

Tugas Akhir Aplikom

Nandhita putri shalsabila

November 2024

1 pengenalan

EMT_{NandhitaPutriSNama} : *NandhitaPutriShalsabila*

NIM : 23030630075

Prodi/Kelas : Matematika B 2023

Pendahuluan dan Pengenalan Cara Kerja EMT

Selamat datang! Ini adalah pengantar pertama ke Euler Math Toolbox (disingkat EMT atau Euler). EMT adalah sistem terintegrasi yang merupakan perpaduan kernel numerik Euler dan program komputer aljabar Maxima.

* Bagian numerik, GUI, dan komunikasi dengan Maxima telah dikembangkan * oleh R. Grothmann, seorang profesor matematika di Universitas * Eichstätt, Jerman. Banyak algoritma numerik dan pustaka software open * source yang digunakan di dalamnya.

* Maxima adalah program open source yang matang dan sangat kaya untuk * perhitungan simbolik dan aritmatika tak terbatas. Software ini * dikelola oleh sekelompok pengembang di internet.

* Beberapa program lain (LaTeX, Povray, Tiny C Compiler, Python) dapat * digunakan di Euler untuk memungkinkan perhitungan yang lebih cepat * maupun tampilan atau grafik yang lebih baik.

Yang sedang Anda baca (jika dibaca di EMT) ini adalah berkas notebook di EMT. Notebook aslinya bawaan EMT (dalam bahasa Inggris) dapat dibuka melalui menu File, kemudian pilih "Open Tutorias and Example", lalu pilih file "00 First Steps.en". Perhatikan, file notebook EMT memiliki ekstensi ".en". Melalui notebook ini Anda akan belajar menggunakan software Euler untuk menyelesaikan berbagai masalah matematika.

Panduan ini ditulis dengan Euler dalam bentuk notebook Euler, yang berisi teks (deskriptif), baris-baris perintah, tampilan hasil perintah (numerik, ekspresi matematika, atau gambar/plot), dan gambar yang disisipkan dari file gambar.

Untuk menambah jendela EMT, Anda dapat menekan [F11]. EMT akan menampilkan jendela grafik di layar desktop Anda. Tekan [F11] lagi untuk kembali ke tata letak favorit Anda. Tata letak disimpan untuk sesi berikutnya.

Anda juga dapat menggunakan [Ctrl]+[G] untuk menyembunyikan jendela grafik. Selanjutnya Anda dapat beralih antara grafik dan teks dengan tombol [TAB].

Seperti yang Anda baca, notebook ini berisi tulisan (teks) berwarna hijau, yang dapat Anda edit dengan mengklik kanan teks atau tekan menu Edit -gt; Edit Comment atau tekan [F5], dan juga baris perintah EMT yang ditandai dengan "gt;" dan berwarna merah. Anda dapat menyisipkan baris perintah baru dengan cara menekan tiga tombol bersamaan: [Shift]+[Ctrl]+[Enter].

Komentar (Teks Uraian)

Komentar atau teks penjelasan dapat berisi beberapa "markup" dengan sintaks sebagai berikut.

- * Judul - ** Sub-Judul - latex: $F(x) = \int_a^x f(t) dt$ - mathjax : $\frac{x^2-1}{x-1} = x+1$ - maxima : $\text{integrate}(x^3, x) = \text{integrate}(x^3, x) + C$ - http : //www.euler-math-toolbox.de - See : http : //www.google.de|Google - image : hati.png - - -

Hasil sintaks-sintaks di atas (tanpa diawali tanda strip) adalah sebagai berikut.

Judul

Sub-Judul

$$F(x) = \int_a^x f(t) dt$$

$$\frac{x^2 - 1}{x - 1} = x + 1$$

$$\int x^3 dx = C + \frac{x^4}{4}$$

ja href="http://www.euler-math-toolbox.de"ihttp://www.euler-math-toolbox.dej/a,ja href="http://www.google.de"iGooglej/a, image: hati.png

Gambar diambil dari folder images di tempat file notebook berada dan tidak dapat dibaca dari Web. Untuk "See:", tautan (URL)web lokal dapat digunakan.

Paragraf terdiri atas satu baris panjang di editor. Pergantian baris akan memulai baris baru. Paragraf harus dipisahkan dengan baris kosong.

// baris perintah diawali dengan , komentar (keterangan) diawali dengan //

Baris Perintah

Mari kita tunjukkan cara menggunakan EMT sebagai kalkulator yang sangat canggih.

EMT berorientasi pada baris perintah. Anda dapat menuliskan satu atau lebih perintah dalam satu baris perintah. Setiap perintah harus diakhiri dengan koma atau titik koma.

- * Titik koma menyembunyikan output (hasil) dari perintah.
- * Sebuah koma mencetak hasilnya.
- * Setelah perintah terakhir, koma diasumsikan secara otomatis (boleh tidak * ditulis).

Dalam contoh berikut, kita mendefinisikan variabel r yang diberi nilai 1,25. Output dari definisi ini adalah nilai variabel. Tetapi karena tanda titik koma,

nilai ini tidak ditampilkan. Pada kedua perintah di belakangnya, hasil kedua perhitungan tersebut ditampilkan.

```
r=1.25; pi^2, 2*pi  
4.90873852123 7.85398163397
```

Latihan untuk Anda

* Sisipkan beberapa baris perintah baru

* Tulis perintah-perintah baru untuk melakukan suatu perhitungan yang *
Anda inginkan, boleh menggunakan variabel, boleh tanpa variabel.

Contoh soal

```
A=[5,7;6,4]
```

```
5 7 6 4
```

```
det (A)
```

```
-22
```

```
B=[6,8;6,3]
```

```
6 8 6 3
```

```
AB
```

```
30 56 36 12
```

```
C=A+B
```

```
11 15 12 7
```

Beberapa catatan yang harus Anda perhatikan tentang penulisan sintaks perintah EMT.

* Pastikan untuk menggunakan titik desimal, bukan koma desimal untuk * bilangan!

* Gunakan * untuk perkalian dan *^* untuk eksponen (pangkat).

* Seperti biasa, * dan / bersifat lebih kuat daripada + atau -.

* *^* mengikat lebih kuat dari *, sehingga *r^2* merupakan rumus * luas lingkaran.

* Jika perlu, Anda harus menambahkan tanda kurung, seperti pada $2^{(2^3)}$.

Perintah `r = 1.25` adalah menyimpan nilai ke variabel di EMT. Anda juga dapat menulis `r: = 1.25` jika mau. Anda dapat menggunakan spasi sesuka Anda.

Anda juga dapat mengakhiri baris perintah dengan komentar yang diawali dengan dua garis miring (`//`).

```
r := 1.25 // Komentar: Menggunakan := sebagai ganti =  
1.25
```

Argumen atau input untuk fungsi ditulis di dalam tanda kurung.

```
sin(45°), cos(pi), log(sqrt(E))
```

```
0.707106781187 -1 0.5
```

Seperti yang Anda lihat, fungsi trigonometri bekerja dengan radian, dan derajat dapat diubah dengan `°`. Jika keyboard Anda tidak memiliki karakter derajat tekan `[F7]`, atau gunakan fungsi `deg()` untuk mengonversi.

EMT menyediakan banyak sekali fungsi dan operator matematika. Hampir semua fungsi matematika sudah tersedia di EMT. Anda dapat melihat daftar lengkap fungsi-fungsi matematika di EMT pada berkas Referensi (klik menu `Help -gt; Reference`)

Untuk membuat rangkaian komputasi lebih mudah, Anda dapat merujuk ke hasil sebelumnya dengan "perhitungan dalam baris perintah yang sama.

$(\sqrt{5}+1)/2,$

1.61803398875 2

Latihan untuk Anda

* Buka berkas Reference dan baca fungsi-fungsi matematika yang * tersedia di EMT.

* Sisipkan beberapa baris perintah baru.

* Lakukan contoh-contoh perhitungan menggunakan fungsi-fungsi * matematika di EMT.

Contoh soal

$(\sqrt{3}+1)/2,$

1.36602540378 1.5

Satuan

EMT dapat mengubah unit satuan menjadi sistem standar internasional (SI). Tambahkan satuan di belakang angka untuk konversi sederhana.

1miles // 1 mil = 1609,344 m

1609.344

Beberapa satuan yang sudah dikenal di dalam EMT adalah sebagai berikut.

Semua unit diakhiri dengan tanda dolar (\$), *namun bolehtidakperluitulisdenganmengaktifkaneasyunits.*

kilometer:= 1000;

km:= *kilometer*;

cm:= 0.01;

mm:= 0.001;

minute:= 60;

min:= *minute*;

minutes:= *minute*;

hour:= 60 * *minute*;

h:= *hour*;

hours:= *hour*;

day:= 24 * *hour*;

days:= *day*;

d:= *day*;

year:= 365.2425 * *day*;

years:= *year*;

y:= *year*;

inch:= 0.0254;

in:= *inch*;

feet:= 12 * *inch*;

foot:= *feet*;

ft:= *feet*;

yard:= 3 * *feet*;

yards:= *yard*;

yd:= *yard*;

mile:= 1760 * *yard*;

miles:= *mile*;

kg:= 1;

sec:= 1;

```

ha:= 10000;
Ar:= 100;
Tagwerk:= 3408;
Acre:= 4046.8564224;
pt:= 0.376mm;

```

Untuk konversi ke dan antar unit, EMT menggunakan operator khusus, yakni -gt;.

```

4km - miles, 4inch - " mm"
2.48548476895 101.6 mm

```

Format Tampilan Nilai

Akurasi internal untuk nilai bilangan di EMT adalah standar IEEE, sekitar 16 digit desimal. Aslinya, EMT tidak mencetak semua digit suatu bilangan. Ini untuk menghemat tempat dan agar terlihat lebih baik. Untuk mengatrtampilan satu bilangan, operator berikut dapat digunakan.

```

pi
3.14159265359
longest pi
3.141592653589793
long pi
3.14159265359
short pi
3.1416
shortest pi
3.1
fraction pi
312689/99532
short 12001.0310, longE, longestpi
1612.7 2.71828182846 3.141592653589793

```

Format aslinya untuk menampilkan nilai menggunakan sekitar 10 digit. Format tampilan nilai dapat diatur secara global atau hanya untuk satu nilai.

Anda dapat mengganti format tampilan bilangan untuk semua perintah selanjutnya. Untuk mengembalikan ke format aslinya dapat digunakan perintah "defformat" atau "reset".

```

longestformat; pi, defformat; pi
3.141592653589793 3.14159265359

```

Kernel numerik EMT bekerja dengan bilangan titik mengambang (floating point) dalam presisi ganda IEEE (berbeda dengan bagian simbolik EMT). Hasil numerik dapat ditampilkan dalam bentuk pecahan.

```

1/7+1/4, fraction
0.392857142857 11/28

```

Perintah Multibaris

Perintah multi-baris membentang di beberapa baris yang terhubung dengan "..." di setiap akhir baris, kecuali baris terakhir. Untuk menghasilkan tanda pindah baris tersebut, gunakan tombol [Ctrl]+[Enter]. Ini akan menyambung perintah ke baris berikutnya dan menambahkan "..." di akhir baris se-

belumnya. Untuk menggabungkan suatu baris ke baris sebelumnya, gunakan [Ctrl]+[Backspace].

Contoh perintah multi-baris berikut dapat dijalankan setiap kali kursor berada di salah satu barisnya. Ini juga menunjukkan bahwa ... harus berada di akhir suatu baris meskipun baris tersebut memuat komentar.

```
a=4; b=15; c=2; // menyelesaikan  $ax^2 + bx + c = 0$ secaramanual... D =  
sqrt(b2/(a24) - c/a); ... - b/(2a) + D, ... - b/(2a) - D  
-0.138444501319 -3.61155549868
```

Menampilkan Daftar Variabel

Untuk menampilkan semua variabel yang sudah pernah Anda definisikan sebelumnya (dan dapat dilihat kembali nilainya), gunakan perintah "listvar".

```
listvar  
r 1.25 a 4 b 15 c 2 D 1.73655549868123
```

Perintah listvar hanya menampilkan variabel buatan pengguna. Dimungkinkan untuk menampilkan variabel lain, dengan menambahkan string termuat di dalam nama variabel yang diinginkan.

Perlu Anda perhatikan, bahwa EMT membedakan huruf besar dan huruf kecil. Jadi variabel "d" berbeda dengan variabel "D".

Contoh berikut ini menampilkan semua unit yang diakhiri dengan "m" dengan mencari semua variabel yang berisi "m".

```
listvar m  
km1000cm 0.01 mm0.001nm 1853.24496 gram0.001m 1 hquantum6.62606957e-  
34atm 101325
```

Untuk menghapus variabel tanpa harus memulai ulang EMT gunakan perintah "remvalue".

```
remvalue a,b,c,D  
D
```

Variable D not found! Error in: D ...

Menampilkan Panduan

Untuk mendapatkan panduan tentang penggunaan perintah atau fungsi di EMT, buka jendela panduan dengan menekan [F1] dan cari fungsinya. Anda juga dapat mengklik dua kali pada fungsi yang tertulis di baris perintah atau di teks untuk membuka jendela panduan.

Coba klik dua kali pada perintah "inrandom" berikut ini!

```
inrandom(10,6)  
[4, 2, 6, 2, 4, 2, 3, 2, 2, 6]
```

Di jendela panduan, Anda dapat mengklik kata apa saja untuk menemukan referensi atau fungsi.

Misalnya, coba klik kata "random" di jendela panduan. Kata tersebut boleh ada dalam teks atau di bagian "See:" pada panduan. Anda akan menemukan penjelasan fungsi "random", untuk menghasilkan bilangan acak berdistribusi uniform antara 0,0 dan 1,0. Dari panduan untuk "random" Anda dapat menampilkan panduan untuk fungsi "normal", dll.

```
random(10)  
[0.270906, 0.704419, 0.217693, 0.445363, 0.308411, 0.914541, 0.193585, 0.463387,  
0.095153, 0.595017]
```

```
normal(10)
[-0.495418, 1.6463, -0.390056, -1.98151, 3.44132, 0.308178, -0.733427, -0.526167,
1.10018, 0.108453]
```

Matriks dan Vektor

EMT merupakan suatu aplikasi matematika yang mengerti "bahasa matriks". Artinya, EMT menggunakan vektor dan matriks untuk perhitungan-perhitungan tingkat lanjut. Suatu vektor atau matriks dapat didefinisikan dengan tanda kurung siku. Elemen-elemennya dituliskan di dalam tanda kurung siku, antar elemen dalam satu baris dipisahkan oleh koma(,), antar baris dipisahkan oleh titik koma (;).

Vektor dan matriks dapat diberi nama seperti variabel biasa.

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

Karena EMT mengerti bahasa matriks, EMT memiliki kemampuan yang sangat canggih untuk melakukan perhitungan matematis untuk masalah-masalah aljabar linier, statistika, dan optimisasi.

Vektor juga dapat didefinisikan dengan menggunakan rentang nilai dengan interval tertentu menggunakan tanda titik dua (:), seperti contoh berikut ini.

```
c=1:5
[1, 2, 3, 4, 5]
w=0:0.1:1
[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
mean(w2)
0.35
```

Bilangan Kompleks

EMT juga dapat menggunakan bilangan kompleks. Tersedia banyak fungsi untuk bilangan kompleks di EMT. Bilangan imajiner

dituliskan dengan huruf I (huruf besar I), namun akan ditampilkan dengan huruf i (i kecil).

re(x) : bagian riil pada bilangan kompleks x. im(x) : bagian imajiner pada bilangan kompleks x. complex(x) : mengubah bilangan riil x menjadi bilangan kompleks. conj(x) : Konjugat untuk bilangan bilangan kompleks x. arg(x) : argumen (sudut dalam radian) bilangan kompleks x. real(x) : mengubah x menjadi bilangan riil.

Apabila bagian imajiner x terlalu besar, hasilnya akan menampilkan pesan kesalahan.

```
gt;sqrt(-1) // Error! gt;sqrt(complex(-1))
z=2+3I, re(z), im(z), conj(z), arg(z), deg(arg(z)), deg(arctan(3/2))
2+3i 2 3 2-3i 0.982793723247 56.309932474 56.309932474
deg(arg(I)) // 90°
90
sqrt(-1)
Floating point error! Error in sqrt Error in: sqrt(-1) ...
sqrt(complex(-1))
```

0+1i

EMT selalu menganggap semua hasil perhitungan berupa bilangan riil dan tidak akan secara otomatis mengubah ke bilangan kompleks.

Jadi akar kuadrat -1 akan menghasilkan kesalahan, tetapi akar kuadrat kompleks didefinisikan untuk bidang koordinat dengan cara seperti biasa. Untuk mengubah bilangan riil menjadi kompleks, Anda dapat menambahkan 0i atau menggunakan fungsi "complex".

```
complex(-1), sqrt(-1+0i) 0+1i
```

Matematika Simbolik

EMT dapat melakukan perhitungan matematika simbolis (eksak) dengan bantuan software Maxima. Software Maxima otomatis sudah terpasang di komputer Anda ketika Anda memasang EMT. Meskipun demikian, Anda dapat juga memasang software Maxima tersendiri (yang terpisah dengan instalasi Maxima di EMT).

Pengguna Maxima yang sudah mahir harus memperhatikan bahwa terdapat sedikit perbedaan dalam sintaks antara sintaks asli Maxima dan sintaks ekspresi simbolik di EMT.

Untuk melakukan perhitungan matematika simbolis di EMT, awali perintah Maxima dengan tanda "amp;". Setiap ekspresi yang dimulai dengan "amp;" adalah ekspresi simbolis dan dikerjakan oleh Maxima.

```
(a+b)2
2 (b + a)
expand((a+b)2), factor(x2 + 5x + 6)
2 2 b + 2 a b + a
(x + 2) (x + 3)
solve(ax2 + bx + c, x) // rumus abc
2 2 (- sqrt(b - 4 a c)) - b sqrt(b - 4 a c) - b [x = —————, x =
—————] 2 a 2 a
(a2 - b2)/(a + b), ratsimp(
2 2 a - b ———- b + a
a - b
10! // nilai faktorial (modus EMT)
3628800
10! // nilai faktorial (simbolik dengan Maxima)
3628800
```

Untuk menggunakan perintah Maxima secara langsung (seperti perintah pada layar Maxima) awali perintahnya dengan tanda "::" pada baris perintah EMT. Sintaks Maxima disesuaikan dengan sintaks EMT (disebut "modus kompatibilitas").

```
factor(1000) // mencari semua faktor 1000 (EMT)
[2, 2, 2, 5, 5, 5]
:: factor(1000) // faktorisasi prima 1000 (dengan Maxima)
3 3 2 5
:: factor(20!)
18 8 4 2 2 3 5 7 11 13 17 19
```


Jika Anda sudah mahir menggunakan Maxima, Anda dapat menggunakan sintaks asli perintah Maxima dengan menggunakan tanda ":" untuk mengawali setiap perintah Maxima di EMT. Perhatikan, harus ada spasi antara ":" dan perintahnya.

```
::: binomial(5,2); // nilai C(5,2)
10
::: binomial(m,4); // C(m,4)=m!/(4!(m-4)!)
(m - 3) (m - 2) (m - 1) m ----- 24
::: trigexpand(cos(x+y)); // rumus cos(x+y)=cos(x) cos(y)-sin(x)sin(y)
cos(x) cos(y) - sin(x) sin(y)
::: trigexpand(sin(x+y));
cos(x) sin(y) + sin(x) cos(y)
::: trigsimp(((1-sin(x)^2)cos(x))/cos(x)^2+tan(x)sec(x)^2)//menyederhanakanfungsi trigonometri
4 sin(x) + cos(x) ----- 3 cos(x)
```

Untuk menyimpan ekspresi simbolik ke dalam suatu variabel digunakan tanda "&=".

```
p1 = (x^3 + 1)/(x + 1)
3 x + 1 ----- x + 1
ratsimp(p1)
2 x - x + 1
```

Untuk mensubstitusikan suatu nilai ke dalam variabel dapat digunakan perintah "with".

```
p1 with x=3 // (3^3 + 1)/(3 + 1)
7
p1 with x=a+b, ratsimp(
3 (b + a) + 1 ----- b + a + 1
2 2 b + (2 a - 1) b + a - a + 1
diff(p1,x) //turunan p1 terhadap x
2 3 3 x x + 1 ----- x + 1 2 (x + 1)
integrate(p1,x) // integral p1 terhadap x
3 2 2 x - 3 x + 6 x ----- 6
```

Tampilan Matematika Simbolik dengan LaTeX

Anda dapat menampilkan hasil perhitungan simbolik secara lebih bagus menggunakan LaTeX. Untuk melakukan hal ini, tambahkan tanda dolar (\$) di depan tanda & pada setiap perintah.

Perhatikan, hal ini hanya dapat menghasilkan tampilan yang diinginkan apabila komputer Anda sudah terpasang software LaTeX.

```
(a + b)^2
expand((a + b)^2), factor(x^2 + 5x + 6)
solve(ax^2 + bx + c, x) // rumus abc
(a^2 - b^2)/(a + b), ratsimp(
```

Selamat Belajar dan Berlatih!

Baik, itulah sekilas pengantar penggunaan software EMT. Masih banyak kemampuan EMT yang akan Anda pelajari dan praktikkan.

Sebagai latihan untuk memperlancar penggunaan perintah-perintah EMT yang sudah dijelaskan di atas, silakan Anda lakukan hal-hal sebagai berikut.

* Carilah soal-soal matematika dari buku-buku Matematika.

* Tambahkan beberapa baris perintah EMT pada notebook ini.
 * Selesaikan soal-soal matematika tersebut dengan menggunakan EMT. *
 Pilih soal-soal yang sesuai dengan perintah-perintah yang sudah * dijelaskan
 dan dicontohkan di atas.

Contoh soal
 carilah nilai dari 7!
 7! // nilai faktorial (modus EMT)
 5040
 carilah nilai semua faktor dari 800
 factor(800) // mencari semua faktor 800 (EMT)
 [2, 2, 2, 2, 2, 5, 5]

2 aljabar

Aljabar_{NandhitaPutriShalsabila}Nama : NandhitaPutriShalsabila

NIM : 23030630075

Kelas : Matematika B 2023

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:

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

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

$$-\frac{42}{xy^4}$$

Menjabarkan:

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

showev('expand((6x⁶ - 3) + y⁵)(-7x² - y⁶ - 9)))

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

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*r2*h/3
```

```
16.7551608191
```

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

```
pi*2rh,
```

```
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 garis terhubung dengan "..." kedua garis 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 "

```
// This line will not be visible once the cursor is off the line
```

```
Garis yang dimulai dengan
```

```
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 $E^p i \pi^E$; then "Thoughtso!", 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.

Sintaks Dasar

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, $x^{(1/2)}$ jugadimungkinkan.

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; 1/2gt^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 + 4log(0.6)) + 1/7)
8.77908249441
```

Untuk menghitung ekspresi rumit seperti

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

Anda harus memasukkannya dalam bentuk baris.

```
((1/7 + 1/8 + 2) / (1/3 + 1/2))^2*pi
23.2671801626
```

Letakkan tanda kurung dengan hati-hati di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyorot ekspresi 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* *posisi* *fata* *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,lantai,langit-langit,bulat,abs,tanda tangan conj,re,im,arg,conj,nyata,kompleks beta,betai,gamma,gamma kompleks,ellrf,elf,ellrd,elle bitand,bitor,bitxor,bitnot

Beberapa perintah memiliki alias, mis. Untuk log.

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


5.555555555555554*16⁻¹

Contoh Soal

membuat deret bilangan dari 1 hingga 10, kemudian menghitung jumlah dari bilangan-bilangan tersebut.

(A) := 1:10;

sum(A)

55

1:10 menghasilkan deret bilangan dari 1 hingga 10. Jumlah dari deret ini adalah $1 + 2 + 3 + \dots + 10$, yang totalnya adalah 55.

membuat deret bilangan ganjil dari 1 hingga 100, kemudian menghitung jumlah dari bilangan-bilangan tersebut.

(B) := 1:2:100;

sum(B)

2500

1:2:100 menghasilkan deret bilangan ganjil dari 1 hingga 99. Jumlah bilangan-bilangan ini adalah total dari semua bilangan ganjil dalam rentang tersebut. menghitung jumlah semua bilangan prima yang kurang dari 50.

(A) := primes(50);

sum(A)

328

primes(50) menghasilkan deret bilangan prima yang kurang dari 50. Jumlah dari bilangan-bilangan prima ini adalah total dari bilangan-bilangan prima di bawah 50, yaitu 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, dan 47.

membuat deret bilangan dari 1 hingga 20, menghitung kuadrat dari setiap bilangan dalam deret tersebut, kemudian menghitung jumlah dari bilangan-bilangan kuadrat tersebut.

(A) := 1:20;

(B) := A.^2;

sum (B)

2870

1:20 menghasilkan deret bilangan dari 1 hingga 20. $A.^2$ menghasilkan kuadrat dari setiap bilangan dalam deret $2^2 + 3^2 + \dots + 20^2$, yang totalnya adalah 2870.

membuat deret bilangan dari 1 hingga 10, menghitung kubik dari setiap bilangan dalam deret tersebut, kemudian menghitung jumlah dari bilangan-bilangan kubik tersebut.

(A) := 1:10;

(B) := A.^3;

sum(B)

3025

1:10 menghasilkan deret bilangan dari 1 hingga 10. $A.^3$ menghasilkan kubik dari setiap bilangan dalam deret $2^3 + 3^3 + \dots + 10^3$, yang totalnya adalah 3025.

mengubah dan menampilkan hasil dari 1/5 dalam format bilangan heksadesimal.

printheX(1/5)

3.333333333333334*16⁻¹

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

Contoh Soal

```
A = "bismillah"
```

bismillah

karena variabel A berisi kalimat "bismillah" maka string yang dihasilkan juga berupa kalimat bismillah.

```
A := ["anak", "kecil", "naik", "sepeda"]
```

anak kecil naik sepeda

karena A merupakan list maka string menghasilkan ututan kebawah.

```
B:=[]; B—A—A
```

anak kecil naik sepeda anak kecil naik sepeda

kode di atas menginisialisasi list kosong B dan kemudian menggunakan operator — untuk menggabungkan elemen-elemen dalam variabel A dua kali, menghasilkan pengulangan elemen-elemen tersebut dalam output. Jika variabel A sudah diisi dengan elemen "anak", "kecil", "naik" "sepeda", maka outputnya menunjukkan urutan tersebut dua kali.

```
u"alpha; = " + 30 + u"deg;"
```

= 30°

merepresentasikan alpha = 30 derajat menggunakan string unicode.

```
u"196;cantik"
```

Äcantik

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

$$\frac{2_1E}{1} \quad E_13$$

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]

Contoh Soal
A=6—3:6:99

A[nonzeros(isprime(A))]

(1:70) 2, nonzeros(
[0, 0, 1,
1,
1, 1, 1, 1, 1, 1, 1] [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
62, 63, 64, 65, 66, 67, 68, 69, 70]

$$\begin{array}{c} 4_i E \quad E_i 5 \\ 0 \end{array}$$

4 < E & E < 5: Memeriksa apakah suatu nilai E berada di antara 4 dan 5. Jika E tidak ditentukan atau tidak sesuai, hasilnya adalah 0.

```
2;1, "pink";"hitam"
```

```
0 0
```

ini adalah perbandingan string yang memeriksa apakah string "pink" kurang dari string "hitam".

alam perbandingan string, EMT menggunakan urutan leksikografis atau urutan kamus berdasarkan karakter ASCII. Pada ASCII, huruf 'h' (ASCII 104) memiliki nilai lebih rendah daripada 'p' (ASCII 112).

```
(1:15) 3, nonzeros(
```

```
[0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1] [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

(1:15) > 3, nonzeros(lebih besar dari 3 dalam rentang 1:15 dan mengembangkan elemen-elemen tersebut yang tidak nol.

```
2;1, "nyemil";"makan"
```

```
0 0
```

ini adalah perbandingan string yang memeriksa apakah string "nyemil" kurang dari string "makan".

alam perbandingan string, EMT menggunakan urutan leksikografis atau urutan kamus berdasarkan karakter ASCII. Pada ASCII, huruf 'm' (ASCII 104) memiliki nilai lebih rendah daripada 'n' (ASCII 112).

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 secara lengkap 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.

```
printhe(pi)
```

```
3.243F6A8885A30*160
```

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

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

```
0.33333 3.14159 0.84147
```

Standarnya adalah format(12).

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

```
0.333333333333
```

Fungsi seperti "shortestformat", "shortformat", "longformat" 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 "format terpanjang" juga mengatur format skalar.

```

longestformat; pi
3.141592653589793

```

Sebagai referensi, berikut adalah daftar format keluaran terpenting.

format terpendek, format pendek, format panjang, format terpanjang format(panjang,digit) format bagus(panjang) frakformat(panjang) merusak format

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 deformat().

```

deformat; // default

```

Ada operator pendek yang hanya mencetak satu nilai. Operator "terpanjang" akan mencetak semua digit nomor yang valid.

```

longest pi2/2
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 "format panjang" 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

```

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

Contoh Soal

```

fraction 2+2/4+5/3+1/4
53/12

```

operasi ini menambahkan beberapa bilangan: 2, 2/4, 5/3, dan 1/4.

MT menghitung hasilnya sebagai:

```

2+0.5+1.6667+0.25=4.4167

```

amun, menggunakan fraction akan mengubah hasil tersebut menjadi bentuk pecahan yang paling sederhana: 53/12.

```

longest pi3/2
15.50313834014991

```

Menghitung nilai pi pangkat 3/2. EMT akan menghasilkan ~15.50313834014991

```

format(15,3); pi
3.142
format(15,3): Mengatur format tampilan hasil perhitungan di EMT. Angka
pertama (15) menentukan jumlah karakter total untuk setiap angka yang di-
tampilkan, dan angka kedua (3) menentukan jumlah angka desimal yang dita-
mpilkan.
defformat; // defaultprint(hasil)
defformat: Mengatur format kembali ke format default EMT untuk menampilkan
hasil.
Biasanya, perintah ini tidak melakukan apa-apa dalam konteks EMT, kecuali
ada variabel atau fungsi lain yang tidak ditampilkan di sini.
shortestformat; random(2,10)
0.53 0.5 0.17 0.26 0.87 0.54 0.49 0.6 0.66 0.97 0.19 0.94 0.073 0.99 0.01 0.36
0.52 0.43 0.17 0.18
shortestformat: Mengatur format tampilan EMT ke format paling pendek
(menampilkan angka dengan jumlah digit minimal yang diperlukan).
random(2,10): Menghasilkan matriks 2x10 dengan nilai acak antara 0 dan
1.
Matriks berukuran 2x10 dihasilkan dengan nilai acak
Ekspresi
String atau nama dapat digunakan untuk menyimpan ekspresi matematika,
yang dapat dievaluasi dengan EMT. Untuk ini, gunakan tanda kurung sete-
lah 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^2"; longest fx()
12.56637061435917
Parameter ditetapkan ke x, y, dan z dalam urutan itu. Parameter tambahan
dapat ditambahkan menggunakan parameter yang ditetapkan.
fx:="asin(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 ek-
spresi dalam fungsi dapat memberikan hasil yang sangat membingungkan bagi
pengguna yang memanggil fungsi tersebut.)
at:=4; function f(expr,x,at) := expr(x); ... f("atx^2", 3, 5)//computes 4 ×
3^2 not 5^2
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("atx^2", 3, 5)
45
Sebagai referensi, kami mencatat bahwa koleksi panggilan (dibahas di tem-
pat lain) dapat berisi ekspresi. Jadi kita bisa membuat contoh di atas sebagai
berikut.

```

```
at:=4; function f(expr,x) := expr(x); ... f("atx^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 = 5x;
function f(x) := 6x;
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(xx, 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",
@(a,b) a^2 + b^2
```

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 - ax; ... a = 1.2; fx(0.5)
-0.475
```

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

```
fx(0.5,a=1.1)
-0.425
```

Sebuah ekspresi tidak perlu bersifat simbolis. Hal ini diperlukan, jika ekspresi berisi fungsi, yang hanya diketahui di kernel numerik, bukan di Maxima.

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 secara mulus ke dalam Euler dengan amp;. Ekspresi apa pun yang dimulai dengan amp; adalah ekspresi simbolis. Itu dievaluasi dan dicetak oleh Maxima.

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

44!

2658271574788448768043625811014615890319638528000000000

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

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

44!/(34!10!)/nilaiC(44, 10)

2481256778

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

binomial(44, 10)/menghitungC(44, 10)menggunakanfungsibinomial()

2481256778

Untuk mempelajari lebih lanjut tentang fungsi tertentu, klik dua kali padanya. Misalnya, coba klik dua kali pada "amp;binomial" di baris perintah sebelumnya. Ini membuka dokumentasi Maxima yang disediakan oleh penulis program tersebut.

Anda akan mengetahui bahwa cara berikut juga bisa dilakukan.

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

binomial(x, 3)/C(x, 3)

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

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

binomial(x, 3)withx = 10/substitusix = 10keC(x, 3)

120

Dengan begitu Anda bisa menggunakan solusi suatu persamaan di persamaan lain.

Ekspresi simbolik dicetak oleh Maxima dalam bentuk 2D. Alasannya adalah adanya tanda simbolis khusus pada string tersebut.

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

Untuk mencetak ekspresi simbolik dengan LaTeX, gunakan *didepanamp; (atauAndadapatmenghilangkanan* jika Anda belum menginstal LaTeX.

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

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

Ekspresi simbolik diurai oleh Euler. Jika Anda memerlukan sintaksis kompleks dalam satu ekspresi, Anda dapat mengapit ekspresi tersebut di "...". Menggunakan lebih dari sekadar ekspresi sederhana bisa saja dilakukan, namun sangat tidak disarankan.

"v := 5; v²"

25

Untuk kelengkapannya, kami mencatat bahwa ekspresi simbolik dapat digunakan dalam program, namun perlu diapit dalam tanda kutip. Selain itu, akan jauh lebih efektif untuk memanggil Maxima pada waktu kompilasi jika memungkinkan.

expand((1 + x)⁴),*factor*(*diff*(

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

![images/Aljabar_{Nandhita}

Sekali lagi,

Untuk mempermudah kami menyimpan solusi ke variabel simbolik. Variabel simbolik didefinisikan dengan "amp;=".

fx = (x+1)/(x⁴ + 1);fx

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

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

factor(*diff*(fx, x))

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

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

::: av:gav²;

2 g

fx = x³*exp*(x),fx

3 x x E

$$x^3 e^x$$

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

```
(fx with x=5),
5 125 E
```

$$125 e^5$$

```
18551.64488782208
```

```
fx(5)
```

```
18551.6448878
```

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

Baris perintah berikut juga menunjukkan bahwa Maxima dapat mengevaluasi 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))
```

$$x (x^2 + 6x + 6) e^x$$

Untuk mendapatkan kode Lateks 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 simbolis, ini tidak berhasil, karena Maxima tidak mendukungnya. Sebagai gantinya, gunakan sintaks "dengan" (bentuk perintah at(...) yang lebih bagus dari Maxima).

```
fxwithx = 1/2
```

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

Penugasannya juga bisa bersifat simbolis.

```
fxwithx = 1 + t
```

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

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

```
solve(x^2 + x = 4, x)
```

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

Bandungkan dengan perintah numerik "solve" di Euler, yang memerlukan nilai awal, dan 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 membacakan tugas x= dst. Jika Anda tidak memerlukan hasil numerik untuk perhitungan lebih lanjut, Anda juga dapat membiarkan Maxima menemukan nilai numeriknya.

```
sol = solve(x^2 + 2x = 4, x); sol, sol(), float(sol)
```

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

[-3.2, 1.2]

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

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

```
solve(x^2 + x = 1, x), x2 = xwith
```

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

![images/Aljabar_Nandhita

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

```
sol = solve([x+y=3, x^2 + y^2 = 5], [x, y]); sol, xywithsol[1]
```

2

![images/Aljabar_Nandhita

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

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

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

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

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

```
factor(
```

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

Fungsi

Dalam EMT, fungsi adalah program yang didefinisikan dengan perintah "fungsi". Ini bisa berupa fungsi satu baris atau fungsi multibaris.

Fungsi satu baris dapat berupa numerik atau simbolik. Fungsi satu baris numerik didefinisikan oleh ":=".

```
function f(x) := xsqrt(x^2 + 1)
```

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

```
f(2)
4.472135955
```

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

```
f(0:0.1:1)
[0, 0.1, 0.2, 0.31, 0.43, 0.56, 0.7, 0.85, 1, 1.2, 1.4]
```

Fungsi dapat diplot. Daripada ekspresi, kita hanya perlu memberikan nama fungsinya.

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

```
solve("f",1,y=1)
0.786151377757
```

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

Anda masih dapat memanggil fungsi bawaan sebagai "...", *jika fungsitersebutadadiintiEuler*.

```
function overwrite sin (x) := _sin(x°) // redine sine in degrees
sin(45)
0.707106781187
```

Sebaiknya kita menghilangkan redefinisi dosa ini.

```
forget sin; sin(pi/4)
0.707106781187
```

Parameter Bawaan

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini akan menggunakan nilai default.

```
f(4)
16
```

Menyetelnya akan menimpa nilai default.

```
f(4,5)
80
```

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

f(4,a=1)

16

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

function f(x) := ax²

a=6; f(2)

24

Namun parameter yang ditetapkan mengesampingkan 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 "amp;=". Mereka didefinisikan di Euler dan Maxima, dan bekerja di kedua dunia. Ekspresi yang menentukan dijalankan melalui Maxima sebelum definisi.

function g(x) = x³ - exp(-x);g(x)

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

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

diff(g(x), x),

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

!images/Aljabar_Nandhita

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

g(5+g(1))

178.635099908

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolik lainnya.

function G(x) = factor(integrate(g(x),x)); G(c)//integrate : mengintegralkan

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

solve(g(x),0.5)

0.703467422498

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

solve(g,0.5)

0.703467422498

function P(x,n) = (2x-1)ⁿ;P(x,n)

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

function Q(x,n) = (x+2)ⁿ;Q(x,n)

$$(x+2)^n$$

P(x,4),expand(

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

! [images/Aljabar_Nandhita

P(3,4)

625

P(x,4) + Q(x,3),expand(

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

! [images/Aljabar_Nandhita

P(x,4) - Q(x,3),expand(

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

! [images/Aljabar_Nandhita

! [images/Aljabar_Nandhita

P(x,4)Q(x,3),expand(

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

! [images/Aljabar_Nandhita

! [images/Aljabar_Nandhita

P(x,4)/Q(x,1),expand(

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

! [images/Aljabar_Nandhita

! [images/Aljabar_Nandhita

function f(x) = x³ - x;f(x)

$$x^3 - x$$

Dengan amp;= fungsinya bersifat simbolis, dan dapat digunakan dalam ekspresi simbolik lainnya.

integrate(f(x),x)

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

Dengan := fungsinya numerik. Contoh yang baik adalah integral tertentu

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

yang tidak dapat dievaluasi secara simbolis.

Jika kita mendefinisikan ulang fungsi tersebut dengan kata kunci “peta” maka dapat digunakan untuk vektor x. Secara internal, fungsi ini dipanggil untuk semua nilai x satu kali, dan hasilnya disimpan dalam vektor.

```
function map f(x) := integrate("x^x", 1, x)
f(0:0.5:2)
[-0.78, -0.41, 0, 0.68, 2.1]
```

Fungsi dapat memiliki nilai default untuk parameter.

```
function mylog (x,base=10) := ln(x)/ln(base);
Sekarang fungsinya bisa dipanggil dengan atau tanpa parameter "base".
mylog(100), mylog(2^6.7, 2)
2 6.7
```

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

2

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

```
function f([a,b]) = a^2 + b^2 - ab + b; f(a,b), f(x,y)
```

$$y^2 - xy + y + x^2$$

![images/Aljabar_Nandhita

Fungsi simbolik seperti ini dapat digunakan untuk variabel simbolik.

Namun fungsinya juga dapat digunakan untuk vektor numerik.

```
v=[3,4]; f(v)
17
```

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

```
function lapl(expr,x,y) = diff(expr,x,2)+diff(expr,y,2)//turunan parsial ke-
dua
diff(expr, y, 2) + diff(expr, x, 2)
realpart((x + Iy)^4),lapl(
```

0

![images/Aljabar_Nandhita

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

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

$$10 (y^2 + x)^3 (9y^2 + x + 2)$$

Untuk meringkas

- * amp;= mendefinisikan fungsi simbolik,
- * := mendefinisikan fungsi numerik,
- * amp;amp;= mendefinisikan fungsi simbolik murni.

Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolis.

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

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

```
1.41421356237
```

Ini juga berfungsi untuk ekspresi simbolik. Ambil fungsi berikut.

```
solve(x^2 = 2, x)
```

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

```
solve(x^2 - 2, x)
```

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

```
solve(ax^2 + bx + c = 0, x)
```

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

```
solve([ax + by = c, dx + ey = f], [x, y])
```

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

```
px = 4x^8 + x^7 - x^4 - x;px
```

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

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

Kami menggunakan `y=2` dan memeriksa dengan mengevaluasi polinomial pada hasil sebelumnya.

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

```
0.966715594851 2
```

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

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

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

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

```
longest sol()
```

```
-0.6180339887498949 1.618033988749895
```

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

x^2 withsol[1],expand($x^2 - x - 1$ withsol[2])

0

!images/Aljabar_Nandhita

Penyelesaian sistem persamaan secara simbolis dapat dilakukan dengan vektor persamaan dan solver simbolis solve(). Jawabannya adalah daftar persamaan.

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

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

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

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

dengan a=3.

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

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

solve("f",3,2,y=0.1)

2.54116291558

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

solve(" $x^a - a^x$ ", a = 3, 2, y = 0.1)

2.54116291558

Menyelesaikan Pertidaksamaan

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

load(fourier_elim)

C:/Program Files/Euler x64/maxima/share/maxima/5.35.1/share/fourier_elim/fourier_elim.lisp

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

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

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

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

fourier_elim($[x^2 - 10], [x]$)/ $x^2 - 10 < 0$

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

fourier_elim($[x^2 - 10], [x]$)

$[x < 6] \vee [6 < x]$
fourier_elim($[x < 1, x \leq 1], [x]$)//tidakmemilikipenyelesaian
emptyset
fourier_elim($[\min f < x, x < \max f], [x]$)//solusinyaR
universalset
fourier_elim($[x^3 - 1 \leq 0], [x]$)
 $[1 < x, x^2 + x + 1 > 0] \vee [x < 1, -x^2 - x - 1 > 0]$
fourier_elim($[\cos(x) < 1/2], [x]$)//???gagal
 $[1 - 2 \cos x > 0]$
fourier_elim($[y - x < 5, x - y < 7, 10 < y], [x, y]$)//sistempertidaksamaan
 $[y - 5 < x, x < y + 7, 10 < y]$
fourier_elim($[y - x < 5, x - y < 7, 10 < y], [y, x]$)
 $[\max(10, x - 7) < y, y < x + 5, 5 < x]$
fourier_elim($((x + y < 5) \text{ and } (x - y < 8)), [x, y]$)
 $\left[y + 8 < x, x < 5 - y, y < -\frac{3}{2} \right]$
fourier_elim($((x + y < 5) \text{ and } x < 1) \text{ or } (x - y < 8), [x, y]$)
 $[y + 8 < x] \vee [x < \min(1, 5 - y)]$
fourier_elim($[\max(x, y) \leq 6, x \leq 8, \text{abs}(y - 1) \leq 12], [x, y]$)
 $[6 \leq x; x \leq 8; y \leq -11] \text{ or } [8 \leq x; x \leq -11] \text{ or } [x \leq 8; 13 \leq y] \text{ or } [x = y, 13 \leq y] \text{ or } [8 \leq x; x \leq y, 13 \leq y] \text{ or } [y \leq x, 13 \leq y]$
fourier_elim($[(x + 6)/(x - 9) \leq 6], [x]$)
 $[x = 12] \vee [12 < x] \vee [x < 9]$

Bahasa Matriks

Dokumentasi inti EMT berisi pembahasan rinci 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//perpangkatanelemen2A
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)
0.33 0.67 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
AA //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//perpangkatanelemen – elemenmatriks/vektor
9 16

```

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

```

2A
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
Adup([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 `lt`; 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 yang sama dengan 25 (hanya ada 1)
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
```

Dari vektor tersebut, "bukan nol" 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, 1e+02]
```

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

```
t=1:1000; nonzeros(mod(t2, 11) == 5 mod(t2, 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(t2 + 1)))
```

```
112
```

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

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

```
0.77 0.4 0.41 0.27 0.14 0.39 0.5 0.95 0.55 0 0.44 0.54
```

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
```

```
0.77 0.4 0.41 0 0 0 0.5 0.95 0.55 0 0.44 0.54
```

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

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

```
0.77 0.4 0.41 -0.13 -0.12 -0.69 0.5 0.95 0.55 -0.48 0.44 0.54
```

Dan dimungkinkan untuk mendapatkan elemen dalam vektor.

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

```
[0.27, 0.14, 0.39, 0]
```

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

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

```
0.27 4 0.77 1 0.14 1 0.95 4 0 2 0.55 1
```

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

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

```
[0.77, 0.95, 0.55]
```

Ini tentu saja sama dengan fungsi max().

```
max(A)'
```

```
[0.77, 0.95, 0.55]
```

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 [-0.77, -0.95, -0.55]
```

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'
0.032 0.053 0.6 0.56 1 0.84 0.18 0.4 0.84 2 0.026 0.66 0.63 0.77 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
0.032 0.053 0.6 0.56 1 0.84 0.18 0.4 0.84 1 0.026 0.66 0.63 0.77 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 panjang().

```
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 (distribusi seragam) atau normal (distribusi Gauß).

```
random(2,2)
0.67 0.83 0.98 0.54
```

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,ncols(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 1 0 0 0 0 1 0 0 0 0 1 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 ma-
trijs baru sebagai hasil dari setdiag().
Berikut adalah fungsi yang mengembalikan matriks tri-diagonal.
function tridiag (n,a,b,c) := setdiag(setdiag(bid(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
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
```

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, 1.4, 1.7]
```

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 ditampilkan
```

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

```
lateks: s = t3 - t
```

```
t=-1:0.1:1; s=t3 - t
```

```
[0, 0.17, 0.29, 0.36, 0.38, 0.38, 0.34, 0.27, 0.19, 0.099, 0, -0.099, -0.19, -0.27, -0.34, -0.38, -0.38, -0.36, -0.29, -0.17, 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
```

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

```
[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',
```

```
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 rasional 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]; vv'
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
vv, sum(vv), cumsum(vv)
[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 pendekan 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.

A4

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

Sekarang kita menambahkan kolom ke matriks.

M = deg(w)—w—cos(w)—sin(w)

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

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

Dalam contoh berikut, kita menghitung $t[j]^i$ untuk i dari 1 hingga n . Kita mendapatkan sebuah matriks, yang selnya $a_{i,j} = t_j^i$, $1 \leq j \leq 101$, $1 \leq i \leq n$

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 "peta" membuat vektorisasi fungsi tersebut. Fungsinya sekarang akan berfungsi

untuk vektor bilangan.

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

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

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] ...

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)
```

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

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.

```
inrandom(1,10,10), unique(
```

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

Ini juga berfungsi untuk vektor string.

```
unique(s)
```

```
a aa d e
```

Aljabar Linier

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` 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.

$A=\text{normal}(200,200)$; $b=\text{sum}(A)$; `longest totalmax(abs(inv(A).b-1))`
8.790745908981989e-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

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 `amp;:=`, 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`

$(a)111a111a$

`det(A),factor(`

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

![[images/Aljabar_Nandhita

`invert(A)witha = 0`

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

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

$(1)ab2$

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

`det(A-xident(2)),solve(`

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

!images/Aljabar_{Nandhita}

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

eigenvalues([a, 1; 1, a])

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

To extract a specific eigenvector needs careful indexing.

eigenvectors([a, 1; 1, a]),

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

[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.

Awith[a = 4, b = 5]

$$(1) 452$$

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

$$(t) + 1ab2$$

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)

$$(-) 21 \frac{3}{2} - \frac{1}{2}$$

!images/Aljabar_{Nandhita}

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 `amp;:=` 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)
```

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

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 `"amp;:="`.

```
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
```

$$(1) \frac{1146408}{364913}45$$

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

```
mxmset(A); A
```

$$(1) 3.14159265358979345$$

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))
```

$$1.414213562373095$$

!images/Aljabar_Nandhita

Ketepatan angka floating point besar dapat diubah.

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

```
3.14159265358979323846264338327950288419716939937510582097494 45923078164062862089986280348253
```

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

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% tambahkan satu tarif sederhana dan hitung hasilnya.

```
K1.03
```

```
5150
```

Euler would understand the following syntax too.

