

Ayu Setyorini (24030130050)

Visualisasi dan Perhitungan Geometri dengan EMT

Euler menyediakan beberapa fungsi untuk melakukan visualisasi dan perhitungan geometri, baik secara numerik maupun analitik (seperti biasanya tentunya, menggunakan Maxima). Fungsi-fungsi untuk visualisasi dan perhitungan geometri tersebut disimpan di dalam file program "geometry.e", sehingga file tersebut harus dipanggil sebelum menggunakan fungsi-fungsi atau perintah-perintah untuk geometri.

Pendahuluan Geometri adalah cabang ilmu dari matematika yang

mempelajari hubungan antara titik, garis, sudut, bidang serta bangun datar dan bangun ruang. Konsep geometri bersifat abstrak namun bisa ditunjukkan dengan cara semi nyata atau semi kongkrit. Bangun geometri terdapat dua macam yaitu geometri bangun datar dan geometri bangun ruang. Geometri bangun datar merupakan suatu bentuk geometris yang terdiri dua dimensi atau hanya sekedar memiliki luas namun tidak memiliki volume contohnya seperti segiempat, lingkaran, segitiga, dan lain-lain.

Terdapat dua pendekatan berbeda dalam menyelesaikan masalah geometris, terutama yang berhubungan dengan komputasi. Geometri numerik berfokus pada penghitungan dan perkiraan berbasis angka, sementara geometri simbolik menggunakan logika dan manipulasi simbol untuk mencapai solusi yang eksak.

```
>load geometry
```

```
Numerical and symbolic geometry.
```

Funsgi-funsgi Geometri

Funsgi-funsgi 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 vektor v 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 $\angle ABC$
angleBisector(A, B, C): garis bagi sudut $\angle ABC$
circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r
getCircleCenter(c): pusat lingkaran c
getCircleRadius(c): jari-jari lingkaran c
circleThrough(A,B,C): lingkaran melalui A, B, C
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan lingkaran c
circleCircleIntersections (c1, c2): titik potong lingkaran c1 dan

c2

`planeThrough(A, B, C):` bidang melalui titik A, B, C

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

`getLineEquation (g,x,y):` persamaan garis g dinyatakan dalam x dan y
`getHesseForm (g,x,y,A):` bentuk Hesse garis g dinyatakan dalam x dan

y dengan titik A pada

sisi positif (kanan/atas) garis
`quad(A,B):` kuadrat jarak AB
`spread(a,b,c):` Spread segitiga dengan panjang sisi-sisi a,b,c, yakni

$\sin(\alpha)^2$ dengan

α sudut yang menghadap sisi a.
`crosslaw(a,b,c,sa):` persamaan 3 quads dan 1 spread pada segitiga

dengan panjang sisi a, b, c.

`triplespread(sa,sb,sc):` persamaan 3 spread sa,sb,sc yang memebntuk

suatu segitiga

`doublespread(sa): Spread sudut rangkap Spread  $2\phi$ , dengan`

`sa=sin(phi)^2 spread a.`

Contoh 1: Luas, Lingkaran Luar, Lingkaran Dalam Segitiga

1. Menggambar Objek Geometri

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.

a. Mendefinisikan Bidang Koordinat

```
>setPlotRange(-0.5,2.5,-0.5,2.5); //(batas sumbu x, batas sumbu y)
```

b. Mendefinisikan Titik Koordinat

```
>A=[1,0]; plotPoint(A,"A"); // definisi dan gambar titik  
>B=[0,1]; plotPoint(B,"B");  
>C=[2,2]; plotPoint(C,"C");
```

c. Menghubungkan Titik Sudut

```
>plotSegment(A,B,"c"); // c=AB  
>plotSegment(B,C,"a"); // a=BC  
>plotSegment(C,A,"b"); // b=AC
```

2. Persamaan Garis

Fungsi geometri mencakup fungsi untuk membuat garis dan lingkaran. Format garisnya adalah [a,b,c], yang merepresentasikan garis dengan persamaan $ax+by=c$.

```
>lineThrough(B,C) // persamaan garis yang melalui B dan C
```

```
[-1, 2, 2]
```

```
maxima: b*x+a*y=c  
maxima: -1*x+2*y=2  
maxima: 2*y=x+2  
maxima: y=1/2*x+1
```

3. Garis Tegak

Mencari garis tegak lurus melalui A pada BC.

a. Mendefinisikan garis tegak

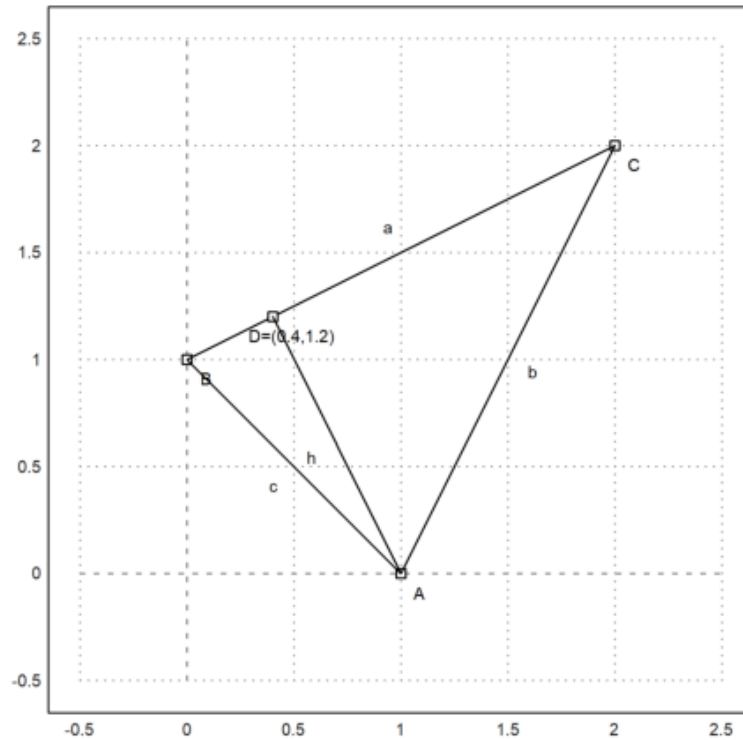
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

b. Titik Potong

```
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
```

c. Menggambar Garis Tegak

```
>plotPoint(D,value=1); // koordinat D ditampilkan  
>aspect(1); plotSegment(A,D,"h"): // tampilkan semua gambar hasil plot...()
```



4. Menghitung Luas

maxima: $L_{ABC} = \frac{1}{2} \cdot a \cdot h$

```
>norm(A-D)*norm(B-C)/2 // AD=norm(A-D), BC=norm(B-C)
```

