gunakan u"..." dengan 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 yang sama seperti , dll. dapat digunakan. Ini mungkin menjadi alternatif cepat dibandingkan Latex. (Detail lebih lanjut tentang komentar akan dijelaskan di bawah).

Terdapat 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 yang berisi angka Unicode. Fungsi kebalikannya adalah chartoutf().

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

```
  Ü is a German letter
```

Fungsi utf() dapat menerjemahkan 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 =.
```

Selain itu, entitas numerik juga dapat digunakan.

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

```
  Ähnliches
```

Nilai Boolean

Nilai Boolean dalam Euler dinyatakan dengan 1 untuk true (benar) dan 0 untuk false (salah). String dapat dibandingkan seperti halnya angka.

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

```
  0
  1
```

Operator "and" ditulis sebagai "&&" dan operator "or" ditulis sebagai "||", seperti dalam bahasa C. (Kata "and" dan "or" hanya dapat digunakan dalam kondisi untuk pernyataan "if".)

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

```
  1
```

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]
```

Anda dapat menggunakan fungsi nonzeros() untuk mengambil elemen tertentu dari sebuah vektor. Dalam contoh, 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]
```

Format Output

Format output bawaan EMT menampilkan 12 digit. Untuk memastikan kita melihat format bawaan, kita reset formatnya.

```
>defformat; pi
```

```
  3.14159265359
```

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

```
>longest pi
```

```
        3.141592653589793
```

Berikut representasi heksadesimal internal dari bilangan double.

```
>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
```

Default-nya adalah format(12).

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

```
  0.333333333333
```

Fungsi seperti "shortestformat", "shortformat", dan "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 bawaan untuk skalar adalah format(12). Namun ini dapat diubah.

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

```
  3.1416
```

Fungsi "longestformat" juga mengatur format skalar.

```
>longestformat; pi
```

```
  3.141592653589793
```

Sebagai referensi, berikut daftar format output paling penting:

```
shortestformat shortformat longformat, longestformat
format(panjang,digit) goodformat(panjang)
fracformat(panjang)
defformat
```

Akurasi internal EMT adalah sekitar 16 digit desimal, yang mengikuti standar IEEE. Angka disimpan dalam format internal ini.
Namun, format output EMT dapat diatur dengan cara yang fleksibel.

```
>longestformat; pi,
```

```
  3.141592653589793
```

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

```
    3.14159
```

Default-nya adalah defformat().

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

Terdapat operator singkat yang hanya menampilkan satu nilai. Operator "longest" akan menampilkan semua digit valid dari suatu angka.

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

```
        4.934802200544679
```

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
```

Karena format internal menggunakan representasi biner untuk menyimpan angka, nilai 0.1 tidak akan direpresentasikan secara tepat. Kesalahan ini sedikit terakumulasi, seperti yang terlihat dalam komputasi berikut.

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

```
   -1.110223024625157e-16
```

Namun dengan format bawaan "longformat", Anda tidak akan menyadarinya. Untuk memudahkan, output dari angka yang sangat kecil ditampilkan sebagai 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 ingin menggunakan string sebagai ekspresi, gunakan konvensi penamaan seperti "fx" atau "fxy", dll. Ekspresi memiliki prioritas lebih tinggi 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 sesuai urutan tersebut. Parameter tambahan dapat ditambahkan menggunakan parameter yang ditetapkan.

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

```
  -0.919535764538
```

Perlu diperhatikan bahwa ekspresi akan selalu menggunakan variabel global, meskipun terdapat variabel dengan nama yang sama dalam suatu fungsi. (Jika tidak, evaluasi ekspresi dalam fungsi dapat menghasilkan hal yang 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 ingin menggunakan nilai lain untuk "at" selain nilai global, tambahkan "at=nilai".

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

```
  45
```

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

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

```
  45
```

Ekspresi yang menggunakan variabel x biasanya diperlakukan seperti fungsi.
Perlu diingat jika Anda membuat fungsi dengan nama yang sama dengan ekspresi simbolik global yang sudah ada, variabel ekspresi tersebut akan dihapus untuk mencegah kerancuan antara ekspresi simbolik dan fungsi.