```
K+K3
```

```
5150
```

Namun lebih mudah menggunakan faktor tersebut

```
q=1+3
```

```
1.03 5150
```

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

```
Kq^10
```

```
6719.58189672
```

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

```
format(12,2); Kq^10
```

```
6719.58
```

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

```
"Starting from " + K + "you get" + round(Kq^10,2) + " ."
```

```
Starting from 5000you get6719.58.
```

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

```
Kq(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 q0 sampai q10. InidikalikandenganK, dankitamendapatkanvektornilainya.
VK=Kq(0 : 10);
Tentu saja, cara realistis untuk menghitung suku bunga ini adalah dengan
membulatkan ke sen terdekat setiap tahunnya. Mari kita tambahkan fungsi
untuk ini.
function oneyear (K) := round(Kq,2)
Mari kita bandingkan kedua hasil tersebut, dengan dan tanpa pembulatan.
longest oneyear(1234.57), longest 1234.57q
1271.61 1271.6071
Sekarang tidak ada rumus sederhana untuk tahun ke-n, dan kita harus men-
gulanginya selama bertahun-tahun. Euler memberikan banyak solusi untuk ini.
Cara termudah adalah fungsi iterate, yang mengulangi fungsi tertentu be-
berapa 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.
Memecahkan Persamaan
Sekarang kita mengambil fungsi yang lebih maju, yang menambahkan tingkat
uang tertentu setiap tahunnya.
function onepay (K) := Kq+R
Kita tidak perlu menentukan q atau R untuk definisi fungsi. Hanya jika kita
menjalankan perintah, kita harus mendefinisikan nilai-nilai ini. Kami memilih
R=200.
R=200; iterate("onepay",5000,10)

```


Real 1 x 11 matrix

5000.00 5350.00 5710.50 6081.82 ...

Bagaimana jika kita menghapus jumlah yang sama setiap tahun?

R=-200; iterate("onipay",5000,10)

Real 1 x 11 matrix

5000.00 4950.00 4898.50 4845.45 ...

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("onipay",5000,50)

Real 1 x 51 matrix

5000.00 4950.00 4898.50 4845.45 ...

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

min(nonzeros(VKR;0))

48.00

Alasannya adalah bukan nol (VKR[i]<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("onipay",5000,till="x<0"); x, n,

-19.83 47.00

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 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 tahun.

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

-336.08

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

Solusi 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) = Kq+R; op(K)

$$R + q K$$

Sekarang kita dapat mengulanginya.

op(op(op(op(K)))),expand(

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

!images/Aljabar_{Nandhita}

Kami melihat sebuah pola. Setelah n periode yang kita miliki

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

Rumusnya adalah rumus jumlah geometri yang diketahui Maxima.

sum(q^k, k, 0, n - 1);

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

Ini agak rumit. Jumlahnya dievaluasi dengan tanda "simpsum" untuk mengurangnya menjadi hasil bagi.

Mari kita membuat fungsi untuk ini.

function fs(K,R,P,n) = (1+P/100)ⁿK+((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=100i); equ

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

Kita dapat menyelesaikan nilai R secara simbolis.

solve(equ, R)

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

Seperti yang Anda lihat dari 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)

$$\lim_{x \rightarrow 0} R(5000,0,x,10)$$

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

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

3 plot 2d

Plot2D_{NandhitaPutriShalsabila}Nama : NandhitaPutriShalsabila

NIM : 23030630075

Kelas : Matematika B 2023

Menggambar Grafik 2D dengan EMT

Notebook ini menjelaskan tentang cara menggambar berbagai kurva 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. Terdapat koordinat layar yang selalu berkisar antara 0 hingga 1024 di setiap sumbu, tidak peduli apakah layarnya berbentuk persegi atau tidak. Semut terdapat koordinat plot, yang dapat diatur dengan `setplot()`. Pemetaan antar koordinat bergantung pada jendela plot saat ini. Misalnya, `shrinkwindow()` default menyisakan ruang untuk label sumbu dan judul plot.

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

```
clg; // clear screen
```

Perintah ini untuk membersihkan layar grafik atau jendela grafik.

```
window(0,0,1024,1024); // use all of the window
```

Perintah `window(0,0,1024,1024);`, untuk mengatur jendela grafik atau tampilan utama agar memanfaatkan seluruh ruang dengan ukuran 1024x1024 piksel dan mulai dari posisi (0,0) di layar.

```
setplot(0,1,0,1); // set plot coordinates
```

Perintah `setplot(0,1,0,1);`, untuk mengatur sumbu X untuk berkisar dari 0 hingga 1 dan sumbu Y untuk berkisar dari 0 hingga 1

```
hold on; // start overwrite mode
```

Perintah ini untuk menambahkan beberapa plot atau elemen grafis ke dalam satu jendela grafik tanpa menghapus plot sebelumnya.

```
n=100; X=random(n,2); Y=random(n,2); // get random points
```

`n = 100;`: Menetapkan variabel `n` dengan nilai 100. Ini menentukan jumlah baris yang akan digunakan dalam matriks yang akan dihasilkan.

`X = random(n,2);`: Menghasilkan matriks `X` dengan ukuran 100x2 yang berisi angka acak. Setiap elemen di `X` adalah angka acak antara 0 dan 1.

`Y = random(n,2);`: Menghasilkan matriks `Y` dengan ukuran 100x2 yang juga berisi angka acak antara 0 dan 1.

```
colors=rgb(random(n),random(n),random(n)); // get random colors
```

`random(n)`: Menghasilkan vektor atau matriks dengan `n` elemen yang berisi angka acak antara 0 dan 1. Misalnya, jika `n = 100`, maka `random(n)` akan menghasilkan vektor dengan 100 elemen acak dalam rentang `[0,1]`.

`rgb(random(n), random(n), random(n))`: Menggunakan tiga vektor acak dari fungsi `random(n)` untuk mengatur komponen merah, hijau, dan biru dari warna. Setiap komponen (merah, hijau, biru) adalah vektor dengan `n` elemen acak yang mengindikasikan intensitas warna dalam rentang `[0,1]`.

colors: Menyimpan hasil dari `rgb(random(n), random(n), random(n))`, yang merupakan matriks warna dengan n baris, di mana setiap baris berisi nilai RGB acak. Setiap baris dari `colors` mewakili satu warna acak.

```
loop 1 to n; color(colors[]); plot(X[],Y[]); end; // plot
```

Perintah ini digunakan untuk memplot n titik, di mana setiap titik memiliki koordinat $(X[i], Y[i])$ dan warna yang berbeda sesuai dengan warna acak yang telah dihasilkan dan disimpan dalam variabel `colors`. Setiap iterasi dari loop akan memplot satu titik dengan warna yang sesuai, sehingga menghasilkan visualisasi di mana setiap titik mungkin memiliki warna yang berbeda.

```
hold off; // end overwrite mode
```

perintah `gt;hold off`; untuk memastikan bahwa setiap kali grafik baru dibuat, grafik sebelumnya dihapus dan tidak ditampilkan lagi.

```
insimg; // insert to notebook
```

```
!images/Plot2DNandhita
```

Perintah `gt;insimg`; untuk menyisipkan gambar grafik atau visualisasi yang baru saja dihasilkan ke dalam notebook. Setelah menghasilkan grafik atau visualisasi, perintah ini akan menambahkannya sebagai gambar yang dapat dilihat di bagian notebook.

```
reset;
```

perintah ini mengembalikan EMT ke keadaan seperti baru dibuka, sehingga Anda dapat memulai sesi baru tanpa harus menutup dan membuka ulang aplikasi.

Grafik perlu ditahan, 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.

Contoh lain, kita menggambar plot sebagai sisipan di plot lain. Hal ini dilakukan dengan mendefinisikan jendela plot yang lebih kecil. Perhatikan bahwa jendela ini tidak memberikan 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 inset.

```
plot2d("x3 - x");
```

Perintah ini akan menghasilkan grafik kurva dari fungsi $y = x^3 - x$ di bidang koordinat 2D.

```
xw=200; yw=100; ww=300; hw=300;
```

Jendela grafik akan diposisikan pada koordinat (200, 100) di layar.

Jendela grafik akan memiliki ukuran lebar 300 dan tinggi 300 piksel.

engan kata lain, perintah ini untuk mengatur ukuran dan posisi jendela grafik yang akan muncul di layar.

```
ow=window();
```

Perintah `gt;ow=window()`; untuk mendapatkan informasi tentang posisi dan ukuran jendela grafik saat ini.

```
window(xw,yw,xw+ww,yw+hw);
```

Perintah tersebut akan memindahkan jendela grafik ke koordinat (200, 100) di layar dengan ukuran lebar 300 piksel dan tinggi 300 piksel.

`hold on;`

Perintah `gt;hold on;` digunakan untuk menahan grafik saat ini sehingga grafik baru yang ditambahkan akan digabungkan dengan grafik yang sudah ada tanpa menghapus grafik sebelumnya.

`barclear(xw-50,yw-10,ww+60,ww+60);`

`xw-50`: Koordinat x dari pojok kiri atas area yang ingin dihapus, dikurangi 50 piksel dari `xw` (posisi horizontal jendela grafik).

`yw-10`: Koordinat y dari pojok kiri atas area yang ingin dihapus, dikurangi 10 piksel dari `yw` (posisi vertikal jendela grafik).

`w+60`: Lebar area yang akan dihapus, ditambah 60 piksel dari `ww` (lebar jendela grafik).

`w+60`: Tinggi area yang akan dihapus, sama dengan lebar tambahan, sehingga membentuk area persegi atau bujur sangkar.

Dengan perintah ini, dapat menghapus area tertentu dari jendela grafik yang mungkin diperlukan untuk membersihkan ruang sebelum menggambar atau mengubah grafik yang ada.

`plot2d("x4 - x", grid = 6) :`

`!images/Plot2DNandhita`

`plot2d("x4 - x"); untuk membuat grafik fungsi $y = x^4 - x$ dalam sistem koordinat 2D.`

`rid=6`: Menentukan interval grid atau skala grid pada grafik. Angka 6 di sini menunjukkan jarak atau interval garis-garis grid pada grafik. Misalnya, jika `grid=6`, maka garis grid akan muncul setiap 6 unit pada sumbu.

`hold off;`

Dengan perintah `gt;hold off;`, mode `hold` dinonaktifkan, dan setiap grafik baru yang dibuat akan menggantikan grafik sebelumnya. Dengan kata lain, grafik sebelumnya akan dihapus, dan hanya grafik yang baru akan ditampilkan.

`window(ow);`

Perintah `gt;window(ow);` digunakan untuk mengatur jendela grafik ke posisi dan ukuran yang telah disimpan sebelumnya dalam variabel `ow`.

Plot dengan banyak gambar dicapai dengan cara yang sama. Ada fungsi `figure()` untuk ini.

Contoh soal

`plot2d("x4 - x", grid = 5) :`

`!images/Plot2DNandhita`

Aspek Plot

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

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

`aspect(2);` // rasio panjang dan lebar 2:1

`aspect(2);` // rasio panjang dan lebar 2:1: Perintah ini mengatur rasio aspek dari plot yang dihasilkan. Dengan `aspect(2)`, maka dapat mengatur rasio

panjang dan lebar plot menjadi 2:1, yang berarti lebar plot akan dua kali lipat dari tingginya. Ini berguna untuk memastikan bahwa grafik ditampilkan dengan proporsi yang sesuai sehingga data lebih mudah dianalisis.

```
plot2d(["sin(x)", "cos(x)"], 0, 2pi);
!images/Plot2DNandhita
plot2d(["sin(x)", "cos(x)"], 0, 2pi);
```

Perintah ini menghasilkan grafik 2D dari dua fungsi, yaitu $\sin(x)$ dan $\cos(x)$, dalam rentang 0 hingga 2π . `plot2d` adalah fungsi yang digunakan untuk membuat plot dua dimensi, dan parameter `["sin(x)", "cos(x)"]` menentukan fungsi yang akan diplot. Rentang 0 hingga 2π menunjukkan domain x untuk fungsi-fungsi tersebut.

```
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 membuat plot di Maxima menggunakan Gnuplot atau dengan Python menggunakan Math Plot Lib.

Euler dapat membuat plot 2D

- * ekspresi

- * fungsi, variabel, atau kurva berparameter,

- * vektor nilai x - y ,

- * awan titik di pesawat,

- * kurva implisit dengan level atau wilayah level.

- * Fungsi kompleks

Gaya plot mencakup berbagai gaya untuk garis dan titik, plot batang, dan plot berbayang.

Plot Ekspresi atau Variabel

Ekspresi tunggal dalam " x " (misalnya " $4*x^2$ ") atau nama suatu fungsi (misalnya " f ") menghasilkan grafik fungsi.

Berikut adalah contoh paling dasar, yang menggunakan rentang default dan menetapkan rentang y yang tepat agar sesuai dengan plot fungsinya.

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") :
```

```
!images/Plot2DNandhita
```

`plot2d("x^2");` akan menghasilkan grafik dari fungsi x^2 dalam rentang default, menampilkan parabola yang memotong titik $(0, 0)$ dan membesar ke arah sumbu $-x$ positif dan negatif.

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

```
!images/Plot2DNandhita
```

Perintah `aspect(1.5); plot2d("x^3 - x");` akan mengatur rasio aspek plot menjadi 1.5 :

1 dan kemudian menggambar grafik dari fungsi $x^3 - x$. Grafik ini akan menampilkan bentuk kurva dari fungsi tersebut.

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

```
!images/Plot2DNandhita
```

Perintah `a:=5.6; plot2d("exp(-a*x^2)/a"); insimg(30);`

Mendefinisikan nilai a sebagai 5.6 kemudian membuat plot 2D dari fungsi $(\exp(-5.6 \cdot x^2))/5.6$ dan menampilkan hasil plot dengan ukuran gambar yang tinggi 25 baris (atau sesuai dengan spesifikasi).

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 sebagai berikut

- * a,b: rentang x (default -2,2)
- * c,d: rentang y (default: skala dengan nilai)
- * r: alternatifnya radius di sekitar pusat plot
- * cx,cy: koordinat pusat plot (default 0,0)

plot2d("x³ - x", -1, 2) :

!images/Plot2D_{Nandhita}

Perintah plot2d("x³ - x", -1, 2); untuk membuat grafik dari fungsi

$x^3 - x$ dengan sumbu-x yang terbentang dari -1 hingga 2. Hal ini untuk melihat perilaku fungsi dalam rentang

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

!images/Plot2D_{Nandhita}

Perintah plot2d("sin(x)", -2*pi, 2*pi); digunakan untuk membuat grafik 2D dari fungsi sin(x) dengan sumbu-x yang terbentang dari -2pi hingga 2pi. Ini akan menampilkan gelombang sinusoidal yang penuh dalam rentang tersebut, untuk melihat beberapa periode dari fungsi sinus.

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

!images/Plot2D_{Nandhita}

Perintah plot2d("cos(x)", "sin(3*x)", xmin=0, xmax=2pi); akan menghasilkan grafik dua fungsi:

cos(x) dan sin(3*x), dalam rentang sumbu-x dari 0 hingga 2pi.

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

Dalam opsi, plot dapat diatur agar 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.gambar(0) mengatur ulang jendela default.

reset;

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

!images/Plot2D_{Nandhita}

Perintah reset; figure(2,2); for n=1 to 4; figure(n); plot2d("x" + n); end; figure(0); untuk melakukan hal berikut

1. Mengatur ulang lingkungan kerja.
2. Mengatur ukuran jendela grafik menjadi 2x2 unit.
3. Menggunakan loop untuk menghasilkan empat jendela grafik terpisah, masing-masing menampilkan grafik dari fungsi x^n untuk n dari 1 hingga 4.
4. Mengembalikan fokus ke jendela grafik utama atau jendela grafik default.

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

```
figure(3,3); ... for k=1:9; figure(k); plot2d("x^3-x", -2, 1, grid = k); end; ...figure(0) :
!images/Plot2DNandhita
```

Perintah ini melakukan hal berikut:

1. Mengatur ukuran jendela grafik menjadi 3x3 unit.
2. Menggunakan loop untuk menghasilkan sembilan jendela grafik terpisah, masing-masing menampilkan grafik dari fungsi $x^3 - x$ dalam rentang dari -2 hingga 1.
3. Mengatur grid untuk setiap jendela grafik dengan intensitas grid yang berbeda, berdasarkan nilai `k` dari 1 hingga 9.
4. Mengembalikan fokus ke jendela grafik default setelah semua grafik telah digambar.

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

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

Pada 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)");
!images/Plot2DNandhita
```

Perintah ini menghasilkan grafik dari fungsi $\sin(x)$ dalam interval dari 0 hingga 2π , dengan sumbu-`y` yang mencakup rentang dari -1.2 hingga 1.2. Grafik ini akan ditampilkan dengan rasio aspek 1.5:1, grid dengan tingkat kepadatan 3, serta label yang sesuai pada sumbu-`x` dan sumbu-`y`.

```
plot2d("sin(x)+cos(2x)", 0, 4pi);
!images/Plot2DNandhita
```

Perintah `plot2d("sin(x)+cos(2*x)", 0, 4pi)`; untuk membuat grafik dari fungsi $\sin(x) + \cos(2x)$ dalam rentang `x` dari 0 hingga 2π . Ini memungkinkan visualisasi pola gelombang yang dihasilkan oleh kombinasi fungsi sinus dan kosinus dalam interval yang luas.

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

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

Fungsi atau ekspresi di `plot2d` dievaluasi secara adaptif. Agar lebih cepat, nonaktifkan plot adaptif dengan `lt;adaptive` dan tentukan jumlah subinterval dengan `n=...`. Hal ini hanya diperlukan pada kasus yang jarang terjadi.

```
plot2d("sign(x)exp(-x^2)", -1, 1, < adaptive, n = 10000) :
!images/Plot2DNandhita
```

Perintah `plot2d("sign(x)*exp(-x^2)", -1, 1, lt;adaptive, n = 10000)`; untuk menghasilkan grafik dua dimensi 1 hingga 1. Grafik ini dibuat dengan menggunakan metode adaptif untuk menyesuaikan titik data dengan resolusi

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

!images/Plot2D_Nandhita

Perintah `plot2d("x^x", r = 1.2, cx = 1, cy = 1);` dapat menghasilkan grafik dari fungsi x^x dengan pengaturan

1. Rasio aspek diatur dengan $r=1.2$ untuk mengubah proporsi grafik.
2. Grafik dipusatkan pada koordinat (1, 1) dengan $cx=1$ dan $cy=1$, yang bisa mempengaruhi penempatan grafik dalam jendela plot.

Perhatikan bahwa x^x tidak ditentukan untuk $x \leq 0$. Fungsi `plot2d` menangkap kesalahan ini, dan mulai memplot

`plot2d("log(x)", -0.1, 2);`

!images/Plot2D_Nandhita

Perintah `plot2d("log(x)", -0.1, 2);` dapat menghasilkan grafik dua dimensi dari fungsi $\log(x)$ dengan rentang sumbu-x dari -0.1 hingga 2. Dalam praktiknya, grafik akan valid hanya untuk bagian dari sumbu-x di atas 0, dan mungkin hanya menunjukkan plot dari x yang mendekati 0 hingga 2, dengan bagian dari sumbu-x yang negatif tidak digunakan dalam grafik.

Parameter `square=true` (atau `gt;square`) memilih rentang y secara otomatis sehingga hasilnya adalah jendela plot persegi. Perhatikan bahwa secara default, Euler menggunakan spasi persegi di dalam jendela plot.

`plot2d("x^3 - x", square);`

!images/Plot2D_Nandhita

Perintah `plot2d("x^3-x", gt;square);` dapat menghasilkan grafik dua dimensi dari fungsi $x^3 - x$ dengan jendela plot yang diatur dalam bentuk persegi, memastikan bahwa grafik tidak terdistorsi dan sumbu-x serta sumbu-y memiliki skala yang sama.

`plot2d("integrate("sin(x)exp(-x^2)", 0, x)", 0, 2); //plot integral`

!images/Plot2D_Nandhita

Perintah `plot2d('integrate("sin(x)*exp(-x^2)", 0, x)', 0, 2);` untuk melakukan hal berikut :

1. Menghitung integral dari fungsi $\sin(x) \cdot \exp(-x^2)$ dari 0 hingga x , menghasilkan fungsi integral kumulatif.
2. Memplot hasil integral ini dalam rentang x dari 0 hingga 2.

Ini dapat memvisualisasikan bagaimana nilai integral kumulatif dari fungsi $\sin(x) \cdot \exp(-x^2)$ berubah sepanjang interval x dari 0 hingga 2.

Jika Anda memerlukan lebih banyak ruang untuk label y, panggil `shrinkwindow()` dengan parameter lebih kecil, atau tetapkan nilai positif untuk "lebih kecil" di `plot2d()`.

`plot2d("gamma(x)", 1, 10, yl="y-values", smaller=6, ivertical);`

!images/Plot2D_Nandhita

Perintah `plot2d("gamma(x)", 1, 10, yl="y-values", smaller=6, lt;vertical);` untuk melakukan hal berikut:

1. Menghasilkan grafik dua dimensi dari fungsi $\Gamma(x)$ dalam rentang x dari 1 hingga 10.
2. Menetapkan label sumbu-y sebagai "y-values".
3. Mengatur ukuran elemen teks dalam grafik menjadi lebih kecil dengan ukuran font 6.
4. Mengubah orientasi teks atau elemen grafik menjadi vertikal.

Eksresi simbolik juga dapat digunakan karena disimpan sebagai ekspresi string sederhana.

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

!images/Plot2D_{Nandhita}

Perintah `x=linspace(0,2pi,1000); plot2d(sin(5x), cos(7x))`; untuk melakukan hal berikut:

1. Menghasilkan vektor `x` dengan 1000 titik dari 0 hingga 2π menggunakan `linspace`.

2. Membuat grafik dua dimensi dari fungsi $\sin(5x)$ dan $\cos(7x)$ menggunakan vektor `x` yang dihasilkan, menampilkan kedua fungsi dalam grafik yang sama.

Hal ini untuk melihat bagaimana fungsi sinus dan kosinus dengan frekuensi berbeda berperilaku dan dibandingkan satu sama lain dalam interval dari 0 hingga 2π .

`a:=5.6; expr = exp(-a*x^2)/a; //defineexpression`

`plot2d(expr,-2,2): // plot from -2 to 2`

!images/Plot2D_{Nandhita}

Perintah `a:=5.6; expr amp:= exp(-a*x^2)/a; untukmendefinisikansebuahfungsi(exp(-5.6.x^2)/5.6)dimana 2hingga2`.

Grafik ini akan menunjukkan bentuk lonceng dari fungsi eksponensial tereduksi dengan parameter $a=5.6$, mencakup interval sumbu- x dari -2 hingga 2.

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

!images/Plot2D_{Nandhita}

Perintah `plot2d(expr, r=1, thickness=2)`; untuk menghasilkan grafik dari fungsi yang didefinisikan (`expr`), dengan tampilan sebagai berikut:

1. `r=1`: Memastikan bahwa plot ditampilkan dalam format persegi (rasio aspek 1:1), sehingga sumbu- x dan sumbu- y memiliki skala yang sama.

2. `thickness=2`: Mengatur ketebalan garis grafik menjadi 2 unit, membuat garis grafik lebih tebal dan lebih jelas terlihat.

Dengan ini, grafik fungsi $(\exp(-5.6.x^2)/5.6)$

akan ditampilkan dalam jendela plot yang berbentuk persegi dengan garis yang lebih tebal, memudahkan visualisasi fungsi tersebut di sekitar titik (0,0).

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

!images/Plot2D_{Nandhita}

Perintah `plot2d(amp;diff(expr,x), gt;add, style="--", color=red)`; untuk melakukan hal berikut:

1. Menghitung turunan pertama dari ekspresi `expr` (yang sebelumnya didefinisikan sebagai $(\exp(-5.6.x^2)/5.6)$)

2. Menambahkan plot dari turunan fungsi ke grafik yang sudah ada.

3. Menggunakan garis putus-putus untuk grafik turunan.

4. Mengatur warna garis grafik turunan menjadi merah.

Dengan ini, grafik turunan dari fungsi $(\exp(-5.6.x^2)/5.6)$

akan ditambahkan ke grafik yang sudah ada, menampilkan fungsi asal bersama dengan grafik dari turunannya dalam warna merah dan gaya garis putus-putus.

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

!images/Plot2D_{Nandhita}

Perintah `plot2d(amp;diff(expr,x,2), a=-2, b=2, c=-2, d=1)`; untuk melakukan hal berikut:

1. Menghitung turunan kedua dari fungsi yang didefinisikan sebelumnya (`expr`), yang dalam hal ini adalah $(\exp(-5.6.x^2)/5.6)$.

2. Memplot turunan kedua fungsi tersebut dalam rentang sumbu-x dari -2 hingga 2 dan sumbu-y dari -2 hingga 1.

Plot ini akan menunjukkan bagaimana kelengkungan dari fungsi $(\exp(5.6.x^2))/5.6$ berubah dalam intervalnya, titik dimana fungsi tersebut cekung atau cembung dalam rentang yang ditentukan.

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

```
!images/Plot2DNandhita
```

Perintah `plot2d(amp;diff(expr,x), a=-2, b=2, gt;square)`; untuk melakukan hal berikut:

1. Menghitung turunan pertama dari fungsi yang didefinisikan sebagai `expr`, yaitu $(\exp(-5.6.x^2))/5.6$.

2. Memplot grafik turunan pertama fungsi tersebut dalam rentang sumbu-x dari -2 hingga 2.

3. Menetapkan tampilan grafik dengan rasio aspek 1:1 menggunakan `gt;square`, sehingga grafik ditampilkan dalam format persegi.

Dengan pengaturan ini, grafik dari turunan pertama fungsi akan ditampilkan dengan skala yang konsisten di sumbu-x dan sumbu-y, memberikan visualisasi yang akurat dari perubahan turunan dalam rentang yang ditentukan.

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

```
!images/Plot2DNandhita
```

Perintah `plot2d("x^2", 0, 1, steps = 1, color = red, n = 10)`; untuk melakukan hal berikut :

1. Menghasilkan grafik dua dimensi dari fungsi x^2 dalam rentang sumbu - x dari 0 hingga 1.

2. Menggunakan metode penggambaran yang melibatkan interpolasi linear antar titik data dengan `steps=1`.

3. Mewarnai grafik dengan warna merah.

4. Menggunakan 10 titik data untuk menggambar grafik, memberikan kekasaran atau kehalusan pada garis grafik.

Dengan ini, grafik dari fungsi x^2 akan digambarkan dalam warna merah, menggunakan 10 titik data dalam rentang titik dengan garis lurus.

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

```
!images/Plot2DNandhita
```

Perintah `plot2d("x^2", gt;add, steps = 2, color = blue, n = 10)`; untuk melakukan hal berikut :

1. Menambahkan grafik dari fungsi x^2 ke grafik yang sudah ada.

2. Menggunakan metode penggambaran dengan interpolasi antar dua titik data.

3. Mengatur warna grafik menjadi biru.

4. Menggunakan 10 titik data untuk menggambar grafik.

Dengan ini, grafik dari fungsi x^2 akan ditambahkan ke grafik yang sudah ada sebelumnya, dengan interpolasi Fungsi dalam satu Parameter

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

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

```

function f(x,a) := x2/a + ax2 - x; //define a function
a=0.3; plot2d("f",0,1;a): // plot with a=0.3
!images/Plot2DNandhita

```

Perintah ini akan menghasilkan grafik dari fungsi f dalam interval [0, 1] dengan parameter a = 0.3.

```

memplot gambar grafik dengan fungsi f nya , interval x nya titik koma a nya.
plot2d("f",0,1;0.4): // plot with a=0.4
!images/Plot2DNandhita

```

Perintah ini akan menggambarkan grafik dari fungsi f(x, 0.4) dalam interval [0, 1], untuk melihat bagaimana perubahan nilai a mempengaruhi bentuk grafik fungsi.

```

plot2d("f",0.2,0,1): // plot with a=0.2
!images/Plot2DNandhita

```

Perintah ini akan menghasilkan grafik dari fungsi f(x, 0.2) dalam interval [0, 1], untuk menganalisis bagaimana nilai a yang berbeda mempengaruhi grafik fungsi tersebut

```

memplot grafik fungsi dengan fungsi f nya, parameter atau a x nya
plot2d("f(x,b)",b=0.1,0,1): // plot with 0.1
!images/Plot2DNandhita

```

Perintah ini akan menghasilkan grafik dari fungsi f(x, 0.1) dalam interval [0, 1]. untuk melihat bagaimana fungsi berperilaku dengan nilai b yang spesifik.

```

memplot grafik fungsi yang sudah diketahui nilai b(parameter)nya.
function f(x) := x3 - x; ... plot2d("f",r = 1) :
!images/Plot2DNandhita

```

Perintah tersebut digunakan untuk mendefinisikan fungsi kubik $f(x)=x^3-x$ dan kemudian memplot grafiknya dalam rentang dari -1 hingga 1.

Berikut ini ringkasan fungsi yang diterima

- * ekspresi atau ekspresi simbolik di x
- * fungsi atau fungsi simbolik dengan nama "f"
- * fungsi simbolik hanya dengan nama f

Fungsi plot2d() juga menerima fungsi simbolik. Untuk fungsi simbolik, namanya saja yang berfungsi.

```

function f(x) = diff(xx, x)
x x (log(x) + 1)

```

Perintah tersebut mendefinisikan fungsi yang mengembalikan turunan dari x^x , yaitu $f(x) = d/dx(x^x) = x^x(\ln(x) + 1)$.

```

plot2d(f,0,2):
!images/Plot2DNandhita

```

Perintah ini akan memplot grafik dari turunan $f(x)=d/dx(x^x)$ untuk x dalam rentang dari 0 hingga 2.

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

```

expr = sin(x)exp(-x)
- x E sin(x)

```

Perintah ini, expr mengandung ekspresi simbolik $\sin(x) \cdot e^{-x}$, dan dapat melakukan operasi matematika lebih lanjut.

- Fillcolor : Lihat di atas untuk mengetahui warna yang tersedia.

Warna isian ditentukan oleh argumen "fillcolor", dan pada lt;outline optional, mencegah menggambar batas untuk semua gaya kecuali gaya default.

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

```
!images/Plot2DNandhita
```

Perintah ini akan memplot grafik dari ekspresi $\sin(x)$. e^{-x} dalam rentang x dari 0 hingga sekitar 9.42 (atau 3π).

```
function f(x) = xx;
```

```
plot2d(f,r=1,cx=1,cy=1,color=blue,thickness=2);
```

```
plot2d(diff(f(x),x), add,color=red,style="-.-"):
```

```
!images/Plot2DNandhita
```

Perintah ini memplot grafik fungsi x^x dalam rentang yang ditentukan, dengan warna biru dan ketebalan garis 2.

Kemudian, perintah kedua memplot turunan dari fungsi tersebut $f'(x) = x^x(\ln(x) + 1)$.

1) di atas grafik yang sama dengan warna merah dan gaya garis yang berupa titik – titik bercampur dengan garis putus – putus.

Hasilnya adalah grafik dari fungsi x^x dan grafik turunannya diplot secara bersamaan untuk perbandingan visual.

Yang biru adalah fungsinya dan yang merah adalah turunan dari fungsinya.

Untuk gaya garis ada berbagai pilihan.

* gaya = "...". Pilih dari "-", "_", "-.", ".", ".-", "-.-".

* Warna: 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 tua, oranye, hijau muda, pirus, biru * muda, oranye muda, kuning

* rgb(merah,hijau,biru): parameternya real di [0,1].

```
plot2d("exp(-x2)", r = 2, color = red, thickness = 3, style = " - -") :
```

```
!images/Plot2DNandhita
```

Perintah ini membuat grafik fungsi e^{-x^2} dengan :

Rentang x dari -2 hingga 2

Warna garis merah

Garis yang lebih tebal (ketebalan 3 unit)

Dan garis bergaya putus-putus.

Grafik ini akan menunjukkan bentuk lonceng simetris yang dikenal sebagai fungsi Gaussian atau kurva lonceng, yang sering digunakan dalam probabilitas dan statistik.

Berikut adalah tampilan warna EMT yang telah ditentukan sebelumnya.

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

```
!images/Plot2DNandhita
```

Rasio aspek grafik diatur menjadi 2 (lebih lebar).

Sebuah plot kolom dengan 16 kolom dibuat, semua kolom memiliki tinggi 1.

Label pada sumbu-x ditetapkan dari 0 hingga 15, dan warna setiap kolom berbeda, diambil dari skala warna yang tersedia (warna 0 hingga 15).

Grid tidak ditampilkan.

Hasil akhirnya adalah plot kolom dengan kolom yang berwarna berbeda, berlabel dari 0 hingga 15, dan rasio lebar grafik dua kali tinggi.

Tapi Anda bisa menggunakan warna apa saja.

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

```
!images/Plot2DNandhita
```

Plot kolom memiliki 16 kolom, semuanya dengan tinggi 1.

Warna kolom bervariasi dari biru gelap ke biru terang berdasarkan gradasi yang dibuat oleh fungsi `linspace`.

Grid dinonaktifkan, sehingga tidak ada garis kisi yang muncul di grafik.

Hasil akhirnya adalah sebuah plot kolom dengan warna biru gradasi yang berubah dari gelap ke terang di sepanjang kolom, tanpa grid.

Menggambar Beberapa Kurva pada bidang koordinat yang sama

Plot lebih dari satu fungsi (multiple function) ke dalam satu jendela dapat dilakukan dengan berbagai cara. Salah satu metodenya adalah menggunakan `gt;add` untuk beberapa panggilan ke `plot2d` secara keseluruhan, kecuali panggilan pertama. Kami telah menggunakan fitur ini pada contoh di atas.

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

```
!images/Plot2DNandhita
```

`aspect()`: Mengatur rasio aspek dari grafik agar proporsional.

`plot2d("cos(x)", r=2, grid=6)`: Menggambar grafik fungsi $\cos(x)$ dengan rentang sumbu $r=2$ dan grid berjumlah 6. Ini menunjukkan fungsi cosinus dalam interval yang ditentukan.

`plot2d("x", style=".", gt;add)`: Menggambar garis lurus

$y=x$ dengan gaya titik, dan menggunakan `gt;add` untuk menambahkan grafik ini ke grafik sebelumnya tanpa menghapusnya.

Jadi, perintah tersebut secara keseluruhan menggambarkan grafik dari

$\cos(x)$ dan garis

$y=x$ dalam satu gambar.

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

```
!images/Plot2DNandhita
```

`spect(1.5)`: Mengatur rasio aspek grafik, sehingga sumbu x dan y memiliki proporsi 1.5:1, memberikan tampilan yang lebih baik pada grafik.

`plot2d("sin(x)", 0, 2pi)`: Menggambar grafik fungsi $\sin(x)$ dalam interval dari 0 hingga 2π .

`plot2d("cos(x)", color=blue, style="--", gt;add)`: Menggambar grafik fungsi

$\cos(x)$ dengan garis putus-putus (`style="--"`) dan warna biru (`color=blue`), serta menambahkan grafik ini ke grafik sebelumnya tanpa menghapus grafik $\sin(x)$.

Salah satu kegunaan `gt;add` adalah untuk menambahkan titik pada kurva.

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

```
!images/Plot2DNandhita
```

`plot2d("sin(x)", 0, pi)`: Menggambar grafik fungsi $\sin(x)$ dari 0 hingga π .

`plot2d(2, sin(2), gt;points, gt;add)`: Menambahkan titik ke grafik pada koordinat $(2, \sin(2))$.

`gt;points`: Menyatakan bahwa titik yang ditambahkan akan ditampilkan sebagai titik pada grafik.

`gt;add`: Menambahkan titik tersebut ke grafik yang sudah ada tanpa menghapus grafik $\sin(x)$.

Jadi, perintah ini menggambarkan grafik $\sin(x)$ dalam interval 0 hingga pi dan menambahkan titik di posisi (2, sin(2)) ke grafik tersebut.

Kita tambahkan titik perpotongan dengan label (pada posisi "cl" untuk kiri tengah), dan masukkan hasilnya ke dalam buku catatan. Kami juga menambahkan judul pada 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);
!images/Plot2DNandhita
```

Secara keseluruhan, perintah ini menggambarkan grafik fungsi $\cos(x)$ dan $y=x$, menunjukkan titik perpotongan antara keduanya, dan menambahkan label untuk memberikan informasi lebih lanjut tentang titik tersebut.

Dalam contoh berikut, kita memplot fungsi $\sin(x)=\sin(x)/x$ dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung perluasan ini menggunakan Maxima melalui ekspresi simbolik.

Plot ini dilakukan dalam perintah multi-baris berikut dengan tiga panggilan ke `plot2d()`. Yang kedua dan ketiga memiliki kumpulan tanda `gt;add`, yang membuat plot menggunakan rentang sebelumnya.

Kami menambahkan kotak label yang menjelaskan fungsinya.

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

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

memplot fungsi $\sin x$ dari pangkat 0 sampai 4

```
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]);
!images/Plot2DNandhita
```

Secara keseluruhan, perintah ini menggambarkan grafik fungsi

$\sin(x)$ serta dua polinomial Taylor untuk membandingkan seberapa baik aproksimasi Taylor mendekati fungsi asli pada interval yang diberikan, dengan label yang jelas untuk setiap grafik.

Dalam contoh berikut, kami menghasilkan Polinomial Bernstein.

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

```
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;
!images/Plot2DNandhita
```

Cara kedua adalah dengan menggunakan pasangan matriks bernilai x dan matriks bernilai y yang berukuran sama.

Kami menghasilkan matriks nilai dengan satu Polinomial Bernstein di setiap baris. Untuk ini, kita cukup menggunakan vektor kolom i. Lihat pendahuluan 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)xk(1-x)(n-k); //y is a matrix then
plot2d(x,y):
!images/Plot2DNandhita
```

Secara keseluruhan, perintah ini menggambarkan grafik distribusi probabilitas binomial untuk n=10 pada interval x dari 0 hingga 1, menunjukkan bagaimana probabilitas berubah berdasarkan nilai k.

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):
!images/Plot2DNandhita
```

Secara keseluruhan, perintah ini menggambarkan grafik dari fungsi

$y = x^k$ untuk k dari 1 hingga 10, dengan setiap kurva ditampilkan dalam warna yang berbeda. Ini memberikan visi

Metode lain adalah menggunakan vektor ekspresi (string). Anda kemudian dapat menggunakan susunan warna, susunan gaya, dan susunan ketebalan dengan panjang yang sama.

```
plot2d(["sin(x)", "cos(x)"], 0, 2pi, color=4:5):
!images/Plot2DNandhita
```

Perintah ini menggambarkan grafik dari fungsi sin(x) dan cos(x) dalam interval dari 0 hingga 2pi, dengan warna yang berbeda untuk masing-masing fungsi, sehingga keduanya dapat dengan mudah dibedakan dalam grafik.

```
plot2d(["sin(x)", "cos(x)"], 0, 2pi): // plot vector of expressions
!images/Plot2DNandhita
```

Perintah ini menggambarkan grafik fungsi sin(x) dan cos(x) dalam interval dari 0 hingga 2pi. Kedua grafik ini akan ditampilkan dalam satu plot, memungkinkan perbandingan visual antara kedua fungsi trigonometri tersebut.

Kita bisa mendapatkan vektor seperti itu dari Maxima menggunakan makelist() dan mxm2str().

```
v = makelist(binomial(10,i)xi(1-x)(10-i), i, 0, 10) //makelist
10 9 8 2 7 3 [(1-x)9, 10(1-x)8x, 45(1-x)7x2, 120(1-x)6x3, 645(1-x)5x4, 252(1-x)4x5, 210(1-x)3x6, 120(1-x)2x7, 45(1-x)x8, 10(1-x)x9, x]
```

Perintah ini menghasilkan daftar v yang berisi nilai distribusi probabilitas binomial untuk n=10 berdasarkan berbagai nilai i dari 0 hingga 10. Setiap elemen dalam daftar v mewakili probabilitas untuk masing-masing jumlah keberhasilan i dalam 10 percobaan, tergantung pada nilai x.

fungsi makelist digunakan untuk membuat list dari elemen-elemen yang dihasilkan oleh ekspresi didalamnya berdasarkan parameter yang diberikan.

```
mxm2str(v) // get a vector of strings from the symbolic vector
(1-x)1010*(1-x)9*x45*(1-x)8*x2120*(1-x)7*x3210*(1-x)6*x4252*
(1-x)5*x5210*(1-x)4*x6120*(1-x)3*x745*(1-x)2*x810*(1-x)*x9x10
```

Perintah ini bertujuan untuk menghasilkan vektor string dari vektor simbolik yang ada, memudahkan manipulasi dan presentasi data.

```
mxm2str untuk vektor string
plot2d(mxm2str(v),0,1): // plot functions
!images/Plot2DNandhita
```

Perintah ini menggambarkan grafik dari fungsi-fungsi yang terdapat dalam vektor simbolik v dalam rentang x dari 0 hingga 1.

Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

Jika suatu ekspresi menghasilkan matriks fungsi, dengan satu fungsi di setiap baris, semua fungsi tersebut akan diplot ke dalam satu plot.

Untuk ini, gunakan vektor parameter dalam bentuk vektor kolom. Jika array warna ditambahkan maka akan digunakan untuk setiap baris plot.

```
n=(1:10)'; plot2d("x^n", 0, 1, color = 1 : 10) :
!images/Plot2DNandhita
```

Perintah ini bertujuan untuk menghasilkan grafik dari fungsi x^n untuk n dari 1 hingga 10 dalam interval x dari 0 hingga 1, masing-masing berdasarkan warnanya.

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 meletakkan semua parameter yang ditetapkan di akhir perintah plot2d. Dalam contoh ini kita meneruskan $a=5$ ke fungsi f , yang kita plot dari -10 hingga 10.

```
function f(x,a) := 1/aexp(-x^2/a); ...plot2d("f", -10, 10; 5, thickness = 2, title =
"a = 5") :
```

```
!images/Plot2DNandhita
```

Perintah ini bertujuan untuk menggambarkan grafik fungsi distribusi normal dengan parameter $a=5$ dalam rentang x dari -10 hingga 10. Grafik ini akan menunjukkan bagaimana fungsi eksponensial ini berperilaku dalam rentang tersebut.

Alternatifnya, gunakan koleksi dengan nama fungsi dan semua parameter tambahan. Daftar khusus ini disebut kumpulan panggilan, dan ini adalah cara yang lebih disukai untuk meneruskan argumen ke suatu fungsi yang kemudian diteruskan sebagai argumen ke fungsi lain.

Pada contoh berikut, kita menggunakan loop untuk memplot beberapa fungsi (lihat tutorial tentang pemrograman loop).

```
plot2d("f",1,-10,10); ... for a=2:10; plot2d("f",a, add); end:
!images/Plot2DNandhita
```

Perintah ini untuk menggambarkan grafik fungsi f untuk parameter a dari 1 hingga 10 dalam rentang x dari -10 hingga 10. Setiap grafik akan ditambahkan ke grafik sebelumnya, sehingga dapat melihat bagaimana fungsi berubah dengan variasi nilai a . Dengan demikian, grafik akan menampilkan fungsi distribusi normal untuk berbagai nilai a dalam satu plot.

Kita dapat mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris matriks $f(x,a)$ merupakan satu fungsi. Selain

itu, kita dapat mengatur warna untuk setiap baris matriks. Klik dua kali pada fungsi `getspectral()` untuk penjelasannya.

```
x=-10:0.01:10; a=(1:10)'; plot2d(x,f(x,a),color=getspectral(a/10));
```

```
!images/Plot2DNandhita
```

Perintah ini untuk menggambarkan grafik fungsi $f(x,a)$ untuk nilai dari 1 hingga 10 dalam rentang x dari -10 hingga 10. Dengan menggunakan palet warna spektral, grafik akan ditampilkan dengan warna yang berbeda untuk setiap nilai a , memungkinkan perbandingan visual antara fungsi untuk berbagai parameter.

Label Teks

Dekorasi sederhana pun bisa

```
* judul dengan judul = "..."
```

```
* label x dan y dengan xl="...", yl="..."
```

```
* label teks lain dengan label("...",x,y)
```

Perintah label akan memplot ke plot saat ini pada koordinat plot (x,y) . Hal ini memerlukan argumen posisional.

```
plot2d("x3 - x", -1, 2, title = "y = x3 - x", yl = "y", xl = "x") :
```

```
!images/Plot2DNandhita
```

Perintah ini menghasilkan grafik dari fungsi $y=x^3 - x$ dalam rentang

x elemen atau anggota $[-1,2]$, dengan judul dan label pada kedua sumbu.

```
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");
```

```
!images/Plot2DNandhita
```

Perintah ini menggambar grafik fungsi

$$y = \frac{\log(x)}{x}$$

menambahkan label pada titik $(1,0)$, dan juga menandai titik maksimum di $x=e$ dengan label "Max".

Ada juga fungsi `labelbox()`, yang dapat menampilkan fungsi dan teks. Dibutuhkan vektor string dan warna, satu item untuk setiap fungsi.

```
function f(x) = x2exp(-x2); ... 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) :
```

```
!images/Plot2DNandhita
```

Perintah ini menggambar grafik dari fungsi $f(x)=x^2 \cdot e^{-x^2}$ dan turunannya $f'(x)$ pada rentang elemen atau anggota

Kotak ini berlabuh di kanan atas secara default, tetapi `gt`; kiri berlabuh di kiri atas. Anda dapat memindahkannya ke tempat mana pun yang Anda suka. Posisi jangkar berada di pojok kanan atas kotak, dan angkanya merupakan pecahan dari ukuran jendela grafis. Lebar nya otomatis.

Untuk plot titik, kotak label juga berfungsi. Tambahkan parameter `gt`; points, atau vektor bendera, satu untuk setiap label.

Pada contoh berikut, hanya ada satu fungsi. Jadi kita bisa menggunakan string sebagai pengganti 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):
```

!images/Plot2D_Nandhita

Perintah ini menggambar distribusi koefisien binomial ($10/k$)

untuk $k=0$ hingga 10, dengan titik-titik pada setiap nilai binomialnya, dan menambahkan kotak label "Binomials" untuk menjelaskan grafik yang digambar.

Gaya plot ini juga tersedia di `statplot()`. Seperti di `plot2d()` warna dapat diatur untuk setiap baris plot. Masih banyak lagi plot khusus untuk keperluan statistik (lihat tutorial tentang statistik).

```
statplot(1:10,random(2,10),color=[red,blue]):
```

!images/Plot2D_Nandhita

Perintah ini membuat plot statistik dari dua set data acak(masing-masing terdiri dari 10 titik), dengan nilai sumbu x dari 1 hingga 10, dan warna yang berbeda (merah dan biru) untuk masing-masing set data.

Fitur serupa adalah fungsi `textbox()`.

Lebar nya secara default adalah lebar maksimal baris teks. Tapi itu bisa diatur oleh pengguna juga.

```
function f(x) = exp(-x)sin(2pix); ... plot2d("f(x)",0,2pi); ... textbox(latex("
textExample of a damped oscillation
```

```
f(x)=e-xsin(2
pix")), w = 0.85) :
```

!images/Plot2D_Nandhita

Perintah ini mendefinisikan fungsi osilasi teredam, memplot grafik dari fungsi tersebut dalam satu siklus, dan menambahkan teks deskriptif dengan format LaTeX untuk memberikan keterangan visual pada grafik.

Label teks, judul, kotak label, dan teks lainnya dapat berisi string Unicode (lihat sintaks EMT untuk mengetahui lebih lanjut tentang string Unicode).

```
plot2d("x3 - x", title = u"xrarr; xsup3; -x") :
```

!images/Plot2D_Nandhita

Judul tersebut menggunakan simbol Unicode untuk menggambarkan transformasi dari x ke $x^3 - x$, *membuat representasi yang lebih rapid dan estetis pada grafik.*

Label pada sumbu x dan y bisa vertikal, begitu juga dengan sumbunya.

```
plot2d("sinc(x)",0,2pi,xl="x",yl=u"x rarr; sinc(x)", vertical):
```

!images/Plot2D_Nandhita

Perintah ini menggambar grafik fungsi $\text{sinc}(x)$ dari 0 hingga 2π , dengan label sumbu y yang menggambarkan transformasi dari x ke $\text{sinc}(x)$, dan orientasi grafik dalam posisi vertikal.

LaTeX

Anda juga dapat memplot rumus LaTeX jika Anda telah menginstal sistem LaTeX. Saya merekomendasikan MiKTeX. Jalur ke biner "latex" dan "dvi2png" harus berada di jalur sistem, atau Anda harus mengatur LaTeX di menu opsi.

Perhatikan, penguraian LaTeX lambat. Jika Anda ingin menggunakan LaTeX dalam plot animasi, Anda harus memanggil `latex()` sebelum loop satu kali dan menggunakan hasilnya (gambar dalam matriks RGB).

Pada 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("
textFunction
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) :
```

!images/Plot2D_Nandhita

Perintah ini untuk membuat plot fungsi teredam sinusoidal, memberi judul, label pada sumbu, serta tambahan teks dan label pada bagian tertentu dari plot untuk memperjelas.

Seringkali, kita menginginkan spasi dan label teks yang tidak konformal pada sumbu x. Kita bisa menggunakan xaxis() dan yaxis() seperti yang akan kita tunjukkan nanti.

Cara termudah adalah membuat plot kosong dengan bingkai menggunakan grid=4, lalu menambahkan grid dengan ygrid() dan xgrid(). Pada contoh berikut, kami 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):
```

!images/Plot2D_Nandhita

Perintah ini untuk menghasilkan plot yang informatif dan mudah dibaca untuk fungsi sinc, dengan pengaturan yang jelas pada sumbu x dan y serta grid yang memudahkan analisis visual.

Tentu saja fungsinya juga bisa digunakan.

```
function map f(x) ...
if x<0 then return x^4else return x^2endifendfunction < /pre > function map f(x) :
mende finisikan sebuah fungsi bernama f yang menerima satu argumen input x.
if x > 0 then return x^4 : sebuah pernyataan kondisi. Jika x lebih besar dari 0, fungsi akan mengembalikan x^4.
else return x^2 : jika kondisi sebelumnyatidak terpenuhi (yaitu, jika x tidak lebih besar dari 0), fungsi akan men
endif : menandai akhir dari pernyataan kondisi.
endfunction : menandai akhir dari definisi fungsi.
```

Parameter "peta" membantu menggunakan fungsi untuk vektor. Untuk plot, itu tidak perlu. Tapi untuk menunjukkan vektorisasi itu berguna, kita menambahkan beberapa poin penting ke plot di 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 bisa menggunakannya
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) =
begincases x^3 x 0
```

x^2x
 $le0$

endcases"), ... $x = 0.7, y = 0.2$) :

!images/Plot2D_Nandhita

Perintah ini menciptakan plot yang informatif dari fungsi bersyarat
f(x), dengan tambahan label, titik-titik penting, dan kotak teks yang men-
jelaskan fungsi tersebut secara rinci.

Interaksi Pengguna

Saat memplot suatu fungsi atau ekspresi, parameter gt;pengguna memu-
ngkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol
kursor atau mouse. Pengguna bisa

- * perbesar dengan + atau -
- * pindahkan plot dengan tombol kursor
- * pilih jendela plot dengan mouse
- * atur ulang tampilan dengan spasi
- * keluar dengan kembali

Tombol spasi akan mengatur ulang plot ke jendela plot aslinya.

Saat memplot data, flag gt;user hanya akan menunggu penekanan tombol.

