Tugas Aplikasi Komputer

Romadhona Enggal Wilujeng

November 2024

1 Pendahuluan

Materi 1 Nama: Romadhona Enggal Wilujeng
Kelas: Matematika E 2023
NIM: 23030630074
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) = $\int_a^x f(t)\,dt - mathjax : \frac{x^2-1}{x-1} = x+1 - maxima :' integrate(x^3, x) = integrate(x^3, x)+C - http : //www.euler - math - toolbox.de - See : http : //www.google.de|Google - image : hati.png - - - -$

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

Judul

Sub-Judul

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

maxima: 'integrate(x$^3, x) = integrate(x^3, x) + C$

¡a href="http://www.euler-math-toolbox.de"¿http://www.euler-math-toolbox.de¡/a¿

¡a href="http://www.google.de"¿Google¡/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; pir$^2, 2pir$

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. * —

Jawaban

r=2.50; pir^2, $2pir$

19.6349540849 15.7079632679

2538/12

79.1666666667

35^5

52521875

—

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

1.25

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 $^{u}ntukeksponen(pangkat)$.

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

* $^{m}engikatlebihkuatdari*, sehinggapi*r^2merupakanrumus*luaslingkaran.$

* 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 (//).

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. * — * Jawaban

cos(45°), sin(pi), log(sqrt(E))
0.707106781187 0 0.5
7km - miles, 9inch - ” mm”
4.34959834566 228.6 mm
(sqrt(10)+6/2),
6.16227766017 8881.83783488
—
Satuan
1miles // 1 mil = 1609,344 m
1609.344
1miles // 1 mil = 1609,344 m
1609.344

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

Beberapa satuan yang sudah dikenal di dalam EMT adalah sebagai berikut. Semua unit diakhiri dengan tanda dolar (), $namunbolehtidakperluditulisdenganmengaktifkaneasyunits$.

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 mengatrtamilan 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 $12001.03^10, longE, longestpi$

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 sebelumnya. 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 $ax^2 + bx + c = 0 secara manual ... D = sqrt(b^2/(a^2 4) - c/a); ... -b/(2a) + D, ... -b/(2a) - D$

-0.138444501319 -3.61155549868

Menampilkan Daftar Variabe

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

km $1000 cm$ 0.01 mm $0.001 nm$ 1853.24496 gram $0.001 m$ 1 hquantum $6.62606957 e-34 atm$ 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 "intrandom" berikut ini!

intrandom(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(w^2)

0.35

Bilangan Kompleks

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

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 imaginer pada bilangan kompleks x. complex(x) : mengubah bilangan riil x menjadi bilangan kompleks. conj(x) : Konjugat untuk bilangan bilangan komplkes x. arg(x) : argumen (sudut dalam radian) bilangan kompleks x. real(x) : mengubah x menjadi bilangan riil.

Apabila bagian imaginer 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), $factor(x^2 + 5x + 6)$

2 2 b + 2 a b + a

(x + 2) (x + 3)

solve(ax^2 + $bx + c, x$)//$rumusabc$

2 2 (- sqrt(b - 4 a c)) - b sqrt(b - 4 a c) - b [x = ——————————, x = ——————————] 2 a 2 a

(a^2 − b^2)/$(a + b), ratsimp($

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)$cos(x)$)/$cos(x)^2$+$tan(x)sec(x)^2$)//$menyederhanakanfungsitrigonometri$

4 sin(x) + cos (x) ————-- 3 cos (x)

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

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 perhitunagn simbolik secara lebih bagus menggunakan LaTeX. Untuk melakukan hal ini, tambahkan tanda dolar ()$didepantandaamp;padasetiapperinta$

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)

$solve(ax^2+bx+c,x)//rumusabc$

$(a^2-b^2)/(a+b)$,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.

—

Jawaban

image: soal2.jpg

$2^3 2^5$

256

$5^-7/5^-5$

0.04

$(4^-32^4)^-2$

16

2 Aljabar

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:

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

Menjabarkan:

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

Baris Perintah

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

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

r:=2; h:=4; $pir^2h/3$

16.7551608191

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

pi2rh,

50.2654824574 100.530964915

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

x := 1;

x := cos(x) // nilai cosinus (x dalam radian)

0.540302305868

x := cos(x)

0.857553215846

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

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

1.41666666667 1.41421568627 1.41421356237

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

Untuk melipat semua multi-garis tekan Ctrl + L. Kemudian garis-garis berikutnya hanya akan terlihat, jika salah satunya memiliki 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, selama mereka masuk ke dalam satu baris atau multi-baris. Dalam program, pembatasan ini tidak berlaku, tentu saja. Untuk informasi lebih lanjut, lihat pengantar berikut.

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

1.5 1.41666666667 1.41421568627 1.41421356237 1.41421356237

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

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

1.41421356237

Struktur bersyarat juga berfungsi.

if E^pi pi^E; $then$"$Thoughtso$!", $endif$;

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 ke bagian komentar di atas perintah untuk menuju ke perintah.

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

sqrt(sin(10°)/cos(20°))
0.429875017772

Untuk melihat bantuan 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 di baris perintah.

exp(log(2.5))
2.5

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

Sintaks Dasar

Euler tahu 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 kuadrat dalam Euler. Tentu saja, $x^(1/2) jugadimungkinkan$.

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

Beberapa perintah dalam satu baris dipisahkan dengan "," atau ";". Titik koma menekan output dari 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. Memasuki
lateks: $e^2 \cdot \left(\frac{1}{3+4\log(0.6)} + \frac{1}{7}\right)$

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

E^2(1/(3 + 4*log*(0.6)) + 1/7)
8.77908249441

Untuk menghitung ekspresi rumit seperti
lateks: $\left(\frac{\frac{1}{7}+\frac{1}{8}+2}{\frac{1}{3}+\frac{1}{2}}\right)^2 \pi$
Anda harus memasukkannya dalam bentuk baris.

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

Letakkan tanda kurung dengan hati-hati di sekitar sub-ekspresi yang perlu dihitung terlebih dahulu. EMT membantu Anda dengan menyorot ekspresi

bahwa braket penutup selesai. Anda juga harus memasukkan nama "pi" untuk huruf Yunani pi.

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

1/3+1/7, fraction

0.47619047619 10/21

Perintah Euler dapat berupa ekspresi atau perintah primitif. Ekspresi dibuat dari operator dan fungsi. Jika perlu, itu harus mengandung tanda kurung untuk memaksa urutan eksekusi yang benar. Jika ragu, memasang braket adalah ide yang bagus. Perhatikan bahwa EMT menunjukkan 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**dayauntukpositifaataubilanganbulatb*(*a***bjugaberfungsi*)*n*!*operatorfaktorial*

dan masih banyak lagi.

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

sin,cos,tan,atan,asin,acos,rad,deg log,exp,log10,sqrt,logbase bin,logbin,logfac,mod,lantai,ceil,bulat,abs,tanda
conj,re,im,arg,conj,nyata,kompleks beta,betai,gamma,complexgamma,ellrf,ellf,ellrd,elle
bitand, bitor, bitxor, bitnot

Beberapa perintah memiliki alias, mis. Untuk log.

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

2 0.5

sin(30°)

0.5

Pastikan untuk menggunakan tanda kurung (kurung bulat), setiap kali ada keraguan tentang urutan eksekusi! Berikut ini tidak sama dengan (2^3)^4, *yangmerupakandefaultuntuk*2^34*diEM*

2^34, (2^3)^4, 2^(3^4)

2.41785163923e+24 4096 2.41785163923e+24

Bilangan Asli

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

longest 1/3

0.3333333333333333

Representasi ganda internal membutuhkan 8 byte.

printdual(1/3)

1.0101010101010101010101010101010101010101010101010101*2^-2

printhex(1/3)

5.5555555555554*16^-1

String

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

"A string can contain anything."

A string can contain anything.

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

"The area of the circle with radius " + 2 + " cm is " + pi4 + " cm^2."

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

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

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

Golden Ratio : 1.61803

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

none

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

"1234.5"()

1234.5

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

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

affe charlie bravo

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

w:=[none]; w—v—v

affe charlie bravo affe charlie bravo

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

String Unicode dapat digabungkan seperti string lainnya.

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

= 45°

I

Dalam komentar, entitas yang sama seperti , dll dapat digunakan. Ini mungkin alternatif cepat untuk Lateks. (Lebih detail 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

Nilai Boolean

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

2¡1, "apel"¡"banana"

0 1

"dan" adalah operator "amp;amp;" dan "atau" adalah operator "——", seperti dalam bahasa C. (Kata-kata "dan" dan "atau" hanya dapat digunakan dalam kondisi untuk "jika".)

2¡E E¡3

1

Operator Boolean mematuhi aturan bahasa matriks.

(1:10) 5, nonzeros(

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

Anda dapat menggunakan fungsi bukan nol() untuk mengekstrak elemen tertentu dari vektor. Dalam contoh, kami menggunakan isprima bersyarat(n).

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

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

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

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

Format Keluaran

Format output default EMT mencetak 12 digit. Untuk memastikan bahwa kami melihat default, kami mengatur ulang format.

defformat; pi

3.14159265359

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

longest pi

3.141592653589793

Berikut adalah representasi heksadesimal internal dari bilangan ganda.

printhex(pi)

$3.243F6A8885A30*16^0$

Format output dapat diubah secara permanen dengan perintah format.

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

0.33333 3.14159 0.84147

Standarnya adalah format (12).

format(12); 1/3
0.333333333333

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

shortestformat; random(3,8)
0.66 0.2 0.89 0.28 0.53 0.31 0.44 0.3 0.28 0.88 0.27 0.7 0.22 0.45 0.31 0.91 0.19 0.46 0.095 0.6 0.43 0.73 0.47 0.32

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

setscalarformat(5); pi
3.1416

Fungsi "format terpanjang" mengatur format skalar juga.

longestformat; pi
3.141592653589793

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

format terpendek format pendek format panjang, format terpanjang format(panjang,digit) format baik(panjang) fracformat (panjang) mengubah bentuk

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

Tetapi format output EMT dapat diatur dengan cara yang fleksibel.

longestformat; pi,
3.141592653589793
format(10,5); pi
3.14159

The default is defformat().

defformat; // default

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

longest $pi^2/2$
4.934802200544679

Ada juga operator pendek 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 dengan tepat. Kesalahan bertambah sedikit, seperti yang Anda lihat dalam perhitungan berikut.

longest 0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1-1
-1.110223024625157e-16

Tetapi dengan "format panjang" default Anda tidak akan melihat ini. Untuk kenyamanan, output dari angka yang sangat kecil adalah 0.

0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1+0.1-1
0

Ekspresi

String atau nama dapat digunakan untuk menyimpan ekspresi matematika, yang dapat dievaluasi oleh EMT. Untuk ini, gunakan tanda kurung setelah

ekspresi. Jika Anda bermaksud menggunakan string sebagai ekspresi, gunakan konvensi untuk menamainya "fx" atau "fxy" dll. Ekspresi lebih diutamakan daripada fungsi.

Variabel global dapat digunakan dalam evaluasi.

r:=2; fx:="pir^2"; $longestfx()$

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

at:=4; function f(expr,x,at) := expr(x); ... f("atx^2", 3, 5)//$computes4 \times 3^2not53^2$

36

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

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

45

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

at:=4; function f(expr,x) := expr(x); ... f("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 menghapus variabel ini untuk menghindari kebingungan antara ekspresi simbolik dan fungsi.

f = 5x;

function f(x) := 6x;

f(2)

12

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

fx = diff(x^x, x);fx

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

"@(a,b) a^2 + b^2",

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

Ini memungkinkan untuk memanipulasi ekspresi dalam variabel lain untuk fungsi EMT yang membutuhkan ekspresi dalam "x".

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

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

fx = $x^3 - ax; ... a = 1.2; fx(0.5)$

-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 simbolis. 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 mencatat bahwa ada perbedaan sintaks antara sintaks asli Maxima dan sintaks default ekspresi simbolik di EMT.

Matematika simbolik terintegrasi dengan mulus ke dalam Euler dengan 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 angka yang sangat besar.

44!

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

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

$44!/(34!10!)//nilai C(44,10)$

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

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

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

Anda akan belajar bahwa yang berikut ini juga berfungsi.

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

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

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

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

Dengan begitu Anda dapat menggunakan solusi persamaan dalam persamaan lain.

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

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

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

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

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

"v := 5; v^2"

25

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

$expand((1 + x)^4)$,factor(diff(

Sekali lagi,

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

fx = (x+1)/(x^4 + 1);fx

Ekspresi simbolik dapat digunakan dalam ekspresi simbolik lainnya.

$factor(diff(fx, x))$

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

factor(20!)

2432902008176640000

::: factor(10!)

8 4 2 2 3 5 7

:: factor(20!)

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

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

::: av:gav^2;

2 g

fx = $x^3 exp(x)$,fx

3 x x E

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

(fx with x=5),

5 125 E

18551.64488782208

fx(5)

18551.6448878

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

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

(fx with x=10)-(fx with x=5), float(

10 5 1000 E - 125 E

2.20079141499189e+7

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

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

tex(fx)

$x^3\ e^x$

Ekspresi simbolik dapat dievaluasi seperti ekspresi numerik.

fx(0.5)

0.206090158838

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

$fxwithx = 1/2$

Penugasan juga bisa bersifat simbolis.

$fxwithx = 1 + t$

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

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

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

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

1.56155281281

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

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

[-3.23607, 1.23607]

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

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

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

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

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

$diff((x^3 - 1)/(x + 1), x)//turunanbentukpecahan$

$diff((x^3 - 1)/(x + 1), x)|ratsimp//menyederhanakanpecahan$

$factor($

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

f(0:0.1:1)

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

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

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

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

0.786151377757

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

Anda masih dapat memanggil fungsi bawaan sebagai "_...", $jikaituadalahfungsidiintiEuler.$

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

sin(45)

0.707106781187

Lebih baik kita menghapus redefinisi dosa ini.

forget sin; sin(pi/4)

0.707106781187

Parameter Default

Fungsi numerik dapat memiliki parameter default.

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

Menghilangkan parameter ini menggunakan nilai default.

f(4)

16

Menyetelnya akan menimpa nilai default.

f(4,5) ... 80

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

f(4,a=1)

16

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

function f(x) := ax^2

a=6; f(2)

24

Tetapi parameter yang ditetapkan menimpa nilai global.

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

f(2,a:=5)

20

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

function g(x) = $x^3 - xexp(-x)$;g(x)

Fungsi simbolik dapat digunakan dalam ekspresi simbolik.

$diff(g(x), x)$,

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

g(5+g(1))

178.635099908

Mereka dapat digunakan untuk mendefinisikan fungsi atau ekspresi simbolis lainnya.

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

solve(g(x),0.5)

0.703467422498

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

solve(g,0.5)

0.703467422498

function P(x,n) = $(2x-1)^n$;P(x,n)

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

$P(x, 4)$,expand(

P(3,4)

625

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

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

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

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

function f(x) = $x^3 - x$;f(x)

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

$integrate(f(x), x)$

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

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

yang tidak dapat dinilai secara simbolis.

Jika kita mendefinisikan kembali fungsi dengan kata kunci "peta" dapat digunakan untuk vektor x. Secara internal, fungsi 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)

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

Fungsi dapat memiliki nilai default untuk parameter.

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

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

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

2 6.7

Selain itu, dimungkinkan untuk menggunakan parameter yang ditetapkan.

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

2

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

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

Fungsi simbolik seperti itu dapat digunakan untuk variabel simbolik.

Tetapi fungsi tersebut juga dapat digunakan untuk vektor numerik.

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

17

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

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

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

$realpart((x + Iy)^4)$,lapl(

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

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

Untuk meringkas

* amp;= mendefinisikan fungsi simbolis,
* := mendefinisikan fungsi numerik,
* amp;amp;= mendefinisikan fungsi simbolis murni.

Memecahkan Ekspresi

Ekspresi dapat diselesaikan secara numerik dan simbolis.

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

solve("x^2 − 2", 1)

1.41421356237

Ini juga berfungsi untuk ekspresi simbolis. Ambil fungsi berikut.

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

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

$solve(ax^2 + bx + c = 0, x)$
$solve([ax + by = c, dx + ey = f], [x, y])$
px = $4x^8 + x^7 - x^4 - x$;px

Sekarang kita mencari titik, di mana 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 simbolis dalam bentuk simbolis mengembalikan daftar solusi. Kami menggunakan pemecah simbolik solve() yang disediakan oleh Maxima.

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

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

longest sol()

-0.6180339887498949 1.618033988749895

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

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

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

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

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

lateks: $a^x - x^a = 0.1$

dengan a=3.

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

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

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

2.54116291558

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

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

2.54116291558

Menyelesaikan Pertidaksamaan

Untuk menyelesaikan pertidaksamaan, EMT tidak akan dapat melakukannya, melainkan dengan bantuan Maxima, artinya secara eksak (simbolik). Perintah Maxima yang digunakan adalah $fourier_elim()$, $yangharusdipanggildenganperintah"load(fourier_elim)"ter$

load(fourier_elim)

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

$fourier\_elim([x^2 - 1\ 0], [x])//x^2 - 1\ 0$

$fourier\_elim([x^2 - 1 < 0], [x])//x^2 - 1 < 0$

$fourier\_elim([x^2 - 10], [x])//x^-1 < 0$
$fourier\_elim([x6], [x])$
$fourier\_elim([x < 1, x\ 1], [x])//tidakmemilikipenyelesaian$
$fourier\_elim([minf < x, x < inf], [x])//solusinyaR$
$fourier\_elim([x^3 - 1\ 0], [x])$
$fourier\_elim([cos(x) < 1/2], [x])//???gagal$
$fourier\_elim([y - x < 5, x - y < 7, 10 < y], [x, y])//sistempertidaksamaan$
$fourier\_elim([y - x < 5, x - y < 7, 10 < y], [y, x])$
$fourier\_elim((x + y < 5)and(x - y\ 8), [x, y])$
$fourier\_elim(((x + y < 5)andx < 1)or(x - y\ 8), [x, y])$
fourier_elim([max(x,y) 6, x 8, abs(y-1) 12],[x,y])
[6 lt; x, x lt; 8, y lt; - 11] or [8 lt; x, y lt; - 11] or [x lt; 8, 13 lt; y] or [x = y, 13 lt; y] or [8 lt; x, x lt; y, 13 lt; y] or [y lt; x, 13 lt; y]
$fourier\_elim([(x + 6)/(x - 9) <= 6], [x])$

Bahasa Matriks

Dokumentasi inti EMT berisi diskusi terperinci tentang bahasa matriks Euler.

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

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

Produk matriks dilambangkan dengan titik.

b=[3;4]
3 4
b' // transpose b
[3, 4]
inv(A) //inverse A
-2 1 1.5 -0.5
A.b //perkalian matriks
11 25
A.inv(A)
1 0 0 1

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

A.A
7 10 15 22
$A^2//perpangkatanelemen2A$
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)

0.333333 0.666667 0.75 1

A

b // hasilkali invers A dan b, $A^(-1)b$

-2 2.5

inv(A).b

-2 2.5

A

A //$A^(-1)A$

1 0 0 1

inv(A).A

1 0 0 1

AA //perkalin elemen-elemen matriks seletak

1 4 9 16

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

b^2//$perpangkatanelemen - elemenmatriks/vektor$

9 16

Jika salah satu operan adalah vektor atau skalar, itu 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 itu adalah vektor baris, itu diterapkan ke semua kolom A.

A[2,3]

2 6 6 12

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

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

1 2 1 2

Adup([1,2],2)

1 4 3 8

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

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

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

Sekali lagi, ingat bahwa ini bukan produk matriks!

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

55

sum((1:5)(1:5)) // sama hasilnya

55

Bahkan operator seperti 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.

Kami mendapatkan vektor 0 dan 1, di mana 1 berarti benar.

t=(1:10)^2; $t == 25//mengujielemen2tyangsamadengan25(hanyaada1)$

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

Dari vektor seperti itu, "bukan nol" memilih elemen bukan nol.

Dalam hal ini, kami mendapatkan indeks semua elemen lebih besar dari 50.

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

[8, 9, 10]

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

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

[64, 81, 100]

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

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

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

EMT tidak sepenuhnya efektif untuk perhitungan bilangan bulat. Ini menggunakan titik mengambang presisi ganda secara internal. Namun, seringkali sangat berguna.

Kita dapat memeriksa keutamaan. Mari kita cari tahu, berapa banyak kuadrat ditambah 1 adalah bilangan prima.

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

112

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

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

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

Ini mengembalikan indeks elemen, yang bukan nol.

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

1 4 2 1 2 2 3 2

Indeks ini dapat digunakan untuk mengatur elemen ke beberapa nilai.

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

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

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

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

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

Dan dimungkinkan untuk mendapatkan elemen dalam vektor.

mget(A,k)

[0.267829, 0.13673, 0.390567, 0.006085]

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

ex=extrema(A)

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

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

ex[,3]'

[0.765761, 0.952814, 0.548138]

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

max(A)'

[0.765761, 0.952814, 0.548138]

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

j=(1:rows(A))'—ex[,4], mget(-A,j)

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

Fungsi Matriks Lainnya (Membangun Matriks)

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

v=1:3; v_v

1 2 3 1 2 3

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

A=random(3,4); A—v'

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

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

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

A—1

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

Dimungkinkan untuk membuat matriks vektor baris dan kolom.

[v;v]

1 2 3 1 2 3

[v',v']

1 1 2 2 3 3

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

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

1 1 2 4 3 9

Untuk mendapatkan ukuran A, kita dapat menggunakan fungsi berikut.

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

2 4 [2, 4] 4

Untuk vektor, ada panjang().

length(2:10)

9

Ada banyak fungsi lain, yang menghasilkan matriks.

ones(2,2)

1 1 1 1

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

ones(5)6

[6, 6, 6, 6, 6]

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

random(2,2)

0.66566 0.831835 0.977 0.544258

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

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

1 2 3 4 5 6 7 8 9

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

function rep(v,n) := redim(dup(v,n),1,ncols(v))

Let us test.

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]

Sebuah 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 harus menemukan elemennya terlebih dahulu. Fungsi indexof(v,x) dapat digunakan untuk mencari elemen x dalam vektor terurut v.

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

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

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

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

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

Ada beberapa fungsi khusus untuk mengatur diagonal atau untuk menghasilkan matriks diagonal.

Kita mulai dengan matriks identitas.

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

1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1

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

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

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

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

Berikut adalah fungsi, yang mengembalikan matriks tri-diagonal.

function tridiag (n,a,b,c) := setdiag(setdiag(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 3 0 0 0 1 2

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

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

1 2 3 4 5 6 7 8 9

Sekarang kita dapat mengekstrak diagonal.

d=getdiag(A,0)

[1, 5, 9]

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

fraction A/d'

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

Vektorisasi

Hampir semua fungsi di Euler juga berfungsi untuk input matriks dan vektor, kapan pun ini masuk akal.

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

sqrt(1:3)

[1, 1.41421, 1.73205]

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

x=1:0.01:5; y=log(x)/x^2; *//terlalupanjanguntukditampikan*

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

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

lateks: s = $t^3 - t$

t=-1:0.1:1; s=$t^3 - t$

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

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

Misalnya, vektor kolom dikalikan vektor baris menjadi matriks, jika operator diterapkan. Berikut ini, v' adalah vektor yang ditransposisikan (vektor kolom).

shortest (1:5)(1:5)'

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

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

(1:5).(1:5)'

55

Secara default, vektor baris dicetak dalam format yang ringkas.

[1,2,3,4]

[1, 2, 3, 4]

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

v:=[1,2]; v.v',

5 25

Untuk mentranspos matriks kita menggunakan apostrof.

v=1:4; v'

1 2 3 4

Jadi kita dapat menghitung matriks A kali vektor b.

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

30 70

Perhatikan bahwa v masih merupakan vektor baris. Jadi v'.v berbeda dari 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 bekerja seperti bilangan real.

v.v'

30

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

norm(v)^2

30

Operator dan fungsi mematuhi bahasa matriks Euler.

Berikut ringkasan aturannya.

* Fungsi yang diterapkan ke vektor atau matriks diterapkan ke setiap * elemen.

* Operator yang beroperasi pada dua matriks dengan ukuran yang sama * diterapkan berpasangan ke elemen matriks.

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

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

Berikut ini adalah kasus sederhana dengan operator ·

[1,2,3]^2

[1, 4, 9]

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

v:=[1,2,3]; vv'

1 2 3 2 4 6 3 6 9

Perhatikan bahwa produk skalar menggunakan produk matriks, bukan *!

v.v'

14

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

sum,prod menghitung jumlah dan produk dari baris cumsum,cumprod melakukan hal yang sama secara kumulatif menghitung nilai ekstrem dari setiap baris extrema mengembalikan vektor dengan informasi ekstrim diag(A,i) mengembalikan diagonal ke-i setdiag(A,i,v) mengatur diagonal ke-i id(n) matriks identitas det(A) penentu 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 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 ada operator "—" dan "¦".

[1,2,3]—[4,5], [1,2,3]¦1

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

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

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

6

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

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

6 [7, 8, 9]

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

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

[2, 4] 2 5 8

Bentuk singkat untuk : adalah menghilangkan indeks sepenuhnya.

A[,2:3]

2 3 5 6 8 9

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

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

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

0 45 90 135 180 225 270 315 360

Sekarang kita menambahkan kolom ke matriks.

M = deg(w)—w—cos(w)—sin(w)

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

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

Dalam contoh berikut, kita menghitung t[j]$^i untuk i dari 1 hingga. Kami mendapatkan matriks, dimana setia$

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

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

Integrasi numerik terintegrasi() hanya berfungsi untuk batas interval skalar. Jadi kita perlu membuat vektor.

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

Kata kunci "peta" membuat vektor fungsi. Fungsinya sekarang akan bekerja untuk vektor bilangan.

f([1:5])

[0, 2.05045, 13.7251, 113.336, 1241.03]

Sub-Matriks dan Matriks-Elemen

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 satu baris matriks yang lengkap.

A[2]

[4, 5, 6]

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

v=1:3; v[2]

2

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

A[2,]

[4, 5, 6]

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

Di sini kita menginginkan baris pertama dan kedua dari A.

A[[1,2]]

1 2 3 4 5 6

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

A[[3,2,1]]

7 8 9 4 5 6 1 2 3

Trik indeks bekerja dengan kolom juga.

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

Atau, biarkan indeks pertama kosong.

A[,2:3]

2 3 5 6 8 9

Kita juga bisa mendapatkan baris terakhir dari A.

A[-1]

[7, 8, 9]

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

A[1,1]=4

4 2 3 4 5 6 7 8 9

Kami juga dapat menetapkan nilai ke baris A.

A[1]=[-1,-1,-1]

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

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

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

5 6 -1 7 8 6 7 8 9

Selain itu, beberapa 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, tergantung pada pengaturan sistem. Standarnya adalah pesan kesalahan. Ingat, bagaimanapun, bahwa indeks negatif dapat digunakan untuk mengakses elemen matriks yang dihitung dari akhir.

A[4]

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

Menyortir dan Mengacak

Fungsi sort() mengurutkan vektor baris.

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

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

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

Mari kita mengocok vektor.

v=shuffle(1:10)

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

Indeks berisi urutan yang tepat dari v.

vs,ind=sort(v); v[ind]

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

Ini bekerja untuk vektor string juga.

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

a d e a aa e

ss,ind=sort(s); ss

a a aa d e e

Seperti yang Anda lihat, posisi entri ganda agak acak.

ind

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

Fungsi unik mengembalikan daftar elemen unik vektor yang diurutkan.

intrandom(1,10,10), unique(

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

Ini bekerja untuk vektor string juga.

unique(s)

a aa d e

Aljabar linier

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

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

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

-4 4.5

Untuk contoh lain, kami membuat matriks 200x200 dan jumlah 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 benar.

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

8.790745908981989e-13

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

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

1 2 3 4 5 6 7 8 9

Determinan matriks ini adalah 0.

det(A)

0

Matriks Simbolik

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

A = [a,1,1;1,a,1;1,1,a]; A

$det(A)$,factor(

$invert(A) with a = 0$

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

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

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

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

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

Untuk mengekstrak vektor eigen tertentu perlu pengindeksan yang cermat.

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

[1, - 1]

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

A(a=4,b=5)

1 4 5 2

Dalam ekspresi simbolik, gunakan dengan.

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

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

$A[1]$

Ekspresi simbolis dapat berisi tugas. Dan itu mengubah matriks A.

A[1,1]:=t+1; A

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

B := [1,2;3,4]; B,invert(B)

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

$invert(B)()$

-2 1 1.5 -0.5

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

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

longest B.xinv(B)

1 0 0 1

Misalnya. nilai eigen dari A dapat dihitung secara numerik.

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

[16.1168, -1.11684, 0]

Atau secara simbolis. Lihat tutorial tentang Maxima untuk detailnya.

$eigenvalues(@A)$

Nilai Numerik dalam Ekspresi simbolis

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

A := [1,pi;4,5]

1 3.14159 4 5

Masih ada perbedaan antara bentuk numerik dan simbolik. Saat mentransfer matriks ke bentuk simbolis, pendekatan fraksional untuk real akan digunakan.

A

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

mxmset(A); A

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

$bfloat(sqrt(2))$,float(sqrt(2))

Ketepatan angka floating point besar dapat diubah.

fpprec:=100; bfloat(pi)

3.14159265358979323846264338327950288419716939937510582097494 45923078164062862089986280348253

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

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

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

Demo - Suku Bunga

Di bawah ini, kami menggunakan Euler Math Toolbox (EMT) untuk perhitungan 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 5000 (katakanlah dalam dolar).

K=5000

5000

Sekarang kita asumsikan tingkat bunga 3tambahkan satu tarif sederhana dan hitung hasilnya.

K1.03

5150

Euler akan memahami sintaks berikut juga.

K+K3

5150

Tetapi lebih mudah menggunakan faktornya

q=1+3

1.03 5150

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

Kq^1 0

6719.58189672

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

format(12,2); Kq^1 0

6719.58

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

"Starting from " + K + "*youget*" + $round(Kq^1 0, 2)$ + "."

Starting from 5000*youget*6719.58.

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

Kq^(0 : 10)

Real 1 x 11 matrix

5000.00 5150.00 5304.50 5463.64 ...

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

short 0:10

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

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

short q^(0 : 10)

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

adalah vektor faktor q^0 $sampai q^1 0. Inidikalikandengan K, dankamimendapatkanvektornilai.$

VK=Kq^(0 : 10);

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

function oneyear (K) := round(Kq,2)

Mari kita bandingkan dua hasil, 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 mengulang selama bertahun-tahun. Euler memberikan banyak solusi untuk ini.

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

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

Real 1 x 11 matrix

5000.00 5150.00 5304.50 5463.64 ...

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

VKr'

5000.00 5150.00 5304.50 5463.64 5627.55 5796.38 5970.27 6149.38 6333.86 6523.88 6719.60

Untuk mendapatkan elemen tertentu dari vektor, 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. Ingat bahwa 1:3 menghasilkan vektor [1,2,3].

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

VKr[-1], VK[-1]

6719.60 6719.58

Perbedaannya sangat kecil.

Memecahkan Persamaan

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

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("onepay",5000,10)

Real 1 x 11 matrix

5000.00 4950.00 4898.50 4845.45 ...

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

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

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

Real 1 x 51 matrix

5000.00 4950.00 4898.50 4845.45 ...

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

min(nonzeros(VKR¡0))

48.00

Alasan untuk ini adalah bahwa bukan nol(VKRlt;0) mengembalikan vektor indeks i, di mana VKR[i]lt;0, dan min menghitung indeks minimal.

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

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

x,n=iterate("onepay",5000,till="x¡0"); x, n,

-19.83 47.00

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

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

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

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

Selain itu, kami hanya tertarik pada nilai terakhir. Jadi 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

Rutin memecahkan memecahkan ekspresi=0 untuk variabel x. Jawabannya adalah 3,15Fungsi 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 3tahun.