1.5

Perandingkan dengan rumus lain

a. Langsung menggunakan fungsi

```
>areaTriangle(A,B,C)
```

1.5

Cara lain menghitung luas segitigas ABC:

b. Menggunakan fungsi distance

```
>distance(A,D)*distance(B,C)/2
```

1.5

5. Mencari Besar Sudut

a. Sudut C

```
>degprint(computeAngle(B,C,A))
```

36°52'11.63''

b. Sudut B

```
>degprint(computeAngle(A,B,C))
```

71°33'54.18''

c. Sudut A

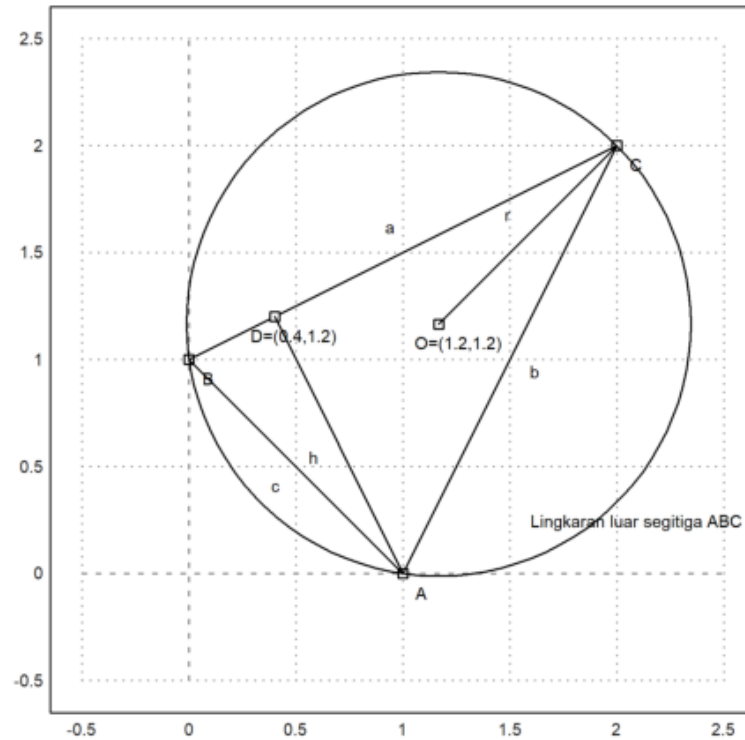
```
>degprint(computeAngle(C,A,B))
```

71°33'54.18''

6. Lingkaran Luar

a. Mendefinisikan lingkaran

```
>c=circleThrough(A,B,C); // definisi lingkaran luar segitiga ABC  
>R=getCircleRadius(c); // definisi jari2 lingkaran luar  
>O=getCircleCenter(c); // definisi titik pusat lingkaran c  
>plotPoint(O,value=1); // gambar titik "O"  
>plotSegment(O,C,"r"); // gambar jari-jari  
>plotCircle(c,"Lingkaran luar segitiga ABC");
```



b. Mencari titik pusat

>0

[1.16667, 1.16667]

c. Mencari jari-jari

```
>R
```

```
1.17851130198
```

```
>
```

7. Lingkaran Dalam

Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

a. Mendefinisikan garis bagi sudut

```
>l=angleBisector(A,C,B); // garis bagi <ACB  
>g=angleBisector(C,A,B); // garis bagi <CAB
```

b. Mencari titik potong garis bagi

```
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

```
[0.86038, 0.86038]
```

```
>plotPoint(P,"P"); // gambar titik potongnya
```

c. Gambar garis bagi

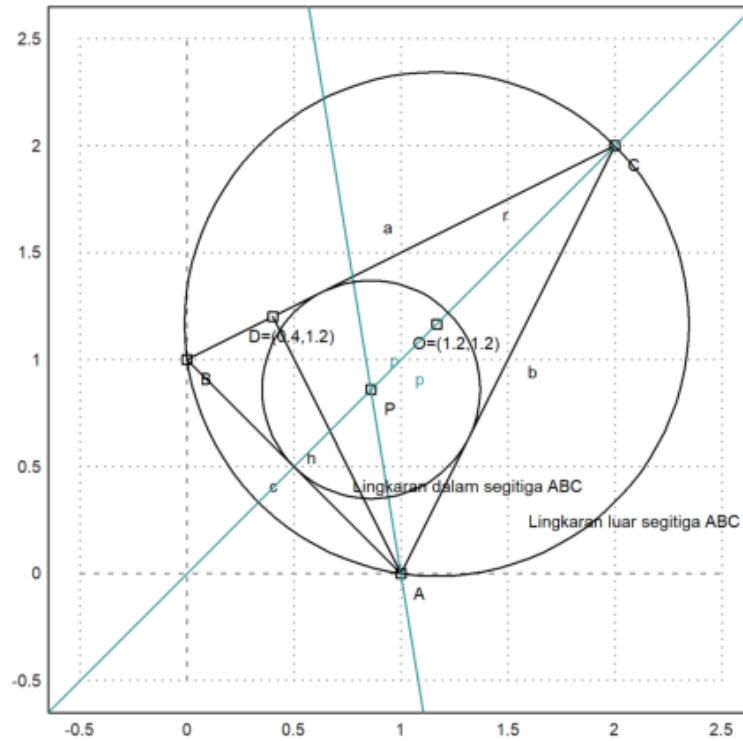
```
>color(5); plotLine(l); plotLine(g); color(1);
```

d. Mencari jari-jari

```
>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
```



Contoh 2: Geometri Simbolik

1. Pengertian Geometri Simbolik

Geometri simbolik merupakan cabang dari geometri yang mempelajari hubungan antara unsur-unsur geometri, seperti titik, garis, bidang, sudut dalam bentuk simbol atau notasi matematis.

2. Unsur-Unsur Geometri dalam Bentuk Simbolik

1). Titik

Unsur titik disimbolkan dengan huruf kapital

Mendefinisikan titik dalam bentuk koordinat pada bidang dua dimensi

```
>A &= [1,0]; B &= [0,1]; C &= [2,2]; // menentukan tiga titik A, B, C
```

Titik A dengan koordinat A(1,0)

Titik B dengan koordinat B(0,1)

Titik C dengan koordinat C(2,2)

2). Garis

Unsur garis disimbolkan dengan huruf kecil

Menentukan persamaan garis yang melalui dua titik

```
>c &= lineThrough(B,C) // c=BC
```

[- 1, 2, 2]

Persamaan garis c yang melalui dua titik, yaitu titik B dan titik C.

Output yang dihasilkan merupakan koefisien dari persamaan garis dalam bentuk umum:

$$Ax + By + C = 0$$

$$-x + 2y + 2 = 0$$

```
>$getLineEquation(c,x,y), $solve(%,y) | expand // persamaan garis c
```

$$2y - x = 2$$

$$\left[y = \frac{x}{2} + 1 \right]$$

Output yang dihasilkan merupakan koefisien dari persamaan garis dalam bentuk eksplisit:

$$y = mx + n$$

ubah dari bentuk umum ke bentuk eksplisit:

$$-x + 2y + 2 = 0$$

$$2y - x = 2$$

$$y = \frac{x}{2} + 1$$

```
>$getLineEquation(lineThrough([x1,y1],[x2,y2]),x,y), $solve(%,y) // persamaan garis melalui(x1, y1)
```

$$x (y_1 - y_2) + (x_2 - x_1) y = x_1 (y_1 - y_2) + (x_2 - x_1) y_1$$

$$\left[y = \frac{-(x_1 - x) y_2 - (x - x_2) y_1}{x_2 - x_1} \right]$$

Menentukan persamaan garis melalui satu titik dengan kemiringan tertentu

```
>$getLineEquation(lineThrough(A,[x1,y1]),x,y) // persamaan garis melalui A dan (x1, y1)
```

$$(x_1 - 1) y - x y_1 = -y_1$$

Menentukan dan mendefinisikan garis yang melalui suatu titik dan tegak lurus terhadap suatu garis

```
>h &= perpendicular(A,lineThrough(B,C)) // h melalui A tegak lurus BC
```

$$[2, 1, 2]$$

Output yang dihasilkan merupakan koefisien dari persamaan garis dalam bentuk umum:

$$Ax + By + C = 0$$

$$2x + y + 2 = 0$$