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

```
  12
```

Sebagai konvensi, ekspresi simbolik atau numerik sebaiknya dinamai fx, fxy, dan sejenisnya. 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 apa pun digunakan sebagai parameter tanpa nama untuk evaluasi ekspresi, tidak hanya "x", "y", dll. Untuk ini, awali ekspresi dengan "@(variabel) ...".

```
>"@(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-fungsi EMT yang memerlukan ekspresi dalam "x".

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

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

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

```
  -0.475
```

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

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

```
  -0.425
```

Ekspresi tidak harus bersifat simbolis. Hal ini diperlukan jika ekspresi mengandung fungsi yang hanya dikenal dalam kernel numerik, bukan di Maxima.

Matematika Simbolis

EMT melakukan matematika simbolis dengan bantuan Maxima. Untuk detailnya, mulailah dengan tutorial berikut, atau jelajahi referensi untuk Maxima. Para ahli Maxima perlu memperhatikan bahwa terdapat perbedaan sintaks antara sintaks asli Maxima dan sintaks default ekspresi simbolis dalam EMT.

Matematika simbolis terintegrasi secara mulus ke dalam Euler dengan tanda &. Ekspresi apa pun yang dimulai dengan & adalah ekspresi simbolis. Ekspresi ini dievaluasi dan dicetak oleh Maxima.

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

```
>$&44!
```

$$265827157478844876804362581101461589031963852800000000$$

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

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

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

$$2481256778$$

Tentu saja, Maxima memiliki fungsi yang lebih efisien untuk ini (begitu juga dengan bagian numerik EMT).

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

$$2481256778$$

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

Anda akan mengetahui bahwa cara berikut juga dapat digunakan.

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

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

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

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

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

$$120$$

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

Ekspresi simbolis dicetak oleh Maxima dalam bentuk 2D. Alasan untuk ini adalah adanya penanda simbolis khusus dalam string.

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

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

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

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

Ekspresi simbolis diurai oleh Euler. Jika Anda membutuhkan sintaks yang kompleks dalam satu ekspresi, Anda dapat membungkus ekspresi tersebut dengan tanda "...". Menggunakan lebih dari satu ekspresi sederhana memang mungkin, tetapi sangat tidak disarankan.

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

25

Sebagai informasi lengkap, kami mencatat bahwa ekspresi simbolis dapat digunakan dalam program, tetapi perlu diapit dengan tanda kutip. Selain itu, akan jauh lebih efektif untuk memanggil Maxima pada saat compile jika memungkinkan.

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

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

Sekali lagi, % merujuk pada hasil sebelumnya.

Untuk mempermudah, kita simpan solusinya ke dalam variabel simbolik. Variabel simbolik didefinisikan dengan "&=".

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

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

Ekspresi simbolis dapat digunakan dalam ekspresi simbolis.

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

$$x^2\ (x+3)\ e^x$$

Input langsung untuk 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 melakukan ini dengan ":::".

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

```
                            2
                           g
```

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

```
                          3   x
                         x   E
```

$$x^3\,e^x$$

Variabel seperti itu dapat digunakan dalam ekspresi simbolis lainnya. Perhatikan bahwa pada perintah berikut, sisi kanan &= dievaluasi terlebih dahulu sebelum ditetapkan ke Fx.

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

fx

$$fx$$

fx

```
>fx(5)
```

```
  Function fx not found.
  Try list ... to find functions!
  Error in:
  fx(5) ...
       ^
```

Untuk mengevaluasi 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(%)
```

0

0.0

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

$$0$$

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

```
>tex(fx)
```

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

Ekspresi simbolis dapat dievaluasi seperti halnya ekspresi numerik.

```
>fx(0.5)
```

```
  Function fx not found.
  Try list ... to find functions!
  Error in:
  fx(0.5) ...
         ^
```

Dalam ekspresi simbolis, cara ini tidak berlaku karena Maxima tidak mendukungnya. Sebagai gantinya, gunakan sintaks "with" (bentuk yang lebih baik dari perintah at(...) dalam Maxima).

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

$$fx$$

Penetapan nilai juga dapat bersifat simbolis.

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

$$fx$$

Perintah solve digunakan untuk menyelesaikan ekspresi simbolis untuk suatu variabel dalam 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 "solve" numerik dalam Euler, yang memerlukan nilai awal, dan secara opsional memerlukan nilai target.

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

```
  1.56155281281
```

Nilai numerik dari solusi simbolis dapat dihitung dengan mengevaluasi hasil simbolis tersebut. Euler akan membaca penetapan seperti x=, dll. Jika Anda tidak memerlukan hasil numerik untuk komputasi lebih lanjut, Anda juga dapat meminta Maxima untuk mencari nilai numeriknya.