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

-336.08

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

Solusi Simbolik untuk Masalah Suku Bunga

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

function op(K) = Kq+R; $op(K)$

Kita sekarang dapat mengulangi ini.

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

Kami melihat sebuah pola. Setelah n periode yang kita miliki

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

Rumusnya adalah rumus untuk jumlah geometri, yang diketahui Maxima.

$\text{sum}(\text{q}^k, k, 0, n-1)$;

Ini agak rumit. Jumlahnya dievaluasi dengan bendera "simpsum" untuk menguranginya menjadi hasil bagi.

Mari kita membuat fungsi untuk ini.

function fs(K,R,P,n) = $(1+\text{P}/100)^n K+((1+P/100)^n-1)/(P/100)R$;fs(K,R,P,n)

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

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

-19.82504734650985 -19.82504734652684

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

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

20.51

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

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

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

equ = fs(K,R,P,n)=Kn; equ

Biasanya rumus ini diberikan dalam bentuk

lateks: i = $\text{P}\frac{}{100}$

equ = (equ with P=100i); equ

Kita dapat memecahkan tingkat R secara simbolis.

$solve(equ, R)$

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

Tentu saja, kami memiliki batasan berikut.

$limit(R(5000, 0, x, 10), x, 0)$

Jelas, tanpa bunga kita harus membayar kembali 10 tarif 500.

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

fn = solve(equ,n) — ratsimp; fn

3 Plot2D

Romadhona Enggal Wilujeng$_p lot2DNama$: $RomadhonaEnggalWilujeng$

NIM : 23030630074

Kelas : Matematika E

Menggambar Grafik 2D dengan EMT

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

Plot Dasar

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

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

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

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.

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

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

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

Aspek Plot

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

Tetapi Anda juga dapat 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
plot2d(["sin(x)","cos(x)"],0,2pi):
aspect();
reset;

Fungsi reset() mengembalikan default plot termasuk rasio aspek.

Plot 2D di Euler

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

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

Euler dapat memplot plot 2D dari

* ekspresi
* fungsi, variabel, atau kurva parameter,
* 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" (mis. "4*x^2")$ataunamafungsi(mis."f")menghasilkangrafikfungsi.$

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

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

plot2d("x^2") :
aspect(1.5); plot2d("$x^3 - x$") :
a:=5.6; plot2d("exp(-ax^2)/a"); $insimg(30); //menampilkangambarhasilplotsetinggi25baris$

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

Rentang plot diatur dengan parameter yang ditetapkan berikut:

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

plot2d("$x^3 - x$", $-1, 2$) :
plot2d("sin(x)",-2pi,2pi): // plot sin(x) pada interval [-2pi, 2pi]
plot2d("cos(x)","sin(3x)",xmin=0,xmax=2pi):

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

Dalam opsi, plot dapat diatur untuk muncul

* di jendela terpisah yang dapat diubah ukurannya,

* di jendela buku catatan.

Lebih banyak gaya dapat dicapai dengan perintah plot tertentu.

Bagaimanapun, tekan tombol tabulator untuk melihat plot, jika disembunyikan.

Untuk membagi jendela menjadi beberapa plot, gunakan perintah figure(). Dalam contoh, kami memplot x$^1 hinggax^4 menjadi4 bagian jendela. figure(0) mengaturulangjendeladefault.$

reset;

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

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

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

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

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

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

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

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

Gambar yang dihasilkan dengan memasukkan plot ke dalam jendela teks disimpan di direktori yang sama dengan buku catatan, secara default di subdirektori bernama "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 mengekspor grafik saat ini dengan fungsi di menu File.

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

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

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

Perhatikan bahwa x$^x tidakdidefinisikanuntukxlt; = 0. Fungsiplot2dmenangkapkesalahanini, danmulaime$

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

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

plot2d("x$^3 - x$", *square*) :

plot2d("integrate("sin(x)exp(-x^2)", $0, x)'', 0, 2)$: *//plotintegral*

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

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

Ekspresi simbolik juga dapat digunakan, karena disimpan sebagai ekspresi string sederhana.

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

a:=5.6; expr = exp(-ax^2)/a; //$defineexpression$

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

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

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

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

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

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

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

Fungsi dalam satu Parameter

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

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

function f(x,a) := x^2/a + ax^2 − x; //$defineafunction$

a=0.3; plot2d("f",0,1;a): // plot with a=0.3

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

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

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

function f(x) := x^3 − x; ... $plot2d("f", r = 1)$:

Berikut adalah ringkasan dari fungsi yang diterima

* ekspresi atau ekspresi simbolik dalam x

* fungsi atau fungsi simbolis dengan nama sebagai "f"

* fungsi simbolis hanya dengan nama f

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

function f(x) = diff(x^x, x)

x x (log(x) + 1)

plot2d(f,0,2):

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

expr = sin(x)exp(-x)

- x E sin(x)

plot2d(expr,0,3pi):

function f(x) = x^x;

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

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

Untuk gaya garis ada berbagai pilihan.

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

* warna: Lihat di bawah untuk warna.

* ketebalan: Default 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 terang, kuning

* rgb(merah, hijau, biru): parameter adalah real dalam [0,1].

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

Berikut adalah tampilan warna EMT yang telah ditentukan sebelumnya.

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

But you can use any color.

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

Menggambar Beberapa Kurva pada bidang koordinat yang sama

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

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

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

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

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

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

plot2d(["cos(x)","x"],r=1.1,cx=0.5,cy=0.5, ... color=[black,blue],style=["-","."], ... grid=1);

x0=solve("cos(x)-x",1); ... plot2d(x0,x0, points, add,title="Intersection Demo"); ... label("cos(x) = x",x0,x0,pos="cl",offset=20):

Dalam demo berikut, kami memplot fungsi sinc(x)=sin(x)/x dan ekspansi Taylor ke-8 dan ke-16. Kami menghitung ekspansi ini menggunakan Maxima melalui ekspresi simbolis.

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

Kami menambahkan kotak label yang menjelaskan fungsi.

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

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]):

Dalam contoh berikut, kami menghasilkan Bernstein-Polinomial.

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

plot2d("(1-x)^{1}0", 0, 1); *//plotfirstfunction*

for i=1 to 10; plot2d("bin(10,i)x^i(1 − x)(10 − i)", *add*); *end*;

insimg;

Metode kedua menggunakan pasangan matriks nilai-x dan matriks nilai-y yang berukuran sama.

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

x=linspace(0,1,500);

n=10; k=(0:n)'; // n is row vector, k is column vector

y=bin(n,k)$x^k(1-x)^(n-k)$; //yisamatrixthen

plot2d(x,y):

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

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

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

plot2d(["sin(x)","cos(x)"],0,2pi,color=4:5):

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

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

v = makelist(binomial(10,i)$x^i(1-x)^(10-i), i, 0, 10)$//$makelist$

10 9 8 2 7 3 [(1 - x) , 10 (1 - x) x, 45 (1 - x) x , 120 (1 - x) x , 6 4 5 5 4 6 3 7 210 (1 - x) x , 252 (1 - x) x , 210 (1 - x) x , 120 (1 - x) x , 2 8 9 10 45 (1 - x) x , 10 (1 - x) x , x]

mxm2str(v) // get a vector of strings from the symbolic vector

(1-x)^{1}010$*(1-x)^9*x$45$*(1-x)^8*x^2$120$*(1-x)^7*x^3$210$*(1-x)^6*x^4$252$*(1-x)^5*x^5$210$*(1-x)^4*x^6$120$*(1-x)^3*x^7$45$*(1-x)^2*x^8$10$*(1-x)*x^9x^1$0

plot2d(mxm2str(v),0,1): // plot functions

Alternatif lain adalah dengan menggunakan bahasa matriks Euler.

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

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

n=(1:10)'; plot2d("x^n", 0, 1, $color = 1 : 10$) :

Ekspresi dan fungsi satu baris dapat melihat variabel global.

Jika Anda tidak dapat menggunakan variabel global, Anda perlu menggunakan fungsi dengan parameter tambahan, dan meneruskan parameter ini sebagai parameter titik koma.

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

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

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

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

plot2d("f",1,-10,10); ... for a=2:10; plot2d("f",a, add); end:

Kami dapat mencapai hasil yang sama dengan cara berikut menggunakan bahasa matriks EMT. Setiap baris matriks f(x,a) adalah satu fungsi. Selain itu, kita dapat mengatur warna untuk setiap baris matriks. Klik dua kali pada fungsi getspectral() untuk penjelasannya.

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

Label Teks

Dekorasi sederhana bisa

* judul dengan judul="..."
* x- dan y-label dengan xl="...", yl="..."
* label teks lain dengan label("...",x,y)

Perintah label akan memplot ke dalam plot saat ini pada koordinat plot (x,y). Itu bisa mengambil argumen posisi.

plot2d("$x^3 - x$", $-1, 2, title = "y = x^3 - x", yl = "y", xl = "x")$:
expr := "log(x)/x"; ... plot2d(expr,0.5,5,title="y="+expr,xl="x",yl="y"); ... label("(1,0)",1,0); label("Max",E,expr(E),pos="lc"):

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

function f(x) = $x^2 exp(-x^2)$; ... $plot2d(f(x), a = -3, b = 3, c = -1, d = 1)$; ...$plot2d(diff(f(x), x), add, color = blue, style = "--")$; ...$labelbox(["function", "derivative"], styles = [" - "," - -"]$, ... $colors = [black, blue], w = 0.4)$:

Kotak ditambatkan di kanan atas secara default, tetapi gt; kiri menambatkannya di kiri atas. Anda dapat memindahkannya ke tempat yang Anda suka. Posisi jangkar adalah sudut kanan atas kotak, dan angkanya adalah pecahan dari ukuran jendela grafik. Lebarnya otomatis.

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

Dalam 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):

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

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

Fitur serupa adalah fungsi textbox().

Lebar secara default adalah lebar maksimal dari 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)=$e^{-x} sin(2$
$pix)"), w = 0.85)$:

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

plot2d("x$^3 - x$", $title = u$"$xrarr; xsup3; -x$") :

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

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

LaTeX

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

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

Dalam plot berikut, kami menggunakan LaTeX untuk label x dan y, label, kotak label, dan judul plot.

plot2d("exp(-x)sin(x)/x",a=0,b=2pi,c=0,d=1,grid=6,color=blue, ... title=latex("
textFunction
Phi"), ... xl=latex("
phi"),yl=latex("
Phi(
phi)")); ... textbox(... latex("
Phi(
phi) = e^{-phi}
$fracsin(phi)phi$"), $x = 0.8, y = 0.5$); ... $label(latex($"
Phi", $color = blue$), 1, 0.4) :

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

Cara termudah adalah dengan membuat plot kosong dengan bingkai menggunakan grid=4, lalu menambahkan grid dengan ygrid() dan xgrid(). Dalam 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):

Tentu saja, fungsi juga dapat digunakan.

function map f(x) ...

if x¿0 then return x$^4 elsereturnx^2 endifendfunction$ < /pre > $Parameter$"$peta$"$membantumenggunakanf$

plot, itu tidak perlu. Tetapi untuk mendemonstrasikan vektorisasi itu

berguna, kami menambahkan beberapa poin kunci 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 akan dapat menggunakan

LaTeX jika Anda telah menginstal LaTeX dengan benar.

plot2d("f",-1,1,xl="x",yl="f(x)",grid=6); ... plot2d([-1,0,1],f([-1,0,1]), points, add); ... label(latex("x^3"), 0.72, $f(0.72)$); ...$label(latex("x^2"), -0.52, f(-0.52), pos =$ "ll"); ... $textbox(...\ latex("f(x) =$
$begincasesx^3x\ 0$

x^2x
$le0$
$endcases$"), ... $x = 0.7, y = 0.2)$:

Interaksi pengguna

Saat memplot fungsi atau ekspresi, parameter gt;user memungkinkan pengguna untuk memperbesar dan menggeser plot dengan tombol kursor atau mouse. Pengguna dapat

* 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 asli.

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

plot2d("x^3 $- ax$", $a = 1$, $user, title =$ "$Pressanykey$!") :

plot2d("exp(x)sin(x)",user=true, ... title="+/- or cursor keys (return to exit)"):

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

Fungsi bawaan mousedrag() menunggu event mouse atau keyboard. Ini melaporkan mouse ke bawah, mouse dipindahkan 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. Sebagai contoh, kita interpolasi dalam 5 titik dengan polinomial. Fungsi harus diplot ke area plot tetap.

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

d=interp(xp,yp); plot2d("interpval(xp,d,x)";d,xp,r=2); plot2d(xp,yp,¿points,¿add); if select¿0 then plot2d(xp[select],yp[select],color=red,¿points,¿add); endif; title("Drag one point, or press space or return!"); endfunction ¡/pre¿ Perhatikan parameter titik koma di plot2d (d dan xp), yang diteruskan ke evaluasi fungsi interp(). Tanpa ini, kita harus menulis fungsi plotinterp() terlebih dahulu, mengakses nilai secara global.

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

t=-1:0.5:1; dragpoints("plotf",t,random(size(t))-0.5):

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

Pertama kita membutuhkan fungsi plot.

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

Kemudian kita membutuhkan nama untuk parameter, nilai awal dan matriks rentang nx2, opsional baris judul.

Ada slider interaktif, yang dapat mengatur nilai oleh pengguna. Fungsi dragvalues() menyediakan ini.

dragvalues("plotf",["a","b"],[-1,2],[[-2,2];[1,10]], ... heading="Drag these values:",hcolor=black):

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

function plotf(n) ...

plot2d("cos(x)",0,2pi,¿square,grid=6); plot2d("taylor(cos(x),x,0,@n)",color=blue,¿add); textbox("Taylor polynomial of degree "+n,0.1,0.02,style="t",¿left); endfunction ¡/pre¿ Sekarang kami mengizinkan derajat n bervariasi dari 0 hingga 20 dalam 20 pemberhentian. 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):

Berikut ini adalah demonstrasi sederhana dari fungsi tersebut. Pengguna dapat menggambar di atas jendela plot, meninggalkan jejak poin.

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 ¡/pre¿ 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 dimodifikasi. Selain itu, label dan judul dapat ditambahkan secara manual. Untuk mengatur ulang ke gaya default, gunakan reset().

aspect();

figure(3,4); ... figure(1); plot2d("x^3−x", $grid = 0$); .../$/nogrid, frameoraxis$

figure(2); plot2d("x$^3 - x$", $grid = 1$); .../$/x - y - axis$

figure(3); plot2d("x$^3 - x$", $grid = 2$); .../$/defaultticks$

figure(4); plot2d("x$^3 - x$", $grid = 3$); .../$/x - y - axiswithlabelsinside$

figure(5); plot2d("x$^3 - x$", $grid = 4$); .../$/noticks, onlylabels$

figure(6); plot2d("x$^3 - x$", $grid = 5$); .../$/default, butnomargin$

figure(7); plot2d("x$^3 - x$", $grid = 6$); .../$/axesonly$

figure(8); plot2d("x$^3 - x$", $grid = 7$); .../$/axesonly, ticksataxis$

figure(9); plot2d("x$^3 - x$", $grid = 8$); .../$/axesonly, finerticksataxis$

figure(10); plot2d("x$^3 - x$", $grid = 9$); .../$/default, smallticksinside$

figure(11); plot2d("x$^3 - x$", $grid = 10$); .../$/noticks, axesonly$

figure(0):

Parameter lt;frame mematikan frame, dan framecolor=blue mengatur frame ke warna biru.

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

aspect(1.5);

plot2d("x$^3 - x$", $grid = 0$); //$plot$

frame; xgrid([-1,0,1]); ygrid(0): // add frame and grid

Untuk judul plot dan label sumbu, lihat contoh berikut.

plot2d("exp(x)",-1,1);
textcolor(black); // set the text color to black
title(latex("y=e^x")); //*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

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

plot2d("x$^3 - x$", $< grid, < frame$);
xaxis(0,xx=-2:1,style="- "); yaxis(0,yy=-5:5,style="- "):

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

function f(x) = x$^3 - x$
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

Ada juga kotak teks.

plot2d(f(x),-1,1,-2,2); // function
plot2d(diff(f(x),x), add,style="–",color=red); // derivative
labelbox(["f","f'"],["-","–"],[black,red]): // label box
plot2d(["exp(x)","1+x"],color=[black,blue],style=["-","-."]):
gridstyle("- ",color=gray,textcolor=gray,framecolor=gray); ... plot2d("x$^3-$ x", $grid = 1$); ... $settitle$("$y = x^3 - x$", $color = black$); ... $label$("x", 2, 0, $pos =$ "bc", $color = gray$); ... $label$("y", 0, 6, $pos =$ "cl", $color = gray$); ... $reset$() :

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

Perintah fullwindow() memperluas jendela plot karena kita tidak lagi membutuhkan tempat untuk label di luar jendela plot. Gunakan shrinkwindow() atau reset() untuk mengatur ulang ke default.

fullwindow; ... gridstyle(color=darkgray,textcolor=darkgray); ... plot2d(["2^x","1","2$^(-$ x)"], $a = -2, b = 2, c = 0, d = 4, < grid, color = 4 : 6, < frame$); ...$xaxis$(0, −2 : 1, $style =$ "−"); $xaxis$(0, 2, "x", $< axis$); ...$yaxis$(0, 4, "y", $style =$ "−"); ...$yaxis$(−2, 1 : 4, $left$); ...$yaxis$(2, 2$^(-2 : 2), style =$ ".", $< left$); ...$labelbox$(["2^x","1","2^-x"], $colors = 4 : 6, x = 0.8, y = 0.2$); ... $reset$:

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

aspect(1.5);
plot2d(["sin(x)","cos(x)"],0,2pi,color=[red,green],¡grid,¡frame); ... xaxis(-1.1,(0:2)pi,xt=["0",u"pi;",u"2pi;"],style="-", ticks, zero); ... xgrid((0:0.5:2) × pi,¡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;)"):

Merencanakan Data 2D

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

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

Memplot ekspresi hanyalah singkatan untuk 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 campuran dan titik gunakan "gt;tambahan".

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)$x; plot2d(x, y)$:

Data juga dapat diplot sebagai titik. Gunakan poin=true untuk ini. Plotnya bekerja seperti poligon, tetapi hanya menggambar sudut-sudutnya.

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

Untuk memplot set poin gunakan gt;points. Jika warna adalah vektor warna, setiap titik

mendapat warna yang berbeda. Untuk matriks koordinat dan vektor kolom, warna berlaku untuk 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
p=polyfit(xdata,ydata,1); // get regression line
plot2d("polyval(p,x)", add,color=red): // add plot of line
```

Menggambar Daerah Yang Dibatasi Kurva

Plot data benar-benar poligon. Kita juga dapat memplot kurva atau kurva terisi.

* terisi=benar mengisi plot.

* style="...": Pilih dari "", "/", ";; "".

* fillcolor: Lihat di atas untuk warna yang tersedia.

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

```
t=linspace(0,2pi,1000); // parameter for curve
x=sin(t)exp(t/pi); y=cos(t)exp(t/pi); // x(t) and y(t)
figure(1,2); aspect(16/9)
figure(1); plot2d(x,y,r=10); // plot curve
figure(2); plot2d(x,y,r=10, filled,style="/",fillcolor=red); // fill curve
figure(0):
```

Dalam contoh berikut kami 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="/"):
t=linspace(0,2pi,6); ...  plot2d(cos(t),sin(t), filled,style="/",fillcolor=red,r=1.2):
t=linspace(0,2pi,6); plot2d(cos(t),sin(t), filled,style=""):
```

Contoh lainnya adalah segi empat, 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):

Berikut ini 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):

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

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

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

sebagai kurva yang menghubungkan titik-titik. Plotnya bisa diisi. Pada kasus ini

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

isi.

plot2d(x,y,¡grid,¡frame, filled):

Sebuah vektor interval diplot terhadap nilai x sebagai daerah terisi

antara nilai interval bawah dan atas.

Hal ini dapat berguna untuk memplot kesalahan perhitungan. Tapi itu bisa juga digunakan untuk memplot kesalahan statistik.

t=0:0.1:1; ... plot2d(t,interval(t-random(size(t)),t+random(size(t))),style="—"); ... plot2d(t,t,add=true):

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

t=-1:0.01:1; x= t-0.01,t+0.01 ; y=$x^3 - x$;

plot2d(t,y):

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

expr := "$2x^2 + xy + 3y^4 + y$"; //$defineanexpressionf(x, y)$

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

Kami juga dapat mengisi rentang nilai seperti

plot2d("$(x^2 + y^2)^2 - x^2 + y^2$", $r = 1.2, level = [-1; 0], style = "/")$:

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

Grafik Fungsi Parametrik

Nilai-x tidak perlu diurutkan. (x,y) hanya menggambarkan kurva. Jika x diurutkan, kurva tersebut merupakan grafik fungsi.

Dalam contoh berikut, kami memplot spiral

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

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

Atau, 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):
t=linspace(0,1,1000); r=exp(-t); x=rcos(2pit); y=rsin(2pit);
plot2d(x,y,r=1):

Dalam contoh berikutnya, kami memplot kurva
dengan
t=linspace(0,2pi,1000); r=1+sin(3t)/2; x=rcos(t); y=rsin(t); ... plot2d(x,y, filled,fillcolor=red,style="/",r

Menggambar Grafik Bilangan Kompleks

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

Matriks bilangan kompleks akan secara otomatis diplot sebagai kisi di bidang kompleks.

Dalam contoh berikut, kami 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):
aspect(1.25); r=linspace(0,1,50); a=linspace(0,2pi,200)'; z=rexp(Ia);
plot2d(exp(z),cgrid=[40,10]):
r=linspace(0,1,10); a=linspace(0,2pi,40)'; z=rexp(Ia);
plot2d(exp(z), points, add):

Sebuah vektor bilangan kompleks secara otomatis diplot sebagai kurva pada bidang kompleks dengan bagian real dan bagian imajiner.

Dalam contoh, kami memplot lingkaran satuan dengan
t=linspace(0,2pi,1000); ... plot2d(exp(It)+exp(4It),r=2):

Plot Statistik

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

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

plot2d(cumsum(randnormal(1,1000))):

Menggunakan dua baris menunjukkan jalan dalam dua dimensi.

X=cumsum(randnormal(2,1000)); plot2d(X[1],X[2]):
columnsplot(cumsum(random(10)),style="/",color=blue):

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];
columnsplot(values,lab=months,color=red,style="-");
title("Temperature"):
k=0:10;
plot2d(k,bin(10,k), bar):
plot2d(k,bin(10,k)); plot2d(k,bin(10,k), points, add):
plot2d(normal(1000),normal(1000), points,grid=6,style=".."):
plot2d(normal(1,1000), distribution,style="O"):
plot2d("qnormal",0,5;2.5,0.5, filled):

Untuk memplot distribusi statistik eksperimental, Anda dapat menggunakan distribution=n dengan plot2d.

w=randexponential(1,1000); // exponential distribution

plot2d(w, distribution): // or distribution=n with n intervals

Atau Anda dapat menghitung distribusi dari data dan memplot hasilnya dengan 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):

Fungsi statplot() menyetel gaya dengan string sederhana.

statplot(1:10,cumsum(random(10)),"b"):

n=10; i=0:n; ... plot2d(i,bin(n,i)/$2^n, a = 0, b = 10, c = 0, d = 0.3$); ...$plot2d(i, bin(n, i)/2^n, points = true, style = "ow", add = true, color = blue)$:

Selain itu, data dapat diplot sebagai batang. Dalam hal ini, x harus diurutkan dan satu elemen lebih panjang dari y. Bilah akan memanjang dari x[i] ke x[i+1] dengan nilai y[i]. Jika x memiliki ukuran yang sama dengan y, maka 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):

Data untuk plot batang (bar=1) dan histogram (histogram=1) dapat dinyatakan 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;genap ditentukan, nilai xv akan dihitung dalam interval bilangan bulat.

plot2d(normal(10000),distribution=50):

k=0:10; m=bin(10,k); x=(0:11)-0.5; plot2d(x,m, bar):

columnsplot(m,k):

plot2d(random(600)6,histogram=6):

Untuk distribusi, ada parameter distribution=n, yang menghitung nilai secara otomatis dan mencetak distribusi relatif dengan n sub-interval.

plot2d(normal(1,1000),distribution=10,style="
/"):

Dengan parameter even=true, ini akan menggunakan interval integer.

plot2d(intrandom(1,1000,10),distribution=10,even=true):

Perhatikan bahwa ada banyak plot statistik, yang mungkin berguna. Silahkan lihat tutorial tentang statistik.

columnsplot(getmultiplicities(1:6,intrandom(1,6000,6))):

plot2d(normal(1,1000), distribution); ... plot2d("qnormal(x)",color=red,thickness=2, add):

Ada juga banyak plot khusus untuk statistik. Boxplot menunjukkan kuartil dari distribusi ini dan banyak outlier. Menurut definisi, outlier dalam boxplot adalah data yang melebihi 1,5 kali kisaran 50tengah plot.

M=normal(5,1000); boxplot(quartiles(M)):

Fungsi Implisit

Plot implisit menunjukkan garis level yang menyelesaikan f(x,y)=level, di mana "level" dapat berupa nilai tunggal atau vektor nilai. Jika level="auto",

akan ada garis level nc, yang akan menyebar antara 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 dari parameter x dan y, atau, sebagai alternatif, xv dapat berupa matriks nilai.

Euler dapat menandai garis level

dari fungsi apapun.

Untuk menggambar himpunan f(x,y)=c untuk satu atau lebih konstanta c, Anda dapat menggunakan plot2d() dengan plot implisitnya di dalam bidang. Parameter untuk c adalah level=c, di mana c dapat berupa vektor garis level. Selain itu, skema warna dapat digambar di latar belakang untuk menunjukkan nilai fungsi untuk setiap titik dalam plot. Parameter "n" menentukan kehalusan plot.

aspect(1.5);

plot2d("x$^2 + y^2 - xy - x$", $r = 1.5, level = 0, contourcolor = red$) :

expr := "2x$^2 + xy + 3y^4 + y$"; //$defineanexpressionf(x, y)$

plot2d(expr,level=0): // Solutions of f(x,y)=0

plot2d(expr,level=0:0.5:20, hue,contourcolor=white,n=200): // nice

plot2d(expr,level=0:0.5:20, hue, spectral,n=200,grid=4): // nicer

Ini berfungsi untuk plot data juga. Tetapi 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):

plot2d("x$^3 - y^2$", $contour, hue, spectral$) :

plot2d("x$^3 - y^2$", $level = 0, contourwidth = 3, add, contourcolor = red$) :

z=z+normal(size(z))0.2;

plot2d(z,level=0.5,a=-1,b=2,c=-2,d=1):

plot2d(expr,level=[0:0.2:5;0.05:0.2:5.05],color=lightgray):

plot2d("x$^2 + y^3 + xy$", $level = 1, r = 4, n = 100$) :

plot2d("x$^2 + 2y^2 - xy$", $level = 0 : 0.1 : 10, n = 100, contourcolor = white, hue$) :

Juga dimungkinkan untuk mengisi set

dengan rentang tingkat.

Dimungkinkan untuk mengisi wilayah nilai untuk fungsi tertentu. Untuk ini, level harus berupa matriks 2xn. Baris pertama adalah batas bawah dan baris kedua berisi batas atas.

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

Plot implisit juga dapat menunjukkan rentang level. Kemudian level harus berupa matriks 2xn dari interval level, di mana baris pertama berisi awal dan baris kedua adalah akhir dari setiap interval. Atau, vektor baris sederhana dapat digunakan untuk level, dan parameter dl memperluas nilai level ke interval.

plot2d("x$^4 + y^4$", $r = 1.5, level = [0; 1], color = blue, style = "/"$) :

plot2d("x$^2 + y^3 + xy$", $level = [0, 2, 4; 1, 3, 5], style = "/", r = 2, n = 100$) :

plot2d("x$^2 + y^3 + xy$", $level = -10 : 20, r = 2, style = " - ", dl = 0.1, n = 100$) :

plot2d("sin(x)cos(y)",r=pi, hue, levels,n=100):

Dimungkinkan juga untuk menandai suatu wilayah

Ini dilakukan dengan menambahkan level dengan dua baris.

plot2d("(x^2+y^2−1)^3−x^2y^3", $r = 1.3, ...style = "", color = red, < outline, ...level = [-2; 0], n = 100)$:

Dimungkinkan untuk menentukan level tertentu. Misalnya, kita dapat memplot solusi persamaan seperti

plot2d("x^3 − $xy + x^2y^2$", $r = 6, level = 1, n = 100)$:

function starplot1 (v, style="/", color=green, lab=none) ... ¡pre class="udf"¿ if !holding() then clg; endif; w=window(); window(0,0,1024,1024); h=holding(1); r=max(abs(v))*1.2; setplot(-r,r,-r,r); n=cols(v); t=linspace(0,2pi,n); v=v—v[1]; c=v*cos(t); s=v*sin(t); cl=barcolor(color); st=barstyle(style); loop 1 to n polygon([0,c[],c[+1]],[0,s[],s[+1]],1); if lab!=none then rlab=v[]+r*0.1; col,row=toscreen(cos(t[])*rlab,sin(t[])*rlab); ctext(""+lab[],col,row-textheight()/2); endif; end; barcolor(cl); barstyle(st); holding(h); window(w); endfunction ¡/pre¿ Tidak ada kotak atau sumbu kutu di sini. Selain itu, kami menggunakan jendela penuh untuk plot.

Kami memanggil reset sebelum kami menguji plot ini untuk mengembalikan default grafis. Ini tidak perlu, jika Anda yakin plot Anda berhasil.

reset; starplot1(normal(1,10)+5,color=red,lab=1:10):

Terkadang, Anda mungkin ingin merencanakan sesuatu yang tidak dapat dilakukan plot2d, tetapi hampir.

Dalam fungsi berikut, kami melakukan 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,¿bar,grid=0); h=holding(1); frame(); xgrid(ticks(x)); p=plot(); for i=-10 to 10; if i¡=p[4] and i¿=p[3] then ygrid(i,yt="10"+$i); endif; end; holding(h); endfunction < /pre > Marikitaujidengannilaiyangterdistribusis$

aspect(1.5); x=1:10; y=-log(random(size(x)))200; ... logimpulseplot1(x,y):

Mari kita menganimasikan kurva 2D menggunakan plot langsung. Perintah plot(x,y) hanya memplot kurva ke jendela plot. setplot(a,b,c,d) mengatur jendela ini.

Fungsi wait(0) memaksa plot untuk muncul di jendela grafik. Jika tidak, menggambar ulang terjadi dalam interval waktu yang jarang.

function animliss (n,m) ...

t=linspace(0,2pi,500); f=0; c=framecolor(0); l=linewidth(2); setplot(-1,1,-1,1); repeat clg; plot(sin(n*t),cos(m*t+f)); wait(0); if testkey() then break; endif; f=f+0.02; end; framecolor(c); linewidth(l); endfunction ¡/pre¿ Tekan sembarang tombol 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 baik menggunakan skala logaritma dalam y dengan logplot=1, atau menggunakan skala logaritma dalam x dan y dengan logplot=2, atau dalam x dengan logplot=3.