```
plot2d("x^3 - ax", a = 1, user, title = "Press any key!") :
```

!images/Plot2D_Nandhita

Perintah ini membuat plot dari fungsi $x^3 - x$ (*karena a diatur ke 1*) dan mengizinkan pengguna untuk berinteraksi
plot2d("exp(x)sin(x)", user=true, ... title="+/- or cursor keys (return to
exit)"):

!images/Plot2D_Nandhita

Perintah ini untuk menciptakan plot interaktif dari fungsi

$\exp(x) \cdot \sin(x)$ dan memberikan instruksi kepada pengguna tentang cara berin-
teraksi dengan grafik tersebut.

Berikut ini menunjukkan cara interaksi pengguna tingkat lanjut (lihat tuto-
rial tentang pemrograman untuk detailnya).

Fungsi bawaan mousedrag() menunggu aktivitas mouse atau keyboard. Ini
melaporkan mouse ke bawah, gerakan mouse atau mouse ke atas, dan penekanan
tombol. Fungsi dragpoints() memanfaatkan ini, dan memungkinkan pengguna
menyeret titik plot.

Kita membutuhkan fungsi plot terlebih dahulu. Misalnya, kita melakukan
interpolasi pada 5 titik dengan polinomial. Fungsi tersebut harus diplot ke
dalam area plot yang tetap.

```
function plotf(xp,yp,select) ...
```

```
d=interp(xp,yp); plot2d("interpval(xp,d,x)" ;d,xp,r=2); plot2d(xp,yp,i,points,i,add);
if select i 0 then plot2d(xp[select],yp[select],color=red,i,points,i,add); endif; ti-
tle("Drag one point, or press space or return!"); endfunction i/pre i
```

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 nilainya secara global.

Sekarang kita menghasilkan beberapa nilai acak, dan membiarkan pengguna menyeret titiknya.

```
t=-1:0.5:1; dragpoints("plotf",t,random(size(t))-0.5):
!images/Plot2DNandhita
```

Ada juga fungsi yang memplot fungsi lain bergantung pada vektor parameter, dan memungkinkan pengguna menyesuaikan parameter ini.

Pertama kita membutuhkan fungsi plot.

```
function plotf([a,b]) := plot2d("exp(ax)cos(2pibx)",0,2pi;a,b);
```

Kemudian kita memerlukan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional garis judul.

Ada penggeser interaktif, yang dapat menetapkan nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

```
dragvalues("plotf",["a","b"],[-1,2],[[-2,2],[1,10]], ... heading="Drag these
values:",hcolor=black):
!images/Plot2DNandhita
```

Dimungkinkan untuk membatasi nilai yang diseret menjadi bilangan bulat. Sebagai contoh, kita menulis fungsi plot, yang memplot polinomial Taylor berderajat n ke fungsi kosinus.

```
function plotf(n) ...
plot2d("cos(x)",0,2pi,i,square,grid=6); plot2d("taylor(cos(x),x,0,@n)",color=blue,i,add);
textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",i,left); endfunction
i/pre i
```

Sekarang kita izinkan derajat n bervariasi dari 0 hingga 20 dalam 20 perhentian. Hasil 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):
```

```
!images/Plot2DNandhita
```

Berikut ini adalah demonstrasi sederhana dari fungsinya. Pengguna dapat menggambar 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 i/pre i dragtest //
lihat hasilnya dan cobalah lakukan!
```

Gaya Plot 2D

Secara default, EMT menghitung tick sumbu otomatis dan menambahkan label ke setiap tick. Ini dapat diubah dengan parameter grid. Gaya default sumbu dan label dapat diubah. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk menyetel ulang ke gaya default, gunakan reset().

```
aspect();
```

```

figure(3,4); ... figure(1); plot2d("x3-x", grid = 0); ...//nograd, frameoraxis
figure(2); plot2d("x3 - x", grid = 1); ...//x - y - axis
figure(3); plot2d("x3 - x", grid = 2); ...//defaultticks
figure(4); plot2d("x3 - x", grid = 3); ...//x - y - axiswithlabelsinside
figure(5); plot2d("x3 - x", grid = 4); ...//noticks, onlylabels
figure(6); plot2d("x3 - x", grid = 5); ...//default, butnomargin
figure(7); plot2d("x3 - x", grid = 6); ...//axesonly
figure(8); plot2d("x3 - x", grid = 7); ...//axesonly, ticksataxis
figure(9); plot2d("x3 - x", grid = 8); ...//axesonly, finerticksataxis
figure(10); plot2d("x3 - x", grid = 9); ...//default, smallticksinside
figure(11); plot2d("x3 - x", grid = 10); ...//noticks, axesonly
figure(0):

```

!images/Plot2D_Nandhita

Parameter lt:frame mematikan frame, dan framecolor=blue mengatur frame menjadi warna biru.

Jika Anda menginginkan tanda centang Anda sendiri, Anda dapat menggunakan style=0, dan menambahkan semuanya nanti.

```

aspect(1.5);
plot2d("x3 - x", grid = 0); //plot
frame; xgrid([-1,0,1]); ygrid(0); // add frame and grid

```

!images/Plot2D_Nandhita

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=ex")); //titleabovetheplot
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

```

!images/Plot2D_Nandhita

Sumbu dapat digambar secara terpisah dengan xaxis() dan yaxis().

```

plot2d("x3 - x", < grid, < frame);
xaxis(0,xx=-2:1,style="- "); yaxis(0,yy=-5:5,style="- ");

```

!images/Plot2D_Nandhita

Teks pada plot dapat diatur dengan label(). Dalam contoh berikut, "lc" berarti bagian tengah bawah. Ini menetapkan posisi label relatif terhadap koordinat plot.

```

function f(x) = x3 - x
3 x - 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

```

!images/Plot2D_Nandhita

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

```


!images/Plot2D_Nandhita

Perintah ini menghasilkan plot dari fungsi dan turunan pertama, serta memberikan penjelasan visual yang jelas dengan kotak label untuk masing-masing grafik.

```
plot2d(["exp(x)", "1+x"], color=[black, blue], style=["-", "-."]):
```

!images/Plot2D_Nandhita

Perintah ini menciptakan plot yang jelas dan terpisah untuk kedua fungsi dengan pengaturan warna dan gaya yang membedakan keduanya, sehingga memudahkan visualisasi dan perbandingan antara $\exp(x)$ dan $1+x$.

Guna tingkat lanjut (lihat tutorial tentang pemrograman untuk detailnya).

Fungsi bawaan `mousedrag()` menunggu aktivitas mouse atau keyboard. Ini melaporkan mouse ke bawah, gerakan mouse atau mouse ke atas, dan penekanan tombol. Fungsi `dragpoints()` memanfaatkan ini, dan memungkinkan pengguna menyeret titik mana pun dalam plot.

Kita membutuhkan fungsi plot terlebih dahulu. Misalnya, kita melakukan interpolasi pada 5 titik dengan polinomial. Fungsi tersebut harus diplot ke dalam area plot yang tetap.

```
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() :
```

!images/Plot2D_Nandhita

Untuk kontrol lebih lanjut, sumbu x dan sumbu y dapat dilakukan secara manual.

Perintah `fullwindow()` memperluas jendela plot karena kita tidak lagi memerlukan tempat untuk label di luar jendela plot. Gunakan `shrinkwindow()` atau `reset()` untuk menyetel 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 :
```

!images/Plot2D_Nandhita

Berikut adalah contoh lain, di mana string Unicode digunakan dan sumbu berada di luar area plot.

```
aspect(1.5);  
plot2d(["sin(x)", "cos(x)"], 0, 2pi, color=[red, green], igrid, iframe); ... xaxis(-  
1.1, (0:2)pi, xt=["0", u"pi", u"2pi"], style="-", ticks, zero); ... xgrid((0:0.5:2) ×  
pi, i ticks); ... yaxis(-0.1pi, -1:0.2:1, style="-", zero, grid); ... labelbox(["sin", "cos"], colors=[red, green], x=0.5, y=  
... xlabel(u"phi"); ylabel(u"f(phi)");
```

!images/Plot2D_Nandhita

Perintah ini untuk membuat plot yang informatif dan estetik dari fungsi sinus dan kosinus dengan pengaturan yang jelas pada sumbu dan label, serta grid untuk membantu visualisasi.

Merencanakan Data 2D

Jika x dan y adalah vektor data, maka data tersebut akan digunakan sebagai koordinat x dan y pada suatu kurva. Dalam hal ini, a , b , c , dan d , atau radius r

dapat ditentukan, atau jendela plot akan menyesuaikan secara otomatis dengan data. Alternatifnya, `gt;persegi` dapat diatur untuk mempertahankan rasio aspek persegi.

Merencanakan ekspresi hanyalah singkatan dari plot data. Untuk plot data, Anda memerlukan satu atau beberapa baris nilai x , dan satu atau beberapa baris nilai y . Dari rentang dan nilai x , fungsi `plot2d` akan menghitung data yang akan diplot, secara default dengan evaluasi fungsi yang adaptif. Untuk plot titik gunakan `"gt;titik"`, untuk garis dan titik campuran gunakan `"gt;addpoints"`.

Tapi Anda bisa 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) :
```

```
!images/Plot2DNandhita
```

Data juga dapat diplot sebagai poin. Gunakan `points=true` untuk ini. Plot-nya berfungsi seperti poligon, tetapi hanya menggambar sudutnya saja.

- * `style="..."`: Pilih dari `"[]"`, `"lt;gt;"`, `"o"`, `"."`, `".."`, `"+"`, `"*"`, `"[]"`, `"*"`, `"lt;gt;"`, `"o"`, `"."`, `".."`, `"_"`.

Untuk memplot kumpulan titik, gunakan `gt;titik`. Jika warna merupakan vektor warna, masing-masing titik

mendapat warna berbeda. Untuk matriks koordinat dan vektor kolom, warna diterapkan pada baris matriks.

Parameter `gt;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
```

```
!images/Plot2DNandhita
```

Perintah ini untuk membuat plot dari data dengan garis yang menghubungkan titik-titik, serta menambahkan titik-titik tersebut sebagai lingkaran untuk memperjelas lokasi data dalam grafik.

```
p=polyfit(xdata,ydata,1); // get regression line
plot2d("polyval(p,x)", add,color=red): // add plot of line
```

```
!images/Plot2DNandhita
```

Perintah tersebut digunakan untuk menghitung dan menampilkan garis regresi linier berdasarkan data yang diberikan.

Menggambar Daerah Yang Dibatasi Kurva

Plot data sebenarnya berbentuk poligon. Kita juga dapat memplot kurva atau kurva terisi.

- * `filled=benar` mengisi plot.

- * `style="..."`: Pilih dari `"_"`, `"/"`, `"_;"`, `"_;"`.

- * `Fillcolor` : Lihat di atas untuk mengetahui warna yang tersedia.

Warna isian ditentukan oleh argumen `"fillcolor"`, dan pada `lt;outline` opsional, mencegah menggambar batas untuk semua gaya kecuali gaya 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):
!images/Plot2DNandhita

```

Dalam contoh berikut kita memplot elips terisi dan dua segi enam terisi menggunakan kurva tertutup dengan 6 titik dengan gaya isian berbeda.

```

x=linspace(0,2pi,1000); plot2d(sin(x),cos(x)0.5,r=1, filled,style="/" ):
!images/Plot2DNandhita
t=linspace(0,2pi,6); ... plot2d(cos(t),sin(t), filled,style="/" ,fillcolor=red,r=1.2):
!images/Plot2DNandhita
t=linspace(0,2pi,6); plot2d(cos(t),sin(t), filled,style=""):
!images/Plot2DNandhita

```

Contoh lainnya adalah septagon yang kita buat dengan 7 titik pada lingkaran satuan.

```

t=linspace(0,2pi,7); ... plot2d(cos(t),sin(t),r=1, filled,style="/" ,fillcolor=red):
!images/Plot2DNandhita

```

Berikut adalah himpunan nilai maksimal dari empat kondisi linier yang kurang dari atau sama dengan 3. Ini adalah $A[k].vlt;=3$ untuk semua baris A. Untuk mendapatkan sudut yang bagus, kita 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):
!images/Plot2DNandhita

```

Poin utama dari bahasa matriks adalah memungkinkan pembuatan tabel fungsi dengan mudah.

```

t=linspace(0,2pi,1000); x=cos(3t); y=sin(4t);

```

Kami sekarang memiliki nilai vektor x dan y. `plot2d()` dapat memplot nilai-nilai ini

sebagai kurva yang menghubungkan titik-titik tersebut. Plotnya bisa diisi. Dalam hal ini

ini menghasilkan hasil yang bagus karena aturan belitan, yang digunakan untuk

isi.

```

plot2d(x,y,igrid,iframe, filled):
!images/Plot2DNandhita

```

Vektor interval diplot terhadap nilai x sebagai wilayah terisi antara nilai interval yang lebih rendah dan lebih tinggi.

Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga dapat 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):
!images/Plot2DNandhita

```

Jika x adalah vektor yang diurutkan, dan y adalah vektor interval, maka `plot2d` akan memplot rentang interval yang terisi pada bidang. Gaya isian sama dengan gaya poligon.

```

t=-1:0.01:1; x= t-0.01,t+0.01 ; y=x3 - x;

```

```
plot2d(t,y):
```

```
! [images/Plot2DNandhita
```

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.

```
expr := "2x2 + xy + 3y4 + y"; //define an expression f(x,y)
```

```
plot2d(expr,level=[0;1],style="-",color=blue): // 0 i= f(x,y) i= 1
```

```
! [images/Plot2DNandhita
```

Kita juga dapat mengisi rentang nilai seperti

```
plot2d("(x2 + y2)2 - x2 + y2", r = 1.2, level = [-1; 0], style = "/" ) :
```

```
! [images/Plot2DNandhita
```

```
plot2d("cos(x)", "sin(x)3", xmin = 0, xmax = 2pi, filled, style = "/" ) :
```

```
! [images/Plot2DNandhita
```

Grafik Fungsi Parametrik

Nilai x tidak perlu diurutkan. (x,y) hanya menggambarkan sebuah kurva.

Jika x diurutkan, kurva tersebut merupakan grafik suatu fungsi.

Dalam contoh berikut, kita memplot spiral

Kita perlu menggunakan banyak titik untuk tampilan yang halus atau fungsi adaptif() untuk mengevaluasi ekspresi (lihat fungsi adaptif() untuk lebih jelasnya).

```
t=linspace(0,1,1000); ... plot2d(tcos(2pit),tsin(2pit),r=1):
```

```
! [images/Plot2DNandhita
```

Sebagai alternatif, dimungkinkan untuk menggunakan dua ekspresi untuk kurva. Berikut ini plot kurva yang sama seperti di atas.

```
plot2d("xcos(2pix)", "xsin(2pix)", xmin=0, xmax=1, r=1):
```

```
! [images/Plot2DNandhita
```

```
t=linspace(0,1,1000); r=exp(-t); x=rcos(2pit); y=rsin(2pit);
```

```
plot2d(x,y,r=1):
```

```
! [images/Plot2DNandhita
```

Pada contoh berikutnya, kita memplot kurvanya

$$\gamma(t) = (r(t) \cos(t), r(t) \sin(t))$$

with

$$r(t) = 1 + \sin(3t)2.$$

```
t=linspace(0,2pi,1000); r=1+sin(3t)/2; x=rcos(t); y=rsin(t); ... plot2d(x,y, filled, fillcolor=red, style = "/" ,r=1.
```

```
! [images/Plot2DNandhita
```

Menggambar Grafik Bilangan Kompleks

Serangkaian bilangan kompleks juga dapat diplot. Kemudian titik-titik grid akan dihubungkan. Jika sejumlah garis kisi ditentukan (atau vektor garis kisi 1x2) dalam argumen cgrid, hanya garis kisi tersebut yang terlihat.

Matriks bilangan kompleks secara otomatis akan diplot sebagai kisi-kisi pada bidang kompleks.

Pada 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=rexp(Ia);... plot2d(z,a=-
1.25,b=1.25,c=-1.25,d=1.25,cgrid=10):

```

```

!images/Plot2DNandhita
    aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=rexp(Ia);
    plot2d(exp(z),cgrid=[40,10]):
!images/Plot2DNandhita
    r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=rexp(Ia);
    plot2d(exp(z), points, add):
!images/Plot2DNandhita

```

Vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian nyata dan bagian imajiner.

Dalam contoh, kita memplot lingkaran satuan dengan
 In the example, we plot the unit circle with

$$\gamma(t) = e^{it}$$

```

t=linspace(0,2pi,1000); ... plot2d(exp(It)+exp(4It),r=2):
!images/Plot2DNandhita
Plot Statistik

```

Ada banyak fungsi yang dikhususkan pada plot statistik. Salah satu plot yang sering digunakan adalah plot kolom.

Jumlah kumulatif dari nilai terdistribusi normal 0-1 menghasilkan jalan acak.

```

plot2d(cumsum(randnormal(1,1000))):
!images/Plot2DNandhita
Penggunaan dua baris menunjukkan jalan dalam dua dimensi.
X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
!images/Plot2DNandhita
columnplot(cumsum(random(10)),style="/" ,color=blue):
!images/Plot2DNandhita

```

Itu 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];
columnplot(values,lab=months,color=red,style="-");
title("Temperature");
!images/Plot2DNandhita

```

```

k=0:10;
plot2d(k,bin(10,k), bar):
!images/Plot2DNandhita
plot2d(k,bin(10,k)); plot2d(k,bin(10,k), points, add):
!images/Plot2DNandhita
plot2d(normal(1000),normal(1000), points,grid=6,style=".."):
!images/Plot2DNandhita
plot2d(normal(1,1000), distribution,style="O"):
!images/Plot2DNandhita
plot2d("qnormal",0,5;2.5,0.5, filled):
!images/Plot2DNandhita

```

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
!images/Plot2DNandhita
```

Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan `gt;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):
!images/Plot2DNandhita
```

Fungsi `statplot()` mengatur gaya dengan string sederhana.
`statplot(1:10,cumsum(random(10)),"b"):`

```
!images/Plot2DNandhita
n=10; i=0:n; ... plot2d(i,bin(n,i)/2n, a = 0, b = 10, c = 0, d = 0.3); ...plot2d(i, bin(n,i)/2n, points =
true, style = "ow", add = true, color = blue) :
!images/Plot2DNandhita
```

Selain itu, data dapat diplot sebagai batang. Dalam hal ini, `x` harus diurutkan dan satu elemen lebih panjang dari `y`. Batangnya akan memanjang dari `x[i]` hingga `x[i+1]` dengan nilai `y[i]`. Jika `x` berukuran sama dengan `y`, maka `x` akan diperpanjang 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):
!images/Plot2DNandhita
```

Data untuk plot batang (`batang=1`) dan histogram (`histogram=1`) dapat diberikan secara eksplisit dalam `xv` dan `yv`, atau dapat dihitung dari distribusi empiris dalam `xv` dengan `gt;distribusi` (atau `distribusi=n`). Histogram nilai `xv` akan dihitung secara otomatis dengan `gt;histogram`. Jika `gt;even` ditentukan, nilai `xv` akan dihitung dalam interval bilangan bulat.

```
plot2d(normal(10000),distribution=50):
!images/Plot2DNandhita
k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m, bar):
!images/Plot2DNandhita
columnplot(m,k):
!images/Plot2DNandhita
plot2d(random(600)6,histogram=6):
!images/Plot2DNandhita
```

Untuk distribusi, terdapat parameter `distribution=n`, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan `n` sub-interval.

```
plot2d(normal(1,1000),distribution=10,style="
/"):
!images/Plot2DNandhita
```

Dengan parameter `even=true`, ini akan menggunakan interval bilangan bulat.

```
plot2d(intrandom(1,1000,10),distribution=10,even=true):
!images/Plot2DNandhita
```

Perhatikan bahwa ada banyak plot statistik yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

```
columnsplot(getmultiplicities(1:6,intrandom(1,6000,6))):
```

```
!images/Plot2DNandhita
```

```
plot2d(normal(1,1000), distribution); ... plot2d("qnormal(x)",color=red,thickness=2, add):
```

```
!images/Plot2DNandhita
```

Ada juga banyak plot khusus untuk statistik. Plot kotak menunjukkan kuartil distribusi ini dan banyak outlier. Menurut definisinya, outlier dalam plot kotak adalah data yang melebihi 1,5 kali rentang 50

```
M=normal(5,1000); boxplot(quartiles(M)):
```

```
!images/Plot2DNandhita
```

Fungsi Implisit

Plot implisit menunjukkan penyelesaian garis level $f(x,y)=\text{level}$, dengan "level" dapat berupa nilai tunggal atau vektor nilai. Jika level = "auto", akan ada garis level nc, yang akan tersebar antara fungsi minimum dan maksimum secara merata. Warna yang lebih gelap atau lebih terang dapat ditambahkan dengan gt;hue untuk menunjukkan nilai fungsi. Untuk fungsi implisit, xv harus berupa fungsi atau ekspresi parameter x dan y, atau alternatifnya, 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 pada bidang. Parameter c adalah level=c, dimana c dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi setiap titik dalam plot. Parameter "n" menentukan kehalusan plot.

```
aspect(1.5);
```

```
plot2d("x2 + y2 - xy - x", r = 1.5, level = 0, contourcolor = red) :
```

```
!images/Plot2DNandhita
```

```
expr := "2x2 + xy + 3y4 + y"; //define an expression f(x,y)
```

```
plot2d(expr,level=0): // Solutions of f(x,y)=0
```

```
!images/Plot2DNandhita
```

```
plot2d(expr,level=0:0.5:20, hue,contourcolor=white,n=200): // nice
```

```
!images/Plot2DNandhita
```

```
plot2d(expr,level=0:0.5:20, hue, spectral,n=200,grid=4): // nicer
```

```
!images/Plot2DNandhita
```

Ini juga berfungsi untuk plot data. Namun Anda harus menentukan rentangnya 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):
```

```
!images/Plot2DNandhita
```

```
plot2d("x3 - y2", contour, hue, spectral) :
```

```
!images/Plot2DNandhita
```

```
plot2d("x3 - y2", level = 0, contourwidth = 3, add, contourcolor = red) :
```

```
!images/Plot2DNandhita
```

```

z=z+normal(size(z))0.2;
plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):
!images/Plot2DNandhita
plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):
!images/Plot2DNandhita
plot2d("x2 + y3 + xy", level = 1, r = 4, n = 100) :
!images/Plot2DNandhita
plot2d("x2 + 2y2 - xy", level = 0 : 0.1 : 10, n = 100, contourcolor =
white, hue) :
!images/Plot2DNandhita
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 i= f(x,y) i= 1
!images/Plot2DNandhita
Plot implisit juga dapat menunjukkan rentang level. Maka level harus
berupa matriks interval level 2xn, di mana baris pertama berisi awal dan baris
kedua berisi akhir setiap interval. Alternatifnya, vektor baris sederhana dapat
digunakan untuk level, dan parameter dl memperluas nilai level ke interval.
plot2d("x4 + y4", r = 1.5, level = [0; 1], color = blue, style = "/" ) :
!images/Plot2DNandhita
plot2d("x2 + y3 + xy", level = [0, 2, 4; 1, 3, 5], style = "/" , r = 2, n = 100) :
!images/Plot2DNandhita
plot2d("x2 + y3 + xy", level = -10 : 20, r = 2, style = " - ", dl = 0.1, n =
100) :
!images/Plot2DNandhita
plot2d("sin(x)cos(y)", r=pi, hue, levels,n=100):
!images/Plot2DNandhita
Dimungkinkan juga untuk menandai suatu wilayah
Hal ini dilakukan dengan menambahkan level dengan dua baris.
plot2d("(x2+y2-1)3-x2y3", r = 1.3, ...style = "", color = red, < outline, ...level =
[-2; 0], n = 100) :
!images/Plot2DNandhita
Dimungkinkan untuk menentukan level tertentu. Misalnya, kita dapat mem-
plot solusi persamaan seperti

```

$$x^3 - xy + x^2y^2 = 6$$

```

plot2d("x3 - xy + x2y2", r = 6, level = 1, n = 100) :
!images/Plot2DNandhita
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 poly-
    gon([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); hold-
    ing(h); window(w); endfunction i/pre;

```

Tidak ada tanda centang kotak atau sumbu di sini. Selain itu, kami menggunakan jendela penuh untuk plotnya.

Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Ini tidak perlu dilakukan jika Anda yakin plot Anda berhasil.

```

    reset; starplot1(normal(1,10)+5,color=red,lab=1:10):

```

```

    ![images/Plot2DNandhita

```

Terkadang, Anda mungkin ingin merencanakan sesuatu yang plot2d tidak bisa lakukan, tapi hampir.

Dalam fungsi berikut, kita membuat plot impuls logaritmik. plot2d dapat melakukan plot logaritmik, tetapi tidak untuk batang impuls.

```

    function logimpulseplot1 (x,y) ...
        x0,y0=makeimpulse(x,log(y)/log(10)); plot2d(x0,y0,i,bar,grid=0); h=holding(1);
        frame(); xgrid(ticks(x)); p=plot(); for i=-10 to 10; if ij=p[4] and ij=p[3] then
        ygrid(i,yt="10"+i); endif; end; holding(h); endfunction < /pre > Marikitauidengannilaiyangterdistribusi
        aspect(1.5); x=1:10; y=-log(random(size(x)))200; ... logimpulseplot1(x,y):
    ![images/Plot2DNandhita

```

Mari kita menganimasikan kurva 2D menggunakan plot langsung. Perintah plot(x,y) hanya memplot kurva ke dalam jendela plot. setplot(a,b,c,d) menyeting jendela ini.

Fungsi wait(0) memaksa plot muncul di jendela grafis. Jika tidak, pengundian ulang akan dilakukan 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; en-
        dif; f=f+0.02; end; framecolor(c); linewidth(l); endfunction i/pre;

```

Tekan tombol apa saja untuk menghentikan animasi ini.

```

    animliss(2,3); // lihat hasilnya, jika sudah puas, tekan ENTER

```

Plot Logaritmik

EMT menggunakan parameter "logplot" untuk skala logaritmik.

Plot logaritma dapat diplot 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: y-logaritma - logplot=2: x-y-logaritma - logplot=3: x-logaritma
plot2d("exp(x3 - x)x2", 1, 5, logplot = 1) :

```

```

    ![images/Plot2DNandhita

```

$\exp(x^3 - x)x^2$: fungsi yang akan diplot, dimana e adalah basis logaritma natural. Fungsi ini menghitung nilai x dan kemudian mengalikannya dengan x^2

1, 5: Ini menunjukkan bahwa grafik akan diplot untuk nilai x dari 1 hingga 5.

logplot=1: Ini berarti bahwa sumbu y akan menggunakan skala logaritmik, sementara sumbu x tetap dalam skala linear.

```
plot2d("exp(x+sin(x))",0,100,logplot=1):
```

!images/Plot2D_{Nandhita}

$\exp(x + \sin(x))$: fungsi eksponensial yang akan diplot, di mana e adalah basis logaritma natural. Fungsi ini menghitung nilai eksponensial dari $x + \sin(x)$.

0, 100: Ini menunjukkan bahwa grafik akan diplot untuk nilai x dari 0 hingga 100.

logplot=1: Ini berarti bahwa sumbu y akan menggunakan skala logaritmik, sementara sumbu x tetap dalam skala linear.

```
plot2d("exp(x+sin(x))",10,100,logplot=2):
```

!images/Plot2D_{Nandhita}

Perintah ini akan menghasilkan grafik yang menunjukkan perilaku fungsi eksponensial dalam rentang yang ditentukan, dengan penekanan pada pertumbuhan fungsi yang cepat karena eksponensial, sambil menggunakan skala logaritmik pada sumbu x .

```
plot2d("gamma(x)",1,10,logplot=1):
```

!images/Plot2D_{Nandhita}

Perintah ini akan menghasilkan grafik yang menunjukkan perilaku fungsi γ di atas rentang yang ditentukan, dengan penekanan pada nilai-nilai yang lebih besar dengan menggunakan skala logaritmik untuk sumbu y .

```
plot2d("log(x(2+sin(x/100)))",10,1000,logplot=3):
```

!images/Plot2D_{Nandhita}

Perintah ini akan menghasilkan grafik yang menunjukkan perilaku fungsi logaritma di atas rentang yang ditentukan, untuk melihat pola dan pertumbuhan fungsi dengan lebih jelas.

Ini juga berfungsi dengan plot data.

```
x=10(1 : 20); y = x^2 - x;
```

```
plot2d(x,y,logplot=2):
```

!images/Plot2D_{Nandhita}

Perintah ini akan menghasilkan grafik yang menunjukkan hubungan antara x dan y dengan sumbu x dalam skala logaritmik, untuk melihat pola pertumbuhan fungsi kuadrat dengan lebih jelas.

Latihan soal

1. buatlah plot 2d dari fungsi

$$y = a \cdot \frac{x^2}{a}$$

dengan $a=4$ dan interval dari $x=-1$ sampai $x=1$. lalu, buatlah plot tersebut dalam grid 3 dan ketebalan 3!.

```
a=4; expr = exp(a(x^2)/a);
```

```
plot2d(expr,-1,1,grid=3,thickness=3):
```

!images/Plot2D_{Nandhita}

2. buatlah plot 2d dari fungsi

$$y(x, a) = x^2 - a \cdot x$$

dengan nilai parameternya adalah 5, dengan interval x nya dari 1 sampai 2.

```
function f(x,a):=x2 - a.x
```

```
a=5; plot2d("f",1,2;a):
```

```
!images/Plot2DNandhita
```

3.selidiki dimanakah fungsi f(x) berikut naik dan turun

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

untuk interval x = -4 sampai x = 5

```
function f(x):= (1/3)x3 - x2 - 3x + 4
```

```
plot2d ("f",-4,5):
```

```
!images/Plot2DNandhita
```

Rujukan Lengkap Fungsi plot2d()

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

```
"", "lt;gt;", ".", "..", "...",
"*", "+", "_", "-", "o"
```

(filled shapes)

```
"", "lt;gt;", "o" (filled shapes)
"w", "lt;gt;w", "ow" (non-transparent)
```

For lines use

```
-, "_", "-.", ".", "-.-", "-.-", "-gt;"
```

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 gt;0, there will be more space to the left for labels.

vertical :

Turns vertical labels on or off. This changes the global variable `vertical` 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 `gt;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 `[-2,2]` 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 `lt;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 `lt;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 in strings must always be expressions in the parameter variable x.

4 plot 3d

Plot3D_{NandhitaPutriShalsabila}Nama : NandhitaPutriShalsabila

Nim : 23030630075

Kelas : Matematika B 2023

Menggambar Plot 3D dengan EMT

Ini adalah pengenalan plot 3D di Euler. Kita memerlukan plot 3D untuk memvisualisasikan fungsi dua variabel.

Euler menggambar fungsi-fungsi tersebut dengan menggunakan algoritme pengurutan untuk menyembunyikan bagian-bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Standarnya adalah dari kuadran x-y positif ke arah asal $x=y=z=0$, tetapi sudut $=0^\circ$ terlihat dari arah sumbu-y. Sudut pandang dan ketinggian dapat diubah.

Euler dapat memetakan

- permukaan dengan bayangan dan garis-garis level atau rentang level,
- awan titik-titik,
- kurva parametrik,
- permukaan implisit.

Plot 3D dari sebuah 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("x² + sin(y)", -5, 5, 0, 6pi) :

![images/Plot3D_{Nandhita}

aspect(1.5);

untuk mengatur rasio aspek dari plot. Dalam hal ini, aspek ditetapkan menjadi 1.5, yang berarti rasio antara sumbu-sumbu grafik akan disesuaikan agar proporsional.

plot3d("x² + sin(y)", -5, 5, 0, 6*pi); untuk membuat grafik 3D dari fungsi $z = x^2 + \sin(y)$, dimana :

$x^2 + \sin(y)$ adalah fungsi yang dipetakan ke sumbu z berdasarkan nilai x dan y.

-5 dan 5 adalah batas-batas untuk nilai x (dari -5 sampai 5).

0 dan 6*pi adalah batas-batas untuk nilai y (dari 0 hingga 6pi).

plot3d("x² + x sin(y)", -5, 5, 0, 6pi) :

![images/Plot3D_{Nandhita}

plot3d("x² + x sin(y)", -5, 5, 0, 6*pi); untuk menghasilkan grafik 3D dari fungsi $z = x^2 + x \sin(y)$, dengan :

Fungsi $x^2 + x \sin(y)$ yang dipetakan ke sumbu z.

Batas nilai x dari -5 hingga 5.

Batas nilai y dari 0 hingga 6pi.

Latihan Soal

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.

1. talang bergelombang melengkung secara mendatar (koordinat polar)

```
plot3d("x^2 + xsin(y)", 0, 5, 0, 2pi) :
```

```
!images/Plot3DNandhita
```

```
plot3d("x^2 + xsin(y)", 0, 5, 0, 4pi) :
```

```
!images/Plot3DNandhita
```

Fungsi dari dua Variabel

Untuk grafik sebuah fungsi, gunakan

-ekspresi sederhana dalam x dan y,

-nama fungsi dari dua variabel

-atau matriks data.

Standarnya adalah kisi-kisi kawat yang terisi dengan warna yang berbeda di kedua sisi. Perhatikan bahwa jumlah default interval grid adalah 10, namun plot menggunakan jumlah default 40x40 persegi panjang untuk membangun permukaan. Hal ini dapat diubah.

-n=40, n=[40,40]: jumlah garis kisi di setiap arah

-grid=10, grid=[10,10]: jumlah garis grid di setiap arah.

Kami menggunakan default n=40 dan grid=10.

```
plot3d("x^2 + y^2") :
```

```
!images/Plot3DNandhita
```

Fungsi $z=x^2+y^2$: *fungsi kuadrat yang menghasilkan permukaan berbentuk paraboloid. Nilai z akan meningkat*

Plot3d: menghasilkan grafik 3D dari fungsi yang diberikan dalam rentang default untuk x dan y (biasanya dari -5 hingga 5 jika tidak ditentukan).

Interaksi pengguna dapat dilakukan dengan `gt;user` Parameter. Pengguna dapat menekan tombol berikut ini.

* kiri, kanan, atas, bawah: memutar sudut pandang

* +,-: memperbesar atau memperkecil

* a: menghasilkan anaglyph (lihat di bawah)

* l: beralih memutar sumber cahaya (lihat di bawah)

* spasi: mengatur ulang ke default

* kembali: mengakhiri interaksi

```
plot3d("exp(-x^2+y^2)", user,...title = "Turn with the vector keys (press return to finish)") :
```

```
!images/Plot3DNandhita
```

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: menskalakan ke nilai fungsi (defaultnya adalah `lt;fscale`).

skala: angka atau vektor 1x2 untuk menskalakan ke arah x dan y.

bingkai: jenis bingkai (default 1).

```
plot3d("exp(-(x2+y2)/5)", r = 10, n = 80, fscale = 4, scale = 1.2, frame = 3, user) :
```

!images/Plot3D_Nandhita

Perintah ini menghasilkan grafik 3D dari fungsi Gaussian berbentuk lonceng dengan distribusi yang diperlebar, rentang dari -10 hingga 10 pada sumbu x dan y, dan dengan resolusi tinggi (80 titik per sumbu). Grafik ini juga diskalakan pada sumbu vertikal (fscale 4) dan keseluruhan grafik diperbesar 1.2 kali.

Tampilan dapat diubah dengan berbagai cara.

- * Jarak: jarak pandang ke plot.
- * zoom: nilai zoom.
- * sudut: sudut terhadap sumbu y negatif dalam radian.
- * tinggi: ketinggian pandangan dalam radian.

Nilai default dapat diperiksa atau diubah dengan fungsi view(). Ini mengembalikan parameter dalam urutan di atas.

```
view
[5, 2.6, 2, 0.4]
```

Jarak yang lebih dekat membutuhkan lebih sedikit zoom. Efeknya lebih seperti lensa sudut lebar.

Pada contoh berikut, sudut=0 dan tinggi=0 dilihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

```
plot3d("x2 + y", distance = 3, zoom = 1, angle = pi/2, height = 0) :
```

!images/Plot3D_Nandhita

Perintah ini menghasilkan grafik 3D dari fungsi $z=x^2+y$ dengan sudut pandang khusus : Kamera berada pada jarak 3 unit dari grafik. Grafik ditampilkan tanpa pembesaran (zoom 1). Kamera diputar 90 derajat (sudut $\pi/2$), sehingga tampilan grafik lebih dari samping.

Kamera ditempatkan sejajar dengan bidang horizontal (ketinggian 0).

Plot selalu terlihat berada di tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter tengah.

```
plot3d("x4 + y2", a = 0, b = 1, c = -1, d = 1, angle = -20, height = 20, ... center = [0.4, 0, 0], zoom = 5) :
```

!images/Plot3D_Nandhita

Perintah ini menghasilkan grafik 3D dari fungsi $z=x^4+y^2$ dengan aturan sebagai berikut : Rentang nilai x dari 0 hingga 1 dan nilai y dari -1 hingga 1. Sudut pandang sedikit diputar (-20 derajat) dan kamera diposisikan 20 derajat di atas bidang horizontal.

Grafik berpusat di titik (0.4, 0, 0).

Grafik diperbesar lima kali dari ukuran normal.

Plotnya diskalakan agar sesuai dengan unit kubus untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung ukuran plot. Namun labelnya mengacu pada ukuran sebenarnya.

Jika Anda memamatikannya dengan `scale=false`, Anda harus berhati-hati agar plot tetap masuk ke dalam jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan bagian tengah.

```
plot3d("5exp(-x^2 - y^2)", r = 2, < fscale, < scale, distance = 13, height =
50, ... center = [0, 0, -2], frame = 3) :
```

!images/Plot3D_Nandhita

Perintah ini menghasilkan grafik 3D dari fungsi $z=5 \cdot \exp(-x^2 - y^2)$ dengan aturan sebagai berikut :

Rentang nilai x dan y dari -2 hingga 2.

Kamera diposisikan pada jarak 13 unit dari grafik dan pada ketinggian 50 derajat, memberikan sudut pandang yang baik.

Grafik berpusat di titik (0, 0, -2).

Skala fungsi dan skala keseluruhan grafik menggunakan nilai default.

Tampilan dengan kerangka atau grid 3D yang membantu visualisasi.

Plot kutub juga tersedia. Parameter `polar=true` menggambar plot kutub. Fungsi tersebut harus tetap merupakan fungsi dari x dan y. Parameter "`fscale`" menskalakan fungsi dengan skalanya sendiri. Kalau tidak, fungsinya 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) :
```

!images/Plot3D_Nandhita

Perintah ini menghasilkan grafik 3D dari fungsi

$z=1/(x^2 + y^2 + 1)$ dengan aturan sebagai berikut :

Rentang nilai x dan y dari -5 hingga 5.

Grafik ditampilkan dalam koordinat polar.

Skala vertikal diperbesar dua kali (`fscale=2`).

Warna permukaan berdasarkan hue (nuansa).

100 titik digunakan untuk resolusi yang lebih halus.

Grafik diperbesar empat kali dari ukuran normal.

Garis kontur juga ditampilkan pada grafik.

Warna grafik ditetapkan ke biru.

```
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) :
```

!images/Plot3D_Nandhita

Perintah ini menghasilkan grafik 3D dari fungsi

$f(r)=e^{-r/2} \cdot \cos(r)$ dengan aturan sebagai berikut :

Fungsi z ditentukan dengan menggunakan $r=x^2 + y^2$

Grafik ditampilkan dalam koordinat polar, sehingga lebih intuitif untuk fungsi radial.

Skala pada sumbu z dikurangi menjadi 0.4 untuk membuat tampilan lebih datar.

Rentang nilai r dibatasi hingga pi.

Tampilan dengan kerangka 3D yang lebih terperinci.

Grafik diperbesar empat kali untuk detail yang lebih baik.

Parameter memutar memutar fungsi di x di sekitar sumbu x.

* putar=1: Menggunakan sumbu x

* putar=2: Menggunakan sumbu z

plot3d("x² + 1", a = -1, b = 1, rotate = true, grid = 5) :
 ![images/Plot3D_Nandhita
 plot3d: digunakan untuk menggambar grafik 3D.
 "x² + 1" : *ekspresi fungsi matematika yang akan digambar, yaitu $y = x^2 + 1$. Fungsi ini menggambarkan parabola yang digeser satu unit ke atas.*
 a=-1, b=1: batas rentang x. Dalam hal ini, x dibatasi dari -1 hingga 1.
 rotate=true: Opsi ini memungkinkan grafik berputar sehingga pengguna bisa melihatnya dari berbagai sudut.

Grid=5: mengatur jumlah grid atau partisi pada sumbu untuk membantu visualisasi. Nilai 5 menunjukkan grafik akan dibagi menjadi 5 bagian pada setiap sumbu.

plot3d("x² + 1", a = -1, b = 1, rotate = 2, grid = 5) :
 ![images/Plot3D_Nandhita
 plot3d: digunakan untuk menggambar grafik 3D.
 "x² + 1" : *ekspresi fungsi matematika yang akan digambar, yaitu $y = x^2 + 1$, sebuah parabola yang digeser satu unit ke atas.*

a=-1, b=1: rentang untuk variabel x, yang dibatasi dari -1 hingga 1.
 rotate=2: Nilai rotate=2 menentukan rotasi awal grafik 3D pada tampilan. Berbeda dengan rotate=true yang memungkinkan rotasi interaktif, rotate=2 merujuk pada sudut atau posisi tetap rotasi tertentu yang telah ditentukan.

grid=5: Mengatur jumlah grid atau partisi pada sumbu untuk membantu visualisasi. Grid akan dibagi menjadi 5 bagian per sumbu, memberikan tampilan grafik dengan detail yang lebih sederhana.

plot3d("sqrt(25-x²)", a = 0, b = 5, rotate = 1) :
 ![images/Plot3D_Nandhita
 plot3d: untuk menggambar grafik 3D.
 sqrt(25-x²) : *fungsi matematika yang digambarkan, yaitu $y = \sqrt{25-x^2}$. Fungsi ini adalah setengah lingkaran jari 5 pada bidang xy, di definisikan hanya untuk x dari -5 sampai 5 karena nilai dalam karkuadratnya positif.*

a=0, b=5: menentukan rentang x dari 0 hingga 5. Karena fungsi ini merepresentasikan setengah lingkaran, rentang x hanya setengah dari lingkaran tersebut (bagian positif).

rotate=1: menentukan rotasi awal grafik 3D, memberikan sudut pandang tertentu pada grafik saat ditampilkan.

plot3d("xsin(x)", a=0, b=6pi, rotate=2):
 ![images/Plot3D_Nandhita
 plot3d: digunakan untuk menggambar grafik 3D.
 "x*sin(x)": *ekspresi fungsi matematika yang akan digambar, yaitu $y = x \sin(x)$. Fungsi ini menghasilkan grafik gelombang sinus yang amplitudonya berubah seiring dengan bertambahnya nilai x.*

a=0, b=6*pi: rentang untuk variabel x. Nilai x diatur dari 0 hingga 6pi (sekitar 18.85), yang mencakup beberapa siklus dari fungsi sinus.

rotate=2: Parameter ini menentukan rotasi awal grafik dalam tampilan 3D. Nilai rotate=2 berarti grafik akan dimulai dengan sudut atau posisi rotasi tertentu yang telah ditetapkan.

Berikut adalah plot dengan tiga fungsi.

plot3d("x", "x² + y²", "y", r = 2, zoom = 3.5, frame = 3) :

!images/Plot3D_{Nandhita}
 plot3d: digunakan untuk menggambar grafik 3D.
 "x", "x²+y²", "y" : *tiga ekspresi fungsi matematikayang merepresentasikan tiga dimensi (sumbu x, z, dan y)*
 x berfungsi sebagai sumbu x,
 x²+y² *menggambarkan ketinggian (sumbu z, biasanya terkait dengan fungsi atau permukaan),*
 y merepresentasikan sumbu y.
 Secara keseluruhan, ini menggambarkan permukaan 3D yang berbentuk paraboloid (karena $x^2 + y^2$ adalah persamaan paraboloid di ruang 3D).
 r=2: parameter untuk mengatur radius dari grid yang dipetakan dalam grafik, sehingga grafik akan diplot dalam rentang dengan jari-jari 2 unit.
 zoom=3.5: mengatur tingkat pembesaran (zoom) pada grafik. Nilai ini memperbesar atau memperkecil tampilan grafik agar lebih jelas.
 frame=3: Parameter ini menentukan tipe bingkai yang akan digunakan untuk grafik. Nilai frame=3 mungkin menunjukkan pengaturan tertentu untuk bingkai grafik, seperti bagaimana sumbu-sumbu ditampilkan atau bagaimana grid ditata di sekitar grafik.
 Plot Kontur
 Untuk plotnya, Euler menambahkan garis grid. Sebaliknya, dimungkinkan untuk menggunakan garis datar dan rona satu warna atau rona warna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan arsiran. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.
 -gt;hue: Mengaktifkan bayangan cahaya, bukan kabel.
 -gt;kontur: Membuat plot garis kontur otomatis pada plot.
 - level=... (atau level): Vektor nilai garis kontur.
 Standarnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level-level tersebut sebenarnya adalah rentang level.
 Gaya default dapat diubah. Untuk plot kontur berikut, kami menggunakan grid yang lebih halus dengan ukuran 100x100 titik, menskalakan fungsi dan plot, dan menggunakan sudut pandang yang berbeda.
 plot3d("exp(-x²-y²)", r = 2, n = 100, level = "thin", ...contour, spectral, fscale = 1, scale = 1.1, angle = 45, height = 20) :
 !images/Plot3D_{Nandhita}
 plot3d: digunakan untuk menggambar grafik 3D.
 "exp(-x²-y²)" : *ekspresi fungsi matematikayang akan digambar. Fungsi ini adalah fungsi Gaussian 2D (fungsi 2D) dan menurun dengan cepat menuju nol saat x²+y² bertambah. Grafik ini menggambarkan permukaannya*
 r=2: mengatur rentang x dan y dari -2 hingga 2. Grafik akan diplot dalam lingkup jari-jari 2 unit pada sumbu x dan y.
 n=100: Ini menentukan resolusi grid, dengan 100 titik di sepanjang setiap sumbu. Semakin tinggi nilainya, semakin halus grafik yang dihasilkan.
 level="thin": Ini mengatur tampilan level (kontur) yang lebih tipis pada grafik, sehingga garis kontur yang dihasilkan lebih halus dan lebih presisi.
 gt;contour: menambahkan garis kontur pada grafik 3D. Garis-garis kontur digunakan untuk menunjukkan nilai-nilai yang sama pada permukaan grafik, memberikan informasi tambahan tentang bentuk fungsi pada bidang tertentu.

gt;spectral: mengatur spektrum warna untuk grafik, yang berarti warna yang digunakan akan didasarkan pada skala spektral, biasanya warna pelangi atau gradasi warna lainnya.

fscale=1: mengatur faktor skala untuk kontur dan grafik ke nilai standar (tanpa pembesaran atau pengecilan tambahan).

scale=1.1: memperbesar grafik secara keseluruhan sebesar 1.1 kali lipat, sehingga grafik terlihat sedikit lebih besar dari ukuran aslinya.

angle=45°: mengatur sudut rotasi tampilan grafik 3D sebesar 45 derajat di sekitar sumbu vertikal (sumbu z), sehingga grafik akan terlihat dari sudut miring.

height=20°: Ini mengatur sudut elevasi atau ketinggian dari mana grafik dilihat. Dalam hal ini, grafik akan dilihat dari sudut 20 derajat di atas bidang horizontal.

```
plot3d("exp(xy)",angle=100°,contour,color=green):
```

```
!images/Plot3DNandhita
```

plot3d: digunakan untuk menggambar grafik 3D.

"exp(x*y)": ekspresi fungsi matematika yang akan digambar, yaitu

$z = \exp(x,y)$. Fungsi ini eksponensial di mana variabel

x dan y saling dikalikan. Grafik ini akan menunjukkan peningkatan yang cepat di satu arah dan penurunan di arah yang lain, membentuk permukaan yang cukup curam di beberapa area.

angle=100°: mengatur sudut rotasi tampilan grafik 3D sebesar 100 derajat di sekitar sumbu vertikal (z). Grafik akan diputar untuk dilihat dari sudut ini, memungkinkan tampilan yang berbeda dari grafik.

gt;contour: Perintah ini menambahkan garis kontur pada grafik 3D. Garis kontur digunakan untuk menunjukkan nilai-nilai yang sama pada permukaan grafik, yang membantu dalam memahami variasi elevasi atau perubahan fungsi dalam bidang tertentu.

color=green: Ini mengatur warna grafik menjadi hijau. Grafik dan kontur akan ditampilkan menggunakan skema warna hijau.

Bayangan defaultnya menggunakan warna abu-abu. Namun rentang warna spektral juga tersedia.