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

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

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

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

Untuk mendapatkan solusi simbolis tertentu, Anda dapat menggunakan "with" dan sebuah 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 simbolis dapat memiliki flag, yang menunjukkan penanganan khusus dalam Maxima. Beberapa flag dapat digunakan sebagai perintah juga, yang lainnya tidak. Flag ditambahkan dengan "|" (bentuk yang lebih baik 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". Fungsi dapat berupa satu baris atau multi-baris.
Fungsi satu baris dapat bersifat numerik atau simbolik. Fungsi satu baris numerik didefinisikan dengan ":=".

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

Sebagai gambaran, kami tunjukkan semua definisi yang mungkin untuk fungsi satu baris. Fungsi dapat dievaluasi seperti fungsi bawaan Euler lainnya.

```
>f(2)
```

```
  4.472135955
```

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

```
>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 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
```

Secara bawaan, jika Anda perlu menimpa fungsi bawaan, Anda harus menambahkan kata kunci "overwrite".
Menimpa fungsi bawaan berisiko dan dapat menyebabkan masalah pada fungsi lain yang bergantung padanya.
Anda masih dapat memanggil fungsi bawaan sebagai "_..." jika itu adalah fungsi inti Euler.

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

```
  0.707106781187
```

Lebih baik kita hapus definisi ulang dari 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
```

Jika parameter ini dihilangkan, nilai default akan digunakan.

```
>f(4)
```

```
  16
```

Mengaturnya akan menimpa nilai default.

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

```
  80
```

Parameter yang ditetapkan juga akan menimpanya. Hal ini digunakan oleh banyak fungsi Euler seperti plot2d, plot3d.

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

```
  16
```

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

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

```
  24
```

Namun, parameter yang ditetapkan akan mengesampingkan nilai global.

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

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

```
  20
```

Fungsi simbolis didefinisikan dengan "&=". Fungsi ini didefinisikan dalam Euler dan Maxima, serta bekerja di kedua lingkungan. Ekspresi pendefinisinya diproses melalui Maxima sebelum didefinisikan.

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

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

Fungsi simbolis dapat digunakan dalam ekspresi simbolis.

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

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

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

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

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

```
  178.635099908
```

Fungsi simbolis dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

```
>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 dapat bekerja, karena Euler menggunakan ekspresi simbolis dalam fungsi g jika tidak menemukan variabel simbolis g, dan jika terdapat fungsi simbolis 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 &=, fungsi bersifat simbolis dan dapat digunakan dalam ekspresi simbolis lainnya.

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

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

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

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

yang tidak dapat dievaluasi secara simbolis.
Jika kita mendefinisikan ulang fungsi dengan kata kunci "map", fungsi dapat digunakan untuk vektor x. Secara internal, fungsi dipanggil untuk semua nilai x sekaligus, 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, juga memungkinkan untuk menggunakan parameter yang ditetapkan.

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

```
  2
```

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

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

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

Fungsi simbolis seperti ini dapat digunakan untuk variabel simbolis.
Namun, fungsi ini juga dapat digunakan untuk vektor numerik.

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

```
  17
```

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

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

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

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

$$0$$

Tapi, tentu saja fungsi tersebut dapat digunakan dalam ekspresi simbolis atau dalam definisi fungsi simbolis.

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

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

Untuk menyingkat:
- &= mendefinisikan fungsi simbolis,
- := mendefinisikan fungsi numerik,
- &&= mendefinisikan fungsi murni simbolis.

Menyelesaikan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolis.

Untuk menyelesaikan 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
```

Ini juga berlaku untuk ekspresi simbolis. Ambil contoh 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 polinomial bernilai 2. Dalam solve(), nilai target default y=0 dapat diubah dengan variabel yang ditetapkan.
Kita gunakan y=2 dan periksa dengan mengevaluasi polinomial pada hasil sebelumnya.

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

```
  0.966715594851
  2
```

Menyelesaikan ekspresi simbolis dalam bentuk simbolis mengembalikan daftar solusi. Kami menggunakan pemecah simbolik solve() yang disediakan oleh Maxima.

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

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

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

```
>longest sol()
```

```
      -0.6180339887498949       1.618033988749895
```

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

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

$$0$$

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

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

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

Fungsi f() dapat mengakses variabel global. Namun sering kali kita ingin menggunakan parameter lokal.

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

dengan a=3.

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

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

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

```
  2.54116291558
```

Ini juga berlaku untuk ekspresi. Namun, elemen daftar yang dinamai harus digunakan. (Informasi lebih lanjut tentang daftar dapat dilihat di tutorial tentang sintaks EMT).

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

```
  2.54116291558
```

Menyelesaikan Pertidaksamaan

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

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

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

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

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

```
>$&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 pembahasan mendetail tentang bahasa matriks Euler.