- logplot=1: y-logaritma - logplot=2: x-y-logaritma - logplot=3: x-logaritma

plot2d("exp(x^3 − x)x^2", $1, 5, logplot = 1)$:

plot2d("exp(x+sin(x))",0,100,logplot=1):

plot2d("exp(x+sin(x))",10,100,logplot=2):
plot2d("gamma(x)",1,10,logplot=1):
plot2d("log(x(2+sin(x/100)))",10,1000,logplot=3):

Ini juga berfungsi dengan plot data.

x=10^(1:20); $y = x^2 - x$;
plot2d(x,y,logplot=2):

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"
"[]", "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
verticallabels locally for one plot. The value 1 sets only vertical
text, the value 2 uses vertical numerical labels on the y axis.
filled : fill the plot of a curve
fillcolor : fill color for bar and filled curves
outline : boundary for filled polygons
logplot : set logarithmic plots
1 = logplot in y,
2 = logplot in xy,
3 = logplot in x
own :
A string, which points to an own plot routine. With 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 Plot3D

Romadhona Enggal Wilujeng$_2$3030630074$_PLOT3DNama$: $RomadhonaEnggalWilujeng$

NIM : 23030630074

Kelas : Matematika E

Menggambar Plot 3D dengan EMT

Ini adalah pengenalan plot 3D di Euler. Kita membutuhkan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

Euler menggambar fungsi tersebut menggunakan algoritma pengurutan untuk menyembunyikan bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Standarnya adalah dari kuadran x-y positif menuju titik asal x=y=z=0, tetapi sudut=0° terlihat dari arah sumbu y. Sudut pandang dan ketinggian dapat diubah.

Euler dapat merencanakan

* permukaan dengan bayangan dan garis level atau rentang level,
* awan poin,
* kurva parametrik,
* permukaan implisit.

Plot 3D dari suatu fungsi menggunakan plot3d. Cara termudah adalah dengan memplot ekspresi dalam x dan y. Parameter r mengatur kisaran plot di sekitar (0,0).

aspect(1.5); plot3d("x^2 + $sin(y)$", $r = pi$) :

aspect(1.5); plot3d("x^2 + $sin(y)$", $0, 10, 0, 2pi$) :

aspect(1.5); plot3d("x^2 + $sin(y)$", $-5, 5, 0, 2pi$) :

aspect(1.5); plot3d("x^2 + $sin(y)$", $-5, 5, 0, 6pi$) :

aspect(1.5); plot3d("x^2 + $sin(y)$", $-2, 7, 0, 5pi$) :

aspect(1.5); plot3d("x$^2 sin(y)$", $-5, 5, 0, 2pi$) :

aspect(1.5); plot3d("x^2 − $sin(y)$", $-5, 5, 0, 2pi$) :

aspect(1.5); plot3d("x$^2(1 + sin(y))$", $-5, 5, 0, 6pi$) :

aspect(1.5); plot3d("x^3 + $sin(y)$", $-5, 5, 0, 2pi$) :

Fungsi dua Variabel

Untuk grafik fungsi, gunakan

* ekspresi sederhana dalam x dan y,
* nama fungsi dari dua variabell
* atau matriks data.

Standarnya adalah kisi kawat yang diisi dengan warna berbeda di kedua sisi. Perhatikan bahwa jumlah default interval grid adalah 10, tetapi plot menggunakan jumlah default 40x40 persegi panjang untuk membangun permukaan. Ini bisa 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$^2 + y^2$") :

plot3d("x$^3 + y^3$") :

Interaksi pengguna dimungkinkan dengan gt;parameter pengguna. Pengguna dapat menekan tombol berikut.

* kiri, kanan, atas, bawah: putar sudut pandang
* +,-: memperbesar atau memperkecil
* a: menghasilkan anaglyph (lihat di bawah)
* l: beralih memutar sumber cahaya (lihat di bawah)
* spasi: reset ke default
* kembali: akhiri interaksi

plot3d("exp(-x^2+y^2)", $user, ...title = "Turnwiththevectorkeys(pressreturntofinish)")$:
plot3d("exp(-x^3+y^4)", $user, ...title = "Turnwiththevectorkeys(pressreturntofinish)")$:
plot3d("exp(x^5+y^2)", $user, ...title = "Turnwiththevectorkeys(pressreturntofinish)")$:

Rentang plot untuk fungsi dapat ditentukan dengan
* a,b: rentang-x
* c,d: rentang-y
* r: persegi simetris di sekitar (0,0).
* n: jumlah subinterval untuk plot.

Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: skala ke nilai fungsi (defaultnya adalah lt;fscale).

skala: angka atau vektor 1x2 untuk skala ke arah x dan y.

bingkai: jenis bingkai (default 1).

plot3d("exp(-(x^2+y^2)/5)", $r = 10, n = 80, fscale = 4, scale = 1.2, frame = 3)$:

plot3d("exp(-(x^2+y^2)/3)", $r = 10, n = 80, fscale = 4, scale = 1.2, frame = 3)$:

plot3d("exp(-(x^2 + y)/5)", $r = 10, n = 80, fscale = 4, scale = 1.2, frame = 3)$:

Tampilan dapat diubah dengan berbagai cara.
* jarak: jarak pandang ke plot.
* zoom: nilai zoom.
* sudut: sudut terhadap sumbu y negatif dalam radian.
* tinggi: ketinggian tampilan 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.

Dalam contoh berikut, sudut=0 dan tinggi=0 terlihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

plot3d("x^2 + y", $distance = 3, zoom = 2, angle = 0, height = 0)$:
plot3d("x^3 + y^2", $distance = 2, zoom = 2, angle = 0, height = 0)$:
plot3d("x+y",distance=3,zoom=1,angle=0,height=0):

Plot terlihat selalu ke pusat kubus plot. Anda dapat memindahkan pusat dengan parameter tengah.

plot3d("x^4 + y^2", $a = 0, b = 1, c = -1, d = 1, angle = -20, height = 20, ... center = [0.4, 0, 0], zoom = 5)$:

plot3d("$x^6 + y^3$", $a = 0, b = 1, c = -1, d = 1, angle = 60, height = 15, ... center = [0.6, 0, 0], zoom = 4$) :

plot3d("$x^3 + y^2$", $a = 0, b = 1, c = -1, d = 1, angle = 90, height = 45, ... center = [0.5, 0, 0], zoom = 5$) :

Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung pada ukuran plot. Namun, label mengacu pada ukuran sebenarnya.

Jika Anda mematikannya dengan scale=false, Anda harus berhati-hati, agar plot tetap pas dengan jendela plot, dengan mengubah jarak pandang atau zoom, dan memindahkan bagian tengahnya.

plot3d("5exp(-$x^2 - y^2$)", $r = 2, < fscale, < scale, distance = 13, height = 50, ... center = [0, 0, -2], frame = 3$) :

plot3d("5exp(-$x^2 - y^2$)", $r = 1, < fscale, < scale, distance = 15, height = 30, ... center = [0, 0, -2], frame = 3$) :

plot3d("3exp(-$x^2 - y^2$)", $r = 1, < fscale, < scale, distance = 15, height = 70, ... center = [0, 0, 2], frame = 3$) :

Sebuah plot kutub juga tersedia. Parameter polar=true menggambar plot polar. Fungsi tersebut harus tetap merupakan fungsi dari x dan y. Parameter "fscale" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi diskalakan agar sesuai dengan kubus.

plot3d("$1/(x^2 + y^2 + 1)$", $r = 5,\ polar, ... fscale = 2,\ hue, n = 100, zoom = 4,\ contour, color = gray$) :

plot3d("$1/(x^2 + y^2)$", $r = 5,\ polar, ... fscale = 2,\ hue, n = 100, zoom = 3,\ contour, color = blue$) :

plot3d("$1/(x^2 + y^2 + 2)$", $r = 5,\ polar, ... fscale = 2,\ hue, n = 100, zoom = 5,\ contour, color = black$) :

function f(r) := exp(-r/2)cos(r); ... plot3d("f($x^2 + y^2$)", $polar, scale = [1, 1, 0.4], r = 2pi, frame = 3, zoom = 4$) :

function f(r) := exp(r/2)cos(r); ... plot3d("f($x^2 + y^2$)", $polar, scale = [1, 1, 0.4], r = 2pi, frame = 2, zoom = 4$) :

function f(r) := exp(-r/2)cos(r); ... plot3d("f($x^2 + y^2$)", $polar, scale = [1, 1, 0.4], r = pi, frame = 3, zoom = 4$) :

Rotasi parameter memutar fungsi dalam x di sekitar sumbu x.

* rotate=1: Menggunakan sumbu x
* rotate=2: Menggunakan sumbu z

plot3d("$x^2 + 1$", $a = -1, b = 1, rotate = true, grid = 5$) :

plot3d("$x^3 + 2$", $a = -1, b = 1, rotate = true, grid = 5$) :

plot3d("$x^4 + 3$", $a = -1, b = 1, rotate = true, grid = 6$) :

plot3d("x","$x^2 + y^2$","y", $r = 2, zoom = 3.5, frame = 3$) :

plot3d("x","$x^2 + y^2$","y", $r = 2, zoom = 2.5, frame = 3$) :

plot3d("x","$x^2 + y$","y", $r = 2, zoom = 3, frame = -3$) :

Plot Kontur

Untuk plot, Euler menambahkan garis grid. Sebagai gantinya dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona berwarna spektral. Euler dapat menggambar tinggi fungsi pada plot dengan bayangan. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah/sian.

-gt; hue: Menyalakan bayangan cahaya alih-alih kabel.

-gt; kontur: Memplot garis kontur otomatis pada plot.

- level=... (atau level): Sebuah vektor nilai untuk garis kontur.

Standarnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut, kami menggunakan grid yang lebih halus untuk 100x100 poin, skala fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

plot3d("exp(-x$^2-y^2$)", $r = 2, n = 100, level =$ "$thin$", $...contour, spectral, fscale = 1, scale = 1.1, angle = 45, height = 20$) :

plot3d("exp(x$^2-y^2$)", $r = 2, n = 100, level =$ "$thin$", $...contour, spectral, fscale = 1, scale = 1.1, angle = 90, height = 15$) :

plot3d("exp(xy)",angle=100°, contour,color=green):

plot3d("exp(xy)",angle=120°, contour,color=blue):

Bayangan default menggunakan warna abu-abu. Tetapi rentang warna spektral juga tersedia.

-gt; spektral: 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 halus.

plot3d("x$^2 + y^2$", $spectral$, $contour, n = 100$) :

plot3d("x$^4 + y^3$", $spectral$, $contour, n = 100$) :

Alih-alih garis level otomatis, kita juga dapat mengatur nilai garis level. Ini akan menghasilkan garis level tipis alih-alih rentang level.

plot3d("x$^2-y^2$", $0, 1, 0, 1, angle = 220, level = -1 : 0.2 : 1, color = redgreen$) :

plot3d("x$^4 - y^4$", $0, 1, 0, 1, angle = 110, level = -1 : 0.2 : 1, color = blue$) :

plot3d("x$^4 - y^3$", $0, 1, 0, 1, angle = 100, level = -1 : 0.2 : 1, color = red$) :

plot3d("x$^2-y^2$", $0, 5, 0, 5, angle = 220, level = -1 : 0.1 : 1, color = redgreen$) :

plot3d("x$^2-y^2$", $0, 5, 0, 5, angle = 100, level = -1 : 0.1 : 1, color = redgreen$) :

plot3d("x$^2 - y^2$", $-5, 5, 5, 5, level = -1 : 0.2 : 1, color = redgreen$) :

Dalam plot berikut, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas level sebagai kolom.

Selain itu, kami melapisi kisi dengan 10 interval di setiap arah.

plot3d("x$^2 + y^3$", $level = [-0.1, 0.9; 0, 1], ...$ $spectral, angle = 30, grid = 10, contourcolor = gray$) :

plot3d("x$^4 + y^5$", $level = [-0.1, 0.9; 0, 1], ...$ $spectral, angle = 60, grid = 10, contourcolor = gray$) :

Dalam contoh berikut, kami memplot himpunan, di mana

Kami menggunakan satu garis tipis untuk garis level.

plot3d("x$^y - y^x$", $level = 0, a = 0, b = 6, c = 0, d = 6, contourcolor = red, n = 100$) :

plot3d("x$^2y - 2y^x$", $level = 0, a = 0, b = 6, c = 0, d = 6, contourcolor = green, n = 100$) :

plot3d("x^y − y^x", level = 0, a = 0, b = 6, c = 0, d = 6, contourcolor = green, n = 100) :

Dimungkinkan untuk menunjukkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

plot3d("$x^2 + y^4$", cp, cpcolor = green, cpdelta = 0.2) :

plot3d("$x^3 + y^6$", cp, cpcolor = black, cpdelta = 0.2) :

plot3d("$x^2 + y^4$", cp, cpcolor = red, cpdelta = 0.5) :

Berikut adalah beberapa gaya lagi. Kami selalu mematikan frame, dan menggunakan berbagai skema warna untuk plot dan grid.

figure(2,2); ... expr="$y^3 - x^2$"; ...figure(1); ...plot3d(expr, < frame, cp, cpcolor = spectral); ...figure(2); ...plot3d(expr, < frame, spectral, grid = 10, cp = 2); ...figure(3); ...plot3d(expr, < frame, contour, color = gray, nc = 5, cp = 3, cpcolor = greenred); ...figure(4); ...plot3d(expr, < frame, hue, grid = 10, transparent, cp, cpcolor = gray); ... figure(0) :

figure(2,2); ... expr="$x^2 - y^2$"; ...figure(1); ...plot3d(expr, < frame, cp, cpcolor = spectral); ...figure(2); ...plot3d(expr, < frame, spectral, grid = 10, cp = 2); ...figure(3); ...plot3d(expr, < frame, contour, color = gray, nc = 5, cp = 3, cpcolor = greenred); ...figure(4); ...plot3d(expr, < frame, hue, grid = 10, transparent, cp, cpcolor = gray); ... figure(0) :

figure(2,2); ... expr="$x^3 - y^2$"; ...figure(1); ...plot3d(expr, < frame, cp, cpcolor = spectral); ...figure(2); ...plot3d(expr, < frame, spectral, grid = 10, cp = 2); ...figure(3); ...plot3d(expr, < frame, contour, color = gray, nc = 5, cp = 3, cpcolor = greenred); ...figure(4); ...plot3d(expr, < frame, hue, grid = 10, transparent, cp, cpcolor = gray); ... figure(0) :

Ada beberapa skema spektral lainnya, bernomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan warna=nilai, di mana nilai

* spektral: untuk rentang dari biru ke merah
* putih: untuk rentang yang lebih redup
* kuningbiru,ungu hijau,birukuning,hijaumerah
* birukuning, hijau ungu, kuning biru, merah hijau

figure(3,3); ... for i=1:9; ... figure(i); plot3d("$x^2 + y^2$", spectral = i, contour, cp, < frame, zoom = 4); ... end; ... figure(0) :

figure(4,4); ... for i=1:9; ... figure(i); plot3d("$x^3 + y^3$", spectral = i, contour, cp, < frame, zoom = 4); ... end; ... figure(0) :

figure(3,3); ... for i=1:10; ... figure(i); plot3d("$x^4 - y^4$", spectral = i, contour, cp, < frame, zoom = 4); ... end; ... figure(0) :

Sumber cahaya dapat diubah dengan l dan tombol kursor selama interaksi pengguna. Itu juga dapat diatur dengan parameter.

* cahaya: arah untuk cahaya
* amb: cahaya sekitar antara 0 dan 1

Perhatikan bahwa program tidak membuat perbedaan antara sisi plot. Tidak ada bayangan. Untuk ini, Anda perlu Povray.

plot3d("-$x^2 - y^2$", ...hue = true, light = [0, 1, 1], amb = 0, user = true, ...title = "Presslandcursorkeys(returntoexit)") :

plot3d("$x^2 + y^2$", ...hue = true, light = [0, 1, 1], amb = 0, user = true, ...title = "Presslandcursorkeys(returntoexit)") :

Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

plot3d("-x$^2-y^2$", $color = rgb(0.2, 0.2, 0), hue = true, frame = false, ...zoom = 3, contourcolor = red, level = -2 : 0.1 : 1, dl = 0.01)$:

plot3d("x$^2-y^2$", $color = rgb(0.2, 0.2, 0), hue = true, frame = false, ...zoom = 3, contourcolor = green, level = -2 : 0.1 : 1, dl = 0.01)$:

plot3d("x$^2+y^2$", $color = rgb(0.2, 0.2, 0), hue = true, frame = false, ...zoom = 3, contourcolor = blue, level = -2 : 0.1 : 1, dl = 0.01)$:

plot3d("x$^2+y^4$", $color = rgb(0.2, 0.2, 0), hue = true, frame = false, ...zoom = 3, contourcolor = yellow, level = -2 : 0.1 : 1, dl = 0.01)$:

plot3d("x$^2+y^3$", $color = rgb(0.2, 0.2, 0), hue = true, frame = false, ...zoom = 2, contourcolor = green, level = -2 : 0.1 : 1, dl = 0.01)$:

Warna 0 memberikan efek pelangi khusus.

plot3d("x$^2/(x^2 + y^2 + 1)$", $color = 0, hue = true, grid = 10)$:

plot3d("x$^3/(x^3 + y^3 + 1)$", $color = 0, hue = true, grid = 10)$:

plot3d("x$^4/(x^4 + y^4 - 1)$", $color = 0, hue = true, grid = 10)$:

plot3d("x/(x+y-1)",color=0,hue=true,grid=10):

Permukaannya juga bisa transparan.

plot3d("x$^2 + y^2$", $transparent, grid = 10, wirecolor = red)$:

plot3d("x$^3 + y^3$", $transparent, grid = 15, wirecolor = yellow)$:

plot3d("x+y^3", $transparent, grid = 15, wirecolor = green)$:

plot3d("x$^5 - y^3$", $transparent, grid = 15, wirecolor = blue)$:

plot3d("x$^5 - y^2$", $transparent, grid = 13, wirecolor = red)$:

plot3d("x$^7 + y^3$", $transparent, grid = 20, wirecolor = blue)$:

plot3d("x$^8 - y^4$", $transparent, grid = 15, wirecolor = green)$:

plot3d("x$^10 - y^5$", $transparent, grid = 15, wirecolor = yellow)$:

plot3d("x$^7 + y^3$", $transparent, grid = 15, wirecolor = black)$:

Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan pemotongan melalui objek. Fitur plot3d termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari suatu fungsi dalam tiga variabel.

Solusi dari

dapat divisualisasikan dalam potongan sejajar dengan bidang x-y-, x-z- dan y-z.

* implisit=1: potong sejajar dengan bidang y-z
* implisit=2: potong sejajar dengan bidang x-z
* implisit=4: potong sejajar dengan bidang x-y

Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh kita plot

plot3d("x$^2 + y^3 + zy - 1$", $r = 5, implicit = 3)$:

plot3d("x$^3 + y^3 + zy - 2$", $r = 5, implicit = 4)$:

plot3d("x$^2 + y^2 + 4xz + z^3$", $implicit, r = 2, zoom = 2.5)$:

plot3d("x$^4 + y^4 + 5xz + z^2$", $implicit, r = 2, zoom = 3)$:

plot3d("x$^7 + y^2 + 3xz + z^4$", $implicit, r = 2, zoom = 2)$:

plot3d("x$^5 + y^9 - 3xz + z^5$", $implicit, r = 2, zoom = 2.5)$:

plot3d("x$^4 + y^5 + 5xz + z^2$", $implicit, r = 2, zoom = 2)$:

plot3d("x$^2 - y^2 + 4xz + z^3$", $implicit, r = 2, zoom = 2)$:

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) melalui sebuah kotak 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, kami menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ... x=cos(s)cos(t); y=cos(s) × sin(t); z=sin(s); ... plot3d(x,y,z, hue, ... color=blue,¡frame,grid=[10,20], ... values=s,contourcolor=red,level=[90°-24°;90°-22°], ... scale=1.4,height=50°):

t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ... x=cos(s)cos(t); y=cos(s) × sin(t); z=sin(s); ... plot3d(x,y,z, hue, ... color=blue,¡frame,grid=[15,30], ... values=s,contourcolor=green,level=[90°-24°;90°-22°], ... scale=1.4,height=50°):

t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ... x=cos(s)cos(t); y=cos(s) × sin(t); z=sin(s); ... plot3d(x,y,z, hue, ... color=blue,¡frame,grid=[10,20], ... values=s,contourcolor=red,level=[90°-20°;90°-18°], ... scale=1.4,height=50°):

Berikut adalah contoh, yang merupakan grafik fungsi.

t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,ts,grid=10):

t=-1:0.5:1; s=(-1:0.5:1)'; plot3d(t,s,ts,grid=10):

t=-2:0.2:2; s=(-2:0.2:2)'; plot3d(t,s,ts,grid=15):

t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,ts,grid=11):

Namun, kita bisa membuat segala macam permukaan. Berikut adalah permukaan yang sama dengan fungsi

plot3d(ts,t,s,angle=180°,grid=10):

plot3d(ts,t,s,angle=90°,grid=16):

plot3d(ts,t,s,angle=270°,grid=5):

plot3d(ts,t,s,angle=45°,grid=7):

plot3d(ts,t,s,angle=60°,grid=10):

Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

Dalam contoh berikut, kita membuat tampilan bayangan dari bola yang terdistorsi. Koordinat biasa untuk bola 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):

t=linspace(0,2pi,270); s=linspace(-pi/2,pi/2,80)'; ... 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):

Dalam percobaan saya ini ternyata dengan fungsi tersebut dapat membentuk sebuah bunga.

berikut beberapa bentuk contohnya

t=linspace(0,pi,320); s=linspace(-pi/2,pi/2,150)'; ... d=1+5(cos(2t)+cos(4 × s)); ... plot3d(cos(t)cos(s)d,sin(t)cos(s)d,sin(s)d,hue=1, ... light=[1,0,1],frame=0,zoom=5):

t=linspace(0,pi,320); s=linspace(-pi/2,pi/2,150)'; ... d=1+7(cos(2t)+cos(4 ×
s)); ... plot3d(cos(t)cos(s)d,sin(t)cos(s)d,sin(s)d,hue=1, ... light=[1,0,1],frame=0,zoom=5):
t=linspace(0,pi,260); s=linspace(-pi/2,pi/2,120)'; ... d=1+10(cos(3t)+cos(7 ×
s)); ... plot3d(cos(t)cos(s)d,sin(t)cos(s)d,sin(s)d,hue=1, ... light=[1,0,1],frame=0,zoom=5):
t=linspace(0,pi,260); s=linspace(-pi/2,pi/2,120)'; ... d=1+15(cos(7t)+cos(8 ×
s)); ... plot3d(cos(t)cos(s)d,sin(t)cos(s)d,sin(s)d,hue=1, ... light=[1,0,1],frame=0,zoom=5):
t=linspace(0,pi,270); s=linspace(-pi/2,pi/2,180)'; ... d=1+15(cos(10 ×
t)+cos(12s)); ... plot3d(cos(t)cos(s)d,sin(t)cos(s)d,sin(s)d,hue=1, ... light=[1,0,1],frame=0,zoom=5):

Tentu saja, titik cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan 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="."):
n=1000; ... plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="–
"):
n=1000; ... plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=".."):
n=500; ... plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=" ×
"):

Dimungkinkan juga untuk memplot kurva dalam 3D. Dalam hal ini, lebih mudah untuk menghitung titik-titik kurva. Untuk kurva di pesawat kami menggunakan urutan koordinat dan parameter wire=true.

t=linspace(0,8pi,500); ... plot3d(sin(t),cos(t),t/10, wire,zoom=3):
t=linspace(0,7pi,1000); ... plot3d(sin(t),cos(t),t/6, wire,zoom=3):
t=linspace(0,8pi,1000); ... plot3d(sin(t),cos(t),t/5, wire,zoom=3):
t=linspace(0,8pi,3000); ... plot3d(sin(t),cos(t),t/10, wire,zoom=3):
t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi, wire, ... linewidth=3,wirecolor=blue):
t=linspace(0,4pi,4000); plot3d(cos(t),sin(t),t/2pi, wire, ... linewidth=3,wirecolor=red):
t=linspace(0,4pi,4000); plot3d(cos(t),sin(t),t/pi, wire, ... linewidth=3,wirecolor=red):
X=cumsum(normal(3,1000)); ... plot3d(X[1],X[2],X[3], anaglyph, wire):
X=cumsum(normal(5,100)); ... plot3d(X[1],X[2],X[3], anaglyph, wire):
X=cumsum(normal(6,600)); ... plot3d(X[1],X[2],X[3], anaglyph, wire):

EMT juga dapat memplot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda memerlukan kacamata merah/sian.

plot3d("x$^2 + y^3$", *anaglyph*, *contour*, *angle* = 30) :
plot3d("x$^2 + y^2$", *anaglyph*, *contour*, *angle* = 60) :
plot3d("x$^3 - y^3$", *anaglyph*, *contour*, *angle* = 75) :
plot3d("x$^4 + y^5$", *anaglyph*, *contour*, *angle* = 60) :
plot3d("x$^2 + y^7$", *anaglyph*, *contour*, *angle* = 90) :
plot3d("x$^5 - y^3$", *anaglyph*, *contour*, *angle* = 80) :
plot3d("x$^2 + y^8$", *anaglyph*, *contour*, *angle* = 180) :
plot3d("x$^9 - y^6$", *anaglyph*, *contour*, *angle* = 30) :

Seringkali, skema warna spektral digunakan untuk plot. Ini menekankan ketinggian fungsi.

plot3d("x$^2y^3 - y$", *spectral*, *contour*, *zoom* = 3.2) :
plot3d("x$^3y^4 + y$", *spectral*, *contour*, *zoom* = 3) :
plot3d("x$^6y^3 - y$", *spectral*, *contour*, *zoom* = 2.5) :
plot3d("x$^9y^4 + y$", *spectral*, *contour*, *zoom* = 2.5) :

plot3d("$x^3y^9 + 3y$", *spectral*, *contour*, *zoom* = 3.2) :

Euler juga dapat memplot permukaan berparameter, ketika parameternya adalah nilai x-, y-, dan z dari gambar kotak persegi panjang dalam ruang.

Untuk demo berikut, kami mengatur parameter u- dan v-, dan menghasilkan koordinat ruang dari ini.

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); ... plot3d(X,Y,Z, anaglyph,¡frame, wire,scale=2.3):

u=linspace(-1,1,10); v=linspace(0,2pi,100)'; ... X=(4+ucos(v/2))cos(v); Y=(4+ucos(v/2))sin(v); Z=usin(v/2); ... plot3d(X,Y,Z, anaglyph,¡frame, wire,scale=2.3):

u=linspace(-1,1,10); v=linspace(0,4pi,70)'; ... X=(5-ucos(v/2))cos(v); Y=(5-ucos(v/2))sin(v); Z=usin(v/2); ... plot3d(X,Y,Z, anaglyph,¡frame, wire,scale=2.3):

Berikut adalah contoh yang lebih rumit, yang megah dengan kacamata merah/sian.

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):

u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,500)'; ... x:=(4(2+.25sin(3 × v))+cos(u))cos(2v); ... y:=(4(2+.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=3, anaglyph):

u:=linspace(-pi,pi,170); v:=linspace(-pi,pi,700)'; ... x:=(5(1+.25sin(3 × v))+cos(u))cos(2v); ... y:=(5(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, anaglyph):

u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)'; ... x:=(4(1+.75sin(3 × v))+cos(u))cos(4v); ... y:=(4(1+.75sin(3v))+cos(u))sin(4v); ... 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):

Plot Statistik

Plot bar juga dimungkinkan. Untuk ini, kita harus menyediakan

* x: vektor baris dengan n+1 elemen
* y: vektor kolom dengan n+1 elemen
* z: matriks nilai nxn.

z bisa lebih besar, tetapi hanya nilai nxn yang akan digunakan.

Dalam contoh, pertama-tama kita menghitung nilainya. Kemudian kita sesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

x=-1:0.1:1; y=x'; z=x^2+y^2; ...$xa = (x|1.1) - 0.05; ya = (y\_1.1) - 0.05;$ ...$plot3d(xa, ya, z, bar = true)$:

x=-1:0.1:1; y=x'; z=x^3+y^3; ...$xa = (x|1.1) - 0.05; ya = (y\_1.1) - 0.05;$ ...$plot3d(xa, ya, z, bar = true)$:

x=-1:0.5:1; y=x'; z=x^2-y^2; ...$xa = (x|1.1) - 0.05; ya = (y\_1.1) - 0.05;$ ...$plot3d(xa, ya, z, bar = true)$:

x=-1:0.3:1; y=x'; z=x^4+y^4; ...$xa = (x|1.1) - 0.05; ya = (y\_1.1) - 0.05;$ ...$plot3d(xa, ya, z, bar = true)$:

Dimungkinkan untuk membagi plot permukaan menjadi dua atau lebih bagian.

x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ... plot3d(x,y,z,disconnect=2:2:20):

x=-1:0.5:1; y=x'; z=x+y; d=zeros(size(x)); ... plot3d(x,y,z,disconnect=2:2:20):

x=-2:0.2:2; y=x'; z=x+y; d=zeros(size(x)); ... plot3d(x,y,z,disconnect=2:2:10):

x=-3:0.3:3; y=x'; z=x+y; d=zeros(size(x)); ... plot3d(x,y,z,disconnect=2:2:30):

Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke [-1,1] dengan scale(M), atau menskalakan matriks dengan gt;zscale. Ini dapat dikombinasikan dengan faktor penskalaan individu yang diterapkan sebagai tambahan.

i=1:20; j=i'; ... plot3d(ij^2+100$normal(20,20), zscale, scale = [1,1,1.5], angle = -40, zoom = 1.8)$:

i=1:20; j=i'; ... plot3d(ij^2+100$normal(20,20), zscale, scale = [1,1,2], angle = -50, zoom = 1.8)$:

Z=intrandom(5,100,6); v=zeros(5,6); ... loop 1 to 5; v[]=getmultiplicities(1:6,Z[]); end; ... columnsplot3d(v',scols=1:5,ccols=[1:5]):

Permukaan Benda Putar

plot2d("(x^2+$y^2-1)^3-x^2y^3$", $r = 1.3, ...style = "", color = red, < outline, ...level = [-2;0], n = 100)$:

ekspresi = $(x^2 + y^2 - 1)^3 - x^2y^3$;ekspresi

Kami ingin memutar kurva jantung di sekitar sumbu y. Berikut adalah ungkapan, yang mendefinisikan hati:

Selanjutnya kita atur

function fr(r,a) = ekspresi with [x=rcos(a),y=rsin(a)] — trigreduce; $fr(r, a)$

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang memecahkan r, jika a diberikan. Dengan fungsi itu 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,¡frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):

Berikut ini adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kami mendefinisikan fungsi, yang menggambarkan objek.

function f(x,y,z) ...

r=x^2+y^2; $return(r+z^2-1)^3-r*z^3; endfunction < /pre > plot3d("f(x, y, z)", ...xmin = 0, xmax = 1.2, ymin = -1.2, ymax = 1.2, zmin = -1.2, zmax = 1.4, ...implicit = 1, angle = -30, zoom = 2.5, n = [10, 60, 60],$ *anaglyph*) :

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 suka.

Meskipun Euler bukan program 3D, ia dapat menggabungkan beberapa objek dasar. Kami mencoba memvisualisasikan paraboloid dan garis singgungnya.

function myplot ...

y=0:0.01:1; x=(0.1:0.01:1)'; plot3d(x,y,0.2*(x-0.1)/2,¡scale,¡frame,¿hue, .. hues=0.5,¿contour,color=orange) h=holding(1); plot3d(x,y,(x^2+y^2)/2, $< scale, < frame, > contour, > hue); holding(h); endfunction <$ /pre > $Sekarang framedplot() menyediakan frame, dan mengaturtampilan.$

framedplot("myplot",[0.1,1,0,1,0,1],angle=-45°, ... center=[0,0,-0.7],zoom=6):

Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa plot3d() menyetel jendela ke fullwindow() secara default, tetapi plotcontourplane() mengasumsikan itu.

x=-1:0.02:1.1; y=x'; z=$x^2 - y^4$;

function myplot (x,y,z) ... ¡pre class="udf"¿ zoom(2); wi=fullwindow(); plotcontourplane(x,y,z,level="auto",¡scale); plot3d(x,y,z,¿hue,¡scale,¿add,color=white,level="thin"); window(wi); reset(); endfunction ¡/pre¿ myplot(x,y,z):

Animasi

Euler dapat menggunakan frame untuk menghitung animasi terlebih dahulu.