-gt;spectral: Menggunakan skema spektral default

- color=...: Menggunakan warna khusus atau skema spektral

Untuk plot berikut, kami menggunakan skema spektral default dan menambah jumlah titik untuk mendapatkan tampilan yang sangat mulus.

```
plot3d("x2 + y2", spectral, contour, n = 100) :
```

```
!images/Plot3DNandhita
```

Selain garis level otomatis, kita juga dapat menetapkan nilai garis level. Ini akan menghasilkan garis level yang tipis, bukan rentang level.

```
plot3d("x2 - y2", 0, 5, 0, 5, level = -1 : 0.1 : 1, color = redgreen) :
```

```
!images/Plot3DNandhita
```

Dalam plot berikut, kita menggunakan dua pita tingkat yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas tingkat sebagai kolom.

Selain itu, kami melapisi grid dengan 10 interval di setiap arah.

```
plot3d("x2 + y3", level = [-0.1, 0.9; 0, 1], ... spectral, angle = 30, grid = 10, contourcolor = gray) :
```

```
!images/Plot3DNandhita
```

Pada contoh berikut, kita memplot himpunan, di mana

Kami menggunakan satu garis tipis untuk garis level.

```
plot3d("xy - yx", level = 0, a = 0, b = 6, c = 0, d = 6, contourcolor = red, n = 100) :
```

```
!images/Plot3DNandhita
```

Dimungkinkan untuk menampilkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

```
plot3d("x2 + y4", cp, cpcolor = green, cpdelta = 0.2) :
```

```
!images/Plot3DNandhita
```

Berikut beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi.

```
figure(2,2); ... expr="y3-x2"; ...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) :
```

```
!images/Plot3DNandhita
```

Ada beberapa skema spektral lainnya, yang diberi nomor dari 1 hingga 9.

Namun Anda juga dapat menggunakan warna=nilai, di mana nilai

- * spektral: untuk rentang dari biru hingga merah

- * putih: untuk rentang yang lebih redup

- * kuningbiru, unguhijau, birukuning, hijaumerah

- * birukuning, hijauungu, kuningbiru, merahhijau

```
figure(3,3); ... for i=1:9; ... figure(i); plot3d("x2 + y2", spectral = i, contour, cp, < frame, zoom = 4); ... end; ...figure(0) :
```

```
!images/Plot3DNandhita
```

Sumber cahaya dapat diubah dengan l dan tombol kursor selama interaksi pengguna. Itu juga dapat diatur dengan parameter.

- * cahaya : arah datangnya cahaya

- * amb: cahaya sekitar antara 0 dan 1

Perhatikan bahwa program ini tidak membuat perbedaan antara sisi plot.

Tidak ada bayangan. Untuk ini, Anda memerlukan Povray.

```
plot3d("-x2-y2", ...hue = true, light = [0, 1, 1], amb = 0, user = true, ...title = "Presslandcursorkeys(returntoexit)") :
```

```
!images/Plot3DNandhita
```

Parameter warna mengubah warna permukaan. Warna garis level juga bisa diubah.

```
plot3d("-x2-y2", color = rgb(0.2, 0.2, 0), hue = true, frame = false, ...zoom = 3, contourcolor = red, level = -2 : 0.1 : 1, dl = 0.01) :
```

```
!images/Plot3DNandhita
```

Warna 0 memberikan efek pelangi yang istimewa.

```
plot3d("x2/(x2 + y2 + 1)", color = 0, hue = true, grid = 10) :
```

```
!images/Plot3DNandhita
```

Permukaannya juga bisa transparan.

```
plot3d("x2 + y2", transparent, grid = 10, wirecolor = red) :
```

```
!images/Plot3DNandhita
```

Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan pemotongan melalui objek. Fitur plot3d mencakup plot implisit. Plot ini menunjukkan himpunan nol suatu fungsi dalam tiga variabel.

Solusi dari

dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y-, x-z- dan y-z.

* implisit=1: dipotong sejajar bidang y-z

* implisit=2: dipotong sejajar dengan bidang x-z

* implisit=4: dipotong sejajar bidang x-y

Tambahkan nilai-nilai ini, jika Anda mau. Dalam contoh kita memplot

```
plot3d("x2 + y3 + zy - 1", r = 5, implicit = 3) :
```

```
!images/Plot3DNandhita
```

```
c=1; d=1;
```

```
plot3d("((x2 + y2 - c2)2 + (z2 - 1)2)((y2 + z2 - c2)2 + (x2 - 1)2((z2 + x2 - c2)2 + (y2 - 1)2) - d", r = 2, < frame, implicit, user) :
```

```
!images/Plot3DNandhita
```

Perintah ini menggambarkan grafik 3D dari persamaan implisit yang cukup kompleks, yang melibatkan fungsi-fungsi kuadrat dan konstanta c dan d. Rentang plot dibatasi dari -2 hingga 2 di setiap sumbu, dan grafik ini ditampilkan dengan bingkai (frame) sumbu koordinat. Persamaan ini menghasilkan bentuk geometris yang rumit dalam ruang 3D yang mungkin menyerupai permukaan non-linear.

```
plot3d("x2 + y2 + 4xz + z3", implicit, r = 2, zoom = 2.5) :
```

```
!images/Plot3DNandhita
```

Perintah ini menggambarkan grafik tiga dimensi dari persamaan implisit $x^2 + y^2 + 4xz + z^3$, dimana grafik diplot dalam rentang dari -2 hingga 2 pada setiap sumbu, dengan tingkat pembesaran linear yang kompleks dalam ruang 3D sesuai dengan persamaan tersebut.

Latihan soal

1. gambarlah fungsi implisit berikut dalam 3D

```
plot3d("x2 + y2 - z2 - 1", r = 8, implicit = 3) :
```

```
!images/Plot3DNandhita
```

2. gambarlah fungsi 3D dari fungsi implisit berikut ini

dengan r=4

```
plot3d("xy + x3y2 + xz3 - 9", r = 4, implicit = 3) :
```

```
!images/Plot3DNandhita
```

Merencanakan 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).

Karena x,y,z adalah matriks, kita asumsikan bahwa (t,s) melewati grid persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di ruang angkasa.

Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

Dalam contoh berikut, kita menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar kita dapat menandai wilayah, dalam kasus kita wilayah 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,ifframe,grid=[10,20], ...
values=s,contourcolor=red,level=[90°-24°;90°-22°], ... scale=1.4,height=50°):
```

!images/Plot3D_{Nandhita}

Berikut ini contohnya yaitu grafik suatu fungsi.

```
t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,ts,grid=10):
```

!images/Plot3D_{Nandhita}

Namun, kita bisa membuat berbagai macam permukaan. Berikut adalah permukaan yang sama sebagai suatu fungsi

```
plot3d(ts,t,s,angle=180°,grid=10):
```

!images/Plot3D_{Nandhita}

Dengan lebih banyak usaha, kita dapat menghasilkan banyak permukaan.

Dalam contoh berikut kita membuat tampilan bayangan dari bola yang terdistorsi. Koordinat bola yang biasa adalah dengan

Kami mendistorsi ini dengan sebuah faktor

```
t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ... d=1+0.2(cos(4 *
t)+cos(8s)); ... plot3d(cos(t)cos(s)d,sin(t)cos(s)d,sin(s)d,hue=1, ... light=[1,0,1],frame=0,zoom=5):
```

!images/Plot3D_{Nandhita}

Tentu saja, point cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita memerlukan tiga vektor untuk koordinat titik-titik tersebut.

Gayanya sama seperti di plot2d dengan points=true;

```
n=500; ... plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=".");
```

!images/Plot3D_{Nandhita}

Dimungkinkan juga untuk memplot kurva dalam 3D. Dalam hal ini, lebih mudah untuk menghitung terlebih dahulu titik-titik kurva. Untuk kurva pada bidang kita menggunakan barisan koordinat dan parameter wire=true.

```
t=linspace(0,8pi,500); ... plot3d(sin(t),cos(t),t/10, wire,zoom=3):
```

!images/Plot3D_{Nandhita}

Perintah ini menggambar spiral 3D menggunakan fungsi sinus dan cosinus dengan variasi di sumbu x , y , dan z . Grafiknya diplot sebagai kurva wireframe (tanpa permukaan) dengan zoom sebesar 3 kali. Ini menghasilkan grafik yang menyerupai heliks atau spiral, karena nilai $\sin(t)$ dan $\cos(t)$ membentuk lingkaran dalam ruang x - y , sementara $t/10$ secara linear meningkat pada sumbu z .

```
t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi, wire, ... linewidth=3,wirecolor=blue):
```

!images/Plot3D_{Nandhita}

Perintah ini menggambar spiral 3D dalam ruang menggunakan fungsi cosinus dan sinus untuk menghasilkan kurva melingkar dalam bidang x - y , sementara variabel $t/2\pi$ memberikan pergerakan linier pada sumbu z . Grafik diplot sebagai kurva wireframe dengan ketebalan garis 3 dan warna biru, sehingga menghasilkan spiral biru yang tebal dalam ruang 3D.

```
X=cumsum(normal(3,100)); ... plot3d(X[1],X[2],X[3], anaglyph, wire):
```

!images/Plot3D_{Nandhita}

Perintah ini menggambar grafik 3D dari jalur acak yang dihasilkan berdasarkan distribusi normal dalam ruang 3D, diplot sebagai kurva wireframe (hanya garis) dengan efek anaglyph untuk memberikan pengalaman visual 3D. Jalur ini dihasilkan melalui jumlah kumulatif dari vektor-vektor acak, yang menghasilkan jalur yang terus terhubung tetapi acak.

EMT juga dapat membuat plot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda memerlukan kacamata berwarna merah/sian.

```
plot3d("x2 + y3", anaglyph, contour, angle = 30) :
```

!images/Plot3D_{Nandhita}

Perintah ini digunakan untuk menggambar grafik 3D dari fungsi $x^2 + y^3$, dengan tampilan dalam mode anaglyph garis kontur.

Seringkali skema warna spektral digunakan untuk plot. Ini menekankan ketinggian fungsinya.

```
plot3d("x2y3 - y", spectral, contour, zoom = 3.2) :
```

!images/Plot3D_{Nandhita}

Perintah ini menggambar grafik 3D dari fungsi $x^2 * y^3 - y$ dalam ruang tiga dimensi, dengan tampilan warna spektral linear yang ditampilkan secara visual dengan warna dan kontur untuk menunjukkan perbedaan ketinggian di permukaan.

Euler juga dapat memplot permukaan yang diparameterisasi, jika parameter-nya adalah nilai x, y, dan z dari gambar kotak persegi panjang di ruang tersebut.

Untuk demo berikut, kami menyiapkan parameter u- dan v-, dan menghasilkan koordinat ruang dari parameter tersebut.

```
u=linspace(-1,1,10); v=linspace(0,2pi,50)'; ... X=(3+ucos(v/2))cos(v);
Y=(3+ucos(v/2))sin(v); Z=usin(v/2); ...
```

Perintah ini menghasilkan titik-titik dalam ruang 3D yang digunakan untuk membentuk permukaan kompleks. Parameter u dan v digunakan untuk mendefinisikan koordinat X, Y, dan Z berdasarkan fungsi trigonometrik, yang memungkinkan penciptaan bentuk yang menarik. Setelah ini, Anda dapat menggunakan fungsi plot3d untuk memvisualisasikan permukaan ini. Permukaan yang dihasilkan akan memiliki bentuk yang menarik dan berputar, mungkin menyerupai struktur seperti torus atau permukaan kompleks lainnya tergantung pada rumus yang digunakan.

```
plot3d(X,Y,Z, anaglyph, iframe, wire, scale=2.3):
```

!images/Plot3D_{Nandhita}

Perintah ini digunakan untuk menggambar grafik 3D dari titik-titik yang telah dihitung sebelumnya menggunakan fungsi plot3d, dengan tampilan dalam mode anaglyph untuk efek 3D, menampilkan bingkai di sekitar grafik, dan menggunakan representasi wireframe untuk menyoroti struktur permukaan. Skala 2,3 kali memperbesar tampilan grafik, memungkinkan pengguna untuk melihat detail lebih dekat pada bentuk yang dihasilkan. Ini sangat berguna untuk menganalisis bentuk kompleks atau permukaan dalam ruang 3D.

Berikut adalah contoh yang lebih rumit, yang megah dengan kacamata merah/cyan.

```
u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)'; ... x:=(4*(1+.25sin(3 *
v))+cos(u))cos(2v); ... y:=(4*(1+.25sin(3v))+cos(u))sin(2v); ... z:=sin(u)+2 *
cos(3v); ... plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8, anaglyph):
```



```
!images/Plot3DNandhita
```

Perintah ini menghasilkan grafik 3D dari permukaan yang kompleks berdasarkan fungsi x , y , dan z yang telah ditentukan dengan parameter yang menciptakan bentuk menarik. Grafik tersebut diplot dalam mode anaglyph untuk efek 3D, tanpa bingkai, dengan skala dan zoom yang ditingkatkan, serta pencahayaan yang diatur untuk meningkatkan visualisasi. Hasilnya adalah tampilan 3D yang menarik dan kaya warna, memungkinkan analisis bentuk dan struktur permukaan yang dihasilkan oleh fungsi tersebut.

Plot Statistik

Plot batang juga dimungkinkan. Untuk itu, kita harus menyediakannya

* x : vektor baris dengan $n+1$ elemen

* y : vektor kolom dengan $n+1$ elemen

* z : matriks nilai $n \times n$.

z bisa lebih besar, tetapi hanya nilai $n \times n$ yang akan digunakan.

Dalam contoh ini, pertama-tama kita menghitung nilainya. Kemudian kita sesuaikan x dan y , sehingga vektor-vektornya berpusat pada 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) :
```

```
!images/Plot3DNandhita
```

Dimungkinkan untuk membagi plot suatu 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):
```

```
!images/Plot3DNandhita
```

Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke $[-1,1]$ dengan `skala(M)`, atau menskalakan matriks dengan `gt;zscale`. Hal ini dapat dikombinasikan dengan faktor penskalaan individual yang diterapkan sebagai tambahan.

```
i=1:20; j=i'; ... plot3d(ij^2+100normal(20,20), zscale, scale = [1, 1, 1.5], angle = -40, zoom = 1.8) :
```

```
!images/Plot3DNandhita
```

Perintah ini menghasilkan grafik 3D dari permukaan yang ditentukan oleh kombinasi variabel i dan j dengan penambahan noise acak dari distribusi normal. Grafik ini diplot dengan penyesuaian skala pada sumbu z untuk memberikan visualisasi yang lebih baik dari ketinggian, dengan sudut dan zoom yang diatur untuk meningkatkan tampilan keseluruhan. Hasilnya adalah permukaan yang kompleks dengan variasi acak, memberikan representasi visual yang menarik dan informatif.

```
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]):
```

```
!images/Plot3DNandhita
```

Perintah ini digunakan untuk menghasilkan matriks acak dari angka bulat, menghitung frekuensi kemunculan angka 1 hingga 6 dalam setiap baris dari matriks tersebut, dan kemudian memvisualisasikan frekuensi tersebut dalam bentuk grafik 3D. Hasil grafik ini akan memberikan pemahaman yang lebih baik tentang distribusi angka yang dihasilkan secara acak dalam matriks Z .

Permukaan Benda Putar
`plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3,...style="",color=red,<outline,...level=[-2;0],n=100):`

!images/Plot3D_{Nandhita}

Secara keseluruhan, perintah ini digunakan untuk menggambar grafik 2D dari fungsi implicit yang kompleks, dengan parameter untuk menentukan gaya garis, warna, level, dan jumlah titik yang digunakan untuk membentuk kurva. Hasilnya akan menjadi visualisasi yang jelas dan informatif tentang bentuk fungsi yang digambarkan, dengan detail pada dua level yang ditentukan.

$\text{ekspresi} = (x^2 + y^2 - 1)^3 - x^2 y^3$; `ekspresi`

$$(y^2 + x^2 - 1)^3 - x^2 y^3$$

Kami ingin memutar kurva hati di sekitar sumbu y. Inilah ungkapan yang mendefinisikan hati:

Selanjutnya kita atur

`function fr(r,a) = ekspresi with [x=rcos(a),y=rsin(a)] — trigreduce; fr(r,a)`

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2\sin a)r^5}{16}$$

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang menyelesaikan r, jika a diberikan. Dengan fungsi tersebut kita dapat memplot 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(rcos(t)sin(s),rcos(t)cos(s),r ×
sin(t), ... hue,i)frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):`

!images/Plot3D_{Nandhita}

Berikut ini adalah plot 3D dari gambar di atas yang diputar mengelilingi sumbu z. Kami mendefinisikan fungsi yang mendeskripsikan objek.

`function f(x,y,z) ...
r=x^2+y^2;return(r+z^2-1)^3-r*z^3;endfunction </pre> 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):`

!images/Plot3D_{Nandhita}

perintah ini memplot grafik 3D dengan

batas x dari 0 hingga 1.2,

batas y -1.2 hingga 1.2,

batas z dari -1.2 hingga 1.4,

dengan implisit 1

angel 30 derajat berarti sudut pandangnya 30 derajat,

grafik di zoom 2.5

Plot 3D Khusus

Fungsi `plot3d` bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, dimungkinkan untuk mendapatkan plot berbingkai dari objek apa pun yang Anda sukai.

Meskipun Euler bukan program 3D, ia dapat menggabungkan beberapa objek dasar. Kami mencoba memvisualisasikan paraboloid dan garis singgungnya.

```
function myplot ...
y=-1:0.01:1; x=(-1:0.01:1)'; plot3d(x,y,0.2*(x-0.1)/2,jscale,i,frame,i,hue, .. hues=0.5,i,contour,color=orange);
h=holding(1); plot3d(x,y,(x^2+y^2)/2, < scale, < frame, > contour, > hue); holding(h); endfunction <
/pre > Sekarang framedplot() menyediakan bingkai, dan mengatur tampilan.
framedplot("myplot",[-1,1,-1,1,0,1],height=0,angle=-30°, ... center=[0,0,-
0.7],zoom=3):
```

!images/Plot3D_Nandhita

Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d() menyetel jendela ke fullwindow() secara default, tetapi plotcontourplane() berasumsi demikian.

```
x=-1:0.02:1.1; y=x'; z=x^2 - y^4;
function myplot (x,y,z) ... i

```


```


```

!images/Plot3D_Nandhita

perintah ini mendefinisikan fungsi yang menggambar grafik kontur dan grafik 3D dari fungsi matematis $z=x^2-y^4$. Fungsi ini memungkinkan pengguna untuk dengan mudah menghasilkan visualisasi Animasi

Euler dapat menggunakan frame untuk melakukan pra-komputasi animasi.

Salah satu fungsi yang memanfaatkan teknik ini adalah memutar. Itu dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi ini memanggil addpage() untuk setiap plot baru. Akhirnya ia menganimasikan plotnya.

Silakan pelajari sumber rotasi untuk melihat lebih detail.

```
function testplot () := plot3d("x^2 + y^3"); ... rotate("testplot"); testplot() :
```

!images/Plot3D_Nandhita

Secara keseluruhan, perintah ini menghasilkan grafik 3D dari fungsi $z=x^2+y^3$ melalui fungsi testplot, kemudian memutar tampilan grafik tersebut untuk memberikan perspektif yang berbeda.

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" Povray ke jalur lingkungan, atau mengatur variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe". <http://www.povray.org/>, dan meletakkan sub-direktori "bin" Povray ke jalur lingkungan, atau mengatur variabel "defaultpovray" dengan jalur lengkap yang mengarah ke "pvengine.exe". <http://www.povray.org/>.

Antarmuka Povray Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk menguraikan file-file ini. Nama file default adalah current.pov, dan direktori default adalah eulerhome(), biasanya c:. Povray menghasilkan file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan file-file ini, gunakan povclear().

Fungsi pov3d memiliki semangat yang sama dengan plot3d. Ini dapat menghasilkan grafik fungsi $f(x,y)$, atau permukaan dengan koordinat X,Y,Z dalam matriks, termasuk garis level opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat adegan ke dalam notebook Euler.

Selain `pov3d()`, ada banyak fungsi yang menghasilkan objek Povray. Fungsi ini mengembalikan string, yang berisi kode Povray untuk objek. Untuk menggunakan fungsi ini, mulai file Povray dengan `povstart()`. Kemudian gunakan `writeln(...)` untuk menulis objek ke file adegan. Terakhir, akhiri file dengan `povend()`. Secara default, raytracer akan dimulai, dan PNG akan dimasukkan ke dalam notebook Euler.

Fungsi objek memiliki parameter yang disebut "tampilan", yang memerlukan string dengan kode Povray untuk tekstur dan penyelesaian objek. Fungsi `povlook()` dapat digunakan untuk menghasilkan string ini. Ini memiliki parameter untuk warna, transparansi, Phong Shading dll.

Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk vertikal ke atas, dan sumbu x,y,z di tangan kanan.

Anda perlu memuat file povray.

```
load povray;
```

Pastikan, direktori Povray bin ada di jalurnya. Jika tidak, edit variabel berikut sehingga berisi jalur ke povray yang dapat dieksekusi.

```
defaultpovray="C:
```

Program Files

POV-Ray

v3.7

bin

pvruntime.exe"

```
C:\Files-Ray\3.7.exe
```

Untuk kesan pertama, kami memplot fungsi sederhana. Perintah berikut menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk penelusuran sinar 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 batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK di jendela Povray untuk mengonfirmasi dialog pengaktifan Povray.

```
plot3d("x2 + y2", zoom = 2) :
```

```
!images/Plot3DNandhita
```

Secara keseluruhan, perintah ini digunakan untuk menghasilkan grafik 3D dari fungsi $Z = x^2 + y^2$ dengan tingkat zoom senilai 2.

```
pov3d("x2 + y2", zoom = 3);
```

```
!images/Plot3DNandhita
```

Secara keseluruhan, perintah ini digunakan untuk menghasilkan grafik 3D dari fungsi $z = x^2 + y^2$ dengan tingkat zoom senilai 3.

Kita dapat membuat fungsinya transparan dan menambahkan penyelesaian lainnya. Kita juga dapat menambahkan garis level ke plot fungsi.

```
pov3d("x2 + y3", axiscolor = red, angle = -45, anaglyph, ...look = povlook(cyan, 0.2), level = -1 : 0.5 : 1, zoom = 3.8);
```

```
!images/Plot3DNandhita
```

```
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);
```

```
!images/Plot3DNandhita
```

Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi secara manual.

Kita memplot himpunan titik pada bidang kompleks, dimana hasil kali jarak ke 1 dan -1 sama dengan 1.

Latihan Soal

```
pov3d("((x-1)^2 + y^2)((x + 1)^2 + y^2)/40", r = 1.5, ... angle = 120, level =
1/40, dlevel = 0.005, light = [-1, 1, 1], height = 10, n = 50, ... < fscale, zoom =
3.8);
```

```
!images/Plot3DNandhita
```

Merencanakan dengan Koordinat

Daripada menggunakan fungsi, kita bisa memplotnya dengan koordinat. Seperti di plot3d, kita memerlukan tiga matriks untuk mendefinisikan objek.

Dalam contoh ini kita memutar suatu fungsi di sekitar 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]);
```

```
!images/Plot3DNandhita
```

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 cocok dengan kubus satuan.

```
r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ... x=rcos(phi); y=rsin(phi);
z=exp(-5r)cos(8pi*r)/3; ... pov3d(x,y,z,zoom=6,axis=0,height=30°,add=povsphere([0.5,0,0.25],0.15,povlook(r
... w=500,h=300);
```

```
!images/Plot3DNandhita
```

Dengan metode peneduh canggih Povray, sangat sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya pada batas-batas dan dalam bayangan, triknya mungkin terlihat jelas.

Untuk ini, kita perlu menjumlahkan vektor normal di setiap titik matriks.

$$Z = x^2 y^3$$

$$\begin{matrix} 2 & 3 & x & y \end{matrix}$$

Persamaan permukaannya adalah $[x, y, Z]$. Kami menghitung dua turunan dari x dan y dan mengambil perkalian silangnya sebagai normal.

$$dx = \text{diff}([x, y, Z], x); dy = \text{diff}([x, y, Z], y);$$

Kami mendefinisikan normal sebagai produk silang dari turunan ini, dan mendefinisikan fungsi koordinat.

$$N = \text{crossproduct}(dx, dy); NX = N[1]; NY = N[2]; NZ = N[3]; N,$$

$$\begin{matrix} 3 & 2 & 2 \\ - & 2 & x & y \\ , & - & 3 & x & y \\ , & & & & 1 \end{matrix}$$

Kami hanya menggunakan 25 poin.

$$x=-1:0.5:1; y=x';$$

$$\text{pov3d}(x, y, Z(x, y), \text{angle}=10^\circ, \dots \quad xv=NX(x, y), yv=NY(x, y), zv=NZ(x, y), \text{jshadow});$$

!images/Plot3D_{Nandhita}

Berikut ini adalah simpul Trefoil yang dilakukan oleh A. Busser di Povray. Ada versi yang lebih baik dalam contoh ini.

ja href="ExamplesKnot.html";Trefoil Knot|a;

Untuk tampilan yang bagus dengan poin yang tidak terlalu banyak, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normalnya bagi kami. Pertama, tiga fungsi koordinat sebagai ekspresi simbolik.

```
X = ((4+sin(3y))+cos(x))cos(2y); ... Y = ((4+sin(3y))+cos(x))sin(2y);
... Z = sin(x)+2cos(3y);
```

Kemudian kedua vektor turunan ke x dan y.

```
dx = diff([X,Y,Z],x); dy = diff([X,Y,Z],y);
```

Sekarang normalnya, yaitu perkalian silang kedua turunannya.

```
dn = crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

```
x:=linspace(-
```

Vektor normal adalah evaluasi ekspresi simbolik dn[i] untuk i=1,2,3. Sintaknya adalah amp;"ekspresi"(parameter). Ini adalah 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));
```

!images/Plot3D_{Nandhita}

Kami juga dapat menghasilkan grid dalam 3D.

```
povstart(zoom=4); ... x=-1:0.5:1; r=1-(x+1)2/6; ... t = (0 : 30 : 360)'; y =
rcos(t); z = rsin(t); ...writeln(povgrid(x,y,z,d = 0.02,dballs = 0.05)); ...povend();
```

!images/Plot3D_{Nandhita}

Dengan povgrid(), kurva dimungkinkan.

```
povstart(center=[0,0,1],zoom=3.6); ... t=linspace(0,2,1000); r=exp(-t); ...
x=cos(2pi10t)r; y=sin(2pi10t)r; z=t; ... writeln(povgrid(x,y,z,povlook(red)));
... writeAxis(0,2,axis=3); ... povend();
```

!images/Plot3D_{Nandhita}

Objek Povray

Di atas, kami menggunakan pov3d untuk memplot permukaan. Antarmuka povray di Euler juga dapat menghasilkan objek Povray. Objek ini disimpan sebagai string di Euler, dan perlu ditulis ke file Povray.

Kami memulai output dengan povstart().

```
povstart(zoom=4);
```

Pertama kita mendefinisikan tiga silinder, dan menyimpannya dalam 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 tersebut berisi
kode Povray, yang tidak perlu kita pahami pada saat itu.
```

c2

```
cylinder lt;0,0,-1gt;; lt;0,0,1gt;; 1 texture pigment color rgb lt;0.941176,0.941176,0.392157gt;
finish ambient 0.2
```

Seperti yang Anda lihat, kami menambahkan tekstur pada objek dalam tiga warna berbeda.

Hal ini dilakukan oleh `povlook()`, yang mengembalikan 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 lt;0.101961,0.2,0.301961,0.1gt; finish ambient
0.5
```

Sekarang kita mendefinisikan objek persimpangan, dan menulis hasilnya ke file.

```
writeln(povintersection([c1,c2,c3]));
```

Persimpangan tiga silinder sulit untuk divisualisasikan jika Anda belum pernah melihatnya sebelumnya.

```
povend;
!images/Plot3DNandhita
```

Fungsi berikut menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan bagaimana Euler menangani objek Povray sederhana. Fungsi `povbox()` mengembalikan 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) ... i/pre class="udf" i 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 i/pre i povstart(fade=10,ishadow);
fractal(-1,-1,-1,2,4);
povend();
!images/Plot3DNandhita
```

Perbedaan memungkinkan pemisahan satu objek dari objek lainnya. Seperti persimpangan, ada bagian dari objek CSG di Povray.

```
povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kami mendefinisikan objek di Povray, alih-alih menggunakan string di Euler. Definisi segera ditulis ke file.

Koordinat kotak -1 berarti [-1,-1,-1].

```
povdefine("mycube",povbox(-1,1));
```

Kita bisa menggunakan objek ini di `povobject()`, yang mengembalikan string seperti biasa.

```
c1=povobject("mycube",povlook(red));
```

Kami membuat kubus kedua, dan memutar serta menskalakannya sedikit.

```
c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ... rotate=xrotate(10°)+yrotate(10°),
scale=1.2);
```

Lalu kita ambil selisih kedua benda 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();
```

![images/Plot3D_Nandhita

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 fungsinya sedikit berbeda. Anda tidak dapat menggunakan keluaran ekspresi Maxima atau Euler.

```
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)-pow(c,2),2)+pow(pow(x,2)-1,2))"+(pow(pow(z,2)+pow(x,2)-
pow(c,2),2)+pow(pow(y,2)-1,2))-d",povlook(red))); ... 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();
```

![images/Plot3D_Nandhita

povstart(angle=70°,height=50°,zoom=4);

Buat permukaan implisit. Perhatikan sintaksis yang berbeda dalam ekspresi.

```
writeln(povsurface("pow(x,2)y-pow(y,3)-pow(z,2)",povlook(green))); ... writeAxes();
... povend();
```

![images/Plot3D_Nandhita

Objek Jaring

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambaranya dengan informasi tambahan.

Kita ingin memaksimalkan xy pada kondisi $x+y=1$ dan mendemonstrasikan sentuhan tangensial garis datar.

```
povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kita tidak dapat menyimpan objek dalam string seperti sebelumnya, karena terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan declare. Fungsi povtriangle() melakukan ini secara otomatis. Ia dapat menerima vektor normal seperti pov3d().

Berikut ini definisi objek mesh, dan segera menuliskannya ke dalam file.

```
x=0:0.02:1; y=x'; z=xy; vx=-y; vy=-x; vz=1;
mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kita mendefinisikan 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);
```

Tulis permukaannya 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)));
Tulis poin maksimal.
writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2defaultpointsize));
Tambahkan sumbu dan selesai.
writeAxes(0,1,0,1,0,1,d=0.015); ... povend();
!images/Plot3DNandhita
```

perintah ini digunakan untuk menggambar sumbu koordinat 3D (x, y, z) dengan rentang dari 0 hingga 1 pada masing-masing sumbu dan dengan margin kecil 0.015, lalu menutup blok POV rendering.

Anaglyph di Povray

Untuk menghasilkan anaglyph untuk kacamata merah/cyan, Povray harus dijalankan dua kali dari posisi kamera berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph().

Tentu saja, Anda memerlukan kacamata berwarna merah/cyan untuk melihat contoh berikut dengan benar.

Fungsi pov3d() memiliki saklar sederhana untuk menghasilkan anaglyph.

```
pov3d("-exp(-x2-y2)/2", r = 2, height = 45, anaglyph, ...center = [0, 0, 0.5], zoom = 3.5);
!images/Plot3DNandhita
```

Jika Anda membuat adegan dengan objek, Anda perlu memasukkan pembuatan adegan ke dalam fungsi, dan menjalankannya dua kali dengan nilai 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
i/prej Fungsi povanaglyph() melakukan semua ini. Parameternya seperti gabungan povstart() dan povend().
povanaglyph("myscene",zoom=4.5);
!images/Plot3DNandhita
```

perintah ini digunakan untuk menghasilkan tampilan 3D dari adegan yang bernama "myscene" dengan tingkat pembesaran tertentu, yang dapat dilihat menggunakan kacamata anaglyph (misalnya merah-biru atau merah-hijau).

Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak objek. Namun Anda tidak dibatasi pada hal ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau merupakan objek yang benar-benar baru.

Kami mendemonstrasikan torus. Perintah Povray untuk ini adalah "torus". Jadi kami mengembalikan 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 i/prej Ini torus pertama kami.
t1=povdonat(0.8,0.2)
```

torus 0.8,0.2

Mari kita gunakan objek ini untuk membuat torus kedua, diterjemahkan dan diputar.

```
t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
object torus 0.8,0.2 rotate 90 *x translate lt;0.8,0,0gt;
```

Sekarang kita tempatkan objek-objek tersebut ke dalam sebuah adegan. 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=
... writeln(povobject(t2,povlook(green,phong=1))); ... gt;povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, kesalahan tersebut tidak ditampilkan. Oleh karena itu Anda harus menggunakan

```
gt;povend(lt;keluar);
```

jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray terbuka.

```
povend(h=320,w=480);
```

```
![images/Plot3DNandhita
```

Berikut adalah contoh yang lebih rumit. Kami memecahkannya 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 punya solusinya.

```
x=simplex(A,b,c, max, check)'
```

```
[0, 1, 0.5]
```

Ya, sudah.

Selanjutnya kita mendefinisikan dua objek. Yang pertama adalah pesawat

```
function oneplane (a,b,look="") ...
```

return povplane(a,b,look) endfunction i/prej 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 i/prej Sekarang kita dapat merencanakan adegannya.

```
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 sekitar optimal.

```
writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ... povlook(red,0.9)));
```

Dan kesalahan ke arah optimal.

```
writeln(povarrow(x,c0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah objek 3D. Kita perlu menempatkan dan memutarinya sesuai dengan pandangan kita.

```
writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=5°)); ... povend();
```

```
![images/Plot3DNandhita
```

perintah ini digunakan untuk menempatkan teks "Linear Problem" pada posisi tertentu dalam grafik atau ruang 3D, dengan ukuran dan rotasi tertentu, dan kemudian mengakhiri proses render gambar

Lebih Banyak Contoh

Anda dapat menemukan beberapa contoh Povray di Euler di file berikut.

```

ja href="Examples/Dandelin Spheres.html" ;Examples/Dandelin Spheres;a;
ja href="Examples/Donat Math.html" ;Examples/Donat Math;a;
ja href="Examples/Trefoil Knot.html" ;Examples/Trefoil Knot;a;
ja href="Examples/Optimization by Affine Scaling.html" ;Examples/Optimization
by Affine Scaling;a;

```

Latihan Soal

1. buatlah gabungan 2 silinder dengan fungsi povx() berwarna merah dan povz() berwarna kuning dan zoom 4

```

povstart(zoom=4)
c1 = povcylinder (-povx,povx,1,povlook(red));
c2 = povcylinder (-povz,povz,1,povlook(yellow));
writeln (povintersection([c1,c2]));
povend()
!images/Plot3D_Nandhita

```

5 kalkulus

Kalkulus_{NandhitaPutriShalsabila} Nama : NandhitaPutriShalsabila

NIM : 23030630075

Kelas : Matematika B 2023

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, namundapatdihitungsecaranumerik),
- * Menggunakan format nama_{fungsi} & amp; = rumus_{fungsi}(untuk_{fungsi}* simbolikmurni, tidakdapatdihitunglangsung),
- * Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah function (bukan sebagai ekspresi).

Berikut adalah beberapa contoh cara mendefinisikan fungsi:

$$f(x) = 2x^2 + e^{\sin(x)}.$$

```

function f(x) := 2x2 + exp(sin(x))//fungsinumerik
f(0), f(1), f(pi)
1 4.31977682472 20.7392088022
f(a) // tidak dapat dihitung nilainya
Variable or function a not found. Error in: f(a) // tidak dapat dihitung
nilainya ...
Silakan Anda plot kurva fungsi di atas!
Berikutnya kita definisikan fungsi:

```

$$g(x) = \frac{\sqrt{x^2 - 3x}}{x + 1}.$$

```

function g(x) := sqrt(x2 - 3x)/(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 vari-
ables after errors. g: useglobal; return sqrt(x2 - 3 * x)/(x + 1)Errorin :
g(1)//kompleks,tidakdapatdihitungolehfungsinumerik...
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^3else return x^2endif;endfunction < /pre > 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):
![images/Kalkulus_Nandhita
function f(x) = 2E^x // fungsisimbolik
x 2 E
f(a) // nilai fungsisecarasimbolik
```

$$2e^a$$

f(E) // nilai fungsi berupa bilangan desimal
30.308524483
f(E),float(

30.30852448295852

```
![images/Kalkulus_Nandhita
function g(x) = 3x+1
3 x + 1
function h(x) = f(g(x)) // komposisi fungsi
3 x + 1 2 E
plot2d("h(x)",-1,1):
![images/Kalkulus_Nandhita
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.

1. Fungsi 1

```

function f(x) := x2 + 3x + 2
f(2)
12
f(-3:3)
[2, 0, 0, 2, 6, 12, 20]
plot2d("f(x)"):
!images/KalkulusNandhita
2. Fungsi 2
function g(x) := sin(x)
g(pi/4)
0.707106781187
g(-2pi:2pi)
[0, 0.841471, 0.909297, 0.14112, -0.756802, -0.958924, -0.279415, 0.656987,
0.989358, 0.412118, -0.544021, -0.99999, -0.536573]
plot2d("g(x)"):
!images/KalkulusNandhita
3. Fungsi 3
function f(x) := exp(x)
f(0.5)
1.6487212707
f(-1:1)
[0.367879, 1, 2.71828]
plot2d("f(x)"):
!images/KalkulusNandhita
4. Fungsi 4
function f(x,y) := x2 + y2 - 12
f(2,4)
8
f(-2:2,3:3)
[1, -2, -3, -2, 1]
aspect=1.5; plot3d("f(x,y)",a=-80,b=80,c=-60,d=60,angle=35°,height=30°,r=pi,n=80):
!images/KalkulusNandhita
5. Fungsi 5
function g(x,y) := y2/(x2/3)
g(2,3)
6.75
g(2:2,-2:2)
[3, 0.75, 0, 0.75, 3]
aspect=1.5; plot3d("g(x,y)",a=-80,b=80,c=-60,d=60,angle=35°,height=30°,r=pi,n=80):
!images/KalkulusNandhita
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 - 3x)/(x + 1), x, inf))

$$\lim_{x \rightarrow \infty} \frac{\sqrt{x^2 - 3x}}{x + 1} = 1$$

limit((x^3 - 13x^2 + 51x - 63)/(x^3 - 4x^2 - 3x + 18), x, 3)

$$-\frac{4}{5}$$

$$\lim_{x \rightarrow 3} \frac{x^3 - 13x^2 + 51x - 63}{x^3 - 4x^2 - 3x + 18} = -\frac{4}{5}$$

Fungsi tersebut diskontinu di titik x=3. Berikut adalah grafik fungsinya.

aspect(1.5); plot2d("(x^3-13x^2+51x-63)/(x^3-4x^2-3x+18)", 0, 4); plot2d(3, -4/5, points, style = "ow", add) :

*![images/Kalkulus_Nandhita
limit(2xsin(x)/(1 - cos(x)), x, 0)*

$$2 \left(\lim_{x \rightarrow 0} \frac{x \sin x}{1 - \cos x} \right) = 4$$

Fungsi tersebut diskontinu di titik x=0. Berikut adalah grafik fungsinya.

plot2d("2xsin(x)/(1-cos(x))", -pi, pi); plot2d(0, 4, points, style="ow", add):

*![images/Kalkulus_Nandhita
limit(cot(7h)/cot(5h), h, 0)*

$$\lim_{h \rightarrow 0} \frac{\cot(7h)}{\cot(5h)} = \frac{5}{7}$$

Fungsi tersebut juga diskontinu (karena tidak terdefinisi) di x=0. Berikut adalah grafiknya.

plot2d("cot(7x)/cot(5x)", -0.001, 0.001); plot2d(0, 5/7, points, style="ow", add):

*![images/Kalkulus_Nandhita
showev('limit(((x/8)^(1/3) - 1)/(x - 8), x, 8))*

$$\lim_{x \rightarrow 8} \frac{\frac{x^{\frac{1}{3}}}{2} - 1}{x - 8} = \frac{1}{24}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

showev('limit(1/(2x - 1), x, 0))

$$\lim_{x \rightarrow 0} \frac{1}{2x-1} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

`showev('limit((x^2 - 3x - 10)/(x - 5), x, 5))`

$$\lim_{x \rightarrow 5} \frac{x^2 - 3x - 10}{x - 5} = 7$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

`showev('limit(sqrt(x^2 + x) - x, x, inf))`

$$\lim_{x \rightarrow \infty} \sqrt{x^2 + x} - x = \frac{1}{2}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

`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.

`showev('limit(sin(x)/x, x, 0))`

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

`plot2d("sin(x)/x", -pi, pi); plot2d(0, 1, points, style="ow", add):`

`!images/Kalkulus_Nandhita`

`showev('limit(sin(x^3)/x, x, 0))`

$$\lim_{x \rightarrow 0} \frac{\sin x^3}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

`showev('limit(log(x), x, minf))`

$$\lim_{x \rightarrow -\infty} \log x = \text{infinity}$$

`showev('limit((-2)^x, x, inf))`

$$\lim_{x \rightarrow \infty} (-2)^x = \text{infinity}$$

`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))//Perhatikanhasilnya

$$\lim_{t \downarrow 5} t - \sqrt{2 - t} = 5 - \sqrt{3} i$$

plot2d("x-sqrt(2-x)", 0, 2):

![images/Kalkulus_Nandhita

showev('limit((x² - 9)/(2x² - 5x - 3), x, 3))

$$\lim_{x \rightarrow 3} \frac{x^2 - 9}{2x^2 - 5x - 3} = \frac{6}{7}$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

showev('limit((1 - cos(x))/x, x, 0))

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} = 0$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

showev('limit((x² + abs(x))/(x² - abs(x)), x, 0))

$$\lim_{x \rightarrow 0} \frac{|x| + x^2}{x^2 - |x|} = -1$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

showev('limit((1 + 1/x)^x, x, inf))

$$\lim_{x \rightarrow \infty} \left(\frac{1}{x} + 1 \right)^x = e$$

plot2d("(1+1/x)^x", 0, 1000) :

![images/Kalkulus_Nandhita

showev('limit((1 + k/x)^x, x, inf))

$$\lim_{x \rightarrow \infty} \left(\frac{k}{x} + 1 \right)^x = e^k$$

showev('limit((1 + x)^(1/x), x, 0))

$$\lim_{x \rightarrow 0} (x + 1)^{\frac{1}{x}} = e$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

showev('limit((x/(x + k))^x, x, inf))

$$\lim_{x \rightarrow \infty} \left(\frac{x}{x + k} \right)^x = e^{-k}$$

showev('limit((E^x - E²)/(x - 2), x, 2))

$$\lim_{x \rightarrow 2} \frac{e^x - e^2}{x - 2} = e^2$$

Tunjukkan limit tersebut dengan grafik, seperti contoh-contoh sebelumnya.

showev('limit(sin(1/x), x, 0))

$$\lim_{x \rightarrow 0} \sin\left(\frac{1}{x}\right) = ind$$

showev('limit(sin(1/x), x, inf))

$$\lim_{x \rightarrow \infty} \sin\left(\frac{1}{x}\right) = 0$$

plot2d("sin(1/x)", -0.001, 0.001):

![images/Kalkulus_Nandhita

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.

1. Fungsi 1

$$f(x) = \frac{9x - 3}{x + 4}$$

showev('limit((9x-3)/(x+4), x, 4))

$$\lim_{x \rightarrow 4} \frac{9x - 3}{x + 4} = \frac{33}{8}$$

plot2d("(9x-3)/(x+4)", -2, 4, -1, 5):

![images/Kalkulus_Nandhita

2. Fungsi 2

$$g(x) = 16x^3 - 3$$

showev('limit((16x^3-3), x, 0))

$$\lim_{x \rightarrow 0} 16x^3 - 3 = -3$$

plot2d("(16x^3-3)") :

![images/Kalkulus_Nandhita

3. Fungsi 3

$$f(x) = \frac{4x^2 - 2x + 10}{6x^2 + x - 6}$$

showev('limit(((4x^2-2x+10)/(6x^2+x-6)), x, 3))

$$\lim_{x \rightarrow 3} \frac{4x^2 - 2x + 10}{6x^2 + x - 6} = \frac{40}{51}$$

plot2d("(4x^2-2x+10)/(6x^2+x-6)", -6, 15, -15, 10) :

!images/Kalkulus_Nandhita

4. Fungsi 4

$$g(x) = \frac{\sin 2x}{\sin x - \cos x}$$

showev('limit(sin(2x)/(sin(x) - cos(x)), x, 0))

$$\lim_{x \rightarrow 0} \frac{\sin(2x)}{\sin x - \cos x} = 0$$

plot2d("sin(2x)/(sin(x) - cos(x))",-1,1):

!images/Kalkulus_Nandhita

5. Fungsi 5

$$f(x) = \frac{2x \tan x}{1 - \sin 4x}$$

showev('limit((2xtan(x))/(1 - sin(4x)), x, 0))

$$2 \left(\lim_{x \rightarrow 0} \frac{x \tan x}{1 - \sin(4x)} \right) = 0$$

plot2d("(2xtan(x))/(1-sin(4x))",-pi/2,2pi,0,2pi):

!images/Kalkulus_Nandhita

Turunan Fungsi

Definisi turunan:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).

showev('limit(((x+h)^2 - x^2)/h, h, 0))/turunanx^2

$$\lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2}{h} = 2x$$

p = expand((x+h)^2 - x^2)|simplify;p //pembilang dijabarkan dan disederhanakan

$$2hx + h^2$$

q =ratsimp(p/h); q//ekspresiyangakandihitunglimitnyadisederhanakan

$$2x + h$$

limit(q, h, 0)//nilailimitsebagaiturunan

$$2x$$

showev('limit(((x+h)^n - x^n)/h, h, 0))/turunanx^n

$$\lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} = n x^{n-1}$$

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 $(x+h)^n$ dengan menggunakan teorema binomial.
showev('limit((sin(x+h) - sin(x))/h, h, 0))//turunan sin(x)

$$\lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin x}{h} = \cos x$$

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)

$$\lim_{h \rightarrow 0} \frac{\log(x+h) - \log x}{h} = \frac{1}{x}$$

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

$$\lim_{h \rightarrow 0} \frac{\frac{1}{x+h} - \frac{1}{x}}{h} = -\frac{1}{x^2}$$

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" 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:

$$\lim_{h \rightarrow 0} \frac{e^{x+h} - e^x}{h}.$$

Oleh karena itu diperlukan trik khusus agar hasilnya benar.

showev('limit((E^h - 1)/h, h, 0))

$$\lim_{h \rightarrow 0} \frac{e^h - 1}{h} = 1$$

showev('factor(E(x+h) - E^x))

$$\text{factor}(e^{x+h} - e^x) = (e^h - 1) e^x$$

showev('limit(factor((E(x+h) - E^x)/h), h, 0))//turunanf(x) = e^x

$$\left(\lim_{h \rightarrow 0} \frac{e^h - 1}{h} \right) e^x = e^x$$

function f(x) = x^x

x x

showev('limit(f(x), x, 0))

$$\lim_{x \rightarrow 0} x^x = 1$$

Silakan Anda gambar kurva

$$y = x^x.$$

showev('limit((f(x+h) - f(x))/h, h, 0))//turunanf(x) = x^x

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = \text{infinity}$$

Di sini Maxima juga bermasalah terkait limit:

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h}.$$

Dalam hal ini diperlukan asumsi nilai x.

assume(x 0); showev('limit((f(x+h) - f(x))/h, h, 0))//turunanf(x) = x^x

$$\lim_{h \rightarrow 0} \frac{(x+h)^{x+h} - x^x}{h} = x^x (\log x + 1)$$

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 gt; 0]

forget(x;0)

[x lt; 0]

facts()

□

showev('limit((asin(x+h) - asin(x))/h, h, 0))//turunanarcsin(x)

$$\lim_{h \rightarrow 0} \frac{\arcsin(x+h) - \arcsin x}{h} = \frac{1}{\sqrt{1-x^2}}$$

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))/turunantan(x)

$$\lim_{h \rightarrow 0} \frac{\tan(x+h) - \tan x}{h} = \frac{1}{\cos^2 x}$$

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)

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

Hasilnya adalah cosh(x), karena

$$\frac{e^x + e^{-x}}{2} = \cosh(x).$$

plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]):

![images/Kalkulus_Nandhita

function f(x) = sin(3x⁵ + 7)²

2 5 sin (3 x + 7)

diff(f,3), diffc(f,3)

1198.32948904 1198.72863721

Apakah perbedaan diff dan diffc?

showev('diff(f(x), x))

$$\frac{d}{dx} \sin^2(3x^5 + 7) = 30x^4 \cos(3x^5 + 7) \sin(3x^5 + 7)$$

$$\%at \left(\frac{d}{dx} \sin^2(3x^5 + 7), x = 3 \right) = 2430 \cos 736 \sin 736$$

float(

$$\%at \left(\frac{d^{1.0}}{dx^{1.0}} \sin^2(3.0x^5 + 7.0), x = 3.0 \right) = 1198.728637211748$$

plot2d(f,0,3.1):

![images/Kalkulus_Nandhita

function f(x) = 5cos(2x)-2xsin(2x) // mendefinisikan fungsi f

```

5 cos(2 x) - 2 x sin(2 x)
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

```

!images/KalkulusNandhita
!images/KalkulusNandhita
!images/KalkulusNandhita
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)"],0,2pi,color=[blue,red]): //grafik fungsi dan turunan-nya

```

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.

