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))))
```

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

Latihan : Menyederhanakan Bentuk Aljabar

1. Sederhanakan bentuk eksponen berikut

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

Penyelesaian :

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

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

2. Sederhanakan bentuk eksponen berikut

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

Penyelesaian :

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

$$\frac{q^{80}}{625\,p^{16}\,r^{148}}$$

3. Sederhanakan bentuk eksponen berikut

$$19x^5y^3 \times -8x^2y^5$$

Penyelesaian :

```
>$&19*x^5*y^3*(-8)*x^2*y^5
```

$$-152\,x^7\,y^8$$

4. Sederhanakan bentuk eksponen berikut

$$3x^{-2}y^4 \times 8x^3y^5 \times 4x^3y^3$$

Penyelesaian :

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

$$96\,x^4\,y^{12}$$

5. Sederhanakan bentuk eksponen berikut

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

Penyelesaian :

```
>$&(8*a*b^7)*(-7*a^(-5)*b^2)
```

$$-\frac{56\,b^9}{a^4}$$

Baris Perintah

Baris perintah Euler terdiri dari satu atau beberapa perintah Euler diikuti dengan titik koma ";" atau koma ",". Titik koma mencegah pencetakan hasilnya. Koma setelah perintah terakhir dapat dihilangkan.

Baris perintah berikut hanya akan mencetak hasil 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 dijalankan sesuai 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 jalur dihubungkan dengan "..." kedua jalur 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 dua baris atau lebih. Anda dapat menekan Ctrl+Return untuk membagi baris menjadi dua pada posisi kursor saat ini, atau Ctlr+Back untuk menggabungkan baris.

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

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

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

```
81
```

Euler mendukung loop di baris perintah, asalkan cocok ke dalam satu baris atau multi-baris. Tentu saja, pembatasan ini tidak berlaku dalam program. Untuk informasi lebih lanjut lihat pendahuluan berikut.

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

```
1.5
1.41666666667
1.41421568627
1.41421356237
1.41421356237
```

Tidak apa-apa menggunakan 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 bersyarat juga berfungsi.

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

```
Thought so!
```

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

Saat Anda menggerakkan kursor di sepanjang garis, pasangan tanda kurung atau tanda kurung pembuka dan penutup akan disorot. Juga, perhatikan baris status. Setelah tanda kurung buka dari fungsi sqrt(), baris status akan menampilkan teks bantuan untuk fungsi tersebut. Jalankan perintah dengan kunci kembali.

```
>sqrt(sin(10°)/cos(20°))
```

```
0.429875017772
```

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

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

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

```
2.5
```

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

Latihan : Baris Perintah

1.

```
>sqrt(sin(180°)/cos(45°))
```

```
1.31602131916e-08
```

2. Hitunglah nilai

```
>r:=6; h:=5; pi*r^2*h/4
```

```
141.371669412
```

3. Hitunglah nilai

```
>x := 2;
>x := cos(x)
```

```
-0.416146836547
```

4. Hitunglah nilai

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

```
1.5
1.41666666667
1.41421568627
```

5. Hitunglah nilai

```
>pi*3*r*h, %+2*pi*r*h
```

```
282.743338823
471.238898038
```

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

Untuk menyetel variabel, gunakan "=" atau ":=". Demi kejelasan, pendahuluan ini menggunakan bentuk yang terakhir. Spasi tidak penting. Tapi jarak antar perintah diharapkan.

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

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

```
30.65625
```

EMT menggunakan sintaks pemrograman untuk ekspresi. Untuk masuk

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

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

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

```
8.77908249441
```

Untuk menghitung ekspresi rumit seperti

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

Anda harus memasukkannya dalam formulir 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 yang mengakhiri tanda kurung tutup. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi.

Hasil perhitungan ini berupa bilangan floating point. Ini secara default dicetak dengan akurasi sekitar 12 digit. Di baris perintah berikut, kita juga mempelajari bagaimana kita bisa merujuk ke hasil sebelumnya dalam baris yang sama.

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

```
0.47619047619
10/21
```

Perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi terbuat dari operator dan fungsi. Jika perlu, harus berisi tanda kurung untuk memaksakan urutan eksekusi yang benar. Jika ragu, memasang braket adalah ide yang bagus. Perhatikan bahwa EMT menampilkan tanda kurung buka dan tutup saat mengedit baris perintah.

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

```
14.4978445072
```

Operator numerik Euler meliputi

```
+ unary atau operator plus
- unary atau operator minus
*, /
. produk matriks
a^b pangkat untuk a positif atau bilangan bulat b (a**b juga
```

berfungsi)

```
N! operator faktorial
```

dan masih banyak lagi.

Berikut beberapa fungsi yang mungkin Anda perlukan. Masih banyak lagi.

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

Beberapa perintah memiliki alias, e.g. ln for log.

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

```
2
0.5
```

```
>sin(30°)
```

```
0.5
```

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

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

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

Latihan Sintaks Dasar

1. Hitunglah nilai

$$e^3 \cdot \left(\frac{1}{2+5\log(0.8)} + \frac{4}{9}\right)$$

```
>E^3*(1/(2+5*log(0.8))+4/9)
```

```
31.6408488081
```

2. Hitunglah nilai

$$\left(\frac{\frac{1}{6}+\frac{2}{9}+6}{\frac{2}{3}+\frac{7}{8}}\right)^4 \pi$$

```
>((1/6 + 2/9 + 6) / (2/3 + 7/8))^4 * pi
```

```
926.593574561
```

3. Hitunglah nilai

```
>9/14+3/10, fraction %
```

```
0.942857142857
33/35
```

4. Hitunglah nilai

```
>(cos(pi/5)+2)^4*(sin(pi/2)+3)^3
```

```
3984.71661458
```

5. Hitunglah nilai

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

```
2
0.447052980583
```

Tipe data primer pada Euler adalah bilangan real. Real 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
```

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

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

Latihan : Bilangan Real

1. Jabarkan pecahan berikut dalam digit desimal

```
>longest 1/4
```

```
0.25
```

2. Jabarkan pecahan berikut dalam digit desimal

```
>printdual (1/4)
```

```
1.0000000000000000000000000000000000000000000000000000*2^-2
```

3. Jabarkan pecahan berikut dalam digit desimal

```
>printhex(1/4)
```

```
4.0000000000000*16^-1
```

4. Jabarkan pecahan berikut dalam digit desimal

```
>longest(1/9)
```

```
0.1111111111111111
```

5. Jabarkan pecahan berikut dalam digit desimal

```
>longest(1/21)
```

```
0.04761904761904762
```

Strings

Sebuah string di Euler didefinisikan dengan "...".

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

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

String dapat digabungkan dengan | atau dengan +. Ini juga berfungsi dengan angka, yang dalam hal ini diubah menjadi string.

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

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

Fungsi print juga mengubah angka menjadi string. Ini dapat memerlukan sejumlah digit dan sejumlah tempat (0 untuk keluaran padat), dan optimalnya satuan.

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

```
Golden Ratio : 1.61803
```

Ada string khusus none yang tidak dicetak. Itu dikembalikan oleh beberapa fungsi, ketika hasilnya tidak menjadi masalah. (Ini dikembalikan secara otomatis, jika fungsi tidak memiliki pernyataan return.)

```
>none
```

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

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

```
1234.5
```

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

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

```
affe
charlie
bravo
```

Vektor string kosong dilambangkan dengan [tidak ada]. 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

Di komentar, entitas yang sama seperti , dll. dapat digunakan. Ini mungkin merupakan alternatif cepat untuk Lateks. (Detail lebih lanjut di komentar di bawah).

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

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

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

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

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

```
Ü is a German letter
```

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

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

```
We have =.
```

Dimungkinkan juga untuk menggunakan entitas numerik.

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

```
Ähnliches
```

Latihan Strings

1.

Nilai Boolean

Nilai Boolean diwakili dengan 1=true atau 0=false di 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 ini, kita menggunakan kondisi isprime(n).

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

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

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

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

Latihan : Nilai Boolean

1. Tentukan value dari kalimat berikut.

2<19

```
>2023<2019
```

```
0
```

2. Tentukan value dari anggota himpunan [1:8]>3 dan mana saja yang bernilai benar?

```
>(1:8)>3, nonzeros(%)
```

```
[0,  0,  0,  1,  1,  1,  1,  1]
[4,  5,  6,  7,  8]
```

3. Jika x > 12 maka x < 11. Apakah value dari peryataan tersebut?

```
>x > 12 && x < 11
```

```
0
```

4. Tentukan value dari kalimat berikut.

"komputer"<"handphone"

```
>"komputer"<"handphone"
```

```
0
```

5. Jabarkan bilangan genap dari 1 sampai 100

```
>N=2:2:100
```

```
[2,  4,  6,  8,  10,  12,  14,  16,  18,  20,  22,  24,  26,  28,  30,
32,  34,  36,  38,  40,  42,  44,  46,  48,  50,  52,  54,  56,  58,
60,  62,  64,  66,  68,  70,  72,  74,  76,  78,  80,  82,  84,  86,
88,  90,  92,  94,  96,  98,  100]
```

Format Keluaran

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

```
>defformat; pi
```

```
3.14159265359
```

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

```
>longest pi
```

```
        3.141592653589793
```

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

```
>printhex(pi)
```

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

Format keluaran dapat diubah secara permanen dengan perintah format.

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

```
0.33333
3.14159
0.84147
```

Standarnya adalah format(12).

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

```
0.333333333333
```

Funngsi seperti"shortestformat", "shortformat", "longformat" berfungsi untuk vektor dengan cara berikut.

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

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

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

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

```
3.1416
```

Fungsi "longestformat" juga mengatur format skalar.

```
>longestformat; pi
```

```
3.141592653589793
```

Sebagai referensi, berikut adalah daftar format keluaran terpenting.

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

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

Namun format keluaran 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 nomor yang valid.

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

```
4.934802200544679
```

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

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

```
25/12
```

Karena format internal menggunakan cara biner untuk menyimpan angka, nilai 0,1 tidak akan direpresentasikan secara tepat. Kesalahannya bertambah sedikit, seperti yang Anda lihat pada 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 "long format" default Anda tidak akan menyadarinya. Untuk kenyamanan, keluaran angka yang sangat kecil adalah 0.

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

```
0
```

Latihan : Format Keluaran

```
>fraction 1/3+1/8+1/4+1/5
```

```
109/120
```

```
>fraction 22/3+24/7
```

```
226/21
```

```
>fraction 100/5+100/4
```

```
45
```

```
>fraction 111111111/2+222222222/1
```

```
555555555/2
```

```
>13/13+14/14+15/15+16/16+17
```

```
21
```

Ekspresi

String atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang dapat dievaluasi dengan EMT. Untuk ini, gunakan tanda kurung setelah ekspresi. Jika Anda ingin 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 itu. 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, meskipun ada variabel dalam fungsi dengan nama yang sama. (Jika tidak, evaluasi ekspresi dalam fungsi dapat memberikan hasil yang sangat membingungkan bagi pengguna yang memanggil fungsi tersebut.)

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

```
36
```

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

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

```
45
```

Sebagai referensi, kami mencatat bahwa koleksi 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 akan menghapus variabel ini untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

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

```
12
```

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

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

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

Bentuk ekspresi khusus memungkinkan variabel apa pun sebagai parameter yang tidak disebutkan namanya untuk mengevaluasi ekspresi, bukan hanya "x", "y", dll. Untuk ini, mulailah ekspresi dengan "@(variabel) ...".

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

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

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

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

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

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

```
-0.475
```

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

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

```
-0.425
```

Latihan : Ekspresi

Selesaikan ekspresi-ekspresi berikut.

```
>r:=5; fx:="pi*r^3"; longest fx()
```

```
392.6990816987241
```

```
>fx:="p*sin(x)^2"; fx(7,p=-2)
```

```
-0.863262781792
```

```
>fx &= x^5-a*x;  ...
>a=2; fx(0.3)
```

```
-0.59757
```

```
>function f(x) := 9*x^2
>f(5)
```

```
225
```

Matematika Simbolik

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

Matematika simbolik diintegrasikan ke dalam Euler dengan &. Ekspresi apa pun yang dimulai dengan & adalah ekspresi simbolis. Itu dievaluasi dan dicetak oleh Maxima.

Pertama-tama, Maxima memiliki aritmatika "infinite" yang dapat menangani bilangan yang sangat besar.

```
>$&44!
```

$$2658271574788448768043625811014615890319638528000000000$$

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 hal ini (seperti halnya 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 fungsi tersebut. Sebagai contoh, coba klik dua kali pada "&binomial" di baris perintah sebelumnya. Ini akan membuka dokumentasi Maxima yang disediakan oleh pembuat program tersebut.

Anda akan mengetahui bahwa fungsi-fungsi 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 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 dari sebuah persamaan dalam persamaan lain.

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

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

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

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

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

Ekspresi simbolik diuraikan oleh Euler. Jika Anda membutuhkan sintaks yang kompleks dalam satu ekspresi, Anda dapat mengapit ekspresi dalam “...”. Menggunakan lebih dari satu ekspresi sederhana dimungkinkan, tetapi sangat tidak disarankan.

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

$$25$$

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

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

$$4\left(x+1\right)^{3}$$

Sekali lagi, % mengacu pada hasil sebelumnya.

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

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

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

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

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

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

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

```
2432902008176640000
```

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

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

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

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

Jika Anda adalah seorang ahli dalam Maxima, Anda mungkin ingin menggunakan sintaks asli Maxima. Anda dapat melakukan ini dengan ":::".

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

```
                                  2
                                 g
```

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

```
                                 3  x
                                x  E
```

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

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

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

```
                    5
125 E
```

$$125\,e^5$$

```
18551.64488782208
```

```
>fx(5)
```

```
18551.6448878
```

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

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

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

```
        10        5
1000 E   - 125 E
```

```
2.20079141499189e+7
```

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

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

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

```
>tex(fx)
```

```
x^3\,e^{x}
```

Ekspresi simbolik dapat dievaluasi seperti halnya ekspresi numerik.

```
>fx(0.5)
```

```
0.206090158838
```

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

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

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

Penugasan ini juga bisa bersifat simbolis.

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

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

Perintah solve menyelesaikan ekspresi simbolik untuk sebuah variabel di Maxima. Hasilnya adalah sebuah vektor solusi.

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

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

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

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

```
1.56155281281
```

Nilai numerik dari solusi simbolik dapat dihitung dengan evaluasi hasil simbolik. Euler akan membaca penugasan x= dst. Jika Anda tidak membutuhkan hasil numerik untuk perhitungan lebih lanjut, Anda juga bisa membiarkan Maxima menemukan nilai numeriknya.

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

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

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

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

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

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

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

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

Untuk menyelesaikan sistem persamaan, gunakan vektor persamaan. Hasilnya adalah vektor solusi 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 bendera, yang menunjukkan perlakuan khusus di Maxima. Beberapa flag dapat digunakan sebagai perintah juga, namun ada juga yang tidak. Bendera ditambahkan dengan "|" (bentuk yang lebih baik dari "ev(...,flags)")

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

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

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

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

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

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

Latihan : Matematika Simbolik

1. Berapa nilai dari 15!

```
>$&15!
```

$$1307674368000$$

2. Tentukan nilai dari

$$C(16,5) = \frac{16!}{11! \cdot 5!}$$

```
>$& 16!/(11!*5!)
```

$$4368$$

```
>$& diff((x^3-1)/(x+2),x) | ratsimp
```

$$\frac{2\,x^3 + 6\,x^2 + 1}{x^2 + 4\,x + 4}$$

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

$$\frac{2\,x^3+6\,x^2+1}{(x+2)^2}$$

```
>$(9+x)/(x^3+5)
```

$$\frac{x+9}{x^3+5}$$

Fungsi

Dalam EMT, fungsi adalah program yang ditentukan dengan perintah "function". Fungsi dapat berupa fungsi satu baris atau fungsi multiline.
ungsi satu baris dapat berupa numerik atau simbolik. Fungsi satu baris numerik didefinisikan dengan ":=".

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

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

```
>f(2)
```

```
4.472135955
```

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

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

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

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

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

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

```
0.786151377757
```

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

Anda masih dapat memanggil fungsi bawaan sebagai "_...", jika fungsi tersebut merupakan fungsi dalam inti Euler.

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

```
0.707106781187
```

Sebaiknya kita hilangkan definisi ulang tentang dosa ini.

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

```
0.707106781187
```

Latihan : Fungsi

1.

```
>sin(180)
```

```
-0.801152635734
```

```
>cos(180)
```

```
-0.598460069058
```

```
>tan(180)
```

```
1.33869021035
```

```
>sec(180)
```

```
-1.67095525951
```

```
>cosec(180)
```

```
-1.24820159779
```

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
```

Menetapkannya akan menimpa nilai default.

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

```
80
```

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

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

```
16
```

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

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

```
24
```

Tetapi parameter yang ditetapkan akan menggantikan nilai global.

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

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

```
20
```

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

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

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

Fungsi simbolis dapat digunakan dalam ekspresi simbolis.

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

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

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

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

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

```
178.635099908
```

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

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

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

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

```
0.703467422498
```

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

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

```
0.703467422498
```

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

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

```
>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 ini bersifat simbolis, dan dapat digunakan dalam ekspresi simbolis lainnya.

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

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

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

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

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

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

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

Fungsi dapat memiliki nilai default untuk parameter.

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

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

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

```
2
6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

```
2
```

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

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

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

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

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

```
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

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

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

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

Untuk meringkas

- &= mendefinisikan fungsi simbolik,

```
:= mendefinisikan fungsi numerik,
```

```
&&= mendefinisikan fungsi simbolik murni.
```

Latihan : Parameter Default

1.

```
>function f(x) &= 5*x^7-3*x^2; $&f(x)
```

$$5\,x^7-3\,x^2$$

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

$$\frac{5\,x^8}{8}-x^3$$

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

$$35\,x^6-6\,x$$

```
>function f(x) &= 6*x^5-9*x^3+5*x^8; $&f(x)
```

$$5\,x^8 + 6\,x^5 - 9\,x^3$$

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

$$\frac{5\,x^9}{9} + x^6 - \frac{9\,x^4}{4}$$

Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolik.

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

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

```
1.41421356237
```

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

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

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

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

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

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

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

```
>$&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.
ami menggunakan y=2 dan mengeceknya dengan mengevaluasi polinomial pada hasil sebelumnya.

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

```
0.966715594851
2
```

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

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

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

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

```
>longest sol()
```

```
-0.6180339887498949        1.618033988749895
```

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

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

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

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

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

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

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

dengan a = 3.

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

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

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

```
2.54116291558
```

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

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

```
2.54116291558
```

Latihan : Memecahkan Ekspresi

1.

```
>px &= 2*x^7+x^3-x^3-x; $&px
```

$$2\,x^7 - x$$

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

```
1.06277345116
2
```

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])
```

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

```
>$&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])
```

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

Latihan : Menyelesaikan Pertidaksamaan

1.

```
>&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 - 5>0],[x]) // x^2-5 > 0
```

$$[x^2 - 5 > 0]$$

```
>$&fourier_elim([x^2 - 5<0],[x]) // x^2-5 > 0
```

$$[5 - x^2 > 0]$$

```
>$&fourier_elim([x^2 - 5#0],[x]) // x^5-5 > 0
```

$$[5 - x^2 > 0] \vee [x^2 - 5 > 0]$$

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.

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

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

```
                1               2
                3               4
```

Hasil kali matriks dilambangkan dengan 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
```

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

```
>A.A
```

```
       7         10
      15         22
```

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

```
                1              4
                9             16
```

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

```
               37             54
               81            118
```

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

```
               37             54
               81            118
```

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

```
                1              1
                1              1
```

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

```
       0.333333      0.666667
           0.75             1
```

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

```
            -2
           2.5
```

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

```
            -2
           2.5
```

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

```
             1             0
             0             1
```

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

```
                1                0
                0                1
```

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

```
                1                4
                9               16
```

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

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

```
                9
               16
```

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

```
>2*A
```

```
                2                4
                6                8
```

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

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

```
                1                4
                3                8
```

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

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

```
                2                6
                6               12
```

Kita dapat membayangkan perkalian ini seolah-olah vektor baris v telah diduplikasi untuk membentuk matriks yang berukuran 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 dimana yang satu adalah vektor baris dan yang lainnya adalah vektor kolom. Kita menghitung i*j untuk i,j dari 1 sampai 5. Caranya adalah dengan 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, ingatlah bahwa ini bukan produk matriks!

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

```
55
```

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

```
55
```

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

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

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

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, dimana 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 tersebut, "nonzeros" memilih elemen bukan nol.
Dalam hal ini, kita mendapatkan indeks semua elemen 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 bilangan 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. Ia menggunakan floating point presisi ganda secara internal. Namun, seringkali hal ini sangat berguna.

Kita dapat memeriksa primalitasnya. Mari kita cari tahu, berapa banyak persegi ditambah 1 yang merupakan 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 nilai tertentu.

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

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

Fungsi mset() juga dapat mengatur elemen pada indeks ke entri 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 ekstrem, 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]
```

Namun 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]
```

1.

```
>X=[9,10,11;15,16,19;3,7,9]
```

```
              9            10            11
             15            16            19
              3             7             9
```

```
>inv(X)
```

```
      -0.203704      0.240741     -0.259259
        1.44444     -0.888889      0.111111
       -1.05556      0.611111      0.111111
```

```
>X'
```

```
              9            15             3
             10            16             7
             11            19             9
```

```
>Y=[3,3;4,4;5,5]
```

```
                3               3
                4               4
                5               5
```

```
>X.Y
```

```
              122             122
              204             204
               82              82
```

Fungsi Matriks Lainnya (Matriks Bangunan)

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

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

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

Demikian pula, kita dapat melampirkan matriks ke matriks lain secara berdampingan, jika keduanya mempunyai 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 diisi dengan 0.

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

```
>A|1
```

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

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

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

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

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

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

Tujuan utamanya adalah untuk menafsirkan ekspresi vektor untuk vektor kolom.

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

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

Untuk mendapatkan ukuran A, kita bisa 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
```

Masih banyak fungsi lain yang menghasilkan matriks.

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

```
              1              1
              1              1
```

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

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

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

Matriks bilangan acak juga dapat dihasilkan dengan acak (uniform distribution) atau normal (Gauß distribution).

```
>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 fungsi ini dan fungsi dup untuk menulis fungsi rep(), yang mengulangi vektor sebanyak n kali.

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

Mari kita uji.

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

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

Fungsi multdup() menduplikasi elemen 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 memiliki 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 elemen. Jika Anda ingin menghapus elemen, Anda perlu mencari elemennya terlebih dahulu. Fungsi indexof(v,x) dapat digunakan untuk mencari 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 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 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 kami tidak mengubah matriks A. Kami mendapatkan matriks baru sebagai hasil dari setdiag().

Berikut adalah fungsi yang mengembalikan matriks tri-diagonal.

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

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

Diagonal suatu matriks juga dapat diekstraksi dari matriks tersebut. Untuk mendemonstrasikannya, kami 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 menjaga agar vektor kolom d diterapkan pada matriks baris demi baris.

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

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

Latihan : Fungsi Matriks Lainnya (Matriks Bangunan)

1.

```
>y=1:4, y_y
```

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

```
>y'
```

```
            1
            2
            3
            4
```

```
>A=random(2,4); A|v'
```

```
      0.208566     0.220144     0.855399    0.0288546            2
      0.259286     0.181379     0.293642     0.791497            3
             0            0            0            0            5
             0            0            0            0            7
```

```
0          0          0          0          11
0          0          0          0          13
0          0          0          0          17
0          0          0          0          19
0          0          0          0          23
0          0          0          0          29
0          0          0          0          31
0          0          0          0          37
0          0          0          0          41
0          0          0          0          43
0          0          0          0          47
```

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

```
[0.208566,  0.181379]
```

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

```
         1 68770/65153 301871/73603 3147/22747
134287/93938         1 270427/167039 445869/102175
```

Vektorisasi

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

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

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

Jadi Anda dapat dengan mudah membuat tabel nilai. Ini adalah salah satu cara untuk memplot suatu 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.

Pada contoh berikut, kita menghasilkan vektor nilai t[i] dengan jarak 0,1 dari -1 hingga 1. Kemudian kita menghasilkan vektor nilai fungsi

$$s = t^3 - t$$

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

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

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

Misalnya, vektor kolom dikali vektor baris diperluas ke matriks, jika operator diterapkan. Berikut ini, v' adalah vektor yang dialihkan (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, ini sangat berbeda dengan perkalian matriks. Hasil kali matriks dilambangkan dengan titik "." di EMT.

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

```
55
```

Secara default, vektor baris dicetak dalam format ringkas.

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

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

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

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

```
5
25
```

Untuk mengubah urutan matriks kita menggunakan apostrof.

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

```
1
2
3
4
```

Jadi kita dapat menghitung matriks A dikalikan vektor b.

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

```
30
70
```

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

```
>v'.v
```

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

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

```
>v.v'
```

```
30
```

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

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

```
30
```

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ringkasan peraturannya.

- Suatu fungsi yang diterapkan pada vektor atau matriks diterapkan pada setiap elemen.

- Operator yang mengoperasikan dua matriks dengan ukuran yang sama diterapkan secara berpasangan pada elemen-elemen matriks.

- Jika kedua matriks mempunyai dimensi yang berbeda, keduanya diekspansi secara wajar sehingga mempunyai ukuran yang sama.

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

Berikut ini adalah kasus sederhana dengan operator ^.

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

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

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

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

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

Perhatikan bahwa perkalian skalar menggunakan perkalian matriks, bukan *!

```
>v.v'
```

```
14
```

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

```
sum,prod menghitung jumlah dan hasil kali baris
cumsum,cumprod melakukan hal yang sama secara kumulatif
menghitung nilai ekstrem setiap baris
extreme mengembalikan vektor dengan informasi ekstrem
diag(A,i) mengembalikan diagonal ke-i
setdiag(A,i,v) menyetel diagonal ke-i
id(n) matriks identitas
det(A) determinannya
charpoly(A) polinomial karakteristik
nilai eigen(A) nilai eigen
```

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

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

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

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

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

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

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

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

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

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

```
6
```

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

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

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

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

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

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

Bentuk kependekan dari : menghilangkan indeks sepenuhnya.

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

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

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

```
>A{4}
```

```
4
```

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 pada tabel, mari kita atur ulang ke format default, dan hitung tabel nilai sinus dan kosinus. Perhatikan bahwa sudut dinyatakan dalam radian secara default.

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

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

Kata kunci "peta" membuat vektorisasi fungsi tersebut. Fungsinya sekarang akan berfungsi untuk vektor bilangan.

```
>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 mendapatkan sebuah matriks, yang setiap barisnya merupakan tabel t^i untuk satu i. Artinya, matriks tersebut memiliki elemen

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

Fungsi yang tidak berfungsi untuk masukan vektor harus "divektorkan". Hal ini dapat dicapai dengan kata kunci "peta" dalam definisi fungsi. Kemudian fungsi tersebut akan dievaluasi untuk setiap elemen parameter vektor.

Integrasi numerik integral() hanya berfungsi untuk batas interval skalar. Jadi kita perlu membuat vektorisasinya.

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

Kata kunci "map" membuat vektorisasi fungsi tersebut. Fungsinya sekarang akan berfungsi untuk vektor bilangan.

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

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

```
>sqrt(2:10)
```

```
[1.41421,  1.73205,  2,  2.23607,  2.44949,  2.64575,  2.82843,  3,
3.16228]
```

```
>shortest (3:7)*(3:7)'
```

```
       9      12      15      18      21
      12      16      20      24      28
      15      20      25      30      35
      18      24      30      36      42
      21      28      35      42      49
```

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

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

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

```
9
```

```
>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]
```

Sub-Matriks dan Elemen Matriks

Untuk mengakses elemen matriks, gunakan notasi braket.

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

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

Kita dapat mengakses baris matriks secara 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 yang kosong.

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

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

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

Di sini kita menginginkan baris pertama dan kedua A.

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

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

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

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

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

Trik indeks juga berfungsi dengan kolom.

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

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

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

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

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

```
3
6
9
```

Alternatifnya, 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 suatu nilai. Ini sebenarnya mengubah matriks A yang disimpan.

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

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

Kita juga dapat memberikan nilai pada baris A.

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

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

Kita bahkan dapat menetapkan 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
```

Peringatan: Indeks di luar batas mengembalikan matriks kosong, atau pesan kesalahan, bergantung pada pengaturan sistem. Standarnya adalah pesan kesalahan. Namun perlu diingat bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks yang dihitung dari akhir.

```
>A[4]
```

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

Latihan : Sub Matriks dan Elemen Matriks

1.

```
>X=[11,12,13;14,15,16;17,18,19], X[2,2]
```

```
                11              12              13
                14              15              16
                17              18              19
  15
```

```
>v=11:13; v[2]
```

```
  12
```

```
>X[:,3]
```

```
                13
                16
                19
```

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

```
             12            13
             15            16
             18            19
```

```
>X[10]
```

```
  Row index 10 out of bounds!
  Error in:
  X[10] ...
       ^
```

Menyortir dan Mengacak

Fungsi sort() mengurutkan vektor baris.

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

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

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

Mari kita mengacak sebuah vektor.

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

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

Indeks berisi urutan v.

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

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

Ini juga berfungsi 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 unik mengembalikan daftar elemen unik vektor yang diurutkan.

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

```
[5,  1,  4,  2,  5,  8,  5,  2,  7,  10]
[1,  2,  4,  5,  7,  8,  10]
```

Ini juga berfungsi untuk vektor string.

```
>unique(s)
```

```
a
aa
d
e
```

Latihan : Menyortir dan Mengacak

1.

```
>sort([20,13,2,17,9,52])
```

```
[2,  9,  13,  17,  20,  52]
```

```
>v=shuffle(88:100)
```

```
[95,  93,  99,  97,  91,  98,  96,  94,  90,  88,  89,  100,  92]
```

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

```
[88,  89,  90,  91,  92,  93,  94,  95,  96,  97,  98,  99,  100]
```

```
>Q=["d","a","ll","a","da","la"]
```

```
d
a
ll
a
da
la
```

```
>unique(Q)
```

```
a
d
da
la
ll
```

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

Untuk sistem linier 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
```

Contoh lain, kita membuat matriks berukuran 200x200 dan jumlah baris-barisnya. Kemudian kita selesaikan Ax=b menggunakan matriks invers. Kami mengukur kesalahan sebagai deviasi maksimal semua elemen dari 1, yang tentu saja merupakan solusi yang tepat.

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

```
3.441691376337985e-13
```

Jika sistem tidak mempunyai 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
```

Penentu matriks ini adalah 0.

```
>det(A)
```

```
  0
```

1.

```
>Q=[11,12;13,14]; R=[15;16]; Q\R
```

```
              -9
             9.5
```

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

```
    5.155875726359227e-13
```

```
>Q=[11,12,13;14,15,16;17,18,19]
```

```
              11            12            13
              14            15            16
              17            18            19
```

```
>det(Q)
```

```
  0
```

Matriks Simbolik

Maxima memiliki matriks simbolik. Tentu saja Maxima dapat digunakan untuk permasalahan aljabar linier sederhana seperti itu. Kita dapat mendefinisikan matriks untuk Euler dan Maxima dengan &:=, lalu menggunakannya dalam ekspresi simbolik. Bentuk [...] yang 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 sebuah vektor dengan dua vektor nilai eigen dan multiplisitas.

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

$$\left[\left[a-1,a+1\right],\left[1,1\right]\right]$$

Untuk mengekstrak vektor eigen tertentu memerlukan 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 sama seperti ekspresi simbolik lainnya.

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

```
1              4
5              2
```

Dalam ekspresi simbolik, gunakan dengan.

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

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

Akses ke deretan matriks simbolik berfungsi sama seperti matriks numerik.

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

$$[1, a]$$

Ekspresi simbolis dapat berisi tugas. 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)
```

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

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

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

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

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

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

Perhatikan, bahwa dengan &:= matriks B telah didefinisikan sebagai simbolik dalam ekspresi simbolik dan 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 simbolis. Lihat tutorial tentang Maxima untuk detailnya.

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

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

Latihan : Matriks Simbolik

1.

```
>Q &= [d,2,2;1,d,1;3,3,d]; $Q
```

$$\begin{pmatrix} d & 2 & 2 \\ 1 & d & 1 \\ 3 & 3 & d \end{pmatrix}$$

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

$$d^3 - 11\,d + 12$$

```
>$&invert(Q) with d=0
```

$$\begin{pmatrix} -\frac{1}{4} & \frac{1}{2} & \frac{1}{6} \\ \frac{1}{4} & -\frac{1}{2} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{2} & -\frac{1}{6} \end{pmatrix}$$

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

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

```
>$&Q with [a=7,b=4]
```

$$\begin{pmatrix} 1 & 7 \\ 4 & 2 \end{pmatrix}$$

Nilai Numerik dalam Ekspresi simbolik

Ekspresi simbolis hanyalah string yang berisi ekspresi. Jika kita ingin mendefinisikan nilai untuk ekspresi simbolik dan ekspresi numerik, kita harus menggunakan "&:=".8

```
>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 real akan digunakan.

```
>$&A
```

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

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

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

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

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

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

$$1.414213562373095$$

Ketepatan angka floating point besar dapat diubah.

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

```
        3.14159265358979323846264338327950288419716939937510582097494\
 459230781640628620899862803482534211706 8b0
```

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

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

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

$$-5.424777960769379$$

Latihan : Nilai Numerik dalam Ekspresi Simbolik

1.

```
>Q &:= [9,pi;4,3]
```

```
                9       3.14159
                4             3
```

```
>$&Q
```

$$\begin{pmatrix} 9 & \frac{1146408}{364913} \\ 4 & 3 \end{pmatrix}$$

```
>$&bfloat(sqrt(3)), $&float(sqrt(3))
```

$$1.732050807568877$$

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

```
        3.14159265358979323846264338327950288419716939937510582097494\
459230781640628620899862803482534211706 8b0
```

Demo - Suku Bunga

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

Asumsikan Anda memiliki modal awal sebesar 5.000 (katakanlah dalam dolar).

```
>K=5000
```

```
5000
```

Sekarang kami mengasumsikan tingkat bunga 3% per tahun. Mari kita tambahkan satu tarif sederhana dan hitung hasilnya.

```
>K*1.03
```

```
5150
```

Euler juga akan memahami sintaks berikut.

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

```
5150
```

Namun lebih mudah menggunakan faktor tersebut

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

```
1.03
5150
```

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

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

```
6719.58189672
```

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

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

```
6719.58
```

Mari kita cetak yang dibulatkan menjadi 2 digit dalam satu 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 ke 9? Untuk ini, bahasa matriks Euler sangat membantu. Anda tidak perlu menulis satu perulangan, tetapi cukup masuk

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

```
Real 1 x 11 matrix

     5000.00     5150.00     5304.50     5463.64     ...
```

Bagaimana keajaiban ini terjadi? 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 di Euler dapat diterapkan pada vektor elemen demi elemen. Jadi

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

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

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

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

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

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

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

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

```
                  1271.61
                 1271.6071
```

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

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

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

```
Real 1 x 11 matrix

     5000.00     5150.00     5304.50     5463.64     ...
```

Kami dapat mencetaknya dengan cara yang ramah, menggunakan format kami dengan tempat desimal tetap.

```
>VKr'
```

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

Untuk mendapatkan elemen vektor tertentu, kami menggunakan 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. Ingatlah bahwa 1:3 menghasilkan vektor [1,2,3]. Mari kita bandingkan elemen terakhir dari nilai yang dibulatkan dengan nilai penuh.

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

```
6719.60
6719.58
```

Perbedaannya sangat kecil.

Latihan : Demo - Suku Bunga

1. Cobalah hitung suku bunga. Asumsikan anda memiliki modakl awal 3000(dalam dolar)

```
>K=3000
```

```
      3000.00
```

Asumsikan bunga awal 5% per tahun

```
>q=1+5%, K*q
```

```
         1.05
      3150.00
```

```
>K*q^20
```

```
      7959.89
```

Hasil yang didapatkan dari tahun 1 sampai tahun ke 20

```
>K*q^(0:20)
```

```
Real 1 x 21 matrix

    3000.00     3150.00     3307.50     3472.88     ...
```

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

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

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

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

```
Real 1 x 11 matrix

     5000.00     5450.00     5922.50     6418.63     ...
```

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

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

```
Real 1 x 11 matrix

     5000.00     5050.00     5102.50     5157.63     ...
```

Kami melihat uangnya berkurang. Jelasnya, jika kita hanya mendapat bunga sebesar 150 pada tahun pertama, namun menghapus 200, kita kehilangan uang setiap tahunnya.

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

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

```
Real 1 x 51 matrix

     5000.00     5050.00     5102.50     5157.63     ...
```

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

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

```
         0.00
```

Alasannya adalah bukan nol (VKR<0) mengembalikan vektor indeks i, dengan VKR[i]<0, dan min menghitung indeks minimal.

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

Fungsi iterate() memiliki satu trik lagi. Ini dapat mengambil kondisi akhir sebagai argumen. Kemudian akan mengembalikan nilai dan jumlah iterasi.

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

```
User interrupted!
Try "trace errors" to inspect local variables after errors.
iterate:
    end;
```

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

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

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

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

Iterasinya sama seperti di atas

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

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

Apalagi kami 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
```

Rutinitas penyelesaian menyelesaikan ekspresi=0 untuk variabel x. Jawabannya adalah 3,15% per tahun. Kami mengambil 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 awal habis setelah 20 tahun dengan asumsi tingkat bunga 3% per tahun.

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

```
-336.08
```

Solusi Simbolis Masalah Suku Bunga

Kita dapat menggunakan bagian simbolis dari Euler untuk mempelajari masalahnya. Pertama kita mendefinisikan fungsi onepay() kita secara simbolis.

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

$$R + q\,K$$

Sekarang kita dapat mengulanginya.

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

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

Kami melihat sebuah pola. Setelah n periode yang kita miliki

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

Rumusnya adalah rumus jumlah geometri yang 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 tanda "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)
```

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

Fungsinya sama dengan fungsi f kita sebelumnya. Tapi ini lebih efektif.

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

```
-19.82504734650985
-19.82504734652684
```

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

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

```
20.51
```

Jawaban ini mengatakan akan menjadi negatif setelah 21 tahun.

Kita juga dapat menggunakan sisi simbolis Euler untuk menghitung rumus 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). Rumusnya 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
```

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

Kita dapat menyelesaikan nilai R secara simbolis.

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

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

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

Tentu saja, kami memiliki batasan berikut.

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

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

Yang jelas tanpa bunga kita harus membayar kembali 10 tarif 500.

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

```
>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 : Penyelesaian Soal Aljabar

1. Sederhanakan bentuk eksponen berikut

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

Penyelesaian :

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

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

2. Sederhanakan bentuk eksponen berikut

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

Penyelesaian :

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

$$\frac{q^{80}}{625\,p^{16}\,r^{148}}$$

3. Hitunglah

$$\frac{4(8-6)^2-4\cdot 3+2\cdot 8}{3^1+19^0}$$

Penyelesaian :

```
>&(4*(8-6)^2-4*3+2*8)/(3^1+19^0)
```

```
5
```

4. Hitunglah

$$\frac{[4(8-6)^2+4](3-2\cdot 8)}{2^2(2^3+5)}$$

Penyelesaian :

```
>&((4*(8-6)^2+4)*(3-2*8))/(2^2*(2^3+5))
```

```
- 5
```

5. Hitunglah

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

Penyelesaian :

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

$$12\,x\,y - y + 7\,x^2 - 2\,x - 9$$

6. Hitunglah

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

Penyelesaian :

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

$$-5\,x - 19$$

7. Hitunglah

$$(8y^5)(9y)$$

Penyelesaian :

```
>$&(8*y^5)*(9*y)
```

$$72\,y^6$$

8. Hitunglah

$$(x+3)^2$$

Penyelesaian :

```
>$&showev('expand((x+3)^2))
```

$$\mathit{expand}\left((x+3)^2\right) = x^2 + 6\,x + 9$$

9. Hitunglah

$$(2x+3y+4)(2x+3y-4)$$

Penyelesaian :

```
>$&showev('expand((2*x+3*y+4)*(2*x+3*y-4)))
```

$$expand\left(\left(3\,y+2\,x-4\right)\,\left(3\,y+2\,x+4\right)\right)=9\,y^2+12\,x\,y+4\,x^2-16$$

10. Faktorkan trinomial berikut

$$t^2+8t+15$$

Penyelesaian :

```
>$&factor(t^2+8*t+15)
```

$$\left(t+3\right)\,\left(t+5\right)$$

11. Faktorkan binomial berikut

$$y^2+12y+27$$

Penyelesaian :

```
>$&factor(y^2+12*y+27)
```

$$(y+3)\ (y+9)$$

12. Faktorkan kuadrat berikut.

$$16x^2 - 9$$

Penyelesaian

```
>$&factor(16*x^2-9)
```

$$(4\,x-3)\ (4\,x+3)$$

13. Faktorkan kuadrat berikut.

$$y^2 - 18y + 81$$

Penyelesaian :

```
>$&factor(y^2-18*y+81)
```

$$(y-9)^2$$

14. Faktorkan persamaan berikut.

$$3z^3 - 24$$

Penyelesaian :

```
>$&factor(3*z^3-24)
```

$$3\,(z-2)\,\left(z^2+2\,z+4\right)$$

15. Selesaikan persamaan berikut.

$$x^2 + 5x = 0$$

Penyelesaian :

```
>$&solve(x^2+5*x=0)
```

$$[x = -5, x = 0]$$

16. Selesaikan persamaan berikut.

$$z^2 - 144$$

Penyelesaian :

```
>$&solve(z^2-144)
```

$$[z = -12, z = 12]$$

17. Selesaikan persamaan berikut.

$$y^2 + 25 = 10y$$

Penyelesaian :

```
>$&solve(y^2+25=10*y)
```

$$[y = 5]$$

18. Selesaikan persamaan berikut.

$$6x^2 - 7x = 10$$

Penyelesaian :

```
>$&solve(6*x^2-7*x=10)
```

$$\left[x = -\frac{5}{6}, x = 2\right]$$

19. Selesaikan persamaan berikut.

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

Penyelesaian :

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

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

20. Selesaikan persamaan berikut.

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

Penyelesaian :

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

$$\left[t = \frac{23}{66}\right]$$

Menggambar Grafik 2D dengan EMT

Notebook ini menjelaskan tentang cara menggambar berbagaikurva dan grafik 2D dengan software EMT. EMT menyediakan fungsi plot2d() untuk menggambar berbagai kurva dan grafik dua dimensi (2D).

Basic Plots

Ada fungsi plot yang sangat mendasar. Ada koordinat layar, yang selalu berkisar dari 0 hingga 1024 di setiap sumbu, tidak peduli apakah layarnya persegi atau tidak. Semut ada koordinat plot, yang dapat diatur dengan setplot(). Pemetaan antara koordinat tergantung pada jendela plot saat ini. Misalnya, shrinkwindow() default menyisakan ruang untuk label sumbu dan judul plot.

Dalam contoh, kita hanya menggambar beberapa garis acak dalam berbagai warna. Untuk detail tentang fungsi-fungsi ini, pelajari fungsi inti EMT.

```
>a=clg; // clear screen
```

```
Variable clg not found!
Error in:
a=clg; // clear screen ...
     ^
```

```
>window(0,0,1024,1024); // use all of the window
>setplot(0,1,0,1); // set plot coordinates
>hold on; // start overwrite mode
>n=100; X=random(n,2); Y=random(n,2);  // get random points
>colors=rgb(random(n),random(n),random(n)); // get random colors
>loop 1 to n; color(colors[#]); plot(X[#],Y[#]); end; // plot
>hold off; // end overwrite mode
>insimg; // insert to notebook
```

```
>reset;
```

Penting untuk menahan grafik, karena perintah plot() akan menghapus jendela plot.

Untuk menghapus semua yang kami lakukan, kami menggunakan reset().

Untuk menampilkan gambar hasil plot di layar notebook, perintah plot2d() dapat diakhiri dengan titik dua (:). Cara lain adalah perintah plot2d() diakhiri dengan titik koma (;), kemudian menggunakan perintah insimg() untuk menampilkan gambar hasil plot.

Untuk contoh lain, kita menggambar plot sebagai sisipan di plot lain. Ini dilakukan dengan menentukan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak menyediakan ruang untuk label sumbu di luar jendela plot. Kita harus menambahkan beberapa margin untuk ini sesuai kebutuhan. Perhatikan bahwa kita menyimpan dan memulihkan jendela penuh, dan menahan plot saat ini sementara kita memplot sisipan.

```
>plot2d("x^3-x");
>xw=200; yw=100; ww=300; hw=300;
>ow=window();
>window(xw,yw,xw+ww,yw+hw);
>hold on;
>barclear(xw-50,yw-10,ww+60,ww+60);
>plot2d("x^4-x",grid=6):
```

```
>hold off;
>window(ow);
```

Plot dengan beberapa angka dicapai dengan cara yang sama. Ada fungsi utilitas figure() untuk ini.

Plot default menggunakan jendela plot persegi. Anda dapat mengubahnya dengan fungsi aspect(). Jangan lupa untuk mengatur ulang aspek nanti. Anda juga dapat mengubah default ini di menu dengan "Set Aspect" ke rasio aspek tertentu atau ke ukuran jendela grafis saat ini.

Tetapi Anda dapat mengubahnya juga untuk satu plot. Untuk ini, ukuran area plot saat ini diubah, dan jendela diatur sehingga label memiliki ruang yang cukup.

```
>aspect(2); // rasio panjang dan lebar 2:1
>plot2d(["sin(x)","cos(x)"],0,2pi):
```

```
>aspect();
>reset;
```

Fungsi reset() mengembalikan default plot termasuk rasio aspek.
Plot 2D di Euler

EMT Math Toolbox memiliki plot dalam 2D, baik untuk data maupun fungsi. EMT menggunakan fungsi plot2d. Fungsi ini dapat memplot fungsi dan data.

Dimungkinkan untuk memplot di Maxima menggunakan Gnuplot atau di Python menggunakan Math Plot Lib.

Euler dapat memplot plot 2D dari

- ekspresi
- fungsi, variabel, atau kurva parameter,
- vektor nilai xy,
- awan titik di pesawat,
- kurva implisit dengan level atau area level.
- Fungsi kompleks

Gaya plot mencakup berbagai gaya untuk garis dan titik, plot batang, dan plot berarsir.
Plot Ekspresi atau Variabel

Satu ekspresi dalam "x" (misalnya "4*x^2") atau nama fungsi (misalnya "f") menghasilkan grafik fungsi.

Berikut adalah contoh paling dasar, yang menggunakan rentang default dan mengatur y-range yang tepat agar sesuai dengan plot fungsi.

Catatan: Jika Anda mengakhiri baris perintah dengan titik dua ":", plot akan dimasukkan ke dalam jendela teks. Jika tidak, tekan TAB untuk melihat plot jika jendela plot tertutup.

```
>plot2d("x^2"):
```

```
>aspect(1.5); plot2d("x^3-x"):
```

```
>a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30); // menampilkan gambar hasil plot setinggi 25 baris
```

Dari beberapa contoh sebelumnya Anda dapat melihat bahwa aslinya gambar plot menggunakan sumbu X dengan rentang nilai dari -2 sampai dengan 2. Untuk mengubah rentang nilai X dan Y, Anda dapat menambahkan nilai-nilai batas X (dan Y) di belakang ekspresi yang digambar.

Rentang plot diatur dengan parameter yang ditetapkan berikut

- a,b: X-range (default -2,2)
- c,d: jrange-y (default: skala dengan nilai)
- r: atau jari-jari di sekitar pusat plot
- cx,cy: koordinat pusat plot (default 0,0)

```
>plot2d("x^3-x",-1,2):
```

```
>plot2d("sin(x)",-2*pi,2*pi): // plot sin(x) pada interval [-2pi, 2pi]
```

```
>plot2d("cos(x)","sin(3*x)",xmin=0,xmax=2pi):
```

Alternatif untuk titik dua adalah perintah insimg(lines), yang menyisipkan plot yang menempati sejumlah baris teks tertentu.

Dalam opsi, plot dapat diatur untuk muncul

- di jendela terpisah yang dapat diubah ukurannya,
- di jendela buku catatan.

Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.

Bagaimanapun, tekan tombol tabulator untuk melihat plotnya, jika tersembunyi.

Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Dalam contoh, kita memplot x^1 hingga x^4 menjadi 4 bagian jendela. Figure(0) mengatur ulang jendela default.