Salah satu fungsi yang memanfaatkan teknik ini adalah rotate. Itu dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi memanggil addpage() untuk setiap plot baru. Akhirnya itu menjiwai plot.

Silakan pelajari sumber rotasi untuk melihat lebih detail.

function testplot () := plot3d("x$^2 + y^3$"); ... *rotate*("*testplot*"); *testplot*() :

*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 ¡a href="http://www.povray.org/, dan meletakkan sub-direktori "bin" dari Povray ke jalur lingkungan, atau mengatur variabel "defaultpovray" dengan path lengkap yang menunjuk ke "pvengine.exe"."¿http://www.povray.org dan meletakkan sub-direktori "bin" dari Povray ke jalur lingkungan, atau mengatur variabel "defaultpovray" dengan path lengkap yang menunjuk ke "pvengine.exe".¡/a¿

Antarmuka Povray dari Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah current.pov, dan direktori default adalah eulerhome(), biasanya c:. Povray menghasilkan file PNG, yang dapat dimuat oleh Euler ke dalam buku catatan. 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-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 "look", yang membutuhkan string dengan kode Povray untuk tekstur dan hasil akhir 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, a nd x,y,z sumbu dalam arti tangan kanan.

Anda perlu memuat file povray.

load povray;

Pastikan, direktori bin Povray ada di jalurnya. Jika tidak, edit variabel berikut sehingga berisi path ke povray yang dapat dieksekusi.

defaultpovray="C:
Program Files
POV-Ray"
C:Files-Ray

Untuk kesan pertama, kami memplot fungsi sederhana. Perintah berikut menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk ray tracing 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 untuk dijalankan. Anda dapat menekan batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK di jendela Povray untuk mengakui dialog awal Povray.

pov3d("x$^2 + y^2$", $zoom = 3$);

exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome)$
$if povraythenpovray(currentfile, w, h, w/h); endif;$

pov3d("x$^3 + y^3$", $zoom = 3$);

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray :$
$exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d :$
$if povraythenpovray(currentfile, w, h, w/h); endif;$

Kita dapat membuat fungsi menjadi transparan dan menambahkan hasil akhir lainnya. Kami juga dapat menambahkan garis level ke plot fungsi.

pov3d("x^2+y^3", $axiscolor = red, angle = 20, ...look = povlook(blue, 0.2), level =$
$-1 : 0.5 : 1, zoom = 3.8$);

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray :$
$exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d :$
$if povraythenpovray(currentfile, w, h, w/h); endif;$

pov3d("x^2+y^4", $axiscolor = red, angle = 25, ...look = povlook(blue, 0.2), level =$
$-1 : 0.5 : 1, zoom = 3.8$);

Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.

Kami memplot himpunan titik di bidang kompleks, di mana produk dari jarak ke 1 dan -1 sama dengan 1.

pov3d("(x-1)$^2 + y^2$)((x + 1)$^2 + y^2$)/40", $r = 1.5, ... angle = -120, level =$
$1/40, dlevel = 0.005, light = [-1, 1, 1], height = 45, n = 50, ... < fscale, zoom =$
3.8);

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray :$
$exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d :$
$if povraythenpovray(currentfile, w, h, w/h); endif;$

pov3d("(x-1)$^3 + y^3$)((x + 1)$^2 + y^2$)/40", $r = 1.5, ... angle = -120, level =$
$1/40, dlevel = 0.005, light = [-1, 1, 1], height = 90, n = 50, ... < fscale, zoom =$
3.8);

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray :$
$exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d :$
$if povraythenpovray(currentfile, w, h, w/h); endif;$

Merencanakan dengan Koordinat

Alih-alih fungsi, kita dapat memplot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objek.

Dalam contoh kita memutar fungsi di sekitar sumbu z.

function f(x) := x^3-x+1; ...$x=-1:0.01:1; t=linspace(0,2pi,8)'$; ...$Z=x; X=cos(t)f(x); Y=sin(t)f(x)$; ...$pov3d(X,Y,Z,angle=40,height=20,axis=0,zoom=4,light=[10,-5,5])$;

Command was not allowed! exec: return $_e xec(program,param,dir,print,hidden,wait)$; $povray: exec(program,params,defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d: ifpovraythenpovray(currentfile,w,h,w/h); endif$;

function f(x) := x^4+x+1; ...$x=-1:0.01:1; t=linspace(0,2pi,8)'$; ...$Z=x; X=cos(t)f(x); Y=sin(t)f(x)$; ...$pov3d(X,Y,Z,angle=40,height=20,axis=0,zoom=4,light=[10,-5,5])$;

Command was not allowed! exec: return $_e xec(program,param,dir,print,hidden,wait)$; $povray: exec(program,params,defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d: ifpovraythenpovray(currentfile,w,h,w/h); endif$;

Dalam contoh berikut, kami 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(8pir)/3; ... pov3d(x,y,z,zoom=5,axis=0,add=povsphere([0,0,0.5],0.1,povlook(green)), ... w=500,h=300);

Command was not allowed! exec: return $_e xec(program,param,dir,print,hidden,wait)$; $povray: exec(program,params,defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d: ifpovraythenpovray(currentfile,w,h,w/h); endif$;

r=linspace(0,1,80); phi=linspace(0,pi,80)'; ... x=rcos(phi); y=rsin(phi); z=exp(-5r)cos(8pir)/3; ... pov3d(x,y,z,zoom=5,axis=0,add=povsphere([0,0,0.5],0.1,povlook(green)), ... w=500,h=300);

Command was not allowed! exec: return $_e xec(program,param,dir,print,hidden,wait)$; $povray: exec(program,params,defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d: ifpovraythenpovray(currentfile,w,h,w/h); endif$;

Dengan metode bayangan canggih dari Povray, sangat sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya di perbatasan dan dalam bayang-bayang triknya mungkin menjadi jelas.

Untuk ini, kita perlu menambahkan vektor normal di setiap titik matriks.

Z = x^2y^3

2 3 x y

Persamaan permukaannya adalah [x,y,Z]. Kami menghitung dua turunan ke x dan y ini dan mengambil produk silang sebagai normal.

dx = diff([x,y,Z],x); dy = diff([x,y,Z],y);

Kami mendefinisikan normal sebagai produk silang dari turunan ini, dan mendefinisikan fungsi koordinat.

N = crossproduct(dx,dy); NX = N[1]; NY = N[2]; NZ = N[3]; N,

3 2 2 [- 2 x y , - 3 x y , 1]

Kami hanya menggunakan 25 poin.

x=-1:0.5:1; y=x';
pov3d(x,y,Z(x,y),angle=10°, ... xv=NX(x,y),yv=NY(x,y),zv=NZ(x,y),¡shadow);

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d : ifpovraythenpovray(currentfile, w, h, w/h); endif;$

Berikut ini adalah simpul Trefoil yang dilakukan oleh A. Busser di Povray. Ada versi yang ditingkatkan dari ini dalam contoh.

Lihat: ContohSimpul — Simpul trefoil

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normal bagi kami. Pertama, ketiga 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 normal, yang merupakan produk silang dari dua turunan.

dn = crossproduct(dx,dy);

Kami sekarang mengevaluasi semua ini secara numerik.

x:=linspace(-

Vektor normal adalah evaluasi dari ekspresi simbolik dn[i] untuk i=1,2,3. Sintaks untuk ini adalah amp;"expression"(parameters). 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),axis=0,zoom=5,w=450,h=350, ... ¡shadow,look=povlook(gray), ... xv="dn[1]"(x,y), yv="dn[2]"(x,y), zv="dn[3]"(x,y));

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d : ifpovraythenpovray(currentfile, w, h, w/h); endif;$

Kami juga dapat menghasilkan grid dalam 3D.

povstart(zoom=4); ... x=-1:0.5:1; r=1-(x+1)^2/6; ... $t = (0 : 30 : 360)'; y = rcos(t); z = rsin(t); ...writeln(povgrid(x, y, z, d = 0.02, dballs = 0.05)); ...povend();$

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

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();

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

Objek Povray

Di atas, kami menggunakan pov3d untuk memplot permukaan. Antarmuka povray di Euler juga dapat menghasilkan objek Povray. Objek-objek ini disimpan sebagai string di Euler, dan perlu ditulis ke file Povray.

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(green)); ... c3=povcylinder(-povz,povz,1,povlook(blue)); ... String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

c1

cylinder lt;-1,0,0gt;, lt;1,0,0gt;, 1 texture pigment color rgb lt;0.564706,0.0627451,0.0627451gt; finish ambient 0.2

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.

Itu dilakukan oleh povlook(), yang mengembalikan string dengan kode Povray yang relevan. Kita dapat menggunakan warna Euler default, atau menentukan warna kita sendiri. Kami 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;

Command was not allowed! exec: return $_{e}xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

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) ... ¡pre class="udf"¿ 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 ¡/pre¿ povstart(fade=10,¡shadow);

fractal(-1,-1,-1,2,4);

povend();

Command was not allowed! exec: return $_{e}xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

Perbedaan memungkinkan memotong satu objek dari yang lain. Seperti persimpangan, ada bagian dari objek CSG Povray.

povstart(light=[5,-5,5],fade=10);

Untuk demonstrasi ini, kami mendefinisikan objek di Povray, alih-alih menggunakan string di Euler. Definisi ditulis ke file segera.

Koordinat kotak -1 berarti [-1,-1,-1].

povdefine("mycube",povbox(-1,1));

Kita dapat menggunakan objek ini di povobject(), yang mengembalikan string seperti biasa.

c1=povobject("mycube",povlook(red));

Kami menghasilkan kubus kedua, dan memutar dan menskalakannya sedikit.

c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ... rotate=xrotate(10°)+yrotate(10°), scale=1.2);

Kemudian 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();

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

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 output dari ekspresi Maxima atau Euler.

povstart(angle=70°,height=50°,zoom=4);

Buat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi.

writeln(povsurface("pow(x,2)y-pow(y,3)-pow(z,2)",povlook(green))); ... writeAxes(); ... povend();

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

Objek Jala

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.

Kami ingin memaksimalkan xy di bawah kondisi x+y=1 dan menunjukkan sentuhan tangensial dari garis level.

povstart(angle=-10°,center=[0.5,0.5,0.5],zoom=7);

Kami 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. Itu dapat menerima vektor normal seperti pov3d().

Berikut ini mendefinisikan objek mesh, dan langsung menulisnya 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 permukaan dikurangi dua cakram.

writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));

Tulis dua persimpangan.

writeln(povintersection([mesh,cl],povlook(red))); ... writeln(povintersection([mesh,ll],povlook(gray)));

Tulis titik maksimum.

writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2defaultpointsize));

Tambahkan sumbu dan selesaikan.

writeAxes(0,1,0,1,0,1,d=0.015); ... povend();

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

Anaglyph di Povray

Untuk menghasilkan anaglyph untuk kacamata merah/sian, Povray harus berjalan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi loadanaglyph().

Tentu saja, Anda memerlukan kacamata merah/sian untuk melihat contoh berikut dengan benar.

Fungsi pov3d() memiliki sakelar sederhana untuk menghasilkan anaglyphs.

pov3d("-exp(-x$^2-y^2$)/2", $r = 2, height = 45, anaglyph, ...center = [0, 0, 0.5], zoom =$ 3.5);

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.pov3d : ifpovraythenpovray(currentfile, w, h, w/h); endif;$

Jika Anda membuat adegan dengan objek, Anda perlu menempatkan generasi adegan ke dalam fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

function myscene ...

s=povsphere(povc,1); cl=povcylinder(-povz,povz,0.5); clx=povobject(cl,rotate=xrotate(90°)); cly=povobject(cl,rotate=yrotate(90°)); c=povbox([-1,-1,0],1); un=povunion([cl,clx,cly,c]); obj=povdifference(s,un,povlook(red)); writeln(obj); writeAxes(); endfunction ¡/pre¿ Fungsi povanaglyph() melakukan semua ini. Parameternya seperti di povstart() dan povend() digabungkan.

povanaglyph("myscene",zoom=4.5);

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povanaglyph : povray(currentfile, w, h, aspect, exit);$

Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak objek. Tapi Anda tidak terbatas pada ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau objek yang sama sekali baru.

Kami mendemonstrasikan sebuah torus. Perintah Povray untuk ini adalah "torus". Jadi kami mengembalikan string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat di titik asal.

function povdonat (r1,r2,look="") ...
return "torus "+r1+","+r2+look+""; endfunction ¡/pre¿ =Inilah 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 menempatkan objek-objek ini ke dalam sebuah adegan. Untuk tampilan, 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, itu tidak menampilkan kesalahan. 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);

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

Berikut adalah contoh yang lebih rumit. Kami memecahkan

dan menunjukkan titik layak dan optimal dalam plot 3D.

A=[10,8,4;5,6,8;6,3,2;9,5,6];
b=[10,10,10,10]';
c=[1,1,1];

Pertama, mari kita periksa, apakah contoh ini memiliki solusi sama sekali.

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 ¡/pre¿ Kemudian kita mendefinisikan persimpangan 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 ¡/pre¿ Kita sekarang 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 yang optimal.

writeln(povarrow(x,c0.5,povlook(red)));

Kami menambahkan teks ke layar. Teks hanyalah objek 3D. Kita perlu menempatkan dan memutarnya menurut pandangan kita.

writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=125°)); ... povend();

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

LATIHAN - LATIHAN SOAL

Fungsi Dua Variabel atau Lebih

Grafik dari fungsi f dengan dua variabel yang dimaksud adalah grafik dari persamaan z = f(x,y). Biasanya grafik ini berupa permukaan dan karena setiap (x,y) di daerah asal hanya berpadanan dengan satu nilai z, maka setiap garis tegaklurus bidang-xy memotong permukaan pada paling banyak satu titik.

Soal :

1. Grafik merupakan sebuah paraboloida f(x,y) = $y^2 - x^2$
2. z = -4x^3y^2
3. z = xy exp(-$x^2 - y^2$)
4. z = x - 1/8x^3 − 1/3y^2

aspect(1.5); plot3d("$y^2 - x^2$") :

aspect(1.5); plot3d("-4x^3y^2") :

plot3d("xyexp(-$x^2 - y^2$)", $r = 2, <$ $fscale, <$ $scale, distance = 13, height =$ 20, ... $center = [0, 0, -0.2], frame = 3$) :

aspect(1.5); plot3d("x-1/8$x^3 - 1/3y^2$") :

plot3d("sin(x)cos(y)"):

Membuat Talang (Perosotan pada kolam renang)

Dalam membuat perosotan kolam renang ini saya belum bisa membuat secara bagus, karena perosotan yang saya buat masih kurang jadi dan bentuknya pun masih dari arah kiri ke kanan, bahkan terlihat dari sisi belakang dan dari samping, belum terlihat dari depan.

plot3d("$x^3 - y^3$", $anaglyph$, $contour, angle = 75$) :

plot3d("$x^3 - y^3$", $anaglyph$, $contour, angle = 20$) :

plot3d("$x^3 + y^3$", $anaglyph$, $contour, angle = 116$) :

aspect(1.5); plot3d("$x^3 + sin(y)$", $-5, 5, 0, 2pi$) :

aspect(1.5); plot3d("$x^3 - sin(2y)$", $-5, 5, 0, pi$) :

load povray;

defaultpovray="C:
Program Files
POV-Ray"

C:Files-Ray

povstart(center=[0,0,1],zoom=3.6); ... t=linspace(0,2,1000); r=exp(-t); ... x=cos(2pi20t)r; y=sin(2pi10t)r; z=t; ... writeln(povgrid(x,y,z,povlook(red))); ... writeAxis(0,2,axis=4); ... povend();

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

povstart(center=[0,0,1],zoom=3); ... t=linspace(0,2,1000); r=exp(-t); ... x=cos(pi10t)r; y=sin(pi10t)r; z=t; ... writeln(povgrid(x,y,z,povlook(yellow))); ... writeAxis(0,2,axis=4); ... povend();

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

povstart(zoom=3); ... x=-1:0.5:1; r=1-(x+1)^3/6; ... $t = (0 : 30 : 360)'; y = rcos(t); z = rsin(t); ...writeln(povgrid(x, y, z, d = 0.02, dballs = 0.05)); ...povend();$

Command was not allowed! exec: return $_e xec(program, param, dir, print, hidden, wait); povray : exec(program, params, defaulthome); Try"traceerrors"toinspectlocalvariablesaftererrors.povend : povray(file, w, h, aspect, exit);$

Lebih Banyak Contoh

Anda dapat menemukan beberapa contoh lagi untuk Povray di Euler di file berikut.

¡a href="Examples/Dandelin Spheres.html"¿Examples/Dandelin Spheres¡/a¿

¡a href="Examples/Donat Math.html"¿Examples/Donat Math¡/a¿

¡a href="Examples/Trefoil Knot.html"¿Examples/Trefoil Knot¡/a¿

¡a href="Examples/Optimization by Affine Scaling.html"¿Examples/Optimization by Affine Scaling¡/a¿

5 Kalkulus

$23030630074_R omadhonaEnggalWilujeng_E MTKalkulusNama : RomadhonaEnggalWilujeng$

NIM : 23030630074

Kelas: Matematika E 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$_f ungsi := rumusfungsi(untukfungsi*numerik)$,

* Menggunakan format nama$_f ungsiamp; = rumusfungsi(untukfungsi * simbolik, namundapatdihitungsecaranumerik)$,

* Menggunakan format nama$_f ungsiamp; amp; = rumusfungsi(untukfungsi* simbolikmurni, tidakdapatdihitunglangsung)$,

* Fungsi sebagai program EMT.

Setiap format harus diawali dengan perintah function (bukan sebagai ekspresi).

Berikut adalah adalah beberapa contoh cara mendefinisikan fungsi.

function f(x) := 2x$^2 + exp(sin(x))//fungsinumerik$

f(0), f(1), f(pi)
1 4.31977682472 20.7392088022
function g(x) := sqrt($x^2 - 3x$)/$(x + 1)$
g(3)
0
g(0)
0
g(1)
Floating point error! Error in sqrt Try "trace errors" to inspect local variables after errors. g: useglobal; return sqrt($x^2 - 3 * x$)/$(x + 1) Error in : g(1)$...
f(g(5)) // komposisi fungsi
2.20920171961
g(f(5))
0.950898070639
f(0:10) // nilai-nilai 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]
Misalkan kita akan mendefinisikan fungsi
Fungsi tersebut tidak dapat didefinisikan sebagai fungsi numerik secara "inline" menggunakan format :=, melainkan didefinisikan sebagai program. Perhatikan, kata "map" digunakan agar fungsi dapat menerima vektor sebagai input, dan hasilnya berupa vektor. Jika tanpa kata "map" fungsinya hanya dapat menerima input satu nilai.
function map f(x) ...
if x¿0 then return $x^3 else return x^2 endif; endfunction < /pre > f(1)$
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):
function f(x) = $2E^x //fungsi simbolik$
x 2 E
$f(a) //nilai fungsi secara simbolik$
f(E) // nilai fungsi berupa bilangan desimal
30.308524483
$f(E)$,float(
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):
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 beberapa nilainya, baik untuk satu nilai maupun vektor. Gambar grafik tersebut.

Juga, carilah fungsi beberapa (dua) variabel. Lakukan hal sama seperti di atas.

Jawab:

1. Fungsi 1

function k(x) := x($x^5 + 3$)^3

k(3), k(5), k(7)

44660808 153027765760 3.3250729687e+13

kmap(-3:3)

[4.1472e+07, 48778, -8, 0, 64, 85750, 4.46608e+07]

plot2d("k(x)"):

2. Fungsi 2

function m(x) := (x)^4/($3 - x^2$)

m(2), m(-2), m(1)

-16 -16 0.5

mmap(-5:-5)

-28.4090909091

plot2d("m(x)"):

3. Fungsi 3

function n(x) := 3x/(x+5)+2

n(2), n(-1), n(-3), n(4)

2.85714285714 1.25 -2.5 3.33333333333

nmap(2:5)

[2.85714, 3.125, 3.33333, 3.5]

plot2d("n(x)"):

4. Fungsi 4

function l(x) := 3x^3/($x^4 - 3$)

l(5), l(4), l(3)

0.602893890675 0.758893280632 1.03846153846

lmap(5:8)

[0.602894, 0.50116, 0.429108, 0.375275]

plot2d("l(x)",-3,3,-600,600):

5. Fungsi 5

function j(x) := (cos(x))sin(2x)

j(pi), j(0), j(pi/3)

0 0 0.433012701892

jmap(0:3pi)

[0, 0.491295, 0.314941, 0.276619, -0.646688, -0.154318, -0.515201, 0.746821, 0.0418899, 0.684247]

plot2d("j(x)"):

6. Fungsi 6

function o(x) := xsqrt(x+2)

o(3), o(5), o(7)
6.7082039325 13.2287565553 21
omap(3:12)
[6.7082, 9.79796, 13.2288, 16.9706, 21, 25.2982, 29.8496, 34.641, 39.6611, 44.8999]
plot2d("o(x)"):
1. Fungsi 1
function a(x,y) ...
return $x^2 + y^2 - 24 endfunction < /pre > a(2,1), a(5,4), a(2,4)$
-19 17 -4
amap(-2:2,3:3)
[-11, -14, -15, -14, -11]
aspect=1.5; plot3d("a(x,y)",a=-100,b=100,c=-80,d=80,angle=35°,height=30°,r=pi,n=100):
2. Fungsi 2
function q(x,y) ...
return $y^2/(x^2/3) endfunction < /pre > q(4,2), q(2,3), q(4,3)$
0.75 6.75 1.6875
qmap(2:2,-2:2)
[3, 0.75, 0, 0.75, 3]
aspect=1.5; plot3d("q(x,y)",a=-100,b=100,c=-80,d=80,angle=35°,height=30°,r=pi,n=100):
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.

$showev('limit(sqrt(x^2 - 3x)/(x + 1), x, inf))$

$limit((x^3 - 13x^2 + 51x - 63)/(x^3 - 4x^2 - 3x + 18), x, 3)$

maxima: 'limit((x^3 - 13 * x^2 + 51 * x - 63)/(x^3 - 4 * x^2 - 3 * x + 18), x, 3) = $limit((x^3 - 13 * x^2 + 51 * x - 63)/(x^3 - 4 * x^2 - 3 * x + 18), x, 3)$

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", $add)$:

$limit(2xsin(x)/(1 - cos(x)), x, 0)$

maxima: 'limit(2*x*sin(x)/(1-cos(x)),x,0)=limit(2*x*sin(x)/(1-cos(x)),x,0)

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):

$limit(cot(7h)/cot(5h), h, 0)$

maxima: showev('limit(cot(7*h)/cot(5*h),h,0))

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):

$showev('limit(((x/8)^(1/3) - 1)/(x - 8), x, 8))$

aspect(1.5); plot2d("(((x/8)^(1/3)−1)/(x−8))", 0, 8); $plot2d(0, 1/24, points, style =$ "ow", $add)$:

$showev('limit(1/(2x - 1), x, 0))$

$showev('limit((x^2 - 3x - 10)/(x - 5), x, 5))$

$showev('limit(sqrt(x^2 + x) - x, x, inf))$

$showev('limit(abs(x - 1)/(x - 1), x, 1, minus))$

$showev('limit(sin(x)/x, x, 0))$

plot2d("sin(x)/x",-pi,pi); plot2d(0,1, points,style="ow", add):

$showev('limit(sin(x^3)/x, x, 0))$

$showev('limit(log(x), x, minf))$

$showev('limit((-2)^x, x, inf))$

$showev('limit(t - sqrt(2 - t), t, 2, minus))$

$showev('limit(t - sqrt(2 - t), t, 2, plus))$

$showev('limit(t - sqrt(2 - t), t, 5, plus))//Perhatikanhasilnya$

plot2d("x-sqrt(2-x)",0,2):

$showev('limit((x^2 - 9)/(2x^2 - 5x - 3), x, 3))$

$showev('limit((1 - cos(x))/x, x, 0))$

$showev('limit((x^2 + abs(x))/(x^2 - abs(x)), x, 0))$

$showev('limit((1 + 1/x)^x, x, inf))$

plot2d("(1+1/x)^x", 0, 1000) :

$showev('limit((1 + k/x)^x, x, inf))$

$showev('limit((1 + x)^(1/x), x, 0))$

$showev('limit((x/(x + k))^x, x, inf))$

$showev('limit((E^x - E^2)/(x - 2), x, 2))$

$showev('limit(sin(1/x), x, 0))$

$showev('limit(sin(1/x), x, inf))$

plot2d("sin(1/x)",-0.001,0.001):

Latihan

Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, hitung nilai limit fungsi tersebut di beberapa nilai dan di tak hingga. Gambar grafik fungsi tersebut untuk mengkonfirmasi nilai-nilai limit tersebut.

Jawab:

1. Fungsi 1

$showev('limit((3x - 6)/(x + 2), x, 2))$

plot2d("(3x-6)/(x+2)",-2,3.5,-1,5):

2. Fungsi 2

$showev('limit(cos(2x)/(sin(x) - cos(x)), x, 0))$

plot2d("cos(2x)/(sin(x) - cos (x))",-1,1):

3. Fungsi 3

$showev('limit(((2x^2 - 2x + 5)/(3x^2 + x - 6)), x, 3))$

plot2d("(2x^2 − 2x + 5)/(3x^2 + x − 6)", −2, 10, −10, 5) :

4. Fungsi 4

$showev('limit((4x^2-3),x,0))$
plot2d("$(4x^2-3)$") :
5. Fungsi 5
$showev('limit((x^{(}x^{(}x))),x,0,plus))$
plot2d("$(x^{(}x^{(}x)))$",$-3,3,-1,7$) :
6. Fungsi 6
$showev('limit((3xtan(x))/(1-cos(4x)),x,0))$
plot2d("(3xtan(x))/(1-cos(4x))",-pi/2,2pi,0,2pi):
Turunan Fungsi
Definisi turunan:
Berikut adalah contoh-contoh menentukan turunan fungsi dengan menggunakan definisi turunan (limit).
$showev('limit(((x+h)^2-x^2)/h,h,0))//turunanx^2$
p = expand((x+h)$^2 - x^2$)|$simplify$;p //pembilang dijabarkan dan disederhanakan
q =ratsimp(p/h); $q//ekspresiyangakandihitunglimitnyadisederhanakan$
$limit(q,h,0)//nilailimitsebagaiturunan$
$showev('limit(((x+h)^n-x^n)/h,h,0))//turunanx^n$
Petunjuk: ekspansikan (x+h)^n $denganmenggunakanteoremabinomial.$
$showev('limit((sin(x+h)-sin(x))/h,h,0))//turunansin(x)$
Petunjuk: ekspansikan sin(x+h) dengan menggunakan rumus jumlah dua sudut.
$showev('limit((log(x+h)-log(x))/h,h,0))//turunanlog(x)$
$showev('limit((1/(x+h)-1/x)/h,h,0))//turunan1/x$
$showev('limit((E^{(}x+h)-E^x)/h,h,0))//turunanf(x)=e^x$
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: $showev('limit((E^{(}x+h)-E^x)/h,h,0))//turunanf(x)=e^x$...
$showev('limit((E^h-1)/h,h,0))$
$showev('factor(E^{(}x+h)-E^x))$
$showev('limit(factor((E^{(}x+h)-E^x)/h),h,0))//turunanf(x)=e^x$
function f(x) = x^x
x x
$showev('limit(f(x),x,0))$
plot2d("x^x") :
$showev('limit((f(x+h)-f(x))/h,h,0))//turunanf(x)=x^x$
Di sini Maxima juga bermasalah terkait limit:
Dalam hal ini diperlukan asumsi nilai x.
assume(x 0); $showev('limit((f(x+h)-f(x))/h,h,0))//turunanf(x)=x^x$
forget(x 0) // jangan lupa, lupakan asumsi untuk kembali ke semula
[x gt; 0]
forget(x¡0)
[x lt; 0]

facts()
[]
$showev('limit((asin(x+h)-asin(x))/h,h,0))//turunanarcsin(x)$
Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.
$showev('limit((tan(x+h)-tan(x))/h,h,0))//turunantan(x)$
Mengapa hasilnya seperti itu? Tuliskan atau tunjukkan bahwa hasil limit tersebut benar, sehingga benar turunan fungsinya benar. Tulis penjelasan Anda di komentar ini.
Jadi, terbukti benar bahwa
function f(x) = sinh(x) // definisikan f(x)=sinh(x)
sinh(x)
function df(x) = limit((f(x+h)-f(x))/h,h,0); $df(x)//df(x) = f'(x)$
Hasilnya adalah cosh(x), karena
plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]):
function f(x) = $\sin(3x^5+7)^2$
2 5 sin (3 x + 7)
diff(f,3), diffc(f,3)
1198.32948904 1198.72863721
Apakah perbedaan diff dan diffc?
diff untuk menghitung turunan numerik ke-n dari fungsi f menggunakan metode perbedaan terbatas (finite difference). Sedangkan diffc mungkin merujuk pada metode yang lebih canggih atau terkompensasi untuk menghitung turunan numerik ke-n dari fungsi f.
$showev('diff(f(x),x))$

$float($
plot2d(f,0,3.1):
function f(x) =5cos(2x)-2xsin(2x) // mendifinisikan 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)
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 turunannya
Perhatikan titik-titik "puncak" grafik y=f(x) dan nilai turunan pada saat grafik fungsinya mencapai titik "puncak" tersebut.
Latihan
Bukalah buku Kalkulus. Cari dan pilih beberapa (paling sedikit 5 fungsi berbeda tipe/bentuk/jenis) fungsi dari buku tersebut, kemudian definisikan di

EMT pada baris-baris perintah berikut (jika perlu tambahkan lagi). Untuk setiap fungsi, tentukan turunannya dengan menggunakan definisi turunan (limit), seperti contoh-contoh tersebut. Gambar grafik fungsi asli dan fungsi turunannya pada sumbu koordinat yang sama.

Jawab:

1. Fungsi 1

function f(x) := x^2

$showev('limit((((x+h)^2-x^2)/h),h,0))//turunan x^2$

function df(x) = limit((((x+h)^2 − x^2)/h),h,0);df(x)// df(x) = f'(x)

plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",2,0.6), label("df(x)",2,0.17):

2. Fungsi 2

function f(x) := sin(x)cos(x)

$showev('limit(((sin(x+h)cos(x+h))-sin(x)cos(x))/h,h,0))//turunan sin(x) \times cos(x)$

function df(x) = limit(((sin(x+h)cos(x+h))-sin(x)cos(x))/h,h,0); $df(x)//df(x) = f'(x)$

plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",1,0), label("df(x)",2.3,1.2):

3. Fungsi 3

function f(x) := sqrt(x)4

$showev('limit((sqrt(x+h)4-sqrt(x)4)/h,h,0))//turunan sqrt(x)4$

function df(x) = limit((sqrt(x+h)4-sqrt(x)4)/h,h,0); $df(x)//df(x) = f'(x)$

plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",-2,11), label("df(x)",-2,-10):

4. Fungsi 4

function f(x) := cos(1/x)

$showev('limit((cos(1/(x+h))-cos(1/x))/h,h,0))//turunan cos(1/x)$

function df(x) = limit((cos(1/(x+h))-cos(1/x))/h,h,0); $df(x)//df(x) = f'(x)$

plot2d(["f(x)","df(x)"],-pi,pi,color=[blue,red]), label("f(x)",2,0.4), label("df(x)",1,-0.5):

5. Fungsi 5

function f(x) := (log(x))^5

$showev('limit(((log(x+h))^5-(log(x))^5)/h,h,0))//turunan (log(x))^5$

function df(x) = limit(((log(x+h))^5−$(log(x))^5$)/h,h,0);df(x)// df(x) = f'(x)

plot2d(["f(x)","df(x)"],-50,100,-10,50,color=[blue,red]), label("f(x)",25,35), label("df(x)",50,1):

6. Fungsi 6

function f(x) := sqrt(tan(x))

$showev('limit((sqrt(tan(x+h))-sqrt(tan(x)))/h,h,0))//turunan exp(x) \times cos(x)$

function df(x) = limit((sqrt(tan(x+h))-sqrt(tan(x)))/h,h,0); $df(x)//df(x) = f'(x)$

plot2d(["f(x)","df(x)"],-10,10,-10,10,color=[blue,red]), label("f(x)",4.5,0), label("df(x)",5.5,5):

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:

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"

$showev('integrate(1/(1+x), x))$

$showev('integrate(1/(1+x^2), x))$

$showev('integrate(1/sqrt(1-x^2), x))$

$showev('integrate(sin(x), x, 0, pi))$

plot2d("sin(x)",0,2pi):

$showev('integrate(sin(x), x, a, b))$

$showev('integrate(x^n, x, a, b))$

Answering "Is n positive, negative or zero?" with "positive"

$showev('integrate(x^2 sqrt(2x+1), x))$

$showev('integrate(x^2 sqrt(2x+1), x, 0, 2))$

$ratsimp($

$showev('integrate((sin(sqrt(x)+a)E^s qrt(x))/sqrt(x), x, 0, pi^2))$

$factor($

function map f(x) = E^(− x^2)

2 - x E

$showev('integrate(f(x), x))$

Fungsi f tidak memiliki antiturunan, integralnya masih memuat integral lain.