1. Fungsi 1

```

function f(x) := x^4
showev('limit(((x+h)^4 - x^4)/h), h, 0))

```

$$\lim_{h \rightarrow 0} \frac{(x+h)^4 - x^4}{h} = 4x^3$$

```

function df(x) = limit(((x+h)^4 - x^4)/h), h, 0); df(x) // df(x) = f'(x)

```

$$4x^3$$

```

plot2d(["f(x)", "df(x)"],-pi,pi,color=[blue,red]), label("f(x)",4,0.8), label("df(x)",2,0):

```

!images/Kalkulus_Nandhita

2. Fungsi 2

```

function f(x) := cosh(x)
showev('limit(((cos(x+h))/h), h, 0))

```

$$\lim_{h \rightarrow 0} \frac{\cos(x+h)}{h} = \lim_{h \rightarrow 0} \frac{\cos(x+h)}{h}$$

```

function df(x) = limit(((cos(x+h))/h),h,0); df(x) // df(x) = f'(x)

```

$$\lim_{h \rightarrow 0} \frac{\cos(x+h)}{h}$$

plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]), label("f(x)", 1, 2), label("df(x)", 1, 2):

Syntax error in expression, or unfinished expression! df: use global; return

'limit(cos(x+h)/h, h, 0) Error in expression: df(x) y0=f(x[1], args()); Try "traceerrors" to inspect local variables.

$u = u_0$

3. Fungsi 3

function f(x) := sin(1/x)

showev('limit((sin(1/(x+h)) - sin(1/x))/h, h, 0))

$$\lim_{h \rightarrow 0} \frac{\sin\left(\frac{1}{x+h}\right) - \sin\left(\frac{1}{x}\right)}{h} = -\frac{\cos\left(\frac{1}{x}\right)}{x^2}$$

function df(x) = limit((sin(1/(x+h))-sin(1/x))/h, h, 0); df(x)/df(x) = f'(x)

$$-\frac{\cos\left(\frac{1}{x}\right)}{x^2}$$

plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]), label("f(x)", 4, 0.2), label("df(x)", 2, 4):

!images/Kalkulus_Nandhita

4. Fungsi 4

function f(x) := sqrt(x)

showev('limit((sqrt(x+h)-sqrt(x))/h, h, 0)) // turunan sqrt(x)

$$\lim_{h \rightarrow 0} \frac{9\sqrt{x+h} - 9\sqrt{x}}{h} = \frac{9}{2\sqrt{x}}$$

function df(x) = limit((sqrt(x+h)-sqrt(x))/h, h, 0); df(x)/df(x) = f'(x)

$$\frac{9}{2\sqrt{x}}$$

plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]), label("f(x)", -3, 6), label("df(x)", -3, -3):

!images/Kalkulus_Nandhita

5. Fungsi 5

function f(x) := tan(x)cos(x)

showev('limit(((tan(x+h)cos(x+h)) - tan(x)cos(x))/h, h, 0))

$$\lim_{h \rightarrow 0} \frac{\cos(x+h) \tan(x+h) - \cos x \tan x}{h} = \frac{1}{\cos x} - \sin x \tan x$$

function df(x) = limit(((tan(x+h)cos(x+h))-tan(x)cos(x))/h, h, 0); df(x)/df(x) = f'(x)

$$\frac{1}{\cos x} - \sin x \tan x$$

plot2d(["f(x)", "df(x)"], -pi, pi, color=[blue, red]), label("f(x)", 1, 2), label("df(x)", 2.5, 2):

![images/Kalkulus_Nandhita

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:

$$\int_a^b f(x) dx = F(b) - F(a), \quad \text{dengan } F'(x) = f(x).$$

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))`

$$\int \frac{1}{x+1} dx = \log(x+1)$$

`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, 2pi):`

![images/Kalkulus_Nandhita

`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(2x+1), x)

$$\int x^2 \sqrt{2x+1} dx = \frac{(2x+1)^{\frac{7}{2}}}{28} - \frac{(2x+1)^{\frac{5}{2}}}{10} + \frac{(2x+1)^{\frac{3}{2}}}{12}$$

showev('integrate(x^2 sqrt(2x+1), x, 0, 2)

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{2 \cdot 5^{\frac{5}{2}}}{21} - \frac{2}{105}$$

ratsimp(

$$\int_0^2 x^2 \sqrt{2x+1} dx = \frac{2 \cdot 5^{\frac{7}{2}} - 2}{105}$$

showev('integrate((sin(sqrt(x)+a)E^sqrt(x))/sqrt(x), x, 0, pi^2)

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x}+a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^{\pi} - 1) \sin a + (e^{\pi} + 1) \cos a$$

factor(

$$\int_0^{\pi^2} \frac{\sin(\sqrt{x}+a) e^{\sqrt{x}}}{\sqrt{x}} dx = (-e^{\pi} - 1) (\sin a - \cos a)$$

function map f(x) = E(-x^2)

2 - x E

showev('integrate(f(x), x)

$$\int e^{-x^2} dx = \frac{\sqrt{\pi} \operatorname{erf}(x)}{2}$$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

$$\operatorname{erf}(x) = \int \frac{e^{-x^2}}{\sqrt{\pi}} dx.$$

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):
!images/KalkulusNandhita
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 $x = 0.1, 0.2, 0.3, \dots, 3.2$.

```
0.1sum(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):
!images/KalkulusNandhita
```

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(3x^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(-,6I) +
6gamma_incomplete(-,-6I))554/51sin(14)+(6Igamma_incomplete(-,6I)54/51pi-
6Igamma_incomplete(-,-6I))cos(14))sin(--) - 60)/120510
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(0.2,6.0I)-4.192962712629476Igam
showev('integrate(xexp(-x),x,0,1))//Integraltentu(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|-x^2, filled, style =
"|" , fillcolor = 1, add) : //Plotdaerahantara2kurva
![images/Kalkulus_Nandhita
a=solve("x^3-x+x^2", 0), b = solve("x^3-x+x^2", 1)//absistitik-titikpotongkeduakurva
0 0.61803398875
integrate("(-x^2) - (x^3 - x)", a, b)//luasdaerahyangdiarsir
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))//Nilaiintegralsecaraeksak
```

$$\int_0^{\frac{\sqrt{5}-1}{2}} -x^3 - x^2 + x dx = \frac{13-5^{\frac{3}{2}}}{24}$$

float(

$$\int_{0.0}^{0.6180339887498949} -1.0 x^3 - 1.0 x^2 + x dx = 0.07581917135421037$$

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

$$r(t) = 1 + \sin(3t)2, \quad 0 \leq t \leq 2\pi.$$

```
t=linspace(0,2pi,1000); r=1+sin(3t)/2; x=rcos(t); y=rsin(t); ... plot2d(x,y, filled,fillcolor=red,style="/" ,r=1.
// Kita gambar kurvanya terlebih dahulu
![images/Kalkulus_Nandhita
function r(t) = 1+sin(3t)/2; 'r(t) = r(t)
```

```
r ([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, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3, 0.31, 0.32, 0.33, 0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4, 0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47, 0.48, 0.49, 0.5, 0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59, 0.6, 0.61, 0.62, 0.63, 0.64, 0.65, 0.66, 0.67, 0.68, 0.69, 0.7, 0.71, 0.72, 0.73, 0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.8, 0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.89, 0.9, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1.0, 1.01, 1.02, 1.03, 1.04, 1.05, 1.06, 1.07, 1.08, 1.09, 1.1, 1.11, 1.12, 1.13, 1.14, 1.15, 1.16, 1.17, 1.18, 1.19, 1.2, 1.21, 1.22, 1.23, 1.24, 1.25, 1.26, 1.27, 1.28, 1.29, 1.3, 1.31, 1.32, 1.33, 1.34, 1.35, 1.36, 1.37, 1.38, 1.39, 1.4, 1.41, 1.42, 1.43, 1.44, 1.45, 1.46, 1.47, 1.48, 1.49, 1.5, 1.51, 1.52, 1.53, 1.54, 1.55, 1.56, 1.57, 1.58, 1.59, 1.6, 1.61, 1.62, 1.63, 1.64, 1.65, 1.66, 1.67, 1.68, 1.69, 1.7, 1.71, 1.72, 1.73, 1.74, 1.75, 1.76, 1.77, 1.78, 1.79, 1.8, 1.81, 1.82, 1.83, 1.84, 1.85, 1.86, 1.87, 1.88, 1.89, 1.9, 1.91, 1.92, 1.93, 1.94, 1.95, 1.96, 1.97, 1.98, 1.99, 2.0, 2.01, 2.02, 2.03, 2.04, 2.05, 2.06, 2.07, 2.08, 2.09, 2.1, 2.11, 2.12, 2.13, 2.14, 2.15, 2.16, 2.17, 2.18, 2.19, 2.2, 2.21, 2.22, 2.23, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.3, 2.31, 2.32, 2.33, 2.34, 2.35, 2.36, 2.37, 2.38, 2.39, 2.4, 2.41, 2.42, 2.43, 2.44, 2.45, 2.46, 2.47, 2.48, 2.49, 2.5, 2.51, 2.52, 2.53, 2.54, 2.55, 2.56, 2.57, 2.58, 2.59, 2.6, 2.61, 2.62, 2.63, 2.64, 2.65, 2.66, 2.67, 2.68, 2.69, 2.7, 2.71, 2.72, 2.73, 2.74, 2.75, 2.76, 2.77, 2.78, 2.79, 2.8, 2.81, 2.82, 2.83, 2.84, 2.85, 2.86, 2.87, 2.88, 2.89, 2.9, 2.91, 2.92, 2.93, 2.94, 2.95, 2.96, 2.97, 2.98, 2.99, 3.0, 3.01, 3.02, 3.03, 3.04, 3.05, 3.06, 3.07, 3.08, 3.09, 3.1, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.2, 3.21, 3.22, 3.23, 3.24, 3.25, 3.26, 3.27, 3.28, 3.29, 3.3, 3.31, 3.32, 3.33, 3.34, 3.35, 3.36, 3.37, 3.38, 3.39, 3.4, 3.41, 3.42, 3.43, 3.44, 3.45, 3.46, 3.47, 3.48, 3.49, 3.5, 3.51, 3.52, 3.53, 3.54, 3.55, 3.56, 3.57, 3.58, 3.59, 3.6, 3.61, 3.62, 3.63, 3.64, 3.65, 3.66, 3.67, 3.68, 3.69, 3.7, 3.71, 3.72, 3.73, 3.74, 3.75, 3.76, 3.77, 3.78, 3.79, 3.8, 3.81, 3.82, 3.83, 3.84, 3.85, 3.86, 3.87, 3.88, 3.89, 3.9, 3.91, 3.92, 3.93, 3.94, 3.95, 3.96, 3.97, 3.98, 3.99, 4.0, 4.01, 4.02, 4.03, 4.04, 4.05, 4.06, 4.07, 4.08, 4.09, 4.1, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, 4.2, 4.21, 4.22, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28, 4.29, 4.3, 4.31, 4.32, 4.33, 4.34, 4.35, 4.36, 4.37, 4.38, 4.39, 4.4, 4.41, 4.42, 4.43, 4.44, 4.45, 4.46, 4.47, 4.48, 4.49, 4.5, 4.51, 4.52, 4.53, 4.54, 4.55, 4.56, 4.57, 4.58, 4.59, 4.6, 4.61, 4.62, 4.63, 4.64, 4.65, 4.66, 4.67, 4.68, 4.69, 4.7, 4.71, 4.72, 4.73, 4.74, 4.75, 4.76, 4.77, 4.78, 4.79, 4.8, 4.81, 4.82, 4.83, 4.84, 4.85, 4.86, 4.87, 4.88, 4.89, 4.9, 4.91, 4.92, 4.93, 4.94, 4.95, 4.96, 4.97, 4.98, 4.99, 5.0, 5.01, 5.02, 5.03, 5.04, 5.05, 5.06, 5.07, 5.08, 5.09, 5.1, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.17, 5.18, 5.19, 5.2, 5.21, 5.22, 5.23, 5.24, 5.25, 5.26, 5.27, 5.28, 5.29, 5.3, 5.31, 5.32, 5.33, 5.34, 5.35, 5.36, 5.37, 5.38, 5.39, 5.4, 5.41, 5.42, 5.43, 5.44, 5.45, 5.46, 5.47, 5.48, 5.49, 5.5, 5.51, 5.52, 5.53, 5.54, 5.55, 5.56, 5.57, 5.58, 5.59, 5.6, 5.61, 5.62, 5.63, 5.64, 5.65, 5.66, 5.67, 5.68, 5.69, 5.7, 5.71, 5.72, 5.73, 5.74, 5.75, 5.76, 5.77, 5.78, 5.79, 5.8, 5.81, 5.82, 5.83, 5.84, 5.85, 5.86, 5.87, 5.88, 5.89, 5.9, 5.91, 5.92, 5.93, 5.94, 5.95, 5.96, 5.97, 5.98, 5.99, 6.0, 6.01, 6.02, 6.03, 6.04, 6.05, 6.06, 6.07, 6.08, 6.09, 6.1, 6.11, 6.12, 6.13, 6.14, 6.15, 6.16, 6.17, 6.18, 6.19, 6.2, 6.21, 6.22, 6.23, 6.24, 6.25, 6.26, 6.27, 6.28, 6.29, 6.3, 6.31, 6.32, 6.33, 6.34, 6.35, 6.36, 6.37, 6.38, 6.39, 6.4, 6.41, 6.42, 6.43, 6.44, 6.45, 6.46, 6.47, 6.48, 6.49, 6.5, 6.51, 6.52, 6.53, 6.54, 6.55, 6.56, 6.57, 6.58, 6.59, 6.6, 6.61, 6.62, 6.63, 6.64, 6.65, 6.66, 6.67, 6.68, 6.69, 6.7, 6.71, 6.72, 6.73, 6.74, 6.75, 6.76, 6.77, 6.78, 6.79, 6.8, 6.81, 6.82, 6.83, 6.84, 6.85, 6.86, 6.87, 6.88, 6.89, 6.9, 6.91, 6.92, 6.93, 6.94, 6.95, 6.96, 6.97, 6.98, 6.99, 7.0, 7.01, 7.02, 7.03, 7.04, 7.05, 7.06, 7.07, 7.08, 7.09, 7.1, 7.11, 7.12, 7.13, 7.14, 7.15, 7.16, 7.17, 7.18, 7.19, 7.2, 7.21, 7.22, 7.23, 7.24, 7.25, 7.26, 7.27, 7.28, 7.29, 7.3, 7.31, 7.32, 7.33, 7.34, 7.35, 7.36, 7.37, 7.38, 7.39, 7.4, 7.41, 7.42, 7.43, 7.44, 7.45, 7.46, 7.47, 7.48, 7.49, 7.5, 7.51, 7.52, 7.53, 7.54, 7.55, 7.56, 7.57, 7.58, 7.59, 7.6, 7.61, 7.62, 7.63, 7.64, 7.65, 7.66, 7.67, 7.68, 7.69, 7.7, 7.71, 7.72, 7.73, 7.74, 7.75, 7.76, 7.77, 7.78, 7.79, 7.8, 7.81, 7.82, 7.83, 7.84, 7.85, 7.86, 7.87, 7.88, 7.89, 7.9, 7.91, 7.92, 7.93, 7.94, 7.95, 7.96, 7.97, 7.98, 7.99, 8.0, 8.01, 8.02, 8.03, 8.04, 8.05, 8.06, 8.07, 8.08, 8.09, 8.1, 8.11, 8.12, 8.13, 8.14, 8.15, 8.16, 8.17, 8.18, 8.19, 8.2, 8.21, 8.22, 8.23, 8.24, 8.25, 8.26, 8.27, 8.28, 8.29, 8.3, 8.31, 8.32, 8.33, 8.34, 8.35, 8.36, 8.37, 8.38, 8.39, 8.4, 8.41, 8.42, 8.43, 8.44, 8.45, 8.46, 8.47, 8.48, 8.49, 8.5, 8.51, 8.52, 8.53, 8.54, 8.55, 8.56, 8.57, 8.58, 8.59, 8.6, 8.61, 8.62, 8.63, 8.64, 8.65, 8.66, 8.67, 8.68, 8.69, 8.7, 8.71, 8.72, 8.73, 8.74, 8.75, 8.76, 8.77, 8.78, 8.79, 8.8, 8.81, 8.82, 8.83, 8.84, 8.85, 8.86, 8.87, 8.88, 8.89, 8.9, 8.91, 8.92, 8.93, 8.94, 8.95, 8.96, 8.97, 8.98, 8.99, 9.0, 9.01, 9.02, 9.03, 9.04, 9.05, 9.06, 9.07, 9.08, 9.09, 9.1, 9.11, 9.12, 9.13, 9.14, 9.15, 9.16, 9.17, 9.18, 9.19, 9.2, 9.21, 9.22, 9.23, 9.24, 9.25, 9.26, 9.27, 9.28, 9.29, 9.3, 9.31, 9.32, 9.33, 9.34, 9.35, 9.36, 9.37, 9.38, 9.39, 9.4, 9.41, 9.42, 9.43, 9.44, 9.45, 9.46, 9.47, 9.48, 9.49, 9.5, 9.51, 9.52, 9.53, 9.54, 9.55, 9.56, 9.57, 9.58, 9.59, 9.6, 9.61, 9.62, 9.63, 9.64, 9.65, 9.66, 9.67, 9.68, 9.69, 9.7, 9.71, 9.72, 9.73, 9.74, 9.75, 9.76, 9.77, 9.78, 9.79, 9.8, 9.81, 9.82, 9.83, 9.84, 9.85, 9.86, 9.87, 9.88, 9.89, 9.9, 9.91, 9.92, 9.93, 9.94, 9.95, 9.96, 9.97, 9.98, 9.99, 10.0, 10.01, 10.02, 10.03, 10.04, 10.05, 10.06, 10.07, 10.08, 10.09, 10.1, 10.11, 10.12, 10.13, 10.14, 10.15, 10.16, 10.17, 10.18, 10.19, 10.2, 10.21, 10.22, 10.23, 10.24, 10.25, 10.26, 10.27, 10.28, 10.29, 10.3, 10.31, 10.32, 10.33, 10.34, 10.35, 10.36, 10.37, 10.38, 10.39, 10.4, 10.41, 10.42, 10.43, 10.44, 10.45, 10.46, 10.47, 10.48, 10.49, 10.5, 10.51, 10.52, 10.53, 10.54, 10.55, 10.56, 10.57, 10.58, 10.59, 10.6, 10.61, 10.62, 10.63, 10.64, 10.65, 10.66, 10.67, 10.68, 10.69, 10.7, 10.71, 10.72, 10.73, 10.74, 10.75, 10.76, 10.77, 10.78, 10.79, 10.8, 10.81, 10.82, 10.83, 10.84, 10.85, 10.86, 10.87, 10.88, 10.89, 10.9, 10.91, 10.92, 10.93, 10.94, 10.95, 10.96, 10.97, 10.98, 10.99, 11.0, 11.01, 11.02, 11.03, 11.04, 11.05, 11.06, 11.07, 11.08, 11.09, 11.1, 11.11, 11.12, 11.13, 11.14, 11.15, 11.16, 11.17, 11.18, 11.19, 11.2, 11.21, 11.22, 11.23, 11.24, 11.25, 11.26, 11.27, 11.28, 11.29, 11.3, 11.31, 11.32, 11.33, 11.34, 11.35, 11.36, 11.37, 11.38, 11.39, 11.4, 11.41, 11.42, 11.43, 11.44, 11.45, 11.46, 11.47, 11.48, 11.49, 11.5, 11.51, 11.52, 11.53, 11.54, 11.55, 11.56, 11.57, 11.58, 11.59, 11.6, 11.61, 11.62, 11.63, 11.64, 11.65, 11.66, 11.67, 11.68, 11.69, 11.7, 11.71, 11.72, 11.73, 11.74, 11.75, 11.76, 11.77, 11.78, 11.79, 11.8, 11.81, 11.82, 11.83, 11.84, 11.85, 11.86, 11.87, 11.88, 11.89, 11.9, 11.91, 11.92, 11.93, 11.94, 11.95, 11.96, 11.97, 11.98, 11.99, 12.0, 12.01, 12.02, 12.03, 12.04, 12.05, 12.06, 12.07, 12.08, 12.09, 12.1, 12.11, 12.12, 12.13, 12.14, 12.15, 12.16, 12.17, 12.18, 12.19, 12.2, 12.21, 12.22, 12.23, 12.24, 12.25, 12.26, 12.27, 12.28, 12.29, 12.3, 12.31, 12.32, 12.33, 12.34, 12.35, 12.36, 12.37, 12.38, 12.39, 12.4, 12.41, 12.42, 12.43, 12.44, 12.45, 12.46, 12.47, 12.48, 12.49, 12.5, 12.51, 12.52, 12.53, 12.54, 12.55, 12.56, 12.57, 12.58, 12.59, 12.6, 12.61, 12.62, 12.63, 12.64, 12.65, 12.66, 12.67, 12.68, 12.69, 12.7, 12.71, 12.72, 12.73, 12.74, 12.75, 12.76, 12.77, 12.78, 12.79, 12.8, 12.81, 12.82, 12.83, 12.84, 12.85, 12.86, 12.87, 12.88, 12.89, 12.9, 12.91, 12.92, 12.93, 12.94, 12.95, 12.96, 12.97, 12.98, 12.99, 13.0, 13.01, 13.02, 13.03, 13.04, 13.05, 13.06, 13.07, 13.08, 13.09, 13.1, 13.11, 13.12, 13.13, 13.14, 13.15, 13.16, 13.17, 13.18, 13.19, 13.2, 13.21, 13.22, 13.23, 13.24, 13.25, 13.26, 13.27, 13.28, 13.29, 13.3, 13.31, 13.32, 13.33, 13.34, 13.35, 13.36, 13.37, 13.38, 13.39, 13.4, 13.41, 13.42, 13.43, 13.44, 13.45, 13.46, 13.47, 13.48, 13.49, 13.5, 13.51, 13.52, 13.53, 13.54, 13.55, 13.56, 13.57, 13.58, 13.59, 13.6, 13.61, 13.62, 13.63, 13.64, 13.65, 13.66, 13.67, 13.68, 13.69, 13.7, 13.71, 13.72, 13.73, 13.74, 13.75, 13.76, 13.77, 13.78, 13.79, 13.8, 13.81, 13.82, 13.83, 13.84, 13.85, 13.86, 13.87, 13.88, 13.89, 13.9, 13.91, 13.92, 13.93, 13.94, 13.95, 13.96, 13.97, 13.98, 13.99, 14.0, 14.01, 14.02, 14.03, 14.04, 14.05, 14.06, 14.07, 14.08, 14.09, 14.1, 14.11, 14.12, 14.13, 14.14, 14.15, 14.16, 14.17, 14.18, 14.19, 14.2, 14.21, 14.22, 14.23, 14.24, 14.25, 14.26, 14.27, 14.28, 14.29, 14.3, 14.31, 14.32, 14.33, 14.34, 14.35, 14.36, 14.37, 14.38, 14.39, 14.4, 14.41, 14.42, 14.43, 14.44, 14.45, 14.46, 14.47, 14.48, 14.49, 14.5, 14.51, 14.52, 14.53, 14.54, 14.55, 14.56, 14.57, 14.58, 14.59, 14.6, 14.61, 14.62, 14.63, 14.64, 14.65, 14.66, 14.67, 14.68, 14.69, 14.7, 14.71, 14.72, 14.73, 14.74, 14.75, 14.76, 14.77, 14.78, 14.79, 14.8, 14.81, 14.82, 14.83, 14.84, 14.85, 14.86, 14.87, 14.88, 14.89, 14.9, 14.91, 14.92, 14.93, 14.94, 14.95, 14.96, 14.97, 14.98, 14.99, 15.0, 15.01, 15.02, 15.03, 15.04, 15.05, 15.06, 15.07, 15.08, 15.09, 15.1, 15.11, 15.12, 15.13, 15.14, 15.15, 15.16, 15.17, 15.18, 15.19, 15.2, 15.21, 15.22, 15.23, 15.24, 15.25, 15.26, 15.27, 15.28, 15.29, 15.3, 15.31, 15.32, 15.33, 15.34, 15.35, 15.36, 15.37, 15.38, 15.39, 15.4, 15.41, 15.42, 15.43, 15.44, 15.45, 15.46, 15.47, 15.48, 15.49, 15.5, 15.51, 15.52, 15.53, 15.54, 15.55, 15.56, 15.57, 15.58, 15.59, 15.6, 15.61, 15.62, 15.63, 15.64, 15.65, 15.66, 15.67, 15.68, 15.69, 15.7, 15.71, 15.72, 15.73, 15.74, 15.75, 15.76, 15.77, 15.78, 15.79, 15.8, 15.81, 15.82, 15.83, 15.84, 15.85, 15.86, 15.87, 15.88, 15.89, 15.9, 15.91, 15.92, 15.93, 15.94, 15.95, 15.96, 15.97, 15.98, 15.99, 16.0, 16.01, 16.02, 16.03, 16.04, 16.05, 16.06, 16.07, 16.08, 16.09, 16.1, 16.11, 16.12, 16.13, 16.14, 16.15, 16.16, 16.17, 16.18, 16.19, 16.2, 16.21, 16.22, 16.23, 16.24, 16.25, 16.26, 16.27, 16.28, 16.29, 16.3, 16.31, 16.32, 16.33, 16.34, 16.35, 16.36, 16.37, 16.38, 16.39, 16.4, 16.41, 16.42, 16.43, 16.44, 16.45, 16.46, 16.47, 16.48, 16.49, 16.5, 16.51, 16.52, 16.53, 16.54, 16.55, 16.56, 16.57, 16.58, 16.59, 16.6, 16.61, 16.62, 16.63, 16.64, 16.65, 16.66, 16.67, 16.68, 16.69, 16.7, 16.71, 16.72, 16.73, 16.74, 16.75, 16.76, 16.77, 16.78, 16.79, 16.8, 16.81, 16.82, 16.83, 16.84, 16.85, 16.86, 16.87, 16.88, 16.89, 16.9, 16.91, 16.92, 16.93, 16.94, 16.95, 16.96, 16.97, 16.98, 16.99, 17.0, 17.01, 17.02, 17.03, 17.04, 17.05, 17.06, 17.07, 17.08, 17.09, 17.1, 17.11, 17.12, 17.13, 17.14, 17.15, 17.16, 17.17, 17.18, 17.19, 17.2, 17.21, 17.22, 17.23, 17.24, 17.25, 17.26, 17.27, 17.28, 17.29, 17.3, 17.31, 17.32, 17.33, 17.34, 17.35, 17.36, 17.37, 17.38, 17.39, 17.4, 17.41, 17.42, 17.43, 17.44, 17.45, 17.46, 17.47, 17.48, 17.49, 17.5, 17.51, 17.52, 17.53, 17.54, 17.55, 17.56, 17.57, 17.58, 17.59, 17.6, 17.61, 17.62, 17.63, 17.64, 17.65, 17.66, 17.67, 17.68, 17.69, 17.7, 17.71, 17.72, 17.73, 17.74, 17.75, 17.76, 17.77, 17.78, 17.79, 17.8, 17.81, 17.82, 17.83, 17.84, 17.85, 17.86, 17.87, 17.88, 17.89, 17.9, 17.91, 17.92, 17.93, 17.94, 17.95, 17.96, 17.97, 17.98, 17.99, 18.0, 18.01, 18.02, 18.03, 18.04, 18.05, 18.06, 18.07, 18.08, 18.09, 18.1, 18.11, 18.12, 18.13, 18.14, 18.15, 18.16, 18.17, 18.18, 18.19, 18.2, 18.21, 18.22, 18.23, 18.24, 18.25, 18.26, 18.27, 18.28, 18.29, 18.3, 18.31, 18.32, 18.33, 18.34, 18.35, 18.36, 18.37, 18.38, 18.39, 18.4, 18.41, 18.42, 18.43, 18.44, 18.45, 18.46, 18.47, 18.48, 18.49, 18.5, 18.51, 18.52, 18.53, 18.54, 18.5
```

Spiral Logaritmik

$$x = e^{ax} \cos x, \quad y = e^{ax} \sin x.$$

```
a=0.1; plot2d("exp(ax)cos(x)", "exp(ax)sin(x)", r=2, xmin=0, xmax=2pi):
!images/KalkulusNandhita
kill(a) // hapus ekspresi a
done
function fx(t) = exp(at)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,
```

```
function fy(t) = exp(at)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,
```

```
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 errexpl – 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

Maxima said: defint: variable of integration cannot be a constant; found errexpl – an error. To debug this try: debugmode(true);

Error in: S amp;=integrate(df(t),t,0,2*

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) :
```

```
!images/KalkulusNandhita
```

```
showev('integrate(sqrt(1+diff(x^2,x)^2), x, -1, 1))
```

$$\int_{-1}^1 \sqrt{4x^2 + 1} \, dx = \frac{\operatorname{asinh} 2 + 2\sqrt{5}}{2}$$

```
float(
```

$$\int_{-1.0}^{1.0} \sqrt{4.0x^2 + 1.0} \, dx = 2.957885715089195$$

```
x=-1:0.2:1; y=x^2; plot2d(x,y); ... plot2d(x,y, points = 1, style = "o", add = 1) :
```

```
!images/KalkulusNandhita
```

Panjang tersebut dapat dihipotesis 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:

$$x^3 + y^3 - 3xy = 0.$$

$$z = x^3 + y^3 - 3xy; z$$

$$y^3 - 3xy + x^3$$

```
plot2d(z,r=2,level=0,n=100):
```

```
![images/KalkulusNandhita
```

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=" /");
```

```
![images/KalkulusNandhita
```

Kita selesaikan persamaannya untuk x.

```
zwithy = lx, sol = solve(
```

$$\left[x = \frac{3l}{l^3 + 1}, x = 0 \right]$$

```
![images/KalkulusNandhita
```

Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

```
function f(l) = rhs(sol[1]); 'f(l) = f(l)
```

$$f(l) = \frac{3l}{l^3 + 1}$$

Fungsi tersebut juga dapat digunakan 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),xf(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):
```

```
![images/KalkulusNandhita
```

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(lf(l),l)^2)); ds(l)=ds(l)
```

$$ds(l) = \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}}$$

```
integrate(ds(l),l,0,1)
```

$$\int_0^1 \frac{\sqrt{9l^8 + 36l^6 - 36l^5 - 36l^3 + 36l^2 + 9}}{\sqrt{l^{12} + 4l^9 + 6l^6 + 4l^3 + 1}} dl$$

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral tersebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

```
2integrate("ds(x)",0,1)
4.91748872168
2romberg(ds(x),0,1)// perintah Euler lain untuk menghitung nilai hampiran
integral yang sama
4.91748872168
Perhitungan di atas 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)");returnromberg(ds,a,b);endfunction <
/pre > panjangkurva("x","x^2",-1,1)//cekuntukmenghitungpanjangkurvaparabolasebelumnya
2.95788571509
Bandingkan dengan nilai eksak di atas.
2panjangkurva(mxm("f(x)"),mxm("xf(x)"),0,1) // cek contoh terakhir, band-
ingkan hasilnya!
4.91748872168
Kita hitung panjang spiral Archimedes berikut ini dengan fungsi tersebut.
plot2d("xcos(x)","xsin(x)",xmin=0,xmax=2pi,square=1):
![images/Kalkulus_Nandhita
panjangkurva("xcos(x)","xsin(x)",0,2pi)
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(xcos(x),xsin(x)); sol//Kitagunakanuntukmenghitungpanjangkurvaterakhir di atas
```

$$\sqrt{(\cos x - x \sin x)^2 + (\sin x + x \cos x)^2}$$

sol|trigreduce|expand,integrate(

$$\frac{\operatorname{asinh}(2\pi) + 2\pi\sqrt{4\pi^2 + 1}}{2}$$

![images/Kalkulus_Nandhita
21.2562941482

Hasilnya sama dengan perhitungan menggunakan fungsi EMT.

Berikut adalah contoh lain penggunaan fungsi Maxima tersebut.
`plot2d("3x^2-1", "3x^3-1", xmin = -1/sqrt(3), xmax = 1/sqrt(3), square = 1) :`
`![images/KalkulusNandhita`
`sol = radcan(ds(3x^2 - 1, 3x^3 - 1));sol`

$$3x\sqrt{9x^2+4}$$

`showev('integrate(sol, x, 0, 1/sqrt(3))),2float(`

$$6.0 \int_{0.0}^{0.5773502691896258} x \sqrt{9.0x^2 + 4.0} dx = 2.337835372767141$$

`![images/KalkulusNandhita`
 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 $t=0$, $t=\pi/2$, $t=\pi$.

`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,col
... 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(1.33330666692022e-6*r),4.499797504338432e-6*r-sin(4.499797504338432e-
6*r),1.066581336583994e-5*r-sin(1.066581336583994e-5*r),2.083072932167196e-
5*r-sin(2.083072932167196e-5*r),3.599352055540239e-5*r-sin(3.599352055540239e-

$5^*r), 5.71526624672386e-5^*r-\sin(5.71526624672386e-5^*r), 8.530603082730626e-5^*r-$
 $\sin(8.530603082730626e-5^*r), 1.214508019889565e-4^*r-\sin(1.214508019889565e-4^*r), 1.665833531718508e-$
 $4^*r-\sin(1.665833531718508e-4^*r), 2.216991628251896e-4^*r-\sin(2.216991628251896e-$
 $4^*r), 2.877927110806339e-4^*r-\sin(2.877927110806339e-4^*r), 3.658573803051457e-$
 $4^*r-\sin(3.658573803051457e-4^*r), 4.5688535576352e-4^*r-\sin(4.5688535576352e-4^*r), 5.618675264007778e-$
 $4^*r-\sin(5.618675264007778e-4^*r), 6.817933857540259e-4^*r-\sin(6.817933857540259e-$
 $4^*r), 8.176509330039827e-4^*r-\sin(8.176509330039827e-4^*r), 9.704265741758145e-$
 $4^*r-\sin(9.704265741758145e-4^*r), 0.001141105023499428^*r-\sin(0.001141105023499428^*r), 0.001330669204938795^*r-$
 $\sin(0.001330669204938795^*r), 0.001540100153900437^*r-\sin(0.001540100153900437^*r), 0.001770376919130678^*r-$
 $\sin(0.001770376919130678^*r), 0.002022476464811601^*r-\sin(0.002022476464811601^*r), 0.002297373572865413^*r-$
 $\sin(0.002297373572865413^*r), 0.002596040745477063^*r-\sin(0.002596040745477063^*r), 0.002919448107844891^*r-$
 $\sin(0.002919448107844891^*r), 0.003268563311168871^*r-\sin(0.003268563311168871^*r), 0.003644351435886262^*r-$
 $\sin(0.003644351435886262^*r), 0.004047774895164447^*r-\sin(0.004047774895164447^*r), 0.004479793338660443^*r-$
 $\sin(0.004479793338660443^*r), 0.0049413635565565^*r-\sin(0.0049413635565565^*r), 0.005433439383882244^*r-$
 $\sin(0.005433439383882244^*r), 0.005956971605131645^*r-\sin(0.005956971605131645^*r), 0.006512907859185624^*r-$
 $\sin(0.006512907859185624^*r), 0.007102192544548636^*r-\sin(0.007102192544548636^*r), 0.007725766724910044^*r-$
 $\sin(0.007725766724910044^*r), 0.00838456803503801^*r-\sin(0.00838456803503801^*r), 0.009079530587017326^*r-$
 $\sin(0.009079530587017326^*r), 0.009811584876838586^*r-\sin(0.009811584876838586^*r), 0.0105816576913495^*r-$
 $\sin(0.0105816576913495^*r), 0.01139067201557714^*r-\sin(0.01139067201557714^*r), 0.01223954694042984^*r-$
 $\sin(0.01223954694042984^*r), 0.01312919757078923^*r-\sin(0.01312919757078923^*r), 0.01406053493400045^*r-$
 $\sin(0.01406053493400045^*r), 0.01503446588876983^*r-\sin(0.01503446588876983^*r), 0.01605189303448024^*r-$
 $\sin(0.01605189303448024^*r), 0.01711371462093175^*r-\sin(0.01711371462093175^*r), 0.01822082445851714^*r-$
 $\sin(0.01822082445851714^*r), 0.01937411182884202^*r-\sin(0.01937411182884202^*r), 0.02057446139579705^*r-$
 $\sin(0.02057446139579705^*r), 0.02182275311709253^*r-\sin(0.02182275311709253^*r), 0.02311986215626333^*r-$
 $\sin(0.02311986215626333^*r), 0.02446665879515308^*r-\sin(0.02446665879515308^*r), 0.02586400834688696^*r-$
 $\sin(0.02586400834688696^*r), 0.02731277106934082^*r-\sin(0.02731277106934082^*r), 0.02881380207911666^*r-$
 $\sin(0.02881380207911666^*r), 0.03036795126603076^*r-\sin(0.03036795126603076^*r), 0.03197606320812652^*r-$
 $\sin(0.03197606320812652^*r), 0.0336389770872163^*r-\sin(0.0336389770872163^*r), 0.03535752660496472^*r-$
 $\sin(0.03535752660496472^*r), 0.03713253989951881^*r-\sin(0.03713253989951881^*r), 0.03896483946269502^*r-$
 $\sin(0.03896483946269502^*r), 0.0408552420577305^*r-\sin(0.0408552420577305^*r), 0.04280455863760801^*r-$
 $\sin(0.04280455863760801^*r), 0.04481359426396048^*r-\sin(0.04481359426396048^*r), 0.04688314802656623^*r-$
 $\sin(0.04688314802656623^*r), 0.04901401296344043^*r-\sin(0.04901401296344043^*r), 0.05120697598153157^*r-$
 $\sin(0.05120697598153157^*r), 0.05346281777803219^*r-\sin(0.05346281777803219^*r), 0.05578231276230905^*r-$
 $\sin(0.05578231276230905^*r), 0.05816622897846346^*r-\sin(0.05816622897846346^*r), 0.06061532802852698^*r-$
 $\sin(0.06061532802852698^*r), 0.0631303649963022^*r-\sin(0.0631303649963022^*r), 0.06571208837185505^*r-$
 $\sin(0.06571208837185505^*r), 0.06836123997666599^*r-\sin(0.06836123997666599^*r), 0.07107855488944881^*r-$
 $\sin(0.07107855488944881^*r), 0.07386476137264342^*r-\sin(0.07386476137264342^*r), 0.07672058079958999^*r-$
 $\sin(0.07672058079958999^*r), 0.07964672758239233^*r-\sin(0.07964672758239233^*r), 0.08264390910047736^*r-$
 $\sin(0.08264390910047736^*r), 0.0857128256298576^*r-\sin(0.0857128256298576^*r), 0.08885417027310427^*r-$
 $\sin(0.08885417027310427^*r), 0.09206862889003742^*r-\sin(0.09206862889003742^*r), 0.09535688002914089^*r-$
 $\sin(0.09535688002914089^*r), 0.0987195948597075^*r-\sin(0.0987195948597075^*r), 0.1021574371047232^*r-$
 $\sin(0.1021574371047232^*r), 0.1056710629744951^*r-\sin(0.1056710629744951^*r), 0.1092611211010309^*r-$
 $\sin(0.1092611211010309^*r), 0.1129282524731764^*r-\sin(0.1129282524731764^*r), 0.1166730903725168^*r-$
 $\sin(0.1166730903725168^*r), 0.1204962603100498^*r-\sin(0.1204962603100498^*r), 0.1243983799636342^*r-$
 $\sin(0.1243983799636342^*r), 0.1283800591162231^*r-\sin(0.1283800591162231^*r), 0.1324418995948859^*r-$
 $\sin(0.1324418995948859^*r), 0.1365844952106265^*r-\sin(0.1365844952106265^*r), 0.140808431699002^*r-$

sin(0.140808431699002*r),0.1451142866615502*r-sin(0.1451142866615502*r),0.1495026295080298*r-sin(0.1495026295080298*r),0.1539740213994798*r-sin(0.1539740213994798*r)] does not produce a real or column vector

Error generated by error() command

adaptiveeval: error(f]"doesnotproducearealorcolumnvector");Try"traceerrors"toinspectlocalvariablesaf $dw/n, dw/n^2, dw/n; args()$);

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 amp;= radcan(sqrt(diff(ex,x)² + diff(ey,x)²));ds=trigsimp(ds

...

ds = trigsimp(ds); ds

showev('integrate(ds,x,0,2pi))//hitungpanjangsikloidsatuputaranpenuh

Maxima said: defint: variable of integration must be a simple or subscripted variable. defint: found errexp1 0: showev(f='integrate(ds,[0,1.6666583335744e-7*r,1.33330666692022e-6*r,4.499797504338432e-6*r,1.06658133658399...)) - an error. To debug this try: debugmode(true);

Error in: showev('integrate(ds,x,0,2*pi))//hitungpanjangsikloidsat...

integrate(mxm("ds"),0,2pi) // hitung secara numerik

Illegal function result in map. if maps then return gauss: if maps then y=adaptivegauss: t1=gauss(f,c,c+h,args(,),=maps);Try"traceerrors"toinspectlocalvariablesaftererrors.in returnadaptivegauss(f,a,b,eps*1000;args(,),=maps);

romberg(mxm("ds"),0,2pi) // cara lain hitung secara numerik

Wrong argument!

Cannot combine a symbolic expression here. Did you want to create a symbolic expression? Then start with amp;.

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 2π sejauh $2\pi R$.)

Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekuivalen.

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 $\mathbf{T}'(s)$ ada. Vektor ini merupakan normal kurva yang arahnya menuju pusat kurvatur, norm-nya merupakan nilai kurvatur (kelengkungan):

$$\mathbf{T}(s) = \gamma'(s), \mathbf{T}^2(s) = 1 \text{ (konstanta)} \Rightarrow \mathbf{T}'(s) \cdot \mathbf{T}(s) = 0 \kappa(s) = \|\mathbf{T}'(s)\| = \|\gamma''(s)\| = \sqrt{x''(s)^2 + y''(s)^2}.$$

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

$$x = r \cos t, \quad y = r \sin t.$$

`fx = rcos(t); fy = rsin(t);`

`assume(t 0,r 0); s=integrate(sqrt(diff(fx,t)^2+diff(fy,t)^2),t,0,t); s//elemenpanjangkurva,panjangbusur`

Maxima said: diff: second argument must be a variable; found errexpl - 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 = rcos(s/r); fy = rsin(s/r); // definisi ulang persamaan parametrik terhadap s dengan substitusi t=s/r`

$k = \text{trigsimp}(\text{sqrt}(\text{diff}(fx,s,2)^2 + \text{diff}(fy,s,2)^2)); k //$ nilai kurvatur lingkaran dengan menggunakan definisi di atas

Untuk representasi parametrik umum, misalkan

$$x = x(t), \quad y = y(t)$$

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

$$\kappa = \frac{d\phi}{ds} = \frac{\frac{d\phi}{dt}}{\frac{ds}{dt}} \quad (\phi \text{ adalah sudut kemiringan garis singgung dan } s \text{ adalah panjang kurva}) = \frac{\frac{d\phi}{dt}}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}} = \frac{1}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}} \frac{d\phi}{dt}$$

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

$$\sec^2 \phi \frac{d\phi}{dt} = \frac{d}{dt} (\tan \phi) = \frac{d}{dt} \left(\frac{dy}{dx} \right) = \frac{d}{dt} \left(\frac{dy/dt}{dx/dt} \right) = \frac{d}{dt} \left(\frac{y'(t)}{x'(t)} \right) = \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2} \cdot \frac{d\phi}{dt} = \frac{1}{\sec^2 \phi} \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2}$$

Jadi, rumus kurvatur untuk kurva parametrik

$$x = x(t), \quad y = y(t)$$

adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}.$$

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kutub

$$x = r(\theta) \cos \theta, \quad y = r(\theta) \sin \theta,$$

maka rumus kurvturnya adalah

$$\kappa(\theta) = \frac{r(\theta)^2 + 2r'(\theta)^2 - r(\theta)r''(\theta)}{(r'(\theta)^2 + r(\theta)^2)^{3/2}}.$$

(Silakan Anda turunkan rumus tersebut!)

Contoh:

Lingkaran dengan pusat (0,0) dan jari-jari r dapat dinyatakan dengan persamaan parametrik

$$x = r \cos t, \quad y = r \sin t.$$

Nilai kelengkungan lingkaran tersebut adalah

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{3/2}} = \frac{r^2}{r^3} = \frac{1}{r}.$$

Hasil cocok dengan definisi kurvatur suatu kelengkungan.

Kurva

$$y = f(x)$$

dapat dinyatakan ke dalam persamaan parametrik

$$x = t, \quad y = f(t), \quad \text{dengan } x'(t) = 1, \quad x''(t) = 0,$$

sehingga kurvturnya adalah

$$\kappa(t) = \frac{y''(t)}{(1 + y'(t)^2)^{3/2}}.$$

Contoh:

Akan ditentukan kurvatur parabola

$$y = ax^2 + bx + c.$$

```
function f(x) = ax^2 + bx + c; y=f(x)
function k(x) = (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2); k(x)=k(x) // kelengkun-
gan parabola
```

Maxima said: diff: second argument must be a variable; found errexp1 - an error. To debug this try: debugmode(true);

```
Error in: ... (x) amp;= (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) // kelengkun-
gan parabola
```

Maxima said: diff: second argument must be a variable; found errexp1 - an error. To debug this try: debugmode(true);

```
Error in: ... (x) amp;= (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

$$(-1/2, 3/4), \quad 1/k(2) = 1/2,$$

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

```
error(f)"doesnotproducearealorcolumnvector"); adaptiveevalone : s = Try"traceerrors"toinspectlocalvar
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 kurvturnya adalah

$$\kappa = \frac{F_y^2 F_{xx} - 2F_x F_y F_{xy} + F_x^2 F_{yy}}{(F_x^2 + F_y^2)^{3/2}}.$$

(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) = ax² + bx + 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),
Fyy = diff(F(x,y),y,2)

Maxima said: diff: second argument must be a variable; found errexp1 - an
error. To debug this try: debugmode(true);

Error in: Fx amp;:= diff(F(x,y),x), Fxx amp;:= diff(F(x,y),x,2), Fy amp;:= diff(F(x,y)

...

function k(x) = (Fy²Fxx - 2FxFyFxy + Fx²Fyy)/(Fx² + Fy²)^(3/2); k(x) = k(x)
// kurvatur parabola tersebut

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 per-
intah berikut (jika perlu tambahkan lagi).

* Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah * inte-
gral tentu dengan batas-batas yang menarik (Anda tentukan * sendiri), seperti
contoh-contoh tersebut.

* Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat * diinte-
gralkan (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 $t=t_0$ dan $x=b$ bersesuaian dengan $t=t_1$, 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 implisit $F(x,y)=0$

* Tentukan kurvatur masing-masing kurva dengan menggunakan keempat * representasi tersebut (hasilnya harus sama).

* Gambarkan kurva asli, kurva kurvatur, kurva jari-jari lingkaran * oskulasi, dan salah satu lingkaran oskulasinya.

1. Fungsi 1

function $f(x) = 7x^3$; $f(x)$

$$7x^3$$

showev('integrate(f(x),x))

$$7 \int x^3 dx = \frac{7x^4}{4}$$

showev('integrate(f(x),x,1,3))

$$7 \int_1^3 x^3 dx = 140$$

$x=0.01:0.02:1$; *plot2d(x,f(x+0.01),bar); plot2d("f(x)",2,3,add):*

![images/Kalkulus_Nandhita

2. Fungsi 2

function f(x) = sqrt(80-x⁴);f(x)

$$\sqrt{80 - x^4}$$

showev('integrate(f(x), x))

$$\int \sqrt{80 - x^4} \, dx = \int \sqrt{80 - x^4} \, dx$$

showev('integrate(f(x), x, 4, 8))

$$\int_4^8 \sqrt{80 - x^4} \, dx = \int_4^8 \sqrt{80 - x^4} \, dx$$

x=-2:5.6:12; plot2d(x,f(x+0.01), bar); plot2d("f(x)",2,4, add):

![images/Kalkulus_Nandhita

3. Fungsi 3

function f(x) = (tan(x))(cos((x)))²;f(x)

$$\cos^2 x \tan x$$

showev('integrate(f(x), x))

$$\int \cos^2 x \tan x \, dx = \frac{\sin^2 x}{2}$$

showev('integrate(f(x), x, 0, pi))

$$\int_0^\pi \cos^2 x \tan x \, dx = 0$$

x=-pi:0.03:pi; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,pi, add):

![images/Kalkulus_Nandhita

Barisan dan Deret

(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh penggunaan 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 berbagai 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_ilist" 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

$$x_{k+1} = g(x_k), \quad x_0 = x_0, k = 1, 2, 3, \dots, n.$$

Berikut ini disajikan contoh-contoh penggunaan iterasi dan rekursi dengan EMT. Contoh pertama menunjukkan pertumbuhan dari nilai awal 1000 dengan laju pertambahan 5

`q=1.05; iterate("xq",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 610000 per bulan, tabungan sebesar 1 juta tanpa diambil selama sekitar 10 tahunan akan habis diambil oleh bank!

`r=0.005; plot2d(iterate("(1+0.8r)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(0.8rA - a, A),`

Berikut didefinisikan fungsi untuk menghitung saldo tabungan, kemudian dilakukan iterasi.

`function saldo(x,r,a) := round(((1+0.8r)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, $x[1], \dots, x[n-1]$ 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)
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 $n = 2, 4, 5, \dots$

```
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 $\log_2(n)$.

$$x_n = A.x_{n-1} = A^2.x_{n-2} = A^3.x_{n-3} = \dots = A^{n-1}.x_1.$$

```
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 dihubungkan titik 0 dengan titik-titik kompleks tersebut menggunakan loop.

```
for i=1:cols(x); plot2d([0,x[i]], add); end:
```

Spiral tersebut juga dapat didefinisikan menggunakan fungsi rekursif, yang tidak memerlukan indeks dan bilangan kompleks. Dalam hal ini digunakan vektor kolom pada bidang.

```
function gstep (v) ...
```

```
w=[-v[2];v[1]]; return v+w/norm(w); endfunction ;/pre> 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):
```

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)
```

0.739085133211,0.739085133213

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 meningkatkan 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

```
0.999999999999999778,1.000000000000000022 ... 4.99999999999999911,5.000000000000000089 ...
```

Iterasi berikut konvergen sangat lambat.