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

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

```
                1                2
                3                4
```

Produk matriks dilambangkan dengan titik.

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

```
                3
                4
```

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

```
  [3,  4]
```

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

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

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

```
               11
               25
```

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

```
                1                0
                0                1
```

Inti utama dari bahasa matriks adalah bahwa semua fungsi dan operator bekerja secara elemen-per-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 operand adalah vektor atau skalar, operand tersebut akan diperluas secara alami.

```
>2*A
```

```
               2              4
               6              8
```

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

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

```
               1              4
               3              8
```

Jika operand adalah vektor baris, operand tersebut diterapkan ke semua kolom dari A.

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

```
               2              6
               6             12
```

Perkalian ini dapat dibayangkan seolah-olah vektor baris v telah diduplikasi untuk membentuk matriks dengan ukuran yang sama seperti A.

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

```
                1              2
                1              2
```

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

```
                1              4
                3              8
```

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

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

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

Sekali lagi, 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 berisi 0 dan 1, di mana 1 berarti benar (true).

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

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

Dari vektor seperti itu, "nonzeros" memilih elemen-elemen yang bukan nol.
Dalam kasus 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-nilai yang sesuai dalam t.

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

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

Sebagai contoh, mari kita cari semua kuadrat dari angka 1 hingga 1000, yang bersisa 5 ketika dibagi 11 dan bersisa 3 ketika dibagi 13.

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

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

EMT tidak sepenuhnya efektif untuk komputasi integer karena secara internal menggunakan floating point presisi ganda. Namun, seringkali sangat berguna.
Kita dapat memeriksa bilangan prima. Mari kita cari tahu berapa banyak kuadrat ditambah 1 yang merupakan bilangan prima.

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

```
  112
```

Fungsi nonzeros() hanya bekerja untuk vektor. Untuk matriks, terdapat 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
```

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

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

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

Indeks-indeks ini dapat digunakan untuk mengatur elemen-elemen tersebut ke suatu nilai.

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

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

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

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

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

Dan juga memungkinkan untuk mendapatkan elemen-elemen tersebut dalam sebuah vektor.

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

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

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

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

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

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

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

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

Tentu saja, ini sama dengan fungsi max().

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

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

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

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

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

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

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

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

Demikian pula, kita dapat menggabungkan matriks secara berdampingan jika kedua matriks 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 jumlah barisnya tidak sama, matriks yang lebih pendek akan diisi dengan 0.

Ada pengecualian untuk aturan ini. Angka riil yang digabungkan dengan matriks akan digunakan sebagai kolom yang diisi dengan angka 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
```

Memungkinkan untuk membuat matriks dari vektor baris dan kolom.

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

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

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

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

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

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

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

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut.

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

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

Untuk vektor, terdapat 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 cara berikut.

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

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

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

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

```
        0.66566      0.831835
          0.977      0.544258
```

Berikut adalah fungsi berguna lainnya, yang mengubah susunan 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
```

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

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

Mari kita coba.

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

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

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]
```

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

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

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

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

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

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

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

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

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

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

```
>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 masalah jika memasukkan 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]
```

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

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

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

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

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

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

Perhatikan bahwa kita tidak mengubah matriks A. Kita mendapatkan matriks baru sebagai hasil dari setdiag().
Berikut adalah fungsi yang mengembalikan matriks tridiagonal.

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

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

Diagonal sebuah matriks juga dapat diekstraksi dari matriks. Untuk mendemonstrasikan ini, kita menyusun ulang 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 diagonalnya.

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

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

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

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

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

Vektorisasi

Hampir semua fungsi di Euler dapat menerima input berupa matriks dan vektor, asalkan sesuai konteks.

Contohnya, fungsi sqrt() mampu menghitung akar kuadrat untuk setiap elemen dalam vektor atau matriks secara bersamaan.