```
>Q &= lineIntersection(c,h) // Q titik potong garis c=BC dan h
```

$$\begin{bmatrix} 2 & 6 \\ - & - \\ 5 & 5 \end{bmatrix}$$

Output yang dihasilkan merupakan titik kaki tegak lurus dari A ke garis BC

```
>$projectToLine(A,lineThrough(B,C)) // proyeksi A pada BC
```

$$\begin{bmatrix} \frac{2}{5} & \frac{6}{5} \\ \frac{2}{5} & \frac{6}{5} \end{bmatrix}$$

```
>$distance(A,Q) // jarak AQ
```

$$\frac{3}{\sqrt{5}}$$

```
>cc &= circleThrough(A,B,C); $cc // (titik pusat dan jari-jari) lingkaran melalui A, B, C
```

$$\left[\frac{7}{6}, \frac{7}{6}, \frac{5}{3\sqrt{2}}\right]$$

```
>r&=getCircleRadius(cc); $r , $float(r) // tampilkan nilai jari-jari
```

$$\frac{5}{3\sqrt{2}}$$

1.178511301977579

```
>$computeAngle(A,C,B) // nilai <ACB
```

$$\arccos\left(\frac{4}{5}\right)$$

```
>$solve(getLineEquation(angleBisector(A,C,B),x,y),y)[1] // persamaan garis bagi <ACB
```

$$y = x$$

```
>P &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A)); $P // titik potong 2 garis bagi s
```

$$\left[\frac{\sqrt{2}\sqrt{5}+2}{6}, \frac{\sqrt{2}\sqrt{5}+2}{6} \right]$$

```
>P() // hasilnya sama dengan perhitungan sebelumnya
```

```
[0.86038, 0.86038]
```

Garis dan Lingkaran yang Berpotongan

Definisi Garis dan Lingkaran

Garis adalah serangkaian titik-titik tak hingga. Persamaan garis dalam bentuk umum yaitu:

$$ax + by + c = 0$$

dengan a, b, c sebagai koefisien dengan setidaknya salah satu dari a dan b tidak boleh nol.

Lingkaran adalah himpunan titik-titik yang berjarak sama terhadap suatu titik tertentu. Jarak yang sama tersebut disebut panjang jari-jari lingkaran dan titik tertentu disebut pusat lingkaran. Persamaan umum lingkaran yaitu:

$$(x - a)^2 + (y - b)^2 = r^2$$

1. Mendefinisikan lingkaran dengan titik pusat dan jari jari r

```
>A &:= [1,0]; c=circleWithCenter(A,4);
```

2. Mendefinisikan garis yang melalui dua titik

```
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
```

3. Menggambar lingkaran dan garis

```
>setPlotRange(5); plotCircle(c); plotLine(l);
```

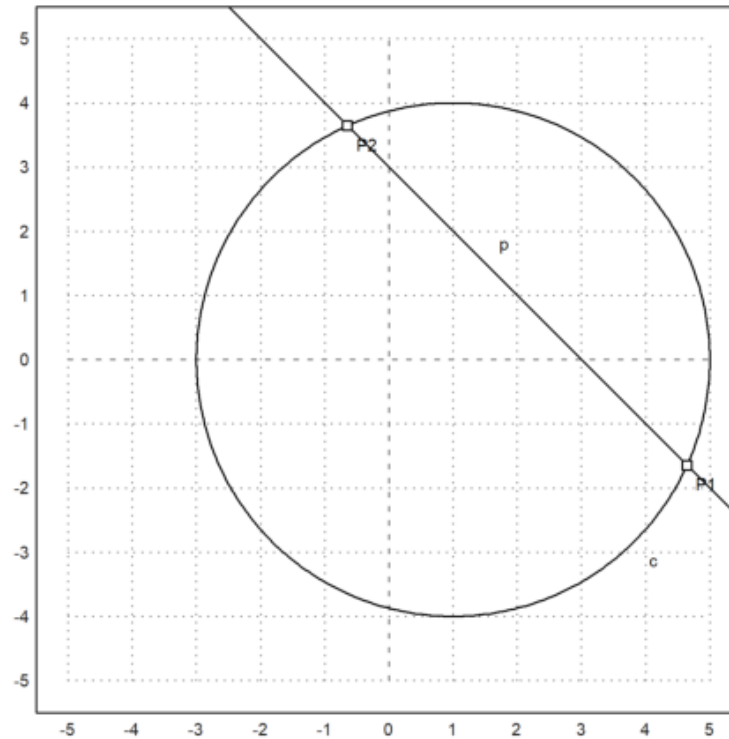
Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik perpotongan.

4. Menentukan titik potong antara garis dan lingkaran

```
>{P1,P2,f}=lineCircleIntersections(l,c);  
>P1, P2, f
```

```
[4.64575,  -1.64575]  
[-0.645751,  3.64575]  
2
```

```
>plotPoint(P1); plotPoint(P2):
```



Lakukan hal yang sama menggunakan maxima

1. Mendefinisikan lingkaran dengan titik pusat dan jari-jari r

```
>c = circleWithCenter(A,4) // lingkaran dengan pusat A jari-jari 4
```

[1, 0, 4]

Persamaan lingkaran:

$$(x - a)^2 + (y - b)^2 = r^2$$

Dengan titik pusat di A(1,0) dan jari-jari 4

$$(x - 1)^2 + y^2 = 4^2$$

$$(x - 1)^2 + y^2 = 16$$

2. Mendefinisikan garis yang melalui dua titik

```
>l &= lineThrough(B,C) // garis l melalui B dan C
```

[1, 1, 3]

Persamaan garis:

$$Ax + By + C = 0$$

$$x + y - 3 = 0$$

3. Mencari titik potong antara garis dan lingkaran

```
>$lineCircleIntersections(l,c) | radcan, // titik potong lingkaran c dan garis l
```

$$\left[\left[\sqrt{7} + 2, 1 - \sqrt{7} \right], \left[2 - \sqrt{7}, \sqrt{7} + 1 \right] \right]$$

Akan ditunjukkan bahwa sudut-sudut yang menghadap busur yang sama adalah sama besar.

4. Menghitung sudut

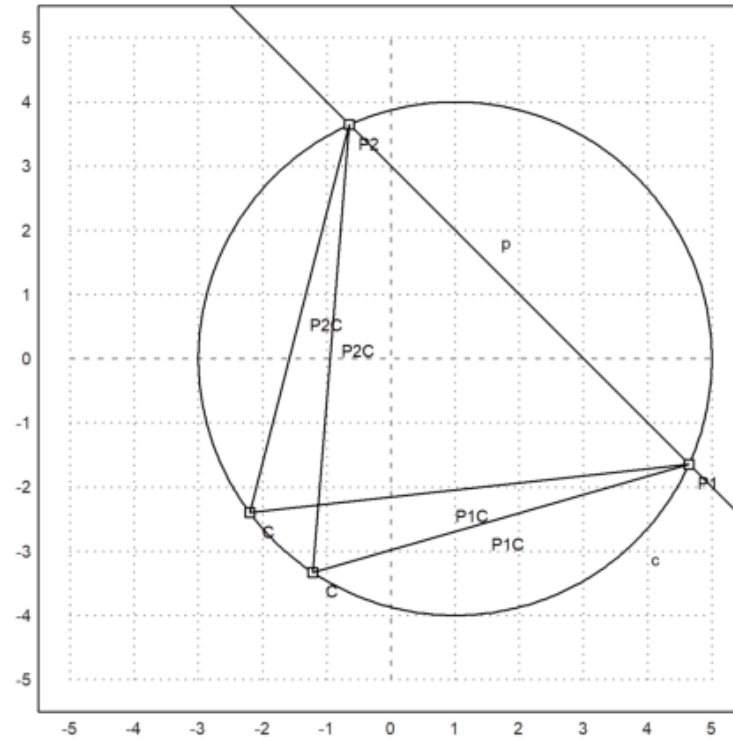
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>degprint(computeAngle(P1,C,P2))
```

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 adalah garis yang merupakan tempat kedudukna titik-titik yang berjarak sama dari dua titik tertentu.