```
>reset;
>figure(2,2); ...
>for n=1 to 4; figure(n); plot2d("x^"+n); end; ...
>figure(0):
```

Di plot2d(), ada gaya alternatif yang tersedia dengan grid=x. Untuk gambaran umum, kami menunjukkan berbagai gaya kisi dalam satu gambar (lihat di bawah untuk perintah figure(). Grid gaya grid = 0 tidak disertakan. Ini tidak menunjukkan kisi dan tidak ada bingkai.

```
>figure(3,3); ...
>for k=1:9; figure(k); plot2d("x^3-x",-2,1,grid=k); end; ...
>figure(0):
```

Jika argumen untuk plot2d() adalah ekspresi diikuti oleh empat angka, angka-angka ini adalah rentang x dan y untuk plot.

Atau, a, b, c, d dapat ditentukan sebagai parameter yang ditetapkan sebagai a=... dll.

Dalam contoh berikut, kita mengubah gaya kisi, menambahkan label, dan menggunakan label vertikal untuk sumbu y.

```
>aspect(1.5); plot2d("sin(x)",0,2pi,-1.2,1.2,grid=3,xl="x",yl="sin(x)"):
```

```
>plot2d("sin(x)+cos(2*x)",0,4pi):
```

Gambar yang dihasilkan dengan memasukkan plot ke dalam jendela teks disimpan di direktori yang sama dengan buku catatan, secara default di subdirektori bernama "images". Mereka juga digunakan oleh ekspor HTML.

Anda cukup menandai gambar apa pun dan menyalinnya ke clipboard dengan Ctrl-C. Tentu saja, Anda juga dapat mengekspor grafik saat ini dengan fungsi di menu File.

Fungsi atau ekspresi dalam plot2d dievaluasi secara adaptif. Untuk kecepatan lebih, matikan plot adaptif dengan <adaptif dan tentukan jumlah subinterval dengan n=... Ini seharusnya diperlukan dalam kasus yang jarang terjadi saja.

```
>plot2d("sign(x)*exp(-x^2)",-1,1,<adaptive,n=10000):
```

```
>plot2d("x^x",r=1.2,cx=1,cy=1):
```

Perhatikan bahwa x^x tidak didefinisikan untuk x<=0. Fungsi plot2d menangkap kesalahan ini, dan mulai merencanakan segera setelah fungsi ditentukan. Ini berfungsi untuk semua fungsi yang mengembalikan NAN di luar jangkauan definisi mereka.

```
>plot2d("log(x)",-0.1,2):
```

Parameter square=true (atau >square) memilih jrange-y secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan ruang persegi di dalam jendela plot.

```
>plot2d("x^3-x",>square):
```

```
>plot2d(''integrate("sin(x)*exp(-x^2)",0,x)'',0,2): // plot integral
```

Jika Anda membutuhkan lebih banyak ruang untuk label-y, panggil shrinkwindow() dengan parameter yang lebih kecil, atau atur nilai positif untuk "smaller" di plot2d().

```
>plot2d("gamma(x)",1,10,yl="y-values",smaller=6,<vertical):
```

Symbolic expressions can also be used, since they are stored as simple string expressions.

```
>x=linspace(0,2pi,1000); plot2d(sin(5x),cos(7x)):
```

```
>a:=5.6; expr &= exp(-a*x^2)/a; // define expression
>plot2d(expr,-2,2): // plot from -2 to 2
```

```
>plot2d(expr,r=1,thickness=2): // plot in a square around (0,0)
```

```
>plot2d(&diff(expr,x),>add,style="--",color=red): // add another plot
```

```
>plot2d(&diff(expr,x,2),a=-2,b=2,c=-2,d=1): // plot in rectangle
```

```
>plot2d(&diff(expr,x),a=-2,b=2,>square): // keep plot square
```

```
>plot2d("x^2",0,1,steps=1,color=red,n=10):
```

```
>plot2d("x^2",>add,steps=2,color=blue,n=10):
```

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Fungsi dalam satu Parameter

Fungsi plot yang paling penting untuk plot planar adalah plot2d(). Fungsi ini diimplementasikan dalam bahasa Euler dalam file "plot.e", yang dimuat di awal program.

Berikut adalah beberapa contoh menggunakan fungsi. Seperti biasa di EMT, fungsi yang berfungsi untuk fungsi atau ekspresi lain, Anda dapat meneruskan parameter tambahan (selain x) yang bukan variabel global ke fungsi dengan parameter titik koma atau dengan koleksi panggilan.

```
>function f(x,a) := x^2/a+a*x^2-x; // define a function
>a=0.3; plot2d("f",0,1;a): // plot with a=0.3
```

```
>plot2d("f",0,1;0.4): // plot with a=0.4
```

```
>plot2d({{"f",0.2}},0,1): // plot with a=0.2
```

```
>plot2d({{"f(x,b)",b=0.1}},0,1): // plot with 0.1
```

```
>function f(x) := x^3-x; ...
>plot2d("f",r=1):
```

Berikut adalah ringkasan fungsi yang diterima

- ekspresi atau ekspresi simbolis dalam x
- fungsi atau fungsi simbolis dengan nama sebagai "f"
- fungsi simbolis hanya dengan nama f

Fungsi plot2d() juga menerima fungsi simbolis. Untuk fungsi simbolis, nama saja berfungsi.

```
>function f(x) &= diff(x^x,x)
```

```
                                  x
                                 x  (log(x) + 1)
```

```
>plot2d(f,0,2):
```

Tentu saja, untuk ekspresi atau ekspresi simbolis, nama variabel sudah cukup untuk memplotnya.

```
>expr &= sin(x)*exp(-x)
```

$$E^{-x} \sin(x)$$

```
>plot2d(expr,0,3pi):
```

```
>function f(x) &= x^x;
>plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
>plot2d(&diff(f(x),x),>add,color=red,style="-.-"):
```

Untuk gaya garis ada berbagai opsi.

- style="...". Pilih dari "-", "–", "-.", ".", ".-.", "-.-".
- color: Lihat di bawah untuk warna.
- ketebalan: Defaultnya adalah 1.

Warna dapat dipilih sebagai salah satu warna default, atau sebagai warna RGB.

- 0..15: indeks warna default.
- Konstanta warna: putih, hitam, merah, hijau, biru, cyan, zaitun, abu-abu muda, abu-abu, abu-abu gelap, oranye, hijau muda, pirus, biru muda, oranye muda, kuning
- RGB(Merah,Hijau,Biru): Parameter adalah real di [0,1].

```
>plot2d("exp(-x^2)",r=2,color=red,thickness=3,style="--"):
```

Berikut adalah tampilan warna EMT yang telah ditentukan sebelumnya.

```
>aspect(2); columnsplot(ones(1,16),lab=0:15,grid=0,color=0:15):
```

Tapi Anda bisa menggunakan warna apa saja.

```
>columnsplot(ones(1,16),grid=0,color=rgb(0,0,linspace(0,1,15))):
```

Menggambar Beberapa Kurva pada bidang koordinat yang sama

Plot lebih dari satu fungsi (beberapa fungsi) ke dalam satu jendela dapat dilakukan dengan cara yang berbeda. Salah satu metode adalah menggunakan >add untuk beberapa panggilan ke plot2d secara keseluruhan, tetapi panggilan pertama. Kami telah menggunakan fitur ini dalam contoh di atas.

```
>aspect(); plot2d("cos(x)",r=2,grid=6); plot2d("x",style=".",>add):
```

```
>aspect(1.5); plot2d("sin(x)",0,2pi); plot2d("cos(x)",color=blue,style="--",>add):
```

Salah satu kegunaan >add adalah untuk menambahkan titik pada kurva.

```
>plot2d("sin(x)",0,pi); plot2d(2,sin(2),>points,>add):
```

Kami menambahkan titik persimpangan dengan label (pada posisi "cl" untuk kiri tengah), dan memasukkan hasilnya ke dalam buku catatan. Kami juga menambahkan judul ke plot.

```
>plot2d(["cos(x)","x"],r=1.1,cx=0.5,cy=0.5, ...
>  color=[black,blue],style=["-",".“], ...
>  grid=1);
>x0=solve("cos(x)-x",1);  ...
>  plot2d(x0,x0,>points,>add,title="Intersection Demo");  ...
>  label("cos(x) = x",x0,x0,pos="cl",offset=20):
```

Dalam demo berikut, kita memplot fungsi sinc(x)=sin(x)/x dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolik.
Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke plot2d(). Yang kedua dan ketiga memiliki set bendera (>add), yang membuat plot menggunakan rentang sebelumnya.

Kami menambahkan kotak label yang menjelaskan fungsinya.

```
>$taylor(sin(x)/x,x,0,4)
```

$$\frac{x^4}{120}-\frac{x^2}{6}+1$$

```
>plot2d("sinc(x)",0,4pi,color=green,thickness=2); ...
>  plot2d(&taylor(sin(x)/x,x,0,8),>add,color=blue,style="--"); ...
>  plot2d(&taylor(sin(x)/x,x,0,16),>add,color=red,style="-.-"); ...
>  labelbox(["sinc","T8","T16"],styles=["-","--","-.-"], ...
>    colors=[black,blue,red]):
```

Dalam contoh berikut, kami menghasilkan Bernstein-Polynomial.

$$B_i(x) = \binom{n}{i} x^i (1-x)^{n-i}$$

```
>plot2d("(1-x)^10",0,1); // plot first function
>for i=1 to 10; plot2d("bin(10,i)*x^i*(1-x)^(10-i)",>add); end;
>insimg;
```

Metode kedua adalah menggunakan sepasang matriks nilai-x dan matriks nilai-y dengan ukuran yang sama.

Kami menghasilkan matriks nilai dengan satu Bernstein-Polynomial di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihat pengantar tentang bahasa matriks untuk mempelajari lebih detail.

```
>x=linspace(0,1,500);
>n=10; k=(0:n)'; // n is row vector, k is column vector
>y=bin(n,k)*x^k*(1-x)^(n-k); // y is a matrix then
>plot2d(x,y):
```

Perhatikan bahwa parameter warna dapat berupa vektor. Kemudian setiap warna digunakan untuk setiap baris matriks.

```
>x=linspace(0,1,200); y=x^(1:10)'; plot2d(x,y,color=1:10):
```

Metode lain adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan array warna, array gaya, dan array ketebalan dengan panjang yang sama.

```
>plot2d(["sin(x)","cos(x)"],0,2pi,color=4:5):
```

```
>plot2d(["sin(x)","cos(x)"],0,2pi): // plot vector of expressions
```

We can get such a vector from Maxima using makelist() and mxm2str().

```
>v &= makelist(binomial(10,i)*x^i*(1-x)^(10-i),i,0,10) // make list
```

```
                10            9              8  2             7  3
        [(1 - x)  , 10 (1 - x)  x, 45 (1 - x)  x , 120 (1 - x)  x ,
            6  4             5  5             4  6             3  7
  210 (1 - x)  x , 252 (1 - x)  x , 210 (1 - x)  x , 120 (1 - x)  x ,
           2  8              9   10
  45 (1 - x)  x , 10 (1 - x) x , x  ]
```

```
>mxm2str(v) // get a vector of strings from the symbolic vector
```

```
(1-x)^10
10*(1-x)^9*x
45*(1-x)^8*x^2
120*(1-x)^7*x^3
210*(1-x)^6*x^4
252*(1-x)^5*x^5
210*(1-x)^4*x^6
120*(1-x)^3*x^7
45*(1-x)^2*x^8
10*(1-x)*x^9
x^10
```

```
>plot2d(mxm2str(v),0,1): // plot functions
```

Alternatif lain adalah menggunakan bahasa matriks Euler.

Jika sebuah ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, semua fungsi ini akan diplot menjadi satu plot.

Untuk ini, gunakan vektor parameter dalam bentuk vektor kolom. Jika array warna ditambahkan, itu akan digunakan untuk setiap baris plot.

```
>n=(1:10)'; plot2d("x^n",0,1,color=1:10):
```

Ekspresi dan fungsi satu baris dapat melihat variabel global.

Jika Anda tidak dapat menggunakan variabel global, Anda perlu menggunakan fungsi dengan parameter tambahan, dan meneruskan parameter ini sebagai parameter titik koma.

Berhati-hatilah, untuk menempatkan semua parameter yang ditetapkan ke akhir perintah plot2d. Dalam contoh kita meneruskan a=5 ke fungsi f, yang kita plot dari -10 hingga 10.

```
>function f(x,a) := 1/a*exp(-x^2/a); ...
>plot2d("f",-10,10;5,thickness=2,title="a=5"):
```

Ekspresi dan fungsi satu baris dapat melihat variabel global. Atau, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut koleksi panggilan, dan itu adalah cara yang lebih disukai untuk meneruskan argumen ke fungsi yang dengan sendirinya diteruskan sebagai argumen ke fungsi lain.

Dalam contoh berikut, kita menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman untuk loop).

Jika Anda tidak dapat menggunakan variabel global, Anda perlu menggunakan fungsi dengan parameter tambahan, dan meneruskan parameter ini sebagai parameter titik koma.

Berhati-hatilah, untuk menempatkan semua parameter yang ditetapkan ke akhir perintah plot2d. Dalam contoh kita meneruskan a=5 ke fungsi f, yang kita plot dari -10 hingga 10.

```
>plot2d({{"f",1}},-10,10); ...
>for a=2:10; plot2d({{"f",a}},>add); end:
```

Kita dapat mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris matriks f(x,a) adalah satu fungsi. Selain itu, kita dapat mengatur warna untuk setiap baris matriks. Klik dua kali pada fungsi getspectral() untuk penjelasan.

```
>x=-10:0.01:10; a=(1:10)'; plot2d(x,f(x,a),color=getspectral(a/10)):
```

Label Teks

Dekorasi sederhana bisa

- judul dengan title="..."
- label x dan y dengan xl="...", yl="..."
- label teks lain dengan label("...",x,y)

Perintah label akan memplot ke dalam plot saat ini pada koordinat plot (x,y). Itu bisa mengambil argumen posisional.

```
>plot2d("x^3-x",-1,2,title="y=x^3-x",yl="y",xl="x"):
```

```
>expr := "log(x)/x"; ...
>  plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ...
>  label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):
```

Ada juga fungsi labelbox(), yang dapat menampilkan fungsi dan teks. Dibutuhkan vektor string dan warna, satu item untuk setiap fungsi.

```
>function f(x) &= x^2*exp(-x^2);  ...
>plot2d(&f(x),a=-3,b=3,c=-1,d=1);  ...
>plot2d(&diff(f(x),x),>add,color=blue,style="--"); ...
>labelbox(["function","derivative"],styles=["-","--"], ...
>   colors=[black,blue],w=0.4):
```

Kotak ini ditambatkan di kanan atas secara default, tetapi >kiri menambatkan di kiri atas. Anda dapat memindahkannya ke tempat mana pun yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafis. Lebarnya otomatis.

Untuk plot titik, kotak label juga berfungsi. Tambahkan parameter >poin, atau vektor bendera, satu untuk setiap label.

Dalam contoh berikut, hanya ada satu fungsi. Jadi kita bisa menggunakan string alih-alih vektor string. Kami mengatur warna teks menjadi hitam untuk contoh ini.

```
>n=10; plot2d(0:n,bin(n,0:n),>addpoints); ...
>labelbox("Binomials",styles="[]",>points,x=0.1,y=0.1, ...
>tcolor=black,>left):
```

Gaya plot ini juga tersedia di statplot(). Seperti pada plot2d() warna dapat diatur untuk setiap baris plot. Ada lebih banyak plot khusus untuk tujuan statistik (lihat tutorial tentang statistik).

```
>statplot(1:10,random(2,10),color=[red,blue]):
```

Fitur serupa adalah fungsi textbox().

Lebar secara default adalah lebar maksimum baris teks. Tapi itu bisa diatur oleh pengguna juga.

```
>function f(x) &= exp(-x)*sin(2*pi*x); ...
>plot2d("f(x)",0,2pi); ...
>textbox(latex("\text{Example of a damped oscillation}\ f(x)=e^{-x}sin(2\pi x)"),w=0.85):
```

Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk selengkapnya tentang string Unicode).

```
>plot2d("x^3-x",title=u"x &rarr; x&sup3; - x"):
```

Label pada sumbu x dan y dapat vertikal, serta sumbu.

```
>plot2d("sinc(x)",0,2pi,xl="x",yl=u"x &rarr; sinc(x)",>vertical):
```

**Getah

Anda juga dapat memplot rumus LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke biner "latex" dan "dvipng" harus berada di jalur sistem, atau Anda harus mengatur LaTeX di menu opsi.

Perhatikan, penguraian LaTeX itu lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil latex() sebelum loop sekali dan menggunakan hasilnya (gambar dalam matriks RGB).

Dalam plot berikut, kami menggunakan LaTeX untuk label x dan y, label, kotak label, dan judul plot.

```
>plot2d("exp(-x)*sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ...
>  title=latex("\text{Function $\Phi$}"), ...
>  xl=latex("\phi"),yl=latex("\Phi(\phi)")); ...
>textbox( ...
>  latex("\Phi(\phi) = e^{-\phi} \frac{\sin(\phi)}{\phi}"),x=0.8,y=0.5); ...
```

```
>label(latex("\Phi",color=blue),1,0.4):
```

Seringkali, kita menginginkan spasi non-konformal dan label teks pada sumbu x. Kita dapat menggunakan xaxis() dan yaxis() seperti yang akan kita tunjukkan nanti.

Cara termudah adalah dengan melakukan plot kosong dengan bingkai menggunakan grid=4, dan kemudian menambahkan kisi dengan ygrid() dan xgrid(). Dalam contoh berikut, kita menggunakan tiga string LaTeX untuk label pada sumbu x dengan xtick().

```
>plot2d("sinc(x)",0,2pi,grid=4,<ticks); ...
>ygrid(-2:0.5:2,grid=6); ...
>xgrid([0:2]*pi,<ticks,grid=6);  ...
>xtick([0,pi,2pi],["0","\pi","2\pi"],>latex):
```

Tentu saja, fungsi juga dapat digunakan.

```
>function map f(x) ...
```

```
if x>0 then return x^4
else return x^2
endif
endfunction
```

Parameter "peta" membantu menggunakan fungsi untuk vektor. Untuk plot, itu tidak perlu. Tetapi untuk menunjukkan bahwa vektorisasi berguna, kami menambahkan beberapa poin kunci ke plot pada x=-1, x=0 dan x=1.

Pada plot berikut, kami juga memasukkan beberapa kode LaTeX. Kami menggunakannya untuk dua label dan kotak teks. Tentu saja, Anda hanya akan dapat menggunakan LaTeX jika Anda telah menginstal LaTeX dengan benar.

```
>plot2d("f",-1,1,xl="x",yl="f(x)",grid=6);  ...
>plot2d([-1,0,1],f([-1,0,1]),>points,>add); ...
>label(latex("x^3"),0.72,f(0.72)); ...
>label(latex("x^2"),-0.52,f(-0.52),pos="ll"); ...
>textbox( ...
>  latex("f(x)=\begin{cases} x^3 & x>0 \\ x^2 & x \le 0\end{cases}"), ...
>  x=0.7,y=0.2):
```

Saat memplot fungsi atau ekspresi, parameter >user memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat

- memperbesar dengan + atau -
- Pindahkan plot dengan tombol kursor
- Pilih jendela plot dengan mouse
- Setel ulang tampilan dengan spasi
- keluar dengan pengembalian

Tombol spasi akan mengatur ulang plot ke jendela plot asli.

Saat memplot data, bendera >user hanya akan menunggu penekanan tombol.

```
>plot2d({{"x^3-a*x",a=1}},>user,title="Press any key!"):
```

```
>plot2d("exp(x)*sin(x)",user=true, ...
>  title="+/- or cursor keys (return to exit)"):
```

Berikut ini menunjukkan cara lanjutan interaksi pengguna (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi bawaan mousedrag() menunggu peristiwa mouse atau keyboard. Ini melaporkan mouse ke bawah, mouse bergerak atau mouse ke atas, dan penekanan tombol. Fungsi dragpoints() memanfaatkan ini, dan memungkinkan pengguna menyeret titik apa pun dalam plot.

Kita membutuhkan fungsi plot terlebih dahulu. Sebagai contoh, kita melakukan interpolasi dalam 5 titik dengan polinomial. Fungsi harus diplot ke dalam area plot tetap.

```
>function plotf(xp,yp,select) ...
```

```
    d=interp(xp,yp);
    plot2d("interpval(xp,d,x)";d,xp,r=2);
    plot2d(xp,yp,>points,>add);
    if select>0 then
```

```
    plot2d(xp[select],yp[select],color=red,>points,>add);
  endif;
  title("Drag one point, or press space or return!");
endfunction
```

Perhatikan parameter titik koma di plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, mengakses nilai-nilai secara global.

Sekarang kita menghasilkan beberapa nilai acak, dan biarkan pengguna menyeret titik-titiknya.

```
>t=-1:0.5:1; dragpoints("plotf",t,random(size(t))-0.5):
```

Ada juga fungsi, yang memplot fungsi lain tergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

Pertama kita membutuhkan fungsi plot.

```
>function plotf([a,b]) := plot2d("exp(a*x)*cos(2pi*b*x)",0,2pi;a,b);
```

Kemudian kita membutuhkan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional garis judul.
Ada slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

```
>dragvalues("plotf",["a","b"],[-1,2],[[-2,2];[1,10]], ...
>  heading="Drag these values:",hcolor=black):
```

Dimungkinkan untuk membatasi nilai yang diseret ke bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor derajat n ke fungsi kosinus.

```
>function plotf(n) ...
```

```
  plot2d("cos(x)",0,2pi,>square,grid=6);
  plot2d(&"taylor(cos(x),x,0,@n)",color=blue,>add);
  textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",>left);
  endfunction
```

Sekarang kita membiarkan derajat n bervariasi dari 0 hingga 20 dalam 20 pemberhentian. Hasil dari dragvalues() digunakan untuk memplot sketsa dengan n ini, dan untuk memasukkan plot ke dalam buku catatan.

```
>nd=dragvalues("plotf","degree",2,[0,20],20,y=0.8, ...
>   heading="Drag the value:"); ...
>plotf(nd):
```

Berikut ini adalah demonstrasi sederhana dari fungsi tersebut. Pengguna dapat menggambar di atas jendela plot, meninggalkan jejak titik.

```
>function dragtest ...
```

```
  plot2d(none,r=1,title="Drag with the mouse, or press any key!");
  start=0;
  repeat
    {flag,m,time}=mousedrag();
    if flag==0 then return; endif;
    if flag==2 then
      hold on; mark(m[1],m[2]); hold off;
    endif;
  end
endfunction
```

```
>dragtest // lihat hasilnya dan cobalah lakukan!
```

Gaya Plot 2D

Secara default, EMT menghitung kutu sumbu otomatis dan menambahkan label ke setiap centang. Ini dapat diubah dengan parameter grid. Gaya default sumbu dan label dapat dimodifikasi. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang ke gaya default, gunakan reset().

```
>aspect();
>figure(3,4); ...
> figure(1); plot2d("x^3-x",grid=0); ... // no grid, frame or axis
> figure(2); plot2d("x^3-x",grid=1); ... // x-y-axis
> figure(3); plot2d("x^3-x",grid=2); ... // default ticks
> figure(4); plot2d("x^3-x",grid=3); ... // x-y- axis with labels inside
> figure(5); plot2d("x^3-x",grid=4); ... // no ticks, only labels
> figure(6); plot2d("x^3-x",grid=5); ... // default, but no margin
> figure(7); plot2d("x^3-x",grid=6); ... // axes only
> figure(8); plot2d("x^3-x",grid=7); ... // axes only, ticks at axis
> figure(9); plot2d("x^3-x",grid=8); ... // axes only, finer ticks at axis
> figure(10); plot2d("x^3-x",grid=9); ... // default, small ticks inside
> figure(11); plot2d("x^3-x",grid=10); ...// no ticks, axes only
> figure(0):
```

Parameter <frame mematikan bingkai, dan framecolor=blue mengatur bingkai ke warna biru.

Jika Anda menginginkan centang Anda sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

```
>aspect(1.5);
>plot2d("x^3-x",grid=0); // plot
>frame; xgrid([-1,0,1]); ygrid(0): // add frame and grid
```

Untuk judul plot dan label sumbu, lihat contoh berikut.

```
>plot2d("exp(x)",-1,1);
>textcolor(black); // set the text color to black
>title(latex("y=e^x")); // title above the plot
>xlabel(latex("x")); // "x" for x-axis
>ylabel(latex("y"),>vertical); // vertical "y" for y-axis
>label(latex("(0,1)"),0,1,color=blue): // label a point
```

Sumbu dapat digambar secara terpisah dengan xaxis() dan yaxis().

```
>plot2d("x^3-x",<grid,<frame);
>xaxis(0,xx=-2:1,style="->"); yaxis(0,yy=-5:5,style="->"):
```

Teks pada plot dapat diatur dengan label(). Dalam contoh berikut, "lc" berarti pusat bawah. Ini mengatur posisi label relatif terhadap koordinat plot.

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

$$x^3 - x$$

```
>plot2d(f,-1,1,>square);
>x0=fmin(f,0,1); // compute point of minimum
>label("Rel. Min.",x0,f(x0),pos="lc"): // add a label there
```

Ada juga kotak teks.

```
>plot2d(&f(x),-1,1,-2,2); // function
>plot2d(&diff(f(x),x),>add,style="--",color=red); // derivative
>labelbox(["f","f'"],["-","--"],[black,red]): // label box
```

```
>plot2d(["exp(x)","1+x"],color=[black,blue],style=["-","-.-"]):
```

```
>gridstyle("->",color=gray,textcolor=gray,framecolor=gray);  ...
> plot2d("x^3-x",grid=1);   ...
> settitle("y=x^3-x",color=black); ...
> label("x",2,0,pos="bc",color=gray);  ...
> label("y",0,6,pos="cl",color=gray); ...
> reset():
```

Untuk kontrol yang lebih baik, sumbu x dan sumbu y dapat dilakukan secara manual.

Perintah fullwindow() memperluas jendela plot karena kita tidak lagi membutuhkan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk mengatur ulang ke default.

```
>fullwindow; ...
> gridstyle(color=darkgray,textcolor=darkgray); ...
> plot2d(["2^x","1","2^(-x)"],a=-2,b=2,c=0,d=4,<grid,color=4:6,<frame); ...
> xaxis(0,-2:1,style="->"); xaxis(0,2,"x",<axis); ...
> yaxis(0,4,"y",style="->"); ...
> yaxis(-2,1:4,>left); ...
> yaxis(2,2^(-2:2),style=".",<left); ...
> labelbox(["2^x","1","2^-x"],colors=4:6,x=0.8,y=0.2); ...
> reset:
```

Berikut adalah contoh lain, di mana string Unicode digunakan dan sumbu di luar area plot.

```
>aspect(1.5);
>plot2d(["sin(x)","cos(x)"],0,2pi,color=[red,green],<grid,<frame); ...
> xaxis(-1.1,(0:2)*pi,xt=["0",u"&pi;",u"2&pi;"],style="-",>ticks,>zero);  ...
> xgrid((0:0.5:2)*pi,<ticks); ...
```

```
> yaxis(-0.1*pi,-1:0.2:1,style="-",>zero,>grid); ...
> labelbox(["sin","cos"],colors=[red,green],x=0.5,y=0.2,>left); ...
> xlabel(u"&phi;"); ylabel(u"f(&phi;)"):
```

Jika x dan y adalah vektor data, data ini akan digunakan sebagai koordinat x dan y dari kurva. Dalam hal ini, a, b, c, dan d, atau jari-jari r dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Atau, > persegi dapat diatur untuk mempertahankan rasio aspek persegi.

Memplot ekspresi hanyalah singkatan dari plot data. Untuk plot data, Anda memerlukan satu atau lebih baris nilai-x, dan satu atau lebih baris nilai-y. Dari rentang dan nilai-x, fungsi plot2d akan menghitung data untuk diplot, secara default dengan evaluasi adaptif dari fungsi tersebut. Untuk plot titik gunakan ">titik", untuk garis campuran dan titik gunakan ">titik tambahan".

Tetapi Anda dapat memasukkan data secara langsung.

- Gunakan vektor baris untuk x dan y untuk satu fungsi.
- Matriks untuk x dan y diplot baris demi baris.

Berikut adalah contoh dengan satu baris untuk x dan y.

```
>x=-10:0.1:10; y=exp(-x^2)*x; plot2d(x,y):
```

Data juga dapat diplot sebagai titik. Gunakan points=true untuk ini. Plotnya bekerja seperti poligon, tetapi hanya menggambar sudutnya.

- style="...": Pilih dari "[]", "<>", "o", ".", "..", "+", "*", "[]", "<>", "o", ".. ", "", "|".

Untuk memplot kumpulan titik, gunakan >titik. Jika warna adalah vektor warna, masing-masing menunjuk
mendapat warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna berlaku untuk baris matriks.
Parameter >addpoints menambahkan titik ke segmen garis untuk plot data.

```
>xdata=[1,1.5,2.5,3,4]; ydata=[3,3.1,2.8,2.9,2.7]; // data
>plot2d(xdata,ydata,a=0.5,b=4.5,c=2.5,d=3.5,style="."); // lines
>plot2d(xdata,ydata,>points,>add,style="o"): // add points
```

```
>p=polyfit(xdata,ydata,1); // get regression line
>plot2d("polyval(p,x)",>add,color=red): // add plot of line
```

3.5
3
2.5
0.5
1
1.5
2
2.5
3
3.5
4
4.5

Menggambar Daerah Yang Dibatasi Kurva

Plot data benar-benar poligon. Kita juga dapat memplot kurva atau kurva yang diisi.

- filled=true mengisi plot.
- style="...": Pilih dari "", "/", "", "/".
- fillcolor: Lihat di atas untuk warna yang tersedia.

Warna isian ditentukan oleh argumen "fillcolor", dan pada <outline opsional mencegah menggambar batas untuk semua gaya kecuali yang default.

```
>t=linspace(0,2pi,1000); // parameter for curve
>x=sin(t)*exp(t/pi); y=cos(t)*exp(t/pi); // x(t) and y(t)
>figure(1,2); aspect(16/9)
>figure(1); plot2d(x,y,r=10); // plot curve
>figure(2); plot2d(x,y,r=10,>filled,style="/",fillcolor=red); // fill curve
>figure(0):
```

Dalam contoh berikut, kita memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian yang berbeda.

```
>x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)*0.5,r=1,>filled,style="/"):
```

```
>t=linspace(0,2pi,6); ...
>plot2d(cos(t),sin(t),>filled,style="/",fillcolor=red,r=1.2):
```

```
>t=linspace(0,2pi,6); plot2d(cos(t),sin(t),>filled,style="#"):
```

Contoh lain adalah septagon, yang kami buat dengan 7 titik pada lingkaran satuan.

```
>t=linspace(0,2pi,7);  ...
> plot2d(cos(t),sin(t),r=1,>filled,style="/",fillcolor=red):
```

Berikut ini adalah himpunan nilai maksimal dari empat kondisi linier kurang dari atau sama dengan 3. Ini adalah A[k].v<=3 untuk semua baris A. Untuk mendapatkan sudut yang bagus, kami menggunakan n yang relatif besar.

```
>A=[2,1;1,2;-1,0;0,-1];
>function f(x,y) := max([x,y].A');
>plot2d("f",r=4,level=[0;3],color=green,n=111):
```

Poin utama dari bahasa matriks adalah memungkinkan untuk menghasilkan tabel fungsi dengan mudah.

```
>t=linspace(0,2pi,1000); x=cos(3*t); y=sin(4*t);
```

Kita sekarang memiliki vektor x dan y dari nilai. plot2d() dapat memplot nilai-nilai ini sebagai kurva yang menghubungkan titik-titik. Plot bisa diisi. Dalam hal ini
Ini menghasilkan hasil yang bagus karena aturan penggulitan, yang digunakan untuk isi.

```
>plot2d(x,y,<grid,<frame,>filled):
```

Vektor interval diplot terhadap nilai x sebagai wilayah yang diisi antara nilai interval yang lebih rendah dan atas.

Ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga digunakan untuk memplot kesalahan statistik.

```
>t=0:0.1:1; ...
> plot2d(t,interval(t-random(size(t)),t+random(size(t))),style="|");  ...
> plot2d(t,t,add=true):
```

Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka plot2d akan memplot rentang interval yang terisi dalam bidang. Gaya isian sama dengan gaya poligon.

```
>t=-1:0.01:1; x=~t-0.01,t+0.01~; y=x^3-x;
>plot2d(t,y):
```

Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Bagi ini, level harus matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```

We can also fill ranges of values like

$$-1 \leq (x^2+y^2)^2 - x^2 + y^2 \leq 0.$$

```
>plot2d("(x^2+y^2)^2-x^2+y^2",r=1.2,level=[-1;0],style="/"):
```

```
>plot2d("cos(x)","sin(x)^3",xmin=0,xmax=2pi,>filled,style="/"):
```

Nilai x tidak perlu diurutkan. (x,y) hanya menggambarkan kurva. Jika x diurutkan, kurva adalah grafik dari suatu fungsi.

Dalam contoh berikut, kita memplot spiral

lateks: gamma(t) = t cdot (cos(2pi t),sin(2pi t))

Kita perlu menggunakan sangat banyak titik untuk tampilan yang halus atau fungsi adaptive() untuk mengevaluasi ekspresi (lihat fungsi adaptive() untuk lebih jelasnya).

```
>t=linspace(0,1,1000); ...
>plot2d(t*cos(2*pi*t),t*sin(2*pi*t),r=1):
```

Atau, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini memplot kurva yang sama seperti di atas.

```
>plot2d("x*cos(2*pi*x)","x*sin(2*pi*x)",xmin=0,xmax=1,r=1):
```

```
>t=linspace(0,1,1000); r=exp(-t); x=r*cos(2pi*t); y=r*sin(2pi*t);
>plot2d(x,y,r=1):
```

Pada contoh berikutnya, kita memplot kurva

$$\gamma(t) = (r(t)\cos(t), r(t)\sin(t))$$

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}.$$

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
>plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5):
```

1
0
-1
-2
-1
0
1
2

Menggambar Grafik Bilangan Kompleks

Array bilangan kompleks juga dapat diplot. Kemudian titik jaringan akan terhubung. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1x2) dalam argumen cgrid hanya garis kisi yang terlihat.

Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi dalam bidang kompleks.

Dalam contoh berikut, kita memplot gambar lingkaran satuan di bawah fungsi eksponensial. Parameter cgrid menyembunyikan beberapa kurva grid.

```
>aspect(); r=linspace(0,1,50); a=linspace(0,2pi,80)'; z=r*exp(I*a);...
>plot2d(z,a=-1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):
```

```
>aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=r*exp(I*a);
>plot2d(exp(z),cgrid=[40,10]):
```

```
>r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=r*exp(I*a);
>plot2d(exp(z),>points,>add):
```

Vektor bilangan kompleks secara otomatis diplot sebagai kurva dalam bidang kompleks dengan bagian real dan bagian imajiner.

Dalam contoh, kita memplot lingkaran satuan dengan

$$\gamma(t) = e^{it}$$

```
>t=linspace(0,2pi,1000); ...
>plot2d(exp(I*t)+exp(4*I*t),r=2):
```

2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1
0
1
2

Plot Statistik

Ada banyak fungsi yang mengkhususkan diri pada plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

Jumlah kumulatif dari nilai terdistribusi 0-1-normal menghasilkan jalan acak.

```
>plot2d(cumsum(randnormal(1,1000))):
```

Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

```
>X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
```

```
>columnsplot(cumsum(random(10)),style="/",color=blue):
```

Ini juga dapat menampilkan string sebagai label.

```
>months=["Jan","Feb","Mar","Apr","May","Jun", ...
>  "Jul","Aug","Sep","Oct","Nov","Dec"];
>values=[10,12,12,18,22,28,30,26,22,18,12,8];
>columnsplot(values,lab=months,color=red,style="-");
>title("Temperature"):
```

```
>k=0:10;
>plot2d(k,bin(10,k),>bar):
```

```
>plot2d(k,bin(10,k)); plot2d(k,bin(10,k),>points,>add):
```

```
>plot2d(normal(1000),normal(1000),>points,grid=6,style=".."):
```

```
>plot2d(normal(1,1000),>distribution,style="O"):
```

```
>plot2d("qnormal",0,5;2.5,0.5,>filled):
```

Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

```
>w=randexponential(1,1000); // exponential distribution
>plot2d(w,>distribution): // or distribution=n with n intervals
```

Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan >bar di plot3d, atau dengan plot kolom.

```
>w=normal(1000); // 0-1-normal distribution
>{x,y}=histo(w,10,v=[-6,-4,-2,-1,0,1,2,4,6]); // interval bounds v
>plot2d(x,y,>bar):
```

Fungsi statplot() mengatur gaya dengan string sederhana.

```
>statplot(1:10,cumsum(random(10)),"b"):
```

```
>n=10; i=0:n; ...
>plot2d(i,bin(n,i)/2^n,a=0,b=10,c=0,d=0.3); ...
>plot2d(i,bin(n,i)/2^n,points=true,style="ow",add=true,color=blue):
```

Selain itu, data dapat diplot sebagai batang. Dalam hal ini, x harus diurutkan dan satu elemen lebih panjang dari y. Bilah akan memanjang dari x[i] ke x[i+1] dengan nilai y[i]. Jika x memiliki ukuran yang sama dengan y, itu akan diperpanjang oleh satu elemen dengan spasi terakhir.

Gaya isian dapat digunakan seperti di atas.

```
>n=10; k=bin(n,0:n); ...
>plot2d(-0.5:n+0.5,k,bar=true,fillcolor=lightgray):
```

Data untuk plot batang (bar=1) dan histogram (histogram=1) dapat diberikan secara eksplisit dalam xv dan yv, atau dapat dihitung dari distribusi empiris dalam xv dengan >distribusi (atau distribusi=n). Histogram nilai xv akan dihitung secara otomatis dengan >histogram. Jika >genap ditentukan, nilai xv akan dihitung dalam interval bilangan bulat.

```
>plot2d(normal(10000),distribution=50):
```

```
>k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m,>bar):
```

```
>columnsplot(m,k):
```

```
>plot2d(random(600)*6,histogram=6):
```

Untuk distribusi, ada parameter distribution=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

```
>plot2d(normal(1,1000),distribution=10,style="\/"):
```

Dengan parameter even=true, ini akan menggunakan interval bilangan bulat.

```
>plot2d(intrandom(1,1000,10),distribution=10,even=true):
```

Perhatikan bahwa ada banyak plot statistik, yang mungkin berguna. Lihatlah tutorial tentang statistik.

```
>columnsplot(getmultiplicities(1:6,intrandom(1,6000,6))):
```

```
>plot2d(normal(1,1000),>distribution); ...
>  plot2d("qnormal(x)",color=red,thickness=2,>add):
```

Ada juga banyak plot khusus untuk statistik. Boxplot menunjukkan kuartil distribusi ini dan banyak outlier. Menurut definisi, outlier dalam boxplot adalah data yang melebihi 1,5 kali rentang 50% tengah plot.

```
>M=normal(5,1000); boxplot(quartiles(M)):
```

3
2
1
0
-1
-2
1
2
3
4
5

Fungsi Implisit

Plot implisit menunjukkan garis level yang menyelesaikan f(x,y)=level, di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika level="auto", akan ada garis level nc, yang akan menyebar antara minimum dan maksimum fungsi secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan >hue untuk menunjukkan nilai fungsi. Untuk fungsi implisit, xv harus berupa fungsi atau ekspresi dari parameter x dan y, atau, sebagai alternatif, xv dapat berupa matriks nilai.

Euler dapat menandai garis level

lateks: f(x,y) = c

dari fungsi apa pun.

Untuk menggambar himpunan f(x,y)=c untuk satu atau lebih konstanta c, Anda dapat menggunakan plot2d() dengan plot implisitnya di bidang. Parameter untuk c adalah level=c, di mana c dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi untuk setiap titik dalam plot. Parameter "n" menentukan kehalusan plot.

```
>aspect(1.5);
>plot2d("x^2+y^2-x*y-x",r=1.5,level=0,contourcolor=red):
```

```
>expr := "2*x^2+x*y+3*y^4+y"; // define an expression f(x,y)
>plot2d(expr,level=0): // Solutions of f(x,y)=0
```

```
>plot2d(expr,level=0:0.5:20,>hue,contourcolor=white,n=200): // nice
```

```
>plot2d(expr,level=0:0.5:20,>hue,>spectral,n=200,grid=4): // nicer
```

Ini juga berfungsi untuk plot data. Tetapi Anda harus menentukan rentang untuk label sumbu.

```
>x=-2:0.05:1; y=x'; z=expr(x,y);
>plot2d(z,level=0,a=-1,b=2,c=-2,d=1,>hue):
```

```
>plot2d("x^3-y^2",>contour,>hue,>spectral):
```

```
>plot2d("x^3-y^2",level=0,contourwidth=3,>add,contourcolor=red):
```

```
>z=z+normal(size(z))*0.2;
>plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
```

```
>plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
```

```
>plot2d("x^2+y^3+x*y",level=1,r=4,n=100):
```

```
>plot2d("x^2+2*y^2-x*y",level=0:0.1:10,n=100,contourcolor=white,>hue):
```

Dimungkinkan juga untuk mengisi set

$$a \leq f(x,y) \leq b$$

dengan rentang level.

Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

```
>plot2d(expr,level=[0;1],style="-",color=blue): // 0 <= f(x,y) <= 1
```

Plot implisit juga dapat menunjukkan rentang level. Kemudian level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua akhir dari setiap interval. Atau, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level ke interval.

```
>plot2d("x^4+y^4",r=1.5,level=[0;1],color=blue,style="/"):
```

```
>plot2d("x^2+y^3+x*y",level=[0,2,4;1,3,5],style="/",r=2,n=100):
```

```
>plot2d("x^2+y^3+x*y",level=-10:20,r=2,style="-",dl=0.1,n=100):
```

```
>plot2d("sin(x)*cos(y)",r=pi,>hue,>levels,n=100):
```

Dimungkinkan juga untuk menandai suatu wilayah

$$a \leq f(x,y) \leq b.$$

Ini dilakukan dengan menambahkan level dengan dua baris.

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>  style="#",color=red,<outline, ...
>  level=[-2;0],n=100):
```

Dimungkinkan untuk menentukan tingkat tertentu. Misalnya, kita dapat memplot solusi dari persamaan seperti

$$x^3 - xy + x^2y^2 = 6$$

```
>plot2d("x^3-x*y+x^2*y^2",r=6,level=1,n=100):
```

```
>function starplot1 (v, style="/", color=green, lab=none) ...
```

```
if !holding() then clg; endif;
w=window(); window(0,0,1024,1024);
h=holding(1);
r=max(abs(v))*1.2;
setplot(-r,r,-r,r);
n=cols(v); t=linspace(0,2pi,n);
v=v|v[1]; c=v*cos(t); s=v*sin(t);
cl=barcolor(color); st=barstyle(style);
loop 1 to n
  polygon([0,c[#],c[#+1]],[0,s[#],s[#+1]],1);
  if lab!=none then
    rlab=v[#]+r*0.1;
    {col,row}=toscreen(cos(t[#])*rlab,sin(t[#])*rlab);
    ctext(""+lab[#],col,row-textheight()/2);
```

```
      endif;
    end;
    barcolor(cl); barstyle(st);
    holding(h);
    window(w);
  endfunction
```

Tidak ada kutu kisi atau sumbu di sini. Selain itu, kami menggunakan jendela penuh untuk plot.

Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Ini tidak perlu, jika Anda yakin bahwa plot Anda berhasil.

```
>reset; starplot1(normal(1,10)+5,color=red,lab=1:10):
```

Terkadang, Anda mungkin ingin merencanakan sesuatu yang tidak dapat dilakukan plot2d, tetapi hampir.

Dalam fungsi berikut, kita melakukan plot impuls logaritma. Plot2d dapat melakukan plot logaritma, tetapi tidak untuk bilah impuls.

```
>function logimpulseplot1 (x,y) ...
```

```
  {x0,y0}=makeimpulse(x,log(y)/log(10));
  plot2d(x0,y0,>bar,grid=0);
  h=holding(1);
  frame();
  xgrid(ticks(x));
  p=plot();
  for i=-10 to 10;
    if i<=p[4] and i>=p[3] then
       ygrid(i,yt="10^"+i);
    endif;
  end;
  holding(h);
endfunction
```

Mari kita ujinya dengan nilai yang didistribusikan secara eksponensial.

```
>aspect(1.5); x=1:10; y=-log(random(size(x)))*200; ...
>logimpulseplot1(x,y):
```

Mari kita animasikan kurva 2D menggunakan plot langsung. Perintah plot(x,y) hanya memplot kurva ke jendela plot. setplot(a,b,c,d) mengatur jendela ini.

Fungsi wait(0) memaksa plot muncul di jendela grafis. Jika tidak, penggambaran ulang terjadi dalam interval waktu yang jarang.

```
>function animliss (n,m) ...
```

```
  t=linspace(0,2pi,500);
  f=0;
  c=framecolor(0);
  l=linewidth(2);
  setplot(-1,1,-1,1);
  repeat
    clg;
```

```
    plot(sin(n*t),cos(m*t+f));
    wait(0);
    if testkey() then break; endif;
    f=f+0.02;
  end;
  framecolor(c);
  linewidth(l);
  endfunction
```

Tekan tombol apa saja untuk menghentikan animasi ini.

```
>animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER
```

Plot Logaritma

EMT menggunakan parameter "logplot" untuk skala logaritma.
Plot logaritma dapat diplot baik menggunakan skala logaritma di y dengan logplot=1, atau menggunakan skala logaritma di x dan y dengan logplot=2, atau di x dengan logplot=3.

- logplot=1: logaritma y
- logplot=2: x-y-logaritma
- logplot=3: x-logaritma

```
>plot2d("exp(x^3-x)*x^2",1,5,logplot=1):
```

```
>plot2d("exp(x+sin(x))",0,100,logplot=1):
```

```
>plot2d("exp(x+sin(x))",10,100,logplot=2):
```

```
>plot2d("gamma(x)",1,10,logplot=1):
```

```
>plot2d("log(x*(2+sin(x/100)))",10,1000,logplot=3):
```

Ini juga berfungsi dengan plot data.

```
>x=10^(1:20); y=x^2-x;
>plot2d(x,y,logplot=2):
```

10^{40}
10^{35}
10^{30}
10^{25}
10^{20}
10^{15}
10^{10}
10^{5}
10^{2}
10^{4}
10^{6}
10^{8}
10^{10}
10^{12}
10^{14}
10^{16}
10^{18}
10^{20}

Menggambar Plot 3D dengan EMT

Ini adalah pengantar plot 3D di Euler. Kita memerlukan plot 3D untuk memvisualisasikan fungsi dua variabel.

Euler menggambar fungsi tersebut menggunakan algoritma pengurutan untuk menyembunyikan bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Defaultnya adalah dari kuadran x-y positif ke arah titik asal x=y=z=0, tetapi sudut=0° terlihat dari arah sumbu y. Sudut pandang dan tinggi dapat diubah.

Euler dapat memplot

- permukaan dengan bayangan dan garis datar atau rentang datar,
- awan titik,
- kurva parametrik,
- permukaan implisit.

Plot 3D suatu fungsi menggunakan plot3d. Cara termudah adalah memplot ekspresi dalam x dan y. Parameter r mengatur rentang plot di sekitar (0,0).