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

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

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

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

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

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

$$s = t^3 - t$$

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

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

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

Misalnya, vektor kolom dikalikan vektor baris akan diperluas menjadi matriks jika operator diterapkan. Pada contoh berikut, 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 "." dalam EMT.

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

```
  55
```

Secara bawaan, 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 1x1 dapat digunakan seperti bilangan real.

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

```
  5
  25
```

Untuk mentranspos matriks, kita menggunakan tanda apostrof.

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

```
             1
             2
             3
             4
```

Jadi kita dapat menghitung matriks A dikali vektor b.

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

```
            30
            70
```

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

```
>v'.v
```

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

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

```
>v.v'
```

```
  30
```

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

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

```
  30
```

Operator dan fungsi mengikuti bahasa matriks Euler.

Berikut ringkasan aturannya:

- Fungsi yang diterapkan pada vektor atau matriks akan diproses untuk setiap elemen.
- Operator yang bekerja pada dua matriks berukuran sama akan diproses berpasangan untuk setiap elemen matriks.
- Jika dua matriks memiliki dimensi berbeda, keduanya akan diperluas secara logis agar memiliki ukuran yang sama.

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

Berikut adalah contoh sederhana dengan operator ^.

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

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

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

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

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

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan *!

```
>v.v'
```

```
  14
```

Ada banyak fungsi untuk matriks. Kami berikan daftar singkat. Anda sebaiknya merujuk dokumentasi untuk informasi lebih lanjut tentang perintah-perintah ini.

```
sum, prod menghitung jumlah dan produk dari baris
cumsum, cumprod melakukan hal yang sama secara kumulatif
extrema menghitung nilai ekstrem setiap baris dan 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]
```

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

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

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

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

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

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

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

```
  6
```

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

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

```
  6
  Row index 3 out of bounds!
  Error in:
  v:=[2,4,6,8]; v[3], A[3] ...
                          ^
```

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

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

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

Bentuk singkat untuk : adalah dengan menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

```
  4
```

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

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

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

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

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

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

Sekarang kita menambahkan kolom ke sebuah matriks.

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

```
  Variable or function w not found.
  Error in:
  M = deg(w)|w|cos(w)|sin(w) ...
           ^
```

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

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

Fungsi yang tidak bekerja untuk input vektor perlu "divektorisasi". Hal ini dapat dilakukan dengan kata kunci "map" dalam definisi fungsi. Dengan demikian, fungsi akan dievaluasi untuk setiap elemen dari parameter vektor.

Integrasi numerik integrate() hanya bekerja untuk batas interval skalar. Jadi kita perlu melakukan vektorisasi.

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

Kata kunci "map" melakukan vektorisasi pada fungsi. Fungsi sekarang akan bekerja untuk vektor angka.

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

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

Sub-Matriks dan Elemen Matriks

Untuk mengakses elemen matriks, gunakan notasi kurung 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 seluruh baris dari sebuah matriks.

```
>A[2]
```

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

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

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

```
  2
```

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

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

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

Jika indeks adalah vektor berisi beberapa indeks, Euler akan mengembalikan baris-baris matriks yang sesuai.
Di sini, kita ingin baris pertama dan kedua dari A.

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

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

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

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

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

Trik indeks juga bekerja dengan kolom.
Contoh ini memilih semua baris dari A dan kolom kedua serta ketiga.

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

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

Untuk menyingkat, ":" 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 dapat mengambil baris terakhir dari A.

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

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

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

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

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

Kita juga dapat menetapkan nilai ke suatu baris dari A.

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

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

Kita bahkan dapat menetapkan nilai ke sub-matriks jika ukurannya sesuai.

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

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

Selain itu, beberapa jalan pintas diperbolehkan.

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

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

Perhatian: Jika indeks yang digunakan berada di luar ukuran matriks, hasilnya bisa berupa matriks kosong atau pesan error, tergantung pengaturan sistem. Secara default, EMT akan menampilkan pesan error. Namun, indeks negatif dapat digunakan untuk mengakses elemen dari urutan terakhir (misalnya -1 untuk elemen terakhir, -2 untuk elemen kedua terakhir, dan seterusnya).

```
>A[4]
```

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

Pengurutan dan Pengacakan

Fungsi sort() mengurutkan vektor baris.