```
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 dipercepat dengan 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 x2 = 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 < /pre > Misalnya, berikut adalah iterasi untuk mendapatkan h
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 penambahannya 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 = x3 - 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 amp;= x3 - 1; newton amp;= x - p/diff(p, x); newton ...
Selanjutnya didefinisikan fungsi untuk melakukan iterasi (aslanya 10 kali).
function iterasi(f, x, n = 10)...
loop 1 to n; x=f(x); end; return x; endfunction < /pre > Kitamula idenganmenentukan 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=amp;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 sim-
bolik 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 ex
Maxima said: taylor: 0.1539740213994798*r cannot be a variable. - an
error. To debug this try: debugmode(true);
Error in: deret amp;= 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 ekspresi pada EMT.
plot2d("exp(x)",0,3); // plot fungsi aslanya, ex
plot2d(mxm2str("deret"), add,color=4:6): // plot ketiga deret Taylor ham-
piran 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]

plot2d(["exp(x)",deret[1],deret[2],deret[3]],0,3,color=1:4):

deret is not a variable! Error in expression: deret[1] y0=f(x[1],args()); Try "trace errors" to inspect local variables

$u = u_0$

sum(sin(kx)/k, k, 1, 5)

Berikut adalah cara menggambar kurva

$$y = \sin(x) + \sin 3x + \sin 5x + \dots$$

plot2d(sum(sin((2k+1)x)/(2k+1),k,0,20),0,2pi):

Maxima output too long! Error in: plot2d(amp;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

$$y = \sum_{k=1}^{100} \sin(kx)k, \quad 0 \leq x \leq 2\pi.$$

x=linspace(0,2pi,1000); k=1:100; y=sum(sin(kx')/k)'; plot2d(x,y):

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 $x^x dx = 1$.

mxmtable("diffat(x^x, x = 1, n)", "n", 1, 8, frac = 1);

Maxima said: diff: second argument must be a variable; found errexpl 0: diffat(expr=[0,1.66665833335744e-7*r,1.33330666692022e-6*r,4.499797504338432e-6*r,1.066581336583994e-5*r,2.08307...,x=[[0,1.66665833335744e-7*r,1.33330666692022e-6*r,4.499797504338432e-6*r,1.066581336583994e-5*r,2.0830...)] – an error. To debug this try: debugmode(true);

return mxm("@expr,@var=@value")(); Try "trace errors" to inspect local variables after errors. mxmtable: y[1]=

'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n), simpsum = true))/simpsum : menghitungderetsecarasimbolik

'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf), simpsum = true))

Di sini masih gagal, hasilnya tidak dihitung.

'sum(1/x², x, 1, inf) = ev(sum(1/x², x, 1, inf), simpsum = true)/ev : menghitungnilaieksresi

'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf), simpsum = true))

Di sini masih gagal, hasilnya tidak dihitung.

```
'sum((-1)^k/(2k-1), k, 1, inf) = factor(ev(sum((-1)^k/(2k-1), k, 1, inf), simpsum = true))
```

```
ev(sum(1/n!, n, 0, inf), simpsum = true)
```

Di sini masih gagal, hasilnya tidak dihitung, harusnya hasilnya e.

```
assume(abs(x)<1); 'sum(ax^k, k, 0, inf) = ev(sum(ax^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), amp; 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)
```

```
limit(sum(x^k/k!, k, 0, n), n, inf)
```

```
function d(n) = sum(1/(k^2 - k), k, 2, n); d(n)=d(n)
```

```
d(10) = ev(d(10), simpsum = true)
```

```
d(100) = ev(d(100), simpsum = true)
```

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 Taylore^x disekitar x = 0, sampaisukuke - 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 Taylore^x disekitar x = ...
```

```
'log(x) = taylor(log(x), x, 1, 10) // deret log(x) disekitar 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) disekitar x = 1...
```

6 geometri

Geometri_{NandhitaPutriShalsabila}Nama : NandhitaPutriShalsabila

NIM : 23030630075

Kelas : Matematika B 2023

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 geometri 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 setPlotrangex1,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 vektor v dan w
 lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] s.d. $ax+by=c$
 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 $\angle ABC$
 angleBisector(A, B, C): garis bagi sudut $\angle ABC$
 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 lingkaran 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 α sudut di A
 persamaan3quadsdan1spreadpada segitiga dengan panjang sisi a, b, c
 triplespread(a,b,c) persamaan3spreadsa, sb, sc yang memebentuk suatu segitiga
 double spread(sa) : $\sin(\phi)$, dengan $\phi = \sin(\phi)^2 \text{spread}a$.

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.

```

    Sekarang tetapkan tiga titik dan gambarkan.setPlotRange(-0.5,2.5,-0.5,2.5);
// mendefinisikan bidang koordinat baru
    Variable Sekarang not found! Error in: Sekarang tetapkan tiga titik dan
gambarkan.setPlotRange(-0.5,2 ...
    Sekarang tetapkan tiga titik dan gambarkan.
    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 meliputi fungsi untuk membuat garis dan lingkaran. Format
garisnya adalah [a,b,c] yang mewakili garis dengan persamaan ax+by=c.
    lineThrough(B,C) // garis yang melalui B dan C
    [-1, 2, 2]
    Hitung garis tegak lurus yang melalui A di BC.
    h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui
A
    Dan persimpangannya dengan SM.
    D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan
BC
    Rencanakan itu.
    plotPoint(D,value=1); // koordinat D ditampilkan
    aspect(1); plotSegment(A,D); // tampilkan semua gambar hasil plot...()
    ![images/GeometriNandhita
    Hitung luas ABC:
    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 langsung dengan fungsi
    1.5
    Cara lain menghitung luas segitiga 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"):
    ![images/GeometriNandhita
    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 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"): // gam-
bar lingkaran dalam
!images/GeometriNandhita
Latihan
1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga
ABC.
2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.
Merupakan segitiga apakah itu?
3. Hitung luas segitiga tersebut.
4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat
lingkaran dalam.
5. Gambar jari-jari lingkaran dalam.
6. Hitung luas lingkaran luar dan luas lingkaran dalam segitiga ABC.
Adakah hubungan antara luas kedua lingkaran tersebut dengan luas segitiga
ABC?
setPlotRange(0,3,0,3);
A=[1,1]; plotPoint(A,"A");
B=[1,2]; plotPoint(B,"B");
C=[2,1]; plotPoint(C,"C");
l=angleBisector(A,C,B);
g=angleBisector(C,A,B);
P=lineIntersection(l,g)
[1.29289, 1.29289]
color(5); plotLine(l); plotLine(g); color(1);
plotPoint(P,"P");
plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):
!images/GeometriNandhita
r=norm(P-projectToLine(P,lineThrough(A,B)))
0.292893218813
plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):
!images/GeometriNandhita
Contoh 2: Geometri Smbolik

```

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.
 File geometri.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, sekarang kita dapat menggunakan perhitungan simbolik.
 $A = [1,0]; B = [0,1]; C = [2,2];$ // menentukan tiga titik A, B, C
 Fungsi garis dan lingkaran berfungsi sama seperti fungsi Euler, namun disediakan komputasi simbolik.

$c = \text{lineThrough}(B,C)$ // $c=BC$

$[-1, 2, 2]$

Kita bisa mendapatkan persamaan garis dengan mudah.

$\text{getLineEquation}(c, x, y), \text{solve}(\$

$$\left[y = \frac{x}{2} + 1 \right]$$

! $\text{images/Geometri}_{Nandhita}$

$\text{getLineEquation}(\text{lineThrough}([x_1, y_1], [x_2, y_2]), x, y), \text{solve}(\$

$$\left[y = \frac{-(x_1 - x) y_2 - (x - x_2) y_1}{x_2 - x_1} \right]$$

! $\text{images/Geometri}_{Nandhita}$

$\text{getLineEquation}(\text{lineThrough}(A, [x_1, y_1]), x, y) // \text{persamaan garis melalui } A \text{ dan } (x_1, y_1)$

$$(x_1 - 1) y - x y_1 = -y_1$$

$h = \text{perpendicular}(A, \text{lineThrough}(B,C))$ // h melalui A tegak lurus BC

$[2, 1, 2]$

$Q = \text{lineIntersection}(c,h)$ // Q titik potong garis $c=BC$ dan h

$2.6 [-, -] 5.5$

$\text{projectToLine}(A, \text{lineThrough}(B,C)) // \text{proyeksi } A \text{ pada } BC$

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

$\text{distance}(A, Q) // \text{jarak } AQ$

$$\frac{3}{\sqrt{5}}$$

$cc = \text{circleThrough}(A,B,C); cc // (\text{titik pusat dan jari-jari}) \text{ lingkaran melalui } A, B, C$

$$\left[\frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}} \right]$$

$r = \text{getCircleRadius}(cc); r, \text{float}(r) // \text{tampilkan nilai jari-jari}$

1.178511301977579

! $\text{images/Geometri}_{Nandhita}$

$\text{computeAngle}(A, C, B) // \text{nilai } \angle ACB$

$$\arccos\left(\frac{4}{5}\right)$$

solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1]//persamaangarisbagi < ACB

$$y = x$$

P = lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); P//titikpotong2garisbagisudut

$$\left[\frac{\sqrt{2}\sqrt{5}+2}{6}, \frac{\sqrt{2}\sqrt{5}+2}{6} \right]$$

P() // hasilnya sama dengan perhitungan sebelumnya

[0.86038, 0.86038]

Perpotongan Garis dan Lingkaran

Tentu saja, kita juga bisa memotong garis 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):

![images/Geometri_Nandhita

Hal yang sama 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, //titikpotonglingkarancdangarisl

$$\left[\left[\sqrt{7}+2, 1-\sqrt{7} \right], \left[2-\sqrt{7}, \sqrt{7}+1 \right] \right]$$

Akan ditunjukkan bahwa sudut-sudut yang menghadap busur 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;

![images/Geometri_Nandhita

Garis Sumbu

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 melalui 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);

![images/Geometri_Nandhita

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 persimpangan cukup rumit. Tapi kita bisa menyederhanakannya jika kita mencari y.

g = getLineEquation(lineThrough(P1,P2),x,y);

solve(g, y)

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

Ini memang sama dengan garis tengah tegak lurus, yang dihitung dengan cara yang sangat berbeda.

solve(getLineEquation(middlePerpendicular(A,B),x,y),y)

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

h = getLineEquation(lineThrough(A,B),x,y);

solve(h, y)

$$\left[y = \frac{(b_2 - a_2)x - a_1b_2 + a_2b_1}{b_1 - a_1} \right]$$

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);
!images/GeometriNandhita
assume(a 0); sol = solve([x^2 + y^2 = b^2, (x-a)^2 + y^2 = c^2], [x, y])
[]
Ekstrak larutan y.
ysol = y with sol[2][2]; 'y = sqrt(factor(ysol^2))
Maxima said: part: invalid index of list or matrix. - an error. To debug
this try: debugmode(true);
Error in: y amp;= y with sol[2][2]; 'y = sqrt(factor(ysol^2))...
Kami mendapatkan rumus Heron.
function H(a,b,c) = sqrt(factor((ysola/2)^2)); 'H(a,b,c)=H(a,b,c)
```

$$H(a, b, [1, 0, 4]) = \frac{a |ysol|}{2}$$

'Luas = H(2,5,6) // luas segitiga dengan panjang sisi - sisi 2, 5, 6

$$Luas = |ysol|$$

Tentu saja, setiap segitiga siku-siku adalah kasus yang terkenal.

H(3,4,5) // luas segitiga siku-siku dengan panjang sisi 3, 4, 5

Variable or function ysol not found. 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 juga bahwa ini adalah segitiga dengan luas maksimal dan kedua 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)

Variable or function ysol not found. Error in expression: 3*abs(ysol)/2
y0=f(x[1],args()); adaptiveevalone : s = Try "traceerrors" to inspect local variables after errors. plot2d :
dw/n, dw/n^2, dw/n, auto; args());

Kasus umum juga berhasil.

$\text{solve}(\text{diff}(H(a, b, c)^2, c) = 0, c)$

Maxima said: diff: second argument must be a variable; found [1,0,4] - an error. To debug this try: debugmode(true);

Error in: $\text{solve}(\text{diff}(H(a, b, c)^2, c) = 0, c)$...

Sekarang mari kita cari himpunan semua titik di mana $b+c=d$ untuk suatu konstanta d. Diketahui bahwa ini adalah elips.

$s1 = \text{subst}(d-c, b, \text{sol}[2]); s1$

Maxima said: part: invalid index of list or matrix. - an error. To debug this try: debugmode(true);

Error in: $s1 \text{ amp; } = \text{subst}(d-c, b, \text{sol}[2]); s1$...

Dan buatlah fungsinya.

$\text{function fx}(a, c, d) = \text{rhs}(s1[1]); \text{fx}(a, c, d), \text{function fy}(a, c, d) = \text{rhs}(s1[2]); \text{fy}(a, c, d)$

0

![images/Geometri_Nandhita

Sekarang kita bisa menggambar setnya. Sisi b bervariasi dari 1 sampai 4. Diketahui bahwa kita memperoleh elips.

aspect(1); plot2d(fx(3,x,5),fy(3,x,5),xmin=1,xmax=4,square=1):

Kita dapat memeriksa persamaan umum elips ini, yaitu.

$$\frac{(x - x_m)^2}{u^2} + \frac{(y - y_m)^2}{v^2} = 1,$$

dimana (xm,ym) adalah pusat, dan u dan v adalah setengah sumbu.

$\text{ratsimp}((\text{fx}(a, c, d) - a/2)^2/u^2 + \text{fy}(a, c, d)^2/v^2 \text{ with } [u = d/2, v = \text{sqrt}(d^2 - a^2)/2])$

$$\frac{a^2}{d^2}$$

Kita melihat bahwa tinggi dan luas segitiga adalah maksimal untuk $x=0$. Jadi luas segitiga dengan $a+b+c=d$ adalah maksimal jika segitiga tersebut sama sisi. Kami ingin memperolehnya secara analitis.

$\text{eqns} = [\text{diff}(H(a, b, d-(a+b))^2, a) = 0, \text{diff}(H(a, b, d-(a+b))^2, b) = 0]; \text{eqns}$

$$\left[\frac{a \text{ ysol}^2}{2} = 0, 0 = 0 \right]$$

Kita mendapatkan nilai minimum yang dimiliki oleh segitiga dengan salah satu sisinya 0, dan solusinya $a=b=c=d/3$.

$\text{solve}(\text{eqns}, [a, b])$

$$[[a = 0, b = \%r_1]]$$

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 situasinya

Pertama atur poin 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
```

$$[a, 0]$$

!images/Geometri_Nandhita

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");
```

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: ... otPoint(mxmeval("C"),"C"); plotPoint(mxmeval("A"),"A"): ...

Plot 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" ...

Hitung garis tengah tegak lurus di Maxima.

```
h = middlePerpendicular(A,B); g = middlePerpendicular(B,C);
```

Dan pusat lingkarannya.

```
U = lineIntersection(h,g);
```

Kita mendapatkan rumus 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 plot.

```
plotPoint(U()); ... plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
```

Variable a2 not found! Use global variables or parameters for string evaluation. Error in *Errorinexpression* : $[a/2, (a^2 + a_1^2 - a * a_1)/(2 * a_2)]$ *Errorin* : `plotPoint(U());plotCircle(circleWithCenter(mxmeval("U"),mxmev...`

Dengan menggunakan geometri, kita memperoleh rumus sederhana untuk radius. Kita bisa cek, apakah ini benar adanya pada Maxima. Maxima akan memfaktorkan ini hanya jika kita mengkuadratkannya.

$$c^2 / \sin(\text{computeAngle}(A, B, C))^2 |factor$$

$$\left[\frac{a_2^2 + a_1^2}{a_2^2}, 0, \frac{16 (a_2^2 + a_1^2)}{a_2^2} \right]$$

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari sembarang segitiga yang tidak sama sisi. Merupakan garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, antara lain ortocenter, sirkumcenter, centroid, titik Exeter dan pusat lingkaran sembilan titik segitiga.

Untuk demonstrasinya, kita menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kita mendefinisikan sudut-sudut segitiga di 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 ke dalamnya. Semua plot objek geometris ditambahkan ke plot saat ini.

$$\text{setPlotRange}(3); \text{plotPoint}(A, "A"); \text{plotPoint}(B, "B"); \text{plotPoint}(C, "C");$$

Kita juga bisa menjumlahkan sisi-sisi segitiga.

$$\text{plotSegment}(A, B, ""); \text{plotSegment}(B, C, ""); \text{plotSegment}(C, A, "");$$

!images/Geometri_Nandhita

Berikut luas segitiga menggunakan rumus determinan. Tentu saja kami harus mengambil nilai absolut dari hasil ini.

$$\text{areaTriangle}(A, B, C)$$

$$-\frac{7}{2}$$

Kita dapat menghitung koefisien sisi c.

$$c = \text{lineThrough}(A, B)$$

$$[-1, 3, -2]$$

Dan dapatkan juga rumus untuk baris ini.

$$\text{getLineEquation}(c, x, y)$$

$$3y - x = -2$$

Untuk bentuk Hesse, kita perlu menentukan sebuah titik, sehingga titik tersebut berada di sisi positif dari Hesseform. Memasukkan titik akan menghasilkan jarak positif ke garis..

$$\text{getHesseForm}(c, x, y, C), \text{at}(\text{$$

$$\frac{7}{\sqrt{10}}$$

!images/Geometri_Nandhita

Sekarang kita menghitung 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

$$\left[\frac{3}{14}, \frac{5}{14}\right]$$

Plot lingkaran dan pusatnya. Cu dan U bersifat simbolis. Kami mengevaluasi ekspresi ini untuk Euler.

plotCircle(LL()); plotPoint(O(), "O");

!images/Geometri_Nandhita

Kita dapat menghitung perpotongan ketinggian di ABC (ortocenter) 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 segitiga Euler.

el = lineThrough(H,O); *getLineEquation*(el, x, y)

$$-\frac{19y}{14} - \frac{x}{14} = -\frac{1}{2}$$

Tambahkan ke plot kami.

plotPoint(H(), "H"); plotLine(el(), "Garis Euler");

!images/Geometri_Nandhita

Pusat gravitasi seharusnya berada di 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

!images/Geometri_Nandhita

Teorinya memberitahu kita MH=2*MO. Kita perlu menyederhanakan dengan radcan untuk mencapai hal ini.

distance(M, H)/*distance*(M, O)|radcan

Fungsinya mencakup fungsi untuk sudut juga.

computeAngle(A, C, B), degprint(

$$\arccos\left(\frac{4}{\sqrt{5}\sqrt{13}}\right)$$

60°15'18.43"

Persamaan pusat lingkaran tidak terlalu bagus.

Q = lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))—radcan;

Q

$$\left[\frac{\left(2^{\frac{3}{2}} + 1\right) \sqrt{5} \sqrt{13} - 15 \sqrt{2} + 3}{14}, \frac{(\sqrt{2} - 3) \sqrt{5} \sqrt{13} + 5 \cdot 2^{\frac{3}{2}} + 5}{14} \right]$$

Mari kita hitung juga ekspresi jari-jari lingkaran yang tertulis.

r = distance(Q,projectToLine(Q,lineThrough(A,B)))—ratsimp; r

$$\frac{\sqrt{(-41\sqrt{2} - 31) \sqrt{5} \sqrt{13} + 115\sqrt{2} + 614}}{7\sqrt{2}}$$

LD = circleWithCenter(Q,r); // Lingkaran dalam

Mari kita tambahkan ini ke dalam plot.

color(5); plotCircle(LD());

![images/Geometri_Nandhita

Parabola

Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

p = getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); p = ' 0

$$\frac{3y - x + 2}{\sqrt{10}} - \sqrt{(2 - y)^2 + (1 - x)^2} = 0$$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

plot2d(p,level=0,add=1,contourcolor=6):

![images/Geometri_Nandhita

Ini seharusnya merupakan suatu fungsi, tetapi pemecah default Maxima hanya dapat menemukan solusinya, jika kita mengkuadratkan persamaannya.

Akibatnya, kami mendapatkan solusi palsu.

akar = solve(getHesseForm(lineThrough(A,B),x,y,C)-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

maxima: akar[1]

Menambahkan solusi pertama pada plot menunjukkan, bahwa itu memang jalan yang kita cari. Teorinya memberitahu kita bahwa itu adalah parabola yang diputar.

```

plot2d(rhs(akar[1]),add=1):
![images/GeometriNandhita
function g(x) = rhs(akar[1]); 'g(x) = g(x)//fungsiyangmende finisikankurvadiatas

$$g(x) = -3x - \sqrt{70}\sqrt{9-2x} + 26$$

T = [-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
dTC = distance(T,C); fullratsimp(dTC),float(
2.135605779339061

![images/GeometriNandhita
U = projectToLine(T,lineThrough(A,B)); U//proyeksiTpadagarisAB

$$\left[ \frac{80 - 3\sqrt{11}\sqrt{70}}{10}, \frac{20 - \sqrt{11}\sqrt{70}}{10} \right]$$

dU2AB = distance(T,U); fullratsimp(dU2AB),float(
2.135605779339061

![images/GeometriNandhita
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 gagasan klasik tentang
jarak dan sudut dengan kuadran dan penyebaran. Dengan menggunakan hal ini,
memang mungkin untuk menghindari fungsi trigonometri dalam banyak contoh,
dan tetap "rasional".
Berikut ini, saya memperkenalkan konsep, dan memecahkan beberapa masalah.
Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembun-
yikan keunggulan utama trigonometri rasional yaitu perhitungan hanya dapat
dilakukan dengan kertas dan pensil. Anda diundang untuk memeriksa hasilnya
tanpa komputer.
Intinya adalah perhitungan rasional simbolik seringkali memberikan hasil
yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri
yang rumit, yang hanya mengevaluasi perkiraan numerik saja.
load geometry;
Untuk pengenalan pertama, kami menggunakan segitiga siku-siku dengan
proporsi Mesir yang terkenal 3, 4 dan 5. Perintah berikut adalah perintah Euler
untuk memplot geometri bidang yang terdapat dalam file Euler "geometry.e".
C:= [2,1]; A:= [1,1]; B:= [1,2]; ... setPlotRange(0,3,0,3); ... plotPoint(A,"A");
plotPoint(B,"B"); plotPoint(C,"C"); ... plotSegment(B,A,"c"); plotSegment(A,C,"b");
plotSegment(C,B,"a"); ... insimg(30);
![images/GeometriNandhita

```

Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

dimana w_a adalah sudut di A. Cara umum untuk menghitung sudut ini adalah dengan mengambil invers dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak secara kasar.

`wa := arcsin(3/5); degprint(wa)`

`36°52'11.63"`

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama tentang trigonometri rasional adalah kuadran, yang menggantikan jarak. Faktanya, itu hanyalah jarak yang dikuadratkan. Di bawah ini, a , b , dan c menyatakan kuadran sisi-sisinya.

Teorema Pythagoras menjadi $a+b=c$.

$$a = 3^2; b = 4^2; c = 5^2; a + b = c$$

$$25 = 25$$

Pengertian trigonometri rasional yang kedua adalah penyebaran. Penyebaran mengukur pembukaan antar garis. Nilainya 0 jika garisnya sejajar, dan 1 jika garisnya persegi panjang. Ini adalah kuadrat sinus sudut antara dua garis.

Luas garis AB dan AC pada gambar di atas didefinisikan sebagai

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

dimana a dan c adalah kuadran suatu segitiga siku-siku yang salah satu sudutnya berada di A.

$$sa = a/c; sa$$

$$\frac{9}{25}$$

Tentu saja ini lebih mudah dihitung daripada sudutnya. Namun Anda kehilangan properti bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengonversi nilai perkiraan sudut w_a menjadi sprad, dan mencetaknya sebagai pecahan.

$$\text{fracprint}(\sin(w_a)^2)$$

$$9/25$$

Hukum kosinus trigonometri klasik diterjemahkan menjadi "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 jarak di sudut A. Sisi a , seperti biasa, berhadapan dengan sudut A.

Hukum-hukum ini diterapkan dalam file `geometri.e` yang kami muat ke Euler.

$$\text{crosslaw}(aa, bb, cc, saa)$$

$$\left[\left(bb - aa + \frac{7}{6} \right)^2, \left(bb - aa + \frac{7}{6} \right)^2, \left(bb - aa + \frac{5}{3\sqrt{2}} \right)^2 \right] = \left[\frac{14 bb (1 - saa)}{3}, \frac{14 bb (1 - saa)}{3}, \frac{5 \cdot 2^{\frac{3}{2}} bb (1 - saa)}{3} \right]$$

Dalam kasus kami, kami mendapatkan

$$\text{crosslaw}(a, b, c, sa)$$

$$1024 = 1024$$

Mari kita gunakan hukum silang ini untuk mencari penyebaran di A. Untuk melakukannya, kita buat hukum silang untuk kuadran a, b, dan c, dan selesaikan untuk penyebaran yang tidak diketahui sa.

Anda bisa melakukannya dengan tangan dengan mudah, tapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasilnya, kami sudah mendapatkannya.

$$\text{crosslaw}(a, b, c, x), \text{solve}(\text{$$

$$\left[x = \frac{9}{25} \right]$$

![images/Geometri_Nandhita

Kami sudah mengetahui hal ini. Pengertian penyebaran merupakan kasus khusus dari hukum silang.

Kita juga dapat menyelesaikannya untuk persamaan umum a,b,c. Hasilnya adalah rumus yang menghitung penyebaran sudut suatu segitiga berdasarkan kuadran ketiga sisinya.

$$\text{solve}(\text{crosslaw}(aa, bb, cc, x), x)$$

$$\left[\left[\frac{168 \, bb \, x + 36 \, bb^2 + (-72 \, aa - 84) \, bb + 36 \, aa^2 - 84 \, aa + 49}{36}, \frac{168 \, bb \, x + 36 \, bb^2 + (-72 \, aa - 84) \, bb + 36 \, aa^2 - 84 \, aa + 49}{36} \right] \right]$$

Kita bisa membuat fungsi dari hasilnya. Fungsi seperti itu sudah didefinisikan dalam file geometri.e Euler.

$$\text{spread}(a, b, c)$$

$$\frac{9}{25}$$

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi-sisinya

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak mudah didapat jika kita menggunakan trigonometri klasik.

$$\text{spread}(a, a, 4a/7)$$

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat.

$$\text{degprint}(\arcsin(\text{sqrt}(6/7)))$$

67°47'32.44"

Contoh Lain

Sekarang, mari kita coba contoh lebih lanjut.

Kita tentukan tiga sudut segitiga sebagai berikut.

```
A:=[1,2]; B:=[4,3]; C:=[0,4]; ... setPlotRange(-1,5,1,7); ... plot-
Point(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ... plotSegment(B,A,"c");
plotSegment(A,C,"b"); plotSegment(C,B,"a"); ... insimg;
```

!images/Geometri_Nandhita

Dengan menggunakan Pythagoras, mudah untuk menghitung jarak antara dua titik. Saya pertama kali menggunakan fungsi jarak file Euler untuk geometri. Fungsi jarak menggunakan geometri klasik.

$distance(A, B)$

$$\sqrt{10}$$

Euler juga memuat fungsi kuadran antara dua titik.

Pada contoh berikut, karena c+b bukan a, maka segitiga tersebut bukan persegi panjang.

$c = \text{quad}(A, B)$; $c, b = \text{quad}(A, C)$; $b, a = \text{quad}(B, C)$; $a,$

17

!images/Geometri_Nandhita

!images/Geometri_Nandhita

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan perkalian titik dua vektor. Hasilnya adalah beberapa perkiraan floating point.

$$A = \langle 1, 2 \rangle \quad B = \langle 4, 3 \rangle, \quad C = \langle 0, 4 \rangle$$

$$\mathbf{a} = C - B = \langle -4, 1 \rangle, \quad \mathbf{c} = A - B = \langle -3, -1 \rangle, \quad \beta = \angle ABC$$

$$\mathbf{a} \cdot \mathbf{c} = |\mathbf{a}| \cdot |\mathbf{c}| \cos \beta$$

$$\cos \angle ABC = \cos \beta = \frac{\mathbf{a} \cdot \mathbf{c}}{|\mathbf{a}| \cdot |\mathbf{c}|} = \frac{12 - 1}{\sqrt{17} \sqrt{10}} = \frac{11}{\sqrt{17} \sqrt{10}}$$

$wb = \text{computeAngle}(A, B, C)$; $wb, (wb/\pi 180)()$

$$\arccos \left(\frac{11}{\sqrt{10} \sqrt{17}} \right)$$

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 x.

$crosslaw(a, b, c, x), \text{solve}()$

$$\left[x = \frac{49}{50} \right]$$

!images/Geometri_{Nandhita}

Yaitu, fungsi penyebaran yang didefinisikan dalam "geometri.e".

sb = spread(b,a,c); sb

$$\frac{49}{170}$$

Maxima mendapatkan hasil yang sama dengan menggunakan trigonometri biasa, jika kita memaksakannya. Itu menyelesaikan suku $\sin(\arccos(\dots))$ menjadi hasil pecahan. Kebanyakan siswa tidak dapat melakukan hal ini.

$$\sin(\text{computeAngle}(A, B, C))^2$$

$$\frac{49}{170}$$

Setelah kita mendapatkan sebaran di B, kita dapat menghitung tinggi h_a pada sisi a. Ingat itu

$$s_b = \frac{h_a}{c}$$

menurut definisi.

$$h_a = cs_b; h_a$$

$$\frac{49}{17}$$

Gambar berikut dihasilkan dengan program geometri C.a.R., yang dapat menggambar kuadran dan sebaran.

image: (20) Rational_{Geometry}_{C.a.R.}.png

Menurut definisi, panjang h_a adalah akar kuadrat dari kuadrannya.

$$\sqrt{h_a}$$

$$\frac{7}{\sqrt{17}}$$

Sekarang kita dapat menghitung luas segitiga tersebut. Jangan lupa, bahwa kita sedang berhadapan dengan kuadran!

$$\sqrt{h_a}\sqrt{a}/2$$

$$\frac{7}{2}$$

Rumus determinan biasa memberikan hasil yang sama.

$$\text{areaTriangle}(B, A, C)$$

$$\frac{7}{2}$$

Rumus Bangau

Sekarang, mari kita selesaikan masalah ini secara umum!

$$\text{remvalue}(a,b,c,sb,ha);$$

Pertama-tama kita menghitung penyebaran di B untuk sebuah segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat ("quadrea"?), memfaktorkannya dengan Maxima, dan kita mendapatkan rumus Heron yang terkenal.

Memang benar, hal ini sulit dilakukan dengan pensil dan kertas.

$spread(b^2, c^2, a^2), factor($

$$\frac{(-c + b + a)(c - b + a)(c + b - a)(c + b + a)}{16}$$

!images/Geometri_Nandhita

Aturan Penyebaran Tiga Kali Lipat

Kerugian dari spread adalah bahwa mereka tidak lagi hanya menambahkan sudut yang sama.

Namun, tiga spread segitiga memenuhi aturan "triple spread" berikut.

$remvalue(sa, sb, sc); triplespread(sa, sb, sc)$

$$(sc + sb + sa)^2 = 2(sc^2 + sb^2 + sa^2) + 4sa sb sc$$

Aturan ini berlaku untuk tiga sudut mana pun yang besarnya 180°.

$$\alpha + \beta + \gamma = \pi$$

Sejak menyebarnya

$$\alpha, \pi - \alpha$$

sama, aturan penyebaran tiga kali lipat juga benar, jika

$$\alpha + \beta = \gamma$$

Karena penyebaran sudut negatifnya sama, maka aturan penyebaran tiga kali lipat juga berlaku, jika

$$\alpha + \beta + \gamma = 0$$

Misalnya, kita dapat menghitung penyebaran sudut 60°. Ini 3/4. Namun persamaan tersebut memiliki solusi kedua, dimana semua spread adalah 0.

$solve(triplespread(x, x, x), x)$

$$\left[x = \frac{3}{4}, x = 0 \right]$$

Penyebaran 90° jelas sama dengan 1. Jika dua sudut dijumlahkan menjadi 90°, penyebarannya menyelesaikan persamaan penyebaran rangkap tiga dengan a,b,1. Dengan perhitungan berikut kita mendapatkan a+b=1.

$triplespread(x, y, 1), solve($

$$[x = 1 - y]$$

!images/Geometri_Nandhita

Karena penyebaran 180° -t sama dengan penyebaran t, rumus penyebaran tiga kali lipat juga berlaku, jika salah satu sudut adalah jumlah atau selisih dua sudut lainnya.

Sehingga kita dapat mencari penyebaran sudut dua kali lipat tersebut. Perhatikan bahwa ada dua solusi lagi. Kami menjadikan ini sebuah fungsi.

$\text{solve}(\text{triplespread}(a, a, x), x), \text{functiondoublespread}(a) = \text{factor}(\text{rhs}(\text{$

$$[x = 4a - 4a^2, x = 0]$$

- 4 (a - 1) a

Pembagi Sudut

Inilah situasinya, kita sudah tahu.

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;

!images/Geometri_Nandhita

Mari kita hitung panjang garis bagi sudut di A. Namun kita ingin menyelesaikannya secara umum a,b,c.

remvalue(a,b,c);

Jadi pertama-tama kita menghitung penyebaran sudut yang dibagi dua di A, menggunakan rumus penyebaran tiga kali lipat.

Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut membagi dua 180° -wa.

$\text{triplespread}(x, x, a/(a + b)), \text{solve}(\text{$

$$\frac{-\sqrt{b}\sqrt{b+a}+b+a}{2b+2a}$$

!images/Geometri_Nandhita

!images/Geometri_Nandhita

Mari kita periksa persegi panjang Mesir.

$\text{sa2with}[a = 3^2, b = 4^2]$

$$\frac{1}{10}$$

Kita dapat mencetak sudut dalam Euler, setelah mentransfer penyebarannya 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):

!images/Geometri_Nandhita

Mari kita periksa sudut dalam contoh spesifik kita.

computeAngle(C,A,P), computeAngle(P,A,B)

0.321750554397 0.321750554397

Sekarang kita menghitung panjang garis bagi AP.

Kita menggunakan teorema sinus pada segitiga APC. Teorema ini menyatakan bahwa

berlaku di segitiga mana pun. Jika digabungkan, maka hal ini akan diterjemahkan ke dalam apa yang disebut dengan “hukum penyebaran” dimana a,b,c menunjukkan qudrance.

Karena spread CPA adalah 1-sa2, kita memperolehnya bisa/1=b/(1-sa2) dan dapat menghitung bisa (kuadran dari garis bagi sudut).

factor(ratsimp(b/(1-sa2))); bisa =

$$\frac{2b(b+a)}{\sqrt{b}\sqrt{b+a}+b+a}$$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

sqr(mxmeval("at(bisa,[a=3², b=4²])")), distance(A,P)
4.21637021356 4.21637021356

Kita juga bisa menghitung P menggunakan rumus spread.

py=factor(ratsimp(sa2bisa)); py

$$-\frac{b(\sqrt{b}\sqrt{b+a}-b-a)}{\sqrt{b}\sqrt{b+a}+b+a}$$

Nilainya sama dengan yang kita peroleh dengan rumus trigonometri.

sqr(mxmeval("at(py,[a=3², b=4²])"))
1.33333333333

Sudut Akord

Lihatlah 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;
![images/GeometriNandhita
```

Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran rangkap tiga untuk sudut di pusat O untuk r. Jadi kita mendapatkan rumus jari-jari kuadrat dari perilingkaran dalam kuadran sisi-sisinya.

Kali ini, Maxima menghasilkan beberapa angka 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
```

$$-\frac{abc}{c^2-2bc+a(-2c-2b)+b^2+a^2}$$

Kita dapat menjadikannya fungsi Euler.

```
function periradius(a,b,c) = rabc;
```

Mari kita periksa hasil untuk poin kita A,B,C.

```
a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

Jari-jarinya memang 1.

```
periradius(a,b,c)
```

1

Faktanya, penyebaran CBA hanya bergantung pada b dan c. Ini adalah teorema sudut tali busur.

```
spread(b,a,c)rabc|ratsimp
```

$$\frac{b}{4}$$

Faktanya, penyebarannya adalah $b/(4r)$, dan kita melihat bahwa sudut tali busur b adalah setengah sudut pusatnya.

```
doublespread(b/(4r)) - spread(b,r,r)|ratsimp
```

0

Contoh 6: Jarak Minimal pada Bidang

Catatan pendahuluan

Fungsi yang, ke titik M pada bidang, menetapkan jarak AM antara titik tetap A dan M, mempunyai garis datar yang cukup sederhana: lingkaran 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");
```

```
![images/GeometriNandhita
```

dan grafiknya juga cukup sederhana: bagian atas kerucut::

```
plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```

```
![images/GeometriNandhita
```

Tentu saja minimum 0 dicapai di A.

Dua poin

Sekarang kita lihat fungsi MA+MB dimana A dan B adalah dua titik (tetap).

Merupakan "fakta yang diketahui" bahwa kurva tingkat berbentuk elips, titik fokusnya adalah A dan B; kecuali AB minimum yang konstan pada ruas [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):
```

```
![images/GeometriNandhita
```

Grafiknya lebih menarik:

```
plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```

```
![images/GeometriNandhita
```

Pembatasan pada garis (AB) lebih terkenal:

```
plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

```
![images/GeometriNandhita
```

Tiga poin

Kini segalanya menjadi lebih sederhana: Tidak diketahui secara luas bahwa $MA+MB+MC$ mencapai nilai minimumnya pada satu titik pada bidang tersebut, namun untuk menentukannya tidaklah mudah:

1) Jika salah satu sudut segitiga ABC lebih dari 120° (katakanlah di A), maka sudut minimum dicapai pada titik tersebut (katakanlah 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;
!images/Geometri_Nandhita
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;
!images/Geometri_Nandhita
```

2) Tetapi jika semua sudut segitiga ABC kurang dari 120° , maka titik minimum ada di titik F di bagian dalam segitiga, yaitu satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (maka masing-masing sudutnya 120°):

```
C=[-0.5,1];
plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2);
!images/Geometri_Nandhita
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;
!images/Geometri_Nandhita
```

Merupakan kegiatan yang menarik untuk merealisasikan gambar di atas dengan perangkat lunak geometri; misalnya, saya tahu soft tertulis di Java yang memiliki instruksi "garis kontur"...

Semua hal di atas ditemukan oleh seorang hakim Perancis bernama Pierre de Fermat; dia menulis surat kepada para penggila lainnya seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di bagian pajak penghasilan. Jadi titik unik F sehingga $FA+FB+FC$ minimal disebut titik Fermat segitiga. Namun nampaknya beberapa tahun sebelumnya, Torricelli dari Italia telah menemukan titik ini sebelum Fermat menemukannya! Pokoknya tradisinya adalah memperhatikan hal ini F...


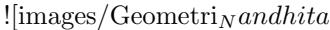

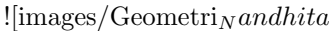
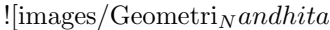
Empat poin

Langkah selanjutnya adalah menambahkan poin ke-4 D dan mencoba meminimalkan $MA+MB+MC+MD$; katakanlah Anda seorang operator TV kabel dan ingin mencari di bidang mana Anda harus memasang antena sehingga Anda dapat memberi makan empat desa dan menggunakan 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 ada nilai minimum dan tidak tercapai di simpul A, B, C, atau D:
function f(x):=d4(x[1],x[2])
neldermin("f",[0.2,0.2])
[0.142858, 0.142857]
Nampaknya dalam hal ini koordinat titik optimal bersifat rasional atau
mendekati rasional...
Sekarang ABCD adalah persegi, kita berharap titik optimalnya adalah 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;
!
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 diperhatikan gambar di bawah, terlihat bahwa kita membutuhkan dua
lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis
yang membentuk bidang yang memotong kerucut.
Kami menggunakan file geometri.e 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]
Kemudian saya baris ketiga.
g = lineThrough([-1,0],[1,1])
[- 1, 2, 1]
Kami merencanakan semuanya 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]

```

Hitung jarak ke g1.

`d1 = distance(P,projectToLine(P,g1)); d1`

$$\sqrt{\left(\frac{a^2 u}{a^2 + 1} - u\right)^2 + \frac{a^2 u^2}{(a^2 + 1)^2}}$$

Hitung jarak ke g.

`d = distance(P,projectToLine(P,g)); d`

$$\sqrt{\left(\frac{u + 2}{5} - u\right)^2 + \frac{(2u - 1)^2}{25}}$$

Dan tentukan pusat kedua lingkaran yang jaraknya sama.

`sol = solve(d1^2 = d^2, u);sol`

$$\left[u = \frac{-\sqrt{5}\sqrt{a^2 + 1} + 2a^2 + 2}{4a^2 - 1}, u = \frac{\sqrt{5}\sqrt{a^2 + 1} + 2a^2 + 2}{4a^2 - 1} \right]$$

Ada dua solusi.

Kami mengevaluasi solusi simbolis, dan menemukan kedua pusat, dan kedua jarak.

`u := sol()`

`[0.333333, 1]`

`dd := d()`

`[0.149071, 0.447214]`

Plot lingkaran ke dalam gambar.

`color(red);`

`plotCircle(circleWithCenter([0,u[1]],dd[1]),"");`

`plotCircle(circleWithCenter([0,u[2]],dd[2]),"");`

`insimg;`

`!images/GeometriNandhita`

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 kita memuat fungsi povray.

`load povray;`

`defaultpovray="C:`

Program Files

POV-Ray

v3.7

bin

pvruntime.exe"

`C:\Files-Ray\3.7.exe`

We setup the scene appropriately.

`povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);`

```

Selanjutnya kita menulis kedua bola tersebut ke 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)));
Kami menghasilkan 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, dimana bola menyentuh keru-
cut.
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)));
Lalu kita buat dua titik di mana bola menyentuh bidang. 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)));
Selanjutnya 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)));
Kami 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 menghasilkan pita abu-abu, dimana 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();
![images/GeometriNandhita
Untuk mendapatkan Anaglyph ini kita perlu memasukkan semuanya ke
dalam fungsi scene. Fungsi ini akan digunakan dua kali kemudian.
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 i/prej. Anda memerlukan kacamata merah/cyan untuk melihat efek
berikut.

```

```

povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
!images/GeometriNandhita

```

Contoh 8: Geometri Bumi

Di buku catatan ini, kami ingin melakukan beberapa perhitungan bola. Fungsi-fungsi tersebut terdapat dalam file "spherical.e" di folder contoh. Kita perlu memuat file itu terlebih dahulu.

```
load "spherical.e";
```

Untuk memasukkan posisi geografis, kita menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut koordinat 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 menghitung vektor dari satu bola ke bola ideal lainnya. Vektor ini adalah [pos, 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. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak sedekat itu, 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)-gt;" km" // perkiraan jarak FMIPA-Semarang ...

Judulnya ada fungsinya, dengan mempertimbangkan bentuk bumi yang elips. 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(asum)
 180°0'10.77"

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, ini tidak akurat karena kesalahan pengurangan pada asum-pi.
 (asum-pi)rearth(48°)² - "km²"//perkiraanluassegitigaFMIPA - 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-gt;"km²"//perkiraanluassegitigaFM...

Ada fungsi untuk ini, yang menggunakan garis lintang rata-rata dari segitiga untuk menghitung jari-jari bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.
 esarea(Solo,FMIPA,Semarang)- " km²" ,//perkiraanyangsamadenganfungsiesarea()
 2123.64310526 km²

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 suatu 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'

Fungsi-fungsi ini mengasumsikan bola ideal. Hal yang sama terjadi di bumi.
 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 429,66km. Kami mendapatkan perkiraan yang bagus.
 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)-gt;" km" // perkiraan jarak Tugu Jogja - Monas ...

Judulnya sama dengan yang dihitung di Google Earth.
 degprint(esdir(Tugu,Monas))
 294°17'2.85"

Namun kita tidak lagi mendapatkan posisi sasaran yang tepat, jika kita menambahkan heading dan jarak ke posisi semula. Hal ini terjadi karena kita tidak menghitung fungsi invers secara tepat, namun melakukan perkiraan jari-jari 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 kita tidak bisa berlayar dengan tujuan yang sama dari satu tujuan ke tujuan lainnya, jika ingin mengambil jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana saja 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 tujuan yang sama selama perjalanan.
 dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
 Sekarang kita tambah 10 kali sepersepuluh jarak, pakai jurusan Monas, kita sampai di Tugu.
 p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
 Hasilnya jauh sekali.
 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°];
 Jalur terpendek dari P1 ke P2 bukanlah lingkaran dengan garis lintang 30°, melainkan jalur 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 perjalanan kita. Untuk tujuan kasarnya, kita sesuaikan 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 mendapatkan perkiraan yang baik, jika kita menyesuaikan arah setiap 1/100 dari total jarak 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 sebuah 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 i/prej. Sekarang rencanakan semuanya.
```

```
plot3d("testplot",angle=25, height=6, own, user,zoom=4):
```

```
!images/GeometriNandhita
```

Atau gunakan plot3d untuk mendapatkan tampilan anaglyph. Ini terlihat sangat bagus dengan kacamata merah/cyan.

```
plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```

```
!images/GeometriNandhita
```

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-sisinya 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. Gambarkanlah 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).

Jawab

1.

load geometry

Numerical and symbolic geometry.

setPlotRange(-4.5,4.5,-4.5,4.5):

!images/Geometri_Nandhita

A=[-1,-1]; plotPoint(A,"A");

!images/Geometri_Nandhita

B=[2,-2]; plotPoint(B,"B");

!images/Geometri_Nandhita

C=[0,2]; plotPoint(C,"C");

!images/Geometri_Nandhita

plotSegment(A,B,"a");

!images/Geometri_Nandhita

plotSegment(B,C,"b");

!images/Geometri_Nandhita

plotSegment(A,C,"c");

!images/Geometri_Nandhita

l=angleBisector(A,B,C);

color(2); plotLine(l); color(1):

!images/Geometri_Nandhita

plotCircle(c, "Lingkaran luar segitiga ABC");

Index 3 out of bounds! Try "trace errors" to inspect local variables after errors. plotCircle: t=linspace(0,2pi,100*floor(max(1,c[3])));

2.

load geometry;

setPlotRange(3); P=[2,0]; Q=[6,0]; R=[0,-6];

plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R");

!images/Geometri_Nandhita

sol = solve([a+b=-c,36a+6b=-c,c=-6],[a,b,c])

[[a = - 1, b = 7, c = - 6]]

function y=-x² + 7x - 6

2 - x + 7 x - 6

plot2d("-x² + 7x - 6", -7, 7, -7, 7) :

!images/Geometri_Nandhita

3.

load geometry

Numerical and symbolic geometry.

setPlotRange(-2.4,2.4,-2.4,2.4):

!images/Geometri_Nandhita

A=[-1,-1]; plotPoint(A,"A");

!images/Geometri_Nandhita

B=[1,-1]; plotPoint(B,"B");


```

!images/GeometriNandhita
C=[1,1]; plotPoint(C,"C");
!images/GeometriNandhita
D=[-1,1]; plotPoint(D,"D");
!images/GeometriNandhita
plotSegment(A,B,"");
plotSegment(B,C,"");
plotSegment(C,D,"");
plotSegment(A,D,"");
aspect(1);
!images/GeometriNandhita
l=angleBisector(A,B,C);
m=angleBisector(B,C,D);
P=lineIntersection(l,m);
color(5); plotLine(l); plotLine(m); color(1);
plotPoint(P,"P");
!images/GeometriNandhita
r=norm(P-projectToLine(P,lineThrough(A,B)));
plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat ABCD");
!images/GeometriNandhita
AB=norm(A-B) //panjang sisi AB
2
CD=norm(C-D) //panjang sisi CD
2
AD=norm(A-D) //panjang sisi AD
2
BC=norm(B-C) //panjang sisi BC
2
AB.CD
4
AD.BC
4
4.
P=[-2,-2]; Q=[2,-2];
function d1(x,y):=sqrt((x-P[2])2 + (y - P[2])2)
Q=[2,-2]; function d2(x,y):=sqrt((x-P[2])2 + (y - P[2])2) + sqrt((x - Q[2])2 +
(y - Q[2])2)
fcontour("d2",xmin=-3,xmax=3,ymin=-4,ymax=2,hue=2):
!images/GeometriNandhita
plot3d("d2",xmin=-3,xmax=3,ymin=-4,ymax=2):
!images/GeometriNandhita
5.
P=[-2,-2]; Q=[2,-2];
function d1(x,y):=sqrt((x-p[2])2 + (y - p[2])2)
Q=[1,-1]; function d2(x,y):=sqrt((x-P[2])2 + (y - P[2])2) + sqrt((x + Q[2])2 +
(y + Q[2])2)

```

```
fcontour("d2",xmin=-3,xmax=3,ymin=-4,ymax=2,hue=1):
!images/GeometriNandhita
plot3d("d2",xmin=-3,xmax=3,ymin=-4,ymax=2):
!images/GeometriNandhita
plot2d("abs(x+1)+abs(x-1)",xmin=-4,xmax=4):
!images/GeometriNandhita
```

7 statistika

Statistika_{NandhitaPutriShalsabila} — — —

NAMA :Nandhita Putri Shalsabila

NIM :23030630075

KELAS :MATEMATIKA B

—
EMT untuk Statistika

Di buku catatan ini, kami mendemonstrasikan plot statistik utama, pengujian, dan distribusi di Euler.

Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar statistik. Jadi, Anda mungkin memerlukan latar belakang untuk memahami detailnya.

Asumsikan pengukuran berikut. Kami ingin menghitung nilai rata-rata dan deviasi standar yang diukur.

```
M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ... median(M), mean(M),
dev(M),
999 999.9 2.72641400622
```

Kita dapat memplot plot kotak-dan-kumis untuk datanya. Dalam kasus kami, tidak ada outlier.

```
aspect(1.75); boxplot(M):
```

aspect (1.75) digunakan untuk mengatur rasio aspek dari plot (perbandingan antara lebar dan tinggi.

boxplot(M) digunakan untuk membuat boxplot atau diagram kotak dari data di dalam variabel M. Boxplot adalah visualisasi statistik yang menunjukkan persebaran data, termasuk nilai minimum, median, dan nilai maksimum.

Contoh, kita asumsikan jumlah pria berikut dalam rentang ukuran tertentu.

```
r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Berikut adalah alur pendistribusiannya.

```
plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="
/"):

```

Kita bisa memasukkan data mentah tersebut ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal jangkauan, akhir jangkauan, jumlah pria dalam jangkauan.

Tabel dapat dicetak dengan header. Kami menggunakan vektor string untuk mengatur header.

```
T:=r[1:8]' — r[2:9]' — v'; writetable(T,labc=["BB","BA","Frek"])
```

```
BB BA Frek 155.5 159.5 22 159.5 163.5 71 163.5 167.5 136 167.5 171.5 169
171.5 175.5 139 175.5 179.5 71 179.5 183.5 32 183.5 187.5 8
```

Jika kita memerlukan nilai rata-rata dan statistik ukuran lainnya, kita perlu menghitung titik tengah rentang tersebut. Kita bisa menggunakan dua kolom pertama tabel kita untuk ini.

Symbol "—" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan opsi "labc" untuk menentukan header kolom.

```
(T[,1]+T[,2])/2
```

```
157.5 161.5 165.5 169.5 173.5 177.5 181.5 185.5
```

```
M=fold(r,[0.5,0.5])
```

```
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
```

Sekarang kita dapat menghitung mean dan deviasi sampel dengan frekuensi tertentu.

```
m,d=meandev(M,v); m, d,
```

```
169.901234568 5.98912964449
```

Mari kita tambahkan distribusi nilai normal ke diagram batang di atas. Rumus distribusi normal dengan mean m dan simpangan baku d adalah:

Karena nilainya antara 0 dan 1, maka untuk memplotnya pada bar plot harus dikalikan dengan 4 kali jumlah data.

```
plot2d("qnormal(x,m,d)sum(v)4", ... xmin=min(r),xmax=max(r),thickness=3,add=1):
```

Tabel

Di direktori buku catatan ini Anda menemukan file dengan tabel. Data tersebut merupakan hasil survei. Berikut adalah empat baris pertama file tersebut. Datanya berasal dari buku online Jerman "Einführung in die Statistik mit R" oleh A. Handl.

```
printfile("table.dat",4);
```

```
Person Sex Age Titanic Evaluation Tip Problem 1 m 30 n . 1.80 n 2 f 23 y g
1.80 n 3 f 26 y g 1.80 y
```

Tabel berisi 7 kolom angka atau token (string). Kami ingin membaca tabel dari file. Pertama, kami menggunakan terjemahan kami sendiri untuk tokennya.

Untuk ini, kami mendefinisikan kumpulan token. Fungsi strtokens() mendapatkan vektor string token dari string tertentu.

```
mf:=["m","f"]; yn:=["y","n"]; ev:=strtokens("g vg m b vb");
```

Sekarang kita membaca tabel dengan terjemahan ini.

Argumen tok2, tok4 dll. adalah terjemahan dari kolom tabel. Argumen ini tidak ada dalam daftar parameter readtable(), jadi Anda perlu menyediakannya dengan ":=".

```
MT,hd=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
load over statistics;
```

```
writetable(MT[1:10],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
Person Sex Age Titanic Evaluation Tip Problem 1 m 30 n . 1.8 n 2 f 23 y g
1.8 n 3 f 26 y g 1.8 y 4 m 33 n . 2.8 n 5 m 37 n . 1.8 n 6 m 28 y g 2.8 y 7 f 31
y vg 2.8 n 8 m 23 n . 0.8 n 9 f 24 y vg 1.8 y 10 m 26 n . 1.8 n
```

Titik "." mewakili nilai-nilai, yang tidak tersedia.

Jika kita tidak ingin menentukan token yang akan diterjemahkan terlebih dahulu, kita hanya perlu menentukan, kolom mana yang berisi token dan bukan angka.

```
ctok=[2,4,5,7]; MT,hd,tok=readtable("table.dat",ctok=ctok);
```

ctok=[2,4,5,7]: Ini adalah untuk menentukan kolom yang akan diambil yaitu kolom ke-2, ke-4, ke-5, dan ke-7.

```
tok
```

```
m n f y g vg
```

Tabel berisi entri dari file dengan token yang diterjemahkan ke dalam angka.

String khusus NA = "." diartikan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) di tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAvail.

```
MT[1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Berikut isi tabel dengan nomor yang belum diterjemahkan.

```
writetable(MT,wc=5)
```

```
1 1 30 2 . 1.8 2 2 3 23 4 5 1.8 2 3 3 26 4 5 1.8 4 4 1 33 2 . 2.8 2 5 1 37 2 .
1.8 2 6 1 28 4 5 2.8 4 7 3 31 4 6 2.8 2 8 1 23 2 . 0.8 2 9 3 24 4 6 1.8 4 10 1 26 2
. 1.8 2 11 3 23 4 6 1.8 4 12 1 32 4 5 1.8 2 13 1 29 4 6 1.8 4 14 3 25 4 5 1.8 4 15
3 31 4 5 0.8 2 16 1 26 4 5 2.8 2 17 1 37 2 . 3.8 2 18 1 38 4 5 . 2 19 3 29 2 . 3.8
2 20 3 28 4 6 1.8 2 21 3 28 4 1 2.8 4 22 3 28 4 6 1.8 4 23 3 38 4 5 2.8 2 24 3 27
4 1 1.8 4 25 1 27 2 . 2.8 4
```

Untuk kenyamanan, Anda dapat memasukkan keluaran readtable() ke dalam daftar.

```
Table=readtable("table.dat",ctok=ctok);
```

Dengan menggunakan kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dll. atau menggunakan tabel daftar.

```
writetable(Table,ctok=ctok,wc=5);
```

```
Person Sex Age Titanic Evaluation Tip Problem 1 m 30 n . 1.8 n 2 f 23 y g
1.8 n 3 f 26 y g 1.8 y 4 m 33 n . 2.8 n 5 m 37 n . 1.8 n 6 m 28 y g 2.8 y 7 f 31
y vg 2.8 n 8 m 23 n . 0.8 n 9 f 24 y vg 1.8 y 10 m 26 n . 1.8 n 11 f 23 y vg 1.8
y 12 m 32 y g 1.8 n 13 m 29 y vg 1.8 y 14 f 25 y g 1.8 y 15 f 31 y g 0.8 n 16 m
26 y g 2.8 n 17 m 37 n . 3.8 n 18 m 38 y g . n 19 f 29 n . 3.8 n 20 f 28 y vg 1.8
n 21 f 28 y m 2.8 y 22 f 28 y vg 1.8 y 23 f 38 y g 2.8 n 24 f 27 y m 1.8 y 25 m
27 n . 2.8 y
```

Fungsi tablecol() mengembalikan nilai kolom tabel, melewati baris apa pun dengan nilai NAN ("." dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

```
c,i=tablecol(MT,[5,6]);
```

Kita bisa menggunakan ini untuk mengekstrak kolom dari tabel untuk tabel baru.

```
j=[1,5,6]; writetable(MT[i,j],labc=hd[j],ctok=[2],tok=tok)
```

```
Person Evaluation Tip 2 g 1.8 3 g 1.8 6 g 2.8 7 vg 2.8 9 vg 1.8 11 vg 1.8 12
g 1.8 13 vg 1.8 14 g 1.8 15 g 0.8 16 g 2.8 20 vg 1.8 21 m 2.8 22 vg 1.8 23 g 2.8
24 m 1.8
```

Tentu saja, kita perlu mengekstrak tabel itu sendiri dari daftar Tabel dalam kasus ini.

```
MT=Table[1];
```

Tentu saja, kita juga dapat menggunakannya untuk menentukan nilai rata-rata suatu kolom atau nilai statistik lainnya.

```
mean(tablecol(MT,6))
2.175
```

Fungsi `getstatistics()` mengembalikan elemen dalam vektor, dan jumlahnya. Kami menerapkannya pada nilai "m" dan "f" di kolom kedua tabel kami.

```
xu,count=getstatistics(tablecol(MT,2)); xu, count,
[1, 3] [12, 13]
```

Kita bisa mencetak hasilnya di tabel baru.

```
writetable(count',labr=tok[xu])
m 12 f 13
```

Fungsi `selecttable()` mengembalikan tabel baru dengan nilai dalam satu kolom yang dipilih dari vektor indeks. Pertama kita mencari indeks dari dua nilai kita di tabel token.

```
v:=indexof(tok,["g","vg"])
[5, 6]
```

Sekarang kita dapat memilih baris tabel, yang memiliki salah satu nilai v pada baris ke-5.

```
MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5);
```

Sekarang kita dapat mencetak tabel, dengan nilai yang diekstraksi dan diurutkan di kolom ke-5.

```
writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7);
```

```
Person Sex Age Titanic Evaluation Tip Problem 2 f 23 y g 1.8 n 3 f 26 y g
1.8 y 6 m 28 y g 2.8 y 18 m 38 y g . n 16 m 26 y g 2.8 n 15 f 31 y g 0.8 n 12 m
32 y g 1.8 n 23 f 38 y g 2.8 n 14 f 25 y g 1.8 y 9 f 24 y vg 1.8 y 7 f 31 y vg 2.8
n 20 f 28 y vg 1.8 n 22 f 28 y vg 1.8 y 13 m 29 y vg 1.8 y 11 f 23 y vg 1.8 y
```

Untuk statistik selanjutnya, kami ingin menghubungkan dua kolom tabel. Jadi kita ekstrak kolom 2 dan 4 dan urutkan tabelnya.

```
i:=sortedrows(MT,[2,4]); ... writetable(tablecol(MT[i],[2,4]),ctok=[1,2],tok=tok)
m n m n m n m n m n m n m y m y m y m y f n f y f y f y f y
f y f y f y f y f y f y f y
```

Dengan `getstatistics()`, kita juga bisa menghubungkan jumlah dalam dua kolom tabel satu sama lain.

```
MT24=tablecol(MT,[2,4]); ... xu1,xu2,count=getstatistics(MT24[1],MT24[2]);
... writetable(count,labr=tok[xu1],labc=tok[xu2])
n y m 7 5 f 1 12
```

Sebuah tabel dapat ditulis ke file.

```
filename="test.dat"; ... writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
```

Kemudian kita bisa membaca tabel dari file tersebut.

```
MT2,hd,tok2,hdr=readtable(filename, clabs, rlabs); ... writetable(MT2,labr=hdr,labc=hd)
n y m 7 5 f 1 12
```

Dan hapus filenya.

```
fileremove(filename);
```

Distribusi

Dengan `plot2d`, ada metode yang sangat mudah untuk memplot sebaran data eksperimen.

```
p=normal(1,1000);
plot2d(p,distribution=20,style="
/");
plot2d("qnormal(x,0,1)",add=1):
p=normal(1,1000); digunakan untuk menciptakan 1000 sampel acak yang
terdistribusi normal dengan mean (rata-rata) 1 dan standar deviasi 1000.
plot2d("qnormal(x,0,1)",add=1);
digunakan untuk menambahkan plot dari distribusi normal standar (dengan
mean 0 dan standar deviasi 1) ke grafik yang sama. Fungsi qnormal(x,0,1)
mengacu pada distribusi kumulatif dari variabel acak normal standar. add=1
menunjukkan bahwa grafik ini harus ditambahkan ke grafik yang sudah ada,
bukan dibuat baru.
```

Perlu diperhatikan perbedaan antara bar plot (sampel) dan kurva normal (distribusi sebenarnya). Masukkan kembali ketiga perintah untuk melihat hasil pengambilan sampel lainnya.

Berikut adalah perbandingan 10 simulasi dari 1000 nilai terdistribusi normal menggunakan apa yang disebut plot kotak. Plot ini menunjukkan median, kuartil 25, dan outlier.

```
p=normal(10,1000); boxplot(p):
```

Untuk menghasilkan bilangan bulat acak, Euler memiliki `intrandom`. Mari kita simulasi lemparan dadu dan plot distribusinya.

Kita menggunakan fungsi `getmultiplicities(v,x)`, yang menghitung seberapa sering elemen `v` muncul di `x`. Kemudian kita plot hasilnya menggunakan `kolomplot()`.

```
k=intrandom(1,6000,6); ... columnsplot(getmultiplicities(1:6,k)); ...
ygrid(1000,color=red):
```

Meskipun `intrandom(n,m,k)` mengembalikan bilangan bulat yang terdistribusi secara seragam dari 1 hingga `k`, distribusi bilangan bulat lainnya dapat digunakan dengan `randpint()`.

Dalam contoh berikut, probabilitas untuk 1,2,3 masing-masing adalah 0,4,0.1,0.5.

```
randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,
[378, 102, 520])
```

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Lihat referensinya.

Misalnya, kita mencoba distribusi eksponensial. Variabel acak kontinu `X` dikatakan berdistribusi eksponensial, jika PDF-nya diberikan oleh

```
with parameter
plot2d(randexponential(1,1000,2), distribution):
```

Parameter pertama (1) adalah λ , yang merupakan parameter distribusi eksponensial.

Parameter kedua (1000) menunjukkan jumlah angka acak yang dihasilkan.

Parameter ketiga (2) bisa menunjukkan dimensi atau bentuk output.

Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversnya.

```
plot2d("normaldis",-4,4):
```

Berikut ini adalah salah satu cara untuk memplot kuantil.

```
plot2d("qnormal(x,1,1.5)",-4,6); ... plot2d("qnormal(x,1,1.5)",a=2,b=5, add, filled):
```

Peluang berada di kawasan hijau adalah sebagai berikut.

```
normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

```
0.248662156979
```

Ini dapat dihitung secara numerik dengan integral berikut.

```
gauss("qnormal(x,1,1.5)",2,5)
```

```
0.248662156979
```

Mari kita bandingkan distribusi binomial dengan distribusi normal yang mean dan deviasinya sama. Fungsi `invbindis()` menyelesaikan interpolasi linier antara nilai integer.

```
invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5sqrt(1000))
```

```
525.516721219 526.007419394
```

Fungsi `qdis()` adalah kepadatan distribusi chi-kuadrat. Seperti biasa, Euler memetakan vektor ke fungsi ini. Dengan demikian kita mendapatkan plot semua distribusi chi-kuadrat dengan derajat 5 sampai 30 dengan mudah dengan cara berikut.

```
plot2d("qchidis(x,(5:5:50)')",0,50):
```

Euler memiliki fungsi akurat untuk mengevaluasi distribusi. Mari kita periksa `chidis()` dengan integral.

Penamaannya mencoba untuk konsisten. Misalnya.,

- * distribusi chi-kuadratnya adalah `chidis()`,

- * fungsi kebalikannya adalah `invchidis()`,

- * kepadatannya adalah `qchidis()`.

Pelengkap distribusi (ekor atas) adalah `chicdis()`.

```
chidis(1.5,2), integrate("qchidis(x,2)",0,1.5)
```

```
0.527633447259 0.527633447259
```

Distribusi Diskrit

Distribusi diskret adalah jenis distribusi probabilitas yang digunakan untuk variabel acak diskret, yaitu variabel yang hanya dapat memiliki nilai tertentu, biasanya dalam bentuk bilangan bulat.

Untuk menentukan distribusi diskrit Anda sendiri, Anda dapat menggunakan metode berikut.

Pertama kita atur fungsi distribusinya.

```
wd = 0—((1:6)+[-0.01,0.01,0,0,0,0])/6
```

```
[0, 0.165, 0.335, 0.5, 0.666667, 0.833333, 1]
```

Perintah ini menggunakan operator `—` dan `+` untuk membuat nilai dalam variabel `wd`.

`1:6` Ini menghasilkan vektor `[1, 2, 3, 4, 5, 6]`.

`(1:6) + [-0.01, 0.01, 0, 0, 0, 0]`: Operasi ini menambahkan kedua vektor elemen per elemen.

Hasilnya:

$[1-0.01, 2+0.01, 3, 4, 5, 6] = [0.99, 2.01, 3, 4, 5, 6]/6$ Membagi setiap elemen hasil penjumlahan tadi dengan 6.

Hasilnya:

Artinya dengan probabilitas $wd[i+1]-wd[i]$ kita menghasilkan nilai acak i .

Ini hampir merupakan distribusi yang seragam. Mari kita tentukan generator nomor acak untuk ini. Fungsi $find(v, x)$ mencari nilai x pada vektor v . Fungsi ini juga berfungsi untuk vektor x .

```
function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahannya sangat halus sehingga kita hanya melihatnya dengan banyak iterasi.

Fungsi $wrongdice$ mengembalikan sebuah matriks berukuran $n \times m$, di mana setiap elemen dari matriks ini adalah indeks posisi dari elemen wd yang paling sesuai (atau mendekati) nilai acak dari $random(n, m)$.

```
columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```

Hasil $columnsplot$ akan menunjukkan frekuensi relatif dari setiap angka (1 hingga 6), yang memungkinkan Anda untuk melihat apakah distribusi itu merata atau tidak.

Berikut adalah fungsi sederhana untuk memeriksa keseragaman distribusi nilai $1 \dots K$ dalam v . Kita menerima hasilnya, jika untuk semua frekuensi

Metode tersebut merupakan metode statistik untuk menguji keseragaman distribusi. Distribusi dianggap seragam jika frekuensi setiap nilai dalam v mendekati frekuensi ideal $1/K$, dengan deviasi yang tidak melebihi batas toleransi.

```
function checkrandom (v, delta=1) ...
```

```
K=max(v); n=cols(v); fr=getfrequencies(v,1:K); return max(fr/n-1/K);delta/sqrt(n);
endfunction i/prej. Memang fungsinya menolak distribusi seragam.
```

```
checkrandom(wrongdice(1,1000000))
```

```
0
```

Dan ia menerima generator acak bawaan.

Manual:

- Asumsi dadu, maka peluang setiap sisi = $1/6$

Dalam 1 juta lemparan maka

* Frekuensi setiap sisi fr . Proporsi tiap sisi = fr/n * Misalkan frekuensi munculnya angka adalah * [160000, 170000, 180000, 150000, 170000, 170000]

* Maka proporsi setiap angka: * [160000, 170000, 180000, 150000, 170000, 170000] $\frac{1}{1000000} \cdot [0.16, 0.17, 0.18, 0.15, 0.17, 0.17]$

* Deviasi maksimum $fn/n - 1/K * 1$ $K=\frac{1}{6}=0.1667 \cdot ([0.16, 0.17, 0.18, 0.15, 0.17, 0.17] - 0.1667) \cdot \max(-0.0067, 0.0033, 0.0133, -0.0133, 0.0033, 0.0133)$

* Bandingkan dengan batas toleransi. * Batas= $\delta \frac{1}{\sqrt{n}} = \frac{1}{\sqrt{1000000}} = \frac{1}{1000} = 0.001 \cdot 0.0133gt; 0.001$

Hasil 0 di sini mengindikasikan bahwa fungsi $checkrandom$ telah menentukan bahwa distribusi tidak seragam.

```
checkrandom(intrandom(1,1000000,6))
```

```
1
```

$checkrandom$ mengembalikan 1 atau true yang berarti bahwa distribusi dari 1 juta bilangan acak rentang 1 sampai 6 dianggap cukup seragam dalam batas toleransi yang ditetapkan.

Kita dapat menghitung distribusi binomial. Pertama ada `binomialsom()`, yang mengembalikan probabilitas i atau kurang hit dari n percobaan.

Misal kita akan menghitung probabilitas dari distribusi binomial di mana terdapat 1000 percobaan (misalnya, 1000 kali pelemparan koin), dengan probabilitas sukses pada setiap percobaan sebesar 0.4, dan kita ingin mengetahui probabilitas mendapatkan tepat 410 sukses.

Secara matematis, ini dihitung dengan rumus:

`bindis(410,1000,0.4)`

0.751401349654

`bindis(4,10,0.6)`

0.1662386176

Manual:

Secara matematis, ini dihitung dengan rumus:

* Untuk $k = 0$ * $P(X=0) = 100 \cdot (0.6)^0 \cdot (0.4)^{10} \approx 0.00010$

* Untuk $k = 1$ * $P(X = 1) = 101 \cdot (0.6)^1 \cdot (0.4)^9 \approx 0.00157$

* Untuk $k = 2$ * $P(X = 2) = 102 \cdot (0.6)^2 \cdot (0.4)^8 \approx 0.01061$

* Untuk $k = 3$ * $P(X = 3) = 103 \cdot (0.6)^3 \cdot (0.4)^7 \approx 0.04246$

* Untuk $k = 4$ * $P(X = 4) = 104 \cdot (0.6)^4 \cdot (0.4)^6 \approx 0.11147$ * Maka, $*P(X \leq 4) = P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3) + P(X = 4) * P(X \leq 4) = 0.00010 + 0.00157 + 0.01061 + 0.04246 + 0.11147 * P(X \leq 4) \approx 0.1662$

Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p . Tingkat defaultnya adalah alfa.

Arti dari interval ini adalah jika p berada di luar interval, hasil pengamatan 410 dalam 1000 jarang terjadi.

`clopperpearson(410,1000)`

[0.37932, 0.441212]

Perintah berikut adalah cara langsung untuk mendapatkan hasil di atas. Namun untuk n yang besar, penjumlahan langsungnya tidak akurat dan lambat.

$p=0.4$; $i=0:410$; $n=1000$; $\text{sum}(\text{bin}(n,i)p^i(1-p)^{(n-i)})$

0.751401349655

Omong-omong, `invbinsum()` menghitung kebalikan dari `binomialsom()`.

`invbindis(0.75,1000,0.4)`

409.932733047

Di Bridge, kami mengasumsikan 5 kartu beredar (dari 52) di dua tangan (26 kartu). Mari kita hitung probabilitas distribusi yang lebih buruk dari 3:2 (misalnya 0:5, 1:4, 4:1, atau 5:0).

`2hypergeomsum(1,5,13,26)`

0.321739130435

Ada juga simulasi distribusi multinomial.

`randmultinomial(10,1000,[0.4,0.1,0.5])`

381 100 519 376 91 533 417 80 503 440 94 466 406 112 482 408 94 498 395
107 498 399 96 505 428 87 485 400 99 501

Merencanakan Data/ Plot Data

Untuk memetakan data, kami mencoba hasil pemilu Jerman sejak tahun 1990, diukur dalam jumlah kursi.

```

BW := [ ... 1990,662,319,239,79,8,17; ... 1994,672,294,252,47,49,30; ...
1998,669,245,298,43,47,36; ... 2002,603,248,251,47,55,2; ... 2005,614,226,222,61,51,54;
... 2009,622,239,146,93,68,76; ... 2013,631,311,193,0,63,64];
Untuk beberapa bagian, kami menggunakan rangkaian nama.
P:=["CDU/CSU","SPD","FDP","Gr","Li"];
Mari kita cetak persentasenya dengan baik.
Pertama kita mengekstrak kolom yang diperlukan. Kolom 3 sampai 7 adalah
kursi masing-masing partai, dan kolom 2 adalah jumlah kursi seluruhnya. Kolom
1 adalah tahun pemilihan.
BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
Kemudian statistiknya kita cetak dalam bentuk tabel. Kami menggunakan
nama sebagai header kolom, dan tahun sebagai header untuk baris. Lebar
default untuk kolom adalah wc=10, tetapi kami lebih memilih keluaran yang
lebih padat. Kolom akan diperluas untuk label kolom, jika perlu.
writetable(BT100,wc=6,dc=0, fixed,labc=P,labr=YT)
CDU/CSU SPD FDP Gr Li 1990 48 36 12 1 3 1994 44 38 7 7 4 1998 37 45
6 7 5 2002 41 42 8 9 0 2005 37 36 10 8 9 2009 38 23 15 11 12 2013 49 31 0 10 10
Perkalian matriks berikut ini menjumlahkan persentase dua partai besar
yang menunjukkan bahwa partai-partai kecil berhasil memperoleh suara di par-
lemen hingga tahun 2009.
BT1:=(BT.[1;1;0;0;0])'100
[84.29, 81.25, 81.1659, 82.7529, 72.9642, 61.8971, 79.8732]
Ada juga plot statistik sederhana. Kami menggunakannya untuk menampilkan
garis dan titik secara bersamaan. Alternatifnya adalah memanggil plot2d dua
kali dengan gt;add.
statplot(YT,BT1,"b"):
Tentukan beberapa warna untuk setiap pesta.
CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
Sekarang kita bisa memplot hasil pemilu 2009 dan perubahannya menjadi
satu plot dengan menggunakan gambar. Kita dapat menambahkan vektor kolom
ke setiap plot.
figure(2,1); ... figure(1); columnsplot(BW[6,3:7],P,color=CP); ... fig-
ure(2); columnsplot(BW[6,3:7]-BW[5,3:7],P,color=CP); ... figure(0):
Plot data menggabungkan deretan data statistik dalam satu plot.
J:=BW[,1]'; DP:=BW[,3:7]'; ... dataplot(YT,BT',color=CP); ... label-
box(P,colors=CP,styles="[]", points,w=0.2,x=0.3,y=0.4):
Plot kolom 3D memperlihatkan baris data statistik dalam bentuk kolom.
Kami memberikan label untuk baris dan kolom. sudut adalah sudut pandang.
columnsplot3d(BT,scols=P,srows=YT, ... angle=30°,ccols=CP):
Representasi lainnya adalah plot mosaik. Perhatikan bahwa kolom plot
mewakili kolom matriks di sini. Karena panjang label CDU/CSU, kami mengam-
bil jendela yang lebih kecil dari biasanya.
shrinkwindow( smaller); ... mosaicplot(BT',srows=YT,scols=P,color=CP,style="");
... shrinkwindow():
Kita juga bisa membuat diagram lingkaran. Karena hitam dan kuning mem-
bentuk koalisi, kami menyusun ulang elemen-elemennya.

```

```

i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):

```

Ini adalah jenis plot lainnya.

```

starplot(normal(1,10)+4,lab=1:10, rays):

```

Beberapa plot di plot2d bagus untuk statika. Berikut adalah plot impuls dari data acak, terdistribusi secara seragam di $[0,1]$.

```

plot2d(makeimpulse(1:10,random(1,10)), bar):

```

Namun untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritmik.

```

logimpulseplot(1:10,-log(random(1,10))10):

```

Fungsi Columnplot() lebih mudah digunakan, karena hanya memerlukan vektor nilai. Selain itu, ia dapat mengatur labelnya ke apa pun yang kita inginkan, kami telah mendemonstrasikannya di tutorial ini.

Ini adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan membuat statistik.

```

v=strtochar("the quick brown fox jumps over the lazy dog"); ... w=ascii("a"):ascii("z");
x=getmultiplicities(w,v); ... cw=[]; for k=w; cw=cw+char(k); end; ...
columnplot(x,lab=cw,width=0.05):

```

Dimungkinkan juga untuk mengatur sumbu secara manual.

```

n=10; p=0.4; i=0:n; x=bin(n,i)pi(1 - p)(n - i); ... columnsplot(x,lab =
i,width = 0.05, < frame, < grid); ... yaxis(0,0 : 0.1 : 1, style = "-", left); xaxis(0, style =
"."); ... label("p", 0, 0.25), label("i", 11, 0); ... textbox(["Binomialdistribution", "with p =
0.4"]) :

```

Berikut ini cara memplot frekuensi bilangan dalam suatu vektor.

Kami membuat vektor bilangan acak bilangan bulat 1 hingga 6.

```

v=intrandom(1,10,10)
[8, 5, 8, 8, 6, 8, 8, 3, 5, 5]

```

Kemudian ekstrak nomor unik di v.

```

vu:=unique(v)
[3, 5, 6, 8]

```

Dan plot frekuensi dalam plot kolom.

```

columnplot(getmultiplicities(vu,v),lab=vu,style="/"):

```

Kami ingin mendemonstrasikan fungsi distribusi nilai empiris.

```

x=normal(1,20);

```

Fungsi empdist(x,vs) memerlukan array nilai yang diurutkan. Jadi kita harus mengurutkan x sebelum kita dapat menggunakannya.

```

xs=sort(x);

```

Kemudian kita plot distribusi empiris dan beberapa batang kepadatan ke dalam satu plot. Alih-alih plot batang untuk distribusi kali ini kami menggunakan plot gigi gergaji.

```

figure(2,1); ... figure(1); plot2d("empdist",-4,4,xs); ... figure(2); plot2d(histo(x,v=-
4:0.2:4,i;bar)); ... figure(0):

```

Plot sebar mudah dilakukan di Euler dengan plot titik biasa. Grafik berikut menunjukkan bahwa X dan $X+Y$ jelas berkorelasi positif.

```

x=normal(1,100); plot2d(x,x+rotright(x), points,style=".."):

```

Seringkali kita ingin membandingkan dua sampel dengan distribusi yang berbeda. Hal ini dapat dilakukan dengan plot kuantil-kuantil.

Untuk pengujiannya, kami mencoba distribusi student-t dan distribusi eksponensial.

```
x=randt(1,1000,5); y=randnormal(1,1000,mean(x),dev(x)); ... plot2d("x",r=6,style="-",
",yl="normal",xl="student-t", vertical); ... plot2d(sort(x),sort(y), points,color=red,style="x", add):
```

Plot tersebut dengan jelas menunjukkan bahwa nilai terdistribusi normal cenderung lebih kecil di ujung ekstrim.

Jika kita mempunyai dua distribusi yang ukurannya berbeda, kita dapat memperluas distribusi yang lebih kecil atau mengecilkan distribusi yang lebih besar. Fungsi berikut ini baik untuk keduanya. Dibutuhkan nilai median dengan persentase antara 0 dan 1.

```
function medianexpand (x,n) := median(x,p=linspace(0,1,n-1));
Mari kita bandingkan dua distribusi yang sama.
x=random(1000); y=random(400); ... plot2d("x",0,1,style="-"); ...
plot2d(sort(medianexpand(x,400)),sort(y), points,color=red,style="x", add):
```

Regresi dan Korelasi

Regresi linier dapat dilakukan dengan fungsi polyfit() atau berbagai fungsi fit.

Sebagai permulaan kita menemukan garis regresi untuk data univariat dengan polyfit(x,y,1).

```
x=1:10; y=[2,3,1,5,6,3,7,8,9,8]; writetable(x'—y',labc=["x","y"])
x y 1 2 2 3 3 1 4 5 5 6 6 3 7 7 8 8 9 9 10 8
```

Kami ingin membandingkan kecocokan yang tidak berbobot dan berbobot. Pertama koefisien kecocokan linier.

```
p=polyfit(x,y,1)
[0.733333, 0.812121]
Regresi linear dapat ditulis dalam bentuk:
dengan
```

Kita hitung:

Maka:

Jadi, b, m = 0.733333, 0.812121

Sekarang koefisien dengan bobot yang menekankan nilai terakhir.

```
w = "exp(-(x-10)2/10)"; pw = polyfit(x,y,1,w = w(x))
[4.71566, 0.38319]
```

Kami memasukkan semuanya ke dalam satu plot untuk titik dan garis regresi, dan untuk bobot yang digunakan.

```
figure(2,1); ... figure(1); statplot(x,y,"b",xl="Regression"); ... plot2d("evalpoly(x,p)", add,color=blue,style="b"); ...
plot2d("evalpoly(x,pw)",5,10, add,color=red,style="-"); ... figure(2);
plot2d(w,1,10, filled,style="r",fillcolor=red,xl=w); ... figure(0):
```

Contoh lain kita membaca survei siswa, usia mereka, usia orang tua mereka dan jumlah saudara kandung dari sebuah file.

Tabel ini berisi "m" dan "f" di kolom kedua. Kami menggunakan variabel tok2 untuk mengatur terjemahan yang tepat alih-alih membiarkan readtable() mengumpulkan terjemahannya.

```
MS,hd:=readtable("table1.dat",tok2:["m","f"]); ... writetable(MS,labc=hd,tok2:["m","f"]);
Person Sex Age Mother Father Siblings 1 m 29 58 61 1 2 f 26 53 54 2 3 m
24 49 55 1 4 f 25 56 63 3 5 f 25 49 53 0 6 f 23 55 55 2 7 m 23 48 54 2 8 m 27 56
```

```
58 1 9 m 25 57 59 1 10 m 24 50 54 1 11 f 26 61 65 1 12 m 24 50 52 1 13 m 29
54 56 1 14 m 28 48 51 2 15 f 23 52 52 1 16 m 24 45 57 1 17 f 24 59 63 0 18 f 23
52 55 1 19 m 24 54 61 2 20 f 23 54 55 1
```

Bagaimana usia bergantung satu sama lain? Kesan pertama muncul dari plot sebar berpasangan.

```
scatterplots(tablecol(MS,3:5),hd[3:5]):
```

Jelas terlihat bahwa usia ayah dan ibu saling bergantung satu sama lain.

Mari kita tentukan dan plot garis regresinya.

```
cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1)
[17.3789, 0.740964]
```

Ini jelas merupakan model yang salah. Garis regresinya adalah $s=17+0,74t$, dengan t adalah umur ibu dan s adalah umur ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tapi tidak terlalu banyak.

Sebaliknya, kami mencurigai fungsi seperti $s=a+t$. Maka a adalah mean dari $s-t$. Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

```
da:=mean(cs[2]-cs[1])
3.65
```

Mari kita plot ini menjadi satu plot sebar.

```
plot2d(cs[1],cs[2], points); ... plot2d("evalpoly(x,ps)",color=red,style=".", add);
... plot2d("x+da",color=blue, add):
```

Berikut adalah plot kotak dari dua zaman tersebut. Ini hanya menunjukkan, bahwa usianya berbeda-beda.

```
boxplot(cs,["mothers","fathers"]):
```

Menariknya, perbedaan median tidak sebesar perbedaan mean.

```
median(cs[2])-median(cs[1])
1.5
```

Koefisien korelasi menunjukkan korelasi positif.

```
Koefisien korelasi menunjukkan korelasi positif.correl(cs[1],cs[2])
```

Variable Koefisien not found! Error in: Koefisien korelasi menunjukkan korelasi positif.correl(cs[1],c ...

Korelasi pangkat merupakan ukuran keteraturan yang sama pada kedua vektor. Hal ini juga cukup positif.

```
rankcorrel(cs[1],cs[2])
0.758925292358
```

Membuat Fungsi baru

Tentu saja, bahasa EMT dapat digunakan untuk memprogram fungsi-fungsi baru. Misalnya, kita mendefinisikan fungsi skewness.

```
m adalah rata-rata dari x.
```

```
function skew (x:vector) ...
```

```
m:=mean(x); return sqrt(cols(x))*sum((x-m)^3)/(sum((x-m)^2))^(3/2); endfunction <
```

```
/pre > SepertiyangAndalihat,kitadapatdenganmudahmenggunakanbahasamatriksuntukmendapatkanimpl
```

```
data=normal(20); skew(normal(10))
```

```
-0.198710316203
```

Berikut adalah fungsi lainnya, yang disebut koefisien skewness Pearson.

```
function skew1 (x) := 3(mean(x)-median(x))/dev(x)
skew1(data)
```

-0.0801873249135

Simulasi Monte Carlo

Kita simulasikan variabel acak berdistribusi normal 1000-5 sebanyak sejuta kali. Untuk ini, kita gunakan fungsi `normal(m,n)`, yang menghasilkan matriks nilai berdistribusi 0-1, atau `normal(n)` yang secara default bernilai `m=1`.

`n=1000000; x=normal(n)`

[-1.9943, 1.1795, 1.18591, -0.441316, -0.488192, -0.675366, -1.58021, 0.962882, -1.66963, -0.819503, -0.929235, 0.539573, -0.606864, -0.56878, -0.087391, -0.404174, 0.0788819, 1.04282, 0.876296, -0.877162, 0.689615, 1.66396, 1.10287, 1.81334, -0.679579, -0.739293, -0.0919214, -0.0516635, 1.68059, -0.8748, -0.661822, 1.14053, -0.0725198, -0.653134, 0.338853, 0.158539, -1.33566, -1.21098, 0.252185, 0.750436, -0.335005, 0.79995, 0.237049, 0.10056, -0.909771, -0.896389, 0.807295, 0.420089, -2.60022, -0.334883, 0.563393, -1.51976, -0.618486, -0.757591, -1.5655, -1.14804, -1.62101, -0.767098, 1.41355, 0.28399, 1.31699, 0.325665, 0.123923, -0.301708, 0.224152, 1.61033, -0.928959, 0.112323, 2.06755, -0.415012, 1.227, -0.392912, -0.82682, -1.37726, 1.09679, -2.83431, 0.310441, 1.333, -0.774371, 1.51449, 0.0260218, 0.244161, -1.25954, 0.833535, -1.05672, -3.99147, -1.70256, 0.620494, -0.98936, -0.868208, 0.344441, 0.926404, 0.00735312, -1.07317, 0.26722, 0.469612, -1.44652, -0.0850004, -1.22639, -0.120585, -0.498201, 0.793201, -2.71956, -0.832553, -1.50601, -0.473103, -0.857245, -0.397193, -0.368307, -1.08578, -1.60682, -0.525828, 0.00660302, -2.44655, 0.761081, -0.0405162, -0.440874, -0.733893, 0.275494, 1.14203, ...]

`n=1000000; x=normal(n)5+1000`

[1002.99, 1003.08, 996.596, 990.94, 998.608, 1005.05, 997.111, 1006.2, 1003.62, 988.073, 999.093, 1001.26, 999.921, 994.418, 995.893, 1008.1, 1000.85, 997.523, 997.748, 1000.46, 988.609, 1003.88, 996.944, 994.016, 1008.61, 997.575, 992.025, 993.416, 1003.18, 1003.39, 1001.73, 1005.57, 995.544, 1004.48, 996.802, 1004.15, 1005.39, 997.653, 1002.83, 1005.25, 998.235, 994.648, 998.584, 1000.88, 998.811, 994.607, 995.101, 995.2, 1002.16, 1009.75, 1000.77, 999.994, 1011.51, 1006.36, 990.598, 1007.2, 1000.18, 999.05, 1002.22, 1000.08, 993.234, 1000.81, 1004.78, 1002.42, 993.949, 1002.01, 995.443, 994.59, 1008.39, 1000.62, 995.42, 1000.23, 998.282, 998.84, 992.97, 1000.81, 993.34, 1003.17, 1001.89, 999.305, 1000.63, 992.78, 989.389, 998.162, 1001.79, 1008.18, 1003.34, 1001.35, 1001.08, 998.659, 998.586, 1000.31, 994.955, 994.737, 988.751, 991.572, 995.534, 999.851, 1004.17, 997.659, 993.513, 996.48, 999.944, 996.108, 998.717, 1002.66, 1001.39, 1003.17, 992.563, 998.983, 994.664, 997.407, 1001.87, 999.54, 993.469, 999.279, 1000.47, 996.123, 1000.8, 1003.22, 993.011, 998.241, 1000.07, 1001.6, 1002.9, 1004.22, 1002.22, 1005.54, 999.447, 1002.94, 1003.54, 999.995, 996.819, 1001.89, 1003.54, 1005.62, 1002.17, 998.897, 1005.83, 1006.59, ...]

terdapat juga fungsi `randnormal(n,m,mean,dev)`, yang dapat kita gunakan. Fungsi ini mematuhi skema penamaan "rand..." untuk generator acak.

`n=1000000; x=randnormal(1,n,1000,5)`

[1001.46, 1002.83, 995.424, 1010.42, 993.922, 995.65, 1002.73, 996.528, 998.885, 999.782, 999.165, 996.189, 1007.19, 1002.5, 1000.04, 1001.11, 995.786, 1000.87, 1007.36, 997.776, 989.569, 1000.06, 999.158, 996.89, 1000.07, 1000.75, 1003.66, 995.875, 1003.36, 992.127, 1003.03, 1006.05, 999.733, 999.09, 1005.97, 1003.15, 998.16, 989.205, 1002.45, 996.641, 1008.69, 1006.67, 992.93, 1003.96, 1002.56,

```

997.29, 995.598, 995.616, 1000.14, 1002.68, 984.552, 1004.04, 1008.83, 994.196,
991.508, 1003.67, 1000.44, 1001.38, 996.971, 1001.32, 998.596, 995.479, 1002.12,
990.516, 997.716, 1008.15, 1000.37, 1001.07, 996.014, 1001.07, 1003.76, 996.41,
992.867, 998.441, 993.741, 997.669, 1007.58, 1005.19, 997.211, 1000.75, 994.023,
994.398, 998.096, 994.683, 994.84, 1001.44, 1003.19, 987.72, 1003.07, 1008.01,
1009.52, 1007.39, 997.166, 1007.7, 987.807, 1002.73, 1003.18, 1002.86, 1002.75,
1001.91, 995.783, 998.36, 1002.57, 994.433, 1006.56, 1002.42, 997.431, 1005.47,
992.773, 1007.92, 1007.36, 999.482, 1003.88, 998.68, 1012.77, 1008.83, 1002.04,
1003.13, 1009.78, 997.305, 1003.33, 1001.45, 1004.12, 999.9, 1002.25, 996.054,
994.309, 1006.29, 999.89, 1003.39, 995.42, 990.872, 1000.26, 994.472, 997.125,
998.633, 994.745, 1001.02, 996.873, 1001.29, ... ]

```

10 nilai pertama x adalah

```
x[1:10]
```

```

[1001.46, 1002.83, 995.424, 1010.42, 993.922, 995.65, 1002.73, 996.528, 998.885,
999.782]

```

Distribusi dapat kita plot dengan flag `gt;distribution` dari `plot2d`.

```
plot2d(x, distribution); ... plot2d("qnormal(x,1000,5)",color=red,thickness=2, add):
```

kita juga dapat mengatur jumlah interval untuk distribusi menjadi 100. Kemudian kita akan melihat seberapa dekat kecocokan distribusi yang diamati dan distribusi yang sebenarnya. Bagaimanapun, kita telah menghasilkan satu juta kejadian.

```
plot2d(x,distribution=100); ... plot2d("qnormal(x,1000,5)",color=red,thickness=2, add):
```

kita dapat menghitung nilai rata-rata simulasi dan deviasinya harus sangat dekat dengan nilai yang diharapkan.

```
mean(x), dev(x)
```

```
999.996936044 4.99872526095
```

rumus nilai rata rata

```
xm=sum(x)/n
```

```
999.996936044
```

Rumus simpangan percobaannya (deviasi)

```
sqrt(sum((x-xm)2/(n - 1)))
```

```
4.99872526095
```

Perhatikan bahwa `x-xm` adalah vektor nilai yang dikoreksi, di mana `xm` dikurangi dari semua elemen vektor `x`.

Berikut adalah 10 nilai pertama `x-xm`.

```
short (x-xm)[1:10]
```

```

[1.4664, 2.8367, -4.573, 10.419, -6.075, -4.3474, 2.7294, -3.4693, -1.1123, -
0.21472]

```

Dengan menggunakan bahasa matriks, kita dapat dengan mudah menjawab pertanyaan lainnya. Misalnya, kita ingin menghitung proporsi `x` yang melebihi 1015.

Eksresi `xgt;=1015` menghasilkan vektor 1 dan 0. Menjumlahkan vektor ini menghasilkan jumlah kali `x[i]gt;=1015` terjadi.

```
sum(x =1015)/n
```

```
0.001339
```

Probabilitas yang diharapkan dari hal ini dapat dihitung dengan fungsi `normaldis(x)`. sehingga,

dimana X terdistribusi secara normal m-s.

```
1-normaldis(1015,1000,5)
0.00134989803163
```

cara kerja `gt;distribution` flag dari `plot2d` adalah menggunakan fungsi `histo(x)`, yang menghasilkan histogram frekuensi nilai dalam x . Fungsi ini mengembalikan batas interval dan jumlah dalam interval ini. Kami menormalkan jumlah untuk mendapatkan frekuensi.

```
t,s=histo(x,40); plot2d(t,s/n, bar):
```

Fungsi `histo()` juga dapat menghitung frekuensi dalam interval yang diberikan.

```
t,s=histo(x,v=[950,980,990,1010,1020,1050]); t, s,
[950, 980, 990, 1010, 1020, 1050] [25, 22795, 954608, 22545, 27]
```

hasil tersebut merupakan semua nilai acak yang berada antara 950 dan 1050.

menghitung total jumlah nilai dalam s , yang sama dengan total jumlah elemen dalam x

```
sum(s)
1000000
```

kita akan mensimulasikan 1000 kali lemparan 3 dadu, dan menanyakan pembagian jumlahnya.

```
ds:=sum(intrandom(1000,3,6)); fs=getmultiplicities(3:18,ds)
[3, 18, 23, 45, 65, 106, 108, 137, 136, 133, 88, 65, 36, 20, 11, 6]
```

kita akan plot hasil tersebut

```
columnplot(fs,lab=3:18):
```

kita akan menggunakan rekursi tingkat lanjut.

Fungsi berikut menghitung banyaknya cara bilangan k dapat direpresentasikan sebagai jumlah dari n bilangan dalam rentang 1 sampai m .

```
function map countways (k; n, m) ...
if n==1 then return k; k=1 k=m else sum=0; loop 1 to m; sum=sum+countways(k-
,n-1,m); end; return sum; end; endfunction ;/pre; Berikut hasil pelemparan dadu sebanyak lima kali.
```

```
countways(5:25,5,5)
[1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320, 255, 185, 121, 70, 35,
15, 5, 1]
```

```
cw=countways(3:18,3,6)
[1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3, 1]
```

Kita akan menambahkan nilai yang diharapkan ke plot.

```
plot2d(cw/6^3*1000, add);plot2d(cw/6^3*1000, points, add) :
```

Untuk simulasi lain, deviasi nilai rata-rata n 0-1-variabel acak terdistribusi normal adalah $1/\sqrt{n}$.

```
longformat; 1/sqrt(10)
0.316227766017
```

Mari kita periksa ini dengan simulasi. Kami menghasilkan 10.000 kali 10 vektor acak.

```
M=normal(10000,10); dev(mean(M)')
0.317942172219
```



```
plot2d(mean(M)', distribution):
```

Median dari 10 bilangan acak berdistribusi normal 0-1 mempunyai deviasi yang lebih besar.

Karena kita dapat dengan mudah menghasilkan jalan acak, kita dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami kemudian memplot deviasi standar dan rata-rata langkah ke-n dari proses ini bersama dengan nilai yang diharapkan berwarna merah.

```
n=1000; m=1000; M=cumsum(normal(n,m)/sqrt(m)); ... t=(1:n)/n; figure(2,1); ... figure(1); plot2d(t,mean(M'))'; plot2d(t,0,color=red, add); ... figure(2); plot2d(t,dev(M'))'; plot2d(t,sqrt(t),color=red, add); ... figure(0): uji chi-kuadrat
```

uji chi-kuadrat adalah alat penting dalam statistik. Di Euler, banyak tes yang diterapkan. Semua pengujian ini mengembalikan kesalahan yang kita terima jika kita menolak hipotesis nol.

Misalnya, kami menguji lemparan dadu untuk distribusi yang seragam. Pada 600 kali lemparan, kami mendapatkan nilai berikut, yang kami masukkan ke dalam uji chi-kuadrat.

```
chitest([90,103,114,101,103,89],dup(100,6)')
0.498830517952
```

Ini adalah nilai p-value dari uji chi-kuadrat

Uji chi-kuadrat juga memiliki mode yang menggunakan simulasi Monte Carlo untuk menguji statistiknya, menggunakan Parameter `gt;p` menafsirkan vektor `y` sebagai vektor probabilitas.

```
chitest([90,103,114,101,103,89],dup(1/6,6)', p, montecarlo)
0.505
```

Ini adalah p-value dari uji chi-kuadrat menggunakan pendekatan Monte Carlo. Dengan simulasi Monte Carlo, kita memperoleh p-value yang mirip dengan uji chi-kuadrat standar (0,4988 di uji pertama)

Selanjutnya kita menghasilkan 1000 lemparan dadu menggunakan generator angka acak, dan melakukan tes yang sama.

```
n=1000; t=random([1,n6]); chitest(count(t6,6),dup(n,6)')
0.558957010759
```

Mari kita uji nilai rata-rata 100 dengan uji-t.

```
s=200+normal([1,100])10; ... ttest(mean(s),dev(s),100,200)
0.25577040307
```

Fungsi `ttest()` memerlukan nilai mean, deviasi, jumlah data, dan nilai mean yang akan diuji.

Sekarang mari kita periksa dua pengukuran untuk mean yang sama. Kami menolak hipotesis bahwa keduanya mempunyai mean yang sama, jika hasilnya $lt;0,05$.

```
tcomparedata(normal(1,10),normal(1,10))
0.4848094938
```

Jika kita menambahkan bias pada satu distribusi, kita akan mendapatkan lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

```
tcomparedata(normal(1,10),normal(1,10)+2)
0.000205842407976
```

Menambah nilai 2 ke salah satu distribusi menyebabkan p-value menjadi sangat kecil.

Pada contoh berikutnya, kita membuat 20 lemparan dadu acak sebanyak 100 kali dan menghitung yang ada di dalamnya. Rata-rata harus ada $20/6=3,3$.