```
>aspect(1.5); plot3d("x^2+sin(y)",-5,5,0,6*pi):
```

```
>plot3d("x^2+x*sin(y)",-5,5,0,6*pi):
```

Silakan lakukan modifikasi agar gambar "talang bergelombang" tersebut tidak lurus melainkan melengkung/melingkar, baik melingkar secara mendatar maupun melingkar turun/naik (seperti papan peluncur pada kolam renang. Temukan rumusnya.

Fungsi Dua Variabel

Untuk grafik fungsi, gunakan

- ekspresi sederhana dalam x dan y,
- nama fungsi dua variabel
- atau matriks data.

Defaultnya adalah kisi kawat yang terisi dengan warna berbeda di kedua sisinya. Perhatikan bahwa jumlah interval kisi default adalah 10, tetapi plot menggunakan jumlah persegi panjang 40x40 default untuk membuat permukaan. Ini dapat diubah.

- n=40, n=[40,40]: jumlah garis kisi di setiap arah
- kisi=10, kisi=[10,10]: jumlah garis kisi di setiap arah.

Kami menggunakan default n=40 dan grid=10.

```
>plot3d("x^2+y^2"):
```

Interaksi pengguna dimungkinkan dengan parameter >user. Pengguna dapat menekan tombol berikut.

- kiri, kanan, atas, bawah: mengubah sudut pandang
- +, -: memperbesar atau memperkecil
- a: menghasilkan anaglif (lihat di bawah)
- l: mengubah arah sumber cahaya (lihat di bawah)
- spasi: mengatur ulang ke default
- return: mengakhiri interaksi

```
>plot3d("exp(-x^2+y^2)",>user, ...
>  title="Turn with the vector keys (press return to finish)"):
```

Rentang plot untuk fungsi dapat ditentukan dengan

- a,b: rentang x
- c,d: rentang y
- r: persegi simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: skala ke nilai fungsi (default adalah <fscale).
scale: angka atau vektor 1x2 untuk diskalakan ke arah x dan y.
frame: jenis frame (default 1).

```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):
```

Tampilan dapat diubah dengan berbagai cara.

- jarak: jarak pandang ke plot.
- perbesaran: nilai perbesaran.
- sudut: sudut ke sumbu y negatif dalam radian.
- tinggi: tinggi tampilan dalam radian.

Nilai default dapat diperiksa atau diubah dengan fungsi view(). Fungsi ini mengembalikan parameter dalam urutan di atas.

```
>view
```

```
[5,  2.6,  2,  0.4]
```

Jarak yang lebih dekat membutuhkan zoom yang lebih sedikit. Efeknya lebih seperti lensa sudut lebar.

Dalam contoh berikut, sudut=0 dan tinggi=0 terlihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```

Plot selalu mengarah ke tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter center.

```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
>  center=[0.4,0,0],zoom=5):
```

Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung pada ukuran plot. Namun, label merujuk pada ukuran sebenarnya.

Jika Anda menonaktifkannya dengan scale=false, Anda perlu berhati-hati agar plot tetap sesuai dengan jendela plot, dengan mengubah jarak tampilan atau zoom, dan memindahkan bagian tengah.

```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
>  center=[0,0,-2],frame=3):
```

Plot polar juga tersedia. Parameter polar=true menggambar plot polar. Fungsi harus tetap merupakan fungsi dari x dan y. Parameter "fscale" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi akan diskalakan agar sesuai dengan kubus.

```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=blue):
```

```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```

Parameter rotate memutar fungsi dalam x di sekitar sumbu x.

- rotate = 1: Menggunakan sumbu x
- rotate=2: Menggunakan sumbu z

```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```

```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```

```
>plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1):
```

```
>plot3d("x^3-2y+cos(x)",a=1,b=2pi,angle=-50°,zoom=2):
```

```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```

Berikut ini adalah plot dengan tiga fungsi.

```
>plot3d("x","x^2+y^2","y",r=2,zoom=3.5,frame=3):
```

2
z
-2
-2
x
2
0
y
8

Plot Kontur

Untuk plot, Euler menambahkan garis kisi-kisi. Sebagai gantinya, dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona berwarna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan bayangan. Pada semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.

- Rona: Mengaktifkan bayangan cahaya, bukan kabel.
- >contour: Memplot garis kontur otomatis pada plot.
- level=... (atau level): Vektor nilai untuk garis kontur.

Defaultnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut ini, kami menggunakan grid yang lebih halus untuk titik 100x100, skala fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...
> >contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):
```

```
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```

Bayangan default menggunakan warna abu-abu. Tetapi, kisaran warna spektral juga tersedia.

- >spektral: Menggunakan skema spektral default
- color =...: Menggunakan warna khusus atau skema spektral

Untuk plot berikut ini, kami menggunakan skema spektral default dan menambah jumlah titik untuk mendapatkan tampilan yang sangat mulus.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```

Alih-alih garis level otomatis, kita juga dapat menetapkan nilai garis level. Hal ini akan menghasilkan garis level yang tipis, alih-alih rentang level.

```
>plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=redgreen):
```

Pada plot berikut ini, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas-batas level sebagai kolom.

Selain itu, kami menghamparkan grid dengan 10 interval di setiap arah.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...
>  >spectral,angle=30°,grid=10,contourcolor=gray):
```

Pada contoh berikut, kami memplot himpunan, di mana

$$f(x,y) = x^y - y^x = 0$$

Kita menggunakan satu garis tipis untuk garis level.

```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```

Dimungkinkan untuk menampilkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

```
>plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2):
```

Berikut ini beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi-kisi.

```
>figure(2,2); ...
>expr="y^3-x^2"; ...
>figure(1);  ...
>  plot3d(expr,<frame,>cp,cpcolor=spectral); ...
>figure(2);  ...
>  plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
>figure(3);  ...
>  plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
>figure(4);  ...
>  plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...
>figure(0):
```

Ada beberapa skema spektral lainnya, yang diberi nomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan color=value, di mana value

- spektral: untuk rentang dari biru ke merah
- putih: untuk rentang yang lebih redup
- kuningbiru, ungu-hijau, biru-kuning, hijau-merah
- biru-kuning, hijau-ungu, kuning-biru, merah-hijau

```
>figure(3,3); ...
>for i=1:9;  ...
>  figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4);  ...
>end; ...
>figure(0):
```

Sumber cahaya dapat diubah dengan l dan tombol kursor selama interaksi pengguna. Ini juga dapat ditetapkan dengan parameter.

- light: arah cahaya
- amb: cahaya sekitar antara 0 dan 1

Perhatikan, bahwa program ini tidak membuat perbedaan di antara sisi-sisi plot. Tidak ada bayangan. Untuk ini Anda akan membutuhkan Povray.

```
>plot3d("-x^2-y^2", ...
>  hue=true,light=[0,1,1],amb=0,user=true, ...
>  title="Press l and cursor keys (return to exit)"):
```

Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
>  zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01):
```

Warna 0 memberikan efek pelangi yang istimewa.

```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```

Permukaannya juga bisa transparan.

```
>plot3d("x^2+y^2",>transparent,grid=10,wirecolor=red):
```

-1
-1
x
y
1
1
2
z
0

Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari sebuah fungsi dalam tiga variabel. Solusi dari

lateks: f(x,y,z) = 0

dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, bidang x-z, dan bidang y-z.

- implisit = 1: potong sejajar dengan bidang-y-z
- implicit = 2: memotong sejajar dengan bidang x-z
- implicit=4: memotong sejajar dengan bidang x-y

Tambahkan nilai-nilai ini, jika Anda mau. Pada contoh, kami memplot

$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```

```
>c=1; d=1;
>plot3d("((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)-d",r=2,
```

```
>plot3d("x^2-x*y+y^3",>implicit,r=0.5,zoom=3):
```

```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):
```

x
-2
-2
y
2
2
2
z
-2

Memplot Data 3D

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu menyediakan matriks nilai x, y, dan z, atau tiga fungsi atau ekspresi fx(x,y), fy(x,y), fz(x,y).

$$\gamma(t,s) = (x(t,s), y(t,s), z(t,s))$$

Karena x,y,z adalah matriks, kita mengasumsikan bahwa (t,s) berjalan melalui kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang dalam ruang.

Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

Pada contoh berikut, kita menggunakan vektor nilai t dan vektor kolom nilai s untuk memparameterkan permukaan bola. Pada gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...
>plot3d(x,y,z,>hue, ...
>color=blue,<frame,grid=[10,20], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.4,height=50°):
```

Berikut ini adalah contoh, yang merupakan grafik suatu fungsi.

```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```

Namun demikian, kita bisa membuat segala macam permukaan. Berikut ini adalah permukaan yang sama dengan fungsi

$$x = y\,z$$

```
>plot3d(t*s,t,s,angle=180°,grid=10):
```

Dengan lebih banyak upaya, kita bisa menghasilkan banyak permukaan.

Dalam contoh berikut ini, kami membuat tampilan berbayang dari bola yang terdistorsi. Koordinat yang biasa digunakan untuk bola adalah

$$\gamma(t,s) = (\cos(t)\cos(s), \sin(t)\sin(s), \cos(s))$$

dengan

$$0 \leq t \leq 2\pi, \quad \frac{-\pi}{2} \leq s \leq \frac{\pi}{2}.$$

Kami mengubahnya dengan sebuah faktor

$$d(t,s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
>  light=[1,0,1],frame=0,zoom=5):
```

Tentu saja, awan titik juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor untuk koordinat titik.

Gaya-gayanya sama seperti pada plot2d dengan points=true;

```
>n=500;  ...
>  plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

Anda juga dapat memplot kurva dalam bentuk 3D. Dalam hal ini, akan lebih mudah untuk menghitung titik-titik kurva. Untuk kurva pada bidang, kami menggunakan urutan koordinat dan parameter wire = true.

```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```

```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
>linewidth=3,wirecolor=blue):
```

```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```

EMT juga dapat membuat plot dalam mode anaglyph. Untuk melihat plot semacam itu, Anda memerlukan kacamata merah/cyan.

```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```

Sering kali, skema warna spektral digunakan untuk plot. Hal ini menekankan ketinggian fungsi.

```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```

Euler juga dapat memplot permukaan yang diparameterkan, ketika parameternya adalah nilai x, y, dan z dari gambar kisi-kisi persegi panjang di dalam ruang.

Untuk demo berikut ini, kami menyiapkan parameter u dan v, dan menghasilkan koordinat ruang dari parameter ini.

```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...
>X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```

Berikut ini contoh yang lebih rumit, yang tampak megah dengan kacamata merah/cyan.

```
>u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)';  ...
>x:=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...
>y:=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...
> z=sin(u)+2*cos(3*v); ...
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```

Plot batang juga dapat digunakan. Untuk ini, kita harus menyediakan

- x: vektor baris dengan n+1 elemen
- y: vektor kolom dengan n+1 elemen
- z: matriks nilai berukuran nxn.

z dapat lebih besar, tetapi hanya nilai nxn yang akan digunakan.

Pada contoh, pertama-tama kita menghitung nilainya. Kemudian kita sesuaikan x dan y, sehingga vektor berada di tengah-tengah nilai yang digunakan.

```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...
>plot3d(xa,ya,z,bar=true):
```

Dimungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20):
```

Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan scale(M), atau menskalakan matriks dengan >zscale. Hal ini dapat dikombinasikan dengan faktor penskalaan individual yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8):
```

```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v',scols=1:5,ccols=[1:5]):
```

1
2
3
4
5

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...
>style="#",color=red,<outline, ...
>level=[-2;0],n=100):
```

```
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

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

Kami ingin memutar kurva jantung di sekitar sumbu y. Berikut ini adalah ekspresi yang mendefinisikan jantung:

$$f(x,y) = (x^2 + y^2 - 1)^3 - x^2.y^3.$$

Selanjutnya kita atur

$$x = r.cos(a), \quad y = r.sin(a).$$

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

$$\left(r^2 - 1\right)^3 + \frac{\left(\sin\left(5\,a\right) - \sin\left(3\,a\right) - 2\,\sin a\right)\,r^5}{16}$$

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang menyelesaikan untuk r, jika a diberikan. Dengan fungsi tersebut kita dapat memplotkan jantung yang diputar sebagai permukaan parametrik.

```
>function map f(a) := bisect("fr",0,2;a); ...
>t=linspace(-pi/2,pi/2,100); r=f(t);  ...
>s=linspace(pi,2pi,100)'; ...
>plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```

Berikut ini adalah plot 3D dari gambar di atas yang diputar mengelilingi sumbu-z. Kami mendefinisikan fungsi, yang menggambarkan objek.

```
>function f(x,y,z) ...
```

```
  r=x^2+y^2;
  return (r+z^2-1)^3-r*z^3;
   endfunction
```

```
>plot3d("f(x,y,z)", ...
>xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...
>implicit=1,angle=-30°,zoom=2.5,n=[10,100,60],>anaglyph):
```

1.4
z
-1.2
1.2
x
0
-1.2
y
1.2

Plot 3D Khusus

Fungsi plot3d memang bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda juga bisa mendapatkan plot berbingkai dari objek apa pun yang Anda sukai.

Meskipun Euler bukan program 3D, namun dapat menggabungkan beberapa objek dasar. Kami mencoba memvisualisasikan parabola dan garis singgungnya.

```
>function myplot ...
```

```
    y=-1:0.01:1; x=(-1:0.01:1)';
    plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ..
      hues=0.5,>contour,color=orange);
    h=holding(1);
    plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);
    holding(h);
  endfunction
```

Sekarang framedplot() menyediakan frame, dan mengatur tampilan.

```
>framedplot("myplot",[-1,1,-1,1,0,1],height=0,angle=-30°, ...
>  center=[0,0,-0.7],zoom=3):
```

Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d() mengatur jendela ke fullwindow() secara default, namun plotcontourplane() mengasumsikannya.

```
>x=-1:0.02:1.1; y=x'; z=x^2-y^4;
>function myplot (x,y,z) ...
```

```
    zoom(2);
    wi=fullwindow();
    plotcontourplane(x,y,z,level="auto",<scale);
    plot3d(x,y,z,>hue,<scale,>add,color=white,level="thin");
    window(wi);
    reset();
  endfunction
```

```
>myplot(x,y,z):
```

Animasi

Euler dapat menggunakan frame untuk melakukan pra-komputasi animasi.

Salah satu fungsi yang memanfaatkan teknik ini adalah rotate. Fungsi ini dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi ini memanggil addpage() untuk setiap plot baru. Akhirnya fungsi ini menganimasikan plot tersebut.

Silakan pelajari sumber dari rotate untuk melihat lebih detail.

```
>function testplot () := plot3d("x^2+y^3"); ...
>rotate("testplot"); testplot():
```

-1
-1
y
x
1
1
2
z
-1

Menggambar Povray

Dengan bantuan file Euler povray.e, Euler dapat menghasilkan file Povray. Hasilnya sangat bagus untuk dilihat.

Anda perlu menginstal Povray (32bit atau 64bit) dari http://www.povray.org/, dan meletakkan sub-direktori "bin" dari Povray ke dalam jalur lingkungan, atau mengatur variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe".

Antarmuka Povray dari Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah current.pov, dan direktori defaultnya adalah eulerhome(), biasanya c:\Users\Username\Euler. Povray menghasilkan sebuah file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan berkas-berkas ini, gunakan povclear().

Fungsi pov3d memiliki semangat yang sama dengan plot3d. Fungsi ini dapat menghasilkan grafik dari sebuah fungsi f(x,y), atau sebuah permukaan dengan koordinat X,Y,Z dalam bentuk matriks, termasuk garis-garis level yang bersifat opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat adegan ke dalam notebook Euler.

Selain pov3d(), ada banyak fungsi yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string, yang berisi kode Povray untuk objek. Untuk menggunakan fungsi-fungsi ini, mulai file Povray dengan povstart(). Kemudian gunakan writeln(...) untuk menulis objek ke file scene. Terakhir, akhiri file dengan povend(). Secara default, raytracer akan dimulai, dan PNG akan dimasukkan ke dalam buku catatan Euler.

Fungsi objek memiliki parameter yang disebut "look", yang membutuhkan string dengan kode povray untuk tekstur dan hasil akhir objek. Fungsi povlook() dapat digunakan untuk menghasilkan string ini. Fungsi ini memiliki parameter untuk warna, transparansi, Phong Shading, dll.

Perhatikan bahwa Povray universe memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat tetap berpikir dalam sistem koordinat Euler dengan z yang mengarah vertikal ke atas, dan sumbu x, y, z di tangan kanan.
Anda perlu memuat file povray.

```
>load povray;
```

Pastikan direktori bin povray berada di dalam path. Jika tidak, edit variabel berikut sehingga berisi jalur ke povray yang dapat dieksekusi.

```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Untuk kesan pertama, kita plot sebuah fungsi sederhana. Perintah berikut ini menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk melacak sinar pada file ini.

Jika Anda memulai perintah berikut, GUI Povray akan terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya, apakah Anda ingin mengizinkan file exe dijalankan. Anda dapat menekan cancel untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK pada jendela Povray untuk mengetahui dialog awal Povray.

```
>plot3d("x^2+y^2",zoom=2):
```

```
>pov3d("sin(2x)",zoom=3,angle=-15°);
```

```
>pov3d("x^2+y^2",zoom=3);
```

Kita dapat membuat fungsi menjadi transparan dan menambahkan hasil akhir lainnya. Kita juga dapat menambahkan garis level ke plot fungsi.

```
>pov3d("x^2+y^3",axiscolor=red,angle=-45°,>anaglyph, ...
>  look=povlook(cyan,0.2),level=-1:0.5:1,zoom=3.8);
```

Kadang-kadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.

Kami memplot kumpulan titik pada bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=2, ...
>  angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=10°,n=50, ...
>  <fscale,zoom=3.8);
```

Merencanakan dengan Koordinat

Sebagai pengganti fungsi, kita dapat membuat plot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objek.

Pada contoh, kita memutar sebuah fungsi pada sumbu z.

```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,50)'; ...
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light=[10,5,15]);
```

Pada contoh berikut, kita memplot gelombang teredam. Kami menghasilkan gelombang dengan bahasa matriks Euler.

Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut. Perhatikan bahwa plot3d menskalakan plot, sehingga sesuai dengan kubus satuan.

```
>r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ...
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3;  ...
>pov3d(x,y,z,zoom=6,axis=0,height=30°,add=povsphere([0.5,0,0.25],0.15,povlook(red)), ...
>  w=500,h=300);
```

Dengan metode bayangan canggih Povray, hanya sedikit titik yang bisa menghasilkan permukaan yang sangat halus. Hanya pada batas-batas dan bayangan, trik ini bisa terlihat jelas.

Untuk itu, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>Z &= x^2*y^3
```

```
 2  3
x  y
```

Persamaan permukaannya adalah [x,y,Z]. Kami menghitung dua turunan terhadap x dan y dari persamaan ini dan mengambil hasil perkalian silang sebagai normal.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

Kami mendefinisikan normal sebagai hasil kali silang dari turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

```
          3       2  2
[- 2 x y , - 3 x  y , 1]
```

Kami hanya menggunakan 25 poin.

```
>x=-1:0.5:1; y=x';
>pov3d(x,y,Z(x,y),angle=10°, ...
>  xv=NX(x,y),yv=NY(x,y),zv=NZ(x,y),<shadow);
```

Berikut ini adalah simpul Trefoil yang dibuat oleh A. Busser di Povray. Ada versi yang lebih baik dari ini dalam contoh.

See: Examples\Trefoil Knot | Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kita tambahkan vektor normal di sini. Kita menggunakan Maxima untuk menghitung normal untuk kita. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolik.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...
>Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...
>Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunan terhadap x dan y.

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang yang normal, yang merupakan produk silang dari dua turunan.

```
>dn &= crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

Vektor normal adalah evaluasi dari ekspresi simbolik dn[i] untuk i=1,2,3. Sintaks untuk ini adalah &"ekspresi"(parameter). Ini adalah sebuah alternatif dari metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),>anaglyph,axis=0,zoom=5,w=450,h=350, ...
>  <shadow,look=povlook(blue), ...
>  xv=&"dn[1]"(x,y), yv=&"dn[2]"(x,y), zv=&"dn[3]"(x,y));
```

Kami juga dapat menghasilkan kisi-kisi dalam bentuk 3D.

```
>povstart(zoom=4); ...
>x=-1:0.5:1; r=1-(x+1)^2/6; ...
>t=(0°:30°:360°)'; y=r*cos(t); z=r*sin(t); ...
>writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...
>povend();
```

Dengan povgrid(), kurva dapat dibuat.

```
>povstart(center=[0,0,1],zoom=3.6); ...
>t=linspace(0,2,1000); r=exp(-t); ...
>x=cos(2*pi*10*t)*r; y=sin(2*pi*10*t)*r; z=t; ...
>writeln(povgrid(x,y,z,povlook(red))); ...
>writeAxis(0,2,axis=3); ...
>povend();
```

Di atas, kami menggunakan pov3d untuk memplot permukaan. Antarmuka povray di Euler juga dapat menghasilkan objek Povray. Objek-objek ini disimpan sebagai string di Euler, dan perlu ditulis ke file Povray.

Kita memulai output dengan povstart().

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

```
>povstart(zoom=4);
```

Pertama, kita mendefinisikan tiga silinder, dan menyimpannya dalam bentuk string di Euler.

Fungsi povx() dll. hanya mengembalikan vektor [1,0,0], yang dapat digunakan sebagai gantinya.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...
>c2=povcylinder(-povy,povy,1,povlook(yellow)); ...
>c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

```
>c2
```

```
cylinder { <0,0,-1>, <0,0,1>, 1
 texture { pigment { color rgb <0.941176,0.941176,0.392157> }  }
 finish { ambient 0.2 }
 }
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.

Hal ini dilakukan dengan povlook(), yang mengembalikan sebuah string dengan kode Povray yang relevan. Kita dapat menggunakan warna default Euler, atau menentukan warna kita sendiri. Kita juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

```
 texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> }  }
 finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek perpotongan, dan menulis hasilnya ke file.

```
>writeln(povintersection([c1,c2,c3]));
```

Perpotongan tiga silinder sulit untuk divisualisasikan, jika Anda belum pernah melihatnya.

```
>povend;
```

Fungsi-fungsi berikut ini menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana. Fungsi povbox() mengembalikan sebuah string, yang berisi koordinat kotak, tekstur dan hasil akhir.

```
>function onebox(x,y,z,d) := povbox([x,y,z],[x+d,y+d,z+d],povlook());
>function fractal (x,y,z,h,n) ...
```

```
   if n==1 then writeln(onebox(x,y,z,h));
   else
     h=h/3;
     fractal(x,y,z,h,n-1);
     fractal(x+2*h,y,z,h,n-1);
     fractal(x,y+2*h,z,h,n-1);
     fractal(x,y,z+2*h,h,n-1);
     fractal(x+2*h,y+2*h,z,h,n-1);
     fractal(x+2*h,y,z+2*h,h,n-1);
     fractal(x,y+2*h,z+2*h,h,n-1);
     fractal(x+2*h,y+2*h,z+2*h,h,n-1);
     fractal(x+h,y+h,z+h,h,n-1);
   endif;
  endfunction
```

```
>povstart(fade=10,<shadow);
>fractal(-1,-1,-1,2,4);
>povend();
```

Perbedaan memungkinkan pemotongan satu objek dari objek lainnya. Seperti persimpangan, ada bagian dari objek CSG Povray.

```
>povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kita akan mendefinisikan sebuah objek di Povray, alih-alih menggunakan sebuah string di Euler. Definisi akan langsung dituliskan ke file.

Koordinat kotak -1 berarti [-1,-1,-1].

```
>povdefine("mycube",povbox(-1,1));
```

Kita dapat menggunakan objek ini dalam povobject(), yang mengembalikan sebuah string seperti biasa.

```
>c1=povobject("mycube",povlook(red));
```

Kami menghasilkan kubus kedua, dan memutar serta menskalakannya sedikit.

```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...
>  rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita ambil selisih dari kedua objek tersebut.

```
>writeln(povdifference(c1,c2));
```

Sekarang tambahkan tiga sumbu.

```
>writeAxis(-1.2,1.2,axis=1); ...
>writeAxis(-1.2,1.2,axis=2); ...
>writeAxis(-1.2,1.2,axis=4); ...
>povend();
```

Fungsi Implisit

Povray dapat memplot himpunan di mana f(x,y,z)=0, seperti parameter implisit di plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsi-fungsi tersebut sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

$$((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)=d$$

```
>povstart(angle=70°,height=50°,zoom=4);
>c=0.1; d=0.1; ...
>writeln(povsurface("(pow(pow(x,2)+pow(y,2)-pow(c,2),2)+pow(pow(z,2)-1,2))*(pow(pow(y,2)+pow(z,2)-po
>povend();
```

```
Error : Povray error!

Error generated by error() command

povray:
    error("Povray error!");
Try "trace errors" to inspect local variables after errors.
povend:
    povray(file,w,h,aspect,exit);
```

```
>povstart(angle=25°,height=10°);
>writeln(povsurface("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,2,"")));
>povend();
```

```
>povstart(angle=70°,height=50°,zoom=4);
```

Membuat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi.

```
>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...
>writeAxes(); ...
>povend();
```

Objek Jaring

Pada contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.

Kami ingin memaksimalkan xy di bawah kondisi x+y = 1 dan mendemonstrasikan sentuhan tangensial dari garis level.

```
>povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kita tidak dapat menyimpan objek dalam sebuah string seperti sebelumnya, karena ukurannya terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan declare. Fungsi povtriangle() melakukan hal ini secara otomatis. Fungsi ini dapat menerima vektor normal seperti halnya pov3d().

Berikut ini mendefinisikan objek mesh, dan menuliskannya langsung ke dalam file.

```
>x=0:0.02:1; y=x'; z=x*y; vx=-y; vy=-x; vz=1;
>mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kita tentukan dua cakram, yang akan berpotongan dengan permukaan.

```
>cl=povdisc([0.5,0.5,0],[1,1,0],2); ...
>ll=povdisc([0,0,1/4],[0,0,1],2);
```

Tuliskan permukaan dikurangi kedua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tuliskan kedua perpotongan tersebut.

```
>writeln(povintersection([mesh,cl],povlook(red))); ...
>writeln(povintersection([mesh,ll],povlook(gray)));
```

Tulislah satu titik secara maksimal.

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));
```

Tambahkan sumbu dan selesaikan.

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...
>povend();
```

Anaglyph dalam Povray

Untuk menghasilkan anaglyph untuk kacamata merah/cyan, Povray harus dijalankan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph().

Tentu saja, Anda membutuhkan kacamata merah/cyan untuk melihat contoh berikut dengan benar.

Fungsi pov3d() memiliki tombol sederhana untuk menghasilkan anaglyph.

```
>pov3d("-exp(-x^2-y^2)/2",r=2,height=45°,>anaglyph, ...
>  center=[0,0,0.5],zoom=3.5);
```

Jika Anda membuat scene dengan objek, Anda harus menempatkan pembuatan scene ke dalam suatu fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

```
>function myscene ...
```

```
    s=povsphere(povc,1);
    cl=povcylinder(-povz,povz,0.5);
    clx=povobject(cl,rotate=xrotate(90°));
    cly=povobject(cl,rotate=yrotate(90°));
    c=povbox([-1,-1,0],1);
    un=povunion([cl,clx,cly,c]);
    obj=povdifference(s,un,povlook(red));
    writeln(obj);
    writeAxes();
  endfunction
```

Fungsi povanaglyph() melakukan semua ini. Parameter-parameternya seperti pada povstart() dan povend() yang digabungkan.

```
>povanaglyph("myscene",zoom=4.5);
```

Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak objek. Namun Anda tidak dibatasi pada objek-objek tersebut. Anda dapat membuat objek sendiri, yang menggabungkan objek-objek lain, atau objek yang benar-benar baru.

Kami mendemonstrasikan sebuah torus. Perintah Povray untuk ini adalah "torus". Jadi kita mengembalikan sebuah string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat pada titik asal.

```
>function povdonat (r1,r2,look="") ...
```

```
    return "torus {"+r1+","+r2+look+"}";
  endfunction
```

Inilah torus pertama kami.

```
>t1=povdonat(0.8,0.2)
```

```
  torus {0.8,0.2}
```

Mari kita gunakan objek ini untuk membuat torus kedua, ditranslasikan dan diputar.

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
object { torus {0.8,0.2}
 rotate 90 *x
 translate <0.8,0,0>
 }
```

Sekarang, kita tempatkan semua benda ini ke dalam suatu pemandangan. Untuk tampilannya, kami menggunakan Phong Shading.

```
>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...
>writeln(povobject(t1,povlook(green,phong=1))); ...
>writeln(povobject(t2,povlook(green,phong=1))); ...
```

```
>povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, program ini tidak menampilkan kesalahan. Oleh karena itu, Anda harus menggunakan

```
>povend(<keluar);
```

jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray tetap terbuka.

```
>povend(h=320,w=480);
```

Berikut adalah contoh yang lebih rumit. Kami menyelesaikan

$$Ax \leq b, \quad x \geq 0, \quad c.x \to \text{Max}.$$

dan menunjukkan titik-titik yang layak dan optimal dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];
>b=[10,10,10,10]';
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini memiliki solusi atau tidak.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0,  1,  0.5]
```

Ya, benar.

Selanjutnya kita mendefinisikan dua objek. Yang pertama adalah pesawat

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look="") ...
```

```
    return povplane(a,b,look)
  endfunction
```

Kemudian kita tentukan perpotongan semua setengah ruang dan kubus.

```
>function adm (A, b, r, look="") ...
```

```
    ol=[];
    loop 1 to rows(A); ol=ol|oneplane(A[#],b[#]); end;
    ol=ol|povbox([0,0,0],[r,r,r]);
    return povintersection(ol,look);
  endfunction
```

Sekarang, kita bisa merencanakan adegan tersebut.

```
>povstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); ...
>writeln(adm(A,b,2,povlook(green,0.4))); ...
>writeAxes(0,1.3,0,1.6,0,1.5); ...
```

Berikut ini adalah lingkaran di sekeliling optimal.

```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
>  povlook(red,0.9)));
```

Dan kesalahan pada arah yang optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah sebuah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

```
>writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=5°)); ...
>povend();
```

Contoh Lainnya

```
Anda dapat menemukan beberapa contoh lain untuk Povray di Euler dalam
```

file-file berikut.

See: Examples/Dandelin Spheres
See: Examples/Donat Math
See: Examples/Trefoil Knot
See: Examples/Optimization by Affine Scaling

Kalkulus dengan EMT

Materi Kalkulus mencakup di antaranya:

- Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi)
- Limit Fungsi,
- Turunan Fungsi,
- Integral Tak Tentu,
- Integral Tentu dan Aplikasinya,
- Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

Mendefinisikan Fungsi

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:

- Menggunakan format nama_fungsi := rumus fungsi (untuk fungsi numerik),
- Menggunakan format nama_fungsi &= rumus fungsi (untuk fungsi simbolik, namun dapat dihitung secara numerik),
- Menggunakan format nama_fungsi &&= rumus fungsi (untuk fungsi simbolik murni, tidak dapat dihitung langsung),
- Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah function (bukan sebagai ekspresi).

Berikut adalah adalah beberapa contoh cara mendefinisikan fungsi:

$$f(x) = 2x^2 + e^{\sin(x)}.$$

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
```

```
1
4.31977682472
20.7392088022
```

```
>f(a) // tidak dapat dihitung nilainya
```

```
Real 41 x 1 matrix

             1
       1.21868
       1.55948
       2.01872
       2.58957
       3.26182
       4.02223
       4.85563
       5.74672
       6.68221
       7.65308
       8.65613
       9.69456
       10.7774
       11.9179
       13.1314
       14.4331
       15.8362
       17.3508
```

```
    18.984
...
```

Silakan Anda plot kurva fungsi di atas!

Berikutnya kita definisikan fungsi:

$$g(x) = \frac{\sqrt{x^2 - 3x}}{x + 1}.$$

```
>function g(x) := sqrt(x^2-3*x)/(x+1)
>g(3)
```

```
0
```

```
>g(0)
```

```
0
```

```
>g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik
```

```
Floating point error!
Error in sqrt
Try "trace errors" to inspect local variables after errors.
g:
```

```
    useglobal; return sqrt(x^2-3*x)/(x+1)
Error in:
g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik ...
    ^
```

Silakan Anda plot kurva fungsi di atas!

```
>f(g(5)) // komposisi fungsi
```

```
2.20920171961
```

```
>g(f(5))
```

```
0.950898070639
```

```
>function h(x) := f(g(x)) // definisi komposisi fungsi
>h(5) // sama dengan f(g(5))
```

```
2.20920171961
```

Silakan Anda plot kurva fungsi komposisi fungsi f dan g:

$$h(x) = f(g(x))$$

dan

$$u(x) = g(f(x))$$

bersama-sama kurva fungsi f dan g dalam satu bidang koordinat.

```
>f(0:10) // nilai-nilai f(0), f(1), f(2), ..., f(10)
```

```
[1,  4.31978,  10.4826,  19.1516,  32.4692,  50.3833,  72.7562,
99.929,  130.69,  163.51,  200.58]
```

```
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
```

```
[1,  4.31978,  10.4826,  19.1516,  32.4692,  50.3833,  72.7562,
99.929,  130.69,  163.51,  200.58]
```

```
>gmap(200:210)
```

```
[0.987534,  0.987596,  0.987657,  0.987718,  0.987778,  0.987837,
0.987896,  0.987954,  0.988012,  0.988069,  0.988126]
```

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0. \end{cases}$$

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format :=, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.

```
>function map f(x) ...
```

```
    if x>0 then return x^3
    else return x^2
    endif;
  endfunction
```

```
>f(1)
```

```
1
```

```
>f(-2)
```

```
4
```

```
>f(-5:5)
```

```
[25,  16,  9,  4,  1,  0,  1,  8,  27,  64,  125]
```

```
>aspect(1.5); plot2d("f(x)",-5,5):
```

```
>function f(x) &= 2*E^x // fungsi simbolik
```

```
   x
2 E
```

```
>$f(a) // nilai fungsi secara simbolik
```

$$2\,e^{a}$$

```
>f(E) // nilai fungsi berupa bilangan desimal
```

```
30.308524483
```

```
>$f(E), $float(%)
```

$$30.30852448295852$$

```
>function g(x) &= 3*x+1
```

```
3 x + 1
```

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

```
   3 x + 1
2 E
```

```
>plot2d("h(x)",-1,1):
```

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan fungsi-fungsi tersebut dan komposisinya di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik fungsi-fungsi tersebut dan komposisi-komposisi 2 fungsi.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

Menghitung Limit

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi "plus" atau "minus". Hasil limit dapat berupa nilai, "und" (tak definisi), "ind" (tak tentu namun terbatas), "infinity" (kompleks tak hingga).

Perhatikan beberapa contoh berikut. Perhatikan cara menampilkan perhitungan secara lengkap, tidak hanya menampilkan hasilnya saja.

```
>$showev('limit(sqrt(x^2-3*x)/(x+1),x,inf))
```

$$\lim_{x\to\infty} \frac{\sqrt{x^2-3\,x}}{x+1} = 1$$

```
>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

$$-\frac{4}{5}$$

$$\lim_{x\to 3} \frac{x^3-13\,x^2+51\,x-63}{x^3-4\,x^2-3\,x+18} = -\frac{4}{5}$$

Fungsi tersebut diskontinu di titik x=3. Berikut adalah grafik fungsinya.

```
>aspect(1.5); plot2d("(x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18)",0,4); plot2d(3,-4/5,>points,style="ow
```

```
>$limit(2*x*sin(x)/(1-cos(x)),x,0)
```

4

$$2\left(\lim_{x\to 0}\frac{x\,\sin x}{1-\cos x}\right)=4$$

Fungsi tersebut diskontinu di titik x=0. Berikut adalah grafik fungsinya.

```
>plot2d("2*x*sin(x)/(1-cos(x))",-pi,pi); plot2d(0,4,>points,style="ow",>add):
```

```
>$limit(cot(7*h)/cot(5*h),h,0)
```

$$\frac{5}{7}$$

$$\lim_{h\to 0}\frac{\cot(7\,h)}{\cot(5\,h)}=\frac{5}{7}$$

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di x=0. Berikut adalah grafiknya.

```
>plot2d("cot(7*x)/cot(5*x)",-0.001,0.001); plot2d(0,5/7,>points,style="ow",>add):
```

```
>$showev('limit(((x/8)^(1/3)-1)/(x-8),x,8))
```

$$\lim_{x\to 8} \frac{\frac{x^{\frac{1}{3}}}{2} - 1}{x - 8} = \frac{1}{24}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x-8)^(1/3)-1)/(x-8)",7.99,8.01); plot2d(8,1/24,>points,style="ow",>add):
```

```
>$showev('limit(1/(2*x-1),x,0))
```

$$\lim_{x\to 0} \frac{1}{2\,x-1} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1/(2*x-1))",-0.1,0.1); plot2d(0,-1,>points,style="ow",>add):
```

```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

$$\lim_{x\to 5}\frac{x^2-3\,x-10}{x-5}=7$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x^2-3x-10)/(x-5))",4.9,5.1); plot2d(5,7,>points,style="ow",>add):
```

```
>$showev('limit(sqrt(x^2+x)-x,x,inf))
```

$$\lim_{x\to\infty} \sqrt{x^2+x} - x = \frac{1}{2}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x^2+x)^0.5-x)",0,6); plot2d(5,7,>points,style="ow",>add):
```

```
>$showev('limit(abs(x-1)/(x-1),x,1,minus))
```

$$\lim_{x\uparrow 1} \frac{|x-1|}{x-1} = -1$$

Hitung limit di atas untuk x menuju 1 dari kanan.
Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$limit((abs(x-1)/(x-1),x,plus,1))
```

$$1$$

```
>plot2d("((abs(x-1))/(x-1))",0.999,1.2):
```

```
>$showev('limit(sin(x)/x,x,0))
```

$$\lim_{x\to 0}\frac{\sin x}{x}=1$$

```
>plot2d("sin(x)/x",-pi,pi); plot2d(0,1,>points,style="ow",>add):
```

```
>$showev('limit(sin(x^3)/x,x,0))
```

$$\lim_{x\to 0} \frac{\sin x^3}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("sin(x^3)/x",-pi,pi); plot2d(0,0,>points,style="ow",>add):
```

```
>$showev('limit(log(x), x, minf))
```

$$\lim_{x\to -\infty} \log x = \mathit{infinity}$$

```
>$showev('limit((-2)^x,x, inf))
```

$$\lim_{x\to\infty} (-2)^x = \mathit{infinity}$$

```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

$$\lim_{t\uparrow 2} t - \sqrt{2-t} = 2$$

```
>$showev('limit(t-sqrt(2-t),t,2,plus))
```

$$\lim_{t\downarrow 2} t - \sqrt{2-t} = 2$$

```
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
```

$$\lim_{t\downarrow 5} t - \sqrt{2-t} = 5 - \sqrt{3}\,i$$

```
>plot2d("x-sqrt(2-x)",0,2):
```

```
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
```

$$\lim_{x\to 3}\frac{x^2-9}{2\,x^2-5\,x-3}=\frac{6}{7}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(x^2-9)/(2*x^2-5*x-3)",2,4); plot2d(3,6/7,>points,style="ow",>add):
```

```
>$showev('limit((1-cos(x))/x,x,0))
```

$$\lim_{x\to 0} \frac{1-\cos x}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1-cos(x))/x",-pi,pi); plot2d(0,0,>points,style="ow",>add):
```

```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

$$\lim_{x\to 0}\frac{|x|+x^2}{x^2-|x|}=-1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(abs(x)+x^2)/(x^2-abs(x))",-1,1); plot2d(0,-1,>points,style="ow",>add):
```

```
>$showev('limit((1+1/x)^x,x,inf))
```

$$\lim_{x\to\infty}\left(\frac{1}{x}+1\right)^{x}=e$$

```
>plot2d("(1+1/x)^x",0,1000):
```

```
>$showev('limit((1+k/x)^x,x,inf))
```

$$\lim_{x\to\infty}\left(\frac{k}{x}+1\right)^{x}=e^{k}$$

```
>$showev('limit((1+x)^(1/x),x,0))
```

$$\lim_{x\to 0}(x+1)^{\frac{1}{x}}=e$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(x+1)^(1/x)",-0.5,2); plot2d(0,E,>points,style="ow",>add):
```

```
>$showev('limit((x/(x+k))^x,x,inf))
```

$$\lim_{x\to\infty}\left(\frac{x}{x+k}\right)^{x}=e^{-k}$$

```
>$showev('limit((E^x-E^2)/(x-2),x,2))
```

$$\lim_{x\to 2}\frac{e^x-e^2}{x-2}=e^2$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(E^x-E^2)/(x-2)",1,3); plot2d(2,E^2,>points,style="ow",>add):
```

```
>$showev('limit(sin(1/x),x,0))
```

$$\lim_{x\to 0} \sin\left(\frac{1}{x}\right) = ind$$

```
>$showev('limit(sin(1/x),x,inf))
```

$$\lim_{x\to \infty} \sin\left(\frac{1}{x}\right) = 0$$

```
>plot2d("sin(1/x)",-0.001,0.001):
```

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

Turunan Fungsi

Definisi turunan:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>$showev('limit(((x+h)^2-x^2)/h,h,0)) // turunan x^2
```

$$\lim_{h \to 0} \frac{(x+h)^2 - x^2}{h} = 2\,x$$

```
>p &= expand((x+h)^2-x^2)|simplify; $p //pembilang dijabarkan dan disederhanakan
```

$$2\,h\,x + h^2$$

```
>q &=ratsimp(p/h); $q // ekspresi yang akan dihitung limitnya disederhanakan
```

$$2\,x+h$$

```
>$limit(q,h,0) // nilai limit sebagai turunan
```

$$2\,x$$

```
>$showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
```

$$\lim_{h\to 0}\frac{(x+h)^n-x^n}{h}=n\,x^{n-1}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan (x+h)ˆn dengan menggunakan teorema binomial.

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

$$\lim_{h\to 0}\frac{\sin(x+h)-\sin x}{h}=\cos x$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan sin(x+h) dengan menggunakan rumus jumlah dua sudut.

```
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
```

$$\lim_{h\to 0} \frac{\log(x+h) - \log x}{h} = \frac{1}{x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, gunakan sifat-sifat logaritma dan hasil limit pada bagian sebelumnya di atas.

```
>$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x
```

$$\lim_{h\to 0} \frac{\frac{1}{x+h} - \frac{1}{x}}{h} = -\frac{1}{x^2}$$

```
>$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x
```

```
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Answering "Is x an integer?" with "integer"
Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is x an integer?

Use assume!
Error in:
 $showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x ...
                                      ^
```

Maxima bermasalah dengan limit:

$$\lim_{h\to 0} \frac{e^{x+h}-e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h\to 0} \frac{e^h-1}{h} = 1$$

```
>$showev('factor(E^(x+h)-E^x))
```

$$factor\left(e^{x+h}-e^{x}\right)=\left(e^{h}-1\right)\,e^{x}$$

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

$$\left(\lim_{h\to 0}\frac{e^{h}-1}{h}\right)\,e^{x}=e^{x}$$

```
>function f(x) &= x^x
```

```
  x
x
```

```
>$showev('limit(f(x),x,0))
```

$$\lim_{x\to 0}x^{x}=1$$

Silakan Anda gambar kurva

$$y=x^{x}.$$

```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h\to 0}\frac{(x+h)^{x+h}-x^x}{h}=\mathit{infinity}$$

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h\to 0}\frac{(x+h)^{x+h}-x^x}{h}.$$

Dalam hal ini diperlukan asumsi nilai x.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$\lim_{h\to 0}\frac{(x+h)^{x+h}-x^x}{h}=x^x\ (\log x+1)$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

[x > 0]

```
>&forget(x<0)
```

[x < 0]

```
>&facts()
```

[]

```
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
```

$$\lim_{h\to 0}\frac{\arcsin(x+h)-\arcsin x}{h}=\frac{1}{\sqrt{1-x^2}}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

$$\lim_{h\to 0}\frac{\tan(x+h)-\tan x}{h}=\frac{1}{\cos^2 x}$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

sinh(x)

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

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

Hasilnya adalah cosh(x), karena

$$\frac{e^x + e^{-x}}{2} = \cosh(x).$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```

```
>function f(x) &= sin(3*x^5+7)^2
```

$$\sin^2(3\,x^5 + 7)$$

```
>diff(f,3), diffc(f,3)
```

```
1198.32948904
1198.72863721
```

Apakah perbedaan diff dan diffc?

```
>$showev('diff(f(x),x))
```

$$\frac{d}{d\,x}\,\sin^2\left(3\,x^5+7\right)=30\,x^4\,\cos\left(3\,x^5+7\right)\,\sin\left(3\,x^5+7\right)$$

```
>$% with x=3
```

$$\%at\left(\frac{d}{d\,x}\,\sin^2\left(3\,x^5+7\right),x=3\right)=2430\,\cos 736\,\sin 736$$

```
>$float(%)
```

$$\%at\left(\frac{d^{1.0}}{d\,x^{1.0}}\,\sin^2\left(3.0\,x^5+7.0\right),x=3.0\right)=1198.728637211748$$

```
>plot2d(f,0,3.1):
```

```
>function f(x) &=5*cos(2*x)-2*x*sin(2*x) // mendifinisikan fungsi f
```

$$5\cos(2x) - 2x\sin(2x)$$

```
>function df(x) &=diff(f(x),x) // fd(x) = f'(x)
```

```
- 12 sin(2 x) - 4 x cos(2 x)
```

```
>$'f(1)=f(1), $float(f(1)), $'f(2)=f(2), $float(f(2)) // nilai f(1) dan f(2)
```

$$-0.2410081230863468$$

```
>xp=solve("df(x)",1,2,0) // solusi f'(x)=0 pada interval [1, 2]
```

```
1.35822987384
```

```
>df(xp), f(xp) // cek bahwa f'(xp)=0 dan nilai ekstrim di titik tersebut
```

```
0
-5.67530133759
```

```
>plot2d(["f(x)","df(x)"],0,2*pi,color=[blue,red]): //grafik fungsi dan turunannya
```

Perhatikan titik-titik "puncak" grafik y=f(x) dan nilai turunan pada saat grafik fungsinya mencapai titik "puncak" tersebut.

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), menggunakan perintah diff, dan secara manual (langkah demi langkah yang dihitung dengan Maxima) seperti contoh-contoh di atas. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

Integral

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).

Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

$$\int_a^b f(x)\ dx = F(b) - F(a), \qquad \text{dengan } F'(x) = f(x).$$

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak integrabel, kita tidak akan menggunakan metode ini).