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.
1. Mendefinisikan dua titik

```
>A=[2,2]; B=[-1,-2];
```

2. Mendefinisikan lingkaran 1

```
>c1=circleWithCenter(A,distance(A,B));
```

3. Mendefinisikan lingkaran 2

```
>c2=circleWithCenter(B,distance(A,B));
```

4. Mendefinisikan titik potong kedua lingkaran

```
>{P1,P2,f}=circleCircleIntersections(c1,c2);
```

5. Mendefinisikan garis sumbu tegak lurus

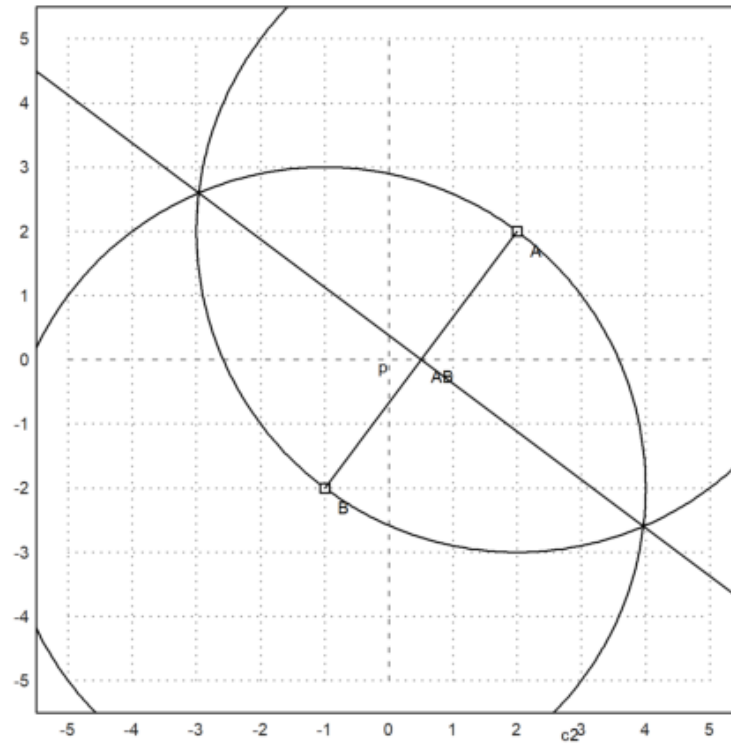
```
>l=lineThrough(P1,P2);
```

6. Menggambar dua lingkaran pada bidang koordinat

```
>setPlotRange(5); plotCircle(c1); plotCircle(c2);
```

7. Menggambar ruas garis perpotongan kedua lingkaran dan garis sumbu tegak lurus

```
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):
```



Selanjutnya, lakukan hal yang sama menggunakan maxima dengan koordinat umum

1. Mendefinisikan dua titik umum

```
>A &= [a1,a2]; B &= [b1,b2];
```

2. Mendefinisikan lingkaran 1

```
>c1 &= circleWithCenter(A,distance(A,B));
```

3. Mendefinisikan lingkaran 2

```
>c2 &= circleWithCenter(B,distance(A,B));
```

4. Menghitung dua titik potong antara kedua lingkaran

```
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];
```

The equations for the intersections are quite involved. But we can simplify, if we solve for y.

5. Mendefinisikan persamaan garis yang melalui dua titik potong

```
>g &= getLineEquation(lineThrough(P1,P2),x,y);  
>$solve(g,y)
```

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

6. Menghitung garis sumbu tegak lurus

```
>$solve(getLineEquation(middlePerpendicular(A,B),x,y),y)
```

$$\left[y = \frac{-(2b_1 - 2a_1)x + b_2^2 + b_1^2 - a_2^2 - a_1^2}{2b_2 - 2a_2} \right]$$

Hasil yang diperoleh saat menghitung garis sumbu tegak lurus AB sama dengan garis yang melalui titik potong dua lingkaran.

Garis yang melalui titik potong dua lingkaran = garis sumbu tegak lurus dari AB

```
>h &=getLineEquation(lineThrough(A,B),x,y);  
>$solve(h,y)
```

$$\left[y = \frac{(b_2 - a_2)x - a_1b_2 + a_2b_1}{b_1 - a_1} \right]$$

Merupakan bentuk umum dari garis yang melalui dua titik A dan B

Perhatikan hasil kali gradien garis g dan h adalah:

$$\frac{-(b_1 - a_1)}{(b_2 - a_2)} \times \frac{(b_2 - a_2)}{(b_1 - a_1)} = -1.$$

Artinya kedua garis tegak lurus.

Contoh 3: Rumus Heron

Rumus Heron menyatakan bahwa luas segitiga dengan panjang sisi-sisi a , b dan c adalah:

$$L = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{dengan } s = (a+b+c)/2,$$

atau bisa ditulis dalam bentuk lain:

$$L = \frac{1}{4} \sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}$$

Untuk membuktikan hal ini kita misalkan $C(0,0)$, $B(a,0)$ dan $A(x,y)$, $b=AC$, $c=AB$. Luas segitiga ABC adalah

$$L_{\triangle ABC} = \frac{1}{2}a \times y.$$

Nilai y didapat dengan menyelesaikan sistem persamaan:

$$x^2 + y^2 = b^2, \quad (x - a)^2 + y^2 = c^2.$$

```
>&assume(a>0);  
>sol &= solve([x^2+y^2=b^2,(x-a)^2+y^2=c^2],[x,y])
```

[]

```
>ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2))
```

Maxima said:

part: invalid index of list or matrix.

-- an error. To debug this try: debugmode(true);

Error in:

```
ysol &= y with sol[2][2]; $'y=sqrt(factor(ysol^2)) ...  
^
```

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
```

$$H(a,b,[1,0,4]) = \frac{a |ysol|}{2}$$

Contoh 1

Mencari luas segitiga dengan panjang sisi $a=3$, $b=4$, dan $c=5$ menggunakan rumus heron

```
>H(3,4,5)
```

```
[1.57143, 0.285714]
```

```
>//Kurva luas segitiga sengan panjang sisi 3, 4, x (1<= x <=7)  
>aspect (1.5); plot2d(&H(3,4,x),1,7):
```

```
Variable or function ysol not found.  
Error in expression: 3*abs(ysol)/2  
%ploteval:  
    y0=f$(x[1],args());  
adaptiveevalone:  
    s=%ploteval(g$,t,args());  
Try "trace errors" to inspect local variables after errors.  
plot2d:  
    dw/n,dw/n^2,dw/n,auto,args());
```

Contoh 2

Mencari luas segitiga dengan panjang sisi $a=2$, $b=5$, dan $c=6$ menggunakan rumus heron

```
>H(2,5,6)
```

[1.57143, 0.285714]