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:

maxima: 'integrate(f(x),x,0,pi)

x=0:0.1:pi-0.1; plot2d(x,f(x+0.1), bar); plot2d("f(x)",0,pi, add):

Integral tentu

maxima: 'integrate(f(x),x,0,pi)

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:

maxima: 'integrate(f(x),x,0,pi) = 0.1*sum(fx[i],i,1,length(fx))

Jumlah tersebut diperoleh dari hasil kali lebar sub-subinterval (=0.1) dan jumlah nilai-nilai f(x) untuk x = 0.1, 0.2, 0.3, ..., 3.2.

0.1sum(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.

$showev('integrate(f(x), x, 0, inf))$

Berikut adalah contoh lain fungsi yang tidak memiliki antiderivatif, sehingga integral tentunya hanya dapat dihitung dengan metode numerik.

function f(x) = x^x;f(x)

$showev('integrate(f(x), x, 0, 1))$

x=0:0.1:1-0.01; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,1, add):

Maxima gagal menghitung integral tentu tersebut secara langsung menggunakan perintah integrate. Berikut kita lakukan seperti contoh sebelumnya untuk mendapat hasil atau pendekatan nilai integral tentu tersebut.

t = makelist(a,a,0,1-0.01,0.01);

fx = makelist(f(t[i]+0.01),i,1,length(t));

maxima: 'integrate(f(x),x,0,1) = 0.01*sum(fx[i],i,1,length(fx))

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$_i$ncomplete$(-, 6I)$ + 6gamma$_i$ncomplete$(-, -6I)$)554/51$sin(14)$+(6Igamma$_i$ncomplete$(-, 6I)$54/51$pi-$ 6Igamma$_i$ncomplete$(-, -6I)$)$cos(14)$)$sin(--) - 60)/120510$

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$_{incomplete(0.2,6.0I)-4.192962712629476Igam}$

$showev('integrate(xexp(-x), x, 0, 1))//Integraltentu(eksak)$

Aplikasi Integral Tentu

plot2d("$x^3 - x$", −0.1, 1.1); $plot2d($" $- x^2$", $add)$; ... b = $solve($"$x^3 - x + x^2$", 0.5); $x = linspace(0, b, 200)$; $xi = flipx(x)$; ...$plot2d(x|xi, x^3-x|-xi^2, filled, style =$ "|", $fillcolor = 1$, $add) : //Plotdaerahantara2kurva$

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

a = solve((-x^2)−(x^3−x), x);a // menentukan absis titik potong kedua kurva secara eksak

$showev('integrate(-x^2-x^3+x, x, 0, (sqrt(5)-1)/2))//Nilaiintegralsecaraeksak$

$float($

Panjang Kurva

Hitunglah panjang kurva berikut ini dan luas daerah di dalam kurva tersebut.

dengan

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

// Kita gambar kurvanya terlebih dahulu

function r(t) = 1+sin(3t)/2; $'r(t) = r(t)$

function fx(t) = r(t)cos(t); $'fx(t) = fx(t)$

function fy(t) = r(t)sin(t); $'fy(t) = fy(t)$

function ds(t) = trigreduce(radcan(sqrt(diff(fx(t),t)^2+$diff(fy(t),t)^2$)));'ds(t)=ds(t)

Maxima said: diff: second argument must be a variable; found errexp1 – an error. To debug this try: debugmode(true);

Error in: ... e(radcan(sqrt(diff(fx(t),t)^2 + $diff(fy(t),t)^2$)));'ds(t)=ds(t ...

$integrate(ds(x), x, 0, 2pi)//panjang(keliling)kurva$

Maxima gagal melakukan perhitungan eksak integral tersebut.

Berikut kita hitung integralnya secara umerik dengan perintah EMT.

integrate("ds(x)",0,2pi)

Function ds not found. Try list ... to find functions! Error in expression: ds(x) return expr(x,args()); Error in map. if maps then return gauss: if maps then y=adaptivegauss: t1=gauss(f, $c, c+h; args(), = maps); Try"traceerrors"toinspectlocalvariablesaftererr$
$returnadaptivegauss(f$,a,b,eps*1000;args(),=maps);

Spiral Logaritmik

a=0.1; plot2d("exp(ax)cos(x)","exp(ax)sin(x)",r=2,xmin=0,xmax=2pi):

kill(a) // hapus expresi a

done

function fx(t) = exp(at)cos(t); $'fx(t) = fx(t)$

function fy(t) = exp(at)sin(t); $'fy(t) = fy(t)$

function df(t) = trigreduce(radcan(sqrt(diff(fx(t),t)^2+$diff(fy(t),t)^2$)));'df(t)=df(t)

Maxima said: diff: second argument must be a variable; found errexp1 – an error. To debug this try: debugmode(true);

Error in: ... e(radcan(sqrt(diff(fx(t),t)^2 + $diff(fy(t),t)^2$)));'df(t)=df(t ...

S =integrate(df(t),t,0,2

Maxima said: expt: undefined: 0 to a negative exponent. 0: df(x=[0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0. – 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", $xmin = -1, xmax = 1$) :

$showev('integrate(sqrt(1 + diff(x^2, x)^2), x, -1, 1))$

$float($

x=-1:0.2:1; y=x^2; $plot2d(x, y); ... plot2d(x, y, points = 1, style = "o", add = 1)$:

Panjang tersebut dapat dihampiri dengan menggunakan jumlah panjang ruas-ruas garis yang menghubungkan titik-titik pada parabola tersebut.

i=1:cols(x)-1; sum(sqrt((x[i+1]-x[i])^2 + $(y[i + 1] - y[i])^2$))

2.95191957027

Hasilnya mendekati panjang yang dihitung secara eksak. Untuk mendapatkan hampiran yang cukup akurat, jarak antar titik dapat diperkecil, misalnya 0.1, 0.05, 0.01, dan seterusnya. Cobalah Anda ulangi perhitungannya dengan nilai-nilai tersebut.

Koordinat Kartesius

Berikut diberikan contoh perhitungan panjang kurva menggunakan koordinat Kartesius. Kita akan hitung panjang kurva dengan persamaan implisit:

z = x^3 + $y^3 - 3xy$;z

plot2d(z,r=2,level=0,n=100):

Kita tertarik pada kurva di kuadran pertama.

plot2d(z,a=0,b=2,c=0,d=2,level=[-10;0],n=100,contourwidth=3,style="/"):

Kita selesaikan persamaannya untuk x.

$zwithy = lx, sol = solve($

Kita gunakan solusi tersebut untuk mendefinisikan fungsi dengan Maxima.

function f(l) = rhs(sol[1]); $'f(l) = f(l)$

Fungsi tersebut juga dapat digunaka untuk menggambar kurvanya. Ingat, bahwa fungsi tersebut adalah nilai x dan nilai y=l*x, yakni x=f(l) dan y=l*f(l).

plot2d(f(x),xf(x),xmin=-0.5,xmax=2,a=0,b=2,c=0,d=2,r=1.5):

Elemen panjang kurva adalah:

function ds(l) = ratsimp(sqrt(diff(f(l),l)^2 + $diff(lf(l), l)^2$));'ds(l)=ds(l)

$integrate(ds(l), l, 0, 1)$

Integral tersebut tidak dapat dihitung secara eksak menggunakan Maxima. Kita hitung integral etrsebut secara numerik dengan Euler. Karena kurva simetris, kita hitung untuk nilai variabel integrasi dari 0 sampai 1, kemudian hasilnya dikalikan 2.

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) ...

endfunction ¡/pre¿ panjangkurva("x","x^2", $-1, 1)//cekuntukmenghitungpanjangkurvaparabolasebelumn$

Bandingkan dengan nilai eksak di atas.

2panjangkurva(mxm("f(x)"),mxm("xf(x)"),0,1)

Wrong argument. Cannot use a string here.

Error in: 2*panjangkurva(mxm("f(x)"),mxm("x*f(x)"),0,1) ...

Kita hitung panjang spiral Archimides berikut ini dengan fungsi tersebut.

plot2d("xcos(x)","xsin(x)",xmin=0,xmax=2pi,square=1):

panjangkurva("xcos(x)","xsin(x)",0,2pi)

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)); $sol//Kitagunakanuntukmenghitungpanjangkurvaterakhirdiatas$

$sol|trigreduce|expand$,integrate(

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) :

sol = radcan(ds(3x$^2 - 1, 3x^3 - 1$));sol

$showev('integrate(sol, x, 0, 1/sqrt(3)))$,2float(

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:

Misalkan posisi titik pada lingkaran tersebut mula-mula (0,0) dan posisinya pada saat t adalah:

Berikut kita plot lintasan tersebut dan beberapa posisi lingkaran ketika t=0, t=pi/2, t=r*pi.

x = r(t-sin(t))

[0, 1.66665833335744e-7 r, 1.33330666692022e-6 r, 4.499797504338432e-6 r, 1.066581336583994e-5 r, 2.083072932167196e-5 r, 3.599352055540239e-5 r, 5.71526624672386e-5 r, 8.530603082730626e-5 r, 1.214508019889565e-4 r, 1.665833531718508e-4 r, 2.216991628251896e-4 r, 2.877927110806339e-4 r, 3.658573803051457e-4 r, 4.568853557635201e-4 r, 5.618675264007778e-4 r, 6.817933857540259e-4 r, 8.176509330039827e-4 r, 9.704265741758145e-4 r, 0.001141105023499428 r, 0.001330669204938795 r, 0.001540100153900437 r, 0.001770376919130678 r, 0.002022476464811601 r, 0.002297373572865413 r, 0.002596040745477063 r, 0.002919448107844891 r, 0.003268563311168871 r, 0.003644351435886262 r, 0.004047774895164447 r, 0.004479793338660443 r, 0.0049413635565565 r, 0.005433439383882244 r, 0.005956971605131645 r, 0.006512907859185624 r, 0.007102192544548636 r, 0.007725766724910044 r, 0.00838456803503801 r, 0.009079530587017326 r, 0.009811584876838586 r, 0.0105816576913495 r, 0.01139067201557714 r, 0.01223954694042984 r, 0.01312919757078923 r, 0.01406053493400045 r, 0.01503446588876983 r, 0.01605189303448024 r, 0.01711371462093175 r, 0.01822082445851714 r, 0.01937411182884202 r, 0.02057446139579705 r, 0.02182275311709253 r, 0.02311986215626333 r, 0.02446665879515308 r, 0.02586400834688696 r, 0.02731277106934082 r, 0.02881380207911666 r, 0.03036795126603076

r, 0.03197606320812652 r, 0.0336389770872163 r, 0.03535752660496472 r, 0.03713253989951881
r, 0.03896483946269502 r, 0.0408552420577305 r, 0.04280455863760801 r, 0.04481359426396048
r, 0.04688314802656623 r, 0.04901401296344043 r, 0.05120697598153157 r, 0.05346281777803219
r, 0.05578231276230905 r, 0.05816622897846346 r, 0.06061532802852698 r, 0.0631303649963022
r, 0.06571208837185505 r, 0.06836123997666599 r, 0.07107855488944881 r, 0.07386476137264342
r, 0.07672058079958999 r, 0.07964672758239233 r, 0.08264390910047736 r, 0.0857128256298576
r, 0.08885417027310427 r, 0.09206862889003742 r, 0.09535688002914089 r, 0.0987195948597075
r, 0.1021574371047232 r, 0.1056710629744951 r, 0.1092611211010309 r, 0.1129282524731764
r, 0.1166730903725168 r, 0.1204962603100498 r, 0.1243983799636342 r, 0.1283800591162231
r, 0.1324418995948859 r, 0.1365844952106265 r, 0.140808431699002 r, 0.1451142866615502
r, 0.1495026295080298 r, 0.1539740213994798 r]

y = r(1-cos(t))

[0, 4.999958333473664e-5 r, 1.999933334222437e-4 r, 4.499662510124569e-
4 r, 7.998933390220841e-4 r, 0.001249739605033717 r, 0.00179946006479581
r, 0.002448999746720415 r, 0.003198293697380561 r, 0.004047266988005727 r,
0.004995834721974179 r, 0.006043902043303184 r, 0.00719136414613375 r, 0.00843810628521191
r, 0.009784003787362772 r, 0.01122892206395776 r, 0.01277271662437307 r, 0.01441523309043924
r, 0.01615630721187855 r, 0.01799576488272969 r, 0.01993342215875837 r, 0.02196908527585173
r, 0.02410255066939448 r, 0.02633360499462523 r, 0.02866202514797045 r, 0.03108757828935527
r, 0.03361002186548678 r, 0.03622910363410947 r, 0.03894456168922911 r, 0.04175612448730281
r, 0.04466351087439402 r, 0.04766643011428662 r, 0.05076458191755917 r, 0.0539576564716131
r, 0.05724533447165381 r, 0.06062728715262111 r, 0.06410317632206519 r, 0.06767265439396564
r, 0.07133536442348987 r, 0.07509094014268702 r, 0.07893900599711501 r, 0.08287917718339499
r, 0.08691105968769186 r, 0.09103425032511492 r, 0.09524833678003664 r, 0.09955289764732322
r, 0.1039475024744748 r, 0.1084317118046711 r, 0.113005077220716 r, 0.1176671413898787
r, 0.1224174381096274 r, 0.1272554923542488 r, 0.1321808203223502 r, 0.1371929294852391
r, 0.1422913186361759 r, 0.1474754779404944 r, 0.152744888986584 r, 0.1580990248377314
r, 0.1635373500848132 r, 0.1690593208998367 r, 0.1746643850903219 r, 0.1803519821545206
r, 0.1861215433374662 r, 0.1919724916878484 r, 0.1979042421157076 r, 0.2039162014509444
r, 0.2100077685026351 r, 0.216178334119151 r, 0.2224272812490723 r, 0.2287539850028937
r, 0.2351578127155118 r, 0.2416381240094921 r, 0.2481942708591053 r, 0.2548255976551299
r, 0.2615314412704124 r, 0.2683111311261794 r, 0.2751639892590951 r, 0.2820893303890569
r, 0.2890864619877229 r, 0.2961546843477643 r, 0.3032932906528349 r, 0.3105015670482534
r, 0.3177787927123868 r, 0.3251242399287333 r, 0.3325371741586922 r, 0.3400168541150183
r, 0.3475625318359485 r, 0.3551734527599992 r, 0.3628488558014202 r, 0.3705879734263036
r, 0.3783900317293359 r, 0.3862542505111889 r, 0.3941798433565377 r, 0.4021660177127022
r, 0.4102119749689023 r, 0.418316910536117 r, 0.4264800139275439 r, 0.4347004688396462
r, 0.4429774532337832 r, 0.451310139418413 r]

Berikut kita gambar sikloid untuk r=1.

ex = x-sin(x); ey = 1-cos(x); aspect(1);

plot2d(ex,ey,xmin=0,xmax=4pi,square=1); ... plot2d("2+cos(x)","1+sin(x)",xmin=0,xmax=2pi, add,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),0.00114114105023499428*r-sin(0.00114114105023499428*r),0.00133066920493879
sin(0.00133066920493879
5*r),0.00154010015390043
7*r-sin(0.00154010015390043
7*r),0.00177037691913067
8*r-
sin(0.00177037691913067
8*r),0.00202247646481160
1*r-sin(0.00202247646481160
1*r),0.00229737357286541
3*r-
sin(0.00229737357286541
3*r),0.00259604074547706
3*r-sin(0.00259604074547706
3*r),0.00291944810784489
1*r-
sin(0.00291944810784489
1*r),0.00326856331116887
1*r-sin(0.00326856331116887
1*r),0.00364435143588626
2*r-
sin(0.00364435143588626
2*r),0.00404777489516444
7*r-sin(0.00404777489516444
7*r),0.00447979333866044
3*r-
sin(0.00447979333866044
3*r),0.0049413635565565
5*r-sin(0.0049413635565565
5*r),0.00543343938388224
4*r-
sin(0.00543343938388224
4*r),0.00595697160513164
5*r-sin(0.00595697160513164
5*r),0.00651290785918562
4*r-
sin(0.00651290785918562
4*r),0.00710219254454863
6*r-sin(0.00710219254454863
6*r),0.00772576672491004
4*r-
sin(0.00772576672491004
4*r),0.00838456803503801
8*r-sin(0.00838456803503801
8*r),0.00907953058701732
6*r-
sin(0.00907953058701732
6*r),0.00981158487683858
6*r-sin(0.00981158487683858
6*r),0.0105816576913495
8*r-
sin(0.0105816576913495
8*r),0.01139067201557714
8*r-sin(0.01139067201557714
8*r),0.01223954694042984
8*r-
sin(0.01223954694042984
8*r),0.01312919757078923
8*r-sin(0.01312919757078923
8*r),0.01406053493400045
8*r-
sin(0.01406053493400045
8*r),0.01503446588876983
8*r-sin(0.01503446588876983
8*r),0.01605189303448024
8*r-
sin(0.01605189303448024
8*r),0.01711371462093175
8*r-sin(0.01711371462093175
8*r),0.01822082445851714
8*r-
sin(0.01822082445851714
8*r),0.01937411182884202
8*r-sin(0.01937411182884202
8*r),0.02057446139579705
8*r-
sin(0.02057446139579705
8*r),0.02182275311709253
8*r-sin(0.02182275311709253
8*r),0.02311986215626333
8*r-
sin(0.02311986215626333
8*r),0.02446665879515308
8*r-sin(0.02446665879515308
8*r),0.02586400834688696
8*r-
sin(0.02586400834688696
8*r),0.02731277106934082
8*r-sin(0.02731277106934082
8*r),0.02881380207911666
8*r-
sin(0.02881380207911666
8*r),0.03036795126603076
8*r-sin(0.03036795126603076
8*r),0.03197606320812652
8*r-
sin(0.03197606320812652
8*r),0.03363897708721638*r-sin(0.03363897708721638*r),0.03535752660496472
8*r-
sin(0.03535752660496472
8*r),0.03713253989951881
8*r-sin(0.03713253989951881
8*r),0.03896483946269502
8*r-
sin(0.03896483946269502
8*r),0.04085552420577305
8*r-sin(0.04085552420577305
8*r),0.04280455863760801
8*r-
sin(0.04280455863760801
8*r),0.04481359426396048
8*r-sin(0.04481359426396048
8*r),0.04688314802656623
8*r-
sin(0.04688314802656623
8*r),0.04901401296344043
8*r-sin(0.04901401296344043
8*r),0.05120697598153157
8*r-
sin(0.05120697598153157
8*r),0.05346281777803219
8*r-sin(0.05346281777803219
8*r),0.05578231276230905
8*r-
sin(0.05578231276230905
8*r),0.05816622897846346
8*r-sin(0.05816622897846346
8*r),0.06061532802852698
8*r-
sin(0.06061532802852698
8*r),0.06313036499630228*r-sin(0.06313036499630228*r),0.06571208837185505
8*r-
sin(0.06571208837185505
8*r),0.06836123997666599
8*r-sin(0.06836123997666599
8*r),0.07107855488944881
8*r-
sin(0.07107855488944881
8*r),0.07386476137264342
8*r-sin(0.07386476137264342
8*r),0.07672058079958999
8*r-
sin(0.07672058079958999
8*r),0.07964672758239233
8*r-sin(0.07964672758239233
8*r),0.08264390910047736
8*r-
sin(0.08264390910047736
8*r),0.08571282562985768*r-sin(0.08571282562985768*r),0.08885417027310427
8*r-
sin(0.08885417027310427
8*r),0.09206862889003742
8*r-sin(0.09206862889003742
8*r),0.09535688002914089
8*r-
sin(0.09535688002914089
8*r),0.09871959485970758*r-sin(0.09871959485970758*r),0.10215743710472328*r-
sin(0.10215743710472328*r),0.10567106297449518*r-sin(0.10567106297449518*r),0.10926112110103098*r-
sin(0.10926112110103098*r),0.11292825247317648*r-sin(0.11292825247317648*r),0.11667309037251688*r-
sin(0.11667309037251688*r),0.12049626031004988*r-sin(0.12049626031004988*r),0.12439837996363428*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|"$doesnotproducearealorcolumnvector$");$Try$"$traceerrors$"$toinspectlocalvariablesaft$
$dw/n, dw/n^2, dw/n; args()$);

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,[0,1.66665833335744e-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 ...

Kurvatur (Kelengkungan) Kurva

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:

dengan R adalah radius kelengkungan. (Setiap lingkaran memiliki kelengkungan ini pada setiap titiknya, dapat diartikan, setiap lingkaran berputar 2pi sejauh 2piR.)

Definisi ini sulit dimanipulasi dan dinyatakan ke dalam rumus untuk kurva umum. Oleh karena itu digunakan definisi lain yang ekivalen.

Definisi Kurvatur dengan Fungsi Parametrik Panjang Kurva

Setiap kurva diferensiabel dapat dinyatakan dengan persamaan parametrik terhadap panjang kurva s:

dengan x dan y adalah fungsi riil yang diferensiabel, yang memenuhi:

Ini berarti bahwa vektor singgung

memiliki norm 1 dan merupakan vektor singgung satuan.

Apabila kurvanya memiliki turunan kedua, artinya turunan kedua x dan y ada, maka T'(s) ada. Vektor ini merupakan normal kurva yang arahnya menuju pusat kurvatur, norm-nya merupakan nilai kurvatur (kelengkungan):

Nilai

disebut jari-jari (radius) kelengkungan kurva.

Bilangan riil

disebut nilai kelengkungan bertanda.

Contoh:

Akan ditentukan kurvatur lingkaran

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 errexp1 – an error. To debug this try: debugmode(true);

Error in: ... =integrate(sqrt(diff(fx,t)^2 + $diff(fy,t)^2$), $t, 0, t$); $s//elemen...$

kill(s); fx = rcos(s/r); fy =rsin(s/r); // definisi ulang persamaan parametrik terhadap s dengan substitusi t=s/r

k = trigsimp(sqrt(diff(fx,s,2)^2+$diff(fy, s, 2)^2$));k // nilai kurvatur lingkaran dengan menggunakan definisi di atas

Untuk representasi parametrik umum, misalkan

merupakan persamaan parametrik untuk kurva bidang yang terdiferensialkan dua kali. Kurvatur untuk kurva tersebut didefinisikan sebagai

Selanjutnya, pembilang pada persamaan di atas dapat dicari sebagai berikut.

Jadi, rumus kurvatur untuk kurva parametrik

adalah

Jika kurvanya dinyatakan dengan persamaan parametrik pada koordinat kutub

maka rumus kurvaturnya adalah

(Silakan Anda turunkan rumus tersebut!)

Contoh:

Lingkaran dengan pusat (0,0) dan jari-jari r dapat dinyatakan dengan persamaan parametrik

Nilai kelengkungan lingkaran tersebut adalah

Hasil cocok dengan definisi kurvatur suatu kelengkungan.

Kurva

dapat dinyatakan ke dalam persamaan parametrik

sehingga kurvaturnya adalah

Contoh:

Akan ditentukan kurvatur parabola

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) // kelengkungan parabola

function f(x) = $x^2 + x + 1$;y=f(x) // akan kita plot kelengkungan parabola untuk a=b=c=1

function k(x) = (diff(f(x),x,2))/(1+diff(f(x),x)^2)^(3/2);'k(x)=k(x) // kelengkungan parabola

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

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):

Untuk kurva yang dinyatakan dengan fungsi implisit

dengan turunan-turunan parsial

berlaku

sehingga kurvaturnya adalah

(Silakan Anda turunkan sendiri!)

Contoh 1:

Parabola

dapat dinyatakan ke dalam persamaan implisit

function F(x,y) =$ax^2 + bx + c - y$;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)

function k(x) = (Fy$^2 Fxx - 2FxFyFxy + Fx^2 Fyy)/(Fx^2 + Fy^2)^(3/2)$;'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 perintah berikut (jika perlu tambahkan lagi).

* Untuk setiap fungsi, tentukan anti turunannya (jika ada), hitunglah * integral tentu dengan batas-batas yang menarik (Anda tentukan * sendiri), seperti contoh-contoh tersebut.

* Lakukan hal yang sama untuk fungsi-fungsi yang tidak dapat * diintegralkan (cari sedikitnya 3 fungsi).

* Gambar grafik fungsi dan daerah integrasinya pada sumbu koordinat * yang sama.

* Gunakan integral tentu untuk mencari luas daerah yang dibatasi oleh * dua kurva yang berpotongan di dua titik. (Cari dan gambar kedua kurva * dan arsir (warnai) daerah yang dibatasi oleh keduanya.)

* Gunakan integral tentu untuk menghitung volume benda putar kurva y= * f(x) yang diputar mengelilingi sumbu x dari x=a sampai x=b, yakni

(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:

(Pilih fungsi dan gambar kurvanya.)

Jawab:

1. Fungsi 1

function f(x) = $5x^2$;f(x)

$showev('integrate(f(x),x))$

Maxima output too long! Error in: $showev('integrate(f(x),x))$...

$showev('integrate(f(x),x,2,3))$

Maxima said: defint: variable of integration must be a simple or subscripted variable. defint: found errexp1 0: showev(f='integrate([0,1.3888750000749e-13*r^2, $8.88853334026953e{-}12{*}r^2$, $1.012408879002519e{-}10{*}r^2$, 5.687978737...)−−$anerror.Todebugthistry : debugmode(true)$;

Error in: $showev('integrate(f(x),x,2,3))$...

x=0.01:0.03:4; plot2d(x,f(x+0.01), bar); plot2d("f(x)",2,3, add):

Error : f(x) does not produce a real or column vector

Error generated by error() command

error(f|"$doesnotproducearealorcolumnvector"); adaptiveevalone : s = Try"traceerrors"toinspectlocalvar$
$dw/n, dw/n^2, dw/n, auto; args());$

2. Fungsi 2

function f(x) = cos(2x+5); $f(x)$

$showev('integrate(f(x),x))$

$showev('integrate(f(x),x,pi,2pi))$

x=0:0.05:pi-0.1; plot2d(x,f(x+0.03), bar); plot2d("f(x)",pi,2pi, add):

3. Fungsi 3

function f(x) = $(\sin(x))(\cos((x)))^2$;f(x)

$showev('integrate(f(x),x))$

$showev('integrate(f(x),x,0,pi))$

x=-pi:0.04:pi; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,pi, add):

4. Fungsi 4

function f(x) = $(x^2(2-x^3)^(1/2))$;f(x)

$showev('integrate(f(x),x))$

$showev('integrate(f(x),x,0,1))$

x=-1:0.04:1; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,1, add):

5. Fungsi 5

function f(x) = sqrt(24-x^2);f(x)

$showev('integrate(f(x),x))$

$showev('integrate(f(x),x,1,2))$

x=-2:0.04:1; plot2d(x,f(x+0.01), bar); plot2d("f(x)",1,2, add):

6. Fungsi 6

t = makelist(a,a,0,1-0.01,0.01);

fx = makelist(f(t[i]+0.01),i,1,length(t));

function f(x) = x^2 + 50;f(x)

x=0:0.1:pi-0.01; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,pi, add):

0.01sum(f(x+0.01))

17.051552

7. Fungsi 7
t = makelist(a,a,0,1-0.01,0.01);
fx = makelist(f(t[i]+0.01),i,1,length(t));
function f(x) = cos(x)/x; $f(x)$
x=-pi:0.07:pi-0.01; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,pi, add):
0.01sum(f(x+0.01))
0.415163991256
8. Fungsi 8
t = makelist(a,a,0,1-0.01,0.01);
fx = makelist(f(t[i]+0.01),i,1,length(t));
function f(x) = sqrt($x^2 - 1$);f(x)
x=3:0.04:pi-0.01; plot2d(x,f(x+0.01), bar); plot2d("f(x)",0,2, add):
0.01sum(f(x+0.01))
0.11610107668
Luas daerah dibatasi 2 kurva
1). Fungsi 1
function f(x) = x^3;f(x)
function g(x) = x; $g(x)$
plot2d(["x^4","x^3"], $-2, 2, -1, 2$) :
function h(x) = f(x)-g(x); $h(x)$
$showev('integrate(h(x), x))$
$solve(f(x) = g(x))$
$showev('integrate(h(x), x, 0, 1))//menghitungluasdaerahyangdibatasi2kurva$
x=-1:0.01:1; plot2d(x,f(x), bar, filled,style="-",fillcolor=orange, grid); plot2d(x,g(x), bar, add, filled,style="
",fillcolor=white); label("f(x)",0,2.1); label("g(x)",0.5,0.3):
2). Fungsi 2
function f(x) = $x^3 + 1$;f(x)
function g(x) = x^2;g(x)
plot2d(["-x^2 + 2","x^2"], $-2, 2, -1, 2$) :
function h(x) = f(x)-g(x); $h(x)$
$solve(f(x) = g(x))$
$showev('integrate(h(x), x, -1, 1))//menghitungluasdaerahyangdibatasi2kurva$
x=-1:0.01:1; plot2d(x,f(x), bar, filled,style="-",fillcolor=orange, grid); plot2d(x,g(x), bar, add, filled,style="
",fillcolor=white); label("f(x)",0,2.1); label("g(x)",0.5,0.3):
Volume benda putar
Menghitung volume hasil perputaran kurva
dari x=-1 sampai x=0. Diputar terhadap sumbu-x.
Jawab:
function m(x) = $x^4 + 3$;m(x)
$showev('integrate(pi(m(x))^2, x, -1, 0))//Menghitungvolumehasilperputaranm(x)$
Daerah di bawah kurva yang akan dirotasi terhadap sumbu x sebagai berikut:
plot2d("m(x)",-1,0,-1,2,grid=7, filled, style="/
"):
Hasil perputaran m(x) terhadap sumbu x sebagai berikut:
plot3d("m(x)",-1,0,-1,1, rotate,angle=6.3, hue, contour,color=redgreen,height=11):
Menghitung panjang kurva

Menghitung panjang kurva
dari x=1 sampai x=3.
function d(x) = $x^2 - x + 1$;d(x)
plot2d("d(x)",-5,6): // gambar kurva d(x)
$showev('limit((d(x+h)-d(x))/h, h, 0))$
function dd(x) = limit((d(x+h)-d(x))/h,h,0); $dd(x)$
function q(x) = $((dd(x))^2)$;q(x)
$showev('integrate(sqrt(1+q(x)), x, 1, 3))//menghitungpanjangkurva$
Barisan dan Deret
(Catatan: bagian ini belum lengkap. Anda dapat membaca contoh-contoh pengguanaan EMT dan Maxima untuk menghitung limit barisan, rumus jumlah parsial suatu deret, jumlah tak hingga suatu deret konvergen, dan sebagainya. Anda dapat mengeksplor contoh-contoh di EMT atau perbagai panduan penggunaan Maxima di software Maxima atau dari Internet.)