```
>$showev('integrate(x^n,x))
```

```
Answering "Is n equal to -1?" with "no"
```

$$\int x^n\ dx = \frac{x^{n+1}}{n+1}$$

```
>$showev('integrate(1/(1+x),x))
```

$$\int \frac{1}{x+1}\,dx = \log\left(x+1\right)$$

```
>$showev('integrate(1/(1+x^2),x))
```

$$\int \frac{1}{x^2+1}\,dx = \arctan x$$

```
>$showev('integrate(1/sqrt(1-x^2),x))
```

$$\int \frac{1}{\sqrt{1-x^2}}\,dx = \arcsin x$$

```
>$showev('integrate(sin(x),x,0,pi))
```

$$\int_0^{\pi} \sin x\,dx = 2$$

```
>plot2d("sin(x)",0,2*pi):
```

```
>$showev('integrate(sin(x),x,a,b))
```

$$\int_a^b \sin x \, dx = \cos a - \cos b$$

```
>$showev('integrate(x^n,x,a,b))
```

```
Answering "Is n positive, negative or zero?" with "positive"
```

$$\int_a^b x^n \, dx = \frac{b^{n+1}}{n+1} - \frac{a^{n+1}}{n+1}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
```

$$\int x^2\,\sqrt{2\,x+1}\,dx=\frac{(2\,x+1)^{\frac{7}{2}}}{28}-\frac{(2\,x+1)^{\frac{5}{2}}}{10}+\frac{(2\,x+1)^{\frac{3}{2}}}{12}$$

```
>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
```

$$\int_0^2 x^2\,\sqrt{2\,x+1}\,dx=\frac{2\,5^{\frac{5}{2}}}{21}-\frac{2}{105}$$

```
>$ratsimp(%)
```

$$\int_0^2 x^2\,\sqrt{2\,x+1}\,dx=\frac{2\,5^{\frac{7}{2}}-2}{105}$$

```
>$showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
```

$$\int_0^{\pi^2}\frac{\sin\left(\sqrt{x}+a\right)\,e^{\sqrt{x}}}{\sqrt{x}}\,dx=\left(-e^{\pi}-1\right)\,\sin a+\left(e^{\pi}+1\right)\,\cos a$$

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

$$\int_0^{\pi^2} \frac{\sin\left(\sqrt{x}+a\right)\, e^{\sqrt{x}}}{\sqrt{x}}\, dx = \left(-e^{\pi}-1\right)\,\left(\sin a - \cos a\right)$$

```
>function map f(x) &= E^(-x^2)
```

```
      2
   - x
E
```

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

$$\int e^{-x^2}\, dx = \frac{\sqrt{\pi}\,\mathrm{erf}\left(x\right)}{2}$$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

$$erf(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}}\, dx.$$

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

$$\int_0^{\pi} e^{-x^2}\, dx$$

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```

Integral tentu

$$\int_0^{\pi} e^{-x^2}\, dx$$

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva y=f(x) tersebut. Langkah-langkahnya adalah sebagai berikut.

```
>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-nilai x
>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)
>// jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!
```

Hasilnya adalah:

$$\int_0^{\pi} e^{-x^2}\, dx = 0.8362196102528469$$

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai f(x) untuk x = 0.1, 0.2, 0.3, ..., 3.2.

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

```
0.836219610253
```

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```

$$\int_0^{\infty} e^{-x^2}\,dx = \frac{\sqrt{\pi}}{2}$$

Tunjukkan kebenaran hasil di atas!

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

```
>function f(x) &= x^x
```

```
 x
x
```

```
>$showev('integrate(f(x),x,0,1))
```

$$\int_0^1 x^x \, dx = \int_0^1 x^x \, dx$$

```
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```

Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
```

$$\int_0^1 x^x \, dx = 0.7834935879025506$$

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
>function f(x) &= sin(3*x^5+7)^2
```

```
   2    5
sin (3 x  + 7)
```

```
>integrate(f,0,1)
```

```
0.542581176074
```

```
>&showev('integrate(f(x),x,0,1))
```

```
 1                          1              pi
/                     gamma(-) sin(14) sin(--)
[     2    5                5              10
I  sin (3 x  + 7) dx = ------------------------
]                                  1/5
/                             10 6
```

```
         0
       4/5                  1          4/5                  1
  - (((6    gamma_incomplete(-, 6 I) + 6    gamma_incomplete(-, - 6 I))
                            5                                5
            4/5                    1
  sin(14) + (6    I gamma_incomplete(-, 6 I)
                                   5
     4/5                    1                        pi
  - 6    I gamma_incomplete(-, - 6 I)) cos(14)) sin(--) - 60)/120
                            5                        10
```

```
>&float(%)
```

```
          1.0
         /
         [       2      5
         I    sin (3.0 x  + 7.0) dx =
         ]
         /
          0.0
0.09820784258795788 - 0.008333333333333333
 (0.3090169943749474 (0.1367372182078336
 (4.192962712629476 I gamma__incomplete(0.2, 6.0 I)
 - 4.192962712629476 I gamma__incomplete(0.2, - 6.0 I))
 + 0.9906073556948704 (4.192962712629476 gamma__incomplete(0.2, 6.0 I)
 + 4.192962712629476 gamma__incomplete(0.2, - 6.0 I))) - 60.0)
```

```
>$showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)
```

$$\int_0^1 x\,e^{-x}\,dx = 1 - 2\,e^{-1}$$

Aplikasi Integral Tentu

```
>plot2d("x^3-x",-0.1,1.1); plot2d("-x^2",>add);  ...
>b=solve("x^3-x+x^2",0.5); x=linspace(0,b,200); xi=flipx(x); ...
>plot2d(x|xi,x^3-x|-xi^2,>filled,style="|",fillcolor=1,>add): // Plot daerah antara 2 kurva
```

```
>a=solve("x^3-x+x^2",0), b=solve("x^3-x+x^2",1) // absis titik-titik potong kedua kurva
```

```
0
0.61803398875
```

```
>integrate("(-x^2)-(x^3-x)",a,b) // luas daerah yang diarsir
```

```
0.0758191713542
```

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```
>a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik potong kedua kurva secara eksak
```

$$\left[x=\frac{-\sqrt{5}-1}{2},x=\frac{\sqrt{5}-1}{2},x=0\right]$$

```
>$showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara eksak
```

$$\int_{0}^{\frac{\sqrt{5}-1}{2}}-x^{3}-x^{2}+x\,dx=\frac{13-5^{\frac{3}{2}}}{24}$$

```
>$float(%)
```

$$\int_{0.0}^{0.6180339887498949}-1.0\,x^{3}-1.0\,x^{2}+x\,dx=0.07581917135421037$$

Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

$$\gamma(t) = (r(t)\cos(t), r(t)\sin(t))$$

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}, \quad 0 \leq t \leq 2\pi.$$

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
>plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5): // Kita gambar kurvanya terlebih dahulu
```

```
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
```

$r\left([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.2200000\right.$

```
>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
```

$fx\left([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.2200000\right.$

```
>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
```

$fy\left([0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19,0.2,0.21,0.2200000\right.$

```
>function ds(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t)
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t ...
                                                      ^
```

```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
```

$$\int_0^{2\,\pi} ds\left(x\right)\,dx$$

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara umerik dengan perintah EMT.

```
>integrate("ds(x)",0,2*pi)
```

```
Function ds not found.
Try list ... to find functions!
Error in expression: ds(x)
 %mapexpression1:
    return expr(x,args());
Error in map.
 %evalexpression:
    if maps then return %mapexpression1(x,f$;args());
gauss:
    if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());
adaptivegauss:
    t1=gauss(f$,c,c+h;args(),=maps);
Try "trace errors" to inspect local variables after errors.
integrate:
    return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
```

Spiral Logaritmik

$$x = e^{ax}\cos x,\ y = e^{ax}\sin x.$$

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi):
```

```
>&kill(a) // hapus expresi a
```

done

```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
```

fx ([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.2200000

```
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
```

$fy([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.2200000$

```
>function df(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t)
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t ...
                                                      ^
```

```
>S &=integrate(df(t),t,0,2*%pi); $S // panjang kurva (spiral)
```

```
Maxima said:
defint: variable of integration cannot be a constant; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
S &=integrate(df(t),t,0,2*%pi); $S // panjang kurva (spiral) ...
                               ^
```

```
>S(a=0.1) // Panjang kurva untuk a=0.1
```

```
Function S not found.
Try list ... to find functions!
Error in:
S(a=0.1) // Panjang kurva untuk a=0.1 ...
        ^
```

Soal:

Tunjukkan bahwa keliling lingkaran dengan jari-jari r adalah K=2.pi.r.

Berikut adalah contoh menghitung panjang parabola.

```
>plot2d("x^2",xmin=-1,xmax=1):
```

```
>$showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))
```

$$\int_{-1}^{1} \sqrt{4\,x^2+1}\,dx = \frac{\operatorname{asinh} 2+2\,\sqrt{5}}{2}$$

```
>$float(%)
```

$$\int_{-1.0}^{1.0} \sqrt{4.0\,x^2+1.0}\,dx = 2.957885715089195$$

```
>x=-1:0.2:1; y=x^2; plot2d(x,y);  ...
>  plot2d(x,y,points=1,style="o#",add=1):
```

Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```
>i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])^2+(y[i+1]-y[i])^2))
```

```
2.95191957027
```

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

Koordinat Kartesius

Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

$$x^3 + y^3 - 3xy = 0.$$

```
>z &= x^3+y^3-3*x*y; $z
```

$$y^3 - 3\,x\,y + x^3$$

```
>plot2d(z,r=2,level=0,n=100):
```

Kita tertarik pada kurva di kuadran pertama.

```
>plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="/"):
```

Kita selesaikan persamaannya untuk x.

```
>$z with y=l*x, sol &= solve(%,x); $sol
```

$$\left[x=\frac{3\,l}{l^3+1},x=0\right]$$

$$\left[x=\frac{3\,l}{l^3+1},x=0\right]$$

Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); $'f(l)=f(l)
```

$$f(l) = \frac{3\,l}{l^3 + 1}$$

Fungsi tersebut juga dapat digunaka untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai y=l*x, yakni x=f(l) dan y=l*f(l).

```
>plot2d(&f(x),&x*f(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):
```

Elemen panjang kurva adalah:

$$ds = \sqrt{f'(l)^2 + (lf'(l) + f(l))^2}.$$

```
>function ds(l) &= ratsimp(sqrt(diff(f(l),l)^2+diff(l*f(l),l)^2)); $'ds(l)=ds(l)
```

$$ds\left(l\right) = \frac{\sqrt{9\,l^8 + 36\,l^6 - 36\,l^5 - 36\,l^3 + 36\,l^2 + 9}}{\sqrt{l^{12} + 4\,l^9 + 6\,l^6 + 4\,l^3 + 1}}$$

```
>$integrate(ds(l),l,0,1)
```

$$\int_0^1 \frac{\sqrt{9\,l^8 + 36\,l^6 - 36\,l^5 - 36\,l^3 + 36\,l^2 + 9}}{\sqrt{l^{12} + 4\,l^9 + 6\,l^6 + 4\,l^3 + 1}}\,dl$$

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral etrsebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
>2*integrate("ds(x)",0,1)
```

```
4.91748872168
```

```
>2*romberg(&ds(x),0,1)// perintah Euler lain untuk menghitung nilai hampiran integral yang sama
```

```
4.91748872168
```

Perhitungan di datas dapat dilakukan untuk sebarang fungsi x dan y dengan mendefinisikan fungsi EMT, misalnya kita beri nama panjangkurva. Fungsi ini selalu memanggil Maxima untuk menurunkan fungsi yang diberikan.

```
>function panjangkurva(fx,fy,a,b) ...
```

```
ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");
return romberg(ds,a,b);
endfunction
```

```
>panjangkurva("x","x^2",-1,1) // cek untuk menghitung panjang kurva parabola sebelumnya
```

```
2.95788571509
```

Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)"),mxm("x*f(x)"),0,1) // cek contoh terakhir, bandingkan hasilnya!
```

```
4.91748872168
```

Kita hitung panjang spiral Archimides berikut ini dengan fungsi tersebut.

```
>plot2d("x*cos(x)","x*sin(x)",xmin=0,xmax=2*pi,square=1):
```

```
>panjangkurva("x*cos(x)","x*sin(x)",0,2*pi)
```

```
21.2562941482
```

Berikut kita definisikan fungsi yang sama namun dengan Maxima, untuk perhitungan eksak.

```
>&kill(ds,x,fx,fy)
```

```
done
```

```
>function ds(fx,fy) &&= sqrt(diff(fx,x)^2+diff(fy,x)^2)
```

```
                    2                2
sqrt(diff (fy, x) + diff (fx, x))
```

```
>sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk menghitung panjang kurva terakhir di atas
```

$$\sqrt{(\cos x - x\,\sin x)^2 + (\sin x + x\,\cos x)^2}$$

```
>$sol | trigreduce | expand, $integrate(%,x,0,2*pi), %()
```

$$\frac{\operatorname{asinh}\,(2\,\pi) + 2\,\pi\,\sqrt{4\,\pi^2 + 1}}{2}$$

$$\frac{\operatorname{asinh}\,(2\,\pi)+2\,\pi\,\sqrt{4\,\pi^2+1}}{2}$$

```
21.2562941482
```

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.

Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
>plot2d("3*x^2-1","3*x^3-1",xmin=-1/sqrt(3),xmax=1/sqrt(3),square=1):
```

```
>sol &= radcan(ds(3*x^2-1,3*x^3-1)); $sol
```

$$3\,x\,\sqrt{9\,x^2+4}$$

```
>$showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%) // panjang kurva di atas
```

$$6.0\int_{0.0}^{0.5773502691896258} x\,\sqrt{9.0\,x^2+4.0}\,dx = 2.337835372767141$$

$$6.0\int_{0.0}^{0.5773502691896258} x\,\sqrt{9.0\,x^2+4.0}\,dx = 2.337835372767141$$

Sikloid

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar ke kanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah r. Posisi titik pusat lingkaran pada saat t adalah:

$$(rt, r).$$

Misalkan posisi titik pada lingkaran tersebut mula-mula (0,0) dan posisinya pada saat t adalah:

$$(r(t-\sin(t)), r(1-\cos(t))).$$

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika t=0, t=pi/2, t=r*pi.

```
>x &= r*(t-sin(t))
```

```
        [0, 1.66665833335744e-7 r, 1.33330666692022e-6 r,
4.499797504338432e-6 r, 1.066581336583994e-5 r,
2.083072932167196e-5 r, 3.599352055540239e-5 r,
5.71526624672386e-5 r, 8.530603082730626e-5 r,
1.214508019889565e-4 r, 1.665833531718508e-4 r,
2.216991628251896e-4 r, 2.877927110806339e-4 r,
3.658573803051457e-4 r, 4.568853557635201e-4 r,
5.618675264007778e-4 r, 6.817933857540259e-4 r,
8.176509330039827e-4 r, 9.704265741758145e-4 r,
0.001141105023499428 r, 0.001330669204938795 r,
0.001540100153900437 r, 0.001770376919130678 r,
0.002022476464811601 r, 0.002297373572865413 r,
0.002596040745477063 r, 0.002919448107844891 r,
0.003268563311168871 r, 0.003644351435886262 r,
0.004047774895164447 r, 0.004479793338660443 r, 0.0049413635565565 r,
0.005433439383882244 r, 0.005956971605131645 r,
0.006512907859185624 r, 0.007102192544548636 r,
0.007725766724910044 r, 0.00838456803503801 r,
0.009079530587017326 r, 0.009811584876838586 r, 0.0105816576913495 r,
0.01139067201557714 r, 0.01223954694042984 r, 0.01312919757078923 r,
0.01406053493400045 r, 0.01503446588876983 r, 0.01605189303448024 r,
0.01711371462093175 r, 0.01822082445851714 r, 0.01937411182884202 r,
0.02057446139579705 r, 0.02182275311709253 r, 0.02311986215626333 r,
0.02446665879515308 r, 0.02586400834688696 r, 0.02731277106934082 r,
0.02881380207911666 r, 0.03036795126603076 r, 0.03197606320812652 r,
0.0336389770872163 r, 0.03535752660496472 r, 0.03713253989951881 r,
0.03896483946269502 r, 0.0408552420577305 r, 0.04280455863760801 r,
0.04481359426396048 r, 0.04688314802656623 r, 0.04901401296344043 r,
```

```
0.05120697598153157 r, 0.05346281777803219 r, 0.05578231276230905 r,
0.05816622897846346 r, 0.06061532802852698 r, 0.0631303649963022 r,
0.06571208837185505 r, 0.06836123997666599 r, 0.07107855488944881 r,
0.07386476137264342 r, 0.07672058079958999 r, 0.07964672758239233 r,
0.08264390910047736 r, 0.0857128256298576 r, 0.08885417027310427 r,
0.09206862889003742 r, 0.09535688002914089 r, 0.0987195948597075 r,
0.1021574371047232 r, 0.1056710629744951 r, 0.1092611211010309 r,
0.1129282524731764 r, 0.1166730903725168 r, 0.1204962603100498 r,
0.1243983799636342 r, 0.1283800591162231 r, 0.1324418995948859 r,
0.1365844952106265 r, 0.140808431699002 r, 0.1451142866615502 r,
0.1495026295080298 r, 0.1539740213994798 r]
```

```
>y &= r*(1-cos(t))
```

```
        [0, 4.999958333473664e-5 r, 1.999933334222437e-4 r,
4.499662510124569e-4 r, 7.998933390220841e-4 r,
0.001249739605033717 r, 0.00179946006479581 r,
0.002448999746720415 r, 0.003198293697380561 r,
0.004047266988005727 r, 0.004995834721974179 r,
0.006043902043303184 r, 0.00719136414613375 r, 0.00843810628521191 r,
0.009784003787362772 r, 0.01122892206395776 r, 0.01277271662437307 r,
0.01441523309043924 r, 0.01615630721187855 r, 0.01799576488272969 r,
0.01993342215875837 r, 0.02196908527585173 r, 0.02410255066939448 r,
0.02633360499462523 r, 0.02866202514797045 r, 0.03108757828935527 r,
0.03361002186548678 r, 0.03622910363410947 r, 0.03894456168922911 r,
0.04175612448730281 r, 0.04466351087439402 r, 0.04766643011428662 r,
0.05076458191755917 r, 0.0539576564716131 r, 0.05724533447165381 r,
0.06062728715262111 r, 0.06410317632206519 r, 0.06767265439396564 r,
0.07133536442348987 r, 0.07509094014268702 r, 0.07893900599711501 r,
0.08287917718339499 r, 0.08691105968769186 r, 0.09103425032511492 r,
```

```
0.09524833678003664 r, 0.09955289764732322 r, 0.1039475024744748 r,
0.1084317118046711 r, 0.113005077220716 r, 0.1176671413898787 r,
0.1224174381096274 r, 0.1272554923542488 r, 0.1321808203223502 r,
0.1371929294852391 r, 0.1422913186361759 r, 0.1474754779404944 r,
0.152744888986584 r, 0.1580990248377314 r, 0.1635373500848132 r,
0.1690593208998367 r, 0.1746643850903219 r, 0.1803519821545206 r,
0.1861215433374662 r, 0.1919724916878484 r, 0.1979042421157076 r,
0.2039162014509444 r, 0.2100077685026351 r, 0.216178334119151 r,
0.2224272812490723 r, 0.2287539850028937 r, 0.2351578127155118 r,
0.2416381240094921 r, 0.2481942708591053 r, 0.2548255976551299 r,
0.2615314412704124 r, 0.2683111311261794 r, 0.2751639892590951 r,
0.2820893303890569 r, 0.2890864619877229 r, 0.2961546843477643 r,
0.3032932906528349 r, 0.3105015670482534 r, 0.3177787927123868 r,
0.3251242399287333 r, 0.3325371741586922 r, 0.3400168541150183 r,
0.3475625318359485 r, 0.3551734527599992 r, 0.3628488558014202 r,
0.3705879734263036 r, 0.3783900317293359 r, 0.3862542505111889 r,
0.3941798433565377 r, 0.4021660177127022 r, 0.4102119749689023 r,
0.418316910536117 r, 0.4264800139275439 r, 0.4347004688396462 r,
0.4429774532337832 r, 0.451310139418413 r]
```

Berikut kita gambar sikloid untuk r=1.

```
>ex &= x-sin(x); ey &= 1-cos(x); aspect(1);
>plot2d(ex,ey,xmin=0,xmax=4pi,square=1); ...
>  plot2d("2+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
>  plot2d([2,ex(2)],[1,ey(2)],color=red,>add); ...
>  plot2d(ex(2),ey(2),>points,>add,color=red); ...
>  plot2d("2pi+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
>  plot2d([2pi,ex(2pi)],[1,ey(2pi)],color=red,>add);  ...
>  plot2d(ex(2pi),ey(2pi),>points,>add,color=red):
```

```
Error : [0,1.66665833335744e-7*r-sin(1.66665833335744e-7*r),1.33330666692022e-6*r-sin(1.3333066669

Error generated by error() command

adaptiveeval:
    error(f$|" does not produce a real or column vector");
Try "trace errors" to inspect local variables after errors.
plot2d:
    dw/n,dw/n^2,dw/n;args());
```

Berikut dihitung panjang lintasan untuk 1 putaran penuh. (Jangan salah menduga bahwa panjang lintasan 1 putaran penuh sama dengan keliling lingkaran!)

```
>ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds) // elemen panjang kurva sikloid
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds ...
                                              ^
```

```
>ds &= trigsimp(ds); $ds
>$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu putaran penuh
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate(ds,[0,1.66665833335744e-7*r,1.33330666692022e-6*r,4.499797504338432e-6*r,1
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid sat ...
                                 ^
```

```
>integrate(mxm("ds"),0,2*pi) // hitung secara numerik
```

```
Illegal function result in map.
 %evalexpression:
    if maps then return %mapexpression1(x,f$;args());
gauss:
    if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());
adaptivegauss:
    t1=gauss(f$,c,c+h;args(),=maps);
Try "trace errors" to inspect local variables after errors.
integrate:
    return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
```

```
>romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik
```

```
Wrong argument!

Cannot combine a symbolic expression here.
Did you want to create a symbolic expression?
```

```
Then start with &.

Try "trace errors" to inspect local variables after errors.
romberg:
    if cols(y)==1 then return y*(b-a); endif;
Error in:
romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik ...
                         ^
```

Perhatikan, seperti terlihat pada gambar, panjang sikloid lebih besar daripada keliling lingkarannya, yakni:

$$2\pi.$$

Kurvatur (Kelengkungan) Kurva

image: Osculating.png

Aslinya, kelengkungan kurva diferensiabel (yakni, kurva mulus yang tidak lancip) di titik P didefinisikan melalui lingkaran oskulasi (yaitu, lingkaran yang melalui titik P dan terbaik memperkirakan, paling banyak menyinggung kurva di sekitar P). Pusat dan radius kelengkungan kurva di P adalah pusat dan radius lingkaran oskulasi. Kelengkungan adalah kebalikan dari radius kelengkungan:

$$\kappa = \frac{1}{R}$$

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar 2pi sejauh 2piR.)
Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekivalen.

Definisi Kurvatur dengan Fungsi Parametrik Panjang Kurva

Setiap kurva diferensiabel dapat dinyatakan dengan persamaan parametrik terhadap panjang kurva s:

$$\gamma(s) = (x(s),\ y(s)),$$

dengan x dan y adalah fungsi riil yang diferensiabel, yang memenuhi:

$$\|\gamma'(s)\| = \sqrt{x'(s)^2 + y'(s)^2} = 1.$$

Ini berarti bahwa vektor singgung

$$\mathbf{T}(s) = (x'(s),\ y'(s))$$

memiliki norm 1 dan merupakan vektor singgung satuan.

Apabila kurvanya memiliki turunan kedua, artinya turunan kedua x dan y ada, maka T'(s) ada. Vektor ini merupakan normal kurva yang arahnya menuju pusat kurvatur, norm-nya merupakan nilai kurvatur (kelengkungan):

$$\begin{aligned}
\mathbf{T}(s) &= \gamma'(s), \\
\mathbf{T}^2(s) &= 1 \text{ (konstanta)} \Rightarrow \mathbf{T}'(s) \cdot \mathbf{T}(s) = 0 \\
\kappa(s) &= \|\mathbf{T}'(s)\| = \|\gamma''(s)\| = \sqrt{x''(s)^2 + y''(s)^2}.
\end{aligned}$$

Nilai

$$R(s) = \frac{1}{\kappa(s)}$$

disebut jari-jari (radius) kelengkungan kurva.

Bilangan riil

$$k(s) = \pm\kappa(s)$$

disebut nilai kelengkungan bertanda.

Contoh:
Akan ditentukan kurvatur lingkaran

$$x = r\cos t,\ y = r\sin t.$$

```
>fx &= r*cos(t); fy &=r*sin(t);
>&assume(t>0,r>0); s &=integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s // elemen panjang kurva,
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
... =integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s // elemen ...
                                                       ^
```

```
>&kill(s); fx &= r*cos(s/r); fy &=r*sin(s/r); // definisi ulang persamaan parametrik terhadap s deng
>k &= trigsimp(sqrt(diff(fx,s,2)^2+diff(fy,s,2)^2)); $k // nilai kurvatur lingkaran dengan menggunak
```

$$\frac{1}{r}$$

Untuk representasi parametrik umum, misalkan

$$x = x(t),\ y = y(t)$$

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

$$\begin{aligned}\kappa &= \frac{d\phi}{ds} = \frac{\frac{d\phi}{dt}}{\frac{ds}{dt}} \quad (\phi \text{ adalah sudut kemiringan garis singgung dan } s \text{ adalah panjang kurva}) \\ &= \frac{\frac{d\phi}{dt}}{\sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2}} = \frac{\frac{d\phi}{dt}}{\sqrt{x'(t)^2 + y'(t)^2}}.\end{aligned}$$

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

$$\sec^2\phi \frac{d\phi}{dt} = \frac{d}{dt}(\tan\phi) = \frac{d}{dt}\left(\frac{dy}{dx}\right) = \frac{d}{dt}\left(\frac{dy/dt}{dx/dt}\right) = \frac{d}{dt}\left(\frac{y'(t)}{x'(t)}\right) = \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2}.$$

$$\begin{aligned}\frac{d\phi}{dt} &= \frac{1}{\sec^2\phi}\frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1+\tan^2\phi}\frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{1}{1+\left(\frac{y'(t)}{x'(t)}\right)^2}\frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \\ &= \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2 + y'(t)^2}.\end{aligned}$$

Jadi, rumus kurvatur untuk kurva parametrik

$$x = x(t),\ y = y(t)$$

adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{\left(x'(t)^2 + y'(t)^2\right)^{3/2}}.$$

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kutub

$$x = r(\theta)\cos\theta,\ \ y = r(\theta)\sin\theta,$$

maka rumus kurvaturnya adalah

$$\kappa(\theta) = \frac{r(\theta)^2 + 2r'(\theta)^2 - r(\theta)r''(\theta)}{\left(r'(\theta)^2 + r'(\theta)^2\right)^{3/2}}.$$

(Silakan Anda turunkan rumus tersebut!)

Contoh:
Lingkaran dengan pusat (0,0) dan jari-jari r dapat dinyatakan dengan persamaan parametrik

$$x = r\cos t,\ \ y = r\sin t.$$

Nilai kelengkungan lingkaran tersebut adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{\left(x'(t)^2 + y'(t)^2\right)^{3/2}} = \frac{r^2}{r^3} = \frac{1}{r}.$$

Hasil cocok dengan definisi kurvatur suatu kelengkungan.

Kurva

$$y = f(x)$$

dapat dinyatakan ke dalam persamaan parametrik

$$x = t,\ y = f(t), \quad \text{dengan } x'(t) = 1,\ x''(t) = 0,$$

sehingga kurvaturnya adalah

$$\kappa(t) = \frac{y''(t)}{\left(1 + y'(t)^2\right)^{3/2}}.$$

Contoh:
Akan ditentukan kurvatur parabola

$$y = ax^2 + bx + c.$$

```
>function f(x) &= a*x^2+b*x+c; $y=f(x)
```

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) // kelengkungan parabola
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
... (x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x)  ...
                                                     ^
```

```
>function f(x) &= x^2+x+1; $y=f(x) // akan kita plot kelengkungan parabola untuk a=b=c=1
```

```
>function k(x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x) // kelengkungan parabola
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
... (x) &= (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); $'k(x)=k(x)  ...
                                                     ^
```

Berikut kita gambar parabola tersebut beserta kurva kelengkungan, kurva jari-jari kelengkungan dan salah satu lingkaran oskulasi di titik puncak parabola. Perhatikan, puncak parabola dan jari-jari lingkaran oskulasi di puncak parabola adalah

$$(-1/2, 3/4),\ 1/k(2) = 1/2,$$

sehingga pusat lingkaran oskulasi adalah (-1/2, 5/4).

```
>plot2d(["f(x)", "k(x)"],-2,1, color=[blue,red]); plot2d("1/k(x)",-1.5,1,color=green,>add); ...
>plot2d("-1/2+1/k(-1/2)*cos(x)","5/4+1/k(-1/2)*sin(x)",xmin=0,xmax=2pi,>add,color=blue):
```

```
Error : f(x) does not produce a real or column vector

Error generated by error() command

 %ploteval:
    error(f$|" does not produce a real or column vector");
adaptiveevalone:
    s=%ploteval(g$,t;args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    dw/n,dw/n^2,dw/n,auto;args());
```

Untuk kurva yang dinyatakan dengan fungsi implisit

$$F(x, y) = 0$$

dengan turunan-turunan parsial

$$F_x = \frac{\partial F}{\partial x},\ F_y = \frac{\partial F}{\partial y},\ F_{xy} = \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial x}\right),\ F_{xx} = \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial x}\right),\ F_{yy} = \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial y}\right),$$

berlaku

$$F_x dx + F_y dy = 0 \text{ atau } \frac{dy}{dx} = -\frac{F_x}{F_y},$$

sehingga kurvaturnya adalah

$$\kappa = \frac{F_y^2 F_{xx} - 2F_x F_y F_{xy} + F_x^2 F_{yy}}{\left(F_x^2 + F_y^2\right)^{3/2}}.$$

(Silakan Anda turunkan sendiri!)

Contoh 1:
Parabola

$$y = ax^2 + bx + c$$

dapat dinyatakan ke dalam persamaan implisit

$$ax^2 + bx + c - y = 0.$$

```
>function F(x,y) &=a*x^2+b*x+c-y; $F(x,y)
```

```
>Fx &= diff(F(x,y),x), Fxx &=diff(F(x,y),x,2), Fy &=diff(F(x,y),y), Fxy &=diff(diff(F(x,y),x),y), Fy
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
Fx &= diff(F(x,y),x), Fxx &=diff(F(x,y),x,2), Fy &=diff(F(x,y) ...
                    ^
```

```
>function k(x) &= (Fy^2*Fxx-2*Fx*Fy*Fxy+Fx^2*Fyy)/(Fx^2+Fy^2)^(3/2); $'k(x)=k(x) // kurvatur parabol
```

$k\left(\left[0, 1.66665833335744 \times 10^{-7}\, r, 1.33330666692022 \times 10^{-6}\, r, 4.499797504338432 \times 10^{-6}\, r, 1.066581336583994 \times 10^{-5}\, r, 2\right.\right.$

Hasilnya sama dengan sebelumnya yang menggunakan persamaan parabola biasa.

Latihan

- Bukalah buku Kalkulus.
- Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi).
- Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah integral tentu dengan batas-batas yang menarik (Anda tentukan sendiri), seperti contoh-contoh tersebut.
- Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat diintegralkan (cari sedikitnya 3 fungsi).
- Gambar grafik fungsi dan daerah integrasinya pada sumbu koordinat yang sama.
- Gunakan integral tentu untuk mencari luas daerah yang dibatasi oleh dua kurva yang berpotongan di dua titik. (Cari dan gambar kedua kurva dan arsir (warnai) daerah yang dibatasi oleh keduanya.)
- Gunakan integral tentu untuk menghitung volume benda putar kurva y= f(x) yang diputar mengelilingi sumbu x dari x=a sampai x=b, yakni

$$V = \int_a^b \pi(f(x)^2 \; dx.$$

(Pilih fungsinya dan gambar kurva dan benda putar yang dihasilkan. Anda dapat mencari contoh-contoh bagaimana cara menggambar benda hasil perputaran suatu kurva.)
- Gunakan integral tentu untuk menghitung panjang kurva y=f(x) dari x=a sampai x=b dengan menggunakan rumus:

$$S = \int_a^b \sqrt{1 + (f'(x))^2} \; dx.$$

(Pilih fungsi dan gambar kurvanya.)
- Apabila fungsi dinyatakan dalam koordinat kutub x=f(r,t), y=g(r,t), r=h(t), x=a bersesuaian dengan t=t0 dan x=b bersesuian dengan t=t1, maka rumus di atas akan menjadi:

$$S = \int_{t_0}^{t_1} \sqrt{x'(t)^2 + y'(t)^2} \; dt.$$

- Pilih beberapa kurva menarik (selain lingkaran dan parabola) dari buku kalkulus. Nyatakan setiap kurva tersebut dalam bentuk:

```
a. koordinat Kartesius (persamaan y=f(x))
b. koordinat kutub ( r=r(theta))
c. persamaan parametrik x=x(t), y=y(t)
d. persamaan implit F(x,y)=0
```

- Tentukan kurvatur masing-masing kurva dengan menggunakan keempat representasi tersebut (hasilnya harus sama).
- Gambarlah kurva asli, kurva kurvatur, kurva jari-jari lingkaran oskulasi, dan salah satu lingkaran oskulasinya.

Barisan dan Deret

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh pengguanaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

- dengan cara yang sama seperti mendefinisikan vektor dengan elemen-elemen beraturan (menggunakan titik dua ":");
- menggunakan perintah "sequence" dan rumus barisan (suku ke -n);
- menggunakan perintah "iterate" atau "niterate";
- menggunakan fungsi Maxima "create_list" atau "makelist" untuk menghasilkan barisan simbolik;
- menggunakan fungsi biasa yang inputnya vektor atau barisan;
- menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

- sum: menghitung jumlah semua elemen suatu barisan
- cumsum: jumlah kumulatif suatu barisan
- differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

```
>1:10 // barisan sederhana
```

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

```
>1:2:30
```

```
[1,  3,  5,  7,  9,  11,  13,  15,  17,  19,  21,  23,  25,  27,  29]
```

Iterasi dan Barisan

EMT menyediakan fungsi iterate("g(x)", x0, n) untuk melakukan iterasi

$$x_{k+1} = g(x_k),\ x_0 = x_0, k = 1, 2, 3, ..., n.$$

Berikut ini disajikan contoh-contoh penggunaan iterasi dan rekursi dengan EMT. Contoh pertama menunjukkan pertumbuhan dari nilai awal 1000 dengan laju pertambahan 5%, selama 10 periode.

```
>q=1.05; iterate("x*q",1000,n=10)'
```

```
           1000
           1050
         1102.5
        1157.63
        1215.51
        1276.28
         1340.1
         1407.1
        1477.46
        1551.33
        1628.89
```

Contoh berikutnya memperlihatkan bahaya menabung di bank pada masa sekarang! Dengan bunga tabungan sebesar 6% per tahun atau 0.5% per bulan dipotong pajak 20%, dan biaya administrasi 10000 per bulan, tabungan sebesar 1 juta tanpa diambil selama sekitar 10 tahunan akan habis diambil oleh bank!

```
>r=0.005; plot2d(iterate("(1+0.8*r)*x-10000",1000000,n=130)):
```

Silakan Anda coba-coba, dengan tabungan minimal berapa agar tidak akan habis diambil oleh bank dengan ketentuan bunga dan biaya administrasi seperti di atas.

Berikut adalah perhitungan minimal tabungan agar aman di bank dengan bunga sebesar r dan biaya administrasi a, pajak bunga 20%.

```
>$solve(0.8*r*A-a,A), $% with [r=0.005, a=10]
```

$$[A = 2500.0]$$

Berikut didefinisikan fungsi untuk menghitung saldo tabungan, kemudian dilakukan iterasi.

```
>function saldo(x,r,a) := round((1+0.8*r)*x-a,2);
>iterate({{"saldo",0.005,10}},1000,n=6)
```

```
[1000,  994,  987.98,  981.93,  975.86,  969.76,  963.64]
```

```
>iterate({{"saldo",0.005,10}},2000,n=6)
```

```
[2000,  1998,  1995.99,  1993.97,  1991.95,  1989.92,  1987.88]
```

```
>iterate({{"saldo",0.005,10}},2500,n=6)
```

```
[2500,  2500,  2500,  2500,  2500,  2500,  2500]
```

Tabungan senilai 2,5 juta akan aman dan tidak akan berubah nilai (jika tidak ada penarikan), sedangkan jika tabungan awal kurang dari 2,5 juta, lama kelamaan akan berkurang meskipun tidak pernah dilakukan penarikan uang tabungan.

```
>iterate({{"saldo",0.005,10}},3000,n=6)
```

```
[3000,  3002,  3004.01,  3006.03,  3008.05,  3010.08,  3012.12]
```

Tabungan yang lebih dari 2,5 juta baru akan bertambah jika tidak ada penarikan.

Untuk barisan yang lebih kompleks dapat digunakan fungsi "sequence()". Fungsi ini menghitung nilai-nilai x[n] dari semua nilai sebelumnya, x[1],...,x[n-1] yang diketahui.
Berikut adalah contoh barisan Fibonacci.

$$x_n = x_{n-1} + x_{n-2}, \quad x_1 = 1, \quad x_2 = 1$$

```
>sequence("x[n-1]+x[n-2]",[1,1],15)
```

```
[1,  1,  2,  3,  5,  8,  13,  21,  34,  55,  89,  144,  233,  377,  610]
```

Barisan Fibonacci memiliki banyak sifat menarik, salah satunya adalah akar pangkat ke-n suku ke-n akan konvergen ke pecahan emas:

```
>$'(1+sqrt(5))/2=float((1+sqrt(5))/2)
```

$$\frac{\sqrt{5}+1}{2} = 1.618033988749895$$

```
>plot2d(sequence("x[n-1]+x[n-2]",[1,1],250)^(1/(1:250))):
```

Barisan yang sama juga dapat dihasilkan dengan menggunakan loop.

```
>x=ones(500); for k=3 to 500; x[k]=x[k-1]+x[k-2]; end;
```

Rekursi dapat dilakukan dengan menggunakan rumus yang tergantung pada semua elemen sebelumnya. Pada contoh berikut, elemen ke-n merupakan jumlah (n-1) elemen sebelumnya, dimulai dengan 1 (elemen ke-1). Jelas, nilai elemen ke-n adalah 2^(n-2), untuk n=2, 4, 5, ...

```
>sequence("sum(x)",1,10)
```

```
[1,  1,  2,  4,  8,  16,  32,  64,  128,  256]
```

Selain menggunakan ekspresi dalam x dan n, kita juga dapat menggunakan fungsi.

Pada contoh berikut, digunakan iterasi

$$x_n = A \cdot x_{n-1},$$

dengan A suatu matriks 2x2, dan setiap x[n] merupakan matriks/vektor 2x1.

```
>A=[1,1;1,2]; function suku(x,n) := A.x[,n-1]
>sequence("suku",[1;1],6)
```

```
Real 2 x 6 matrix

            1            2            5           13     ...
            1            3            8           21     ...
```

Hasil yang sama juga dapat diperoleh dengan menggunakan fungsi perpangkatan matriks "matrixpower()". Cara ini lebih cepat, karena hanya menggunakan perkalian matriks sebanyak log_2(n).

$$x_n = A.x_{n-1} = A^2.x_{n-2} = A^3.x_{n-3} = ... = A^{n-1}.x_1.$$

```
>sequence("matrixpower(A,n).[1;1]",1,6)
```

```
Real 2 x 6 matrix

            1             5            13            34     ...
            1             8            21            55     ...
```

Spiral Theodorus

image: Spiral_of_Theodorus.png
Spiral Theodorus (spiral segitiga siku-siku) dapat digambar secara rekursif. Rumus rekursifnya adalah:

$$x_n = \left(1 + \frac{i}{\sqrt{n-1}}\right) x_{n-1}, \quad x_1 = 1,$$

yang menghasilkan barisan bilangan kompleks.

```
>function g(n) := 1+I/sqrt(n)
```

Rekursinya dapat dijalankan sebanyak 17 untuk menghasilkan barisan 17 bilangan kompleks, kemudian digambar bilangan-bilangan kompleksnya.

```
>x=sequence("g(n-1)*x[n-1]",1,17); plot2d(x,r=3.5); textbox(latex("Spiral\ Theodorus"),0.4):
```

Selanjutnya dihubungan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
>for i=1:cols(x); plot2d([0,x[i]],>add); end:
```

Spiral Theodorus
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

```
>
```

Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memmerlukan indeks dan bilangan kompleks. Dalam hal ini diigunakan vektor kolom pada bidang.

```
>function gstep (v) ...
```

```
  w=[-v[2];v[1]];
  return v+w/norm(w);
  endfunction
```

Jika dilakukan iterasi 16 kali dimulai dari [1;0] akan didapatkan matriks yang memuat vektor-vektor dari setiap iterasi.

```
>x=iterate("gstep",[1;0],16); plot2d(x[1],x[2],r=3.5,>points):
```

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

Kekonvergenan

Terkadang kita ingin melakukan iterasi sampai konvergen. Apabila iterasinya tidak konvergen setelah ditunggu lama, Anda dapat menghentikannya dengan menekan tombol [ESC].

```
>iterate("cos(x)",1) // iterasi x(n+1)=cos(x(n)), dengan x(0)=1.
```

```
0.739085133216
```

Iterasi tersebut konvergen ke penyelesaian persamaan

$$x = \cos(x).$$

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.

```
>hasil := iterate("cos(x)",~1,2~) //iterasi x(n+1)=cos(x(n)), dengan interval awal (1, 2)
```

```
~0.739085133211,0.7390851332133~
```

Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan x=cos(x).

```
>h=expand(hasil,100), cos(h) << h
```

```
~0.73908513309,0.73908513333~
1
```

Iterasi juga dapat digunakan pada fungsi yang didefinisikan.

```
>function f(x) := (x+2/x)/2
```

Iterasi x(n+1)=f(x(n)) akan konvergen ke akar kuadrat 2.

```
>iterate("f",2), sqrt(2)
```

```
1.41421356237
1.41421356237
```

Jika pada perintah iterate diberikan tambahan parameter n, maka hasil iterasinya akan ditampilkan mulai dari iterasi pertama sampai ke-n.

```
>iterate("f",2,5)
```

```
[2,  1.5,  1.41667,  1.41422,  1.41421,  1.41421]
```

Untuk iterasi ini tidak dapat dilakukan terhadap interval.

```
>niterate("f",~1,2~,5)
```

```
[ ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~,  ~1,2~ ]
```

Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya adalah perhitungan dengan interval bersifat terlalu longgar. Untuk meingkatkan perhitungan pada ekspresi dapat digunakan pembagian intervalnya, menggunakan fungsi ieval().

```
>function s(x) := ieval("(x+2/x)/2",x,10)
```

Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal, dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang memuat akar persamaan:

$$x = \frac{1}{2}\left(x + \frac{2}{x}\right).$$

Satu-satunya solusi adalah

$$x = \sqrt{2}.$$

```
>iterate("s",~1,2~)
```

```
~1.41421356236,1.41421356239~
```

Fungsi "iterate()" juga dapat bekerja pada vektor. Berikut adalah contoh fungsi vektor, yang menghasilkan rata-rata aritmetika dan rata-rata geometri.

$$(a_{n+1}, b_{n+1}) = \left(\frac{a_n + b_n}{2}, \sqrt{a_n b_n}\right)$$

Iterasi ke-n disimpan pada vektor kolom x[n].

```
>function g(x) := [(x[1]+x[2])/2;sqrt(x[1]*x[2])]
```

Iterasi dengan menggunakan fungsi tersebut akan konvergen ke rata-rata aritmetika dan geometri dari nilai-nilai awal.

```
>iterate("g",[1;5])
```

```
        2.60401
        2.60401
```

Hasil tersebut konvergen agak cepat, seperti kita cek sebagai berikut.

```
>iterate("g",[1;5],4)
```

```
              1             3       2.61803       2.60403       2.60401
              5       2.23607       2.59002       2.60399       2.60401
```

Iterasi pada interval dapat dilakukan dan stabil, namun tidak menunjukkan bahwa limitnya pada batas-batas yang dihitung.

```
>iterate("g",[~1~;~5~],4)
```

```
  Interval 2 x 5 matrix

  ~0.999999999999999778,1.00000000000000022~     ...
  ~4.99999999999999911,5.00000000000000089~     ...
```

Iterasi berikut konvergen sangat lambat.

$$x_{n+1} = \sqrt{x_n}.$$

```
>iterate("sqrt(x)",2,10)
```

```
[2,  1.41421,  1.18921,  1.09051,  1.04427,  1.0219,  1.01089,
1.00543,  1.00271,  1.00135,  1.00068]
```

Kekonvergenan iterasi tersebut dapat dipercepatdengan percepatan Steffenson:

```
>steffenson("sqrt(x)",2,10)
```

```
[1.04888,  1.00028,  1,  1]
```

Iterasi menggunakan Loop yang ditulis Langsung

Berikut adalah beberapa contoh penggunaan loop untuk melakukan iterasi yang ditulis langsung pada baris perintah.

```
>x=2; repeat x=(x+2/x)/2; until x^2~=2; end; x,
```

```
1.41421356237
```

Penggabungan matriks menggunakan tanda "|" dapat digunakan untuk menyimpan semua hasil iterasi.

```
>v=[1]; for i=2 to 8; v=v|(v[i-1]*i); end; v,
```

```
[1,  2,  6,  24,  120,  720,  5040,  40320]
```

hasil iterasi juga dapat disimpan pada vektor yang sudah ada.

```
>v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]*i; end; ...
>plot2d(v,logplot=1); textbox(latex(&log(n)),x=0.5):
```

```
>A =[0.5,0.2;0.7,0.1]; b=[2;2]; ...
>x=[1;1]; repeat xnew=A.x-b; until all(xnew~=x); x=xnew; end; ...
>x,
```

```
-7.09677
-7.74194
```

Iterasi di dalam Fungsi

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.

Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

```
>function map hiter(f$,x0) ...
```

```
  x=x0;
  maxiter=0;
  repeat
    xnew=f$(x);
    maxiter=maxiter+1;
    until xnew~=x;
    x=xnew;
  end;
  return maxiter;
  endfunction
```

Misalnya, berikut adalah iterasi untuk mendapatkan hampiran akar kuadrat 2, cukup cepat, konvergen pada iterasi ke-5, jika dimulai dari hampiran awal 2.

```
>hiter("(x+2/x)/2",2)
```

```
5
```

Karena fungsinya didefinisikan menggunakan ”map”. maka nilai awalnya dapat berupa vektor.

```
>x=1.5:0.1:10; hasil=hiter("(x+2/x)/2",x); ...
>  plot2d(x,hasil):
```

Dari gambar di atas terlihat bahwa kekonvergenan iterasinya semakin lambat, untuk nilai awal semakin besar, namun penambahnnya tidak kontinu. Kita dapat menemukan kapan maksimum iterasinya bertambah.

```
>hasil[1:10]
```

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

```
>x[nonzeros(differences(hasil))]
```

```
[1.5,  2,  3.4,  6.6]
```

maksimum iterasi sampai konvergen meningkat pada saat nilai awalnya 1.5, 2, 3.4, dan 6.6.

Contoh berikutnya adalah metode Newton pada polinomial kompleks berderajat 3.

```
>p &= x^3-1; newton &= x-p/diff(p,x); $newton
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
p &= x^3-1; newton &= x-p/diff(p,x); $newton ...
                                    ^
```

Selanjutnya didefinisikan fungsi untuk melakukan iterasi (aslinya 10 kali).

```
>function iterasi(f$,x,n=10) ...
```

```
loop 1 to n; x=f$(x); end;
return x;
endfunction
```

Kita mulai dengan menentukan titik-titik grid pada bidang kompleksnya.