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

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

Sering kali diperlukan untuk mengetahui indeks dari vektor yang 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)
```

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

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]
```

Ini juga berlaku untuk vektor string.

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

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

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

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

Seperti yang Anda lihat, posisi entri ganda agak acak.

```
>ind
```

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

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

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

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

Ini juga berlaku untuk vektor string.

```
>unique(s)
```

```
  a
  aa
  d
  e
```

Aljabar Linear

EMT memiliki banyak fungsi untuk menyelesaikan sistem linear, sistem sparse, atau masalah regresi.

Untuk sistem linear Ax=b, Anda dapat menggunakan algoritma Gauss, matriks invers, atau linear fit. Operator A\b menggunakan versi algoritma Gauss.

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

```
      -4
     4.5
```

Untuk contoh lain, kita menghasilkan matriks 200x200 dan jumlah baris-barisnya. Kemudian kita menyelesaikan Ax=b menggunakan matriks invers. Kita mengukur error sebagai deviasi maksimal semua elemen dari 1, yang tentu saja adalah solusi yang benar.

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

```
    3.140820936664568e-13
```

Jika sistem tidak memiliki solusi, sebuah linear fit meminimalkan norma error 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 Simbolis

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

```
>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 simbolis, matriks ini dapat digunakan dalam ekspresi simbolis 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 yang berisi dua vektor nilai eigen dan multiplisitasnya.

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

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

Untuk mengambil eigenvector tertentu, diperlukan pengindeksan yang hati-hati.

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

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

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

Matriks simbolis dapat dievaluasi secara numerik di Euler seperti ekspresi simbolis lainnya.

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

```
                1                4
                5                2
```

Dalam ekspresi simbolis, gunakan with.

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

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

Akses ke baris matriks simbolis bekerja sama seperti pada matriks numerik.

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

$$[1,a]$$

Ekspresi simbolis dapat mengandung penetapan nilai. Dan hal itu mengubah matriks A.

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

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

Terdapat fungsi simbolis dalam Maxima untuk membuat vektor dan matriks. Untuk ini, silakan merujuk ke dokumentasi Maxima atau tutorial tentang Maxima dalam 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 melakukan upaya lebih besar dan mendapatkan hasil yang lebih akurat.

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

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

```
                1                 0
                0                 1
```

Misalnya, nilai eigen dari A dapat dihitung secara numerik.

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

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

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

```
>$&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 Simbolis

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

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

```
                1        3.14159
                4              5
```

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

```
>$&A
```

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

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

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

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

Maxima juga dapat menghitung dengan bilangan floating point (desimal), dan bahkan dengan bilangan floating besar dengan 32 digit. Namun, evaluasinya jauh lebih lambat.

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

$$1.414213562373095$$

Presisi bilangan floating point besar dapat diubah.

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

```
          3.14159265358979323846264338327950288419716939937510582097494\
  459230781640628620899862803482534211706 8b0
```

Variabel numerik dapat digunakan dalam ekspresi simbolis apa pun dengan "@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, kami menggunakan Euler Math Toolbox (EMT) untuk menghitung suku bunga. Kami melakukannya secara numerik dan simbolis untuk menunjukkan bagaimana Euler dapat digunakan untuk memecahkan masalah kehidupan nyata.
Misalkan Anda memiliki modal awal sebesar 5000 (misalnya dalam dolar).

```
>K=5000
```

```
  5000
```

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

```
>K*1.03
```

```
  5150
```

Euler juga akan memahami sintaks berikut.

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

```
  5150
```

Lebih mudah jika menggunakan faktor

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

```
  1.03
  5150
```

Untuk 10 tahun, kita dapat mengalikan faktor-faktornya dan mendapatkan nilai akhir dengan bunga majemuk.

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

```
  6719.58189672
```

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

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

```
      6719.58
```

Mari kita cetak hasil 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 hingga tahun 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menggunakan 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 ke vektor elemen demi elemen. Jadi

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

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

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

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

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

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

Mari kita bandingkan kedua hasilnya, 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 melakukan perulangan untuk setiap tahun. Euler menyediakan banyak solusi untuk ini.

Cara termudah adalah dengan fungsi iterate, yang mengiterasi 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 lebih mudah dipahami, menggunakan format kita dengan tempat desimal tetap.

```
>VKr'
```

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

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

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

```
      5150.00
      5000.00      5150.00      5304.50
```

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

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

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

```
      6719.60
      6719.58
```

Perbedaannya sangat kecil.