```
>//Kurva luas segitiga sengan panjang sisi 2, 5, x (3<= x <=7)  
>aspect(1.5); plot2d(&H(2,5,x),3,7):
```

Variable or function ysol not found.

Error in expression: abs(ysol)

%ploteval:

y0=f\$(x[1],args());

adaptiveevalone:

s=%ploteval(g\$,t,args());

Try "trace errors" to inspect local variables after errors.

plot2d:

dw/n,dw/n^2,dw/n,auto,args());

Contoh 4: Garis Euler dan Parabola

DEFINISI GARIS EULER

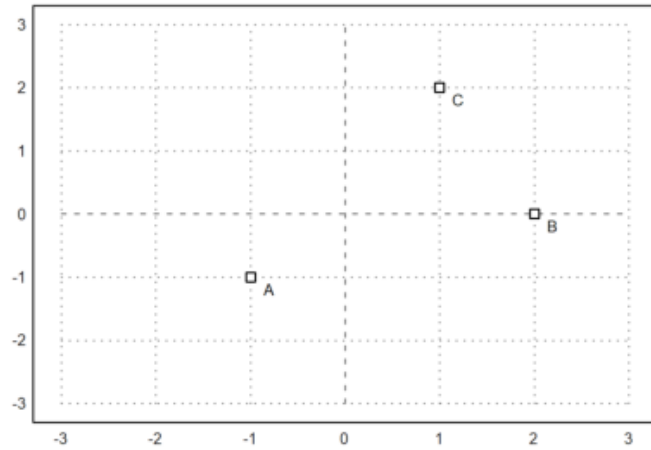
Garis Euler adalah garis yang ditentukan dari segitiga apa pun yang bukan sama sisi. Garis ini merupakan garis pusat segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga tersebut, termasuk orthocenter, circumcenter, centroid, titik Exeter, dan pusat lingkaran sembilan titik pada segitiga tersebut.

1. Mendefinisikan titik-titik segitiga

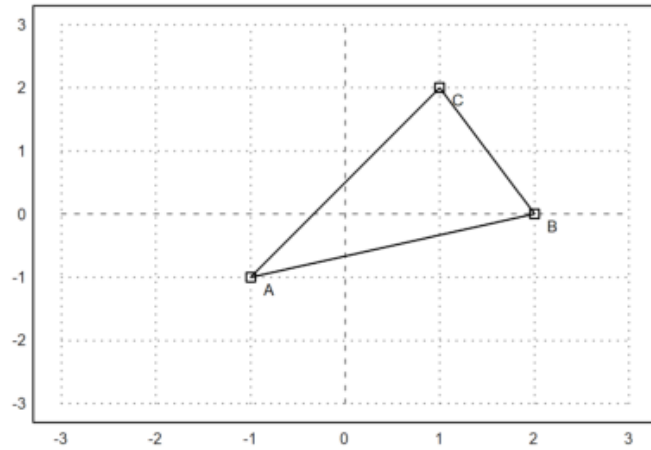
```
>A:=[-1,-1]; B:=[2,0]; C:=[1,2];
```

2. Menampilkan titik dan sisi segitiga

```
>setPlotRange(3);  
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C");
```



```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,A,""):
```



3. Menghitung Luas Segitiga

```
>$areaTriangle(A,B,C)
```

$$-\frac{7}{2}$$

4. Menentukan lingkaran luar segitiga

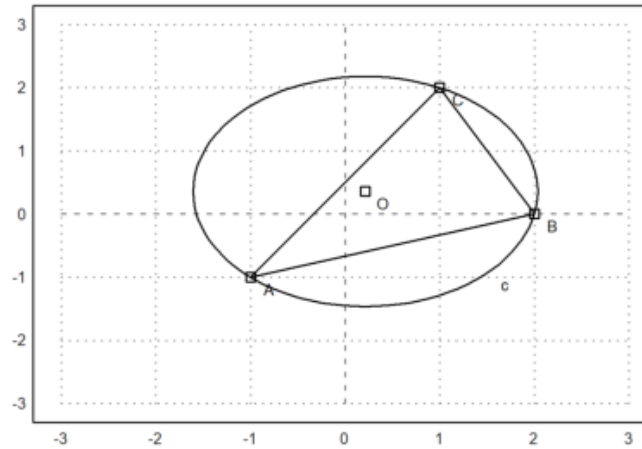
```
>//lingkaran yang melalui A, B, C  
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

$$\left(y - \frac{5}{14}\right)^2 + \left(x - \frac{3}{14}\right)^2 = \frac{325}{98}$$

```
>//titik pusat lingkaran luar  
>O &= getCircleCenter(LL); $O
```

$$\left[\frac{3}{14}, \frac{5}{14}\right]$$

```
>plotCircle(LL()); plotPoint(O(),"O"):
```



5. Menentukan Orthocenter (H)

Orthocenter adalah titik potong antara dua garis tinggi:

```
>H &= lineIntersection(perpendicular(A,lineThrough(C,B)),...
>perpendicular(B,lineThrough(A,C))); $H
```

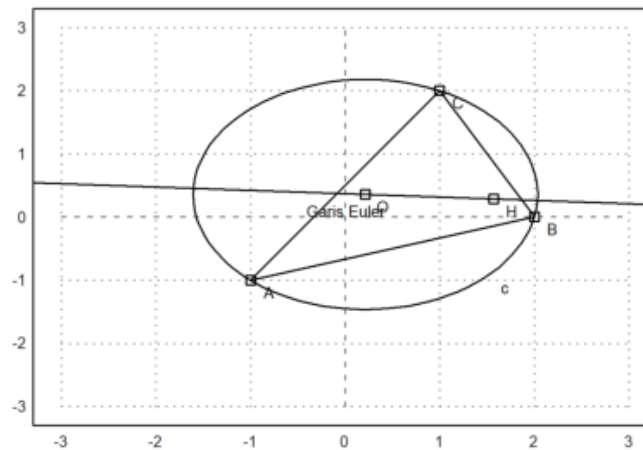
$$\left[\frac{11}{7}, \frac{2}{7} \right]$$

6. Menentukan Garis Euler

```
>el &= lineThrough(H,O); $getLineEquation(el,x,y)
```

$$-\frac{19y}{14} - \frac{x}{14} = -\frac{1}{2}$$

```
>plotPoint(H(),"H"); plotLine(el(),"Garis Euler"):
```