Barisan dapat didefinisikan dengan beberapa cara di dalam EMT, di antaranya:

* dengan cara yang sama seperti mendefinisikan vektor dengan * elemen-elemen beraturan (menggunakan titik dua ":");
* menggunakan perintah "sequence" dan rumus barisan (suku ke -n);
* menggunakan perintah "iterate" atau "niterate";
* menggunakan fungsi Maxima "create$_list$" $atau$ "$makelist$" $untuk * menghasilkan barisan simbolik$;
* menggunakan fungsi biasa yang inputnya vektor atau barisan;
* menggunakan fungsi rekursif.

EMT menyediakan beberapa perintah (fungsi) terkait barisan, yakni:

* sum: menghitung jumlah semua elemen suatu barisan
* cumsum: jumlah kumulatif suatu barisan
* differences: selisih antar elemen-elemen berturutan

EMT juga dapat digunakan untuk menghitung jumlah deret berhingga maupun deret tak hingga, dengan menggunakan perintah (fungsi) "sum". Perhitungan dapat dilakukan secara numerik maupun simbolik dan eksak.

Berikut adalah beberapa contoh perhitungan barisan dan deret menggunakan EMT.

1:10 // barisan sederhana
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
1:2:30
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
sum(1:2:30), sum(1/(1:2:30))
225 2.33587263431
$'sum(k, k, 1, n) = factor(ev(sum(k, k, 1, n), simpsum = true))//simpsum : menghitungderetsecarasimbolik$
$'sum(1/(3^k+k), k, 0, inf) = factor(ev(sum(1/(3^k+k), k, 0, inf), simpsum = true))$
Di sini masih gagal, hasilnya tidak dihitung.
$'sum(1/x^2, x, 1, inf) = ev(sum(1/x^2, x, 1, inf), simpsum = true)//ev : menghitungnilaiekspresi$
$'sum((-1)^(k-1)/k, k, 1, inf) = factor(ev(sum((-1)^(x-1)/x, x, 1, inf), simpsum = true))$

Di sini masih gagal, hasilnya tidak dihitung.

$'sum((-1)^k/(2k-1), k, 1, inf) = factor(ev(sum((-1)^k/(2k-1), k, 1, inf), simpsum = true))$

$ev(sum(1/n!, n, 0, inf), simpsum = true)$

Iterasi dan Barisan

EMT menyediakan fungsi iterate("g(x)", x0, n) untuk melakukan iterasi

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 6bulan dipotong pajak 20tabungan 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],...,x[n-1] yang diketahui.

Berikut adalah contoh barisan Fibonacci.

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 $2^(n-2), untuk n = 2, 4, 5, ...$

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

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 $\log_2(n)$.

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:

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 di hubungan 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 memmerlukan indeks dan bilangan kompleks. Dalam hal ini diigunakan vektor kolom pada bidang.

function gstep (v) ...

w=[-v[2];v[1]]; return v+w/norm(w); endfunction ¡/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

Iterasi ini juga dapat dilakukan pada interval, hasilnya adalah barisan interval yang memuat akar tersebut.

hasil := iterate("cos(x)", 1,2) //iterasi x(n+1)=cos(x(n)), dengan interval awal (1, 2)

0.739085133211,0.7390851332133

Jika interval hasil tersebut sedikit diperlebar, akan terlihat bahwa interval tersebut memuat akar persamaan x=cos(x).

h=expand(hasil,100), cos(h) ¡¡ h

0.73908513309,0.73908513333 1

Iterasi juga dapat digunakan pada fungsi yang didefinisikan.

function f(x) := (x+2/x)/2

Iterasi x(n+1)=f(x(n)) akan konvergen ke akar kuadrat 2.

iterate("f",2), sqrt(2)

1.41421356237 1.41421356237

Jika pada perintah iterate diberikan tambahan parameter n, maka hasil iterasinya akan ditampilkan mulai dari iterasi pertama sampai ke-n.

iterate("f",2,5)

[2, 1.5, 1.41667, 1.41422, 1.41421, 1.41421]

Untuk iterasi ini tidak dapat dilakukan terhadap interval.

niterate("f", 1,2 ,5)

[1,2 , 1,2 , 1,2 , 1,2 , 1,2 , 1,2]

Perhatikan, hasil iterasinya sama dengan interval awal. Alasannya adalah perhitungan dengan interval bersifat terlalu longgar. Untuk meingkatkan perhitungan pada ekspresi dapat digunakan pembagian intervalnya, menggunakan fungsi ieval().

function s(x) := ieval("(x+2/x)/2",x,10)

Selanjutnya dapat dilakukan iterasi hingga diperoleh hasil optimal, dan intervalnya tidak semakin mengecil. Hasilnya berupa interval yang memuat akar persamaan:

Satu-satunya solusi adalah

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.

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

0.99999999999999778,1.00000000000000022 ... 4.99999999999999911,5.00000000000000089 ...

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 dipercepatdengan percepatan Steffenson:

steffenson("sqrt(x)",2,10)

[1.04888, 1.00028, 1, 1]

Iterasi menggunakan Loop yang ditulis Langsung

Berikut adalah beberapa contoh penggunaan loop untuk melakukan iterasi yang ditulis langsung pada baris perintah.

x=2; repeat x=(x+2/x)/2; until $x^2 = 2; end; x$,

1.41421356237

Penggabungan matriks menggunakan tanda "—" dapat digunakan untuk menyimpan semua hasil iterasi.

v=[1]; for i=2 to 8; v=v—(v[i-1]i); end; v,

[1, 2, 6, 24, 120, 720, 5040, 40320]

hasil iterasi juga dapat disimpan pada vektor yang sudah ada.

v=ones(1,100); for i=2 to cols(v); v[i]=v[i-1]i; end; ... plot2d(v,logplot=1); textbox(latex(log(n)),x=0.5):

A =[0.5,0.2;0.7,0.1]; b=[2;2]; ... x=[1;1]; repeat xnew=A.x-b; until all(xnew =x); x=xnew; end; ... x,

-7.09677 -7.74194

Iterasi di dalam Fungsi

Fungsi atau program juga dapat menggunakan iterasi dan dapat digunakan untuk melakukan iterasi. Berikut adalah beberapa contoh iterasi di dalam fungsi.

Contoh berikut adalah suatu fungsi untuk menghitung berapa lama suatu iterasi konvergen. Nilai fungsi tersebut adalah hasil akhir iterasi dan banyak iterasi sampai konvergen.

function map hiter(f, $x0$)...

x=x0; maxiter=0; repeat xnew=f(x); $maxiter = maxiter + 1; until xnew = x; x = xnew; end; return maxiter; endfunction < /pre > Misalnya, berikutadalahiterasiuntukmendapatkanh$
5, $jikadimulaidarihampiranawal2$.

hiter("(x+2/x)/2",2)

5

Karena fungsinya didefinisikan menggunakan "map". maka nilai awalnya dapat berupa vektor.

x=1.5:0.1:10; hasil=hiter("(x+2/x)/2",x); ... plot2d(x,hasil):

Dari gambar di atas terlihat bahwa kekonvergenan iterasinya semakin lambat, untuk nilai awal semakin besar, namun penambahnnya tidak kontinu. Kita dapat menemukan kapan maksimum iterasinya bertambah.

hasil[1:10]

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

x[nonzeros(differences(hasil))]

[1.5, 2, 3.4, 6.6]

maksimum iterasi sampai konvergen meningkat pada saat nilai awalnya 1.5, 2, 3.4, dan 6.6.

Contoh berikutnya adalah metode Newton pada polinomial kompleks berderajat 3.

p = x^3 − 1; $newton = x - p/diff(p, x)$;newton

Selanjutnya didefinisikan fungsi untuk melakukan iterasi (aslinya 10 kali).

function iterasi(f, $x, n = 10$)...

loop 1 to n; x=f(x); $end; returnx; endfunction < /pre > Kitamulaidenganmenentukantitik-titikgridpadabidangkompleksnya.$

r=1.5; x=linspace(-r,r,501); Z=x+Ix'; W=iterasi(newton,Z);

Berikut adalah akar-akar polinomial di atas.

z=solve(p)()

Untuk menggambar hasil iterasinya, dihitung jarak dari hasil iterasi ke-10 ke masing-masing akar, kemudian digunakan untuk menghitung warna yang akan digambar, yang menunjukkan limit untuk masing-masing nilai awal.

Fungsi plotrgb() menggunakan jendela gambar terkini untuk menggambar warna RGB sebagai matriks.

C=rgb(max(abs(W-z[1]),1),max(abs(W-z[2]),1),max(abs(W-z[3]),1)); ... plot2d(none,-r,r,-r,r); plotrgb(C):

Iterasi Simbolik

Seperti sudah dibahas sebelumnya, untuk menghasilkan barisan ekspresi simbolik dengan Maxima dapat digunakan fungsi makelist().

powerdisp:true // untuk menampilkan deret pangkat mulai dari suku berpangkat terkecil

deret = makelist(taylor(exp(x),x,0,k),k,1,3); $deret//barisanderetTayloruntuke^x$

Untuk mengubah barisan deret tersebut menjadi vektor string di EMT digunakan fungsi mxm2str(). Selanjutnya, vektor string/ekspresi hasilnya dapat digambar seperti menggambar vektor eskpresi pada EMT.

plot2d("exp(x)",0,3); // plot fungsi aslinya, e^x

plot2d(mxm2str("deret"), add,color=4:6): // plot ketiga deret taylor hampiran fungsi tersebut

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):

$sum(sin(kx)/k, k, 1, 5)$

Berikut adalah cara menggambar kurva

plot2d(sum(sin((2k+1)x)/(2k+1),k,0,20),0,2pi):

Hal serupa juga dapat dilakukan dengan menggunakan matriks, misalkan kita akan menggambar kurva

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 $x^x dix = 1$.

mxmtable("diffat($x^x, x = 1, n$)","n",1,8,$frac = 1$);

$'sum(k,k,1,n) = factor(ev(sum(k,k,1,n), simpsum = true))//simpsum$: $menghitungderetsecarasimbolik$

$'sum(1/(3^k+k),k,0,inf) = factor(ev(sum(1/(3^k+k),k,0,inf), simpsum = true))$

Di sini hasilnya masih gagal, hasilnya tidak dihitung.

$'sum(1/x^2,x,1,inf) = ev(sum(1/x^2,x,1,inf), simpsum = true)//ev : menghitungnilaiekspresi$

$'sum((-1)^(k-1)/k,k,1,inf) = factor(ev(sum((-1)^(x-1)/x,x,1,inf), simpsum = true))$

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); $'sum(ax^k,k,0,inf) = ev(sum(ax^k,k,0,inf), simpsum = true), forget(abs(x) < 1)$;

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)

$d(10) = ev(d(10), simpsum = true)$

$d(100) = ev(d(100), simpsum = true)$

¡pre class="udf"¿

¡/pre¿ Deret Taylor

Deret Taylor adalah representasi dari suatu fungsi sebagai penjumlahan deret tak hingga dari suku-suku yang didasarkan pada turunan-turunan fungsi tersebut di sekitar suatu titik. Deret ini dinamai berdasarkan matematikawan Inggris, Brook Taylor.

Jika suatu fungsi

(x) memiliki turunan hingga tak hingga di sekitar suatu titik

, maka deret Taylor dari

(x) di sekitar titik

a diberikan oleh:

Deret Taylor suatu fungsi f yang diferensiabel sampai tak hingga di sekitar x=a adalah:

* $f^(k)(a) adalah turunan ke - k dari fungsi f(x) di titik a$,

* k! adalah faktorial dari * (contohnya,3! = 3x2x1=6),

* a adalah titik di sekitar mana deret Taylor dikembangkan (sering * disebut "pusat" deret Taylor).

$'e^x = taylor(exp(x), x, 0, 10)//deretTaylore^x disekitar x = 0, sampai sukuke-11$

exp(1)

$'log(x) = taylor(log(x), x, 1, 10)//deretlog(x) disekitar x = 1$

log(2)

$'sin(x) = taylor(sin(x), x, 1, 5)$

sin(1)

6 Geometri

$23030630074_R omadhona Enggal Wilujeng_E MT Geometry Nama : Romadhona Enggal Wilujeng$

NIM : 23030630074

Kelas : Matematika E 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 geometeri tersebut disimpan di dalam file program "geometry.e", sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

load geometry

Numerical and symbolic geometry.

Fungsi-fungsi Geometri

Fungsi-fungsi untuk Menggambar Objek Geometri:

defaultd:=textheight()*1.5: nilai asli untuk parameter d setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang

koordinat

setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas sumbu-x dan y adalah -r sd r

plotPoint (P, "P"): menggambar titik P dan diberi label "P"

plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label "AB" sejauh d

plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d

plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"

plotLabel (label, P, V, d): menuliskan label pada posisi P

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

turn(v, phi): memutar vektor v sejauh phi turnLeft(v): memutar vektor v ke kiri turnRight(v): memutar vektor v ke kanan normalize(v): normal vektor v crossProduct(v, w): hasil kali silang vektorv dan w. lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh.

ax+by=c.

lineWithDirection(A,v): garis melalui A searah vektor v

getLineDirection(g): vektor arah (gradien) garis g
getNormal(g): vektor normal (tegak lurus) garis g
getPointOnLine(g): titik pada garis g
perpendicular(A, g): garis melalui A tegak lurus garis g
parallel (A, g): garis melalui A sejajar garis g
lineIntersection(g, h): titik potong garis g dan h
projectToLine(A, g): proyeksi titik A pada garis g
distance(A, B): jarak titik A dan B
distanceSquared(A, B): kuadrat jarak A dan B
quadrance(A, B): kuadrat jarak A dan B
areaTriangle(A, B, C): luas segitiga ABC
computeAngle(A, B, C): besar sudut lt;ABC
angleBisector(A, B, C): garis bagi sudut lt;ABC
circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r
getCircleCenter(c): pusat lingkaran c
getCircleRadius(c): jari-jari lingkaran c
circleThrough(A,B,C): lingkaran melalui A, B, C
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan lingkran c
circleCircleIntersections (c1, c2): titik potong lingkaran c1 dan c2
planeThrough(A, B, C): bidang melalui titik A, B, C
Fungsi-fungsi Khusus Untuk Geometri Simbolik:
getLineEquation (g,x,y): persamaan garis g dinyatakan dalam x dan y getHesseForm (g,x,y,A): bentuk Hesse garis g dinyatakan dalam x dan y dengan titik A pada
sisi positif (kanan/atas) garis
quad(A,B): kuadrat jarak AB
spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni $\sin(\text{alpha})^2 dengan$ alpha sudut yang menghadap sisi a.
crosslaw(a,b,c,sa): persamaan 3 quads dan 1 spread pada segitiga dengan panjang sisi a, b, c.
triplespread(sa,sb,sc): persamaan 3 spread sa,sb,sc yang memebntuk suatu segitiga
doublespread(sa): Spread sudut rangkap Spread 2*phi, dengan sa=$\sin(\text{phi})^2 spreada$.
Contoh 1: Luas, Lingkaran Luar, Lingkaran Dalam Segitiga
Untuk menggambar objek-objek geometri, langkah pertama adalah menentukan rentang sumbu-sumbu koordinat. Semua objek geometri akan digambar pada satu bidang koordinat, sampai didefinisikan bidang koordinat yang baru.

```
setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
```

Sekarang tetapkan tiga poin dan plot mereka.

```
A=[1,0]; plotPoint(A,"A"); // definisi dan gambar tiga titik
B=[0,1]; plotPoint(B,"B");
C=[2,2]; plotPoint(C,"C");
```

Kemudian 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 garis adalah [a,b,c], yang mewakili garis dengan persamaan ax+by=c.

lineThrough(B,C) // garis yang melalui B dan C

[-1, 2, 2]

Hitunglah garis tegak lurus yang melalui A pada BC.

h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A

Dan persimpangannya dengan BC.

D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC

Plot itu.

plotPoint(D,value=1); // koordinat D ditampilkan

aspect(1); plotSegment(A,D): // tampilkan semua gambar hasil plot...()

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 langusng dengan fungsi

1.5

Cara lain menghitung luas segitigas ABC:

distance(A,D)distance(B,C)/2

1.5

Sudut di C

degprint(computeAngle(B,C,A))

36°52'11.63"

Sekarang lingkaran luar segitiga.

c=circleThrough(A,B,C); // lingkaran luar segitiga ABC

R=getCircleRadius(c); // jari2 lingkaran luar

O=getCircleCenter(c); // titik pusat lingkaran c

plotPoint(O,"O"); // gambar titik "O"

plotCircle(c,"Lingkaran luar segitiga ABC"):

Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

O, R

[1.16667, 1.16667] 1.17851130198

Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

l=angleBisector(A,C,B); // garis bagi ¡ACB

g=angleBisector(C,A,B); // garis bagi ¡CAB

P=lineIntersection(l,g) // titik potong kedua garis bagi sudut

[0.86038, 0.86038]

Tambahkan semuanya ke plot.

color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut

plotPoint(P,"P"); // gambar titik potongnya

r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam

0.509653732104

plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"): // gambar lingkaran dalam

Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

setPlotRange(-2.5,4.5,-2.5,4.5);

A=[-2,1]; plotPoint(A,"A");

B=[1,-2]; plotPoint(B,"B");

C=[4,4]; plotPoint(C,"C");

2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut.

plotSegment(A,B,"c")

plotSegment(B,C,"a")

plotSegment(A,C,"b")

aspect(1):

3. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

l=angleBisector(A,C,B);

g=angleBisector(C,A,B);

P=lineIntersection(l,g)

[0.581139, 0.581139]

color(5); plotLine(l); plotLine(g); color(1);

plotPoint(P,"P");

r=norm(P-projectToLine(P,lineThrough(A,B)))

1.52896119631

plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):

Jadi, terbukti bahwa garis bagi sudut yang ketiga juga melalui titik pusat lingkaran dalam.

4. Gambar jari-jari lingkaran dalam.

r=norm(P-projectToLine(P,lineThrough(A,B)))

1.52896119631

plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"):

Contoh 2: Geometri Simbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

File geometri.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, kita dapat menggunakan perhitungan simbolis sekarang.

A = [1,0]; B = [0,1]; C = [2,2]; // menentukan tiga titik A, B, C

Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi memberikan perhitungan simbolis.

c = lineThrough(B,C) // c=BC

[- 1, 2, 2]

Kita bisa mendapatkan persamaan garis dengan mudah.

$getLineEquation(c,x,y)$,solve(

$getLineEquation(lineThrough([x1,y1],[x2,y2]),x,y)$,solve(

$getLineEquation(lineThrough(A,[x1,y1]),x,y)//persamaangarismelaluiAdan(x1,y1)$

h = perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC

[2, 1, 2]

Q = lineIntersection(c,h) // Q titik potong garis c=BC dan h

2 6 [-, -] 5 5

$projectToLine(A, lineThrough(B, C))//proyeksiApadaBC$

$distance(A, Q)//jarakAQ$

cc = circleThrough(A,B,C); $cc//(titikpusatdanjari-jari)lingkaranmelaluiA, B, C$

r=getCircleRadius(cc); r,float(r) // tampilkan nilai jari-jari

$computeAngle(A, C, B)//nilai < ACB$

$solve(getLineEquation(angleBisector(A, C, B), x, y), y)[1]//persamaangarisbagi < ACB$

P = lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P//titikpotong2garisbagisudut$

P() // hasilnya sama dengan perhitungan sebelumnya

[0.86038, 0.86038]

Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga dapat 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 potong.

P1,P2,f=lineCircleIntersections(l,c);

P1, P2,

[4.64575, -1.64575] [-0.645751, 3.64575]

plotPoint(P1); plotPoint(P2):

Begitu pula 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]

$lineCircleIntersections(l, c)|radcan, //titikpotonglingkarancdangarisl$

Akan ditunjukkan bahwa sudut-sudut yang menghadap bsuusr yang sama adalah sama besar.

C=A+normalize([-2,-3])4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);

degprint(computeAngle(P1,C,P2))

69°17'42.68"

C=A+normalize([-4,-3])4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);

degprint(computeAngle(P1,C,P2))

69°17'42.68"

insimg:

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 melallui kedua titik potong kedua lingkaran tersebut. Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

A=[2,2]; B=[-1,-2];
c1=circleWithCenter(A,distance(A,B));
c2=circleWithCenter(B,distance(A,B));
P1,P2,f=circleCircleIntersections(c1,c2);
l=lineThrough(P1,P2);
setPlotRange(5); plotCircle(c1); plotCircle(c2);
plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):

Selanjutnya, kami 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 terlibat. Tetapi kita dapat menyederhanakannya, jika kita memecahkan y.

g = getLineEquation(lineThrough(P1,P2),x,y);
$solve(g, y)$

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sama sekali berbeda.

$solve(getLineEquation(middlePerpendicular(A, B), x, y), y)$
h =getLineEquation(lineThrough(A,B),x,y);
$solve(h, y)$

Perhatikan hasil kali gradien garis g dan h adalah:

Artinya kedua garis tegak lurus.

Contoh 3: Rumus Heron

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a, b dan c adalah:

Untuk membuktikan hal ini kita misalkan C(0,0), B(a,0) dan A(x,y), b=AC, c=AB. Luas segitiga ABC adalah

Nilai y didapat dengan menyelesaikan sistem persamaan:

setPlotRange(-1,10,-1,8); plotPoint([0,0], "C(0,0)"); plotPoint([5.5,0], "B(a,0)"); ... plotPoint([7.5,6], "A(x,y)");
plotSegment([0,0],[5.5,0], "a",25); plotSegment([5.5,0],[7.5,6],"c",15); ... plotSegment([0,0],[7.5,6],"b",25);
plotSegment([7.5,6],[7.5,0],"t=y",25):
assume(a 0); sol = solve([$x^2 + y^2 = b^2, (x - a)^2 + y^2 = c^2$], [x, y])
[]

Ekstrak solusi y.

ysol = y with sol[2][2] $'y = sqrt(factor(ysol^2))$:

Kami mendapatkan rumus Heron.

function H(a,b,c) = sqrt(factor((ysola/2)^2));'H(a,b,c)=H(a,b,c)
$'Luas = H(2, 5, 6)//luassegitigadenganpanjangsisi - sisi2, 5, 6$

Tentu saja, setiap segitiga persegi panjang adalah kasus yang terkenal.

$'Luas = H(3, 4, 5)//luassegitigasiku - sikudenganpanjangsisi3, 4, 5$

Dan juga jelas, bahwa ini adalah segitiga dengan luas maksimal dan dua sisi 3 dan 4.

aspect (1.5); plot2d(H(3,4,x),1,7): // Kurva luas segitiga sengan panjang sisi 3, 4, x (1¡= x ¡=7)

Wrong argument!

Cannot combine a symbolic expression here. Did you want to create a symbolic expression? Then start with amp;.

This function can only use real or complex values! Error in abs Error in expression: 3*abs(ysol)/2 y0=f($x[1], args()$); $adaptiveevalone : s = Try"traceerrors"toinspectlocalvariablesafte$ $dw/n, dw/n^2, dw/n, auto; args()$;

Kasus umum juga berfungsi.

$solve(diff(H(a,b,c)^2, c) = 0, c)$

Maxima said: diff: second argument must be a variable; found [1,0,4] – an error. To debug this try: debugmode(true);

Error in: $solve(diff(H(a,b,c)^2, c) = 0, c)$...

Sekarang mari kita cari himpunan semua titik di mana b+c=d untuk beberapa konstanta d. Diketahui bahwa ini adalah elips.

s1 = subst(d-c,b,sol[2]); $s1$

Maxima said: part: invalid index of list or matrix. – an error. To debug this try: debugmode(true);

Error in: s1 amp;= subst(d-c,b,sol[2]); $s1$...

Dan buat fungsi ini.

function fx(a,c,d) = rhs(s1[1]); $fx(a,c,d), functionfy(a,c,d) = rhs(s1[2])$;fy(a,c,d)

Sekarang kita bisa menggambar setnya. Sisi b bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan elips.

aspect(1); plot2d(fx(3,x,5),fy(3,x,5),xmin=1,xmax=4,square=1):

Kita dapat memeriksa persamaan umum untuk elips ini, yaitu.

di mana (xm,ym) adalah pusat, dan u dan v adalah setengah sumbu.

$ratsimp((fx(a,c,d) - a/2)^2/u^2 + fy(a,c,d)^2/v^2 with[u = d/2, v = sqrt(d^2 - a^2)/2])$

Kita lihat bahwa tinggi dan luas segitiga adalah maksimal untuk x=0. Jadi luas segitiga dengan a+b+c=d maksimal jika segitiga sama sisi. Kami ingin menurunkan ini secara analitis.

eqns = [diff(H(a,b,d-(a+b))$^2, a) = 0, diff(H(a,b,d-(a+b))^2, b) = 0]$;eqns

Kami mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusinya a=b=c=d/3.

$solve(eqns, [a,b])$

Ada juga metode Lagrange, memaksimalkan H(a,b,c)$^2 terhadapa + b + d = d$.

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: ... b)=la, diff(H(a,b,c)$^2, c) = la, a + b + c = d], [a, b, c, la])$...

Kita bisa membuat plot situasinya

Pertama-tama atur poin di Maxima.

A = at([x,y],sol[2]) A

[a1, a2]

B = [0,0]; $B, C = [a, 0]$;C

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 segmen.

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 tegak lurus tengah di Maxima.

h = middlePerpendicular(A,B); g = middlePerpendicular(B,C);

Dan pusat lingkaran.

U = lineIntersection(h,g);

Kami mendapatkan rumus untuk jari-jari lingkaran.

assume(a 0,b 0,c 0); $distance(U,B)|radcan$

Mari kita tambahkan ini ke 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 $^{E}rrorinexpression : [a/2, (a2^2 + a1^2 - a * a1)/(2 * a2)]Errorin : plotPoint(U()); plotCircle(circleWithCenter(mxmeval("U"), mxmev...$

Menggunakan geometri, kami memperoleh rumus sederhana

untuk radiusnya. Kami dapat memeriksa, apakah ini benar dengan Maxima. Maxima akan memfaktorkan ini hanya jika kita kuadratkan.

$c^2/sin(computeAngle(A,B,C))^2|factor$

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari sembarang segitiga yang tidak sama sisi. Ini adalah garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk orthocenter, circumcenter, centroid, titik Exeter dan pusat lingkaran sembilan titik segitiga.

Untuk demonstrasi, kami menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kita mendefinisikan sudut-sudut segitiga di Euler. Kami menggunakan definisi, yang terlihat dalam ekspresi simbolis.

A::=[-1,-1]; B::=[2,0]; C::=[1,2];

Untuk memplot objek geometris, kami menyiapkan area plot, dan menambahkan titik ke sana. Semua plot objek geometris ditambahkan ke plot saat ini.

setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");

Kita juga bisa menambahkan sisi segitiga.

plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,""):

Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

$areaTriangle(A, B, C)$

Kita dapat menghitung koefisien sisi c.

c = lineThrough(A,B)

[- 1, 3, - 2]

Dan juga dapatkan rumus untuk baris ini.

$getLineEquation(c, x, y)$

Untuk bentuk Hesse, kita perlu menentukan sebuah titik, sehingga titik tersebut berada di sisi positif dari bentuk Hesse. Memasukkan titik menghasilkan jarak positif ke garis.

$getHesseForm(c, x, y, C)$,at(

Sekarang kita hitung lingkaran luar ABC.

LL = circleThrough(A,B,C); $getCircleEquation(LL, x, y)$

O = getCircleCenter(LL); O

Gambarkan lingkaran dan pusatnya. Cu dan U adalah simbolis. Kami mengevaluasi ekspresi ini untuk Euler.

plotCircle(LL()); plotPoint(O(),"O"):

Kita dapat menghitung perpotongan ketinggian di ABC (orthocenter) secara numerik dengan perintah berikut.

H = lineIntersection(perpendicular(A,lineThrough(C,B)),... perpendicular(B,lineThrough(A,C))); H

Sekarang kita dapat menghitung garis Euler dari segitiga.

el = lineThrough(H,O); $getLineEquation(el, x, y)$

Tambahkan ke plot kami.

plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):

Pusat gravitasi harus berada di garis ini.

M = (A+B+C)/3; $getLineEquation(el, x, y) with [x = M[1], y = M[2]]$

plotPoint(M(),"M"): // titik berat

Teorinya memberitahu kita MH=2*MO. Kita perlu menyederhanakan dengan radcan untuk mencapai ini.

$distance(M, H)/distance(M, O)|radcan$

Fungsi termasuk fungsi untuk sudut juga.

$computeAngle(A, C, B), degprint($

60°15'18.43"

Persamaan untuk pusat incircle tidak terlalu bagus.

Q = lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))—radcan; Q

Mari kita hitung juga ekspresi untuk jari-jari lingkaran yang tertulis.

r = distance(Q,projectToLine(Q,lineThrough(A,B)))—ratsimp; r

LD = circleWithCenter(Q,r); // Lingkaran dalam

Mari kita tambahkan ini ke plot.

color(5); plotCircle(LD()):

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); $p =' 0$

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

plot2d(p,level=0,add=1,contourcolor=6):

Ini seharusnya menjadi beberapa fungsi, tetapi pemecah default Maxima hanya dapat menemukan solusinya, jika kita kuadratkan persamaannya. Akibatnya, kami mendapatkan solusi palsu.

akar = solve(getHesseForm(lineThrough(A,B),x,y,C)$^2 - distance([x,y],C)^2, y)$

[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26, y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 26]

Solusi pertama adalah

maxima: akar[1]

Menambahkan solusi pertama ke plot menunjukkan, bahwa itu memang jalan yang kita cari. Teorinya memberi tahu kita bahwa itu adalah parabola yang diputar.

plot2d(rhs(akar[1]),add=1):

function g(x) = rhs(akar[1]); $'g(x) = g(x) // fungsiyangmendefinisikankurvadiatas$

T =[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut

dTC = distance(T,C); $fullratsimp(dTC)$,float(

U = projectToLine(T,lineThrough(A,B)); $U // proyeksiTpadagarisAB$

dU2AB = distance(T,U); $fullratsimp(dU2AB)$,float(

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

Ini terinspirasi dari ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti pengertian klasik tentang jarak dan sudut dengan kuadrat dan penyebaran. Dengan menggunakan 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 menyembunyikan keuntungan utama dari trigonometri rasional bahwa perhitungan hanya dapat dilakukan dengan kertas dan pensil. Anda diundang untuk memeriksa hasil tanpa komputer.

Intinya adalah bahwa perhitungan rasional simbolis sering kali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

load geometry;

Untuk pengenalan pertama, kami menggunakan segitiga persegi panjang dengan proporsi Mesir terkenal 3, 4 dan 5. Perintah berikut adalah perintah Euler untuk merencanakan geometri bidang yang terdapat dalam file Euler "geometry.e".

C:=[0,0]; A:=[4,0]; B:=[0,3]; ... setPlotRange(-1,5,-1,5); ... plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ... plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ... insimg(30);

Tentu saja,

di mana wa adalah sudut di A. Cara yang biasa untuk menghitung sudut ini, adalah dengan mengambil invers dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak kira-kira.

wa := arcsin(3/5); degprint(wa)

36°52'11.63"

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadran, yang menggantikan jarak. Sebenarnya, itu hanya jarak kuadrat. Berikut ini, a, b, dan c menunjukkan kuadrat dari sisi-sisinya.

Teorema Pythogoras menjadi a+b=c.