```
>r=1.5; x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z);
```

```
Function newton needs at least 3 arguments!
Use: newton (f$: call, df$: call, x: scalar complex {, y: number, eps: none})
Error in:
...  x=linspace(-r,r,501); Z=x+I*x'; W=iterasi(newton,Z); ...
                                                        ^
```

Berikut adalah akar-akar polinomial di atas.

```
>z=&solve(p)()
```

```
Maxima said:
solve: more equations than unknowns.
Unknowns given :
[r]
Equations given:
errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
z=&solve(p)() ...
           ^
```

Untuk menggambar hasil iterasinya, dihitung jarak dari hasil iterasi ke-10 ke masing-masing akar, kemudian digunakan untuk menghitung warna yang akan digambar, yang menunjukkan limit untuk masing-masing nilai awal.

Fungsi plotrgb() menggunakan jendela gambar terkini untuk menggambar warna RGB sebagai matriks.

```
>C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1)); ...
>  plot2d(none,-r,r,-r,r); plotrgb(C):
```

```
Variable W not found!
Error in:
C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1) ...
                     ^
```

Iterasi Simbolik

Seperti sudah dibahas sebelumnya, untuk menghasilkan barisan ekspresi simbolik dengan Maxima dapat digunakan fungsi makelist().

```
>&powerdisp:true // untuk menampilkan deret pangkat mulai dari suku berpangkat terkecil
```

```
true
```

```
>deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // barisan deret Taylor untuk e^x
```

```
Maxima said:
taylor: 0.1539740213994798*r cannot be a variable.
 -- an error. To debug this try: debugmode(true);

Error in:
deret &= makelist(taylor(exp(x),x,0,k),k,1,3); $deret // baris ...
                                              ^
```

Untuk mengubah barisan deret tersebut menjadi vektor string di EMT digunakan fungsi mxm2str(). Selanjutnya, vektor string/ekspresi hasilnya dapat digambar seperti menggambar vektor eskpresi pada EMT.

```
>plot2d("exp(x)",0,3); // plot fungsi aslinya, e^x
>plot2d(mxm2str("deret"),>add,color=4:6): // plot ketiga deret taylor hampiran fungsi tersebut
```

```
Maxima said:
length: argument cannot be a symbol; found deret
 -- an error. To debug this try: debugmode(true);

mxmeval:
    return evaluate(mxm(s));
Try "trace errors" to inspect local variables after errors.
mxm2str:
    n=mxmeval("length(VVV)");
```

Selain cara di atas dapat juga dengan cara menggunakan indeks pada vektor/list yang dihasilkan.

```
>$deret[3]
```

$$deret_3$$

```
>plot2d(["exp(x)",&deret[1],&deret[2],&deret[3]],0,3,color=1:4):
```

```
deret is not a variable!
Error in expression: deret[1]
 %ploteval:
    y0=f$(x[1],args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    u=u_(%ploteval(xx[#],t;args()));
```

```
>$sum(sin(k*x)/k,k,1,5)
```

$$\frac{\sin(0.725714333077512\,r)}{5} + \left[\sin(0.14956262333035794\,r) + \sin(0.299005239331655671488\,r) + \sin(0.0448507888524089975009072331801578\,r) + \sin(0.29801051803211941\,r) + \sin(0.66\,r)\right.$$

Berikut adalah cara menggambar kurva

$$y = \sin(x) + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \ldots.$$

```
>plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi):
```

```
Maxima output too long!
Error in:
plot2d(&sum(sin((2*k+1)*x)/(2*k+1),k,0,20),0,2pi): ...
                                                   ^
```

Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan kita akan menggambar kurva

$$y = \sum_{k=1}^{100} \frac{\sin(kx)}{k}, \quad 0 \le x \le 2\pi.$$

```
>x=linspace(0,2pi,1000); k=1:100; y=sum(sin(k*x')/k)'; plot2d(x,y):
```

1.5
1
0.5
0
-0.5
-1
-1.5
0
1
2
3
4
5
6

Tabel Fungsi

Terdapat cara menarik untuk menghasilkan barisan dengan ekspresi Maxima. Perintah mxmtable() berguna untuk menampilkan dan menggambar barisan dan menghasilkan barisan sebagai vektor kolom.

Sebagai contoh berikut adalah barisan turunan ke-n x^x di x=1.

```
>mxmtable("diffat(x^x,x=1,n)","n",1,8,frac=1);
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
#0: diffat(expr=[0,1.66665833335744e-7*r,1.33330666692022e-6*r,4.499797504338432e-6*r,1.0665813365
 -- an error. To debug this try: debugmode(true);

 %mxmevtable:
    return mxm("@expr,@var=@value")();
Try "trace errors" to inspect local variables after errors.
mxmtable:
    y[#,1]=%mxmevtable(expr,var,x[#]);
```

```
>$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n),simpsum=true)) // simpsum:menghitung deret secara sim
```

$$\sum_{k=1}^{n} k = \frac{n\ (1+n)}{2}$$

```
>$'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf),simpsum=true))
```

$$\sum_{k=0}^{\infty} \frac{1}{k+3^k} = \sum_{k=0}^{\infty} \frac{1}{k+3^k}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum(1/x^2, x, 1, inf)= ev(sum(1/x^2, x, 1, inf),simpsum=true) // ev: menghitung nilai ekspresi
```

$$\sum_{x=1}^{\infty} \frac{1}{x^2} = \frac{\pi^2}{6}$$

```
>$'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^{-1+k}}{k} = -\sum_{x=1}^{\infty} \frac{(-1)^x}{x}$$

Di sini masih gagal, hasilnya tidak dihitung.

```
>$'sum((-1)^k/(2*k-1), k, 1, inf) = factor(ev(sum((-1)^k/(2*k-1), k, 1, inf),simpsum=true))
```

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{-1+2\,k} = \sum_{k=1}^{\infty} \frac{(-1)^k}{-1+2\,k}$$

```
>$ev(sum(1/n!, n, 0, inf),simpsum=true)
```

$$\sum_{n=0}^{\infty} \frac{1}{n!}$$

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
>&assume(abs(x)<1); $'sum(a*x^k, k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forget(abs(x)<1
```

```
Answering "Is -94914474571+15819*r positive, negative or zero?" with "positive"
Maxima said:
sum: sum is divergent.
 -- an error. To debug this try: debugmode(true);

Error in:
... k, 0, inf)=ev(sum(a*x^k, k, 0, inf),simpsum=true), &forget(abs ...
                                                      ^
```

Deret geometri tak hingga, dengan asumsi rasional antara -1 dan 1.

```
>$'sum(x^k/k!,k,0,inf)=ev(sum(x^k/k!,k,0,inf),simpsum=true)
```

$$\left[0, \sum_{k=0}^{\infty} \frac{\left(1.66665833335744 \times 10^{-7}\right)^k r^k}{k!}, \sum_{k=0}^{\infty} \frac{\left(1.33330666692022 \times 10^{-6}\right)^k r^k}{k!}, \sum_{k=0}^{\infty} \frac{\left(4.499797504338432 \times 10^{-6}\right)^k r^k}{k!}, \sum_{k=0}^{\infty} \left(\right.\right.$$

```
>$limit(sum(x^k/k!,k,0,n),n,inf)
```

$$\left[0, \lim_{n\to\infty} \sum_{k=0}^{n} \frac{\left(1.66665833335744 \times 10^{-7}\right)^k r^k}{k!}, \lim_{n\to\infty} \sum_{k=0}^{n} \frac{\left(1.33330666692022 \times 10^{-6}\right)^k r^k}{k!}, \lim_{n\to\infty} \sum_{k=0}^{n} \frac{\left(4.499797504338432 \times 1\right.}{k!}\right.$$

```
>function d(n) &= sum(1/(k^2-k),k,2,n); $'d(n)=d(n)
```

$$d(n) = \sum_{k=2}^{n} \frac{1}{-k + k^2}$$

```
>$d(10)=ev(d(10),simpsum=true)
```

$$\sum_{k=2}^{10} \frac{1}{-k + k^2} = \frac{9}{10}$$

```
>$d(100)=ev(d(100),simpsum=true)
```

$$\sum_{k=2}^{100} \frac{1}{-k+k^2} = \frac{99}{100}$$

Deret Taylor

Deret Taylor suatu fungsi f yang diferensiabel sampai tak hingga di sekitar x=a adalah:

$$f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^k f^{(k)}(a)}{k!}.$$

```
>$'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x=0, sampai suku ke-11
```

```
Maxima said:
taylor: 0.1539740213994798*r cannot be a variable.
 -- an error. To debug this try: debugmode(true);

Error in:
 $'e^x =taylor(exp(x),x,0,10) // deret Taylor e^x di sekitar x= ...
                              ^
```

```
>$'log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1
```

```
Maxima said:
log: encountered log(0).
 -- an error. To debug this try: debugmode(true);

Error in:
 $'log(x)=taylor(log(x),x,1,10)// deret log(x) di sekitar x=1 ...
                               ^
```

Kalkulus dengan EMT

Materi Kalkulus mencakup di antaranya:

- Fungsi (fungsi aljabar, trigonometri, eksponensial, logaritma, komposisi fungsi)
- Limit Fungsi,
- Turunan Fungsi,
- Integral Tak Tentu,
- Integral Tentu dan Aplikasinya,
- Barisan dan Deret (kekonvergenan barisan dan deret).

EMT (bersama Maxima) dapat digunakan untuk melakukan semua perhitungan di dalam kalkulus, baik secara numerik maupun analitik (eksak).

Mendefinisikan Fungsi

Terdapat beberapa cara mendefinisikan fungsi pada EMT, yakni:

- Menggunakan format nama_fungsi := rumus fungsi (untuk fungsi numerik),
- Menggunakan format nama_fungsi &= rumus fungsi (untuk fungsi simbolik, namun dapat dihitung secara numerik),
- Menggunakan format nama_fungsi &&= rumus fungsi (untuk fungsi simbolik murni, tidak dapat dihitung langsung),
- Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah function (bukan sebagai ekspresi).

Berikut adalah adalah beberapa contoh cara mendefinisikan fungsi:

$$f(x) = 2x^2 + e^{\sin(x)}.$$

```
>function f(x) := 2*x^2+exp(sin(x)) // fungsi numerik
>f(0), f(1), f(pi)
```

```
1
4.31977682472
20.7392088022
```

```
>f(a) // tidak dapat dihitung nilainya
```

```
1.12498683033
```

Silakan Anda plot kurva fungsi di atas!

Berikutnya kita definisikan fungsi:

$$g(x) = \frac{\sqrt{x^2 - 3x}}{x + 1}.$$

```
>function g(x) := sqrt(x^2-3*x)/(x+1)
>g(3)
```

```
0
```

```
>g(0)
```

```
0
```

```
>g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik
```

```
Floating point error!
Error in sqrt
Try "trace errors" to inspect local variables after errors.
g:
    useglobal; return sqrt(x^2-3*x)/(x+1)
Error in:
g(1) // kompleks, tidak dapat dihitung oleh fungsi numerik ...
    ^
```

Silakan Anda plot kurva fungsi di atas!

```
>f(g(5)) // komposisi fungsi
```

```
2.20920171961
```

```
>g(f(5))
```

```
0.950898070639
```

```
>function h(x) := f(g(x)) // definisi komposisi fungsi
>h(5) // sama dengan f(g(5))
```

```
2.20920171961
```

Silakan Anda plot kurva fungsi komposisi fungsi f dan g:

$$h(x) = f(g(x))$$

dan

$$u(x) = g(f(x))$$

bersama-sama kurva fungsi f dan g dalam satu bidang koordinat.

```
>f(0:10) // nilai-nilai f(0), f(1), f(2), ..., f(10)
```

```
[1,  4.31978,  10.4826,  19.1516,  32.4692,  50.3833,  72.7562,
99.929,  130.69,  163.51,  200.58]
```

```
>fmap(0:10) // sama dengan f(0:10), berlaku untuk semua fungsi
```

```
[1,  4.31978,  10.4826,  19.1516,  32.4692,  50.3833,  72.7562,
99.929,  130.69,  163.51,  200.58]
```

```
>gmap(200:210)
```

```
[0.987534,  0.987596,  0.987657,  0.987718,  0.987778,  0.987837,
0.987896,  0.987954,  0.988012,  0.988069,  0.988126]
```

Misalkan kita akan mendefinisikan fungsi

$$f(x) = \begin{cases} x^3 & x > 0 \\ x^2 & x \leq 0. \end{cases}$$

Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format :=, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.

```
>function map f(x) ...
```

```
    if x>0 then return x^3
    else return x^2
    endif;
  endfunction
```

```
>f(1)
```

```
1
```

```
>f(-2)
```

```
4
```

```
>f(-5:5)
```

```
[25,  16,  9,  4,  1,  0,  1,  8,  27,  64,  125]
```

```
>aspect(1.5); plot2d("f(x)",-5,5):
```

```
>function f(x) &= 2*E^x // fungsi simbolik
```

```
                1.66665833335744e-7 r     1.33330666692022e-6 r
        [2, 2 E                      , 2 E                      ,
   4.499797504338432e-6 r     1.066581336583994e-5 r
2 E                       , 2 E                       ,
   2.083072932167196e-5 r     3.599352055540239e-5 r
2 E                       , 2 E                       ,
   5.71526624672386e-5 r     8.530603082730626e-5 r
```

```
2 E                     , 2 E                     ,
   1.214508019889565e-4 r     1.665833531718508e-4 r
2 E                      , 2 E                      ,
   2.216991628251896e-4 r     2.877927110806339e-4 r
2 E                      , 2 E                       ,
   3.658573803051457e-4 r     4.568853557635201e-4 r
2 E                      , 2 E                      ,
   5.618675264007778e-4 r     6.817933857540259e-4 r
2 E                      , 2 E                       ,
   8.176509330039827e-4 r     9.704265741758145e-4 r
2 E                      , 2 E                       ,
   0.001141105023499428 r     0.001330669204938795 r
2 E                      , 2 E                       ,
   0.001540100153900437 r     0.001770376919130678 r
2 E                      , 2 E                       ,
   0.002022476464811601 r     0.002297373572865413 r
2 E                      , 2 E                       ,
   0.002596040745477063 r     0.002919448107844891 r
2 E                      , 2 E                       ,
   0.003268563311168871 r     0.003644351435886262 r
2 E                      , 2 E                       ,
   0.004047774895164447 r     0.004479793338660443 r
2 E                      , 2 E                       ,
   0.0049413635565565 r     0.005433439383882244 r
2 E                    , 2 E                      ,
   0.005956971605131645 r     0.006512907859185624 r
2 E                      , 2 E                       ,
   0.007102192544548636 r     0.007725766724910044 r
2 E                      , 2 E                       ,
   0.00838456803503801 r     0.009079530587017326 r
2 E                     , 2 E                      ,
   0.009811584876838586 r     0.0105816576913495 r
2 E                      , 2 E                     ,
   0.01139067201557714 r     0.01223954694042984 r
2 E                     , 2 E                     ,
```

0.01312919757078923 r 0.01406053493400045 r
2 E , 2 E ,
0.01503446588876983 r 0.01605189303448024 r
2 E , 2 E ,
0.01711371462093175 r 0.01822082445851714 r
2 E , 2 E ,
0.01937411182884202 r 0.02057446139579705 r
2 E , 2 E ,
0.02182275311709253 r 0.02311986215626333 r
2 E , 2 E ,
0.02446665879515308 r 0.02586400834688696 r
2 E , 2 E ,
0.02731277106934082 r 0.02881380207911666 r
2 E , 2 E ,
0.03036795126603076 r 0.03197606320812652 r
2 E , 2 E ,
0.0336389770872163 r 0.03535752660496472 r
2 E , 2 E ,
0.03713253989951881 r 0.03896483946269502 r
2 E , 2 E ,
0.0408552420577305 r 0.04280455863760801 r
2 E , 2 E ,
0.04481359426396048 r 0.04688314802656623 r
2 E , 2 E ,
0.04901401296344043 r 0.05120697598153157 r
2 E , 2 E ,
0.05346281777803219 r 0.05578231276230905 r
2 E , 2 E ,
0.05816622897846346 r 0.06061532802852698 r
2 E , 2 E ,
0.0631303649963022 r 0.06571208837185505 r
2 E , 2 E ,
0.06836123997666599 r 0.07107855488944881 r
2 E , 2 E ,
0.07386476137264342 r 0.07672058079958999 r

```
2 E                    , 2 E                    ,
   0.07964672758239233 r     0.08264390910047736 r
2 E                    , 2 E                    ,
   0.0857128256298576 r     0.08885417027310427 r
2 E                    , 2 E                    ,
   0.09206862889003742 r     0.09535688002914089 r
2 E                    , 2 E                    ,
   0.0987195948597075 r     0.1021574371047232 r
2 E                    , 2 E                    ,
   0.1056710629744951 r     0.1092611211010309 r
2 E                    , 2 E                    ,
   0.1129282524731764 r     0.1166730903725168 r
2 E                    , 2 E                    ,
   0.1204962603100498 r     0.1243983799636342 r
2 E                    , 2 E                    ,
   0.1283800591162231 r     0.1324418995948859 r
2 E                    , 2 E                    ,
   0.1365844952106265 r     0.140808431699002 r
2 E                    , 2 E                    ,
   0.1451142866615502 r     0.1495026295080298 r
2 E                    , 2 E                    ,
   0.1539740213994798 r
2 E                    ]
```

```
>$f(a) // nilai fungsi secara simbolik
```

$$\left[2, 2\,e^{1.66665833335744\times10^{-7}\,r}, 2\,e^{1.33330666692022\times10^{-6}\,r}, 2\,e^{4.499797504338432\times10^{-6}\,r}, 2\,e^{1.066581336583994\times10^{-5}\,r}, 2\,e^{2.08307293216719}\right.$$

```
>f(E) // nilai fungsi berupa bilangan desimal
```

```
[2,  2,  2,  2.00001,  2.00003,  2.00006,  2.00011,  2.00017,  2.00026,
2.00036,  2.0005,  2.00067,  2.00086,  2.0011,  2.00137,  2.00169,
2.00205,  2.00245,  2.00291,  2.00343,  2.004,  2.00463,  2.00532,
2.00608,  2.0069,  2.0078,  2.00878,  2.00983,  2.01096,  2.01218,
2.01348,  2.01488,  2.01637,  2.01795,  2.01963,  2.02142,  2.02331,
2.02531,  2.02742,  2.02965,  2.032,  2.03447,  2.03706,  2.03978,
2.04263,  2.04562,  2.04874,  2.05201,  2.05542,  2.05898,  2.06269,
2.06655,  2.07058,  2.07476,  2.07912,  2.08364,  2.08834,  2.09321,
2.09827,  2.10351,  2.10894,  2.11456,  2.12038,  2.1264,  2.13263,
2.13906,  2.14571,  2.15258,  2.15967,  2.16699,  2.17455,  2.18234,
2.19037,  2.19865,  2.20718,  2.21597,  2.22502,  2.23434,  2.24393,
2.2538,  2.26395,  2.2744,  2.28514,  2.29619,  2.30754,  2.31921,
2.3312,  2.34352,  2.35617,  2.36917,  2.38252,  2.39622,  2.41028,
2.42472,  2.43954,  2.45475,  2.47035,  2.48636,  2.50278,  2.51962]
```

```
>$f(E), $float(%)
```

$$\left[2.0, 2.0\,2.718281828459045^{1.66665833335744\times10^{-7}\,r}, 2.0\,2.718281828459045^{1.33330666692022\times10^{-6}\,r}, 2.0\,2.718281828459045^{4.499}\right.$$

```
>function g(x) &= 3*x+1
```

```
        [1, 1 + 4.999975000072321e-7 r, 1 + 3.999920000760659e-6 r,
1 + 1.34993925130153e-5 r, 1 + 3.199744009751981e-5 r,
1 + 6.249218796501588e-5 r, 1 + 1.079805616662072e-4 r,
1 + 1.714579874017158e-4 r, 1 + 2.559180924819188e-4 r,
1 + 3.643524059668696e-4 r, 1 + 4.997500595155524e-4 r,
1 + 6.650974884755689e-4 r, 1 + 8.633781332419016e-4 r,
1 + 0.001097572140915437 r, 1 + 0.00137065606729056 r,
1 + 0.001685602579202333 r, 1 + 0.002045380157262078 r,
1 + 0.002452952799011948 r, 1 + 0.002911279722527443 r,
1 + 0.003423315070498284 r, 1 + 0.003992007614816384 r,
1 + 0.00462030046170131 r, 1 + 0.005311130757392035 r,
1 + 0.006067429394434803 r, 1 + 0.00689212071859624 r,
1 + 0.007788122236431189 r, 1 + 0.008758344323534673 r,
1 + 0.009805689933506612 r, 1 + 0.01093305430765878 r,
1 + 0.01214332468549334 r, 1 + 0.01343938001598133 r,
1 + 0.0148240906696695 r, 1 + 0.01630031815164673 r,
1 + 0.01787091481539493 r, 1 + 0.01953872357755687 r,
1 + 0.02130657763364591 r, 1 + 0.02317730017473013 r,
1 + 0.02515370410511403 r, 1 + 0.02723859176105198 r,
1 + 0.02943475463051576 r, 1 + 0.0317449730740485 r,
1 + 0.03417201604673142 r, 1 + 0.03671864082128951 r,
1 + 0.03938759271236769 r, 1 + 0.04218160480200134 r,
1 + 0.0451033976663095 r, 1 + 0.04815567910344071 r,
1 + 0.05134114386279526 r, 1 + 0.05466247337555141 r,
1 + 0.05812233548652607 r, 1 + 0.06172338418739115 r,
1 + 0.06546825935127759 r, 1 + 0.06935958646878998 r,
1 + 0.07339997638545925 r, 1 + 0.07759202504066087 r,
1 + 0.08193831320802247 r, 1 + 0.08644140623734997 r,
1 + 0.09110385379809227 r, 1 + 0.09592818962437955 r,
1 + 0.1009169312616489 r, 1 + 0.1060725798148942 r,
1 + 0.1113976196985564 r, 1 + 0.1168945183880851 r,
1 + 0.1225657261731915 r, 1 + 0.128413675912824 r,
1 + 0.1344407827918814 r, 1 + 0.1406494440796987 r,
1 + 0.1470420388903213 r, 1 + 0.1536209279445947 r,
```

```
1 + 0.1603884533340966 r, 1 + 0.1673469382869271 r,
1 + 0.1744986869353904 r, 1 + 0.1818459840855809 r,
1 + 0.1893910949889066 r, 1 + 0.1971362651155651 r,
1 + 0.205083719929998 r, 1 + 0.2132356646683464 r,
1 + 0.2215942841179303 r, 1 + 0.23016174239877 r,
1 + 0.238940182747177 r, 1 + 0.2479317273014321 r,
1 + 0.2571384768895728 r, 1 + 0.2665625108193128 r,
1 + 0.2762058866701123 r, 1 + 0.2860706400874227 r,
1 + 0.2961587845791225 r, 1 + 0.3064723113141697 r,
1 + 0.3170131889234854 r, 1 + 0.3277833633030928 r,
1 + 0.3387847574195292 r, 1 + 0.3500192711175505 r,
1 + 0.3614887809301494 r, 1 + 0.3731951398909027 r,
1 + 0.3851401773486692 r, 1 + 0.3973256987846578 r,
1 + 0.4097534856318796 r, 1 + 0.4224252950970061 r,
1 + 0.4353428599846507 r, 1 + 0.4485078885240895 r,
1 + 0.4619220641984394 r]
```

```
>function h(x) &= f(g(x)) // komposisi fungsi
```

```
                1.66665833335744e-7 r     1.33330666692022e-6 r
        [2, 2 E                     , 2 E                    ,
   4.499797504338432e-6 r     1.066581336583994e-5 r
2 E                      , 2 E                      ,
   2.083072932167196e-5 r     3.599352055540239e-5 r
2 E                      , 2 E                      ,
   5.71526624672386e-5 r     8.530603082730626e-5 r
2 E                     , 2 E                      ,
   1.214508019889565e-4 r     1.665833531718508e-4 r
2 E                      , 2 E                      ,
   2.216991628251896e-4 r     2.877927110806339e-4 r
```

```
2 E                      , 2 E                      ,
   3.658573803051457e-4 r     4.568853557635201e-4 r
2 E                      , 2 E                      ,
   5.618675264007778e-4 r     6.817933857540259e-4 r
2 E                      , 2 E                      ,
   8.176509330039827e-4 r     9.704265741758145e-4 r
2 E                      , 2 E                      ,
   0.001141105023499428 r     0.001330669204938795 r
2 E                      , 2 E                      ,
   0.001540100153900437 r     0.001770376919130678 r
2 E                      , 2 E                      ,
   0.002022476464811601 r     0.002297373572865413 r
2 E                      , 2 E                      ,
   0.002596040745477063 r     0.002919448107844891 r
2 E                      , 2 E                      ,
   0.003268563311168871 r     0.003644351435886262 r
2 E                      , 2 E                      ,
   0.004047774895164447 r     0.004479793338660443 r
2 E                      , 2 E                      ,
   0.0049413635565565 r     0.005433439383882244 r
2 E                    , 2 E                      ,
   0.005956971605131645 r     0.006512907859185624 r
2 E                      , 2 E                      ,
   0.007102192544548636 r     0.007725766724910044 r
2 E                      , 2 E                      ,
   0.00838456803503801 r     0.009079530587017326 r
2 E                     , 2 E                      ,
   0.009811584876838586 r     0.0105816576913495 r
2 E                      , 2 E                     ,
   0.01139067201557714 r     0.01223954694042984 r
2 E                     , 2 E                      ,
   0.01312919757078923 r     0.01406053493400045 r
2 E                     , 2 E                      ,
   0.01503446588876983 r     0.01605189303448024 r
2 E                     , 2 E                      ,
```

0.01711371462093175 r 0.01822082445851714 r
2 E , 2 E ,
0.01937411182884202 r 0.02057446139579705 r
2 E , 2 E ,
0.02182275311709253 r 0.02311986215626333 r
2 E , 2 E ,
0.02446665879515308 r 0.02586400834688696 r
2 E , 2 E ,
0.02731277106934082 r 0.02881380207911666 r
2 E , 2 E ,
0.03036795126603076 r 0.03197606320812652 r
2 E , 2 E ,
0.0336389770872163 r 0.03535752660496472 r
2 E , 2 E ,
0.03713253989951881 r 0.03896483946269502 r
2 E , 2 E ,
0.0408552420577305 r 0.04280455863760801 r
2 E , 2 E ,
0.04481359426396048 r 0.04688314802656623 r
2 E , 2 E ,
0.04901401296344043 r 0.05120697598153157 r
2 E , 2 E ,
0.05346281777803219 r 0.05578231276230905 r
2 E , 2 E ,
0.05816622897846346 r 0.06061532802852698 r
2 E , 2 E ,
0.0631303649963022 r 0.06571208837185505 r
2 E , 2 E ,
0.06836123997666599 r 0.07107855488944881 r
2 E , 2 E ,
0.07386476137264342 r 0.07672058079958999 r
2 E , 2 E ,
0.07964672758239233 r 0.08264390910047736 r
2 E , 2 E ,
0.0857128256298576 r 0.08885417027310427 r

```
2 E                    , 2 E                    ,
   0.09206862889003742 r     0.09535688002914089 r
2 E                     , 2 E                     ,
   0.0987195948597075 r     0.1021574371047232 r
2 E                    , 2 E                    ,
   0.1056710629744951 r     0.1092611211010309 r
2 E                    , 2 E                    ,
   0.1129282524731764 r     0.1166730903725168 r
2 E                    , 2 E                    ,
   0.1204962603100498 r     0.1243983799636342 r
2 E                    , 2 E                    ,
   0.1283800591162231 r     0.1324418995948859 r
2 E                    , 2 E                    ,
   0.1365844952106265 r     0.140808431699002 r
2 E                    , 2 E                   ,
   0.1451142866615502 r     0.1495026295080298 r
2 E                    , 2 E                    ,
   0.1539740213994798 r
2 E                    ]
```

```
>plot2d("h(x)",-1,1):
```

```
Error : h(x) does not produce a real or column vector

Error generated by error() command

 %ploteval:
    error(f$|" does not produce a real or column vector");
adaptiveevalone:
    s=%ploteval(g$,t;args());
Try "trace errors" to inspect local variables after errors.
```

```
plot2d:
    dw/n,dw/n^2,dw/n,auto;args());
```

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan fungsi-fungsi tersebut dan komposisinya di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik fungsi-fungsi tersebut dan komposisi-komposisi 2 fungsi.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

Menghitung Limit

Perhitungan limit pada EMT dapat dilakukan dengan menggunakan fungsi Maxima, yakni "limit". Fungsi "limit" dapat digunakan untuk menghitung limit fungsi dalam bentuk ekspresi maupun fungsi yang sudah didefinisikan sebelumnya. Nilai limit dapat dihitung pada sebarang nilai atau pada tak hingga (-inf, minf, dan inf). Limit kiri dan limit kanan juga dapat dihitung, dengan cara memberi opsi "plus" atau "minus". Hasil limit dapat berupa nilai, "und" (tak definisi), "ind" (tak tentu namun terbatas), "infinity" (kompleks tak hingga).

Perhatikan beberapa contoh berikut. Perhatikan cara menampilkan perhitungan secara lengkap, tidak hanya menampilkan hasilnya saja.

```
>$showev('limit(sqrt(x^2-3*x)/(x+1),x,inf))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([0,sqrt(-4.999975000072321e-7*r+2.7777500001498e-14*r^2)/(1+1.66665833335744e-
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(sqrt(x^2-3*x)/(x+1),x,inf)) ...
                                            ^
```

```
>$limit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3)
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
... imit((x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18),x,3) ...
                                                     ^
```

$$\lim_{x\to 3} \frac{x^3 - 13\,x^2 + 51\,x - 63}{x^3 - 4\,x^2 - 3\,x + 18} = -\frac{4}{5}$$

Fungsi tersebut diskontinu di titik x=3. Berikut adalah grafik fungsinya.

```
>aspect(1.5); plot2d("(x^3-13*x^2+51*x-63)/(x^3-4*x^2-3*x+18)",0,4); plot2d(3,-4/5,>points,style="ow
```

```
>$limit(2*x*sin(x)/(1-cos(x)),x,0)
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $limit(2*x*sin(x)/(1-cos(x)),x,0) ...
                                  ^
```

$$2\left(\lim_{x\to 0}\frac{x\,\sin x}{1-\cos x}\right)=4$$

Fungsi tersebut diskontinu di titik x=0. Berikut adalah grafik fungsinya.

```
>plot2d("2*x*sin(x)/(1-cos(x))",-pi,pi); plot2d(0,4,>points,style="ow",>add):
```

```
>$limit(cot(7*h)/cot(5*h),h,0)
```

$$\frac{5}{7}$$

$$\lim_{h\to 0}\frac{\cot(7\,h)}{\cot(5\,h)}=\frac{5}{7}$$

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di x=0. Berikut adalah grafiknya.

```
>plot2d("cot(7*x)/cot(5*x)",-0.001,0.001); plot2d(0,5/7,>points,style="ow",>add):
```

```
>$showev('limit(((x/8)^(1/3)-1)/(x-8),x,8))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([1/8,(-1+0.002751601454741079*r^(1/3))/(-8+1.66665833335744e-7*r),(-1+0.005503
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(((x/8)^(1/3)-1)/(x-8),x,8)) ...
                                            ^
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x-8)^(1/3)-1)/(x-8)",7.99,8.01); plot2d(8,1/24,>points,style="ow",>add):
```

```
>$showev('limit(1/(2*x-1),x,0))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([-1,1/(-1+3.333316666714881e-7*r),1/(-1+2.66661333384044e-6*r),1/(-1+8.9995950
 -- an error. To debug this try: debugmode(true);
```

```
Error in:
 $showev('limit(1/(2*x-1),x,0)) ...
                               ^
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1/(2*x-1))",-0.1,0.1); plot2d(0,-1,>points,style="ow",>add):
```

```
>$showev('limit((x^2-3*x-10)/(x-5),x,5))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([2,(-10-4.999975000072321e-7*r+2.7777500001498e-14*r^2)/(-5+1.66665833335744e-
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((x^2-3*x-10)/(x-5),x,5)) ...
                                        ^
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x^2-3x-10)/(x-5))",4.9,5.1); plot2d(5,7,>points,style="ow",>add):
```

```
>$showev('limit(sqrt(x^2+x)-x,x,inf))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([0,-1.66665833335744e-7*r+sqrt(1.66665833335744e-7*r+2.7777500001498e-14*r^2),
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(sqrt(x^2+x)-x,x,inf)) ...
                                      ^
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("((x^2+x)^0.5-x)",0,6); plot2d(5,7,>points,style="ow",>add):
```

```
>$showev('limit(abs(x-1)/(x-1),x,1,minus))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([-1,abs(-1+1.66665833335744e-7*r)/(-1+1.66665833335744e-7*r),abs(-1+1.33330666
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(abs(x-1)/(x-1),x,1,minus)) ...
                                           ^
```

Hitung limit di atas untuk x menuju 1 dari kanan.
Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>$limit((abs(x-1)/(x-1),x,plus,1))
```

$$1$$

```
>plot2d("((abs(x-1))/(x-1))",0.999,1.2):
```

```
>$showev('limit(sin(x)/x,x,0))
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(sin(x)/x,x,0)) ...
                               ^
```

```
>plot2d("sin(x)/x",-pi,pi); plot2d(0,1,>points,style="ow",>add):
```

```
>$showev('limit(sin(x^3)/x,x,0))
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(sin(x^3)/x,x,0)) ...
                                ^
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("sin(x^3)/x",-pi,pi); plot2d(0,0,>points,style="ow",>add):
```

```
>$showev('limit(log(x), x, minf))
```

```
Maxima said:
log: encountered log(0).
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(log(x), x, minf)) ...
                                 ^
```

```
>$showev('limit((-2)^x,x, inf))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([1,(-2)^(1.66665833335744e-7*r),(-2)^(1.33330666692022e-6*r),(-2)^(4.499797504
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((-2)^x,x, inf)) ...
                               ^
```

```
>$showev('limit(t-sqrt(2-t),t,2,minus))
```

```
Maxima said:
limit: second argument must be a variable, not a constant; found:
```

```
 errexp1
#0: showev(f='limit([-sqrt(2),-1.400673597966589,-1.387124727947029,-1.37356688476182,-1.36,-1.346
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(t-sqrt(2-t),t,2,minus)) ...
                                        ^
```

```
>$showev('limit(t-sqrt(2-t),t,2,plus))
```

```
Maxima said:
limit: second argument must be a variable, not a constant; found:
 errexp1
#0: showev(f='limit([-sqrt(2),-1.400673597966589,-1.387124727947029,-1.37356688476182,-1.36,-1.346
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(t-sqrt(2-t),t,2,plus)) ...
                                        ^
```

```
>$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya
```

```
Maxima said:
limit: second argument must be a variable, not a constant; found:
 errexp1
#0: showev(f='limit([-sqrt(2),-1.400673597966589,-1.387124727947029,-1.37356688476182,-1.36,-1.346
 -- an error. To debug this try: debugmode(true);

Error in:
```

```
$showev('limit(t-sqrt(2-t),t,5,plus)) // Perhatikan hasilnya ...
                                      ^
```

```
>plot2d("x-sqrt(2-x)",0,2):
```

```
>$showev('limit((x^2-9)/(2*x^2-5*x-3),x,3))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([3,(-9+2.7777500001498e-14*r^2)/(-3-8.333291666787201e-7*r+5.5555000002996e-14
```

```
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((x^2-9)/(2*x^2-5*x-3),x,3)) ...
                                            ^
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(x^2-9)/(2*x^2-5*x-3)",2,4); plot2d(3,6/7,>points,style="ow",>add):
```

```
>$showev('limit((1-cos(x))/x,x,0))
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((1-cos(x))/x,x,0)) ...
                                   ^
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(1-cos(x))/x",-pi,pi); plot2d(0,0,>points,style="ow",>add):
```

```
>$showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0))
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((x^2+abs(x))/(x^2-abs(x)),x,0)) ...
                                                ^
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(abs(x)+x^2)/(x^2-abs(x))",-1,1); plot2d(0,-1,>points,style="ow",>add):
```

```
>$showev('limit((1+1/x)^x,x,inf))
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((1+1/x)^x,x,inf)) ...
                                  ^
```

```
>plot2d("(1+1/x)^x",0,1000):
```

```
>$showev('limit((1+k/x)^x,x,inf))
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((1+k/x)^x,x,inf)) ...
                                 ^
```

```
>$showev('limit((1+x)^(1/x),x,0))
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((1+x)^(1/x),x,0)) ...
                                 ^
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(x+1)^(1/x)",-0.5,2); plot2d(0,E,>points,style="ow",>add):
```

```
>$showev('limit((x/(x+k))^x,x,inf))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([0,1.66665833335744e-7*r/(1/r+1.66665833335744e-7*r),1.33330666692022e-6*r/(1/
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((x/(x+k))^x,x,inf)) ...
                                    ^
```

```
>$showev('limit((E^x-E^2)/(x-2),x,2))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([-(1-E^2)/2,(-E^2+E^(1.66665833335744e-7*r))/(-2+1.66665833335744e-7*r),(-E^2+
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((E^x-E^2)/(x-2),x,2)) ...
                                      ^
```

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

```
>plot2d("(E^x-E^2)/(x-2)",1,3); plot2d(2,E^2,>points,style="ow",>add):
```

```
>$showev('limit(sin(1/x),x,0))
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(sin(1/x),x,0)) ...
                               ^
```

```
>$showev('limit(sin(1/x),x,inf))
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(sin(1/x),x,inf)) ...
                                 ^
```

```
>plot2d("sin(1/x)",-0.001,0.001):
```

1
0.5
0
-0.5
-1
-0.001
-0.0005
0
0.0005
0.001

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

Turunan Fungsi

Definisi turunan:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

```
>$showev('limit(((x+h)^2-x^2)/h,h,0)) // turunan x^2
```

```
>p &= expand((x+h)^2-x^2)|simplify; $p //pembilang dijabarkan dan disederhanakan
```

$$\left[h^2, h^2 + 3.333316666714881 \times 10^{-7}\, h\, r, h^2 + 2.66661333384044 \times 10^{-6}\, h\, r, h^2 + 8.999595008676864 \times 10^{-6}\, h\, r, h^2 + 2.133$$

```
>q &=ratsimp(p/h); $q // ekspresi yang akan dihitung limitnya disederhanakan
```

$$\left[h, \frac{94914474571\,h+31638\,r}{94914474571}, \frac{11914363285\,h+31771\,r}{11914363285}, \frac{5341573699\,h+48072\,r}{5341573699}, \frac{2759330113\,h+58861\,r}{2759330113}, \frac{855346912\,h+3}{855346912}\right.$$

```
>$limit(q,h,0) // nilai limit sebagai turunan
```

$$\left[0, \frac{31638\,r}{94914474571}, \frac{31771\,r}{11914363285}, \frac{48072\,r}{5341573699}, \frac{58861\,r}{2759330113}, \frac{35635\,r}{855346912}, \frac{98277\,r}{1365204049}, \frac{102308\,r}{895041417}, \frac{121691\,r}{713261412}, \frac{158455\,r}{652342337}, \frac{2}{85}\right.$$

```
>$showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n
```

```
Answering "Is -1+n positive, negative or zero?" with "positive"
Answering "Is n an integer?" with "integer"
Answering "Is n an integer?" with "integer"
Answering "Is n an integer?" with "integer"
Answering "Is n an integer?" with "integer"
Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is n an integer?

Use assume!
Error in:
 $showev('limit(((x+h)^n-x^n)/h,h,0)) // turunan x^n ...
                                    ^
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan (x+h)ˆn dengan menggunakan teorema binomial.

```
>$showev('limit((sin(x+h)-sin(x))/h,h,0)) // turunan sin(x)
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, ekspansikan sin(x+h) dengan menggunakan rumus jumlah dua sudut.

```
>$showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x)
```

```
Maxima said:
log: encountered log(0).
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((log(x+h)-log(x))/h,h,0)) // turunan log(x) ...
                                          ^
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

Sebagai petunjuk, gunakan sifat-sifat logaritma dan hasil limit pada bagian sebelumnya di atas.

```
>$showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit((1/(x+h)-1/x)/h,h,0)) // turunan 1/x ...
                                      ^
```

```
>$showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x
```

```
Answering "Is 15819*r/94914474571 an integer?" with "integer"
Answering "Is 15819*r/94914474571 an integer?" with "integer"
Answering "Is 15819*r/94914474571 an integer?" with "integer"
Answering "Is 15819*r/94914474571 an integer?" with "integer"
Answering "Is 15819*r/94914474571 an integer?" with "integer"
Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
Is 15819*r/94914474571 an integer?

Use assume!
Error in:
 $showev('limit((E^(x+h)-E^x)/h,h,0)) // turunan f(x)=e^x ...
                                      ^
```

Maxima bermasalah dengan limit:

$$\lim_{h\to 0} \frac{e^{x+h}-e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

```
>$showev('limit((E^h-1)/h,h,0))
```

$$\lim_{h\to 0} \frac{-1+e^h}{h} = 1$$

```
>$showev('factor(E^(x+h)-E^x))
```

$$factor\left(\left[-1+e^h, e^{h+1.66665833335744\times 10^{-7}\,r} - e^{1.66665833335744\times 10^{-7}\,r}, e^{h+1.33330666692022\times 10^{-6}\,r} - e^{1.33330666692022\times 10^{-6}\,r}, e^{h}\right.\right.$$

```
>$showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e^x
```

```
Answering "Is 15819*r/94914474571 an integer?" with "integer"
Answering "Is 15819*r/94914474571 an integer?" with "integer"
Answering "Is 15819*r/94914474571 an integer?" with "integer"
Answering "Is 15819*r/94914474571 an integer?" with "integer"
Answering "Is 15819*r/94914474571 an integer?" with "integer"
Maxima is asking
Acceptable answers are: yes, y, Y, no, n, N, unknown, uk
```

```
Is 15819*r/94914474571 an integer?

Use assume!
Error in:
 $showev('limit(factor((E^(x+h)-E^x)/h),h,0)) // turunan f(x)=e ...
                                                ^
```

```
>function f(x) &= x^x
```

```
        expt([0, 1.66665833335744e-7 r, 1.33330666692022e-6 r,
4.499797504338432e-6 r, 1.066581336583994e-5 r,
2.083072932167196e-5 r, 3.599352055540239e-5 r,
5.71526624672386e-5 r, 8.530603082730626e-5 r,
1.214508019889565e-4 r, 1.665833531718508e-4 r,
2.216991628251896e-4 r, 2.877927110806339e-4 r,
3.658573803051457e-4 r, 4.568853557635201e-4 r,
5.618675264007778e-4 r, 6.817933857540259e-4 r,
8.176509330039827e-4 r, 9.704265741758145e-4 r,
0.001141105023499428 r, 0.001330669204938795 r,
0.001540100153900437 r, 0.001770376919130678 r,
0.002022476464811601 r, 0.002297373572865413 r,
0.002596040745477063 r, 0.002919448107844891 r,
0.003268563311168871 r, 0.003644351435886262 r,
0.004047774895164447 r, 0.004479793338660443 r, 0.0049413635565565 r,
0.005433439383882244 r, 0.005956971605131645 r,
0.006512907859185624 r, 0.007102192544548636 r,
0.007725766724910044 r, 0.00838456803503801 r,
0.009079530587017326 r, 0.009811584876838586 r, 0.0105816576913495 r,
0.01139067201557714 r, 0.01223954694042984 r, 0.01312919757078923 r,
0.01406053493400045 r, 0.01503446588876983 r, 0.01605189303448024 r,
0.01711371462093175 r, 0.01822082445851714 r, 0.01937411182884202 r,
0.02057446139579705 r, 0.02182275311709253 r, 0.02311986215626333 r,
```

0.02446665879515308 r, 0.02586400834688696 r, 0.02731277106934082 r,
0.02881380207911666 r, 0.03036795126603076 r, 0.03197606320812652 r,
0.0336389770872163 r, 0.03535752660496472 r, 0.03713253989951881 r,
0.03896483946269502 r, 0.0408552420577305 r, 0.04280455863760801 r,
0.04481359426396048 r, 0.04688314802656623 r, 0.04901401296344043 r,
0.05120697598153157 r, 0.05346281777803219 r, 0.05578231276230905 r,
0.05816622897846346 r, 0.06061532802852698 r, 0.0631303649963022 r,
0.06571208837185505 r, 0.06836123997666599 r, 0.07107855488944881 r,
0.07386476137264342 r, 0.07672058079958999 r, 0.07964672758239233 r,
0.08264390910047736 r, 0.0857128256298576 r, 0.08885417027310427 r,
0.09206862889003742 r, 0.09535688002914089 r, 0.0987195948597075 r,
0.1021574371047232 r, 0.1056710629744951 r, 0.1092611211010309 r,
0.1129282524731764 r, 0.1166730903725168 r, 0.1204962603100498 r,
0.1243983799636342 r, 0.1283800591162231 r, 0.1324418995948859 r,
0.1365844952106265 r, 0.140808431699002 r, 0.1451142866615502 r,
0.1495026295080298 r, 0.1539740213994798 r],
[0, 1.66665833335744e-7 r, 1.33330666692022e-6 r,
4.499797504338432e-6 r, 1.066581336583994e-5 r,
2.083072932167196e-5 r, 3.599352055540239e-5 r,
5.71526624672386e-5 r, 8.530603082730626e-5 r,
1.214508019889565e-4 r, 1.665833531718508e-4 r,
2.216991628251896e-4 r, 2.877927110806339e-4 r,
3.658573803051457e-4 r, 4.568853557635201e-4 r,
5.618675264007778e-4 r, 6.817933857540259e-4 r,
8.176509330039827e-4 r, 9.704265741758145e-4 r,
0.001141105023499428 r, 0.001330669204938795 r,
0.001540100153900437 r, 0.001770376919130678 r,
0.002022476464811601 r, 0.002297373572865413 r,
0.002596040745477063 r, 0.002919448107844891 r,
0.003268563311168871 r, 0.003644351435886262 r,
0.004047774895164447 r, 0.004479793338660443 r, 0.0049413635565565 r,
0.005433439383882244 r, 0.005956971605131645 r,
0.006512907859185624 r, 0.007102192544548636 r,
0.007725766724910044 r, 0.00838456803503801 r,
0.009079530587017326 r, 0.009811584876838586 r, 0.0105816576913495 r,

```
0.01139067201557714 r, 0.01223954694042984 r, 0.01312919757078923 r,
0.01406053493400045 r, 0.01503446588876983 r, 0.01605189303448024 r,
0.01711371462093175 r, 0.01822082445851714 r, 0.01937411182884202 r,
0.02057446139579705 r, 0.02182275311709253 r, 0.02311986215626333 r,
0.02446665879515308 r, 0.02586400834688696 r, 0.02731277106934082 r,
0.02881380207911666 r, 0.03036795126603076 r, 0.03197606320812652 r,
0.0336389770872163 r, 0.03535752660496472 r, 0.03713253989951881 r,
0.03896483946269502 r, 0.0408552420577305 r, 0.04280455863760801 r,
0.04481359426396048 r, 0.04688314802656623 r, 0.04901401296344043 r,
0.05120697598153157 r, 0.05346281777803219 r, 0.05578231276230905 r,
0.05816622897846346 r, 0.06061532802852698 r, 0.0631303649963022 r,
0.06571208837185505 r, 0.06836123997666599 r, 0.07107855488944881 r,
0.07386476137264342 r, 0.07672058079958999 r, 0.07964672758239233 r,
0.08264390910047736 r, 0.0857128256298576 r, 0.08885417027310427 r,
0.09206862889003742 r, 0.09535688002914089 r, 0.0987195948597075 r,
0.1021574371047232 r, 0.1056710629744951 r, 0.1092611211010309 r,
0.1129282524731764 r, 0.1166730903725168 r, 0.1204962603100498 r,
0.1243983799636342 r, 0.1283800591162231 r, 0.1324418995948859 r,
0.1365844952106265 r, 0.140808431699002 r, 0.1451142866615502 r,
0.1495026295080298 r, 0.1539740213994798 r])
```

```
>$showev('limit(f(x),x,0))
```

```
Maxima said:
limit: variable must be a symbol or subscripted symbol; found:
 errexp1
#0: showev(f='limit([0,1.66665833335744e-7*r,1.33330666692022e-6*r,4.499797504338432e-6*r,1.066581
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('limit(f(x),x,0)) ...
                           ^
```

Silakan Anda gambar kurva

$$y = x^x.$$

```
>$showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$0 = 0$$

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h\to 0} \frac{(x+h)^{x+h} - x^x}{h}.$$

Dalam hal ini diperlukan asumsi nilai x.