Memecahkan Persamaan

Sekarang kita ambil fungsi yang lebih advanced, yang menambahkan sejumlah uang tertentu setiap tahun.

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

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

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

```
  Real 1 x 11 matrix

     5000.00     5350.00     5710.50     6081.82      ...
```

Bagaimana jika kita menarik 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 menarik 200, kita kehilangan uang setiap tahun.

Bagaimana kita dapat menentukan jumlah tahun uang akan bertahan? Kita harus menulis perulangan untuk ini. Cara termudah adalah dengan mengiterasi 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
```

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

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

Fungsi iterate() memiliki satu trik lagi. Fungsi ini dapat menerima 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 jawab pertanyaan yang lebih ambigu. Misalkan kita tahu bahwa nilainya menjadi 0 setelah 50 tahun. Berapa tingkat bunga yang sesuai?

Ini adalah pertanyaan yang hanya dapat dijawab secara numerik. Di bawah ini, kita akan menurunkan rumus yang diperlukan. Kemudian Anda akan melihat bahwa tidak ada rumus sederhana untuk tingkat bunga. Namun untuk sekarang, kita menargetkan 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 tidak lagi menggunakan nilai global R dalam ekspresi kita. Fungsi seperti iterate() memiliki trik khusus dalam Euler. Anda dapat meneruskan nilai variabel dalam ekspresi sebagai parameter titik koma. Dalam hal ini P dan R.

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

Mari kita coba tes.

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

```
       -19.83
```

SEkarang kita dapat menyelesaikan permasalahan kita.

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

```
         3.15
```

Fungsi solve() digunakan untuk mencari nilai variabel x yang membuat ekspresi matematika sama dengan 0. Jawabannya adalah 3,15% per tahun. Kita gunakan 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 tarik per tahun agar modal awal habis setelah 20 tahun dengan asumsi suku bunga 3% per tahun.

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

```
-336.08
```

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

Solusi Simbolis untuk Masalah Suku Bunga

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

```
>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 melihat sebuah pola. Setelah n periode kita memiliki

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

Rumus tersebut adalah rumus untuk deret geometri, yang telah dikenal oleh Maxima.

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

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

Ini agak rumit. Jumlah tersebut dievaluasi dengan flag "simpsum" 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)
```

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

Fungsi ini melakukan hal yang sama seperti fungsi f kita sebelumnya. Namun 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? Tebakan awal kita adalah 30 tahun.

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

```
        20.51
```

Jawaban ini mengatakan bahwa nilainya akan negatif setelah 21 tahun.

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

Misalkan kita mendapatkan pinjaman sebesar K, dan membayar n pembayaran sebesar R (dimulai setelah tahun pertama) menyisakan hutang residual sebesar Kn (pada saat pembayaran terakhir). Rumus untuk ini jelas

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

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

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 untuk tingkat R secara simbolis.

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

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

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

Tentu saja, kita memiliki batasan 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 mengembalikan 10 angsuran sebesar 500.

Persamaan juga dapat diselesaikan untuk n. Ini terlihat lebih baik jika kita menerapkan beberapa penyederhanaan.

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

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

Latihan Soal

1. Tuliskan bentuk paling sederhana dari ekspresi berikut dengan cara mengalikan dan menyederhanakan hasil perkalian:

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

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

$$4\,x^2 - 9\,y^2$$

2. Ubah ekspresi berikut ke dalam bentuk hasil pemfaktoran dengan mengeluarkan faktor persekutuan terbesar:

$$24x^3 - 36x^2 + 72x - 108$$

```
>$factor(24*x^3-36*x^2+72*x-108)
```

$$12\,\left(2\,x - 3\right)\,\left(x^2 + 3\right)$$

3. Sederhanakan hasil pembagian dua suku aljabar berikut:

$$\frac{12x^2y^2}{20x^3y}$$

```
>$((12*x^2*y^2)/(20*x^3*y))
```

$$\frac{3\,y}{5\,x}$$

4. Tentukan hasil dari ekspresi berikut dengan menggunakan sifat eksponen dan tuliskan hasil akhirnya tanpa eksponen negatif:

$$\frac{x^{-3}y^4}{x^{-1}y^{-2}}$$