$a = 3^2; b = 4^2; c = 5^2; a + b = c$

$25 = 25$

Pengertian kedua dari trigonometri rasional adalah penyebaran. Spread mengukur pembukaan antar baris. Ini adalah 0, jika garis-garisnya sejajar, dan 1, jika garis-garisnya persegi panjang. Ini adalah kuadrat sinus sudut antara dua garis.

Penyebaran garis AB dan AC pada gambar di atas didefinisikan sebagai:

di mana a dan c adalah kuadrat dari sembarang segitiga siku-siku dengan salah satu sudut di A.

sa = a/c; *sa*

Ini lebih mudah dihitung daripada sudut, tentu saja. Tetapi Anda kehilangan properti bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengonversi nilai perkiraan untuk sudut wa menjadi sprad, dan mencetaknya sebagai pecahan.

fracprint($\sin(\text{wa})^2$)

9/25

Hukum kosinus trgonometri klasik diterjemahkan menjadi "hukum silang" berikut.

Di sini a, b, dan c adalah kuadrat dari sisi-sisi segitiga, dan sa adalah penyebaran sudut A. Sisi a, seperti biasa, berhadapan dengan sudut A.

Hukum ini diimplementasikan dalam file geometri.e yang kami muat ke Euler.

$crosslaw(aa, bb, cc, saa)$

Dalam kasus kami, kami mendapatkan

$crosslaw(a, b, c, sa)$

Mari kita gunakan crosslaw ini untuk mencari spread di A. Untuk melakukan ini, kita buat crosslaw untuk kuadran a, b, dan c, dan selesaikan untuk spread yang tidak diketahui sa.

Anda dapat melakukannya dengan tangan dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasilnya, kami sudah memilikinya.

$crosslaw(a, b, c, x)$,solve(

Kita sudah tahu ini. Definisi spread adalah kasus khusus dari crosslaw.

Kita juga dapat menyelesaikan ini untuk umum a,b,c. Hasilnya adalah rumus yang menghitung penyebaran sudut segitiga yang diberikan kuadrat dari ketiga sisinya.

$solve(crosslaw(aa, bb, cc, x), x)$

Kita bisa membuat fungsi dari hasilnya. Fungsi seperti itu sudah didefinisikan dalam file geometri.e dari Euler.

$spread(a, b, c)$

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi

Hasilnya rasional, yang tidak begitu mudah didapat jika kita menggunakan trigonometri klasik.

$spread(a, a, 4a/7)$

Ini adalah sudut dalam derajat.

degprint(arcsin(sqrt(6/7)))

67°47'32.44"

Contoh lain

Sekarang, mari kita coba contoh yang lebih maju.

Kami mengatur tiga sudut segitiga sebagai berikut.

A:=[1,2]; B:=[4,3]; C:=[0,4]; ... setPlotRange(-1,5,1,7); ... plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ... plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ... insimg;

Menggunakan Pythogoras, mudah untuk menghitung jarak antara dua titik. Saya pertama kali menggunakan jarak fungsi file Euler untuk geometri. Jarak fungsi menggunakan geometri klasik.

$distance(A, B)$

Euler juga mengandung fungsi untuk kuadran antara dua titik.

Dalam contoh berikut, karena c+b bukan a, maka segitiga itu bukan persegi panjang.

c = quad(A,B); $c, b = quad(A, C)$;b, a = quad(B,C); a,

Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode biasa berdasarkan hasil kali titik dua vektor. Hasilnya adalah beberapa pendekatan floating point.

wb = computeAngle(A,B,C); wb,(wb/pi180)()

32.4711922908

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kami memasukkan kuadran a, b, dan c ke dalam hukum silang dan menyelesaikan x.

$crosslaw(a, b, c, x)$,solve(

Yaitu, apa yang dilakukan oleh penyebaran fungsi yang didefinisikan dalam "geometry.e".

sb = spread(b,a,c); sb

Maxima mendapatkan hasil yang sama menggunakan trigonometri biasa, jika kita memaksanya. Itu menyelesaikan istilah sin(arccos(...)) menjadi hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

$sin(computeAngle(A, B, C))^2$

Setelah kita memiliki spread di B, kita dapat menghitung tinggi ha di sisi a. Ingat bahwa

Menurut definisi.

ha = csb; ha

Gambar berikut telah dihasilkan dengan program geometri C.a.R., yang dapat menggambar kuadrat dan menyebar.

image: (20) Rational$_G eometry_C aR.png$

Menurut definisi, panjang ha adalah akar kuadrat dari kuadratnya.

$sqrt(ha)$

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita berhadapan dengan kuadrat!

$sqrt(ha)sqrt(a)/2$

Rumus determinan biasa menghasilkan hasil yang sama.

$areaTriangle(B, A, C)$

Rumus Bangau

Sekarang, mari kita selesaikan masalah ini secara umum!

remvalue(a,b,c,sb,ha);

Pertama kita hitung spread di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat ("quadrea"?), faktorkan dengan Maxima, dan kita mendapatkan rumus Heron yang terkenal.

Memang, ini sulit dilakukan dengan pensil dan kertas.

$spread(b^2, c^2, a^2)$,factor(

Aturan Triple Spread

Kerugian dari spread adalah mereka tidak lagi hanya menambahkan sudut yang sama.

Namun, tiga spread dari sebuah segitiga memenuhi aturan "triple spread" berikut.

remvalue(sa,sb,sc); $triplespread(sa, sb, sc)$

Aturan ini berlaku untuk setiap tiga sudut yang menambah 180 °.

Sejak menyebar

sama, aturan triple spread juga benar, jika

Karena penyebaran sudut negatif adalah sama, aturan penyebaran rangkap tiga juga berlaku, jika

Misalnya, kita dapat menghitung penyebaran sudut 60°. Ini 3/4. Persamaan memiliki solusi kedua, bagaimanapun, di mana semua spread adalah 0.

$solve(triplespread(x, x, x), x)$

Sebaran 90° jelas 1. Jika dua sudut dijumlahkan menjadi 90°, sebarannya menyelesaikan persamaan sebaran rangkap tiga dengan a,b,1. Dengan perhitungan berikut kita mendapatkan a+b=1.

$triplespread(x, y, 1)$,solve(

Karena sebaran 180°-t sama dengan sebaran t, rumus sebaran rangkap tiga juga berlaku, jika satu sudut adalah jumlah atau selisih dua sudut lainnya.

Jadi kita dapat menemukan penyebaran sudut berlipat ganda. Perhatikan bahwa ada dua solusi lagi. Kami membuat ini fungsi.

$solve(triplespread(a, a, x), x), functiondoublespread(a) = factor(rhs($

- 4 (a - 1) a

Pembagi Sudut

Ini 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;

Mari kita hitung panjang garis bagi sudut di A. Tetapi kita ingin menyelesaikannya untuk umum a,b,c.

remvalue(a,b,c);

Jadi pertama-tama kita hitung penyebaran sudut yang dibagi dua di A, dengan menggunakan rumus sebaran rangkap tiga.

Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut terbelah 180 °-wa.

$triplespread(x, x, a/(a+b))$,solve(

Mari kita periksa persegi panjang Mesir.

$sa2with[a = 3^2, b = 4^2]$

Kami dapat mencetak sudut dalam Euler, setelah mentransfer penyebaran ke radian.

wa2 := arcsin(sqrt(1/10)); degprint(wa2)

18°26'5.82"

Titik P adalah perpotongan garis bagi sudut dengan sumbu y.

P := [0,tan(wa2)4]

[0, 1.33333]

plotPoint(P,"P"); plotSegment(A,P):

Mari kita periksa sudut dalam contoh spesifik kita.

computeAngle(C,A,P), computeAngle(P,A,B)

0.321750554397 0.321750554397

Sekarang kita hitung panjang garis bagi AP.

Kami menggunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

berlaku dalam segitiga apa pun. Kuadratkan, itu diterjemahkan ke dalam apa yang disebut "hukum penyebaran"

di mana a,b,c menunjukkan qudrances.

Karena spread CPA adalah 1-sa2, kita dapatkan darinya bisa/1=b/(1-sa2) dan dapat menghitung bisa (kuadran dari garis-bagi sudut).

factor(ratsimp(b/(1-sa2))); bisa =

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

sqrt(mxmeval("at(bisa,[a=$3^2, b = 4^2$])")), $distance(A, P)$

4.21637021356 4.21637021356

Kita juga dapat menghitung P menggunakan rumus spread.

py=factor(ratsimp(sa2bisa)); py

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

sqrt(mxmeval("at(py,[a=$3^2, b = 4^2$])"))

1.33333333333

Sudut Akord

Perhatikan situasi berikut.

setPlotRange(1.2); ... color(1); plotCircle(circleWithCenter([0,0],1)); ... A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ... plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ... color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ... color(1); O:=[0,0]; plotPoint(O,"0"); ... plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ... insimg;

Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran rangkap tiga untuk sudut-sudut di pusat O untuk r. Jadi kita mendapatkan rumus untuk jari-jari kuadrat dari pericircle dalam hal kuadrat dari sisi.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru

rabc = rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]);

rabc

Kita dapat menjadikannya sebagai fungsi Euler.

function periradius(a,b,c) = rabc;

Mari kita periksa hasilnya untuk poin 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, spread CBA hanya bergantung pada b dan c. Ini adalah teorema sudut chord.

$spread(b,a,c)rabc|ratsimp$

Sebenarnya spreadnya adalah b/(4r), dan kita melihat bahwa sudut chord dari chord b adalah setengah dari sudut pusat.

$doublespread(b/(4r)) - spread(b,r,r)|ratsimp$

Contoh 6: Jarak Minimal pada Bidang

Catatan awal

Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis level yang agak 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"):

dan grafiknya juga agak sederhana: bagian atas kerucut:

plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):

Tentu saja minimal 0 dicapai di A.

Dua poin

Sekarang kita lihat fungsi MA+MB dimana A dan B adalah dua titik (tetap). Ini adalah "fakta yang diketahui" bahwa kurva level adalah elips, titik fokusnya adalah A dan B; kecuali untuk AB minimum yang konstan pada segmen [AB]:

B=[1,-1];

function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2 + (y − B[2])^2)

fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):

Grafiknya lebih menarik:

plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):

Pembatasan garis (AB) lebih terkenal:

plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):

Tiga poin

Sekarang hal-hal yang kurang sederhana: Ini sedikit kurang terkenal bahwa MA+MB+MC mencapai minimum pada satu titik pesawat tetapi untuk menentukan itu kurang sederhana:

1) Jika salah satu sudut segitiga ABC lebih dari 120° (katakanlah di A), maka minimum dicapai pada titik ini (misalnya AB+AC).

Contoh:

C=[-4,1];

function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2 + $(y - C[2])^2$)

plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);

insimg;

fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");

P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);

insimg;

2) Tetapi jika semua sudut segitiga ABC kurang dari 120 °, minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (maka masing-masing 120 °):

C=[-0.5,1];

plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):

fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");

P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);

insimg;

Merupakan kegiatan yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya tahu soft yang ditulis di Jawa yang memiliki instruksi "garis kontur" ...

Semua ini di atas telah ditemukan oleh seorang hakim Perancis bernama Pierre de Fermat; dia menulis surat kepada dilettants lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di pajak penghasilan. Jadi titik unik F sedemikian rupa sehingga FA+FB+FC minimal, disebut titik Fermat segitiga. Tetapi tampaknya beberapa tahun sebelumnya, Torriccelli Italia telah menemukan titik ini sebelum Fermat melakukannya! Bagaimanapun tradisinya adalah mencatat poin ini F...

Empat poin

Langkah selanjutnya adalah menambahkan 4 titik D dan mencoba meminimalkan MA+MB+MC+MD; katakan bahwa Anda adalah operator TV kabel dan ingin mencari di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan panjang kabel sesedikit mungkin!

D=[1,1];

function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2 + $(y - D[2])^2$)

plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):

fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);
insimg;

Masih ada minimum dan tidak tercapai di salah satu simpul A, B, C atau D:

function f(x):=d4(x[1],x[2])
neldermin("f",[0.2,0.2])
[0.142858, 0.142857]

Tampaknya dalam kasus ini, koordinat titik optimal adalah rasional atau mendekati rasional...

Sekarang ABCD adalah persegi, kami berharap bahwa titik optimal akan menjadi pusat ABCD:

C=[-1,1];
plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1):
fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);
P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);
insimg;

Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini, jika Anda telah menginstal Povray, dan pvengine.exe di jalur program.

Pertama kita hitung jari-jari bola.

Jika Anda melihat gambar di bawah, Anda melihat bahwa kita membutuhkan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kami menggunakan file geometri.e dari Euler untuk ini.

load geometry;

Pertama dua garis yang 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)); $d1$

Hitung jarak ke g.

d = distance(P,projectToLine(P,g)); d

Dan temukan pusat kedua lingkaran yang jaraknya sama.

sol = solve($d1^2 = d^2, u$);sol

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;
```

Plot dengan Povray

Selanjutnya kami merencanakan 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
pvengine.exe"
```

C:Files-Ray3̆.7.exe

Kami mengatur adegan dengan tepat.

```
povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Selanjutnya kita menulis dua bidang 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 terbatas 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 menghasilkan dua titik pada lingkaran, di mana bola menyentuh kerucut.

```
function turnz(v) := return [-v[2],v[1],v[3]]
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
```

Kemudian kami menghasilkan dua titik di mana bola menyentuh bidang. Ini adalah fokus dari 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, di mana bola menyentuh kerucut.

```
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc1],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc2],povlook(gray)));
```

Mulai program Povray.

```
povend();
```

Untuk mendapatkan Anaglyph ini kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali 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
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 ¡/pre¿ Anda membutuhkan kacamata merah/sian untuk menghargai efek berikut.
```

```
povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Contoh 8: Geometri Bumi

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan sferis. Fungsi-fungsi tersebut terdapat dalam file "spherical.e" di folder contoh. Kita perlu memuat file itu terlebih dahulu.

```
load "spherical.e";
```

Untuk memasukkan posisi geografis, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut 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 spherical).

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 ke yang lain pada bola ideal. Vektor ini [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 baik. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak yang begitu pendek hasilnya hampir sama.

esdist(FMIPA,Semarang)-" km", // perkiraan jarak FMIPA-Semarang

88.0114026318 km

Ada fungsi untuk heading, dengan mempertimbangkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

sdegprint(esdir(FMIPA,Solo))

65.34°

Sudut 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 dalam asum-pi.

(asum-pi)rearth(48°)^2 − "km^2", //$perkiraanluassegitigaFMIPA - Solo - Semarang$

2116.02948749 km^2

Ada fungsi untuk ini, yang menggunakan garis lintang rata-rata segitiga untuk menghitung jari-jari bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

esarea(Solo,FMIPA,Semarang)-" km^2", //$perkiraanyangsamadenganfungsiesarea$()

2123.64310526 km^2

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi heading dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kami menggunakan vektor. Untuk menambahkan vektor ke posisi, kami menggunakan vektor sadd.

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 yang ideal. Hal yang sama 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 beralih ke contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];

sposprint(Tugu), sposprint(Monas)

S 7°46.998' E 110°21.966' S 6°10.500' E 106°48.717'

Menurut Google Earth, jaraknya adalah 429,66 km. Kami mendapatkan pendekatan yang baik.

esdist(Tugu,Monas)- " km", // perkiraan jarak Tugu Jogja - Monas Jakarta

431.565659488 km

Judulnya sama dengan judul yang dihitung di Google Earth.

degprint(esdir(Tugu,Monas))

294°17'2.85"

Namun, kita tidak lagi mendapatkan posisi target yang tepat, jika kita menambahkan heading dan jarak ke posisi semula. Hal ini terjadi, karena kita tidak menghitung fungsi invers secara tepat, tetapi mengambil perkiraan jari-jari bumi di sepanjang jalan.

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 kita ingin menempuh jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti heading yang konstan!

Perhitungan berikut menunjukkan bahwa kami jauh dari tujuan yang benar, jika kami menggunakan pos yang sama selama perjalanan kami.

dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);

Sekarang kita tambahkan 10 kali sepersepuluh dari jarak, menggunakan pos ke Monas, kita sampai di Tugu.

p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;

Hasilnya jauh.

sposprint(p), skmprint(esdist(p,Monas))

S 6°11.250' E 106°48.372' 1.529km

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

P1=[30°,10°]; P2=[30°,50°];

Jalur terpendek dari P1 ke P2 bukanlah lingkaran garis lintang 30°, melainkan jalur terpendek yang dimulai 10° lebih jauh ke utara di P1.

sdegprint(esdir(P1,P2))

79.69°

Tapi, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kami menyesuaikannya pada 1/10 dari total jarak.

p=P1; dist=esdist(P1,P2); ... loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;

79.69° 81.67° 83.71° 85.78° 87.89° 90.00° 92.12° 94.22° 96.29° 98.33°

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti heading yang sama terlalu lama.

skmprint(esdist(p,P2))

0.203km

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan pos setelah 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 di 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'

Kami 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 ¡/pre¿ Sekarang rencanakan semuanya.

plot3d("testplot",angle=25, height=6, own, user,zoom=4):

Atau gunakan plot3d untuk mendapatkan tampilan anaglyph. Ini terlihat sangat bagus dengan kacamata merah/sian.

plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):

MENCOBA RUMUS-RUMUS PADA MATERI DI ATAS

Geometri Simbolik

A = [2,0]; B = [0,2]; C = [3,3]; // menentukan tiga titik A, B, C

c = lineThrough(B,C) // c=BC

[- 1, 3, 6]

$getLineEquation(c, x, y)$,solve(

h = perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC

[3, 1, 6]

Q = lineIntersection(c,h) // Q titik potong garis c=BC dan h

6 12 [-, –] 5 5

$projectToLine(A, lineThrough(B, C))//proyeksiApadaBC$

$distance(A, Q)//jarakAQ$

cc = circleThrough(A,B,C); $cc//(titikpusatdanjari-jari)lingkaranmelaluiA, B, C$

r=getCircleRadius(cc); r,float(r) // tampilkan nilai jari-jari
$computeAngle(A, C, B)//nilai < ACB$
$solve(getLineEquation(angleBisector(A, C, B), x, y), y)[1]//persamaangarisbagi < ACB$
P = lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P//titikpotong2$
P() //
[1.61803, 1.61803]
Garis dan Lingkaran yang berpotongan
A := [2,0]; c=circleWithCenter(A,4);
B := [2,3]; C := [3,2]; l=lineThrough(B,C);
setPlotRange(5); plotCircle(c); plotLine(l);
P1,P2,f=lineCircleIntersections(l,c);
P1, P2,
[5.89792, -0.897916] [1.10208, 3.89792]
plotPoint(P1); plotPoint(P2):
c = circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
[2, 0, 4]
l = lineThrough(B,C) // garis l melalui B dan C
[1, 1, 5]
$lineCircleIntersections(l, c)|radcan, //titikpotonglingkarancdangarisl$
C=A+normalize([-3,-4])4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
degprint(computeAngle(P1,C,P2))
57°58'20.06"
C=A+normalize([-4,-5])4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
degprint(computeAngle(P1,C,P2))
57°58'20.06"
insimg;
Garis Sumbu
A=[3,3]; B=[-2,-3];
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):
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];
g = getLineEquation(lineThrough(P1,P2),x,y);
$solve(g, y)$
$solve(getLineEquation(middlePerpendicular(A, B), x, y), y)$
h =getLineEquation(lineThrough(A,B),x,y);
$solve(h, y)$
Garis Euler dan Parabola
A::=[-1.5,-1.5]; B::=[3,0]; C::=[1.5,3];

setPlotRange(3); plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"):
plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,""):
$areaTriangle(A,B,C)$
c = lineThrough(A,B)
3 9 9 [- -, -, - -] 2 2 2
$getLineEquation(c,x,y)$
$getHesseForm(c,x,y,C)$,at(
LL = circleThrough(A,B,C); $getCircleEquation(LL,x,y)$
O = getCircleCenter(LL); O
plotCircle(LL()); plotPoint(O(),"O"):
H = lineIntersection(perpendicular(A,lineThrough(C,B)),... perpendicular(B,lineThrough(A,C))); H
el = lineThrough(H,O); $getLineEquation(el,x,y)$
plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
M = (A+B+C)/3; $getLineEquation(el,x,y)with[x=M[1],y=M[2]]$
plotPoint(M(),"M"): // titik berat
$distance(M,H)/distance(M,O)|radcan$
$computeAngle(A,C,B),degprint($
60°15'18.43"
Q = lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))—radcan; Q
r = distance(Q,projectToLine(Q,lineThrough(A,B)))—ratsimp; r
LD = circleWithCenter(Q,r); // Lingkaran dalam
color(5); plotCircle(LD()):
contoh lain dari materi trigonometri rasional
A:=[2,3]; B:=[5,4]; C:=[0,5]; ... setPlotRange(-1,5,1,7); ... plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ... plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ... insimg;
$distance(A,B)$
c = quad(A,B); $c,b=quad(A,C)$;b, a = quad(B,C); a,
wb = computeAngle(A,B,C); wb,(wb/pi180)()
29.7448812969
$crosslaw(a,b,c,x)$,solve(
sb = spread(b,a,c); sb
$sin(computeAngle(A,B,C))^2$
ha = csb; ha
$sqrt(ha)$
$sqrt(ha)sqrt(a)/2$
$areaTriangle(B,A,C)$
Aturan penyebaran 3 kali lipat
setPlotRange(1); ... 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,"0"); ... plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ... insimg;
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

function periradius(a,b,c) = rabc;

a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);

periradius(a,b,c)

1

$spread(b, a, c)rabc|ratsimp$

$doublespread(b/(4r)) - spread(b, r, r)|ratsimp$

Contoh 6: Jarak Minimal pada Bidang

Catatan awal

Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis level yang agak sederhana: lingkaran berpusat di A.

remvalue();

A=[-2,-2];

function d1(x,y):=sqrt((x-A[1])^2 + $(y - A[2])^2$)

fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ... title="If you see ellipses, please set your window square"):

dan grafiknya juga agak sederhana: bagian atas kerucut:

plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):

Ternyata setelah mencoba yang bisa hanya dengan memasukkan angka 1, karena ketika memakai angka 2, plot tidak membentuk kerucut diatas.

Dua poin

B=[2,-2];

function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2 + $(y - B[2])^2$)

fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):

Grafiknya lebih menarik:

plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):

Pembatasan garis (AB) lebih terkenal:

plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):

Tiga poin

Contoh:

C=[-3,2];

function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2 + $(y - C[2])^2$)

plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);

insimg;

fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");

P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);

insimg;

Tetapi jika semua sudut segitiga ABC kurang dari 120 °, minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (maka masing-masing 120 °):

C=[-1,2];

plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):

fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");

P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
insimg;

Empat poin

Langkah selanjutnya adalah menambahkan 4 titik D dan mencoba meminimalkan MA+MB+MC+MD; katakan bahwa Anda adalah operator TV kabel dan ingin mencari di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan panjang kabel sesedikit mungkin!

D=[2,21];
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;

Contoh 7: Bola Dandelin dengan Povray

load geometry;

Pertama dua garis yang membentuk kerucut.

g1 = lineThrough([0,0],[2,a])
[- a, 2, 0]
g2 = lineThrough([0,0],[-2,a])
[- a, - 2, 0]
g = lineThrough([-2,0],[2,2])
[- 2, 4, 4]
setPlotRange(-2,2,0,3);
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)); $d1$

Hitung jarak ke g.

d = distance(P,projectToLine(P,g)); d

Dan temukan pusat kedua lingkaran yang jaraknya sama.

sol = solve(d1^2 = d^2, u);sol

Ada dua solusi.

u := sol()
[0.558482, 4.77485]
dd := d()
[0.394906, 3.37633]

Plot lingkaran ke dalam gambar.

color(red);
plotCircle(circleWithCenter([0,u[1]],dd[1]),"");
plotCircle(circleWithCenter([0,u[2]],dd[2]),"");
insimg;

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.

Penyelesaian :

```
load geometry
Numerical and symbolic geometry.
setPlotRange(-3.5,3.5,-3.5,3.5);
A=[-2,-2]; plotPoint(A,"A");
B=[2,-2]; plotPoint(B,"B");
C=[0,3]; plotPoint(C,"C");
plotSegment(A,B,"c");
plotSegment(B,C,"a");
plotSegment(A,C,"b");
aspect(1):
c=circleThrough(A,B,C);
R=getCircleRadius(c);
O=getCircleCenter(c);
plotPoint(O,"O");
l=angleBisector(A,C,B);
color(2); plotLine(l); color(1);
plotCircle(c,"Lingkaran luar segitiga ABC"):
```

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.

Penyelesaian :

```
load geometry;
setPlotRange(5); P=[2,0]; Q=[4,0]; R=[0,-4];
plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R"):
sol = solve([a+b=-c,16a+4b=-c,c=-4],[a,b,c])
[[a = - 1, b = 5, c = - 4]]
```

Sehingga didapatkan nilai a = -1, b = 5 dan c = -4

```
function y=-x^2 + 5x - 4
2 - x + 5 x - 4
plot2d("-x^2 + 5x - 4", -5, 5, -5, 5) :
```

3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.

- Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung (sisinya-sisintya merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).

- Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya bertemu di satu titik.

- Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya.

- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

Penyelesaian :

load geometry

Numerical and symbolic geometry.

```
setPlotRange(-4.5,4.5,-4.5,4.5);
A=[-3,-3]; plotPoint(A,"A");
B=[3,-3]; plotPoint(B,"B");
C=[3,3]; plotPoint(C,"C");
D=[-3,3]; plotPoint(D,"D");
plotSegment(A,B,"");
plotSegment(B,C,"");
plotSegment(C,D,"");
plotSegment(A,D,"");
aspect(1):
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"):
```

Dari gambar diatas terlihat bahwa keempat garis bagi sudutnya bertemu di satu titik yaitu titik P.

```
r=norm(P-projectToLine(P,lineThrough(A,B)));
plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat ABCD"):
```

Dari gambar diatas, terlihat bahwa sisi-sisinya merupakan garis singgung lingkaran yang sama yaitu lingkaran dalam segiempat.

Akan ditunjukkan bahwa hasil kali panjang sisi-sisi yang berhadapan sama.

```
AB=norm(A-B) //panjang sisi AB
6
CD=norm(C-D) //panjang sisi CD
6
AD=norm(A-D) //panjang sisi AD
6
BC=norm(B-C) //panjang sisi BC
6
AB.CD
36
AD.BC
36
```

Terbukti bahwa hasil kali panjang sisi-sisi yang berhadapan sama yaitu 36. Jadi dapat dipastikan bahwa segiempat tersebut merupakan segiempat garis singgung.

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).

Penyelesaian :

Diketahui kedua titik fokus P = [-1,-1] dan Q = [1,-1]

P=[-1,-1]; Q=[1,-1];

function d1(x,y):=sqrt((x-P[1])^2 + $(y - P[2])^2$)

Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+$(y-P[2])^2$)+$sqrt((x-Q[1])^2+(y-Q[2])^2)$

fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):

Grafik yang lebih menarik

plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):

Batasan ke garis PQ

plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):

5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

Penyelesaian :

P=[-1,-1]; Q=[1,-1];

function d1(x,y):=sqrt((x-p[1])^2 + $(y - p[2])^2$)

Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])^2+$(y-P[2])^2$)+$sqrt((x+Q[1])^2+(y+Q[2])^2)$

fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):

Grafik yang lebih menarik

plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):

plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):

7 Statistika

EMT4Statistika$_{R}omadhonaEnggalWilujeng_{2}3030630074Nama$: $RomadhonaEnggalWilujeng$

NIM : 23030630074

Kelas : Matematika E 2023

EMT untuk Statistika

Dalam 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 beberapa latar belakang untuk memahami detailnya.

Asumsikan pengukuran berikut. Kami ingin menghitung nilai rata-rata dan standar deviasi yang diukur.

M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ... mean(M), dev(M),

999.9 2.72641400622

Kita dapat memplot plot kotak-dan-kumis untuk data. Dalam kasus kami 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 di 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 yang diberikan.

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 bisa memasukkan data mentah tersebut ke dalam sebuah tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal jangkauan, akhir jangkauan, jumlah orang 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=["from","to","count"])

from to count 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 membutuhkan nilai rata-rata dan statistik lain dari ukuran, kita perlu menghitung titik tengah rentang. Kita dapat menggunakan dua kolom pertama dari tabel kita untuk ini.

Sumbul "—" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan opsion "labc" adalah untuk menentukan header kolom.

(T[,1]+T[,2])/2 // the midpoint of each interval

157.5 161.5 165.5 169.5 173.5 177.5 181.5 185.5

Tetapi lebih mudah, untuk melipat rentang dengan vektor [1/2.1/2].

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 yang diberikan.

m,d=meandev(M,v); m, d,

169.901234568 5.98912964449

Mari kita tambahkan distribusi normal dari nilai-nilai ke plot batang di atas. Rumus untuk distribusi normal dengan mean m dan standar deviasi d adalah:

Karena nilainya antara 0 dan 1, untuk memplotnya pada bar plot harus dikalikan dengan 4 kali jumlah total data.

plot2d("qnormal(x,m,d)sum(v)4", ... xmin=min(r),xmax=max(r),thickness=3,add=1):

Meja

Di direktori notebook ini Anda menemukan file dengan tabel. Data tersebut mewakili hasil survei. Berikut adalah empat baris pertama dari file tersebut. Data 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 token.

Untuk ini, kami mendefinisikan set token. Fungsi strtokens() mendapatkan vektor string token dari string yang diberikan.

```
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 harus menyediakannya dengan ":=".

```
MT,hd=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
load over statistics;
```

Untuk mencetak, kita perlu menentukan set token yang sama. Kami mencetak empat baris pertama saja.

```
writetable(MT[1:4],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

Titik "." mewakili nilai-nilai, yang tidak tersedia.

Jika kita tidak ingin menentukan token untuk terjemahan 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);
```

Fungsi readtable() sekarang mengembalikan satu set token.

```
tok
```

m n f y g vg

Tabel berisi entri dari file dengan token yang diterjemahkan ke angka.

String khusus NA="." ditafsirkan sebagai "Tidak Tersedia", dan mendapatkan NAN (bukan angka) dalam tabel. Terjemahan ini dapat diubah dengan parameter NA, dan NAval.

```
MT[1]
```

[1, 1, 30, 2, NAN, 1.8, 2]

Berikut isi tabel dengan angka 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 output readtable() ke dalam daftar.

Table=readtable("table.dat",ctok=ctok);

Menggunakan kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita dapat menentukan ctok, tok, dll. Atau menggunakan daftar Tabel.

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, melewatkan setiap baris dengan nilai NAN("." dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

c,i=tablecol(MT,[5,6]);

Kita dapat 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 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]

Kami dapat mencetak hasilnya dalam 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 dalam v di baris ke-5.

MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5);

Sekarang kita dapat mencetak tabel, dengan nilai yang diekstrak 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 berikutnya, kami ingin menghubungkan dua kolom tabel. Jadi kami mengekstrak kolom 2 dan 4 dan mengurutkan tabel.

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 n m y 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 f y f y

Dengan getstatistics(), kita juga bisa menghubungkan hitungan 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

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 file tersebut.

fileremove(filename);

Distribusi

Dengan plot2d, terdapat metode yang sangat mudah untuk memplot sebaran data eksperimen.

p=normal(1,1000); //1000 random normal-distributed sample p

plot2d(p,distribution=20,style="
/"); // plot the random sample p

plot2d("qnormal(x,0,1)",add=1): // add the standard normal distribution plot

Harap perhatikan perbedaan antara plot batang (sampel) dan kurva normal (distribusi nyata). Masukkan kembali tiga 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

p=normal(10,1000); boxplot(p):

Untuk membangkitkan bilangan bulat acak, Euler memiliki intrarandom. Mari kita simulasikan lemparan dadu dan plot distribusinya.