```
>&assume(x>0); $showev('limit((f(x+h)-f(x))/h,h,0)) // turunan f(x)=x^x
```

$$0 = 0$$

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>&forget(x>0) // jangan lupa, lupakan asumsi untuk kembali ke semula
```

```
        [[0, 1.66665833335744e-7 r, 1.33330666692022e-6 r,
4.499797504338432e-6 r, 1.066581336583994e-5 r,
2.083072932167196e-5 r, 3.599352055540239e-5 r,
5.71526624672386e-5 r, 8.530603082730626e-5 r,
1.214508019889565e-4 r, 1.665833531718508e-4 r,
2.216991628251896e-4 r, 2.877927110806339e-4 r,
3.658573803051457e-4 r, 4.568853557635201e-4 r,
5.618675264007778e-4 r, 6.817933857540259e-4 r,
8.176509330039827e-4 r, 9.704265741758145e-4 r,
0.001141105023499428 r, 0.001330669204938795 r,
0.001540100153900437 r, 0.001770376919130678 r,
0.002022476464811601 r, 0.002297373572865413 r,
0.002596040745477063 r, 0.002919448107844891 r,
0.003268563311168871 r, 0.003644351435886262 r,
0.004047774895164447 r, 0.004479793338660443 r, 0.0049413635565565 r,
0.005433439383882244 r, 0.005956971605131645 r,
0.006512907859185624 r, 0.007102192544548636 r,
0.007725766724910044 r, 0.00838456803503801 r,
0.009079530587017326 r, 0.009811584876838586 r, 0.0105816576913495 r,
0.01139067201557714 r, 0.01223954694042984 r, 0.01312919757078923 r,
0.01406053493400045 r, 0.01503446588876983 r, 0.01605189303448024 r,
0.01711371462093175 r, 0.01822082445851714 r, 0.01937411182884202 r,
0.02057446139579705 r, 0.02182275311709253 r, 0.02311986215626333 r,
0.02446665879515308 r, 0.02586400834688696 r, 0.02731277106934082 r,
0.02881380207911666 r, 0.03036795126603076 r, 0.03197606320812652 r,
0.0336389770872163 r, 0.03535752660496472 r, 0.03713253989951881 r,
0.03896483946269502 r, 0.0408552420577305 r, 0.04280455863760801 r,
```

```
0.04481359426396048 r, 0.04688314802656623 r, 0.04901401296344043 r,
0.05120697598153157 r, 0.05346281777803219 r, 0.05578231276230905 r,
0.05816622897846346 r, 0.06061532802852698 r, 0.0631303649963022 r,
0.06571208837185505 r, 0.06836123997666599 r, 0.07107855488944881 r,
0.07386476137264342 r, 0.07672058079958999 r, 0.07964672758239233 r,
0.08264390910047736 r, 0.0857128256298576 r, 0.08885417027310427 r,
0.09206862889003742 r, 0.09535688002914089 r, 0.0987195948597075 r,
0.1021574371047232 r, 0.1056710629744951 r, 0.1092611211010309 r,
0.1129282524731764 r, 0.1166730903725168 r, 0.1204962603100498 r,
0.1243983799636342 r, 0.1283800591162231 r, 0.1324418995948859 r,
0.1365844952106265 r, 0.140808431699002 r, 0.1451142866615502 r,
0.1495026295080298 r, 0.1539740213994798 r] > 0]
```

```
>&forget(x<0)
```

```
        [[0, - 1.66665833335744e-7 r, - 1.33330666692022e-6 r,
- 4.499797504338432e-6 r, - 1.066581336583994e-5 r,
- 2.083072932167196e-5 r, - 3.599352055540239e-5 r,
- 5.71526624672386e-5 r, - 8.530603082730626e-5 r,
- 1.214508019889565e-4 r, - 1.665833531718508e-4 r,
- 2.216991628251896e-4 r, - 2.877927110806339e-4 r,
- 3.658573803051457e-4 r, - 4.568853557635201e-4 r,
- 5.618675264007778e-4 r, - 6.817933857540259e-4 r,
- 8.176509330039827e-4 r, - 9.704265741758145e-4 r,
- 0.001141105023499428 r, - 0.001330669204938795 r,
- 0.001540100153900437 r, - 0.001770376919130678 r,
- 0.002022476464811601 r, - 0.002297373572865413 r,
- 0.002596040745477063 r, - 0.002919448107844891 r,
- 0.003268563311168871 r, - 0.003644351435886262 r,
- 0.004047774895164447 r, - 0.004479793338660443 r,
```

```
- 0.0049413635565565 r, - 0.005433439383882244 r,
- 0.005956971605131645 r, - 0.006512907859185624 r,
- 0.007102192544548636 r, - 0.007725766724910044 r,
- 0.00838456803503801 r, - 0.009079530587017326 r,
- 0.009811584876838586 r, - 0.0105816576913495 r,
- 0.01139067201557714 r, - 0.01223954694042984 r,
- 0.01312919757078923 r, - 0.01406053493400045 r,
- 0.01503446588876983 r, - 0.01605189303448024 r,
- 0.01711371462093175 r, - 0.01822082445851714 r,
- 0.01937411182884202 r, - 0.02057446139579705 r,
- 0.02182275311709253 r, - 0.02311986215626333 r,
- 0.02446665879515308 r, - 0.02586400834688696 r,
- 0.02731277106934082 r, - 0.02881380207911666 r,
- 0.03036795126603076 r, - 0.03197606320812652 r,
- 0.0336389770872163 r, - 0.03535752660496472 r,
- 0.03713253989951881 r, - 0.03896483946269502 r,
- 0.0408552420577305 r, - 0.04280455863760801 r,
- 0.04481359426396048 r, - 0.04688314802656623 r,
- 0.04901401296344043 r, - 0.05120697598153157 r,
- 0.05346281777803219 r, - 0.05578231276230905 r,
- 0.05816622897846346 r, - 0.06061532802852698 r,
- 0.0631303649963022 r, - 0.06571208837185505 r,
- 0.06836123997666599 r, - 0.07107855488944881 r,
- 0.07386476137264342 r, - 0.07672058079958999 r,
- 0.07964672758239233 r, - 0.08264390910047736 r,
- 0.0857128256298576 r, - 0.08885417027310427 r,
- 0.09206862889003742 r, - 0.09535688002914089 r,
- 0.0987195948597075 r, - 0.1021574371047232 r,
- 0.1056710629744951 r, - 0.1092611211010309 r,
- 0.1129282524731764 r, - 0.1166730903725168 r,
- 0.1204962603100498 r, - 0.1243983799636342 r,
- 0.1283800591162231 r, - 0.1324418995948859 r,
- 0.1365844952106265 r, - 0.140808431699002 r,
- 0.1451142866615502 r, - 0.1495026295080298 r,
- 0.1539740213994798 r] > 0]
```

```
>&facts()
```

```
        [[0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09,
0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21,
0.2200000000000001, 0.2300000000000001, 0.2400000000000001,
0.2500000000000001, 0.2600000000000001, 0.2700000000000001,
0.2800000000000001, 0.2900000000000001, 0.3000000000000001,
0.3100000000000001, 0.3200000000000001, 0.3300000000000001,
0.3400000000000001, 0.3500000000000001, 0.3600000000000002,
0.3700000000000002, 0.3800000000000002, 0.3900000000000002,
0.4000000000000002, 0.4100000000000002, 0.4200000000000002,
0.4300000000000002, 0.4400000000000002, 0.4500000000000002,
0.4600000000000002, 0.4700000000000003, 0.4800000000000003,
0.4900000000000003, 0.5000000000000002, 0.5100000000000002,
0.5200000000000002, 0.5300000000000002, 0.5400000000000003,
0.5500000000000003, 0.5600000000000003, 0.5700000000000003,
0.5800000000000003, 0.5900000000000003, 0.6000000000000003,
0.6100000000000003, 0.6200000000000003, 0.6300000000000003,
0.6400000000000003, 0.6500000000000004, 0.6600000000000004,
0.6700000000000004, 0.6800000000000004, 0.6900000000000004,
0.7000000000000004, 0.7100000000000004, 0.7200000000000004,
0.7300000000000004, 0.7400000000000004, 0.7500000000000004,
0.7600000000000005, 0.7700000000000005, 0.7800000000000005,
0.7900000000000005, 0.8000000000000005, 0.8100000000000005,
0.8200000000000005, 0.8300000000000005, 0.8400000000000005,
0.8500000000000005, 0.8600000000000005, 0.8700000000000006,
0.8800000000000006, 0.8900000000000006, 0.9000000000000006,
0.9100000000000006, 0.9200000000000006, 0.9300000000000006,
0.9400000000000006, 0.9500000000000006, 0.9600000000000006,
0.9700000000000006, 0.9800000000000006, 0.9900000000000007] > 0,
```

```
r > 0, [1, 1 - 1.66665833335744e-7 r, 1 - 1.33330666692022e-6 r,
1 - 4.499797504338432e-6 r, 1 - 1.066581336583994e-5 r,
1 - 2.083072932167196e-5 r, 1 - 3.599352055540239e-5 r,
1 - 5.71526624672386e-5 r, 1 - 8.530603082730626e-5 r,
1 - 1.214508019889565e-4 r, 1 - 1.665833531718508e-4 r,
1 - 2.216991628251896e-4 r, 1 - 2.877927110806339e-4 r,
1 - 3.658573803051457e-4 r, 1 - 4.568853557635201e-4 r,
1 - 5.618675264007778e-4 r, 1 - 6.817933857540259e-4 r,
1 - 8.176509330039827e-4 r, 1 - 9.704265741758145e-4 r,
1 - 0.001141105023499428 r, 1 - 0.001330669204938795 r,
1 - 0.001540100153900437 r, 1 - 0.001770376919130678 r,
1 - 0.002022476464811601 r, 1 - 0.002297373572865413 r,
1 - 0.002596040745477063 r, 1 - 0.002919448107844891 r,
1 - 0.003268563311168871 r, 1 - 0.003644351435886262 r,
1 - 0.004047774895164447 r, 1 - 0.004479793338660443 r,
1 - 0.0049413635565565 r, 1 - 0.005433439383882244 r,
1 - 0.005956971605131645 r, 1 - 0.006512907859185624 r,
1 - 0.007102192544548636 r, 1 - 0.007725766724910044 r,
1 - 0.00838456803503801 r, 1 - 0.009079530587017326 r,
1 - 0.009811584876838586 r, 1 - 0.0105816576913495 r,
1 - 0.01139067201557714 r, 1 - 0.01223954694042984 r,
1 - 0.01312919757078923 r, 1 - 0.01406053493400045 r,
1 - 0.01503446588876983 r, 1 - 0.01605189303448024 r,
1 - 0.01711371462093175 r, 1 - 0.01822082445851714 r,
1 - 0.01937411182884202 r, 1 - 0.02057446139579705 r,
1 - 0.02182275311709253 r, 1 - 0.02311986215626333 r,
1 - 0.02446665879515308 r, 1 - 0.02586400834688696 r,
1 - 0.02731277106934082 r, 1 - 0.02881380207911666 r,
1 - 0.03036795126603076 r, 1 - 0.03197606320812652 r,
1 - 0.0336389770872163 r, 1 - 0.03535752660496472 r,
1 - 0.03713253989951881 r, 1 - 0.03896483946269502 r,
1 - 0.0408552420577305 r, 1 - 0.04280455863760801 r,
1 - 0.04481359426396048 r, 1 - 0.04688314802656623 r,
1 - 0.04901401296344043 r, 1 - 0.05120697598153157 r,
1 - 0.05346281777803219 r, 1 - 0.05578231276230905 r,
```

```
1 - 0.05816622897846346 r, 1 - 0.06061532802852698 r,
1 - 0.0631303649963022 r, 1 - 0.06571208837185505 r,
1 - 0.06836123997666599 r, 1 - 0.07107855488944881 r,
1 - 0.07386476137264342 r, 1 - 0.07672058079958999 r,
1 - 0.07964672758239233 r, 1 - 0.08264390910047736 r,
1 - 0.0857128256298576 r, 1 - 0.08885417027310427 r,
1 - 0.09206862889003742 r, 1 - 0.09535688002914089 r,
1 - 0.0987195948597075 r, 1 - 0.1021574371047232 r,
1 - 0.1056710629744951 r, 1 - 0.1092611211010309 r,
1 - 0.1129282524731764 r, 1 - 0.1166730903725168 r,
1 - 0.1204962603100498 r, 1 - 0.1243983799636342 r,
1 - 0.1283800591162231 r, 1 - 0.1324418995948859 r,
1 - 0.1365844952106265 r, 1 - 0.140808431699002 r,
1 - 0.1451142866615502 r, 1 - 0.1495026295080298 r,
1 - 0.1539740213994798 r] > 0, g26798 >= 0, g26846 >= 0]
```

```
>$showev('limit((asin(x+h)-asin(x))/h,h,0)) // turunan arcsin(x)
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>$showev('limit((tan(x+h)-tan(x))/h,h,0)) // turunan tan(x)
```

Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.

```
>function f(x) &= sinh(x) // definisikan f(x)=sinh(x)
```

```
        [0, sinh(1.66665833335744e-7 r), sinh(1.33330666692022e-6 r),
sinh(4.499797504338432e-6 r), sinh(1.066581336583994e-5 r),
sinh(2.083072932167196e-5 r), sinh(3.599352055540239e-5 r),
sinh(5.71526624672386e-5 r), sinh(8.530603082730626e-5 r),
sinh(1.214508019889565e-4 r), sinh(1.665833531718508e-4 r),
sinh(2.216991628251896e-4 r), sinh(2.877927110806339e-4 r),
sinh(3.658573803051457e-4 r), sinh(4.568853557635201e-4 r),
sinh(5.618675264007778e-4 r), sinh(6.817933857540259e-4 r),
sinh(8.176509330039827e-4 r), sinh(9.704265741758145e-4 r),
sinh(0.001141105023499428 r), sinh(0.001330669204938795 r),
sinh(0.001540100153900437 r), sinh(0.001770376919130678 r),
sinh(0.002022476464811601 r), sinh(0.002297373572865413 r),
sinh(0.002596040745477063 r), sinh(0.002919448107844891 r),
sinh(0.003268563311168871 r), sinh(0.003644351435886262 r),
sinh(0.004047774895164447 r), sinh(0.004479793338660443 r),
sinh(0.0049413635565565 r), sinh(0.005433439383882244 r),
sinh(0.005956971605131645 r), sinh(0.006512907859185624 r),
sinh(0.007102192544548636 r), sinh(0.007725766724910044 r),
sinh(0.00838456803503801 r), sinh(0.009079530587017326 r),
sinh(0.009811584876838586 r), sinh(0.0105816576913495 r),
sinh(0.01139067201557714 r), sinh(0.01223954694042984 r),
sinh(0.01312919757078923 r), sinh(0.01406053493400045 r),
sinh(0.01503446588876983 r), sinh(0.01605189303448024 r),
sinh(0.01711371462093175 r), sinh(0.01822082445851714 r),
```

```
sinh(0.01937411182884202 r), sinh(0.02057446139579705 r),
sinh(0.02182275311709253 r), sinh(0.02311986215626333 r),
sinh(0.02446665879515308 r), sinh(0.02586400834688696 r),
sinh(0.02731277106934082 r), sinh(0.02881380207911666 r),
sinh(0.03036795126603076 r), sinh(0.03197606320812652 r),
sinh(0.0336389770872163 r), sinh(0.03535752660496472 r),
sinh(0.03713253989951881 r), sinh(0.03896483946269502 r),
sinh(0.0408552420577305 r), sinh(0.04280455863760801 r),
sinh(0.04481359426396048 r), sinh(0.04688314802656623 r),
sinh(0.04901401296344043 r), sinh(0.05120697598153157 r),
sinh(0.05346281777803219 r), sinh(0.05578231276230905 r),
sinh(0.05816622897846346 r), sinh(0.06061532802852698 r),
sinh(0.0631303649963022 r), sinh(0.06571208837185505 r),
sinh(0.06836123997666599 r), sinh(0.07107855488944881 r),
sinh(0.07386476137264342 r), sinh(0.07672058079958999 r),
sinh(0.07964672758239233 r), sinh(0.08264390910047736 r),
sinh(0.0857128256298576 r), sinh(0.08885417027310427 r),
sinh(0.09206862889003742 r), sinh(0.09535688002914089 r),
sinh(0.0987195948597075 r), sinh(0.1021574371047232 r),
sinh(0.1056710629744951 r), sinh(0.1092611211010309 r),
sinh(0.1129282524731764 r), sinh(0.1166730903725168 r),
sinh(0.1204962603100498 r), sinh(0.1243983799636342 r),
sinh(0.1283800591162231 r), sinh(0.1324418995948859 r),
sinh(0.1365844952106265 r), sinh(0.140808431699002 r),
sinh(0.1451142866615502 r), sinh(0.1495026295080298 r),
sinh(0.1539740213994798 r)]
```

```
>function df(x) &= limit((f(x+h)-f(x))/h,h,0); $df(x) // df(x) = f'(x)
```

[0, 0, 0,

Hasilnya adalah cosh(x), karena

$$\frac{e^x + e^{-x}}{2} = \cosh(x).$$

```
>plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
```

```
Error : f(x) does not produce a real or column vector

Error generated by error() command

 %ploteval:
    error(f$|" does not produce a real or column vector");
adaptiveevalone:
    s=%ploteval(g$,t;args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    dw/n,dw/n^2,dw/n,auto;args());
```

```
>function f(x) &= sin(3*x^5+7)^2
```

```
            2        2                            5
         [sin (7), sin (7 + 3.857928241984252e-34 r ),
   2                            5
sin (7 + 1.264071117369734e-29 r ),
   2                            5
sin (7 + 5.534598323932662e-27 r ),
   2                            5
```

```
sin (7 + 4.140865788349747e-25 r ),
   2                            5
sin (7 + 1.176640067174869e-23 r ),
   2                            5
sin (7 + 1.812353420702142e-22 r ),
   2                            5
sin (7 + 1.829378577899295e-21 r ),
   2                            5
sin (7 + 1.355251607920358e-20 r ),
   2                            5
sin (7 + 7.927261589391743e-20 r ),
   2                            5
sin (7 + 3.848391561406479e-19 r ),
   2                            5
sin (7 + 1.606724886280612e-18 r ),
   2                            5
sin (7 + 5.922706430403385e-18 r ),
   2                            5
sin (7 + 1.966438444926739e-17 r ),
   2                            5
sin (7 + 5.972517698531353e-17 r ),
   2                            5
sin (7 + 1.679928959568682e-16 r ),
   2                            5
sin (7 + 4.419622473786581e-16 r ),
   2                            5
sin (7 + 1.096379579300723e-15 r ),
   2                            5
sin (7 + 2.581871707303932e-15 r ),
   2                            5
sin (7 + 5.804293182204869e-15 r ),
   2                            5
sin (7 + 1.251617959779806e-14 r ),
   2                            5
sin (7 + 2.599357872032985e-14 r ),
```

```
   2                            5
sin (7 + 5.217349730801036e-14 r ),
   2                            5      2                           5
sin (7 + 1.015169677716456e-13 r ), sin (7 + 1.9199033201535e-13 r ),
   2                           5
sin (7 + 3.53735606461862e-13 r ),
   2                            5
sin (7 + 6.362459865941665e-13 r ),
   2                            5
sin (7 + 1.119195003037803e-12 r ),
   2                            5
sin (7 + 1.928512721307568e-12 r ),
   2                            5
sin (7 + 3.259890537530715e-12 r ),
   2                            5
sin (7 + 5.412665053214709e-12 r ),
   2                            5
sin (7 + 8.838026394825634e-12 r ),
   2                            5
sin (7 + 1.420677121368262e-11 r ),
   2                            5
sin (7 + 2.250343962315998e-11 r ),
   2                            5
sin (7 + 3.515571419086806e-11 r ),
   2                           5      2                           5
sin (7 + 5.42105065269614e-11 r ), sin (7 + 8.25713163932381e-11 r ),
   2                            5
sin (7 + 1.243153394347951e-10 r ),
   2                            5      2                           5
sin (7 + 1.851135607682476e-10 r ), sin (7 + 2.7278286125595e-10 r ),
   2                            5
sin (7 + 3.980061623597386e-10 r ),
   2                            5
sin (7 + 5.752650497791658e-10 r ),
   2                            5
```

```
sin (7 + 8.240393785975726e-10 r ),
   2                            5      2                            5
sin (7 + 1.170340336056673e-9 r ), sin (7 + 1.648657617393293e-9 r ),
   2                            5      2                            5
sin (7 + 2.304418085581079e-9 r ), sin (7 + 3.197072905534091e-9 r ),
   2                            5      2                            5
sin (7 + 4.403953076583591e-9 r ), sin (7 + 6.02505998720235e-9 r ),
   2                            5      2                            5
sin (7 + 8.188988583715194e-9 r ), sin (7 + 1.106021653139266e-8 r ),
   2                            5      2                            5
sin (7 + 1.484803395715922e-8 r ), sin (7 + 1.981743566080698e-8 r ),
   2                            5      2                            5
sin (7 + 2.630235178964736e-8 r ), sin (7 + 3.472165467790206e-8 r ),
   2                            5      2                            5
sin (7 + 4.559844971273112e-8 r ), sin (7 + 5.958323763108241e-8 r ),
   2                            5      2                            5
sin (7 + 7.748162557801914e-8 r ), sin (7 + 1.002873656491318e-7 r ),
   2                          5      2                          5
sin (7 + 1.2922161366027e-7 r ), sin (7 + 1.657794287864104e-7 r ),
   2                            5      2                            5
sin (7 + 2.117846778257662e-7 r ), sin (7 + 2.694546676059425e-7 r ),
   2                            5      2                            5
sin (7 + 3.414760069817971e-7 r ), sin (7 + 4.310933976044147e-7 r ),
   2                            5      2                            5
sin (7 + 5.422132718931149e-7 r ), sin (7 + 6.795244392485226e-7 r ),
   2                            5      2                            5
sin (7 + 8.486381694408711e-7 r ), sin (7 + 1.056250437340941e-6 r ),
   2                          5      2                          5
sin (7 + 1.31032937788846e-6 r ), sin (7 + 1.620331356684433e-6 r ),
   2                            5      2                            5
sin (7 + 1.997449452236024e-6 r ), sin (7 + 2.454898573173933e-6 r ),
   2                            5      2                            5
sin (7 + 3.008241900322126e-6 r ), sin (7 + 3.675763852060344e-6 r ),
   2                            5      2                            5
sin (7 + 4.478895324833735e-6 r ), sin (7 + 5.442697561899192e-6 r ),
```

```
   2                            5      2                            5
sin (7 + 6.596411655537407e-6 r ), sin (7 + 7.974081394203433e-6 r ),
   2                            5      2                            5
sin (7 + 9.615257929748426e-6 r ), sin (7 + 1.156579556434297e-5 r ),
   2                            5      2                            5
sin (7 + 1.387874884559973e-5 r ), sin (7 + 1.661538211526619e-5 r ),
   2                            5      2                            5
sin (7 + 1.984630368547185e-5 r ), sin (7 + 2.365273792070429e-5 r ),
   2                            5      2                            5
sin (7 + 2.812794968737494e-5 r ), sin (7 + 3.337883690003698e-5 r ),
   2                            5      2                           5
sin (7 + 3.952770824811069e-5 r ), sin (7 + 4.67142646335361e-5 r ),
   2                            5      2                            5
sin (7 + 5.509780439231647e-5 r ), sin (7 + 6.485967401573982e-5 r ),
   2                            5      2                            5
sin (7 + 7.620598783449127e-5 r ), sin (7 + 8.937064198526321e-5 r ),
   2                            5      2                            5
sin (7 + 1.046186499492426e-4 r ), sin (7 + 1.222498290394089e-4 r ),
   2                            5      2                            5
sin (7 + 1.426028694233221e-4 r ), sin (7 + 1.660598196044561e-4 r ),
   2                            5      2                            5
sin (7 + 1.930510247525017e-4 r ), sin (7 + 2.240605568757864e-4 r ),
   2                            5
sin (7 + 2.596321785713566e-4 r )]
```

```
>diff(f,3), diffc(f,3)
```

```
[1.59563e-13,  1.59563e-13,  1.59563e-13,  1.59563e-13,  1.59563e-13,
1.59563e-13,  1.59563e-13,  1.59563e-13,  1.59563e-13,  1.59563e-13,
1.59563e-13,  1.59563e-13,  1.59563e-13,  1.59563e-13,  -1.59563e-13,
-1.59563e-13,  -1.59563e-13,  1.59563e-13,  -2.1275e-13,  2.1275e-13,
```

```
-5.31876e-14,  -1.59563e-13,  1.59563e-13,  -1.06375e-13,  0,
-1.59563e-13,  -1.59563e-13,  1.06375e-13,  -5.31876e-14,
-1.59563e-13,  -2.1275e-13,  -1.06375e-13,  2.1275e-13,  1.59563e-13,
-1.59563e-13,  5.31876e-14,  -5.31876e-14,  5.31876e-14,  5.31876e-14,
-5.31876e-14,  -1.59563e-13,  1.59563e-13,  -2.1275e-13,
-1.06375e-13,  1.59563e-13,  -1.59563e-13,  -2.1275e-13,  1.59563e-13,
-1.59563e-13,  1.06375e-13,  2.1275e-13,  -2.1275e-13,  -5.31876e-14,
-1.59563e-13,  -1.59563e-13,  1.59563e-13,  1.59563e-13,  1.59563e-13,
0,  -5.31876e-14,  2.1275e-13,  -1.06375e-13,  -5.31876e-14,
5.31876e-14,  -1.59563e-13,  1.06375e-13,  -1.06375e-13,  0,
2.1275e-13,  1.59563e-13,  1.59563e-13,  1.59563e-13,  5.31876e-14,
5.31876e-14,  2.1275e-13,  -1.59563e-13,  0,  -5.31876e-14,
-5.31876e-14,  -1.59563e-13,  -5.31876e-14,  1.06375e-13,
-1.59563e-13,  0,  -1.06375e-13,  -5.31876e-14,  1.06375e-13,
1.59563e-13,  -1.59563e-13,  -5.31876e-14,  -5.31876e-14,
-1.59563e-13,  -1.59563e-13,  1.59563e-13,  -1.06375e-13,
 ... ]
[0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,  0,  0,  0,  0,  0,  0,  0,  0,  0]
```

Apakah perbedaan diff dan diffc?

```
>$showev('diff(f(x),x))
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
#0: showev(f='diff([sin(7)^2,sin(7+3.857928241984252e-34*r^5)^2,sin(7+1.264071117369734e-29*r^5)^2
```

```
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('diff(f(x),x)) ...
                        ^
```

```
>$% with x=3
```

[0, 0, 0,

```
plot2d:
    dw/n,dw/n^2,dw/n,auto;args());
```

```
>function f(x) &=5*cos(2*x)-2*x*sin(2*x) // mendifinisikan fungsi f
```

```
        [5, 5 cos(3.333316666714881e-7 r)
 - 3.333316666714881e-7 r sin(3.333316666714881e-7 r),
5 cos(2.66661333384044e-6 r) - 2.66661333384044e-6 r
 sin(2.66661333384044e-6 r), 5 cos(8.999595008676864e-6 r)
 - 8.999595008676864e-6 r sin(8.999595008676864e-6 r),
5 cos(2.133162673167988e-5 r) - 2.133162673167988e-5 r
 sin(2.133162673167988e-5 r), 5 cos(4.166145864334392e-5 r)
 - 4.166145864334392e-5 r sin(4.166145864334392e-5 r),
5 cos(7.198704111080478e-5 r) - 7.198704111080478e-5 r
 sin(7.198704111080478e-5 r), 5 cos(1.143053249344772e-4 r)
 - 1.143053249344772e-4 r sin(1.143053249344772e-4 r),
5 cos(1.706120616546125e-4 r) - 1.706120616546125e-4 r
 sin(1.706120616546125e-4 r), 5 cos(2.42901603977913e-4 r)
 - 2.42901603977913e-4 r sin(2.42901603977913e-4 r),
5 cos(3.331667063437016e-4 r) - 3.331667063437016e-4 r
 sin(3.331667063437016e-4 r), 5 cos(4.433983256503793e-4 r)
 - 4.433983256503793e-4 r sin(4.433983256503793e-4 r),
5 cos(5.755854221612677e-4 r) - 5.755854221612677e-4 r
 sin(5.755854221612677e-4 r), 5 cos(7.317147606102914e-4 r)
 - 7.317147606102914e-4 r sin(7.317147606102914e-4 r),
5 cos(9.137707115270399e-4 r) - 9.137707115270399e-4 r
 sin(9.137707115270399e-4 r), 5 cos(0.001123735052801556 r)
 - 0.001123735052801556 r sin(0.001123735052801556 r),
5 cos(0.001363586771508052 r) - 0.001363586771508052 r
 sin(0.001363586771508052 r), 5 cos(0.001635301866007965 r)
 - 0.001635301866007965 r sin(0.001635301866007965 r),
```

```
5 cos(0.001940853148351629 r) - 0.001940853148351629 r
 sin(0.001940853148351629 r), 5 cos(0.002282210046998856 r)
 - 0.002282210046998856 r sin(0.002282210046998856 r),
5 cos(0.002661338409877589 r) - 0.002661338409877589 r
 sin(0.002661338409877589 r), 5 cos(0.003080200307800873 r)
 - 0.003080200307800873 r sin(0.003080200307800873 r),
5 cos(0.003540753838261357 r) - 0.003540753838261357 r
 sin(0.003540753838261357 r), 5 cos(0.004044952929623202 r)
 - 0.004044952929623202 r sin(0.004044952929623202 r),
5 cos(0.004594747145730826 r) - 0.004594747145730826 r
 sin(0.004594747145730826 r), 5 cos(0.005192081490954126 r)
 - 0.005192081490954126 r sin(0.005192081490954126 r),
5 cos(0.005838896215689782 r) - 0.005838896215689782 r
 sin(0.005838896215689782 r), 5 cos(0.006537126622337741 r)
 - 0.006537126622337741 r sin(0.006537126622337741 r),
5 cos(0.007288702871772523 r) - 0.007288702871772523 r
 sin(0.007288702871772523 r), 5 cos(0.008095549790328893 r)
 - 0.008095549790328893 r sin(0.008095549790328893 r),
5 cos(0.008959586677320885 r) - 0.008959586677320885 r
 sin(0.008959586677320885 r), 5 cos(0.009882727113112999 r)
 - 0.009882727113112999 r sin(0.009882727113112999 r),
5 cos(0.01086687876776449 r) - 0.01086687876776449 r
 sin(0.01086687876776449 r), 5 cos(0.01191394321026329 r)
 - 0.01191394321026329 r sin(0.01191394321026329 r),
5 cos(0.01302581571837125 r) - 0.01302581571837125 r
 sin(0.01302581571837125 r), 5 cos(0.01420438508909727 r)
 - 0.01420438508909727 r sin(0.01420438508909727 r),
5 cos(0.01545153344982009 r) - 0.01545153344982009 r
 sin(0.01545153344982009 r), 5 cos(0.01676913607007602 r)
 - 0.01676913607007602 r sin(0.01676913607007602 r),
5 cos(0.01815906117403465 r) - 0.01815906117403465 r
 sin(0.01815906117403465 r), 5 cos(0.01962316975367717 r)
 - 0.01962316975367717 r sin(0.01962316975367717 r),
5 cos(0.021163315382699 r) - 0.021163315382699 r
 sin(0.021163315382699 r), 5 cos(0.02278134403115428 r)
```

```
 - 0.02278134403115428 r sin(0.02278134403115428 r),
5 cos(0.02447909388085967 r) - 0.02447909388085967 r
 sin(0.02447909388085967 r), 5 cos(0.02625839514157846 r)
 - 0.02625839514157846 r sin(0.02625839514157846 r),
5 cos(0.02812106986800089 r) - 0.02812106986800089 r
 sin(0.02812106986800089 r), 5 cos(0.03006893177753966 r)
 - 0.03006893177753966 r sin(0.03006893177753966 r),
5 cos(0.03210378606896047 r) - 0.03210378606896047 r
 sin(0.03210378606896047 r), 5 cos(0.03422742924186351 r)
 - 0.03422742924186351 r sin(0.03422742924186351 r),
5 cos(0.03644164891703428 r) - 0.03644164891703428 r
 sin(0.03644164891703428 r), 5 cos(0.03874822365768404 r)
 - 0.03874822365768404 r sin(0.03874822365768404 r),
5 cos(0.0411489227915941 r) - 0.0411489227915941 r
 sin(0.0411489227915941 r), 5 cos(0.04364550623418506 r)
 - 0.04364550623418506 r sin(0.04364550623418506 r),
5 cos(0.04623972431252665 r) - 0.04623972431252665 r
 sin(0.04623972431252665 r), 5 cos(0.04893331759030617 r)
 - 0.04893331759030617 r sin(0.04893331759030617 r),
5 cos(0.05172801669377391 r) - 0.05172801669377391 r
 sin(0.05172801669377391 r), 5 cos(0.05462554213868165 r)
 - 0.05462554213868165 r sin(0.05462554213868165 r),
5 cos(0.05762760415823331 r) - 0.05762760415823331 r
 sin(0.05762760415823331 r), 5 cos(0.06073590253206151 r)
 - 0.06073590253206151 r sin(0.06073590253206151 r),
5 cos(0.06395212641625303 r) - 0.06395212641625303 r
 sin(0.06395212641625303 r), 5 cos(0.06727795417443261 r)
 - 0.06727795417443261 r sin(0.06727795417443261 r),
5 cos(0.07071505320992943 r) - 0.07071505320992943 r
 sin(0.07071505320992943 r), 5 cos(0.07426507979903763 r)
 - 0.07426507979903763 r sin(0.07426507979903763 r),
5 cos(0.07792967892539004 r) - 0.07792967892539004 r
 sin(0.07792967892539004 r), 5 cos(0.081710484115461 r)
 - 0.081710484115461 r sin(0.081710484115461 r),
5 cos(0.08560911727521603 r) - 0.08560911727521603 r
```

```
 sin(0.08560911727521603 r), 5 cos(0.08962718852792095 r)
 - 0.08962718852792095 r sin(0.08962718852792095 r),
5 cos(0.09376629605313247 r) - 0.09376629605313247 r
 sin(0.09376629605313247 r), 5 cos(0.09802802592688087 r)
 - 0.09802802592688087 r sin(0.09802802592688087 r),
5 cos(0.1024139519630631 r) - 0.1024139519630631 r
 sin(0.1024139519630631 r), 5 cos(0.1069256355560644 r)
 - 0.1069256355560644 r sin(0.1069256355560644 r),
5 cos(0.1115646255246181 r) - 0.1115646255246181 r
 sin(0.1115646255246181 r), 5 cos(0.1163324579569269 r)
 - 0.1163324579569269 r sin(0.1163324579569269 r),
5 cos(0.121230656057054 r) - 0.121230656057054 r
 sin(0.121230656057054 r), 5 cos(0.1262607299926044 r)
 - 0.1262607299926044 r sin(0.1262607299926044 r),
5 cos(0.1314241767437101 r) - 0.1314241767437101 r
 sin(0.1314241767437101 r), 5 cos(0.136722479953332 r)
 - 0.136722479953332 r sin(0.136722479953332 r),
5 cos(0.1421571097788976 r) - 0.1421571097788976 r
 sin(0.1421571097788976 r), 5 cos(0.1477295227452868 r)
 - 0.1477295227452868 r sin(0.1477295227452868 r),
5 cos(0.15344116159918 r) - 0.15344116159918 r
 sin(0.15344116159918 r), 5 cos(0.1592934551647847 r)
 - 0.1592934551647847 r sin(0.1592934551647847 r),
5 cos(0.1652878182009547 r) - 0.1652878182009547 r
 sin(0.1652878182009547 r), 5 cos(0.1714256512597152 r)
 - 0.1714256512597152 r sin(0.1714256512597152 r),
5 cos(0.1777083405462085 r) - 0.1777083405462085 r
 sin(0.1777083405462085 r), 5 cos(0.1841372577800748 r)
 - 0.1841372577800748 r sin(0.1841372577800748 r),
5 cos(0.1907137600582818 r) - 0.1907137600582818 r
 sin(0.1907137600582818 r), 5 cos(0.197439189719415 r)
 - 0.197439189719415 r sin(0.197439189719415 r),
5 cos(0.2043148742094465 r) - 0.2043148742094465 r
 sin(0.2043148742094465 r), 5 cos(0.2113421259489903 r)
 - 0.2113421259489903 r sin(0.2113421259489903 r),
```

```
5 cos(0.2185222422020618 r) - 0.2185222422020618 r
 sin(0.2185222422020618 r), 5 cos(0.2258565049463528 r)
 - 0.2258565049463528 r sin(0.2258565049463528 r),
5 cos(0.2333461807450337 r) - 0.2333461807450337 r
 sin(0.2333461807450337 r), 5 cos(0.2409925206200996 r)
 - 0.2409925206200996 r sin(0.2409925206200996 r),
5 cos(0.2487967599272685 r) - 0.2487967599272685 r
 sin(0.2487967599272685 r), 5 cos(0.2567601182324462 r)
 - 0.2567601182324462 r sin(0.2567601182324462 r),
5 cos(0.2648837991897719 r) - 0.2648837991897719 r
 sin(0.2648837991897719 r), 5 cos(0.2731689904212531 r)
 - 0.2731689904212531 r sin(0.2731689904212531 r),
5 cos(0.2816168633980041 r) - 0.2816168633980041 r
 sin(0.2816168633980041 r), 5 cos(0.2902285733231005 r)
 - 0.2902285733231005 r sin(0.2902285733231005 r),
5 cos(0.2990052590160597 r) - 0.2990052590160597 r
 sin(0.2990052590160597 r), 5 cos(0.3079480427989596 r)
 - 0.3079480427989596 r sin(0.3079480427989596 r)]
```

```
>function df(x) &=diff(f(x),x) // fd(x) = f'(x)
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
function df(x) &=diff(f(x),x)  ...
                              ^
```

```
>$'f(1)=f(1), $float(f(1)), $'f(2)=f(2), $float(f(2)) // nilai f(1) dan f(2)
```

```
>xp=solve("df(x)",1,2,0) // solusi f'(x)=0 pada interval [1, 2]
```

```
Function df not found.
Try list ... to find functions!
Error in expression: df(x)
secant:
    y0=f$(x0,args())-y; y1=f$(x1,args())-y;
Try "trace errors" to inspect local variables after errors.
solve:
    if eps then return secant(f$,a,b,y;args(),eps=eps);
```

```
>df(xp), f(xp) // cek bahwa f'(xp)=0 dan nilai ekstrim di titik tersebut
```

```
Function df not found.
Try list ... to find functions!
Error in:
df(xp), f(xp) // cek bahwa f'(xp)=0 dan nilai ekstrim di titik ...
      ^
```

```
>plot2d(["f(x)","df(x)"],0,2*pi,color=[blue,red]): //grafik fungsi dan turunannya
```

```
Error : f(x) does not produce a real or column vector

Error generated by error() command

 %ploteval:
    error(f$|" does not produce a real or column vector");
adaptiveevalone:
    s=%ploteval(g$,t;args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    dw/n,dw/n^2,dw/n,auto;args());
```

Perhatikan titik-titik "puncak" grafik y=f(x) dan nilai turunan pada saat grafik fungsinya mencapai titik "puncak" tersebut.

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), menggunakan perintah diff, dan secara manual (langkah demi langkah yang dihitung dengan Maxima) seperti contoh-contoh di atas. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

Integral

EMT dapat digunakan untuk menghitung integral, baik integral tak tentu maupun integral tentu. Untuk integral tak tentu (simbolik) sudah tentu EMT menggunakan Maxima, sedangkan untuk perhitungan integral tentu EMT sudah menyediakan beberapa fungsi yang mengimplementasikan algoritma kuadratur (perhitungan integral tentu menggunakan metode numerik).

Pada notebook ini akan ditunjukkan perhitungan integral tentu dengan menggunakan Teorema Dasar Kalkulus:

$$\int_a^b f(x)\ dx = F(b) - F(a), \qquad \text{dengan } F'(x) = f(x).$$

Fungsi untuk menentukan integral adalah integrate. Fungsi ini dapat digunakan untuk menentukan, baik integral tentu maupun tak tentu (jika fungsinya memiliki antiderivatif). Untuk perhitungan integral tentu fungsi integrate menggunakan metode numerik (kecuali fungsinya tidak integrabel, kita tidak akan menggunakan metode ini).

```
>$showev('integrate(x^n,x))
```

```
Maxima output too long!
Error in:
 $showev('integrate(x^n,x)) ...
                           ^
```

```
>$showev('integrate(1/(1+x),x))
```

```
Maxima output too long!
Error in:
 $showev('integrate(1/(1+x),x)) ...
                                ^
```

```
>$showev('integrate(1/(1+x^2),x))
```

```
Maxima output too long!
Error in:
 $showev('integrate(1/(1+x^2),x)) ...
                                 ^
```

```
>$showev('integrate(1/sqrt(1-x^2),x))
```

```
Maxima output too long!
Error in:
 $showev('integrate(1/sqrt(1-x^2),x)) ...
                                      ^
```

```
>$showev('integrate(sin(x),x,0,pi))
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
```

```
defint: found errexp1
#0: showev(f='integrate([0,sin(1.66665833335744e-7*r),sin(1.33330666692022e-6*r),sin(4.49979750433
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('integrate(sin(x),x,0,pi)) ...
                                   ^
```

```
>plot2d("sin(x)",0,2*pi):
```

```
>$showev('integrate(sin(x),x,a,b))
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([0,sin(1.66665833335744e-7*r),sin(1.33330666692022e-6*r),sin(4.49979750433
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('integrate(sin(x),x,a,b)) ...
                                  ^
```

```
>$showev('integrate(x^n,x,a,b))
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([0,1.66665833335744e-7^n*r^n,1.33330666692022e-6^n*r^n,4.499797504338432e-
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('integrate(x^n,x,a,b)) ...
                              ^
```

```
>$showev('integrate(x^2*sqrt(2*x+1),x))
```

```
Maxima output too long!
Error in:
 $showev('integrate(x^2*sqrt(2*x+1),x)) ...
                                      ^
```

```
>$showev('integrate(x^2*sqrt(2*x+1),x,0,2))
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([0,2.7777500001498e-14*sqrt(1+3.333316666714881e-7*r)*r^2,1.77770666805390
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('integrate(x^2*sqrt(2*x+1),x,0,2)) ...
                                            ^
```

```
>$ratsimp(%)
>$showev('integrate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2))
```

```
Maxima said:
expt: undefined: 0 to a negative exponent.
 -- an error. To debug this try: debugmode(true);

Error in:
... ate((sin(sqrt(x)+a)*E^sqrt(x))/sqrt(x),x,0,pi^2)) ...
                                                     ^
```

```
>$factor(%)
>function map f(x) &= E^(-x^2)
```

```
                                  2                            2
            - 2.7777500001498e-14 r    - 1.777706668053906e-12 r
       [1, E                        , E                          ,
                          2                             2
 - 2.024817758005038e-11 r    - 1.137595747549299e-10 r
E                          , E                          ,
                          2                             2
 - 4.339192840727639e-10 r    - 1.295533521972174e-9 r
E                          , E                          ,
                         2                              2
 - 3.266426827094104e-9 r    - 7.277118895509326e-9 r
E                          , E                          ,
                         2                              2
 - 1.475029730376073e-8 r    - 2.775001355397757e-8 r
E                          , E                          ,
                         2                              2
 - 4.915051879738995e-8 r    - 8.28246445511412e-8 r
E                          , E                          ,
                        2                               2
 - 1.33851622723744e-7 r    - 2.087442283111582e-7 r
E                         , E                           ,
                         2                               2
 - 3.156951172237287e-7 r    - 4.64842220857938e-7 r
E                          , E                          ,
                         2                               2
 - 6.685530482422835e-7 r    - 9.417277358666075e-7 r
E                          , E                          ,
                        2                               2
 - 1.30212067465563e-6 r    - 1.770680532972444e-6 r
E                         , E                           ,
                         2                               2
 - 2.371908484044149e-6 r    - 3.134234435790633e-6 r
E                          , E                          ,
                         2                               2
```

```
 - 4.090411050716832e-6 r    - 5.277925333300395e-6 r
E                         , E                         ,
                         2                           2
 - 6.739427552177103e-6 r    - 8.523177254399114e-6 r
E                         , E                         ,
                         2                           2
 - 1.068350611911921e-5 r    - 1.328129738824626e-5 r
E                         , E                         ,
                         2                           2
 - 1.638448160192355e-5 r    - 2.006854835710647e-5 r
E                         , E                         ,
                        2                           2
 - 2.44170737980647e-5 r    - 2.952226353832265e-5 r
E                        , E                         ,
                         2                           2
 - 3.548551070434468e-5 r    - 4.241796878224187e-5 r
E                         , E                         ,
                         2                           2
 - 5.044113893984222e-5 r    - 5.968747148772726e-5 r
E                         , E                         ,
                         2                           2
 - 7.030098113418114e-5 r    - 8.243787568058321e-5 r
E                         , E                         ,
                         2                           2
 - 9.626719779540763e-5 r    - 1.11971479496896e-4 r
E                         , E                         ,
                         2                           2
 - 1.297474089664522e-4 r    - 1.498065093069853e-4 r
E                         , E                         ,
                         2                           2
 - 1.723758288528179e-4 r    - 1.976986426302469e-4 r
E                         , E                         ,
                         2                           2
 - 2.260351645605837e-4 r    - 2.576632699903951e-4 r
E                         , E                         ,
```

```
                        2                          2
 - 2.928792281266932e-4 r    - 3.319984439480964e-4 r
E                        , E                        ,
                        2                          2
 - 3.753562091564763e-4 r    - 4.233084617271431e-4 r
E                        , E                        ,
                        2                         2
 - 4.762325536095718e-4 r    - 5.34528026124617e-4 r
E                        , E                        ,
                        2                          2
 - 5.986173925984417e-4 r    - 6.689469277678383e-4 r
E                        , E                        ,
                        2                          2
 - 7.459874634862211e-4 r    - 8.302351902545073e-4 r
E                        , E                        ,
                        2                          2
 - 9.222124640960191e-4 r    - 0.001022468618290102 r
E                        , E                        ,
                        2                          2
 - 0.001131580779474263 r    - 0.001250154687620788 r
E                        , E                        ,
                        2                          2
 - 0.001378825519389357 r    - 0.001518258714353595 r
E                        , E                        ,
                        2                          2
 - 0.001669150803595751 r    - 0.001832230240160423 r
E                        , E                        ,
                        2                          2
 - 0.002008258230854871 r    - 0.002198029568880921 r
E                        , E                        ,
                        2                          2
 - 0.002402373466780307 r    - 0.002622154389173151 r
E                        , E                        ,
                        2                          2
 - 0.002858272884767075 r    - 0.003111666417112067 r
```

E , E ,
2 2
- 0.003383310193575043 r - 0.003674217992005929 r
E , E ,
2 2
- 0.003985442984566339 r - 0.004318078558190487 r
E , E ,
2 2
- 0.004673259131147316 r - 0.005052160965172387 r
E , E ,
2 2
- 0.005456002972637555 r - 0.005886047518226416 r
E , E ,
2 2
- 0.006343601214583815 r - 0.006830015711407966 r
E , E ,
2 2
- 0.007346688477454374 r - 0.007895063574921807 r
E , E ,
2 2
- 0.008476632425691433 r - 0.009092934568891969 r
E , E ,
2 2
- 0.009745558409264787 r - 0.01043614195580549 r
E , E ,
2 2
- 0.01116637355015972 r - 0.01193799258425414 r
E , E ,
2 2
- 0.01275279020664547 r - 0.01361261001707348 r
E , E ,
2 2
- 0.01451934874870728 r - 0.01547495693757671 r
E , E ,
2 2

```
 - 0.01648143957868493 r    - 0.01754085676830185 r
E                        , E                        ,
                        2                          2
 - 0.01865532433194167 r    - 0.01982701443753252 r
E                        , E                        ,
                        2                          2
 - 0.02105815619329058 r    - 0.02235103622981523 r
E                        , E                        ,
                        2
 - 0.02370799926592746 r
E                        ]
```

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

```
Maxima output too long!
Error in:
 $showev('integrate(f(x),x)) ...
                            ^
```

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

$$erf(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}}\, dx.$$

Kita tidak dapat menggunakan teorema Dasar kalkulus untuk menghitung integral tentu fungsi tersebut jika semua batasnya berhingga. Dalam hal ini dapat digunakan metode numerik (rumus kuadratur).