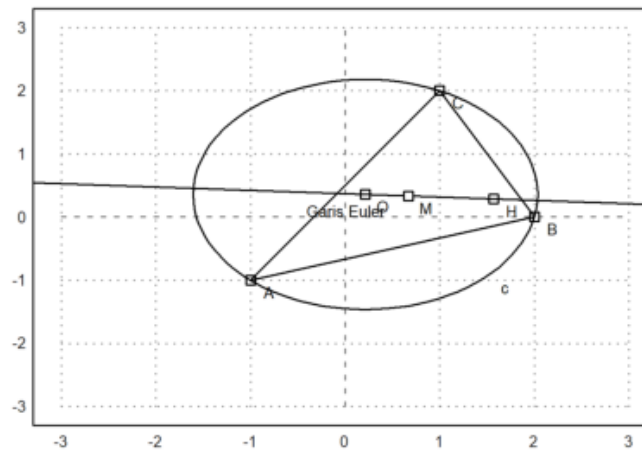


7. Menentukan Titik Berat Segitiga

```
>M &= (A+B+C)/3; $getLineEquation(e1,x,y) with [x=M[1],y=M[2]]
```

$$-\frac{1}{2} = -\frac{1}{2}$$

```
>plotPoint(M(),"M"): // titik berat
```



8. Membuktikan $MH = 2MO$

```
>$distance(M,H)/distance(M,O)|radcan
```

2

9. Membuat lingkaran dalam segitiga

```
>$computeAngle(A,C,B), degprint(%())
```

$$\arccos\left(\frac{4}{\sqrt{5}\sqrt{13}}\right)$$

$60^{\circ}15'18.43''$

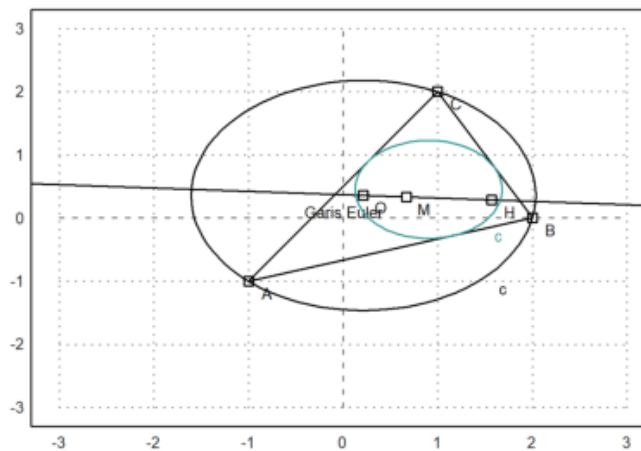
```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $Q
```

$$\left[\frac{\left(2^{\frac{3}{2}} + 1\right) \sqrt{5} \sqrt{13} - 15 \sqrt{2} + 3}{14}, \frac{(\sqrt{2} - 3) \sqrt{5} \sqrt{13} + 5 \cdot 2^{\frac{3}{2}} + 5}{14} \right]$$

```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

$$\frac{\sqrt{(-41\sqrt{2}-31)\sqrt{5}\sqrt{13}+115\sqrt{2}+614}}{7\sqrt{2}}$$

```
>LD &= circleWithCenter(Q,r); // Lingkaran dalam
>color(5); plotCircle(LD());
```



Parabola

1. Menentukan titik dan garis

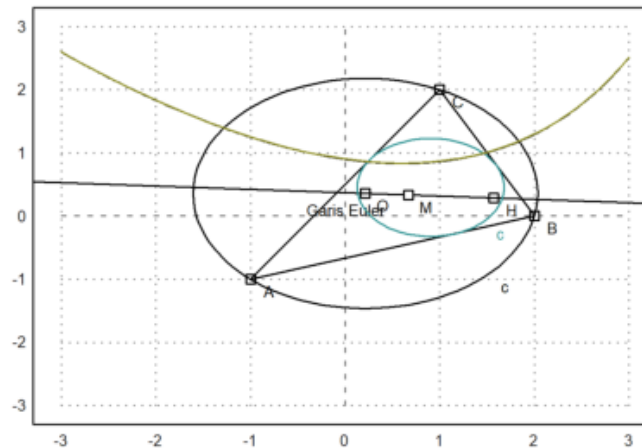
Mencari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>p &= getHesseForm(lineThrough(A,B),x,y,C)-distance([x,y],C); $p='0
```

$$\frac{3y - x + 2}{\sqrt{10}} - \sqrt{(2 - y)^2 + (1 - x)^2} = 0$$

2. Membuat grafik parabola

```
>plot2d(p,level=0,add=1,contourcolor=6):
```



3. Mendapatkan bentuk eksplisit parabola

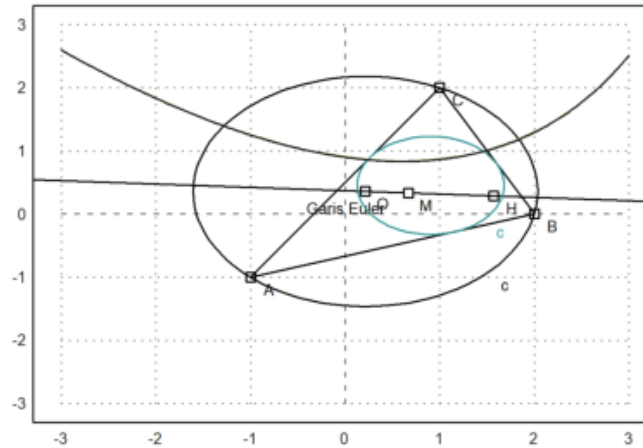
Dikarenakan Maxima kesulitan dengan akar, maka persamaan dikuadratkan:

```
>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)^2-distance([x,y],C)^2,y)
```

$$\begin{aligned} [y = -3x - \sqrt{70} \sqrt{9 - 2x} + 26, \\ y = -3x + \sqrt{70} \sqrt{9 - 2x} + 26] \end{aligned}$$

4. Plot Fungsi Parabola

```
>plot2d(&rhs(akar[1]),add=1):
```



```
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan kurva di atas
```

$$g(x) = -3x - \sqrt{70}\sqrt{9-2x} + 26$$

```
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

$$2.135605779339061$$

```
>U &= projectToLine(T,lineThrough(A,B)); $U // proyeksi T pada garis AB
```

$$\left[\frac{80 - 3\sqrt{11}\sqrt{70}}{10}, \frac{20 - \sqrt{11}\sqrt{70}}{10} \right]$$

```
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jarak T ke AB
```

$$\sqrt{1503 - 54\sqrt{11}\sqrt{70}}$$

2.135605779339061

Ternyata jarak T ke C sama dengan jarak T ke AB. Coba Anda pilih titik T yang lain dan ulangi perhitungan-perhitungan di atas untuk menunjukkan bahwa hasilnya juga sama.

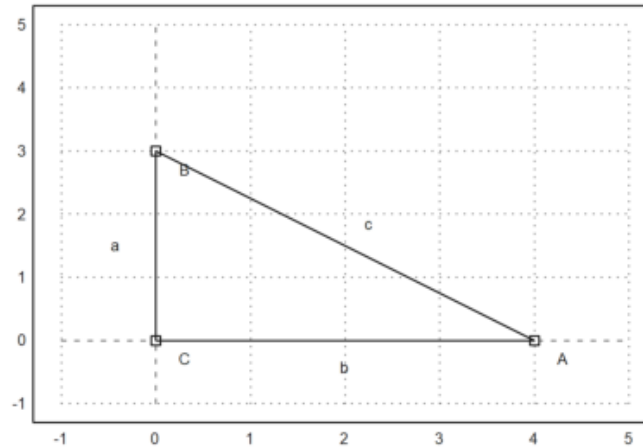
Contoh 5: Trigonometri Rasional

Ini terinspirasi dari presentasi N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti konsep klasik jarak dan sudut dengan quadrance dan spread. Dengan menggunakan konsep ini, memang mungkin untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

Dalam pembahasan berikut, saya memperkenalkan konsep-konsep tersebut dan menyelesaikan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keunggulan utama trigonometri rasional bahwa perhitungan dapat dilakukan hanya dengan kertas dan pensil. Anda diundang untuk memeriksa hasilnya tanpa menggunakan komputer.

Intinya, perhitungan simbolik rasional sering menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya dievaluasi sebagai perkiraan numerik.

```
>load geometry;  
>C&:=[0,0]; A&:=[4,0]; B&:=[0,3]; ...  
>setPlotRange(-1,5,-1,5); ...  
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...  
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...  
>insimg(30);
```



Tentu saja,

$$\sin(w_a) = \frac{a}{c},$$

di mana w_a adalah sudut di titik A. Cara biasa untuk menghitung sudut ini adalah dengan mengambil invers fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicetak secara tepat, sehingga hanya dapat dicetak secara aproksimatif.