```
R=random(100,20); R=sum(R6i=1)'; mean(R)
```

```
3.46
```

Sekarang kita bandingkan jumlah satuan dengan distribusi binomial. Pertama kita plot distribusinya.

```
plot2d(R,distribution=max(R)+1,even=1,style="
/"):

```

kita akan Menghitung frekuensi kemunculan setiap jumlah angka "1" dalam 20 lemparan dadu acak yang telah dilakukan 100 kali

```
t=count(R,21);
```

Kemudian kami menghitung nilai yang diharapkan.

```
n=0:20; b=bin(20,n)(1/6)^n(5/6)^(20-n)100;
```

Kita harus mengumpulkan beberapa angka untuk mendapatkan kategori yang cukup besar.

```
t1=sum(t[1:2])—t[3:7]—sum(t[8:21]); ... b1=sum(b[1:2])—b[3:7]—sum(b[8:21]);
```

Uji chi-square menolak hipotesis bahwa distribusi kita merupakan distribusi binomial, jika hasilnya $lt;0,05$.

```
chitest(t1,b1)
```

```
0.855843823605
```

Contoh berikut berisi hasil dua kelompok orang (misalnya laki-laki dan perempuan) yang memilih satu dari enam partai.

```
A=[23,37,43,52,64,74;27,39,41,49,63,76]; ... writetable(A,wc=6,labr=["m","f"],labc=1:6)
```

```
1 2 3 4 5 6 m 23 37 43 52 64 74 f 27 39 41 49 63 76
```

Kita akan menguji independensi suara dari jenis kelamin.

```
tabletest(A)
```

```
0.990701632326
```

Berikut ini adalah tabel yang diharapkan, jika kita mengasumsikan frekuensi pemungutan suara yang diamati.

```
writetable(expectedtable(A),wc=6,dc=1,labr=["m","f"],labc=1:6)
```

```
1 2 3 4 5 6 m 24.9 37.9 41.9 50.3 63.3 74.7 f 25.1 38.1 42.1 50.7 63.7 75.3
```

Kita dapat menghitung koefisien kontingensi yang dikoreksi. Karena sangat mendekati 0, kami menyimpulkan bahwa pemungutan suara tidak bergantung pada jenis kelamin.

```
contingency(A)
```

```
0.0427225484717
```

```
uji F
```

Selanjutnya kita menggunakan analisis varians (uji F) untuk menguji tiga sampel data yang berdistribusi normal untuk nilai mean yang sama. Metode tersebut disebut ANOVA (analisis varians). Di Euler, fungsi `varanalysis()` digunakan.

```
x1=[109,111,98,119,91,118,109,99,115,109,94]; mean(x1),
```

```
106.545454545
```

```
x2=[120,124,115,139,114,110,113,120,117]; mean(x2),
```

```

119.111111111
x3=[120,112,115,110,105,134,105,130,121,111]; mean(x3)
116.3
varanalysis(x1,x2,x3)
0.0138048221371

```

Dengan p-value sebesar 0.0138 (1,38) bahwa ketiga sampel memiliki mean yang sama pada tingkat signifikansi 5 terdapat perbedaan yang signifikan antara mean dari setidaknya satu sampel.

Ada juga uji median, yang menolak sampel data dengan distribusi rata-rata yang berbeda, menguji median dari sampel yang disatukan.

```

a=[56,66,68,49,61,53,45,58,54]
[56, 66, 68, 49, 61, 53, 45, 58, 54]
b=[72,81,51,73,69,78,59,67,65,71,68,71]
[72, 81, 51, 73, 69, 78, 59, 67, 65, 71, 68, 71]
mediantest(a,b)
0.0241724220052

```

Tes kesetaraan lainnya adalah tes peringkat. Ini jauh lebih tajam daripada tes median.

```

ranktest(a,b)
0.00199969612469

```

Pada contoh berikut, kedua distribusi mempunyai mean yang sama.

```

ranktest(random(1,100),random(1,50)-1)
0.162380065554

```

ini menunjukkan bahwa perbedaan tidak cukup signifikan pada tingkat signifikansi 5 median yang sama tidak dapat ditolak.

Sekarang mari kita coba mensimulasikan dua perlakuan a dan b yang diterapkan pada orang yang berbeda.

```

a=[8.0,7.4,5.9,9.4,8.6,8.2,7.6,8.1,6.2,8.9];
b=[6.8,7.1,6.8,8.3,7.9,7.2,7.4,6.8,6.8,8.1];

```

Tes signum memutuskan, apakah a lebih baik dari b.

```

signtest(a,b)
0.0546875

```

Ini kesalahan yang terlalu besar untuk menolak hipotesis. Kita tidak dapat menolak bahwa a sama baiknya dengan b, Karena $p > 0.05$.

Uji Wilcoxon lebih tajam dibandingkan uji ini, namun mengandalkan nilai kuantitatif perbedaannya.

```

wilcoxon(a,b)
0.0296680599405

```

Mari kita coba dua tes lagi menggunakan rangkaian yang dihasilkan.

```

wilcoxon(normal(1,20),normal(1,20)-1)
9.45060349657e-05

```

ini menunjukkan bahwa ada perbedaan signifikan antara kedua sampel pada tingkat signifikansi 5

```

wilcoxon(normal(1,20),normal(1,20))
0.559353645673

```

hasil ini jauh di atas 0.05, sehingga kita tidak bisa menolak hipotesis bahwa kedua sampel berasal dari distribusi yang sama.

Angka Acak

Berikut ini adalah pengujian pembangkit bilangan acak. Euler menggunakan generator yang sangat bagus, jadi kita tidak perlu mengharapkan adanya masalah.

Pertama kita menghasilkan sepuluh juta angka acak di $[0,1]$.

```
n:=10000000; r:=random(1,n);
```

Selanjutnya kita hitung jarak antara dua angka yang kurang dari 0,05.

```
a:=0.05; d:=differences(nonzeros(rja));
```

Terakhir, kami memplot berapa kali, setiap jarak terjadi, dan membandingkannya dengan nilai yang diharapkan.

```
m=getmultiplicities(1:100,d); plot2d(m); ... plot2d("n(1-a)^(x-1)a^2",color = red, add) :
```

Hapus datanya.

```
remvalue n;
```

Kami ingin menghitung nilai rata-rata dan simpangan baku yang diukur.

```
M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ... mean(M), dev(M), 999.9 2.72641400622
```

Kita dapat membuat diagram kotak dan kumis untuk data tersebut. Dalam kasus kita, tidak ada outlier.

```
boxplot(M):
```

Kami menghitung probabilitas bahwa suatu nilai lebih besar dari 1005, dengan asumsi nilai terukur dan distribusi normal.

Semua fungsi untuk distribusi dalam Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

Kami mencetak hasilnya dalam fungsi cetak.

```
print((1-normaldis(1005,mean(M),dev(M)))100,2,unit="3.07
```

Untuk contoh berikutnya, kami mengasumsikan jumlah pria berikut dalam rentang ukuran tertentu.

```
r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Berikut adalah plot distribusinya.

```
plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="
```

```
/"):
```

Kita dapat memasukkan data mentah tersebut ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal rentang, akhir rentang, jumlah orang dalam rentang.

Tabel dapat dicetak dengan tajuk. Kita menggunakan vektor string untuk mengatur tajuk.

```
T:=r[1:8]' — r[2:9]' — v'; writetable(T,labc=["from","to","count"])
```

```
from to count 155.5 22 159.5 163.5 71 163.5 167.5 136 167.5 171.5 169 171.5 175.5 139 175.5 179.5 71 179.5 183.5 32 183.5 187.5 8
```

Jika kita memerlukan nilai rata-rata dan statistik ukuran lainnya, kita perlu menghitung titik tengah rentang. Kita dapat menggunakan dua kolom pertama tabel kita untuk ini.

```

(T[,1]+T[,2])/2
157.5 161.5 165.5 169.5 173.5 177.5 181.5 185.5
Namun lebih mudah untuk melipat rentang dengan vektor [1/2,1/2].
l=fold(r,[0.5,0.5])
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
Sekarang kita dapat menghitung rata-rata dan deviasi sampel dengan frekuensi
yang diberikan.
m,d=meandev(l,v); m, d,
169.901234568 5.98912964449
Mari kita tambahkan distribusi normal nilai-nilai tersebut ke plot.
plot2d("qnormal(x,m,d)sum(v)4", ... xmin=min(r),xmax=max(r),thickness=3,add=1):
Pengantar untuk Pengguna Proyek R
Jelasnya, EMT tidak bersaing dengan R sebagai paket statistik. Namun,
ada banyak prosedur dan fungsi statistik yang tersedia di EMT juga. Jadi
EMT dapat memenuhi kebutuhan dasar. Bagaimanapun, EMT hadir dengan
paket numerik dan sistem aljabar komputer.
Notebook ini cocok untuk Anda yang sudah familiar dengan R, namun perlu
mengetahui perbedaan sintaksis EMT dan R. Kami mencoba memberikan gam-
baran umum tentang hal-hal yang sudah jelas dan kurang jelas yang perlu Anda
ketahui.
Selain itu, kami mencari cara untuk bertukar data antara kedua sistem.
Note that this is a work in progress.
Sintaks Dasar
Hal pertama yang Anda pelajari di R adalah membuat vektor. Dalam EMT,
perbedaan utamanya adalah operator : dapat mengambil ukuran langkah. Se-
lain itu, ia mempunyai daya ikat yang rendah.
n:=10; 0:n/20:n-1
[0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9]
x:=[10.4, 5.6, 3.1, 6.4, 21.7]; [x,0,x]
[10.4, 5.6, 3.1, 6.4, 21.7, 0, 10.4, 5.6, 3.1, 6.4, 21.7]
Operator titik dua dengan ukuran langkah EMT digantikan oleh fungsi seq()
di R. Kita dapat menulis fungsi ini di EMT.
function seq(a,b,c) := a:b:c; ... seq(0,-0.1,-1)
[0, -0.1, -0.2, -0.3, -0.4, -0.5, -0.6, -0.7, -0.8, -0.9, -1]
function seq(a,b,c) := a:b:c; ... seq(0,-0.5,-5)
[0, -0.5, -1, -1.5, -2, -2.5, -3, -3.5, -4, -4.5, -5]
function rep(x:vector,n:index) := flatten(dup(x,n)); ... rep(x,2)
[10.4, 5.6, 3.1, 6.4, 21.7, 10.4, 5.6, 3.1, 6.4, 21.7]
Fungsi rep() dari R tidak ada di EMT. Untuk masukan vektor dapat dit-
uliskan sebagai berikut.
Perhatikan bahwa "=" atau "==" digunakan untuk tugas. Operator "-gt;"
digunakan untuk satuan dalam EMT.
125km - " miles"
77.6713990297 miles
Operator "lt;-" untuk penugasan memang bukan ide yang baik untuk R.
tetapi di EMT operator "lt;-" itu bukan penugasan melainkan perbandingan

```

Berikut ini akan membandingkan a dan -4 di EMT.

```
a:=2; a[-4]
```

```
0
```

EMT dan R memiliki vektor bertipe boolean. Namun dalam EMT, angka 0 dan 1 digunakan untuk mewakili salah dan benar. Di R, nilai benar dan salah tetap bisa digunakan dalam aritmatika biasa seperti di EMT.

```
xj5,  
[0, 0, 1, 0, 0] [0, 0, 3.1, 0, 0]
```

EMT memunculkan kesalahan atau menghasilkan NAN tergantung pada tanda "kesalahan".

```
errors off; 0/0, isNAN(sqrt(-1)), errors on;
```

```
NAN 1
```

Stringnya sama di R dan EMT. Keduanya berada di lokal saat ini, bukan di Unicode.

Di R ada paket untuk Unicode. Di EMT, string dapat berupa string Unicode. String unicode dapat diterjemahkan ke pengkodean lokal dan sebaliknya. Selain itu, u"... " dapat berisi entitas HTML.

```
u"169; Reneacut; Grothmann"
```

```
© René Grothmann
```

```
karakter khusus (hak cipta © dan karakter aksent é),  
chartoutf([480])
```

Berikut ini mungkin tidak ditampilkan dengan benar pada sistem sebagai A dengan titik dan garis di atasnya. Itu tergantung pada font yang Anda gunakan.

Penggabungan string dilakukan dengan "+" atau "—". Penggabungan ini akan menghasilkan string tunggal, dan angka yang digabungkan akan dikonversi otomatis ke format string. Ini dapat mencakup angka, yang akan dicetak dalam format saat ini.

```
"pi = "+pi
```

```
pi = 3.14159265359
```

```
Pengindeksan
```

Seringkali, ini akan berfungsi seperti di R.

Namun EMT akan menafsirkan indeks negatif dari belakang vektor, sementara R menafsirkan x[n] sebagai x tanpa elemen ke-n.

```
x, x[1:3], x[-2]  
[10.4, 5.6, 3.1, 6.4, 21.7] [10.4, 5.6, 3.1] 6.4
```

```
x, x[1:5], x[-3]  
[10.4, 5.6, 3.1, 6.4, 21.7] [10.4, 5.6, 3.1, 6.4, 21.7] 3.1
```

Untuk meniru perilaku R di EMT, kita dapat menggunakan fungsi drop(x,n)

```
drop(x,2)  
[10.4, 3.1, 6.4, 21.7]
```

Vektor logika tidak diperlakukan berbeda sebagai indeks di EMT, berbeda dengan R. Anda perlu mengekstrak elemen bukan nol terlebih dahulu di EMT.

```
x, x[5], x[nonzeros(x 5)]  
[10.4, 5.6, 3.1, 6.4, 21.7] [1, 1, 0, 1, 1] [10.4, 5.6, 6.4, 21.7]
```

Sama seperti di R, vektor indeks dapat berisi pengulangan.

```
x[[1,2,2,1]]
```

[10.4, 5.6, 5.6, 10.4]

Tipe Data

EMT memiliki lebih banyak tipe data tetap daripada R. Jelasnya, di R terdapat vektor yang berkembang. Anda dapat mengatur vektor numerik kosong `v` dan memberikan nilai ke elemen `v[17]`. Hal ini tidak mungkin dilakukan di EMT.

Berikut ini agak tidak efisien.

```
v=[]; for i=1 to 10000; v=v—i; end;
```

kenapa cara ini kurang efisien? karna setiap elemen baru di tambahkan EMT harus menyalin seluruh isi `v` kembali ke variabel `v`.

Semakin efisien vektor telah ditentukan sebelumnya.

```
v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Untuk mengubah tipe data di EMT, Anda dapat menggunakan fungsi seperti `kompleks()`.

```
kompleks(1:4)
```

```
[ 1+0i , 2+0i , 3+0i , 4+0i ]
```

Konversi ke string hanya dimungkinkan untuk tipe data dasar. Format saat ini digunakan untuk penggabungan string sederhana. Tapi ada fungsi seperti `print()` atau `frac()`.

Untuk vektor, Anda dapat dengan mudah menulis fungsi Anda sendiri.

```
function tostr (v) ...
```

```
s=""; loop 1 to length(v); s=s+print(v[],2,0); if jlength(v) then s=s+", ";  
endif; end; return s+""; endfunction ;/prej * Variabel s diinisialisasi sebagai  
string "" untuk menyimpan hasil * akhir. Awalnya, tanda kurung buka [  
ditambahkan ke variabel s sebagai * pembuka.
```

```
* loop 1 to length(v); menjalankan perulangan dari elemen pertama * hingga  
elemen terakhir dalam v. Fungsi length(v) mengembalikan panjang * atau jum-  
lah elemen dalam vektor v.
```

```
* print(v[], 2, 0); adalah fungsi format yang mengonversi elemen * vektor v  
pada posisi saat ini (v[]) menjadi string. * parameter 2 menunjukkan bahwa dua  
digit setelah titik desimal akan * ditampilkan, sementara 0 memastikan bahwa  
angka ditampilkan tanpa * tambahan simbol atau format lainnya.
```

```
* Bagian if lt;length(v) memeriksa apakah elemen saat ini bukan elemen *  
terakhir. Jika benar, maka koma , akan ditambahkan ke variabel s untuk *  
memisahkan elemen.
```

```
* Setelah loop selesai, tanda kurung tutup ] ditambahkan ke string s, * dan  
string ini kemudian dikembalikan sebagai output.
```

```
tostr(linspace(0,1,10));
```

Untuk komunikasi dengan Maxima, terdapat fungsi `convertmxm()`, yang juga dapat digunakan untuk memformat vektor untuk keluaran.

```
convertmxm(1:10);
```

Untuk Latex perintah `tex` dapat digunakan untuk mendapatkan perintah Latex.

```
tex([1,2,3]);
```

Faktor dan Tabel

Dalam pengantar R ada contoh yang disebut faktor.

Berikut ini adalah daftar wilayah 30 negara bagian.

```
austates = ["tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa", ... "qld",
"vic", "nsw", "vic", "qld", "qld", "sa", "tas", ... "sa", "nt", "wa", "vic",
"qld", "nsw", "nsw", "wa", ... "sa", "act", "nsw", "vic", "vic", "act"];
```

Perintah diatas digunakan untuk mendefinisikan sebuah array (array sendiri adalah sekumpulan variabel yang memiliki tipe data yang sama) karena pada data tersebut ada beberapa nama negara bagian yang terulang. Array ini berisi singkatan untuk negara bagian dan teritori di Australia.

Asumsikan, kita memiliki pendapatan yang sesuai di setiap negara bagian.

```
incomes = [60, 49, 40, 61, 64, 60, 59, 54, 62, 69, 70, 42, 56, ... 61, 61, 61,
58, 51, 48, 65, 49, 49, 41, 48, 52, 46, ... 59, 46, 58, 43];
```

Sekarang mari kita coba mencari nilai mean dan median dari data pendapatan tersebut menggunakan perintah `mean(incomes)` dan `median(incomes)`

```
mean(incomes)
54.7333333333
median(incomes)
57
```

Sekarang, kami ingin menghitung rata-rata pendapatan di suatu wilayah. Menjadi program statistik, R memiliki faktor() dan tapply() untuk ini.

EMT dapat melakukan hal ini dengan menemukan indeks wilayah dalam daftar wilayah unik.

```
auterr=sort(unique(austates)); f=indexofsorted(auterr,austates)
[6, 5, 4, 2, 2, 3, 8, 8, 4, 7, 2, 7, 4, 4, 5, 6, 5, 3, 8, 7, 4, 2, 2, 8, 5, 1, 2, 7, 7, 1]
```

Pada titik itu, kita dapat menulis fungsi perulangan kita sendiri untuk melakukan sesuatu hanya untuk satu faktor.

Atau kita bisa meniru fungsi tapply() dengan cara berikut.

```
function map tappl (i; f: call, cat, x)...
u=sort(unique(cat)); f=indexof(u,cat); return f(x[nonzeros(f == indexof(u,i))]); endfunction <
/pre > i : Parameterpertamabiasanyaadalahnilaiyangdigunakanuntukpencocokanataupemetaan.
f: call : Parameterkedua, yangkemungkinanbesaradalahsebuahfungsiyangdipanggildalamkodetersebut.
di sini merujuk pada fungsi yang diterima sebagai input.
```

cat: Parameter ketiga adalah array atau vektor yang berisi kategori yang akan diproses.

x: Parameter keempat adalah array atau vektor yang akan diproses atau diubah berdasarkan pemetaan kategori yang dilakukan.

Ini agak tidak efisien, karena menghitung wilayah unik untuk setiap i, tetapi berhasil.

```
tappl(auterr,"mean",austates,incomes)
[44.5, 57.3333333333, 55.5, 53.6, 55, 60.5, 56, 52.25]
Perhatikan bahwa ini berfungsi untuk setiap vektor wilayah.
tappl(["act", "nsw"], "mean", austates, incomes)
[44.5, 57.3333333333]
```

Sekarang, paket statistik EMT mendefinisikan tabel seperti di R. Fungsi `readtable()` dan `writetable()` dapat digunakan untuk input dan output.

Sehingga kita bisa mencetak rata-rata pendapatan negara di daerah secara bersahabat.


```
writetable(tappl(auterr,"mean",austates,incomes),labc=auterr,wc=7)
act nsw nt qld sa tas vic wa 44.5 57.33 55.5 53.6 55 60.5 56 52.25
```

Fungsi writetable digunakan untuk menampilkan data dalam bentuk tabel yang terstruktur dengan label kolom dan lebar kolom yang dapat disesuaikan.

Dengan labc=auterr, berarti menetapkan label kolom untuk tabel tersebut berdasarkan kategori yang ada di auterr(yang sudah diurutkan sesuai abjad).

wc(width of columns)=7 berarti setiap kolom dalam tabel akan memiliki lebar minimal 7 karakter.

sebagai contoh 44.5 itu memiliki 4 karakter (termasuk titik desimal).

karena data dalam kolom lebih pendek dari 7 karakter, kolom tersebut diberi ruang ekstra untuk tampilan yang rapi.

Kita juga bisa mencoba meniru perilaku R sepenuhnya.

Faktor-faktor tersebut harus disimpan dengan jelas dalam kumpulan beserta jenis dan kategorinya (negara bagian dan teritori dalam contoh kita). Untuk EMT, kami menambahkan indeks yang telah dihitung sebelumnya.

```
function makef (t) ...
```

Factor data Returns a collection with data t, unique data, indices. See: `tappl u=sort(unique(t)); return t,u,indexofsorted(u,t); endfunction` `i/pre> statef=makef(austates);`

Perintah `statef = makef(austates);` digunakan untuk mengolah data yang ada dalam variabel `austates`, dan mengidentifikasi elemen unik yang ada dalam data tersebut.

Sekarang elemen ketiga dari koleksi akan berisi indeks.

```
statef[3]
```

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

`statef[3]` adalah elemen ketiga dari koleksi yang dikembalikan oleh fungsi `makef`, yaitu indeks posisi dari elemen-elemen dalam `austates` yang sudah dipetakan ke urutan dalam `u` (data unik yang terurut).

`statef[3]` akan mengembalikan indeks posisi dari setiap elemen dalam `austates` berdasarkan urutan yang ada di `u`.

Sekarang kita bisa meniru `tappl()` dengan cara berikut. Ini akan mengembalikan tabel sebagai kumpulan data tabel dan judul kolom.

```
function tappl (t:vector,tf,f: call)...
```

```
Makes a table of data and factors  tf: output of makef() See: makef uf=tf[2];
f=tf[3]; x=zeros(length(uf)); for i=1 to length(uf); ind=nonzeros(f==i); if length(ind)==0
then x[i]=NAN; else x[i]=f(t[ind]); end i f; end; return x, u f; endfunction < /pre >
```

Kami tidak menambahkan banyak pengecekan tipedisini. Satu-satunya tindakan pencegahan menyangkut kategori

Tabel ini dapat dicetak sebagai tabel dengan `writetable()`.

```
writetable(tappl(incomes,statef,"mean"),wc=7)
```

```
act nsw nt qld sa tas vic wa 44.5 57.33 55.5 53.6 55 60.5 56 52.25
```

Array

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Namun, akan mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau perpustakaan C untuk ini.

R memiliki lebih dari dua dimensi. Di R array adalah vektor dengan bidang dimensi.

Dalam EMT, vektor adalah matriks dengan satu baris. Itu dapat dibuat menjadi matriks dengan `redim()`.

```
shortformat; X=redim(1:20,4,5)
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

Fungsi `shortformat` digunakan untuk mengatur format tampilan angka agar lebih ringkas dan mudah dibaca.

Perintah diatas digunakan untuk membuat matrik X dari angka 1 sampai 20 dengan ketentuan matriks dengan 4 baris dan 5 kolom.

Ekstraksi baris dan kolom, atau sub-matriks, mirip dengan R.

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

```
2 3 7 8 12 13 17 18
```

Perintah diatas digunakan untuk menampilkan matriks X kolom kedua sampai ketiga.

```
X[,3:5]
```

```
3 4 5 8 9 10 13 14 15 18 19 20
```

Namun, di R dimungkinkan untuk menyetel daftar indeks vektor tertentu ke suatu nilai. Hal yang sama mungkin terjadi di EMT hanya dengan satu putaran.

```
function setmatrixvalue(M, i, j, v) ...
```

```
loop 1 to max(length(i),length(j),length(v)) M[i,j] = v; end; endfunction
```

`;/pre>` Perintah `setmatrixvalue(M, i, j, v)` adalah fungsi yang digunakan untuk mengubah nilai elemen-elemen dalam matriks berdasarkan indeks tertentu.

M: Matriks yang akan dimodifikasi.

i: Indeks baris atau posisi baris dalam matriks M yang ingin diubah.

j: Indeks kolom atau posisi kolom dalam matriks M yang ingin diubah.

v: Nilai yang akan dimasukkan ke dalam elemen-elemen matriks M pada posisi yang ditentukan oleh indeks i dan j.

Kami mendemonstrasikan ini untuk menunjukkan bahwa matriks dilewatkan dengan referensi di EMT. Jika Anda tidak ingin mengubah matriks M asli, Anda perlu menyalinnya ke dalam fungsi.

```
setmatrixvalue(X,1:3,3:-1:1,0); X,
```

```
1 2 0 4 5 6 0 8 9 10 0 12 13 14 15 16 17 18 19 20
```

Perkalian luar dalam EMT hanya dapat dilakukan antar vektor. Ini otomatis karena bahasa matriks. Satu vektor harus berupa vektor kolom dan vektor lainnya harus berupa vektor baris.

```
(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
```

1:5: Ini adalah vektor baris yang berisi angka-angka dari 1 hingga 5

(1:5)': Tanda ' di sini menunjukkan transposisi dari vektor baris 1:5. Dengan kata lain, ini mengubah vektor baris menjadi vektor kolom.

Dalam PDF pendahuluan untuk R terdapat contoh yang menghitung distribusi ab-cd untuk a,b,c,d yang dipilih dari 0 hingga n secara acak. Solusi dalam R adalah membentuk matriks 4 dimensi dan menjalankan `table()` di atasnya.

Tentu saja, hal ini dapat dicapai dengan satu putaran. Tapi loop tidak efektif di EMT atau R. Di EMT, kita bisa menulis loop di C dan itu akan menjadi solusi tercepat.

Namun kita ingin meniru perilaku R. Untuk melakukannya, kita perlu meratakan perkalian ab dan membuat matriks ab-cd.

```
a=0:6; b=a'; p=flatten(ab); q=flatten(p-p'); ... u=sort(unique(q)); f=getmultiplicities(u,q);
... statplot(u,f,"h");
```

Selain multiplisitas eksak, EMT dapat menghitung frekuensi dalam vektor.

```
getfrequencies(q,-50:10:50)
```

```
[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]
```

Perintah diatas digunakan untuk menghitung distribusi frekuensi nilai-nilai dalam vektor q dalam rentang dari -50 hingga 50, dengan interval 10. Fungsi ini menghitung berapa banyak nilai dalam q yang jatuh dalam setiap interval: [-50, -40), [-40, -30), ..., [40, 50).

Cara paling mudah untuk memplotnya sebagai distribusi adalah sebagai berikut.

```
plot2d(q,distribution=11):
```

Namun dimungkinkan juga untuk menghitung terlebih dahulu penghitungan dalam interval yang dipilih sebelumnya. Tentu saja, berikut ini menggunakan getfrequencies() secara internal.

Karena fungsi histo() mengembalikan frekuensi, kita perlu menskalakannya sehingga integral di bawah grafik batang adalah 1.

```
x,y=histo(q,v=-55:10:55); y=y/sum(y)/differences(x); ... plot2d(x,y, bar,style="/" ):
```

Daftar

EMT memiliki dua jenis daftar. Salah satunya adalah daftar global yang bisa berubah, dan yang lainnya adalah tipe daftar yang tidak bisa diubah. Kami tidak peduli dengan daftar global di sini.

Tipe daftar yang tidak dapat diubah disebut koleksi di EMT. Ini berperilaku seperti struktur di C, tetapi elemennya hanya diberi nomor dan tidak diberi nama.

1. Membuat list dan mengakses elemen dalam list

```
L="Fred","Flintstone",40,[1990,1992]
```

```
Fred Flintstone 40 [1990, 1992]
```

Perintah diatas digunakan untuk membuat list L dengan nama depan Fred, nama belakang Flintstone, usia 40, dan tahun 1990, 1992.

Namun untuk tahun tersebut tidak dapat dipastikan apa arti dari tahun-tahun tersebut, bisa saja tahun kelahiran dan kematian, tahun pendidikan, tahun pekerjaan, atau yang lainnya.

Saat ini unsur-unsur tersebut tidak memiliki nama, meskipun nama dapat ditetapkan untuk tujuan khusus. Mereka diakses dengan nomor.

```
(L[4])[2]
```

```
1992
```

Perintah diatas digunakan untuk menampilkan list L keempat urutan kedua. Karena pada list L keempat berisi tahun yang dimana terdapat 2 tahun, tahun pertama adalah 1990 dan tahun kedua adalah 1992. Perintah tersebut ingin menampilkan tahun kedua, maka outputnya adalah 1992.

2. Menggabungkan dua list

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

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

```

B := [4,5,6]
[4, 5, 6]
C := [A, B]
[1, 2, 3, 4, 5, 6]
3. Mengubah elemen dalam list

```

```

D := [7,8,9,10]
[7, 8, 9, 10]
D[3] := 99
[7, 8, 99, 10]
4. menghitung panjang list
E := [10,20,30,40,50,60,70]
[10, 20, 30, 40, 50, 60, 70]
len := length(E)
7

```

File Input dan Output (Membaca dan Menulis Data)

Anda sering kali ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberi tahu Anda tentang banyak cara untuk mencapai hal ini. Fungsi sederhananya adalah `writematrix()` dan `readmatrix()`.

Mari kita tunjukkan cara membaca dan menulis vektor real ke file.

```

a=random(1,100); mean(a), dev(a),
0.44779 0.29446

```

Untuk menulis data ke file, kita menggunakan fungsi `writematrix()`.

Karena pengenalan ini kemungkinan besar ada di direktori, di mana pengguna tidak memiliki akses tulis, kami menulis data ke direktori home pengguna. Untuk buku catatan sendiri, hal ini tidak diperlukan, karena file data akan ditulis ke dalam direktori yang sama.

```

filename="test.dat";

```

Sekarang kita menulis vektor kolom `a'` ke file. Ini menghasilkan satu nomor di setiap baris file.

```

writematrix(a',filename)

```

Untuk membaca data, kita menggunakan `readmatrix()`

```

a=readmatrix(filename)'

```

```

[0.74157, 0.13184, 0.082656, 0.1975, 0.63019, 0.51127, 0.98091, 0.020501,
0.29063, 0.83916, 0.051539, 0.45459, 0.77198, 0.8754, 0.49257, 0.96765, 0.48685,
0.11645, 0.086971, 0.35536, 0.69515, 0.19161, 0.66654, 0.61295, 0.38779, 0.16748,
0.65044, 0.64784, 0.17712, 0.048322, 0.24795, 0.8682, 0.49634, 0.9125, 0.079817,
0.6544, 0.66279, 0.54949, 0.10807, 0.71727, 0.75473, 0.34844, 0.38634, 0.40978,
0.3234, 0.32117, 0.85977, 0.78211, 0.12933, 0.39591, 0.018822, 0.95697, 0.26985,
0.2409, 0.52222, 0.79655, 0.30484, 0.53489, 0.014971, 0.22246, 0.13021, 0.82292,
0.83054, 0.99006, 0.064585, 0.57221, 0.16539, 0.31818, 0.16037, 0.71343, 0.6188,
0.67365, 0.22367, 0.86965, 0.67439, 0.859, 0.85639, 0.07405, 0.65051, 0.79298,
0.039088, 0.4536, 0.24248, 0.10425, 0.36493, 0.084869, 0.37012, 0.045487, 0.66347,
0.69294, 0.46928, 0.27625, 0.20081, 0.069966, 0.2009, 0.52929, 0.97226, 0.52319,
0.11857, 0.075214]

```

Dan hapus file tersebut.

```

fileremove(filename);

```

```

    mean(a), dev(a),
    0.44779 0.29446
    Fungsi writematrix() atau writetable() dapat dikonfigurasi untuk bahasa
    lain.
    Misalnya, jika Anda memiliki sistem Indonesia (titik desimal dengan koma),
    Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan dengan
    titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan koma). File
    berikut "test.csv" akan muncul di folder saat ini Anda.
        filename="test.csv"; ... writematrix(random(5,3),file=filename,separator=",")
    Anda sekarang dapat membuka file ini dengan Excel bahasa Indonesia secara
    langsung.
        fileremove(filename);
    Terkadang kita memiliki string dengan token seperti berikut.
    s1:="f m m f m m m f f f m m f"; ... s2:="f f f m m f f";
    Untuk melakukan tokenisasi ini, kami mendefinisikan vektor token.
    tok:=["f","m"]
    f m
    Kemudian kita dapat menghitung berapa kali setiap token muncul dalam
    string, dan memasukkan hasilnya ke dalam tabel.
    M:=getmultiplicities(tok,strtokens(s1))-... getmultiplicities(tok,strtokens(s2));
    Tulis tabel dengan header token.
    writetable(M,labc=tok,labr=1:2,wc=8)
    f m 1 6 7 2 5 2
    Untuk statika, EMT dapat membaca dan menulis tabel.
    file="test.dat"; open(file,"w"); ... writeln("A,B,C"); writematrix(random(3,3));
... close();
    The file looks like this.
    printfile(file)
    A,B,C 0.5894214399628446,0.1500948494104958,0.4911562128682984 0.2817236329179764,0.3586251183800
0.2135992153342503,0.3457575959104093,0.3271359940090028
    Fungsi readtable() dalam bentuknya yang paling sederhana dapat membaca
    ini dan mengembalikan kumpulan nilai dan baris judul.
    L=readtable(file, list);
    Koleksi ini dapat dicetak dengan writetable() ke buku catatan, atau ke file.
    writetable(L,wc=10,dc=5)
    A B C 0.58942 0.15009 0.49116 0.28172 0.35863 0.1159 0.2136 0.34576 0.32714
    Matriks nilai adalah elemen pertama dari L. Perhatikan bahwa mean() di
    EMT menghitung nilai rata-rata baris matriks.
    mean(L[1])
    0.41022 0.25208 0.2955
    File CSV
    Pertama, mari kita menulis matriks ke dalam file. Untuk outputnya, kami
    membuat file di direktori kerja saat ini.
    file="test.csv"; ... M=random(3,3); writematrix(M,file);
    Here is the content of this file.
    printfile(file)

```

```
0.5537619367566726,0.6738419762237779,0.04291882768650555 0.2401989793483695,0.0861139922960146,0.
0.1343166294586442,0.3616257867488041,0.3383621511798272
```

CVS ini dapat dibuka pada sistem berbahasa Inggris ke Excel dengan klik dua kali. Jika Anda mendapatkan file seperti itu di sistem Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal.

Namun titik desimal juga merupakan format default untuk EMT. Anda dapat membaca matriks dari file dengan `readmatrix()`.

```
readmatrix(file)
```

```
0.55376 0.67384 0.042919 0.2402 0.086114 0.36787 0.13432 0.36163 0.33836
```

Dimungkinkan untuk menulis beberapa matriks ke satu file. Perintah `open()` dapat membuka file untuk ditulis dengan parameter "w". Standarnya adalah "r" untuk membaca.

```
open(file,"w"); writematrix(M); writematrix(M'); close();
```

Matriks dipisahkan oleh garis kosong. Untuk membaca matriks, buka file dan panggil `readmatrix()` beberapa kali.

```
open(file); A=readmatrix(); B=readmatrix(); A==B, close();
```

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

Di Excel atau spreadsheet serupa, Anda dapat mengekspor matriks sebagai CSV (nilai yang dipisahkan koma). Di Excel 2007, gunakan "save as" dan "other format", lalu pilih "CSV". Pastikan tabel saat ini hanya berisi data yang ingin Anda ekspor.

Ini sebuah contoh.

```
printfile("excel-data.csv")
```

```
0;1000;1000 1;1051,271096;1072,508181 2;1105,170918;1150,273799 3;1161,834243;1233,67806
4;1221,402758;1323,129812 5;1284,025417;1419,067549 6;1349,858808;1521,961556
7;1419,067549;1632,31622 8;1491,824698;1750,6725 9;1568,312185;1877,610579 10;1648,721271;2013,752707
```

Seperti yang Anda lihat, sistem bahasa Jerman saya menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubahnya di pengaturan sistem atau di Excel, tetapi hal ini tidak diperlukan untuk membaca matriks menjadi EMT.

Cara termudah untuk membaca ini ke dalam Euler adalah `readmatrix()`. Semua koma diganti dengan titik dengan parameter `gt;koma`. Untuk CSV bahasa Inggris, hilangkan saja parameter ini.

```
M=readmatrix("excel-data.csv", comma)
```

```
0 1000 1000 1 1051.3 1072.5 2 1105.2 1150.3 3 1161.8 1233.7 4 1221.4 1323.1
5 1284 1419.1 6 1349.9 1522 7 1419.1 1632.3 8 1491.8 1750.7 9 1568.3 1877.6 10
1648.7 2013.8
```

Let us plot this.

```
plot2d(M'[1],M'[2:3], points,color=[red,green]');
```

Ada cara yang lebih mendasar untuk membaca data dari suatu file. Anda dapat membuka file dan membaca angka baris demi baris. Fungsi `getvectorline()` akan membaca angka dari sebaris data. Secara default, ini mengharapkan titik desimal. Tapi bisa juga menggunakan koma desimal, jika Anda memanggil `setdecimaldot(",")` sebelum Anda menggunakan fungsi ini.

Fungsi berikut adalah contohnya. Itu akan berhenti di akhir file atau baris kosong.

```
function myload (file) ...
open(file); M=[]; repeat until eof(); v=getvectorline(3); if length(v)~0 then
M=M_v; else break; end; end; return M; close(file); endfunction < /pre > myload(file)
0.55376 0.67384 0.042919 0.2402 0.086114 0.36787 0.13432 0.36163 0.33836
```

Dimungkinkan juga untuk membaca semua angka dalam file itu dengan
getvector().

```
open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
0.55376 0.67384 0.042919 0.2402 0.086114 0.36787 0.13432 0.36163 0.33836
```

Oleh karena itu sangat mudah untuk menyimpan suatu vektor nilai, satu nilai di setiap baris dan membaca kembali vektor ini.

```
v=random(1000); mean(v)
0.493
writematrix(v',file); mean(readmatrix(file)')
0.493
```

Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Misalnya, kita menulis tabel dengan header baris dan kolom ke sebuah file.

```
file="test.tab"; M=random(3,3); ... open(file,"w"); ... writetable(M,separator=",",labc=["one","two"],"
... close(); ... printfile(file)
```

```
one,two,three 0.56, 0.63, 0.59 0.45, 0.86, 0.44 0.72, 0.19, 0.91
```

Ini dapat diimpor ke Excel.

Untuk membaca file di EMT, kami menggunakan readtable().

```
M,headings=readtable(file, clabs); ... writetable(M,labc=headings)
```

```
one two three 0.56 0.63 0.59 0.45 0.86 0.44 0.72 0.19 0.91
```

Menganalisis Garis

Pada subbab ini sering digunakan untuk memproses atau mengekstrak data dari teks yang berformat khusus, seperti data tabel dalam HTML. Anda bahkan dapat mengevaluasi setiap baris dengan tangan. Misalkan, kita memiliki baris dengan format berikut.

```
line="2020-11-03,Tue,1'114.05"
```

```
2020-11-03,Tue,1'114.05
```

Pertama, kita akan memisahkan string line menjadi bagian-bagian yang lebih kecil, yang dikenal sebagai "token".

```
vt=strtokens(line)
```

```
2020-11-03 Tue 1'114.05
```

Kemudian kita dapat mengevaluasi setiap elemen garis menggunakan evaluasi yang sesuai.

```
day(vt[1]); ... indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2]));
... strrepl(vt[3],"","")();
```

Dengan menggunakan ekspresi reguler, dimungkinkan untuk mengekstrak hampir semua informasi dari sebaris data.

Selanjutnya, kita akan melihat bagaimana mengekstrak data string yang berisi markup HTML menggunakan ekspresi reguler.

```
line="jtr jtd 1145.45j/td jtd 5.6j/td jtd -4.5j/td jtr";
```

Untuk mengekstraknya, kami menggunakan ekspresi reguler, yang mencari - tanda kurung tutup gt;;, untuk mengindikasikan bahwa kita akan

- mencari awal dari elemen yang ada di dalam tag.
- string apa pun yang tidak mengandung tanda kurung akan mencocokkan elemen di dalam tag `lt;tdgt;`.
- braket pembuka dan penutup menggunakan solusi terpendek, dengan tag pembuka `(lt;tdgt;)` dan penutup `(lt;/tdgt;)`.
- sekali lagi string apa pun yang tidak mengandung tanda kurung, ini akan menjamin bahwa kita akan mengambil isi yang relevan di dalam tagnya.
- dan tanda kurung buka `lt;` menandai bahwa ini adalah akhir dari tag dan awal dari tag baru.

Mencari pola tertentu dalam string line yang menggunakan ekspresi reguler.
`pos,s,v=strxfind(line,"([<]+)<.+?([<]+)<");`

Hasilnya adalah posisi kecocokan, string yang cocok, dan vektor string untuk sub-kecocokan.

Kita akan mengeksekusi elemen-elemen di dalam array atau list `vt` satu per satu dalam sebuah perulangan.

```
for k=1:length(vt); vt[k](), end;
1145.5 5.6
```

Berikut adalah fungsi yang membaca semua item numerik antara `lt;tdgt;` dan `lt;/tdgt;`.

```
function readtd (line) ...
v=[]; cp=0; repeat pos,s,v=strxfind(line,"jtd.*?;(.+?)i/td;",cp); until pos==0;
if length(vt);0 then v=v—vt[1]; endif; cp=pos+strlen(s); end; return v; end-
function j/pre; Kita akan mengekstrak dan menampilkan semua nilai yang be-
rada di antara tag lt;tdgt;...lt;/tdgt; dalam baris, dan mencari apakah nilai terse-
but numerik atau bukan.
```

```
readtd(line+"jtd non-numericali/td ")
1145.45 5.6 -4.5 non-numerical
```

Membaca dari Web

Situs web atau file dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris.

Dalam contoh, kita membaca versi terkini dari situs EMT. Kami menggu-
nakan ekspresi reguler untuk memindai "Versi ..." dalam sebuah judul.

```
function readversion () ...
urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html"); re-
peat until urloef(); s=urlgetline(); k=strfind(s,"Version ",1); if k;0 then substring(s,k,strfind(s,"j",k)-
1), break; endif; end; urlclose(); endfunction j/pre; readversion
Version 2024-01-12
```

Contoh lain membaca URL dengan EMT

```
"https://mywebsite.com/version.h"
function readversionmywebsite () ...
urlopen("https://mywebsite.com/version.h"); repeat until urloef(); s=urlgetline();
k=strfind(s,"Release",1); if k;0 then substring(s,k,strfind(s,"j",k)-1); break; en-
dif; end; urlclose(); endfunction j/pre; readversionmywebsite
```

Karena string "Release" tidak ada di dalam file `version.h`, maka `strfind(s, "Release", 1)` akan mengembalikan nilai nol atau tidak menghasilkan indeks yang diperlukan untuk proses pencarian.

Input dan Output Variabel

Anda dapat menulis variabel dalam bentuk definisi Euler ke file atau ke baris perintah.

```
writevar(pi,"mypi");
mypi = 3.141592653589793;
```

Untuk pengujian, kami membuat file Euler di direktori kerja EMT.

```
file="tes.e"; ... writevar(random(2,2),"M",file); ... printfile(file,3)
M = [ .. 0.3888176479607355, 0.3870125417113612; 0.2576115973745406,
0.1750949536837037];
```

Sekarang kita dapat memuat file tersebut. Ini akan mendefinisikan matriks M.

```
load(file); show M,
M = 0.38882 0.38701 0.25761 0.17509
```

Omong-omong, jika `writevar()` digunakan pada suatu variabel, definisi variabel dengan nama variabel tersebut akan dicetak.

```
writevar(M); writevar(inch)
M = [ .. 0.3888176479607355, 0.3870125417113612; 0.2576115973745406,
0.1750949536837037]; inch= 0.0254;
```

Kita juga bisa membuka file baru atau menambahkan file yang sudah ada.

Dalam contoh kita menambahkan file yang dibuat sebelumnya.

```
open(file,"a"); ... writevar(random(2,2),"M1"); ... writevar(random(3,1),"M2");
... close();
load(file); show M1; show M2;
M1 = 0.17881 0.026345 0.63559 0.34332 M2 = 0.22274 0.91626 0.4616
Untuk menghapus file apa pun, gunakan fileremove().
fileremove(file);
```

Vektor baris dalam suatu file tidak memerlukan koma, jika setiap angka berada pada baris baru. Mari kita buat file seperti itu, tulis setiap baris satu per satu dengan `writeln()`.

```
open(file,"w"); writeln("M = ["); ... for i=1 to 5; writeln(" "+random());
end; ... writeln("]"); close(); ... printfile(file)
M = [ 0.743886392671 0.92708388148 0.46506977432 0.0265574069961 0.147047179518
];
load(file); M
[0.74389, 0.92708, 0.46507, 0.026557, 0.14705]
```

LATIHAN

1. Misalkan anda memiliki vektor $x=[2,4,6,8,10]$

a. buatlah vektor yang menggabungkan vektor x , angka 0 dan vektor x lagi

b. tentukan apakah setiap elemen vektor x lebih besar dari 5 (hasil logika 1 untuk benar dan 0 untuk salah)

```
x:=[2,4,6,8,10]; [x,0,x]
[2, 4, 6, 8, 10, 0, 2, 4, 6, 8, 10]
x 5,
[0, 0, 1, 1, 1] [0, 0, 6, 8, 10]
```

2. Tentukan matriks X dengan elemen-elemen yang berurutan dari 1 hingga 20 dan susunlah elemen tersebut menjadi matriks berukuran 5×4 .

```

shortformat; X=redim(1:20,5,4)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
3.Seorang analis memiliki data penjualan harian selama 5 hari(150,200,250,300,350)
yang disimpan dalam bentuk vektor sebagai berikut:
a. mean(rata-rata)
b. deviasi standar
penjualan=[150,200,250,300,350]
[150, 200, 250, 300, 350]
atau anda bisa memanggil data yang sudah dibuat
filename="penjualan.dat";
writematrix(penjualan,filename)
penjualan=readmatrix(filename)'
[150, 200, 250, 300, 350]
mean(penjualan)
250
dev(penjualan)
79.057
4. Buat fungsi yang membuka URL
"https://en.wikipedia.org/wiki/Euler(software)"
dan mencari kata "Versi" di dalam URL tersebut, dan tampilkan hasilnya.
function readversionwebsite () ...
urlopen("https://en.wikipedia.org/wiki/Euler(software)");repeatuntilurleof(); s =
urlgetline(); k = strfind(s,"version",1); if k > 0 then substring(s,k, strfind(s," <
",k) - 1), break; end if; end; urlclose(); endfunction < /pre > readversion
Version 2024-01-12
5.Diberikan data pengukuran tinggi badan pada kelas matematika B adalah
sebagai berikut:
— Rentang Tinggi (cm) — Jumlah Orang — —————
— — 155.5 - 159.5 — 22 — — 159.5 - 163.5 — 71 — — 163.5 - 167.5 — 136
— — 167.5 - 171.5 — 169 — — 171.5 - 175.5 — 139 — — 175.5 - 179.5 — 71
— — 179.5 - 183.5 — 32 — — 183.5 - 187.5 — 8 —
a.)Hitung rata-rata dan deviasi standar dari distribusi tinggi badan ini.
b.)Plot distribusi frekuensi data (diagram batang).
c.)Tambahkan kurva distribusi normal untuk dibandingkan dengan data.
r = 155.5:4:187.5 //Rentang ukuran tinggi badan
[155.5, 159.5, 163.5, 167.5, 171.5, 175.5, 179.5, 183.5, 187.5]
v = [22, 71, 136, 169, 139, 71, 32, 8] //Jumlah orang dalam tiap rentang
[22, 71, 136, 169, 139, 71, 32, 8]
l=fold(r,[0.5,0.5]) //Menghitung titik tengah dari setiap rentang tinggi badan
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
m,d=meandev(l,v); m, d, //Hitung rata-rata dan deviasi standar
169.9 5.9891
plot2d(r, v, a=150, b=200, c=0, d=190, bar=1, style="
/"):
plot2d("qnormal(x, m, d) sum(v) 4", ... xmin=min(r), xmax=max(r),
thickness=3, add=1):

```

```

remvalue();
6. Sebuah survei dilakukan untuk mengetahui jumlah jam belajar siswa
SMA dalam satu minggu. Berikut data jam belajar dari 10 siswa: 8, 10, 7, 6,
9, 10, 11, 9, 8, 12.
a) Hitung nilai rata-rata dari data di atas
b) Tentukan median dari data tersebut.
M=[8,10,7,6,9,10,11,9,8,12];
mean(M)
9
median(M)
9
7. Anda diberikan data yang menunjukkan jumlah penjualan barang selama
12 bulan dalam satu tahun berturut-turut 120, 135, 150, 160, 170, 180, 190,
210, 200, 220, 230, 240.
a) Buatlah plot garis dari data penjualan barang tersebut.
b) Hitung rata-rata penjualan perbulan.
X=[120,135,150,160,170,180,190,210,200,220,230,240]
[120, 135, 150, 160, 170, 180, 190, 210, 200, 220, 230, 240]
Y=[1,2,3,4,5,6,7,8,9,10,11,12]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
statplot(Y,X,"l"):
mean(X)
183.75

```