Misalkan kita akan menghitung:

$$\int_0^{\pi} e^{-x^2}\, dx$$

```
>x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>add):
```

```
Illegal function result in map.
Error in:
x=0:0.1:pi-0.1; plot2d(x,f(x+0.1),>bar); plot2d("f(x)",0,pi,>a ...
                                  ^
```

Integral tentu

$$\int_0^{\pi} e^{-x^2}\, dx$$

dapat dihampiri dengan jumlah luas persegi-persegi panjang di bawah kurva y=f(x) tersebut. Langkah-langkahnya adalah sebagai berikut.

```
>t &= makelist(a,a,0,pi-0.1,0.1); // t sebagai list untuk menyimpan nilai-nilai x
>fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nilai f(x)
```

```
Maxima output too long!
Error in:
fx &= makelist(f(t[i]+0.1),i,1,length(t)); // simpan nilai-nil ...
                                          ^
```

```
>// jangan menggunakan x sebagai list, kecuali Anda pakar Maxima!
```

Hasilnya adalah:

$$\int_0^{\pi} e^{-x^2}\, dx = 0.8362196102528469$$

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai f(x) untuk x = 0.1, 0.2, 0.3, ..., 3.2.

```
>0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik EMT
```

```
Illegal function result in map.
Error in:
0.1*sum(f(x+0.1)) // cek langsung dengan perhitungan numerik E ...
                 ^
```

Untuk mendapatkan nilai integral tentu yang mendekati nilai sebenarnya, lebar sub-intervalnya dapat diperkecil lagi, sehingga daerah di bawah kurva tertutup semuanya, misalnya dapat digunakan lebar subinterval 0.001. (Silakan dicoba!)

Meskipun Maxima tidak dapat menghitung integral tentu fungsi tersebut untuk batas-batas yang berhingga, namun integral tersebut dapat dihitung secara eksak jika batas-batasnya tak hingga. Ini adalah salah satu keajaiban di dalam matematika, yang terbatas tidak dapat dihitung secara eksak, namun yang tak hingga malah dapat dihitung secara eksak.

```
>$showev('integrate(f(x),x,0,inf))
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([1,E^-(2.7777500001498e-14*r^2),E^-(1.777706668053906e-12*r^2),E^-(2.02481
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('integrate(f(x),x,0,inf)) ...
                                   ^
```

Tunjukkan kebenaran hasil di atas!

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

```
>function f(x) &= x^x
```

```
        expt([0, 1.66665833335744e-7 r, 1.33330666692022e-6 r,
4.499797504338432e-6 r, 1.066581336583994e-5 r,
2.083072932167196e-5 r, 3.599352055540239e-5 r,
5.71526624672386e-5 r, 8.530603082730626e-5 r,
1.214508019889565e-4 r, 1.665833531718508e-4 r,
2.216991628251896e-4 r, 2.877927110806339e-4 r,
3.658573803051457e-4 r, 4.568853557635201e-4 r,
5.618675264007778e-4 r, 6.817933857540259e-4 r,
8.176509330039827e-4 r, 9.704265741758145e-4 r,
0.001141105023499428 r, 0.001330669204938795 r,
0.001540100153900437 r, 0.001770376919130678 r,
0.002022476464811601 r, 0.002297373572865413 r,
0.002596040745477063 r, 0.002919448107844891 r,
0.003268563311168871 r, 0.003644351435886262 r,
0.004047774895164447 r, 0.004479793338660443 r, 0.0049413635565565 r,
0.005433439383882244 r, 0.005956971605131645 r,
0.006512907859185624 r, 0.007102192544548636 r,
0.007725766724910044 r, 0.00838456803503801 r,
0.009079530587017326 r, 0.009811584876838586 r, 0.0105816576913495 r,
0.01139067201557714 r, 0.01223954694042984 r, 0.01312919757078923 r,
0.01406053493400045 r, 0.01503446588876983 r, 0.01605189303448024 r,
0.01711371462093175 r, 0.01822082445851714 r, 0.01937411182884202 r,
0.02057446139579705 r, 0.02182275311709253 r, 0.02311986215626333 r,
0.02446665879515308 r, 0.02586400834688696 r, 0.02731277106934082 r,
0.02881380207911666 r, 0.03036795126603076 r, 0.03197606320812652 r,
0.03363897708721 63 r, 0.03535752660496472 r, 0.03713253989951881 r,
0.03896483946269502 r, 0.0408552420577305 r, 0.04280455863760801 r,
0.04481359426396048 r, 0.04688314802656623 r, 0.04901401296344043 r,
0.05120697598153157 r, 0.05346281777803219 r, 0.05578231276230905 r,
0.05816622897846346 r, 0.06061532802852698 r, 0.0631303649963022 r,
0.06571208837185505 r, 0.06836123997666599 r, 0.07107855488944881 r,
0.07386476137264342 r, 0.07672058079958999 r, 0.07964672758239233 r,
0.08264390910047736 r, 0.0857128256298576 r, 0.08885417027310427 r,
0.09206862889003742 r, 0.09535688002914089 r, 0.0987195948597075 r,
```

0.1021574371047232 r, 0.1056710629744951 r, 0.1092611211010309 r,
0.1129282524731764 r, 0.1166730903725168 r, 0.1204962603100498 r,
0.1243983799636342 r, 0.1283800591162231 r, 0.1324418995948859 r,
0.1365844952106265 r, 0.140808431699002 r, 0.1451142866615502 r,
0.1495026295080298 r, 0.1539740213994798 r],
[0, 1.66665833335744e-7 r, 1.33330666692022e-6 r,
4.499797504338432e-6 r, 1.066581336583994e-5 r,
2.083072932167196e-5 r, 3.599352055540239e-5 r,
5.71526624672386e-5 r, 8.530603082730626e-5 r,
1.214508019889565e-4 r, 1.665833531718508e-4 r,
2.216991628251896e-4 r, 2.877927110806339e-4 r,
3.658573803051457e-4 r, 4.568853557635201e-4 r,
5.618675264007778e-4 r, 6.817933857540259e-4 r,
8.176509330039827e-4 r, 9.704265741758145e-4 r,
0.001141105023499428 r, 0.001330669204938795 r,
0.001540100153900437 r, 0.001770376919130678 r,
0.002022476464811601 r, 0.002297373572865413 r,
0.002596040745477063 r, 0.002919448107844891 r,
0.003268563311168871 r, 0.003644351435886262 r,
0.004047774895164447 r, 0.004479793338660443 r, 0.0049413635565565 r,
0.005433439383882244 r, 0.005956971605131645 r,
0.006512907859185624 r, 0.007102192544548636 r,
0.007725766724910044 r, 0.00838456803503801 r,
0.009079530587017326 r, 0.009811584876838586 r, 0.0105816576913495 r,
0.01139067201557714 r, 0.01223954694042984 r, 0.01312919757078923 r,
0.01406053493400045 r, 0.01503446588876983 r, 0.01605189303448024 r,
0.01711371462093175 r, 0.01822082445851714 r, 0.01937411182884202 r,
0.02057446139579705 r, 0.02182275311709253 r, 0.02311986215626333 r,
0.02446665879515308 r, 0.02586400834688696 r, 0.02731277106934082 r,
0.02881380207911666 r, 0.03036795126603076 r, 0.03197606320812652 r,
0.0336389770872163 r, 0.03535752660496472 r, 0.03713253989951881 r,
0.03896483946269502 r, 0.0408552420577305 r, 0.04280455863760801 r,
0.04481359426396048 r, 0.04688314802656623 r, 0.04901401296344043 r,
0.05120697598153157 r, 0.05346281777803219 r, 0.05578231276230905 r,
0.05816622897846346 r, 0.06061532802852698 r, 0.0631303649963022 r,

```
0.06571208837185505 r, 0.06836123997666599 r, 0.07107855488944881 r,
0.07386476137264342 r, 0.07672058079958999 r, 0.07964672758239233 r,
0.08264390910047736 r, 0.0857128256298576 r, 0.08885417027310427 r,
0.09206862889003742 r, 0.09535688002914089 r, 0.0987195948597075 r,
0.1021574371047232 r, 1056710629744951 r, 0.1092611211010309 r,
0.1129282524731764 r, 1166730903725168 r, 0.1204962603100498 r,
0.1243983799636342 r, 1283800591162231 r, 0.1324418995948859 r,
0.1365844952106265 r, 140808431699002 r, 0.1451142866615502 r,
0.1495026295080298 r, 0.1539740213994798 r])
```

```
>$showev('integrate(f(x),x,0,1))
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([0,1.66665833335744e-7*r,1.33330666692022e-6*r,4.499797504338432e-6*r,1.06
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('integrate(f(x),x,0,1)) ...
                                ^
```

```
>x=0:0.1:1-0.01; plot2d(x,f(x+0.01),>bar); plot2d("f(x)",0,1,>add):
```

```
Error : f(x) does not produce a real or column vector

Error generated by error() command
```

```
 %ploteval:
    error(f$|" does not produce a real or column vector");
adaptiveevalone:
    s=%ploteval(g$,t;args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    dw/n,dw/n^2,dw/n,auto;args());
```

Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

```
>t &= makelist(a,a,0,1-0.01,0.01);
>fx &= makelist(f(t[i]+0.01),i,1,length(t));
```

```
Maxima output too long!
Error in:
fx &= makelist(f(t[i]+0.01),i,1,length(t)); ...
                                            ^
```

$$\int_0^1 x^x \, dx = 0.7834935879025506$$

Apakah hasil tersebut cukup baik? perhatikan gambarnya.

```
>function f(x) &= sin(3*x^5+7)^2
```

```
                2        2                            5
            [sin (7), sin (7 + 3.857928241984252e-34 r ),
    2                            5
  sin (7 + 1.264071117369734e-29 r ),
    2                            5
  sin (7 + 5.534598323932662e-27 r ),
    2                            5
  sin (7 + 4.140865788349747e-25 r ),
    2                            5
  sin (7 + 1.176640067174869e-23 r ),
    2                            5
  sin (7 + 1.812353420702142e-22 r ),
    2                            5
  sin (7 + 1.829378577899295e-21 r ),
    2                            5
  sin (7 + 1.355251607920358e-20 r ),
    2                            5
  sin (7 + 7.927261589391743e-20 r ),
    2                            5
  sin (7 + 3.848391561406479e-19 r ),
    2                            5
  sin (7 + 1.606724886280612e-18 r ),
    2                            5
  sin (7 + 5.922706430403385e-18 r ),
    2                            5
  sin (7 + 1.966438444926739e-17 r ),
    2                            5
  sin (7 + 5.972517698531353e-17 r ),
```

```
   2                            5
sin (7 + 1.679928959568682e-16 r ),
   2                            5
sin (7 + 4.419622473786581e-16 r ),
   2                            5
sin (7 + 1.096379579300723e-15 r ),
   2                            5
sin (7 + 2.581871707303932e-15 r ),
   2                            5
sin (7 + 5.804293182204869e-15 r ),
   2                            5
sin (7 + 1.251617959779806e-14 r ),
   2                            5
sin (7 + 2.599357872032985e-14 r ),
   2                            5
sin (7 + 5.217349730801036e-14 r ),
   2                            5      2                           5
sin (7 + 1.015169677716456e-13 r ), sin (7 + 1.9199033201535e-13 r ),
   2                           5
sin (7 + 3.53735606461862e-13 r ),
   2                            5
sin (7 + 6.362459865941665e-13 r ),
   2                            5
sin (7 + 1.119195003037803e-12 r ),
   2                            5
sin (7 + 1.928512721307568e-12 r ),
   2                            5
sin (7 + 3.259890537530715e-12 r ),
   2                            5
sin (7 + 5.412665053214709e-12 r ),
   2                            5
sin (7 + 8.838026394825634e-12 r ),
   2                            5
sin (7 + 1.420677121368262e-11 r ),
   2                            5
```

```
sin (7 + 2.250343962315998e-11 r ),
   2                            5
sin (7 + 3.515571419086806e-11 r ),
   2                           5      2                           5
sin (7 + 5.42105065269614e-11 r ), sin (7 + 8.25713163932381e-11 r ),
   2                            5
sin (7 + 1.243153394347951e-10 r ),
   2                            5      2                           5
sin (7 + 1.851135607682476e-10 r ), sin (7 + 2.7278286125595e-10 r ),
   2                            5
sin (7 + 3.980061623597386e-10 r ),
   2                            5
sin (7 + 5.752650497791658e-10 r ),
   2                            5
sin (7 + 8.240393785975726e-10 r ),
   2                           5      2                           5
sin (7 + 1.170340336056673e-9 r ), sin (7 + 1.648657617393293e-9 r ),
   2                           5      2                           5
sin (7 + 2.304418085581079e-9 r ), sin (7 + 3.197072905534091e-9 r ),
   2                           5      2                          5
sin (7 + 4.403953076583591e-9 r ), sin (7 + 6.02505998720235e-9 r ),
   2                           5      2                           5
sin (7 + 8.188988583715194e-9 r ), sin (7 + 1.106021653139266e-8 r ),
   2                           5      2                           5
sin (7 + 1.484803395715922e-8 r ), sin (7 + 1.981743566080698e-8 r ),
   2                           5      2                           5
sin (7 + 2.630235178964736e-8 r ), sin (7 + 3.472165467790206e-8 r ),
   2                           5      2                           5
sin (7 + 4.559844971273112e-8 r ), sin (7 + 5.958323763108241e-8 r ),
   2                           5      2                           5
sin (7 + 7.748162557801914e-8 r ), sin (7 + 1.002873656491318e-7 r ),
   2                         5      2                          5
sin (7 + 1.2922161366027e-7 r ), sin (7 + 1.657794287864104e-7 r ),
   2                           5      2                           5
sin (7 + 2.117846778257662e-7 r ), sin (7 + 2.694546676059425e-7 r ),
```

```
   2                           5      2                           5
sin (7 + 3.414760069817971e-7 r ), sin (7 + 4.310933976044147e-7 r ),
   2                           5      2                           5
sin (7 + 5.422132718931149e-7 r ), sin (7 + 6.795244392485226e-7 r ),
   2                           5      2                           5
sin (7 + 8.486381694408711e-7 r ), sin (7 + 1.056250437340941e-6 r ),
   2                          5      2                          5
sin (7 + 1.31032937788846e-6 r ), sin (7 + 1.620331356684433e-6 r ),
   2                           5      2                           5
sin (7 + 1.997449452236024e-6 r ), sin (7 + 2.454898573173933e-6 r ),
   2                           5      2                           5
sin (7 + 3.008241900322126e-6 r ), sin (7 + 3.675763852060344e-6 r ),
   2                           5      2                           5
sin (7 + 4.478895324833735e-6 r ), sin (7 + 5.442697561899192e-6 r ),
   2                           5      2                           5
sin (7 + 6.596411655537407e-6 r ), sin (7 + 7.974081394203433e-6 r ),
   2                           5      2                           5
sin (7 + 9.615257929748426e-6 r ), sin (7 + 1.156579556434297e-5 r ),
   2                           5      2                           5
sin (7 + 1.387874884559973e-5 r ), sin (7 + 1.661538211526619e-5 r ),
   2                           5      2                           5
sin (7 + 1.984630368547185e-5 r ), sin (7 + 2.365273792070429e-5 r ),
   2                           5      2                           5
sin (7 + 2.812794968737494e-5 r ), sin (7 + 3.337883690003698e-5 r ),
   2                           5      2                          5
sin (7 + 3.952770824811069e-5 r ), sin (7 + 4.67142646335361e-5 r ),
   2                           5      2                           5
sin (7 + 5.509780439231647e-5 r ), sin (7 + 6.485967401573982e-5 r ),
   2                           5      2                           5
sin (7 + 7.620598783449127e-5 r ), sin (7 + 8.937064198526321e-5 r ),
   2                           5      2                           5
sin (7 + 1.046186499492426e-4 r ), sin (7 + 1.222498290394089e-4 r ),
   2                           5      2                           5
sin (7 + 1.426028694233221e-4 r ), sin (7 + 1.660598196044561e-4 r ),
   2                           5      2                           5
```

```
sin (7 + 1.930510247525017e-4 r ), sin (7 + 2.240605568757864e-4 r ),
   2                                5
sin (7 + 2.596321785713566e-4 r )]
```

```
>integrate(f,0,1)
```

```
Illegal function result in map.
 %evalexpression:
    if maps then return map(f$,x;args());
gauss:
    if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());
adaptivegauss:
    t1=gauss(f$,c,c+h;args(),=maps);
Try "trace errors" to inspect local variables after errors.
integrate:
    return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
```

```
>&showev('integrate(f(x),x,0,1))
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([sin(7)^2,sin(7+3.857928241984252e-34*r^5)^2,sin(7+1.264071117369734e-29*r
 -- an error. To debug this try: debugmode(true);

Error in:
&showev('integrate(f(x),x,0,1)) ...
                                ^
```

```
>&float(%)
```

```
>$showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak)
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([0,1.66665833335744e-7*r*E^-(1.66665833335744e-7*r),1.33330666692022e-6*r*
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('integrate(x*exp(-x),x,0,1)) // Integral tentu (eksak) ...
                                     ^
```

```
>plot2d("x^3-x",-0.1,1.1); plot2d("-x^2",>add);  ...
>b=solve("x^3-x+x^2",0.5); x=linspace(0,b,200); xi=flipx(x); ...
>plot2d(x|xi,x^3-x|-xi^2,>filled,style="|",fillcolor=1,>add): // Plot daerah antara 2 kurva
```

```
>a=solve("x^3-x+x^2",0), b=solve("x^3-x+x^2",1) // absis titik-titik potong kedua kurva
```

```
0
0.61803398875
```

```
>integrate("(-x^2)-(x^3-x)",a,b) // luas daerah yang diarsir
```

```
0.0758191713542
```

Hasil tersebut akan kita bandingkan dengan perhitungan secara analitik.

```
>a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik potong kedua kurva secara eksak
```

```
Maxima said:
solve: all variables must not be numbers.
 -- an error. To debug this try: debugmode(true);

Error in:
a &= solve((-x^2)-(x^3-x),x); $a // menentukan absis titik pot ...
                              ^
```

```
>$showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai integral secara eksak
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate([0,1.66665833335744e-7*r-2.7777500001498e-14*r^2-4.629560185733296e-21*r^3
 -- an error. To debug this try: debugmode(true);

Error in:
... showev('integrate(-x^2-x^3+x,x,0,(sqrt(5)-1)/2)) // Nilai inte ...
                                                     ^
```

```
>$float(%)
```

$$0.07581917135421049$$

Panjang Kurva

Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

$$\gamma(t) = (r(t)\cos(t), r(t)\sin(t))$$

dengan

$$r(t) = 1 + \frac{\sin(3t)}{2}, \quad 0 \leq t \leq 2\pi.$$

```
>t=linspace(0,2pi,1000); r=1+sin(3*t)/2; x=r*cos(t); y=r*sin(t); ...
>plot2d(x,y,>filled,fillcolor=red,style="/",r=1.5): // Kita gambar kurvanya terlebih dahulu
```

```
>function r(t) &= 1+sin(3*t)/2; $'r(t)=r(t)
```

$r\left([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.2200000\right.$

```
>function fx(t) &= r(t)*cos(t); $'fx(t)=fx(t)
```

$fx\left([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.2200000\right.$

```
>function fy(t) &= r(t)*sin(t); $'fy(t)=fy(t)
```

fy ([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.2200000

```
>function ds(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t)
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'ds(t)=ds(t ...
                                                      ^
```

```
>$integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva
```

```
Maxima said:
Too few arguments supplied to ds(fx,fy); found: errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
 $integrate(ds(x),x,0,2*pi) //panjang (keliling) kurva ...
                            ^
```

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara umerik dengan perintah EMT.

```
>integrate("ds(x)",0,2*pi)
```

```
Function ds not found.
Try list ... to find functions!
Error in expression: ds(x)
 %mapexpression1:
    return expr(x,args());
Error in map.
 %evalexpression:
    if maps then return %mapexpression1(x,f$;args());
gauss:
    if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());
adaptivegauss:
    t1=gauss(f$,c,c+h;args(),=maps);
Try "trace errors" to inspect local variables after errors.
integrate:
    return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
```

Spiral Logaritmik

$$x = e^{ax}\cos x,\ y = e^{ax}\sin x.$$

```
>a=0.1; plot2d("exp(a*x)*cos(x)","exp(a*x)*sin(x)",r=2,xmin=0,xmax=2*pi):
```

```
>&kill(a) // hapus expresi a
```

done

```
>function fx(t) &= exp(a*t)*cos(t); $'fx(t)=fx(t)
```

fx ([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.2200000

```
>function fy(t) &= exp(a*t)*sin(t); $'fy(t)=fy(t)
```

$fy([0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.2200000$

```
>function df(t) &= trigreduce(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t)
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
... e(radcan(sqrt(diff(fx(t),t)^2+diff(fy(t),t)^2))); $'df(t)=df(t ...
                                                      ^
```

```
>S &=integrate(df(t),t,0,2*%pi); $S // panjang kurva (spiral)
```

```
Maxima said:
defint: variable of integration cannot be a constant; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
S &=integrate(df(t),t,0,2*%pi); $S // panjang kurva (spiral) ...
                               ^
```

```
>S(a=0.1) // Panjang kurva untuk a=0.1
```

```
Function S not found.
Try list ... to find functions!
Error in:
S(a=0.1) // Panjang kurva untuk a=0.1 ...
        ^
```

Soal:

Tunjukkan bahwa keliling lingkaran dengan jari-jari r adalah K=2.pi.r.

Berikut adalah contoh menghitung panjang parabola.

```
>plot2d("x^2",xmin=-1,xmax=1):
```

```
>$showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1))
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('integrate(sqrt(1+diff(x^2,x)^2),x,-1,1)) ...
                                                  ^
```

```
>$float(%)
>x=-1:0.2:1; y=x^2; plot2d(x,y);  ...
>  plot2d(x,y,points=1,style="o#",add=1):
```

Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

```
>i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])^2+(y[i+1]-y[i])^2))
```

```
2.95191957027
```

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

$$x^3 + y^3 - 3xy = 0.$$

```
>z &= x^3+y^3-3*x*y; $z
```

$$\left[0, -2.499966666877159 \times 10^{-11}\, r^2 + 1.249968796661264 \times 10^{-13}\, r^3, -7.999573343744277 \times 10^{-10}\, r^2 + 7.99920240756305\right.$$

```
>plot2d(z,r=2,level=0,n=100):
```

```
Contour needs a real matrix!
datacontour:
    contour(Z,level);
fcontour:
    contourcolor,contourwidth,style,outline,frame);
Try "trace errors" to inspect local variables after errors.
plot2d:
    =style,=outline,=frame);
```

Kita tertarik pada kurva di kuadran pertama.

```
>plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="/"):
```

```
Contour needs a real matrix!
datacontour:
    contour(Z,level);
fcontour:
    contourcolor,contourwidth,style,outline,frame);
Try "trace errors" to inspect local variables after errors.
plot2d:
    =style,=outline,=frame);
```

Kita selesaikan persamaannya untuk x.

```
>$z with y=l*x, sol &= solve(%,x); $sol
```

$$\left[0, -2.499966666877159 \times 10^{-11}\, r^2 + 1.249968796661264 \times 10^{-13}\, r^3, -7.999573343744277 \times 10^{-10}\, r^2 + 7.9992024075630$$

```
Maxima said:
solve: all variables must not be numbers.
 -- an error. To debug this try: debugmode(true);

Error in:
 $z with y=l*x, sol &= solve(%,x); $sol ...
                                      ^
```

Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
>function f(l) &= rhs(sol[1]); $'f(l)=f(l)
```

$$f(l) = 0$$

Fungsi tersebut juga dapat digunaka untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai y=l*x, yakni x=f(l) dan y=l*f(l).

```
>plot2d(&f(x),&x*f(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):
```

```
Error : [0,0,0,
```

Elemen panjang kurva adalah:

$$ds = \sqrt{f'(l)^2 + (lf'(l) + f(l))^2}.$$

```
>function ds(l) &= ratsimp(sqrt(diff(f(l),l)^2+diff(l*f(l),l)^2)); $'ds(l)=ds(l)
```

$$ds\,(l) = 0$$

```
>$integrate(ds(l),l,0,1)
```

$$0$$

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral etrsebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
>2*integrate("ds(x)",0,1)
```

```
0
```

```
>2*romberg(&ds(x),0,1)// perintah Euler lain untuk menghitung nilai hampiran integral yang sama
```

```
0
```

Perhitungan di datas dapat dilakukan untuk sebarang fungsi x dan y dengan mendefinisikan fungsi EMT, misalnya kita beri nama panjangkurva. Fungsi ini selalu memanggil Maxima untuk menurunkan fungsi yang diberikan.

```
>function panjangkurva(fx,fy,a,b) ...
```

```
ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");
return romberg(ds,a,b);
endfunction
```

```
>panjangkurva("x","x^2",-1,1) // cek untuk menghitung panjang kurva parabola sebelumnya
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Try "trace errors" to inspect local variables after errors.
panjangkurva:
    ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");
```

Bandingkan dengan nilai eksak di atas.

```
>2*panjangkurva(mxm("f(x)"),mxm("x*f(x)"),0,1) // cek contoh terakhir, bandingkan hasilnya!
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Try "trace errors" to inspect local variables after errors.
panjangkurva:
    ds=mxm("sqrt(diff(@fx,x)^2+diff(@fy,x)^2)");
```

Kita hitung panjang spiral Archimides berikut ini dengan fungsi tersebut.

```
>plot2d("x*cos(x)","x*sin(x)",xmin=0,xmax=2*pi,square=1):
```

```
>panjangkurva("x*cos(x)","x*sin(x)",0,2*pi)
```

```
21.2562941482
```

Berikut kita definisikan fungsi yang sama namun dengan Maxima, untuk perhitungan eksak.

```
>&kill(ds,x,fx,fy)
```

```
done
```

```
>function ds(fx,fy) &&= sqrt(diff(fx,x)^2+diff(fy,x)^2)
```

```
                    2                2
sqrt(diff (fy, x) + diff (fx, x))
```

```
>sol &= ds(x*cos(x),x*sin(x)); $sol // Kita gunakan untuk menghitung panjang kurva terakhir di atas
```

$$\sqrt{(\cos x - x\,\sin x)^2 + (\sin x + x\,\cos x)^2}$$

```
>$sol | trigreduce | expand, $integrate(%,x,0,2*pi), %()
```

$$\frac{\operatorname{asinh}\,(2\,\pi) + 2\,\pi\,\sqrt{4\,\pi^2 + 1}}{2}$$

$$\frac{\operatorname{asinh}\,(2\,\pi) + 2\,\pi\,\sqrt{4\,\pi^2 + 1}}{2}$$

```
21.2562941482
```

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.

Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.

```
>plot2d("3*x^2-1","3*x^3-1",xmin=-1/sqrt(3),xmax=1/sqrt(3),square=1):
```

```
>sol &= radcan(ds(3*x^2-1,3*x^3-1)); $sol
```

$$3\,x\,\sqrt{9\,x^2+4}$$

```
>$showev('integrate(sol,x,0,1/sqrt(3))), $2*float(%) // panjang kurva di atas
```

$$6.0\int_{0.0}^{0.5773502691896258} x\sqrt{9.0\,x^2+4.0}\,dx = 2.337835372767141$$

$$6.0\int_{0.0}^{0.5773502691896258} x\sqrt{9.0\,x^2+4.0}\,dx = 2.337835372767141$$

Sikloid

Berikut kita akan menghitung panjang kurva lintasan (sikloid) suatu titik pada lingkaran yang berputar ke kanan pada permukaan datar. Misalkan jari-jari lingkaran tersebut adalah r. Posisi titik pusat lingkaran pada saat t adalah:

$$(rt, r).$$

Misalkan posisi titik pada lingkaran tersebut mula-mula (0,0) dan posisinya pada saat t adalah:

$$(r(t-\sin(t)), r(1-\cos(t))).$$

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika t=0, t=pi/2, t=r*pi.

```
>x &= r*(t-sin(t))
```

```
        [0, 1.66665833335744e-7 r, 1.33330666692022e-6 r,
4.499797504338432e-6 r, 1.066581336583994e-5 r,
2.083072932167196e-5 r, 3.599352055540239e-5 r,
5.71526624672386e-5 r, 8.530603082730626e-5 r,
1.214508019889565e-4 r, 1.665833531718508e-4 r,
2.216991628251896e-4 r, 2.877927110806339e-4 r,
3.658573803051457e-4 r, 4.568853557635201e-4 r,
5.618675264007778e-4 r, 6.817933857540259e-4 r,
8.176509330039827e-4 r, 9.704265741758145e-4 r,
0.001141105023499428 r, 0.001330669204938795 r,
0.001540100153900437 r, 0.001770376919130678 r,
0.002022476464811601 r, 0.002297373572865413 r,
0.002596040745477063 r, 0.002919448107844891 r,
0.003268563311168871 r, 0.003644351435886262 r,
0.004047774895164447 r, 0.004479793338660443 r, 0.0049413635565565 r,
0.005433439383882244 r, 0.005956971605131645 r,
0.006512907859185624 r, 0.007102192544548636 r,
0.007725766724910044 r, 0.00838456803503801 r,
0.009079530587017326 r, 0.009811584876838586 r, 0.0105816576913495 r,
0.01139067201557714 r, 0.01223954694042984 r, 0.01312919757078923 r,
0.01406053493400045 r, 0.01503446588876983 r, 0.01605189303448024 r,
0.01711371462093175 r, 0.01822082445851714 r, 0.01937411182884202 r,
0.02057446139579705 r, 0.02182275311709253 r, 0.02311986215626333 r,
0.02446665879515308 r, 0.02586400834688696 r, 0.02731277106934082 r,
0.02881380207911666 r, 0.03036795126603076 r, 0.03197606320812652 r,
0.0336389770872163 r, 0.03535752660496472 r, 0.03713253989951881 r,
0.03896483946269502 r, 0.0408552420577305 r, 0.04280455863760801 r,
0.04481359426396048 r, 0.04688314802656623 r, 0.04901401296344043 r,
```

```
0.05120697598153157 r, 0.05346281777803219 r, 0.05578231276230905 r,
0.05816622897846346 r, 0.06061532802852698 r, 0.0631303649963022 r,
0.06571208837185505 r, 0.06836123997666599 r, 0.07107855488944881 r,
0.07386476137264342 r, 0.07672058079958999 r, 0.07964672758239233 r,
0.08264390910047736 r, 0.0857128256298576 r, 0.08885417027310427 r,
0.09206862889003742 r, 0.09535688002914089 r, 0.0987195948597075 r,
0.1021574371047232 r, 0.1056710629744951 r, 0.1092611211010309 r,
0.1129282524731764 r, 0.1166730903725168 r, 0.1204962603100498 r,
0.1243983799636342 r, 0.1283800591162231 r, 0.1324418995948859 r,
0.1365844952106265 r, 0.140808431699002 r, 0.1451142866615502 r,
0.1495026295080298 r, 0.1539740213994798 r]
```

```
>y &= r*(1-cos(t))
```

```
        [0, 4.999958333473664e-5 r, 1.999933334222437e-4 r,
4.499662510124569e-4 r, 7.998933390220841e-4 r,
0.001249739605033717 r, 0.00179946006479581 r,
0.002448999746720415 r, 0.003198293697380561 r,
0.004047266988005727 r, 0.004995834721974179 r,
0.006043902043303184 r, 0.00719136414613375 r, 0.00843810628521191 r,
0.009784003787362772 r, 0.01122892206395776 r, 0.01277271662437307 r,
0.01441523309043924 r, 0.01615630721187855 r, 0.01799576488272969 r,
0.01993342215875837 r, 0.02196908527585173 r, 0.02410255066939448 r,
0.02633360499462523 r, 0.02866202514797045 r, 0.03108757828935527 r,
0.03361002186548678 r, 0.03622910363410947 r, 0.03894456168922911 r,
0.04175612448730281 r, 0.04466351087439402 r, 0.04766643011428662 r,
0.05076458191755917 r, 0.0539576564716131 r, 0.05724533447165381 r,
0.06062728715262111 r, 0.06410317632206519 r, 0.06767265439396564 r,
0.07133536442348987 r, 0.07509094014268702 r, 0.07893900599711501 r,
0.08287917718339499 r, 0.08691105968769186 r, 0.09103425032511492 r,
```

```
0.09524833678003664 r, 0.09955289764732322 r, 0.1039475024744748 r,
0.1084317118046711 r, 0.113005077220716 r, 0.1176671413898787 r,
0.1224174381096274 r, 0.1272554923542488 r, 0.1321808203223502 r,
0.1371929294852391 r, 0.1422913186361759 r, 0.1474754779404944 r,
0.152744888986584 r, 0.1580990248377314 r, 0.1635373500848132 r,
0.1690593208998367 r, 0.1746643850903219 r, 0.1803519821545206 r,
0.1861215433374662 r, 0.1919724916878484 r, 0.1979042421157076 r,
0.2039162014509444 r, 0.2100077685026351 r, 0.216178334119151 r,
0.2224272812490723 r, 0.2287539850028937 r, 0.2351578127155118 r,
0.2416381240094921 r, 0.2481942708591053 r, 0.2548255976551299 r,
0.2615314412704124 r, 0.2683111311261794 r, 0.2751639892590951 r,
0.2820893303890569 r, 0.2890864619877229 r, 0.2961546843477643 r,
0.3032932906528349 r, 0.3105015670482534 r, 0.3177787927123868 r,
0.3251242399287333 r, 0.3325371741586922 r, 0.3400168541150183 r,
0.3475625318359485 r, 0.3551734527599992 r, 0.3628488558014202 r,
0.3705879734263036 r, 0.3783900317293359 r, 0.3862542505111889 r,
0.3941798433565377 r, 0.4021660177127022 r, 0.4102119749689023 r,
0.418316910536117 r, 0.4264800139275439 r, 0.4347004688396462 r,
0.4429774532337832 r, 0.451310139418413 r]
```

Berikut kita gambar sikloid untuk r=1.

```
>ex &= x-sin(x); ey &= 1-cos(x); aspect(1);
>plot2d(ex,ey,xmin=0,xmax=4pi,square=1); ...
>  plot2d("2+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
>  plot2d([2,ex(2)],[1,ey(2)],color=red,>add); ...
>  plot2d(ex(2),ey(2),>points,>add,color=red); ...
>  plot2d("2pi+cos(x)","1+sin(x)",xmin=0,xmax=2pi,>add,color=blue); ...
>  plot2d([2pi,ex(2pi)],[1,ey(2pi)],color=red,>add);  ...
>  plot2d(ex(2pi),ey(2pi),>points,>add,color=red):
```

```
Error : [0,1.66665833335744e-7*r-sin(1.66665833335744e-7*r),1.33330666692022e-6*r-sin(1.3333066669

Error generated by error() command

adaptiveeval:
    error(f$|" does not produce a real or column vector");
Try "trace errors" to inspect local variables after errors.
plot2d:
    dw/n,dw/n^2,dw/n;args());
```

Berikut dihitung panjang lintasan untuk 1 putaran penuh. (Jangan salah menduga bahwa panjang lintasan 1 putaran penuh sama dengan keliling lingkaran!)

```
>ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds) // elemen panjang kurva sikloid
```

```
Maxima said:
diff: second argument must be a variable; found errexp1
 -- an error. To debug this try: debugmode(true);

Error in:
ds &= radcan(sqrt(diff(ex,x)^2+diff(ey,x)^2)); $ds=trigsimp(ds ...
                                              ^
```

```
>ds &= trigsimp(ds); $ds
>$showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid satu putaran penuh
```

```
Maxima said:
defint: variable of integration must be a simple or subscripted variable.
defint: found errexp1
#0: showev(f='integrate(ds,[0,1.66665833335744e-7*r,1.33330666692022e-6*r,4.499797504338432e-6*r,1
 -- an error. To debug this try: debugmode(true);

Error in:
 $showev('integrate(ds,x,0,2*pi)) // hitung panjang sikloid sat ...
                                 ^
```

```
>integrate(mxm("ds"),0,2*pi) // hitung secara numerik
```

```
Illegal function result in map.
 %evalexpression:
    if maps then return %mapexpression1(x,f$;args());
gauss:
    if maps then y=%evalexpression(f$,a+h-(h*xn)',maps;args());
adaptivegauss:
    t1=gauss(f$,c,c+h;args(),=maps);
Try "trace errors" to inspect local variables after errors.
integrate:
    return adaptivegauss(f$,a,b,eps*1000;args(),=maps);
```

```
>romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik
```

```
Wrong argument!

Cannot combine a symbolic expression here.
Did you want to create a symbolic expression?
```

```
Then start with &.

Try "trace errors" to inspect local variables after errors.
romberg:
    if cols(y)==1 then return y*(b-a); endif;
Error in:
romberg(mxm("ds"),0,2*pi) // cara lain hitung secara numerik ...
                         ^
```

Perhatikan, seperti terlihat pada gambar, panjang sikloid lebih besar daripada keliling lingkarannya, yakni:

$$2\pi.$$

Visualisasi dan Perhitungan Geometri dengan EMT

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan geometeri tersebut disimpan di dalam file program "geometry.e", sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

```
>load geometry
```

```
Numerical and symbolic geometry.
```

Fungsi-fungsi Geometri

Fungsi-fungsi untuk Menggambar Objek Geometri:

```
defaultd:=textheight()*1.5: nilai asli untuk parameter d
setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang
```

koordinat

```
setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas
```

sumbu-x dan y adalah -r sd r

```
plotPoint (P, "P"): menggambar titik P dan diberi label "P"
plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label
```

”AB” sejauh d

```
plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d
plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"
plotLabel (label, P, V, d): menuliskan label pada posisi P
```

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

```
turn(v, phi): memutar vektor v sejauh phi
turnLeft(v):   memutar vektor v ke kiri
turnRight(v):  memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang vektorv dan w.
lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh.
```

ax+by=c.

```
lineWithDirection(A,v): garis melalui A searah vektor v
getLineDirection(g): vektor arah (gradien) garis g
getNormal(g): vektor normal (tegak lurus) garis g
getPointOnLine(g):  titik pada garis g
perpendicular(A, g):  garis melalui A tegak lurus garis g
parallel (A, g):  garis melalui A sejajar garis g
lineIntersection(g, h):  titik potong garis g dan h
projectToLine(A, g):   proyeksi titik A pada garis g
distance(A, B):  jarak titik A dan B
distanceSquared(A, B):  kuadrat jarak A dan B
quadrance(A, B): kuadrat jarak A dan B
areaTriangle(A, B, C):  luas segitiga ABC
computeAngle(A, B, C):   besar sudut <ABC
angleBisector(A, B, C): garis bagi sudut <ABC
circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r
getCircleCenter(c):  pusat lingkaran c
getCircleRadius(c):  jari-jari lingkaran c
circleThrough(A,B,C):  lingkaran melalui A, B, C
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan lingkran c
circleCircleIntersections (c1, c2):  titik potong lingkaran c1 dan
```

c2

```
planeThrough(A, B, C):  bidang melalui titik A, B, C
```

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

```
getLineEquation (g,x,y): persamaan garis g dinyatakan dalam x dan y
getHesseForm (g,x,y,A): bentuk Hesse garis g dinyatakan dalam x dan
```

y dengan titik A pada

```
sisi positif (kanan/atas) garis
quad(A,B): kuadrat jarak AB
spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni
```

sin(alpha)^2 dengan

```
alpha sudut yang menghadap sisi a.
crosslaw(a,b,c,sa): persamaan 3 quads dan 1 spread pada segitiga
```

dengan panjang sisi a, b, c.

```
triplespread(sa,sb,sc): persamaan 3 spread sa,sb,sc yang memebntuk
```

suatu segitiga

```
doublespread(sa): Spread sudut rangkap Spread 2*phi, dengan
```

sa=sin(phi)^2 spread a.

Contoh 1: Luas, Lingkaran Luar, Lingkaran Dalam Segitiga

Untuk menggambar objek-objek geometri, langkah pertama adalah menentukan rentang sumbu-sumbu koordinat. Semua objek geometri akan digambar pada satu bidang koordinat, sampai didefinisikan bidang koordinat yang baru.

```
>setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
```

Sekarang tetapkan tiga titik dan plotlah.

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
>B=[0,1]; plotPoint(B,"B");
>C=[2,2]; plotPoint(C,"C");
```

Lalu tiga segmen.

```
>plotSegment(A,B,"c"); // c=AB
>plotSegment(B,C,"a"); // a=BC
>plotSegment(A,C,"b"); // b=AC
```

Fungsi geometri mencakup fungsi untuk membuat garis dan lingkaran. Format untuk garis adalah [a,b,c], yang merepresentasikan garis dengan persamaan ax+by=c.

```
>lineThrough(B,C) // garis yang melalui B dan C
```

```
[-1,  2,  2]
```

Hitunglah garis tegak lurus melalui A pada BC.

```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan persimpangannya dengan BC.

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

Gambarkan itu.

```
>plotPoint(D,value=1); // koordinat D ditampilkan
>aspect(1); plotSegment(A,D): // tampilkan semua gambar hasil plot...()
```

Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2} AD.BC.$$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
```

```
1.5
```

Bandingkan dengan rumus determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langusng dengan fungsi
```

```
1.5
```

Cara lain menghitung luas segitigas ABC:

```
>distance(A,D)*distance(B,C)/2
```

```
1.5
```

Sudut di C.

```
>degprint(computeAngle(B,C,A))
```

```
36°52'11.63''
```

Sekarang lingkaran luar segitiga.

```
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
>R=getCircleRadius(c); // jari2 lingkaran luar
>O=getCircleCenter(c); // titik pusat lingkaran c
>plotPoint(O,"O"); // gambar titik "O"
>plotCircle(c,"Lingkaran luar segitiga ABC"):
```

Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>O, R
```

```
[1.16667,  1.16667]
1.17851130198
```

Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

```
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

```
[0.86038,  0.86038]
```

Tambahkan semuanya ke dalam plot.

```
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut
>plotPoint(P,"P"); // gambar titik potongnya
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

```
0.509653732104
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"): // gambar lingkaran dalam
```

2.5
2
1.5
1
0.5
0
-0.5
-0.5
0
0.5
1
1.5
2
2.5
C
a
D=(0.4,1.2)
O
b
B
r
p
P
AD
c
Lingkaran dalam segitiga ABC
Lingkaran luar segitiga ABC
A

Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

```
>setPlotRange(-1.5,3.5,-1.5,3.5);
>A=[-1,1]; plotPoint(A,"A"):
```

```
>B=[1,-1]; plotPoint(B,"B"):
```

```
>C=[3,3]; plotPoint(C,"C"):
```

2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. Merupakan segitiga apakah itu?

```
>plotSegment(A,B,"a")
>plotSegment(B,C,"b")
>plotSegment(A,C,"c")
>aspect(1):
```

3. Hitung luas segitiga tersebut.

4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

```
>l=angleBisector(A,C,B);
>g=angleBisector(C,A,B);
>P=lineIntersection(l,g)
```

```
[0.720759,  0.720759]
```

```
>color(5); plotLine(l); plotLine(g); color(1);
>plotPoint(P,"P");
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):
```

5. Gambar jari-jari lingkaran dalam.

```
>r=norm(P-projectToLine(P,lineThrough(A,B)))
```

```
1.01930746421
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):
```

6. Hitung luas lingkaran luar dan luas lingkaran dalam segitiga ABC. Adakah hubungan antara luas kedua lingkaran tersebut dengan luas segitiga ABC?

Contoh 2: Geometri Simbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometry.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Akan tetapi, kita sekarang dapat menggunakan perhitungan simbolik.

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi menyediakan perhitungan simbolis.

```
>c &= lineThrough(B,C) // c=BC
```

```
[- 1, 2, 2]
```

Kita dapat memperoleh persamaan garis dengan mudah.

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$2\,y - x = 2$$

$$\left[y = \frac{x}{2} + 1\right]$$

```
>$getLineEquation(lineThrough([x1,y1],[x2,y2]),x,y), $solve(%,y) // persamaan garis melalui(x1, y1)
```

$$x\ (y_1 - y_2) + (x_2 - x_1)\ y = x_1\ (y_1 - y_2) + (x_2 - x_1)\ y_1$$

$$\left[y = \frac{-(x_1 - x)\ y_2 - (x - x_2)\ y_1}{x_2 - x_1}\right]$$

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
```

$$(x_1 - 1)\ y - x\ y_1 = -y_1$$

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

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

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

```
 2  6
[-, -]
 5  5
```

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\left[\frac{2}{5},\frac{6}{5}\right]$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[\frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}}\right]$$

```
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

$$\frac{5}{3\sqrt{2}}$$

$$1.178511301977579$$

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 garis bagi s
```

$$\left[\frac{\sqrt{2}\sqrt{5}+2}{6}, \frac{\sqrt{2}\sqrt{5}+2}{6}\right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

```
[0.86038,  0.86038]
```

Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga dapat membuat garis berpotongan dengan lingkaran, dan lingkaran dengan lingkaran.

```
>A &:= [1,0]; c=circleWithCenter(A,4);
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l);
```

Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik perpotongan.

```
>{P1,P2,f}=lineCircleIntersections(l,c);
>P1, P2, f
```

```
[4.64575,  -1.64575]
[-0.645751,  3.64575]
2
```

```
>plotPoint(P1); plotPoint(P2):
```

Sama halnya di Maxima.

```
>c &= circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

```
[1, 0, 4]
```

```
>l &= lineThrough(B,C) // garis l melalui B dan C
```

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

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan garis l
```

$$\left[\left[\sqrt{7}+2, 1-\sqrt{7}\right], \left[2-\sqrt{7}, \sqrt{7}+1\right]\right]$$

Akan ditunjukkan bahwa sudut-sudut yang menghadap bsuusr yang sama adalah sama besar.

```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

```
>C=A+normalize([-4,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
```

```
69°17'42.68''
```

```
>insimg;
```

Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B.
2. Gambar lingkaran dengan pusat B melalui A.
3. Tarik garis melallui kedua titik potong kedua lingkaran tersebut. Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

```
>A=[2,2]; B=[-1,-2];
>c1=circleWithCenter(A,distance(A,B));
>c2=circleWithCenter(B,distance(A,B));
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(5); plotCircle(c1); plotCircle(c2);
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):
```

Selanjutnya, kita melakukan hal yang sama di Maxima dengan koordinat umum.