```
>wa := arcsin(3/5); degprint(wa)
```

36°52'11.63''

Trigonometri rasional berusaha menghindari hal ini.

Konsep pertama dalam trigonometri rasional adalah kuadran, yang menggantikan jarak. Sebenarnya, kuadran hanyalah kuadrat dari jarak. Dalam pembahasan berikut, a , b , dan c mewakili kuadran dari sisi-sisi.

Teorema Pythagoras menjadi $a^2 + b^2 = c^2$.

$$a^2 = 3^2; b^2 = 4^2; c^2 = 5^2; a^2 + b^2 = c^2$$

$$25 = 25$$

Konsep kedua dari trigonometri rasional adalah selisih. Selisih mengukur jarak antara dua garis. Nilainya 0 jika garis-garis tersebut sejajar, dan 1 jika garis-garis tersebut tegak lurus. Selisih ini merupakan kuadrat dari sinus sudut antara dua garis tersebut.

Spread garis AB dan AC pada gambar di atas didefinisikan sebagai

$$s_a = \sin(\alpha)^2 = \frac{a}{c},$$

di mana a dan c adalah sisi-sisi segitiga siku-siku dengan salah satu sudutnya di A.

$$s_a = a/c; s_a$$

$$\frac{9}{25}$$

Tentu saja, ini lebih mudah dihitung daripada sudut. Namun, Anda kehilangan sifat bahwa sudut dapat ditambahkan dengan mudah.

Tentu saja, kita dapat mengubah nilai perkiraan sudut w_a menjadi sprad, dan mencetaknya sebagai pecahan.

```
>fracprint(sin(wa)^2)
```

9/25

Hukum kosinus dalam trigonometri klasik dapat diartikan sebagai "hukum silang" berikut.

$$(c + b - a)^2 = 4bc(1 - s_a)$$

Di sini, a , b , dan c adalah kuadran sisi-sisi segitiga, dan s_a adalah sudut di sudut A . Sisi a , seperti biasa, berhadapan dengan sudut A .

Hukum-hukum ini diimplementasikan dalam berkas `geometry.e` yang telah dimuat ke dalam Euler.

```
>$crosslaw(aa,bb,cc,saa)
```

$$\left[\left(bb - aa + \frac{7}{6} \right)^2, \left(bb - aa + \frac{7}{6} \right)^2, \left(bb - aa + \frac{5}{3\sqrt{2}} \right)^2 \right] = \left[\frac{14\,bb\,(1 - saa)}{3}, \frac{14\,bb\,(1 - saa)}{3}, \frac{5\,2^{\frac{3}{2}}\,bb\,(1 - saa)}{3} \right]$$

Dalam kasus kami, kami mendapatkan

```
>$crosslaw(a,b,c,sa)
```

$$1024 = 1024$$

Mari kita gunakan hukum silang ini untuk menemukan selisih di A. Untuk melakukannya, kita menghitung hukum silang untuk kuadran a, b, dan c, lalu menyelesaikan persamaan tersebut untuk selisih yang tidak diketahui sa.

Anda dapat melakukannya secara manual dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, hasil yang kita dapatkan sama dengan yang sudah kita miliki sebelumnya.

```
>$crosslaw(a,b,c,x), $solve(%,x)
```

$$1024 = 1600 (1 - x)$$

$$\left[x = \frac{9}{25} \right]$$

Kita sudah tahu hal ini. Definisi selisih sudut merupakan kasus khusus dari hukum silang.

Kita juga dapat menyelesaikan ini untuk a, b, c yang umum. Hasilnya adalah rumus yang menghitung selisih sudut segitiga berdasarkan kuadran ketiga sisinya.

```
>$solve(crosslaw(aa,bb,cc,x),x)
```

$$\left[\left[\frac{168bbx + 36bb^2 + (-72aa - 84)bb + 36aa^2 - 84aa + 49}{36}, \frac{168bbx + 36bb^2 + (-72aa - 84)bb + 36aa^2 - 84aa + 49}{36} \right] \right]$$

Kita dapat membuat fungsi dari hasil tersebut. Fungsi semacam itu sudah didefinisikan dalam berkas geometry.e dari Euler.

```
>$spread(a,b,c)
```

$$\frac{9}{25}$$

Sebagai contoh, kita dapat menggunakannya untuk menghitung sudut segitiga dengan sisi-sisi

$$a, \quad a, \quad \frac{4a}{7}$$

Hasilnya rasional, yang tidak begitu mudah didapatkan jika kita menggunakan trigonometri klasik.

```
>$spread(a,a,4*a/7)
```

$$\frac{6}{7}$$

Ini adalah sudut dalam derajat.

```
>degprint(arcsin(sqrt(6/7)))
```

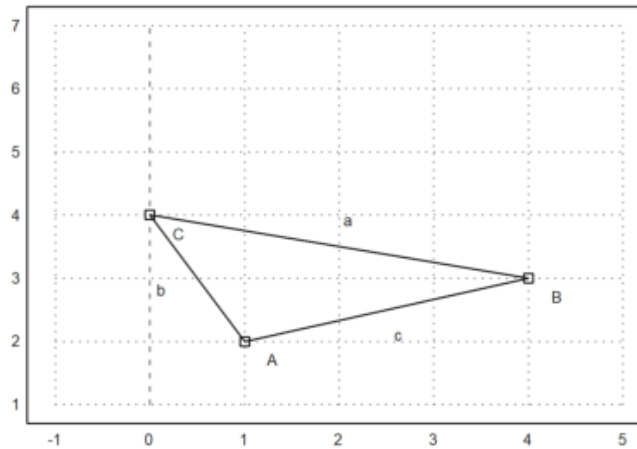
67°47'32.44''

Contoh Lainnya

Sekarang, mari kita coba contoh yang lebih lanjut.

Kita tetapkan tiga sudut segitiga sebagai berikut.

```
>A&:=[1,2]; B&:=[4,3]; C&:=[0,4]; ...  
>setPlotRange(-1,5,1,7); ...  
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...  
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...  
>insimg;
```



Dengan menggunakan Pythagoras, mudah untuk menghitung jarak antara dua titik. Pertama, saya menggunakan fungsi distance dari berkas Euler untuk geometri. Fungsi distance menggunakan geometri klasik.

```
>$distance(A,B)
```

$$\sqrt{10}$$

Euler juga mengandung fungsi untuk kuadran antara dua titik.

Dalam contoh berikut, karena c+b bukan a, segitiga tersebut bukan segitiga siku-siku.

```
>c &= quad(A,B); $c, b &= quad(A,C); $b, a &= quad(B,C); $a,
```

$$10$$

$$5$$

$$17$$

Pertama, mari kita hitung sudut tradisional. Fungsi `computeAngle` menggunakan metode biasa yang didasarkan pada hasil kali dot dua vektor. Hasilnya adalah perkiraan bilangan floating point.

$$A = \langle 1, 2 \rangle \quad B = \langle 4, 3 \rangle, \quad C = \langle 0, 4 \rangle$$

$$\mathbf{a} = C - B = \langle -4, 1 \rangle, \quad \mathbf{c} = A - B = \langle -3, -1 \rangle, \quad \beta = \angle ABC$$

$$\mathbf{a} \cdot \mathbf{c} = |\mathbf{a}| \cdot |\mathbf{c}| \cos \beta$$

$$\cos \angle ABC = \cos \beta = \frac{\mathbf{a} \cdot \mathbf{c}}{|\mathbf{a}| \cdot |\mathbf{c}|} = \frac{12 - 1}{\sqrt{17} \sqrt{10}} = \frac{11}{\sqrt{17} \sqrt{10}}$$

```
>wb &= computeAngle(A,B,C); $wb, $(wb/pi*180)()
```

$$\arccos\left(\frac{11}{\sqrt{10} \sqrt{17}}\right)$$

32.4711922908

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum silang. Kita memasukkan kuadran a, b, dan c ke dalam hukum silang dan menyelesaikan persamaan untuk x.