```
>$((x^(-3)*y^4)/(x^(-1)*y^(-2)))
```

$$\frac{y^6}{x^2}$$

5. Faktorkanlah ekspresi berikut dengan benar-benar lengkap menggunakan pemfaktoran kelompok:

$$3x^3 - x^2 + 18x - 6$$

```
>$factor(3*x^3-x^2+18*x-6)
```

$$(3\,x - 1)\,\left(x^2 + 6\right)$$

6. Diperkirakan terdapat 51,2 miliar potongan sampah di sepanjang 76 juta mil jalan raya di Amerika Serikat pada suatu tahun. Tentukan rata-rata jumlah potongan sampah per mil jalan raya. Bulatkan jawaban ke bilangan bulat terdekat.

$$\text{Rata-rata} = \frac{51.2 \times 10^9 \text{ pcs}}{76 \times 10^6 \text{ mil}}$$

```
>$round((51.2*10^9)/(76*10^6))
```

$$674$$

7. Proyek pembangunan Chesapeake Bay Bridge-Tunnel memiliki panjang 17,6 mil dengan biaya pembangunan sebesar 210 juta dolar pada saat penyelesaiannya. Hitung rata-rata biaya pembangunan per mil. Nyatakan jawaban dalam satuan dolar (dibulatkan sampai dua angka di belakang koma) dan juga dalam notasi ilmiah.

$$\text{Biaya per mil} = \frac{210{,}000{,}000}{17.6}\ \text{(dolar per mil)}$$

```
>$round(210000000/17.6)
```

$$11931818$$

8. Bintang terdekat, Alpha Centauri C, berjarak sekitar 4,22 tahun cahaya dari Bumi. Satu tahun cahaya setara dengan 5,88*10^12 mil. Tentukan jarak dari Bumi ke Alpha Centauri C dalam mil, dan nyatakan jawaban dalam notasi ilmiah.

$$\text{Jarak} = 4.22 \times (5.88 \times 10^{12}) \text{ mil}$$

```
>$4.22*5.88e12
```

$$2.48136 \times 10^{+13}$$

9. Satu gram radium menghasilkan sekitar 37 miliar disintegrasi setiap detik. Tentukan berapa banyak disintegrasi yang terjadi dalam 1 jam. Nyatakan jawaban dalam notasi ilmiah.

$$\text{Disintegrasi per jam} = 37 \times 10^{9} \times 3600$$

```
>$3.26*5.88e12
```

$$1.91688 \times 10^{+13}$$

10. Jarak rata-rata Bumi ke Matahari adalah sekitar 93 juta mil. Dengan menganggap orbit Bumi berbentuk lingkaran, tentukan panjang lintasan yang ditempuh Bumi dalam satu tahun (satu kali revolusi penuh mengelilingi Matahari). Nyatakan hasil dalam juta mil (dibulatkan hingga dua angka di belakang koma) dan dalam notasi ilmiah.

$$\text{Panjang orbit} = 2\pi r = 2\pi(93 \times 10^{6}) \text{ mil}$$

```
>$37e9*3600
```

$$1.332 \times 10^{+14}$$

11. Hitung jumlah dua bilangan kompleks berikut dan nyatakan hasilnya dalam bentuk standar a+bi.

$$(7 - 3i) + (2 + 5i)$$

```
>$((7-3*i)+(2+5*i))
```

$$2\,i + 9$$

12. Diketahui polinom bereal-koefisien memiliki akar 1+i dan 2. Tentukan polinom berderajat terendah (koefisien real) yang memenuhi syarat tersebut, dan tuliskan hasil faktornya.

```
>$expand((x-(1+i))*(x-(1-i))*(x-2))
```

$$x^3 - 4\,x^2 - i^2\,x + 5\,x + 2\,i^2 - 2$$

13. Selesaikan persamaan kuadrat berikut, lalu nyatakan semua akarnya dalam bentuk a+bi:

$$x^2 + 6x + 25 = 0$$

```
>$solve(x^2+6*x+25=0,x)
```

$$[x = -4\,i - 3, x = 4\,i - 3]$$

14. Tentukan semua akar kuadrat dari bilangan kompleks -9.

```
>$solve(w^2+9=0,w)
```

$$[w = -3\,i, w = 3\,i]$$

15. Carilah nilai C

$$F = \frac{9}{5}C + 32$$

```
>$solve(F=(9/5)*C+32, C)
```

$$\left[C = \frac{5\,F - 160}{9}\right]$$