Kita menggunakan fungsi getmultiplicities(v,x), yang menghitung seberapa sering elemen v muncul di x. Kemudian kita memplot hasilnya menggunakan columnplot().

k=intrandom(1,6000,6); ... columnsplot(getmultiplicities(1:6,k)); ... ygrid(1000,color=red):

Sementara intrandom(n,m,k) mengembalikan bilangan bulat yang terdistribusi secara seragam dari 1 ke k, dimungkinkan untuk menggunakan distribusi bilangan bulat lain yang diberikan 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,

[391, 96, 513]

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Coba lihat referensinya.

Misalnya, kami mencoba distribusi eksponensial. Variabel acak kontinu X dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

dengan parameter

plot2d(randexponential(1,1000,2), distribution):

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):

Probabilitas untuk berada di area 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 mean dan deviasi yang sama. Fungsi invbindis() memecahkan interpolasi linier antara nilai bilangan bulat.

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. Jadi kita mendapatkan plot dari 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 yang akurat untuk mengevaluasi distribusi. Mari kita periksa chidis() dengan integral.

Penamaan mencoba untuk konsisten. Misalnya.,

* distribusi chi-kuadrat adalah chidis(),
* fungsi kebalikannya adalah invchidis(),
* densitasnya 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

Untuk menentukan distribusi diskrit Anda sendiri, Anda dapat menggunakan metode berikut.

Pertama kita mengatur fungsi distribusi.

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]

Artinya dengan probabilitas wd[i+1]-wd[i] kita menghasilkan nilai acak i.

Ini hampir merupakan distribusi yang seragam. Mari kita tentukan generator angka acak untuk ini. Fungsi find(v,x) menemukan nilai x dalam vektor v. Fungsi ini juga berlaku untuk vektor x.

function wrongdice (n,m) := find(wd,random(n,m))

Kesalahannya sangat halus sehingga kami melihatnya hanya dengan iterasi yang sangat banyak.

columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):

Berikut adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai 1...K dalam v. Kami menerima hasilnya, jika untuk semua frekuensi

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 ¡/pre¿ Memang fungsi menolak distribusi seragam.

checkrandom(wrongdice(1,1000000))

0

Dan itu menerima generator acak bawaan.

checkrandom(intrandom(1,1000000,6))

1

Kita dapat menghitung distribusi binomial. Pertama ada binomialsum(), yang mengembalikan probabilitas i atau kurang hit dari n percobaan.

bindis(410,1000,0.4)

0.751401349654

Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p. Level default adalah alfa.

Arti 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. Tapi untuk n besar, penjumlahan langsungnya tidak akurat dan lambat.

p=0.4; i=0:410; n=1000; sum(bin(n,i)p$^i(1-p)^(n-i)$)

0.751401349655

Omong-omong, invbinsum() menghitung kebalikan dari binomialsum().

invbindis(0.75,1000,0.4)

409.932733047

Di Bridge, kami mengasumsikan 5 kartu beredar (dari 52) dengan dua tangan (26 kartu). Mari kita hitung probabilitas distribusi yang lebih buruk dari 3:2 (mis. 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])

395 107 498 399 96 505 428 87 485 400 99 501 373 109 518 418 108 474 392 116 492 404 93 503 379 114 507 404 107 489

Merencanakan Data

Untuk memplot data, kami mencoba hasil pemilu Jerman sejak 1990, yang 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 partai, kami menggunakan rangkaian nama.

P:=["CDU/CSU","SPD","FDP","Gr","Li"];

Mari kita cetak persentasenya dengan baik.

Pertama, kami mengekstrak kolom yang diperlukan. Kolom 3 sampai 7 adalah kursi masing-masing partai, dan kolom 2 adalah jumlah kursi. kolom adalah tahun pemilihan.

BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';

Kemudian kami mencetak statistik dalam bentuk tabel. Kami menggunakan nama sebagai tajuk kolom, dan tahun sebagai tajuk untuk baris. Lebar default untuk kolom adalah wc=10, tetapi kami lebih memilih hasil 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 mengekstrak jumlah persentase dari dua partai besar yang menunjukkan bahwa partai-partai kecil telah mendapatkan rekaman di parlemen 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 masing-masing pihak.

CP:=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];

Sekarang kita bisa memplot hasil pemilu 2009 dan mengubahnya menjadi satu plot menggunakan gambar. Kita dapat menambahkan vektor kolom ke setiap plot.

figure(2,1); ... figure(1); columnsplot(BW[6,3:7],P,color=CP); ... figure(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); ... labelbox(P,colors=CP,styles="[]", points,w=0.2,x=0.3,y=0.4):

Plot kolom 3D menunjukkan deretan data statistik dalam bentuk kolom. Kami menyediakan label untuk baris dan kolom. sudut adalah sudut pandang.

columnsplot3d(BT,scols=P,srows=YT, ... angle=30°,ccols=CP):

Representasi lainnya adalah plot mozaik. Perhatikan bahwa kolom plot mewakili kolom matriks di sini. Karena panjang label CDU/CSU, kami mengambil 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 membentuk koalisi, kami menyusun ulang elemennya.

i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):

Ini 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):

Tetapi 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 membutuhkan vektor nilai. Selain itu, ia dapat mengatur labelnya ke apa pun yang kita inginkan, kita telah mendemonstrasikannya di tutorial ini.

Ini adalah aplikasi lain, di mana kami menghitung karakter dalam sebuah kalimat dan memplot 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; ... columnsplot(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)p$^i(1-p)^(n-i)$; ... $columnsplot(x, lab = i, width = 0.05, < frame, < grid)$; ...$yaxis(0, 0 : 0.1 : 1, style = "-", left)$; $xaxis(0, style = ".")$; ...$label("p", 0, 0.25), label("i", 11, 0)$; ...$textbox(["Binomialdistribution","withp = 0.4"])$:

Berikut ini adalah cara memplot frekuensi bilangan dalam vektor.

Kami membuat vektor bilangan acak bilangan bulat 1 hingga 6.

v:=intrandom(1,10,10)

[3, 3, 7, 1, 10, 2, 7, 2, 8, 2]

Kemudian ekstrak angka unik di v.

vu:=unique(v)

[1, 2, 3, 7, 8, 10]

Dan plot frekuensi dalam plot kolom.

columnsplot(getmultiplicities(vu,v),lab=vu,style="/"):

Kami ingin menunjukkan fungsi untuk distribusi nilai empiris.

x=normal(1,20);

Fungsi empdist(x,vs) membutuhkan array nilai yang diurutkan. Jadi kita harus mengurutkan x sebelum kita dapat menggunakannya.

xs=sort(x);

Kemudian kami memplot distribusi empiris dan beberapa batang kepadatan menjadi 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,¡bar)); ... figure(0):

Plot pencar 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, kami ingin membandingkan dua sampel dari distribusi yang berbeda. Ini dapat dilakukan dengan plot kuantil-kuantil.

Untuk pengujian, 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 jelas menunjukkan bahwa nilai terdistribusi normal cenderung lebih kecil di ujung ekstrim.

Jika kita memiliki dua distribusi dengan ukuran berbeda, kita dapat memperluas yang lebih kecil atau mengecilkan 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, kami 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 fit linier.

p=polyfit(x,y,1)

[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 poin 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,st"); ... plot2d("evalpoly(x,pw)",5,10, add,color=red,style="–"); ... figure(2); plot2d(w,1,10, filled,style="/",fillcolor=red,xl=w); ... figure(0):

Sebagai contoh lain kita membaca survei siswa, umur mereka, umur orang tua mereka dan jumlah saudara dari sebuah file.

Tabel ini berisi "m" dan "f" di kolom kedua. Kami menggunakan variabel tok2 untuk menyetel terjemahan yang tepat alih-alih membiarkan readtable() mengumpulkan terjemahan.

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 berasal dari sebar berpasangan.

scatterplots(tablecol(MS,3:5),hd[3:5]):

Jelas terlihat bahwa usia ayah dan ibu saling bergantung. Mari kita tentukan dan plot garis regresi.

cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1)

[17.3789, 0.740964]

Ini jelas model yang salah. Garis regresi adalah s=17+0,74t, di mana t adalah umur ibu dan s adalah umur ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak terlalu banyak.

Sebaliknya, kami mencurigai fungsi seperti s=a+t. Maka a adalah rata-rata 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 pencar.

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. Ini hanya menunjukkan, bahwa usianya berbeda.

boxplot(cs,["mothers","fathers"]):

Menariknya, perbedaan median tidak sebesar perbedaan rata-rata.

median(cs[2])-median(cs[1])

1.5

Koefisien korelasi menunjukkan korelasi positif.

correl(cs[1],cs[2])

0.7588307236

Korelasi peringkat adalah ukuran untuk urutan yang sama di kedua vektor. 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.

di mana 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))

-1.03341851867

Ini adalah fungsi lain, yang disebut koefisien kemiringan Pearson.

function skew1 (x) := 3(mean(x)-median(x))/dev(x)

skew1(data)

0.374685157988

Simulasi Monte Carlo

Euler dapat digunakan untuk mensimulasikan kejadian acak. Kita telah melihat contoh sederhana di atas. Ini satu lagi, yang mensimulasikan 1000 kali 3 lemparan dadu, dan meminta distribusi jumlahnya.

ds:=sum(intrandom(1000,3,6))'; fs=getmultiplicities(3:18,ds)

[4, 8, 27, 48, 70, 105, 94, 139, 122, 116, 96, 73, 43, 35, 13, 7]

Kita bisa merencanakan ini sekarang.

columnsplot(fs,lab=3:18):

Untuk menentukan distribusi yang diharapkan tidak begitu mudah. Kami menggunakan rekursi lanjutan untuk ini.

Fungsi berikut menghitung banyaknya cara bilangan k dapat dinyatakan sebagai jumlah dari n bilangan dalam rentang 1 sampai m. Ini bekerja secara rekursif dengan cara yang jelas.

function map countways (k; n, m) ...

if n==1 then return k¿=1 k¡=m else sum=0; loop 1 to m; sum=sum+countways(k-,n-1,m); end; return sum; end; endfunction ¡/pre¿ Inilah hasil lemparan dadu sebanyak tiga kali.

cw=countways(3:18,3,6)

[1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3, 1]

Kami menambahkan nilai yang diharapkan ke plot.

plot2d(cw/$6^3$1000, *add*); *plot2d*(*cw*/$6^3$1000, *points*, *add*) :

Untuk simulasi lain, penyimpangan 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)')

0.311229033136

plot2d(mean(M)', distribution):

Median dari 10 bilangan acak terdistribusi 0-1-normal memiliki deviasi yang lebih besar.

dev(median(M)')

0.369103707236

Karena kita dapat dengan mudah membuat jalan acak, kita dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami kemudian memplot standar deviasi dan rata-rata langkah ke-n dari proses ini bersama dengan nilai yang diharapkan dalam warna 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):

Tes

Tes adalah alat penting dalam statistik. Di Euler, banyak tes yang diterapkan. Semua tes ini mengembalikan kesalahan yang kami terima jika kami menolak hipotesis nol.

Sebagai contoh, kami menguji lemparan dadu untuk distribusi seragam. Pada 600 lemparan, kami mendapat nilai berikut, yang kami masukkan ke uji chi-square.

chitest([90,103,114,101,103,89],dup(100,6)')

0.498830517952

Tes chi-kuadrat juga memiliki mode yang menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya harus hampir sama. Parameter gt;p menginterpretasikan vektor-y sebagai vektor probabilitas.

chitest([90,103,114,101,103,89],dup(1/6,6)', p, montecarlo)

0.499

Kesalahan ini terlalu besar. Jadi kita tidak bisa menolak pemerataan distribusi. Ini tidak membuktikan bahwa dadu kami adil. Tapi kita tidak bisa menolak hipotesis kita.

Selanjutnya kami menghasilkan 1000 lemparan dadu menggunakan generator angka acak, dan melakukan pengujian yang sama.

n=1000; t=random([1,n6]); chitest(count(t6,6),dup(n,6)')

0.704302790658

Mari kita uji nilai rata-rata 100 dengan uji-t.

s=200+normal([1,100])10; ... ttest(mean(s),dev(s),100,200)

0.441731057071

Fungsi ttest() membutuhkan nilai rata-rata, simpangan, jumlah data, dan nilai rata-rata untuk diuji.

Sekarang mari kita periksa dua pengukuran untuk rata-rata yang sama. Kami menolak hipotesis bahwa mereka memiliki rata-rata yang sama, jika hasilnya lt;0,05.

tcomparedata(normal(1,10),normal(1,10))

0.0374345857136

Jika kami menambahkan bias ke satu distribusi, kami mendapat lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

tcomparedata(normal(1,10),normal(1,10)+2)

0.017224091064

Dalam contoh berikutnya, kami menghasilkan 20 lemparan dadu acak 100 kali dan menghitungnya. Harus ada rata-rata 20/6=3,3.

R=random(100,20); R=sum(R6¡=1)'; mean(R)

3.09

Kami sekarang membandingkan jumlah satu dengan distribusi binomial. Pertama kita memplot distribusi satuan.

plot2d(R,distribution=max(R)+1,even=1,style="
/"):

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 adalah distribusi binomial, jika hasilnya lt;0,05.

chitest(t1,b1)

0.528664525902

Contoh berikut berisi hasil dari dua kelompok orang (pria dan wanita, katakanlah) 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

Kami ingin menguji independensi suara dari jenis kelamin. Tes tabel chi$^2 melakukanini.Hasilnyaterlalubes$

tabletest(A)
0.990701632326

Berikut 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

Beberapa Tes Lagi

Selanjutnya kami menggunakan analisis varians (F-test) untuk menguji tiga sampel data yang terdistribusi normal untuk nilai rata-rata yang sama. Metode tersebut dinamakan ANOVA (analysis of variance). 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

Ini berarti, kami menolak hipotesis dengan nilai rata-rata yang sama. Kami melakukan ini dengan probabilitas kesalahan 1,3

Ada juga uji median yang menolak sampel data dengan distribusi rata-rata yang berbeda menguji median sampel bersatu.

a=[56,66,68,49,61,53,45,58,54];
b=[72,81,51,73,69,78,59,67,65,71,68,71];
mediantest(a,b)
0.0241724220052

Tes lain tentang kesetaraan adalah tes peringkat. Ini jauh lebih tajam daripada tes median.

ranktest(a,b)
0.00199969612469

Dalam contoh berikut, kedua distribusi memiliki rata-rata yang sama.

ranktest(random(1,100),random(1,50)3-1)
0.121380642241

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, jika a lebih baik dari b.

signtest(a,b)

0.0546875

Ini terlalu banyak kesalahan. Kita tidak dapat menolak bahwa a sama baiknya dengan b.

Tes Wilcoxon lebih tajam dari tes ini, tetapi bergantung pada 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)

0.000340305226474

wilcoxon(normal(1,20),normal(1,20))

0.054213372246

Angka Acak

Berikut ini adalah tes untuk generator angka acak. Euler menggunakan generator yang sangat bagus, jadi kita tidak perlu berharap ada masalah.

Pertama kami menghasilkan sepuluh juta angka acak di [0,1].

n:=10000000; r:=random(1,n);

Selanjutnya kita menghitung jarak antara dua angka kurang dari 0,05.

a:=0.05; d:=differences(nonzeros(r¡a));

Akhirnya, 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;

Pengantar untuk Pengguna Proyek R

Jelas, 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. Lagi pula, EMT hadir dengan paket numerik dan sistem aljabar komputer.

Notebook ini cocok untuk Anda jika sudah familiar dengan R, namun perlu mengetahui perbedaan sintaks EMT dan R. Kami mencoba memberikan gambaran umum tentang hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami mencari cara untuk bertukar data antara kedua sistem.

Perhatikan bahwa ini adalah pekerjaan yang sedang berjalan.

Sintaks Dasar

Hal pertama yang Anda pelajari di R adalah membuat vektor. Di EMT, perbedaan utamanya adalah operator : dapat mengambil ukuran langkah. Apalagi daya ikatnya 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]

Fungsi c() tidak ada. Dimungkinkan untuk menggunakan vektor untuk menggabungkan berbagai hal.

Contoh berikut, seperti banyak lainnya, dari "Introduction to R" yang disertakan dengan proyek R. Jika Anda membaca PDF ini, Anda akan menemukan bahwa saya mengikuti jalannya dalam tutorial ini.

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 diganti dengan fungsi seq() di R. Kita bisa 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]

Fungsi rep() dari R tidak ada di EMT. Untuk input vektor, dapat ditulis sebagai berikut.

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]

Perhatikan bahwa "=" atau ":=" digunakan untuk tugas. Operator "-gt;" digunakan untuk unit di EMT.

125km - " miles"

77.6713990297 miles

The "lt;-" operator for assignment is misleading anyway, and not a good idea of R. The following will compare a and -4 in EMT.

a=2; a¡-4

0

Di R, "alt;-4lt;3" berfungsi, tetapi "alt;-4lt;-3" tidak. Saya juga memiliki ambiguitas serupa di EMT, tetapi mencoba menghilangkannya sedikit demi sedikit.

EMT dan R memiliki vektor tipe 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.

x¡5,

[0, 0, 1, 0, 0] [0, 0, 3.1, 0, 0]

EMT melempar kesalahan atau menghasilkan NAN tergantung pada bendera "kesalahan".

errors off; 0/0, isNAN(sqrt(-1)), errors on;

NAN 1

String sama di R dan EMT. Keduanya berada di lokal saat ini, bukan di Unicode.

Di R ada paket untuk Unicode. Di EMT, sebuah 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

Berikut ini mungkin atau mungkin tidak ditampilkan dengan benar di sistem Anda sebagai A dengan titik dan garis di atasnya. Itu tergantung pada font yang Anda gunakan.

chartoutf([480])

Penggabungan string dilakukan dengan "+" atau "—". Itu bisa termasuk angka, yang akan dicetak dalam format saat ini.

"pi = "+pi

pi = 3.14159265359

Pengindeksan

Sebagian besar waktu, ini akan berfungsi seperti di R.

Tetapi EMT akan menginterpretasikan indeks negatif dari belakang vektor, sedangkan R menginterpretasikan 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

Perilaku R dapat dicapai dalam EMT dengan drop().

drop(x,2)

[10.4, 3.1, 6.4, 21.7]

Vektor logis tidak diperlakukan berbeda sebagai indeks di EMT, berbeda dengan R. Anda perlu mengekstraksi 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]

Tetapi nama untuk indeks tidak dimungkinkan di EMT. Untuk paket statistik, hal ini sering diperlukan untuk memudahkan akses ke elemen vektor.

Untuk meniru perilaku ini, kita dapat mendefinisikan fungsi sebagai berikut.

function sel (v,i,s) := v[indexof(s,i)]; ... s=["first","second","third","fourth"]; sel(x,["first","third"],s)

Trying to overwrite protected function sel! Error in: function sel (v,i,s) := v[indexof(s,i)]; ... ...

Tipe Data

EMT memiliki lebih banyak tipe data tetap daripada R. Jelas, di R terdapat vektor yang tumbuh. Anda dapat menyetel vektor numerik kosong v dan menetapkan nilai ke elemen v[17]. Ini tidak mungkin di EMT.

Berikut ini agak tidak efisien.

v=[]; for i=1 to 10000; v=v—i; end;

EMT sekarang akan membuat vektor dengan v dan i ditambahkan pada tumpukan dan menyalin vektor itu kembali ke variabel global v.

Semakin efisien pra-mendefinisikan vektor.

v=zeros(10000); for i=1 to 10000; v[i]=i; end;

Untuk mengubah jenis tanggal di EMT, Anda dapat menggunakan fungsi seperti complex().

complex(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 ¡length(v) then s=s+","; endif; end; return s+"]"; endfunction ¡/pre¿ tostr(linspace(0,1,10))

[0.00,0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1.00]

Untuk komunikasi dengan Maxima, terdapat fungsi convertmxm(), yang juga dapat digunakan untuk memformat vektor untuk output.

convertmxm(1:10)

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

Untuk Lateks, perintah tex dapat digunakan untuk mendapatkan perintah Lateks.

tex([1,2,3])

$[1, 2, 3]$

Faktor dan Tabel

Dalam pengantar R ada contoh dengan apa yang disebut faktor.

Berikut ini adalah daftar wilayah dari 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”];

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, kami ingin menghitung rata-rata pendapatan di wilayah tersebut. Menjadi program statistik, R memiliki factor() dan tappy() untuk ini.

EMT dapat melakukannya dengan menemukan indeks wilayah di daftar unik wilayah.

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 saat itu, kita dapat menulis fungsi loop 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($x[nonzeros(f == indexof(u, i))]$); $endfunction <$ /$pre > Inisedikittidakefisien, karenamenghitungwilayahunikuntuksetiapi, tetapiberhasil.$

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 pada R. Fungsi readtable() dan writetable() dapat digunakan untuk input dan output.

Sehingga kita bisa mencetak rata-rata pendapatan negara di daerah dengan cara yang 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

Kami juga dapat mencoba meniru perilaku R sepenuhnya.

Faktor jelas harus disimpan dalam kumpulan dengan jenis dan kategori (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: tapply u=sort(unique(t)); return t,u,indexofsorted(u,t); endfunction ¡/pre¿ statef=makef(austates);

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]

Sekarang kita bisa meniru tapply() dengan cara berikut. Ini akan mengembalikan tabel sebagai kumpulan data tabel dan judul kolom.

function tapply (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*]); *endif*; *end*; *returnx*, *uf*; *endfunction* < */pre* >
Kamitidakmenambahkanbanyakpengecekantipedisini.Satu−*satunyatindakanpencegahanmenyangkutkateg*

Tabel ini dapat dicetak sebagai tabel dengan writetable().

writetable(tapply(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 pustaka 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

Ekstraksi baris dan kolom, atau sub-matriks, sangat mirip dengan R.

X[,2:3]

2 3 7 8 12 13 17 18

Namun, dalam R dimungkinkan untuk menetapkan daftar indeks spesifik vektor ke suatu nilai. Hal yang sama dimungkinkan 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¿ Kami mendemonstrasikan ini untuk menunjukkan bahwa matriks dilewatkan dengan referensi di EMT. Jika Anda tidak ingin mengubah matriks asli M, 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

Produk luar di EMT hanya dapat dilakukan di antara vektor. Ini otomatis karena bahasa matriks. Satu vektor harus berupa vektor kolom dan yang lainnya 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

Dalam pengantar PDF untuk R ada contoh, yang menghitung distribusi abcd untuk a,b,c,d dipilih dari 0 sampai n secara acak. Solusi dalam R adalah membentuk matriks 4 dimensi dan menjalankan table() di atasnya.

Tentu saja, ini bisa 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.

Tapi kami ingin meniru perilaku R. Untuk ini, kami 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 perkalian yang tepat, EMT dapat menghitung frekuensi dalam vektor.

```
getfrequencies(q,-50:10:50)
```

[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]

Cara paling mudah untuk memplot ini sebagai distribusi adalah sebagai berikut.

```
plot2d(q,distribution=11):
```

Tetapi juga memungkinkan untuk melakukan pra-perhitungan hitungan 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.

Jenis daftar yang tidak dapat diubah disebut koleksi di EMT. Ini berperilaku seperti struktur di C, tetapi elemennya hanya diberi nomor dan tidak diberi nama.

```
L="Fred","Flintstone",40,[1990,1992]
```

Fred Flintstone 40 [1990, 1992]

Saat ini elemen tidak memiliki nama, meskipun nama dapat diatur untuk tujuan khusus. Mereka diakses oleh nomor.

```
(L[4])[2]
```

1992

File Input dan Output (Membaca dan Menulis Data)

Anda sering ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberitahu Anda tentang banyak cara untuk mencapai hal ini. Fungsi sederhana adalah writematrix() dan readmatrix().

Mari kita tunjukkan cara membaca dan menulis vektor real ke file.

```
a=random(1,100); mean(a), dev(a),
```

0.47803 0.26642

Untuk menulis data ke file, kami menggunakan fungsi writematrix().

Karena pengantar 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, kami menggunakan readmatrix().

a=readmatrix(filename)';

Dan hapus file tersebut.

fileremove(filename);

mean(a), dev(a),

0.47803 0.26642

Fungsi writematrix() atau writetable() dapat dikonfigurasi untuk bahasa lain.

Misalnya, jika Anda memiliki sistem bahasa Indonesia (titik desimal dengan koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan koma). File berikut "test.csv" akan muncul di folder cuurent 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 kami memiliki string dengan token seperti berikut ini.

s1:="f m m f m m m f f f m m f"; ... s2:="f f f m m f f";

Untuk menandai 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 0.1666922653880168,0.01319339085437275,0.6365603562945641 0.3446099177430584,0.381469811510
0.7813294829861289,0.480004758946786,0.8449331856422688

Fungsi readtable() dalam bentuknya yang paling sederhana dapat membaca ini dan mengembalikan kumpulan nilai dan baris heading.

L=readtable(file, list);

Koleksi ini dapat dicetak dengan writetable() ke notebook, atau ke file.

writetable(L,wc=10,dc=5)

A B C 0.16669 0.01319 0.63656 0.34461 0.38147 0.39688 0.78133 0.48 0.84493

Matriks nilai adalah elemen pertama dari L. Perhatikan bahwa mean() dalam EMT menghitung nilai rata-rata dari baris matriks.

mean(L[1])

0.27215 0.37432 0.70209

File CSV

Pertama, mari kita menulis matriks ke dalam file. Untuk hasilnya, kami membuat file di direktori kerja saat ini.

file="test.csv"; ... M=random(3,3); writematrix(M,file);

Berikut adalah isi dari file ini.

printfile(file)

0.5958533886110121,0.8024372897144265,0.5760419801639245 0.5682326081673483,0.7367668331717983,0.2
0.4574738051208078,0.4793953284381902,0.3906056574552697

CVS ini dapat dibuka pada sistem bahasa Inggris ke dalam Excel dengan klik dua kali. Jika Anda mendapatkan file seperti itu di sistem Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal.

Tetapi titik desimal juga merupakan format default untuk EMT. Anda dapat membaca matriks dari file dengan readmatrix().

readmatrix(file)

0.33737 0.64037 0.31285 0.64093 0.80522 0.80441 0.25648 0.46055 0.31475

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 "simpan sebagai" dan "format lain", 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 Jerman saya menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubahnya di pengaturan sistem atau di Excel, tetapi tidak perlu membaca matriks ke dalam EMT.

Cara termudah untuk membaca ini ke 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

mari kita rencanakan ini.

plot2d(M'[1],M'[2:3], points,color=[red,green]'):

Ada cara yang lebih mendasar untuk membaca data dari file. Anda dapat membuka file dan membaca angka baris demi baris. Fungsi getvectorline() akan membaca angka dari baris data. Secara default, ini mengharapkan titik desimal. Tapi itu juga bisa menggunakan koma desimal, jika Anda memanggil setdecimaldot(",") sebelum Anda menggunakan fungsi ini.

Fungsi berikut adalah contoh untuk ini. 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$_v$; $else break$; $endif$; end; $return M$; $close(file)$; $endfunction$ < /pre > $myload(file)$

0.59585 0.80244 0.57604 0.56823 0.73677 0.28562 0.45747 0.4794 0.39061

Dimungkinkan juga untuk membaca semua angka dalam file itu dengan getvector().

open(file); v=getvector(10000); close(); redim(v[1:9],3,3)

0.59585 0.80244 0.57604 0.56823 0.73677 0.28562 0.45747 0.4794 0.39061

Thus it is very easy to save a vector of values, one value in each line and read back this vector.

v=random(1000); mean(v)

0.5019

writematrix(v',file); mean(readmatrix(file)')

0.5019

Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Sebagai contoh, kami menulis tabel dengan tajuk baris dan kolom ke file.

file="test.tab"; M=random(3,3); ... open(file,"w"); ... writetable(M,separator=",",labc=["one","two","
... close(); ... printfile(file)

one,two,three 0.21, 0.63, 0.06 0.23, 0.74, 0.43 0.77, 0.79, 0.61

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 0.21 0.63 0.06 0.23 0.74 0.43 0.77 0.79 0.61

Menganalisis Garis

Anda bahkan dapat mengevaluasi setiap baris dengan tangan. Misalkan, kita memiliki garis dengan format berikut.

line="2020-11-03,Tue,1'114.05"

2020-11-03,Tue,1'114.05

Pertama kita dapat menandai garis.

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],"","")()

7.3816e+05 2 1114

Menggunakan ekspresi reguler, dimungkinkan untuk mengekstraksi hampir semua informasi dari sebaris data.

Asumsikan kita memiliki baris berikut sebuah dokumen HTML.

line="¡tr ¡td 1145.45¡/td ¡td 5.6¡/td ¡td -4.5¡/td ¡tr "

lt;trgt;lt;tdgt;1145.45lt;/tdgt;lt;tdgt;5.6lt;/tdgt;lt;tdgt;-4.5lt;/tdgt;lt;trgt;

Untuk mengekstrak ini, kami menggunakan ekspresi reguler, yang mencari

- tanda kurung tutup gt;, - string apa pun yang tidak mengandung tanda kurung dengan

sub-pertandingan "(...)",

- braket pembuka dan penutup menggunakan solusi terpendek,

- sekali lagi string apa pun yang tidak mengandung tanda kurung,

- dan tanda kurung buka lt;.

Ekspresi reguler agak sulit dipelajari tetapi sangat kuat.

pos,s,vt=strxfind(line," ([<]+) < .+? ([<]+) < ");

Hasilnya adalah posisi kecocokan, string yang cocok, dan vektor string untuk sub-kecocokan.

for k=1:length(vt); vt[k](), end;

1145.5 5.6

Ini adalah fungsi yang membaca semua item numerik antara lt;tdgt; dan lt;/tdgt;.

function readtd (line) ...

v=[]; cp=0; repeat pos,s,vt=strxfind(line,"¡td.*?¿(.+?)¡/td¿",cp); until pos==0; if length(vt)¿0 then v=v—vt[1]; endif; cp=pos+strlen(s); end; return v; endfunction ¡/pre¿ readtd(line+"¡td non-numerical¡/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, kami membaca versi terkini dari situs EMT. Kami menggunakan ekspresi reguler untuk memindai "Versi ..." dalam judul.

function readversion () ...

urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html"); repeat until urleof(); s=urlgetline(); k=strfind(s,"Version ",1); if k¿0 then substring(s,k,strfind(s,"¡",k)-1), break; endif; end; urlclose(); endfunction ¡/pre¿ readversion

Version 2024-01-12

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="test.e"; ... writevar(random(2,2),"M",file); ... printfile(file,3)

M = [.. 0.535595099686451, 0.1780708758793885; 0.3125868367822287, 0.3438257576083715];

We can now load the file. It will define the matrix M.

load(file); show M,

M = 0.5356 0.17807 0.31259 0.34383

By the way, jika writevar() digunakan pada variabel, itu akan mencetak definisi variabel dengan nama variabel ini.

writevar(M); writevar(inch)

M = [.. 0.535595099686451, 0.1780708758793885; 0.3125868367822287, 0.3438257576083715]; inch= 0.0254;

Kami juga dapat membuka file baru atau menambahkan file yang sudah ada. Dalam contoh kami menambahkan file yang dihasilkan sebelumnya.

open(file,"a"); ... writevar(random(2,2),"M1"); ... writevar(random(3,1),"M2"); ... close();

load(file); show M1; show M2;

M1 = 0.50911 0.19151 0.99538 0.80856 M2 = 0.88957 0.42152 0.85292

Untuk menghapus file apa pun gunakan fileremove().

fileremove(file);

Vektor baris dalam file tidak memerlukan koma, jika setiap angka berada di baris baru. Mari kita buat file seperti itu, menulis 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 = [0.0892613504954 0.790049102695 0.074988207954 0.93537149276 0.618230259768];

load(file); M

[0.089261, 0.79005, 0.074988, 0.93537, 0.61823]