```
>A &= [a1,a2]; B &= [b1,b2];
>c1 &= circleWithCenter(A,distance(A,B));
>c2 &= circleWithCenter(B,distance(A,B));
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];
```

Persamaan untuk perpotongan cukup rumit. Namun, kita dapat menyederhanakannya, jika kita mencari nilai y.

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);
>$solve(g,y)
```

$$\left[y=\frac{-\left(2\,b_1-2\,a_1\right)\,x+{b_2}^2+{b_1}^2-{a_2}^2-{a_1}^2}{2\,b_2-2\,a_2}\right]$$

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sepenuhnya berbeda.

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[y=\frac{-\left(2\,b_1-2\,a_1\right)\,x+{b_2}^2+{b_1}^2-{a_2}^2-{a_1}^2}{2\,b_2-2\,a_2}\right]$$

```
>h &=getLineEquation(lineThrough(A,B),x,y);
>$solve(h,y)
```

$$\left[y=\frac{\left(b_2-a_2\right)\,x-a_1\,b_2+a_2\,b_1}{b_1-a_1}\right]$$

Perhatikan hasil kali gradien garis g dan h adalah:

$$\frac{-(b_1 - a_1)}{(b_2 - a_2)} \times \frac{(b_2 - a_2)}{(b_1 - a_1)} = -1.$$

Artinya kedua garis tegak lurus.

Contoh 3: Rumus Heron

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a, b dan c adalah:

$$L = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{dengan } s = (a+b+c)/2,$$

atau bisa ditulis dalam bentuk lain:

$$L = \frac{1}{4}\sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}$$

Untuk membuktikan hal ini kita misalkan C(0,0), B(a,0) dan A(x,y), b=AC, c=AB. Luas segitiga ABC adalah

$$L_{\triangle ABC} = \frac{1}{2}a \times y.$$

Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2 + y^2 = b^2, \quad (x-a)^2 + y^2 = c^2.$$

```
>setPlotRange(-1,10,-1,8); plotPoint([0,0], "C(0,0)"); plotPoint([5.5,0], "B(a,0)");  ...
> plotPoint([7.5,6], "A(x,y)");
>plotSegment([0,0],[5.5,0], "a",25); plotSegment([5.5,0],[7.5,6],"c",15);  ...
>plotSegment([0,0],[7.5,6],"b",25);
>plotSegment([7.5,6],[7.5,0],"t=y",25):
```

```
>&assume(a>0); sol &= solve([x^2+y^2=b^2,(x-a)^2+y^2=c^2],[x,y])
```

[]

Ekstrak solusi y.

```
>ysol &= y with sol[2][2]: $'y=sqrt(factor(ysol^2))
```

```
y = mabs(ysol)
```

Kita mendapatkan rumus Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H\left(a,b,[1,0,4]\right)=\frac{a\left|ysol\right|}{2}$$

```
>$'Luas=H(2,5,6) // luas segitiga dengan panjang sisi-sisi 2, 5, 6
```

$$Luas=\left|ysol\right|$$

Tentu saja, setiap segitiga siku-siku adalah kasus yang terkenal.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

```
Wrong argument!

Cannot combine a symbolic expression here.
Did you want to create a symbolic expression?
Then start with &.

This function can only use real or complex values!
Error in abs
Try "trace errors" to inspect local variables after errors.
H:
    useglobal; return a*abs(ysol)/2
Error in:
H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...
        ^
```

Dan jelas pula, bahwa ini adalah segitiga dengan luas maksimal dan dua sisinya 3 dan 4.

```
>aspect (1.5); plot2d(&H(3,4,x),1,7): // Kurva luas segitiga sengan panjang sisi 3, 4, x (1<= x <=7)
```

```
Wrong argument!

Cannot combine a symbolic expression here.
Did you want to create a symbolic expression?
Then start with &.
```

```
This function can only use real or complex values!
Error in abs
Error in expression: 3*abs(ysol)/2
 %ploteval:
    y0=f$(x[1],args());
adaptiveevalone:
    s=%ploteval(g$,t;args());
Try "trace errors" to inspect local variables after errors.
plot2d:
    dw/n,dw/n^2,dw/n,auto;args());
```

Kasus umum juga berfungsi.

```
>$solve(diff(H(a,b,c)^2,c)=0,c)
```

```
Maxima said:
diff: second argument must be a variable; found [1,0,4]
 -- an error. To debug this try: debugmode(true);

Error in:
$solve(diff(H(a,b,c)^2,c)=0,c) ...
                               ^
```

Sekarang mari kita cari himpunan semua titik di mana b+c=d untuk suatu konstanta d. Diketahui bahwa ini adalah elips.

```
>s1 &= subst(d-c,b,sol[2]); $s1
```

```
Maxima said:
part: invalid index of list or matrix.
 -- an error. To debug this try: debugmode(true);

Error in:
s1 &= subst(d-c,b,sol[2]); $s1 ...
                           ^
```

Dan buat fungsi ini.

```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); $fy(a,c,d)
```

$$0$$

$$0$$

Sekarang kita dapat menggambar himpunannya. Sisi b bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan elips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```

Kita dapat memeriksa persamaan umum untuk elips ini, yaitu:

$$\frac{(x-x_m)^2}{u^2}+\frac{(y-y_m)}{v^2}=1,$$

di mana (xm,ym) adalah pusat, dan u dan v adalah sumbu setengah.

```
>$ratsimp((fx(a,c,d)-a/2)^2/u^2+fy(a,c,d)^2/v^2 with [u=d/2,v=sqrt(d^2-a^2)/2])
```

$$\frac{a^2}{d^2}$$

Kita melihat bahwa tinggi dan luas segitiga tersebut adalah maksimum untuk x=0. Jadi luas segitiga dengan a+b+c=d adalah maksimum, jika segitiga tersebut sama sisi. Kita ingin memperolehnya secara analitis.

```
>eqns &= [diff(H(a,b,d-(a+b))^2,a)=0,diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

$$\left[\frac{a\,ysol^2}{2}=0,0=0\right]$$

Kita memperoleh beberapa nilai minimum, yang dimiliki oleh segitiga dengan satu sisi 0, dan solusinya a=b=c=d/3.

```
>$solve(eqns,[a,b])
```

$$[[a = 0, b = \%r_1]]$$

Ada juga metode Lagrange, yang memaksimalkan H(a,b,c)^2 terhadap a+b+d=d.

```
>&solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la, ...
>   diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la])
```

```
Maxima said:
diff: second argument must be a variable; found [1,0,4]
 -- an error. To debug this try: debugmode(true);

Error in:
... la,    diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la]) ...
                                                      ^
```

Kita bisa membuat plot dari situasinya

Pertama-tama atur titik di Maxima.

```
>A &= at([x,y],sol[2]); $A
```

```
Maxima said:
part: invalid index of list or matrix.
 -- an error. To debug this try: debugmode(true);

Error in:
A &= at([x,y],sol[2]); $A ...
                       ^
```

```
>B &= [0,0]; $B, C &= [a,0]; $C
```

$$[0,0]$$

$$[a,0]$$

Kemudian atur rentang plot dan plot titik-titiknya.

```
>$setPlotRange(0,5,-2,3);
>a=4; b=3; c=2;
>$plotPoint(mxmeval("B"),"B"); plotPoint(mxmeval("C"),"C");
>$plotPoint(mxmeval("A"),"A"):
```

Gambarkan segmen-segmennya.

```
>plotSegment(mxmeval("A"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("C")); ...
>plotSegment(mxmeval("B"),mxmeval("A")):
```

```
Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
    return evaluate(mxm(s));
Error in:
plotSegment(mxmeval("A"),mxmeval("C")); plotSegment(mxmeval("B ...
                         ^
```

Hitunglah garis tegak lurus tengah di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan pusat kelilingnya.

```
>U &= lineIntersection(h,g);
```

Kita mendapatkan rumus untuk jari-jari lingkaran luar.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

$$\frac{\sqrt{{a_2}^2+{a_1}^2}\,\sqrt{{a_2}^2+{a_1}^2-2\,a\,a_1+a^2}}{2\,|a_2|}$$

Mari kita tambahkan ini ke dalam alur cerita.

```
>plotPoint(U()); ...
>plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
```

```
Variable a2 not found!
Use global variables or parameters for string evaluation.
Error in ^
Error in expression: [a/2,(a2^2+a1^2-a*a1)/(2*a2)]
Error in:
plotPoint(U()); plotCircle(circleWithCenter(mxmeval("U"),mxmev ...
             ^
```

Dengan menggunakan geometri, kita peroleh rumus sederhana

$$\frac{a}{\sin(\alpha)}=2r$$

untuk jari-jari. Kita dapat memeriksa apakah ini benar dengan Maxima. Maxima akan memfaktorkan ini hanya jika kita mengkuadratkannya.

```
>$c^2/sin(computeAngle(A,B,C))^2  | factor
```

$$\left[\frac{{a_2}^2+{a_1}^2}{{a_2}^2},0,\frac{16\left({a_2}^2+{a_1}^2\right)}{{a_2}^2}\right]$$

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari setiap segitiga yang tidak sama sisi. Garis ini merupakan garis pusat segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga tersebut, termasuk orthocenter, circumcenter, centroid, titik Exeter, dan pusat lingkaran sembilan titik pada segitiga tersebut.

Sebagai contoh, kami menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kami mendefinisikan sudut-sudut segitiga dalam Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolik.

```
>A::=[-1,-1]; B::=[2,0]; C::=[1,2];
```

Untuk memplot objek geometris, kita menyiapkan area plot, dan menambahkan titik-titik ke dalamnya. Semua plot objek geometris ditambahkan ke plot saat ini.

```
>setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```

Kita juga dapat menambahkan sisi-sisi segitiga.

```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,""):
```

Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

```
>$areaTriangle(A,B,C)
```

$$-\frac{7}{2}$$

Kita dapat menghitung koefisien sisi c.

```
>c &= lineThrough(A,B)
```

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

Dan dapatkan juga rumus untuk garis ini.

```
>$getLineEquation(c,x,y)
```

$$3\,y - x = -2$$

Untuk bentuk Hesse, kita perlu menentukan suatu titik, sehingga titik tersebut berada di sisi positif bentuk Hesse. Memasukkan titik akan menghasilkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$$\frac{3\,y-x+2}{\sqrt{10}}$$
$$\frac{7}{\sqrt{10}}$$

Sekarang kita hitung lingkaran luar ABC.

```
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

$$\left(y-\frac{5}{14}\right)^2+\left(x-\frac{3}{14}\right)^2=\frac{325}{98}$$

```
>O &= getCircleCenter(LL); $O
```

$$\left[\frac{3}{14},\frac{5}{14}\right]$$

Gambarkan lingkaran dan titik pusatnya. Cu dan U adalah simbol. Kita evaluasi ekspresi ini untuk Euler.

```
>plotCircle(LL()); plotPoint(O(),"O"):
```

Kita dapat menghitung perpotongan tinggi di ABC (orthocenter) secara numerik dengan perintah berikut.

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...
>  perpendicular(B,lineThrough(A,C))); $H
```

$$\left[\frac{11}{7},\frac{2}{7}\right]$$

Sekarang kita dapat menghitung garis Euler dari segitiga tersebut.

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$-\frac{19\,y}{14}-\frac{x}{14}=-\frac{1}{2}$$

Tambahkan ke plot kita.

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```

Pusat gravitasi seharusnya berada pada garis ini.

```
>M &= (A+B+C)/3; $getLineEquation(el,x,y) with [x=M[1],y=M[2]]
```

$$-\frac{1}{2}=-\frac{1}{2}$$

```
>plotPoint(M(),"M"): // titik berat
```

Teori ini memberi tahu kita MH=2*MO. Kita perlu menyederhanakannya dengan radcan untuk mencapainya.

```
>$distance(M,H)/distance(M,O)|radcan
```

$$2$$

Fungsinya termasuk fungsi untuk sudut juga.

```
>$computeAngle(A,C,B), degprint(%())
```

$$\arccos\left(\frac{4}{\sqrt{5}\,\sqrt{13}}\right)$$

```
60°15'18.43''
```

Persamaan untuk pusat lingkaran dalam tidak terlalu bagus.

```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $Q
```

$$\left[\frac{\left(2^{\frac{3}{2}}+1\right)\,\sqrt{5}\,\sqrt{13}-15\,\sqrt{2}+3}{14},\frac{\left(\sqrt{2}-3\right)\,\sqrt{5}\,\sqrt{13}+5\,2^{\frac{3}{2}}+5}{14}\right]$$

Mari kita hitung juga ekspresi untuk jari-jari lingkaran dalam.

```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{\left(-41\sqrt{2}-31\right)\sqrt{5}\sqrt{13}+115\sqrt{2}+614}}{7\sqrt{2}}$$

```
>LD &=  circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke dalam alur cerita.

```
>color(5); plotCircle(LD()):
```

Parabola

Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang mencapai sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3\,y-x+2}{\sqrt{10}}-\sqrt{(2-y)^2+(1-x)^2}=0$$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

```
>plot2d(p,level=0,add=1,contourcolor=6):
```

Ini seharusnya merupakan suatu fungsi, tetapi penyelesai default Maxima hanya dapat menemukan solusinya, jika kita mengkuadratkan persamaannya. Akibatnya, kita mendapatkan solusi palsu.

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

```
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26,
                      y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 26]
```

Solusi pertama adalah

$$y = -3\,x - \sqrt{70}\,\sqrt{9 - 2\,x} + 26$$

Dengan menambahkan solusi pertama ke dalam plot, terlihat bahwa itu memang jalur yang kita cari. Teori tersebut memberi tahu kita bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1):
```

```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan kurva di atas
```

$$g(x) = -3\,x - \sqrt{70}\,\sqrt{9 - 2\,x} + 26$$

```
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

$$2.135605779339061$$

```
>U &= projectToLine(T,lineThrough(A,B)); $U // proyeksi T pada garis AB
```

$$\left[\frac{80 - 3\sqrt{11}\sqrt{70}}{10}, \frac{20 - \sqrt{11}\sqrt{70}}{10}\right]$$

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jatak T ke AB
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

$$2.135605779339061$$

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

Contoh 5: Trigonometri Rasional

Hal ini terinspirasi dari ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti konsep klasik tentang jarak dan sudut dengan kuadran dan sebaran. Dengan menggunakan konsep-konsep ini, memang memungkinkan untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

Berikut ini, saya memperkenalkan konsep-konsep tersebut, dan memecahkan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keuntungan utama trigonometri rasional bahwa perhitungan dapat dilakukan hanya dengan kertas dan pensil. Anda diundang untuk memeriksa hasilnya tanpa komputer.

Intinya adalah bahwa perhitungan rasional simbolik sering kali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

```
>load geometry;
```

Untuk pengenalan pertama, kami menggunakan segitiga siku-siku dengan proporsi Mesir yang terkenal yaitu 3, 4, dan 5. Perintah berikut adalah perintah Euler untuk memplot geometri bidang yang terdapat dalam file Euler "geometry.e".

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg(30);
```

Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

di mana wa adalah sudut di A. Cara yang biasa untuk menghitung sudut ini adalah dengan mengambil kebalikan dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak secara perkiraan.

```
>wa := arcsin(3/5); degprint(wa)
```

```
36°52'11.63''
```

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadran, yang menggantikan jarak. Faktanya, itu hanyalah kuadrat jarak. Dalam persamaan berikut, a, b, dan c menunjukkan kuadran sisi-sisi.

Teorema Pythogoras menjadi a+b=c.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

```
25 = 25
```

Gagasan kedua trigonometri rasional adalah sebaran. Sebaran mengukur bukaan antara garis. Nilainya 0, jika garis sejajar, dan 1, jika garis persegi panjang. Nilainya adalah kuadrat sinus sudut antara dua garis.

Sebaran garis AB dan AC pada gambar di atas didefinisikan sebagai

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

di mana a dan c adalah kuadran dari setiap segitiga persegi panjang dengan satu sudut di A.

```
>sa &= a/c; $sa
```

$$\frac{9}{25}$$

Tentu saja, ini lebih mudah dihitung daripada sudut. Namun, Anda kehilangan sifat bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengubah nilai perkiraan untuk sudut wa menjadi sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

```
9/25
```

Hukum kosinus dari trgonometri klasik diterjemahkan ke dalam "hukum silang" berikut.

$$(c+b-a)^2 = 4bc\,(1-s_a)$$

Di sini a, b, dan c adalah kuadran sisi-sisi segitiga, dan sa adalah sebaran di sudut A. Sisi a, seperti biasa, berseberangan dengan sudut A.

Hukum-hukum ini diimplementasikan dalam berkas geometry.e yang kami muat ke Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

$$\left[\left(bb-aa+\frac{7}{6}\right)^2,\left(bb-aa+\frac{7}{6}\right)^2,\left(bb-aa+\frac{5}{3\sqrt{2}}\right)^2\right]=\left[\frac{14\,bb\,(1-saa)}{3},\frac{14\,bb\,(1-saa)}{3},\frac{5\,2^{\frac{3}{2}}\,bb\,(1-saa)}{3}\right]$$

Dalam kasus kami, kami mendapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024=1024$$

Mari kita gunakan hukum silang ini untuk menemukan sebaran di A. Untuk melakukannya, kita buat hukum silang untuk kuadran a, b, dan c, dan selesaikan untuk sebaran yang tidak diketahui sa.

Anda dapat melakukannya dengan mudah secara manual, tetapi saya menggunakan Maxima. Tentu saja, kita mendapatkan hasil yang sudah kita miliki.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

$$1024 = 1600\ (1 - x)$$

$$\left[x = \frac{9}{25}\right]$$

Kita sudah tahu ini. Definisi sebaran adalah kasus khusus dari hukum silang.

Kita juga dapat memecahkan ini untuk a,b,c umum. Hasilnya adalah rumus yang menghitung sebaran sudut segitiga yang diberikan kuadran ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[\left[\frac{168\ bb\ x + 36\ bb^2 + (-72\ aa - 84)\ bb + 36\ aa^2 - 84\ aa + 49}{36}, \frac{168\ bb\ x + 36\ bb^2 + (-72\ aa - 84)\ bb + 36\ aa^2 - 84\ aa + 4}{36}\right.\right.$$

Kita dapat membuat fungsi dari hasil tersebut. Fungsi tersebut telah didefinisikan dalam berkas geometry.e milik Euler.

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi-sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak mudah didapat jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat.

```
>degprint(arcsin(sqrt(6/7)))
```

```
67°47'32.44''
```

Contoh Lain

Sekarang, mari kita coba contoh yang lebih maju.

Kita tentukan tiga sudut segitiga sebagai berikut.

```
>A&:=[1,2]; B&:=[4,3]; C&:=[0,4]; ...
>setPlotRange(-1,5,1,7); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```

Dengan menggunakan Pythogoras, mudah untuk menghitung jarak antara dua titik. Pertama-tama saya menggunakan fungsi distance dari file Euler untuk geometri. Fungsi distance menggunakan geometri klasik.

```
>$distance(A,B)
```

$$\sqrt{10}$$

Euler juga memuat fungsi untuk kuadran antara dua titik.

Dalam contoh berikut, karena c+b bukan a, segitiga tersebut bukan persegi panjang.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

$$10$$

$$5$$

$$17$$

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan perkalian titik dua vektor. Hasilnya adalah beberapa perkiraan floating point.

$$A =< 1, 2 > \quad B =< 4, 3 >, \quad C =< 0, 4 >$$

$$\mathbf{a} = C - B =< -4, 1 >, \quad \mathbf{c} = A - B =< -3, -1 >, \quad \beta = \angle ABC$$

$$\mathbf{a}.\mathbf{c} = |\mathbf{a}|.|\mathbf{c}| \cos\beta$$

$$\cos\angle ABC = \cos\beta = \frac{\mathbf{a}.\mathbf{c}}{|\mathbf{a}|.|\mathbf{c}|} = \frac{12-1}{\sqrt{17}\sqrt{10}} = \frac{11}{\sqrt{17}\sqrt{10}}$$

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

$$\arccos\left(\frac{11}{\sqrt{10}\,\sqrt{17}}\right)$$

```
32.4711922908
```

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kita masukkan kuadran a, b, dan c ke dalam hukum silang dan selesaikan untuk x.

```
>$crosslaw(a,b,c,x), $solve(%,x), //(b+c-a)^=4b.c(1-x)
```

$$4 = 200\ (1 - x)$$

$$\left[x = \frac{49}{50}\right]$$

Yaitu, apa yang dilakukan fungsi sebaran yang didefinisikan dalam "geometry.e".

```
>sb &= spread(b,a,c); $sb
```

$$\frac{49}{170}$$

Maxima memperoleh hasil yang sama dengan menggunakan trigonometri biasa, jika kita memaksakannya. Maxima memang menyelesaikan suku sin(arccos(...)) menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{49}{170}$$

Setelah kita memiliki sebaran di B, kita dapat menghitung tinggi ha pada sisi a. Ingat bahwa

$$s_b = \frac{h_a}{c}$$

menurut definisi.

```
>ha &= c*sb; $ha
```

$$\frac{49}{17}$$

Gambar berikut ini dibuat dengan program geometri C.a.R., yang dapat menggambar kuadran dan sebaran.

gambar: (20) Rational_Geometry_CaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadrannya.

```
>$sqrt(ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang kita bisa menghitung luas segitiga. Jangan lupa, bahwa kita sedang membahas tentang kuadran!

```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{7}{2}$$

Rumus determinan yang biasa menghasilkan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

$$\frac{7}{2}$$

Rumus Heron

Sekarang, mari kita selesaikan masalah ini secara umum!

```
>&remvalue(a,b,c,sb,ha);
```

Pertama-tama kita hitung sebaran di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita hitung luas kuadrat (yang disebut "quadrea"?), faktorkan dengan Maxima, dan kita dapatkan rumus Heron yang terkenal.

Memang, ini sulit dilakukan dengan pensil dan kertas.

```
>$spread(b^2,c^2,a^2), $factor(%*c^2*a^2/4)
```

$$\frac{-c^4-\left(-2\,b^2-2\,a^2\right)\,c^2-b^4+2\,a^2\,b^2-a^4}{4\,a^2\,c^2}$$

$$\frac{\left(-c+b+a\right)\,\left(c-b+a\right)\,\left(c+b-a\right)\,\left(c+b+a\right)}{16}$$

Aturan Triple Spread

Kerugian spread adalah tidak lagi sekadar menambahkan sudut yang sama.
Namun, tiga spread segitiga memenuhi aturan "triple spread" berikut.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

$$(sc+sb+sa)^2 = 2\left(sc^2+sb^2+sa^2\right)+4\,sa\,sb\,sc$$

Aturan ini berlaku untuk tiga sudut mana pun yang jika dijumlahkan menjadi 180°.

$$\alpha+\beta+\gamma=\pi$$

Karena sebaran

$$\alpha, \pi-\alpha$$

sama, aturan sebaran rangkap tiga juga berlaku, jika

$$\alpha+\beta=\gamma$$

Karena sebaran sudut negatif sama, aturan sebaran rangkap tiga juga berlaku, jika

$$\alpha+\beta+\gamma=0$$

Misalnya, kita dapat menghitung sebaran sudut 60°. Yaitu 3/4. Persamaan tersebut memiliki solusi kedua, di mana semua sebaran adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

$$\left[x=\frac{3}{4},x=0\right]$$

Sebaran 90° jelas adalah 1. Jika dua sudut dijumlahkan menjadi 90°, sebarannya memecahkan persamaan sebaran rangkap tiga dengan a,b,1. Dengan perhitungan berikut kita memperoleh a+b=1.

```
>$triplespread(x,y,1), $solve(%,x)
```

$$(y+x+1)^2=2\left(y^2+x^2+1\right)+4\,x\,y$$

$$[x=1-y]$$

Karena sebaran 180°-t sama dengan sebaran t, rumus sebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau selisih dari dua sudut lainnya.

Jadi kita dapat menemukan sebaran sudut yang digandakan. Perhatikan bahwa ada dua solusi lagi. Kita buat ini menjadi fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs(%[1]))
```

$$\left[x = 4\,a - 4\,a^2, x = 0\right]$$

```
- 4 (a - 1) a
```

Garis Bagi Sudut

Kita sudah tahu situasinya seperti ini.

```
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...
>setPlotRange(-1,5,-1,5); ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>insimg;
```

Mari kita hitung panjang garis bagi sudut di A. Namun, kita ingin menyelesaikannya untuk a,b,c umum.

```
>&remvalue(a,b,c);
```

Jadi pertama-tama kita hitung sebaran sudut yang dibagi dua di A, menggunakan rumus sebaran rangkap tiga.

Masalah dengan rumus ini muncul lagi. Rumus ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut yang dibagi dua 180°-wa.

```
>$triplespread(x,x,a/(a+b)), $solve(%,x), sa2 &= rhs(%[1]); $sa2
```

$$\left(2\,x+\frac{a}{b+a}\right)^2=2\left(2\,x^2+\frac{a^2}{(b+a)^2}\right)+\frac{4\,a\,x^2}{b+a}$$

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

$$\frac{-\sqrt{b}\,\sqrt{b+a}+b+a}{2\,b+2\,a}$$

Mari kita periksa persegi panjang Mesir.

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

Kita dapat mencetak sudut dalam Euler, setelah mentransfer sebaran ke radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

```
18°26'5.82''
```

Titik P merupakan perpotongan garis bagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
```

```
[0,  1.33333]
```

```
>plotPoint(P,"P"); plotSegment(A,P):
```

Mari kita periksa sudut-sudut pada contoh spesifik kita.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

```
0.321750554397
0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.

Kita gunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

berlaku di sembarang segitiga. Kuadratkan, maka akan menghasilkan apa yang disebut "hukum sebaran"

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_b}$$

di mana a,b,c menunjukkan kuadran.

Karena CPA sebaran adalah 1-sa2, kita peroleh darinya bisa/1=b/(1-sa2) dan dapat menghitung bisa (kuadran garis bagi sudut).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

$$\frac{2\,b\,(b+a)}{\sqrt{b}\,\sqrt{b+a}+b+a}$$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])")), distance(A,P)
```

```
4.21637021356
4.21637021356
```

Kita juga dapat menghitung P menggunakan rumus spread.

```
>py&=factor(ratsimp(sa2*bisa)); $py
```

$$-\frac{b\left(\sqrt{b}\sqrt{b+a}-b-a\right)}{\sqrt{b}\sqrt{b+a}+b+a}$$

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

```
1.33333333333
```

Sudut Tali Busur

Perhatikan situasi berikut.

```
>setPlotRange(1.2); ...
>color(1); plotCircle(circleWithCenter([0,0],1)); ...
>A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>color(1); O:=[0,0];  plotPoint(O,"O"); ...
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
>insimg;
```

Kita dapat menggunakan Maxima untuk memecahkan rumus penyebaran rangkap tiga untuk sudut-sudut di pusat O untuk r. Dengan demikian, kita memperoleh rumus untuk jari-jari kuadrat pericircle dalam bentuk kuadran sisi-sisinya.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

```
>&remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); $rabc
```

$$-\frac{a\,b\,c}{c^2-2\,b\,c+a\,\left(-2\,c-2\,b\right)+b^2+a^2}$$

Kita dapat menjadikannya fungsi Euler.

```
>function periradius(a,b,c) &= rabc;
```

Mari kita periksa hasilnya untuk titik A, B, C.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Radiusnya memang 1.

```
>periradius(a,b,c)
```

```
1
```

Faktanya, sebaran CBA hanya bergantung pada b dan c. Ini adalah teorema sudut tali busur.

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Faktanya, sebarannya adalah b/(4r), dan kita melihat bahwa sudut tali busur b adalah setengah sudut pusat.

```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

$$0$$

Contoh 6: Jarak Minimal pada Bidang

Catatan awal

Fungsi yang, pada titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis datar yang cukup sederhana: lingkaran yang berpusat di A.

```
>&remvalue();
>A=[-1,-1];
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...
>title="If you see ellipses, please set your window square"):
```

dan grafiknya cukup sederhana: bagian atas kerucut:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```

Tentu saja minimum 0 dicapai di A.

Dua titik

Sekarang kita lihat fungsi MA+MB di mana A dan B adalah dua titik (tetap). Merupakan "fakta yang diketahui" bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali untuk minimum AB yang konstan pada segmen [AB]:

```
>B=[1,-1];
>function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2+(y-B[2])^2)
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```

Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```

Pembatasan pada garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

Sekarang semuanya menjadi kurang sederhana: Tidak banyak yang tahu bahwa MA+MB+MC mencapai nilai minimumnya di satu titik bidang, tetapi menentukannya tidaklah sesederhana itu:

1) Jika salah satu sudut segitiga ABC lebih dari 120° (misalkan di A), maka nilai minimumnya tercapai di titik ini (misalkan AB+AC).

Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4):
```

```
>insimg;
```

```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

2) Namun jika semua sudut segitiga ABC kurang dari 120°, maka nilai minimumnya berada di titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang sudut-sudut sisi ABC-nya sama (masing-masing sudutnya 120°):

```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):
```

```
>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

Merupakan aktivitas yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya mengetahui perangkat lunak yang ditulis dalam Java yang memiliki instruksi "garis kontur"...

Semua ini ditemukan oleh seorang hakim Prancis bernama Pierre de Fermat; ia menulis surat kepada para dilettan lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di pajak penghasilan. Jadi titik unik F sehingga FA+FB+FC minimal, disebut titik Fermat dari segitiga tersebut. Namun tampaknya beberapa tahun sebelumnya, Torriccelli dari Italia telah menemukan titik ini sebelum Fermat menemukannya! Bagaimanapun tradisinya adalah mencatat titik F ini...

Empat titik

Langkah berikutnya adalah menambahkan titik ke-4 D dan mencoba meminimalkan MA+MB+MC+MD; katakanlah Anda adalah operator TV kabel dan ingin mencari di bidang mana Anda harus meletakkan antena Anda sehingga Anda dapat menyalurkan sinyal ke empat desa dan menggunakan panjang kabel sesedikit mungkin!

```
>D=[1,1];
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
```

```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
>insimg;
```

Masih terdapat nilai minimum dan tidak tercapai di titik A, B, C, maupun D:

```
>function f(x):=d4(x[1],x[2])
>neldermin("f",[0.2,0.2])
```

```
[0.142858,  0.142857]
```

Tampaknya dalam kasus ini, koordinat titik optimal bersifat rasional atau mendekati rasional...

Sekarang ABCD adalah persegi, kita mengharapkan bahwa titik optimal akan menjadi pusat ABCD:

```
>C=[-1,1];
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1):
```

```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
>P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);
>insimg;
```

1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan pvengine.exe di jalur program.

Pertama, kita hitung jari-jari bola.

Jika Anda melihat gambar di bawah, Anda melihat bahwa kita memerlukan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file geometry.e milik Euler untuk ini.

```
>load geometry;
```

Pertama dua garis membentuk kerucut.

```
>g1 &= lineThrough([0,0],[1,a])
```

```
[- a, 1, 0]
```

```
>g2 &= lineThrough([0,0],[-1,a])
```

```
[- a, - 1, 0]
```

Lalu baris ketiga.

```
>g &= lineThrough([-1,0],[1,1])
```

```
[- 1, 2, 1]
```

Kita merencanakan segalanya sejauh ini.

```
>setPlotRange(-1,1,0,2);
>color(black); plotLine(g(),"")
>a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```

Sekarang kita ambil titik umum pada sumbu y.

```
>P &= [0,u]
```

```
[0, u]
```

Hitunglah jarak ke g1.

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

$$\sqrt{\left(\frac{a^2\,u}{a^2+1}-u\right)^2+\frac{a^2\,u^2}{\left(a^2+1\right)^2}}$$

Hitunglah jarak ke g.

```
>d &= distance(P,projectToLine(P,g)); $d
```

$$\sqrt{\left(\frac{u+2}{5}-u\right)^2+\frac{\left(2\,u-1\right)^2}{25}}$$

Dan temukan pusat kedua lingkaran, yang jaraknya sama.

```
>sol &= solve(d1^2=d^2,u); $sol
```

$$\left[u=\frac{-\sqrt{5}\,\sqrt{a^2+1}+2\,a^2+2}{4\,a^2-1},u=\frac{\sqrt{5}\,\sqrt{a^2+1}+2\,a^2+2}{4\,a^2-1}\right]$$

Ada dua solusi.

Kita mengevaluasi solusi simbolik, dan menemukan kedua pusat, dan kedua jarak.

```
>u := sol()
```

```
[0.333333,  1]
```

```
>dd := d()
```

```
[0.149071,  0.447214]
```

Gambarkan lingkaran-lingkaran tersebut ke dalam gambar.

```
>color(red);
>plotCircle(circleWithCenter([0,u[1]],dd[1]),"");
>plotCircle(circleWithCenter([0,u[2]],dd[2]),"");
>insimg;
```

Plot dengan Povray

Selanjutnya kita plot semuanya dengan Povray. Perhatikan bahwa Anda mengubah perintah apa pun dalam urutan perintah Povray berikut, dan menjalankan kembali semua perintah dengan Shift-Return.

Pertama-tama kita memuat fungsi povray.

```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Kami menyiapkan suasananya dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Berikutnya kita menulis kedua bola itu ke dalam file Povray.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

Dan kerucutnya, transparan.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

Kita buat bidang yang dibatasi pada kerucut.

```
>gp=g();
>pc=povcone([0,0,0],0,[0,0,a],1,"");
>vp=[gp[1],0,gp[2]]; dp=gp[3];
>writeln(povplane(vp,dp,povlook(blue,0.5),pc));
```

Sekarang kita buat dua titik pada lingkaran, di mana bola menyentuh kerucut.

```
>function turnz(v) := return [-v[2],v[1],v[3]]
>P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
>writeln(povpoint(P1,povlook(yellow)));
>P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
>writeln(povpoint(P2,povlook(yellow)));
```

Kemudian kita buat dua titik tempat bola-bola tersebut menyentuh bidang. Titik-titik ini adalah fokus elips.

```
>P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
>writeln(povpoint(P3,povlook(yellow)));
>P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
>writeln(povpoint(P4,povlook(yellow)));
```

Berikutnya kita hitung perpotongan P1P2 dengan bidang.

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
>writeln(povpoint(P5,povlook(yellow)));
```

Kita menghubungkan titik-titik dengan segmen garis.

```
>writeln(povsegment(P1,P2,povlook(yellow)));
>writeln(povsegment(P5,P3,povlook(yellow)));
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Sekarang kita buat pita abu-abu, di mana bola-bola menyentuh kerucut.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);
>pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
>writeln(povintersection([pcw,pc1],povlook(gray)));
>pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
>writeln(povintersection([pcw,pc2],povlook(gray)));
```

Mulai program Povray.

```
>povend();
```

Untuk mendapatkan Anaglyph ini, kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali nanti.

```
>function scene () ...
```

```
global a,u,dd,g,g1,defaultpointsize;
writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
gp=g();
pc=povcone([0,0,0],0,[0,0,a],1,"");
vp=[gp[1],0,gp[2]]; dp=gp[3];
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow)));
writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow)));
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc1],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction
```

Anda memerlukan kacamata merah/cyan untuk menghargai efek berikut.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Contoh 8: Geometri Bumi

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan sferis. Fungsi-fungsi tersebut terdapat dalam berkas "spherical.e" di folder contoh. Kami perlu memuat berkas tersebut terlebih dahulu.

```
>load "spherical.e";
```

Untuk memasukkan posisi geografis, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut ini adalah koordinat untuk Kampus FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569,  1.92657]
```

Anda dapat mencetak posisi ini dengan sposprint (cetak posisi bulat).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
S 7°46.467' E 110°23.050'
```

Mari kita tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,24.533)];
>sposprint(Solo), sposprint(Semarang),
```

```
S 7°34.333’ E 110°49.683’
S 6°59.050’ E 110°24.533’
```

Pertama, kita hitung vektor dari satu ke yang lain pada bola ideal. Vektor ini adalah [arah, jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang 7°.

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan jarak FMIPA-Solo
```

```
65°20’26.60’’
53.8945384608
```

Ini adalah perkiraan yang bagus. Rutin berikut menggunakan perkiraan yang lebih baik lagi. Pada jarak yang pendek, hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->" km" // perkiraan jarak FMIPA-Semarang
```

```
Commands must be separated by semicolon or comma!
Found:  // perkiraan jarak FMIPA-Semarang (character 32)
You can disable this in the Options menu.
Error in:
esdist(FMIPA,Semarang)->" km" // perkiraan jarak FMIPA-Semaran ...
                              ^
```

Ada fungsi untuk judul, yang memperhitungkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

```
65.34°
```

Sudut suatu segitiga melebihi 180° pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,Semarang,Solo); degprint(
```

```
180°0'10.77''
```

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, ini tidak akurat karena kesalahan pengurangan dalam asum-pi.

```
>(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FMIPA-Solo-Semarang
```

```
Commands must be separated by semicolon or comma!
Found:  // perkiraan luas segitiga FMIPA-Solo-Semarang (character 32)
You can disable this in the Options menu.
Error in:
(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FM ...
                                ^
```

Ada fungsi untuk ini, yang menggunakan lintang rata-rata segitiga untuk menghitung jari-jari bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi esarea()
```

```
2123.64310526 km^2
```

Kita juga dapat menambahkan vektor ke posisi. Vektor berisi arah dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kita menggunakan svector. Untuk menambahkan vektor ke posisi, kita menggunakan saddvector.

```
>v=svector(FMIPA,Solo); sposprint(saddvector(FMIPA,v)), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'
S 7°34.333' E 110°49.683'
```

These functions assume an ideal ball. The same on the earth.

```
>sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'
S 7°34.333' E 110°49.683'
```

Mari kita lihat contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];
>sposprint(Tugu), sposprint(Monas)
```

```
S 7°46.998' E 110°21.966'
S 6°10.500' E 106°48.717'
```

Menurut Google Earth, jaraknya adalah 429,66 km. Kami memperoleh perkiraan yang baik.

```
>esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Monas Jakarta
```

```
Commands must be separated by semicolon or comma!
Found:  // perkiraan jarak Tugu Jogja - Monas Jakarta (character 32)
You can disable this in the Options menu.
Error in:
esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Mona ...
                         ^
```

Judulnya sama dengan yang dihitung di Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

```
294°17'2.85''
```

Akan tetapi, kita tidak lagi memperoleh posisi target yang tepat, jika kita menambahkan arah dan jarak ke posisi awal. Hal ini terjadi karena kita tidak menghitung fungsi invers secara tepat, tetapi mengambil perkiraan radius bumi di sepanjang lintasan.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

```
S 6°10.500' E 106°48.717'
```

Namun, kesalahannya tidak besar.

```
>sposprint(Monas),
```

```
S 6°10.500' E 106°48.717'
```

Tentu saja, kita tidak dapat berlayar dengan arah yang sama dari satu tujuan ke tujuan lain, jika kita ingin mengambil jalur terpendek. Bayangkan, Anda terbang ke arah timur laut mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti arah yang konstan!

Perhitungan berikut menunjukkan bahwa kita jauh dari tujuan yang benar, jika kita menggunakan arah yang sama selama perjalanan kita.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 dikalikan sepersepuluh jaraknya, dengan memakai arah ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya sangat jauh.

```
>sposprint(p), skmprint(esdist(p,Monas))
```

```
  S 6°11.250' E 106°48.372'
       1.529km
```

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Lintasan terpendek dari P1 ke P2 bukanlah lingkaran lintang 30°, tetapi lintasan yang lebih pendek yang dimulai 10° lebih jauh ke utara di P1.

```
>sdegprint(esdir(P1,P2))
```

```
       79.69°
```

Namun, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kita menyesuaikannya pada 1/10 dari total jarak.

```
>p=P1;  dist=esdist(P1,P2); ...
>  loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
79.69°
81.67°
83.71°
85.78°
87.89°
90.00°
92.12°
94.22°
96.29°
98.33°
```

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti arah yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

```
0.203km
```

Kita memperoleh perkiraan yang baik, jika kita menyesuaikan arah setelah setiap 1/100 jarak total dari Tugu ke Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...
>  loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;
>skmprint(esdist(p,Monas))
```

```
       0.000km
```

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS sepanjang lingkaran besar menuju Monas dengan fungsi navigasi.

```
>load spherical; v=navigate(Tugu,Monas,10); ...
>  loop 1 to rows(v); sposprint(v[#]), end;
```

```
  S 7°46.998’ E 110°21.966’
  S 7°37.422’ E 110°0.573’
  S 7°27.829’ E 109°39.196’
  S 7°18.219’ E 109°17.834’
  S 7°8.592’ E 108°56.488’
  S 6°58.948’ E 108°35.157’
  S 6°49.289’ E 108°13.841’
  S 6°39.614’ E 107°52.539’
  S 6°29.924’ E 107°31.251’
  S 6°20.219’ E 107°9.977’
  S 6°10.500’ E 106°48.717’
```

Kita menulis suatu fungsi yang memplot bumi, dua posisi, dan posisi di antaranya.

```
>function testplot ...
```

```
useglobal;
plotearth;
plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");
plotposline(v);
endfunction
```

Sekarang rencanakan semuanya.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```

Atau gunakan plot3d untuk mendapatkan tampilan anaglifnya. Ini tampak sangat bagus dengan kaca mata merah/biru kehijauan.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```

N
90°
80°
0°
90
Tangu
Monas

Latihan

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah (360/n).
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan (360/n).
- Untuk n ganjil, pilih salah satu titik sudut adalah di atas.
- Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat.
- Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:
- Misalkan persamaan parabolanya y= ax^2+bx+c.
- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.

```
- Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung (sisinya-sisintya
```

merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).

```
- Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya
```

bertemu di satu titik.

```
- Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya.
- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali
```

panjang sisi-sisi yang berhadapan sama.

4. Gambarlah suatu ellips jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang jumlah jarak ke P dan ke Q selalu sama (konstan).

5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

```
function plot2d (xv, yv, btest, a, b, c, d, xmin, xmax, r, n,  ..
logplot, grid, frame, framecolor, square, color, thickness, style, ..
auto, add, user, delta, points, addpoints, pointstyle, bar, histogram,  ..
distribution, even, steps, own, adaptive, hue, level, contour,  ..
nc, filled, fillcolor, outline, title, xl, yl, maps, contourcolor, ..
contourwidth, ticks, margin, clipping, cx, cy, insimg, spectral,  ..
cgrid, vertical, smaller, dl, niveau, levels)
```

Multipurpose plot function for plots in the plane (2D plots). This function can do plots of functions of one variables, data plots, curves in the plane, bar plots, grids of complex numbers, and implicit plots of functions of two variables.

Parameters
x,y : equations, functions or data vectors
a,b,c,d : Plot area (default a=-2,b=2)
r : if r is set, then a=cx-r, b=cx+r, c=cy-r, d=cy+r

```
r can be a vector [rx,ry] or a vector [rx1,rx2,ry1,ry2].
```

xmin,xmax : range of the parameter for curves
auto : Determine y-range automatically (default)
square : if true, try to keep square x-y-ranges
n : number of intervals (default is adaptive)
grid : 0 = no grid and labels,

```
1 = axis only,
2 = normal grid (see below for the number of grid lines)
3 = inside axis
4 = no grid
5 = full grid including margin
6 = ticks at the frame
7 = axis only
8 = axis only, sub-ticks
```

frame : 0 = no frame
framecolor: color of the frame and the grid
margin : number between 0 and 0.4 for the margin around the plot
color : Color of curves. If this is a vector of colors,

```
it will be used for each row of a matrix of plots. In the case of
point plots, it should be a column vector. If a row vector or a
full matrix of colors is used for point plots, it will be used for
each data point.
```

thickness : line thickness for curves

```
This value can be smaller than 1 for very thin lines.
```

style : Plot style for lines, markers, and fills.

```
For points use
"[]", "<>", ".", "..", "...",
"*", "+", "|", "-", "o"
"[]#", "<>#", "o#" (filled shapes)
"[]w", "<>w", "ow" (non-transparent)
For lines use
"-", "--", "-.", ".", ".-.", "-.-", "->"
For filled polygons or bar plots use
"#", "#O", "O", "/", "\", "\/",
"+", "|", "-", "t"
```

points : plot single points instead of line segments
addpoints : if true, plots line segments and points
add : add the plot to the existing plot
user : enable user interaction for functions
delta : step size for user interaction
bar : bar plot (x are the interval bounds, y the interval values)
histogram : plots the frequencies of x in n subintervals
distribution=n : plots the distribution of x with n subintervals
even : use inter values for automatic histograms.
steps : plots the function as a step function (steps=1,2)
adaptive : use adaptive plots (n is the minimal number of steps)
level : plot level lines of an implicit function of two variables
outline : draws boundary of level ranges.
If the level value is a 2xn matrix, ranges of levels will be drawn
in the color using the given fill style. If outline is true, it
will be drawn in the contour color. Using this feature, regions of
f(x,y) between limits can be marked.
hue : add hue color to the level plot to indicate the function

```
value
```

contour : Use level plot with automatic levels
nc : number of automatic level lines
title : plot title (default "")
xl, yl : labels for the x- and y-axis
smaller : if >0, there will be more space to the left for labels.
vertical :

```
Turns vertical labels on or off. This changes the global variable
verticallabels locally for one plot. The value 1 sets only vertical
text, the value 2 uses vertical numerical labels on the y axis.
```

filled : fill the plot of a curve
fillcolor : fill color for bar and filled curves
outline : boundary for filled polygons
logplot : set logarithmic plots

```
1 = logplot in y,
2 = logplot in xy,
3 = logplot in x
```

own :

```
A string, which points to an own plot routine. With >user, you get
the same user interaction as in plot2d. The range will be set
before each call to your function.
```

maps : map expressions (0 is faster), functions are always mapped.
contourcolor : color of contour lines
contourwidth : width of contour lines
clipping : toggles the clipping (default is true)
title :

```
This can be used to describe the plot. The title will appear above
the plot. Moreover, a label for the x and y axis can be added with
xl="string" or yl="string". Other labels can be added with the
functions label() or labelbox(). The title can be a unicode
string or an image of a Latex formula.
```

cgrid :

```
Determines the number of grid lines for plots of complex grids.
Should be a divisor of the the matrix size minus 1 (number of
subintervals). cgrid can be a vector [cx,cy].
```

Overview

The function can plot

- expressions, call collections or functions of one variable,
- parametric curves,
- x data against y data,
- implicit functions,
- bar plots,
- complex grids,
- polygons.

If a function or expression for xv is given, plot2d() will compute values in the given range using the function or expression. The

expression must be an expression in the variable x. The range must be defined in the parameters a and b unless the default range should be used. The y-range will be computed automatically, unless c and d are specified, or a radius r, which yields the range r,r for x and y. For plots of functions, plot2d will use an adaptive evaluation of the function by default. To speed up the plot for complicated functions, switch this off with <adaptive, and optionally decrease the number of intervals n. Moreover, plot2d() will by default use mapping. I.e., it will compute the plot element for element. If your expression or your functions can handle a vector x, you can switch that off with <maps for faster evaluation.

Note that adaptive plots are always computed element for element. If functions or expressions for both xv and for yv are specified, plot2d() will compute a curve with the xv values as x-coordinates and the yv values as y-coordinates. In this case, a range should be defined for the parameter using xmin, xmax. Expressions contained