```
>$crosslaw(a,b,c,x), $solve(%,x), //(b+c-a)^=4b.c(1-x)
```

$$4 = 200 (1 - x)$$

$$\left[x = \frac{49}{50} \right]$$

Artinya, apa yang dilakukan oleh fungsi `spread` yang didefinisikan dalam "geometry.e".

```
>sb &= spread(b,a,c); $sb
```

$$\frac{49}{170}$$

Maxima mendapatkan hasil yang sama menggunakan trigonometri biasa, jika kita memaksanya. Program ini memang menyelesaikan istilah $\sin(\arccos(\dots))$ menjadi hasil pecahan. Kebanyakan siswa tidak dapat melakukan hal ini.

```
>$sin(computeAngle(A,B,C))^2
```

$$\frac{49}{170}$$

Setelah diperoleh nilai sebaran di titik B, tinggi h_a terhadap sisi a dapat dihitung.

Ingat bahwa, berdasarkan definisi, berlaku hubungan

$$s_b = \frac{h_a}{c}$$

```
>ha &= c*sb; $ha
```

$$\frac{49}{17}$$

Gambar berikut dihasilkan menggunakan program geometri C.a.R., yang dapat menggambar kuadrans dan sebaran (spreads).

Secara definisi, panjang garis ha merupakan akar kuadrat dari kuadransnya.

```
>$sqrt(ha)
```

$$\frac{7}{\sqrt{17}}$$

Sekarang dapat dihitung luas segitiga tersebut.

Perlu diingat bahwa perhitungan dilakukan menggunakan kuadrans, bukan panjang sisi biasa.

```
>$sqrt(ha)*sqrt(a)/2
```

$$\frac{7}{2}$$

Rumus determinan yang biasa digunakan memberikan hasil yang sama.

```
>$areaTriangle(B,A,C)
```

$$\frac{7}{2}$$

Rumus Heron

Selanjutnya, permasalahan ini diselesaikan secara umum.

```
>&remvalue(a,b,c,sb,ha);
```

Pertama, dihitung penyebaran di titik B untuk segitiga dengan sisi a, b, dan c.

Selanjutnya, dihitung luas kuadratnya (disebut juga *‘quadrea’*), difaktorkan menggunakan **Maxima**, dan diperoleh rumus Heron yang terkenal.

Proses ini cukup sulit dilakukan dengan pensil dan kertas.

```
>$spread(b^2,c^2,a^2), $factor(%*c^2*a^2/4)
```

$$\frac{-c^4 - (-2b^2 - 2a^2)c^2 - b^4 + 2a^2b^2 - a^4}{4a^2c^2}$$

$$\frac{(-c + b + a)(c - b + a)(c + b - a)(c + b + a)}{16}$$

Aturan Sebaran Rangkap Tiga (Triple Spread Rule)

Kelemahan dari penggunaan sebaran (spread) adalah nilainya tidak dapat dijumlahkan secara langsung seperti sudut.

Namun, tiga sebaran dalam sebuah segitiga memenuhi aturan berikut yang disebut aturan sebaran rangkap tiga.

```
>&remvalue(sa,sb,sc); $triplespread(sa,sb,sc)
```

$$(sc + sb + sa)^2 = 2(sc^2 + sb^2 + sa^2) + 4sa\,sb\,sc$$

Aturan ini berlaku untuk setiap tiga sudut yang jumlahnya 180° .

$$\alpha + \beta + \gamma = \pi$$

Sejak penyebaran

$$\alpha, \pi - \alpha$$

jika sama, aturan selisih tiga juga berlaku, jika

$$\alpha + \beta = \gamma$$

Karena selisih sudut negatifnya sama, aturan selisih tiga juga berlaku, jika

$$\alpha + \beta + \gamma = 0$$

Misalnya, kita dapat menghitung selisih sudut 60° . Nilainya adalah $3/4$. Persamaan-persamaan tersebut memiliki solusi kedua, di mana semua selisihnya adalah 0.

```
>$solve(triplespread(x,x,x),x)
```

$$\left[x = \frac{3}{4}, x = 0 \right]$$

Sebaran untuk sudut 90° jelas bernilai 1.

ika dua sudut berjumlah 90° , maka sebarannya memenuhi persamaan sebaran rangkap tiga dengan a, b, 1.

ari perhitungan berikut diperoleh hasil bahwa $a + b = 1$.

```
>$triplespread(x,y,1), $solve(%,x)
```

$$(y + x + 1)^2 = 2 (y^2 + x^2 + 1) + 4 x y$$

$$[x = 1 - y]$$

Karena selisih 180° -t sama dengan selisih t, rumus selisih tiga sudut juga berlaku, jika salah satu sudut merupakan penjumlahan atau pengurangan dari dua sudut lainnya.

Jadi, kita dapat menemukan selisih sudut yang digandakan. Perhatikan bahwa ada dua solusi lagi. Kita jadikan ini sebagai fungsi.

```
>$solve(triplespread(a,a,x),x), function doublespread(a) &= factor(rhs( %[1]))
```

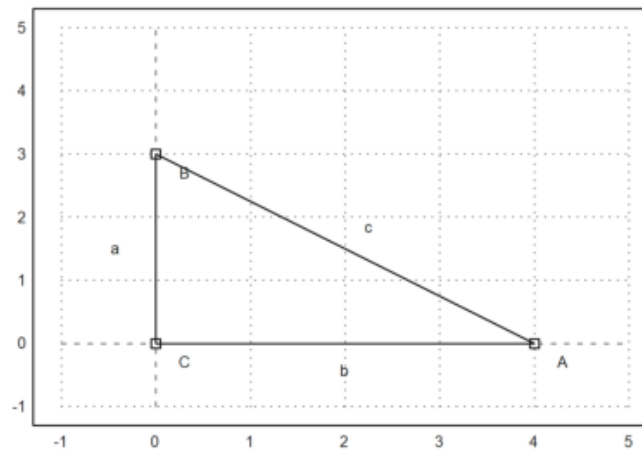
$$[x = 4 a - 4 a^2, x = 0]$$

$$- 4 (a - 1) a$$

Pembagi Sudut (Angle Bisectors)

1. Pertama, ditinjau situasi yang sudah diketahui sebelumnya.

```
>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;
```



2. hitung panjang pembagi sudut di A.
 perhitungan dilakukan untuk sisi umum a, b, dan c.

```
>&remvalue(a,b,c);
```

3. Hitung sebaran sudut yang dibagi di A menggunakan rumus sebaran rangkap tiga (triple spread formula).

Masalah pada rumus ini muncul kembali, karena memiliki dua solusi.

Solusi yang tepat harus dipilih, sedangkan solusi lain merujuk pada sudut yang dibagi 180° - wa.

```
>$triplespread(x,x,a/(a+b)), $solve(%,x), sa2 &= rhs(%[1]); $sa2
```

$$\left(2x + \frac{a}{b+a}\right)^2 = 2 \left(2x^2 + \frac{a^2}{(b+a)^2}\right) + \frac{4ax^2}{b+a}$$

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

$$\frac{-\sqrt{b^2 + ab} + b + a}{2b + 2a}$$

4. Lakukan pemeriksaan pada persegi panjang Mesir (Egyptian rectangle).

```
>$sa2 with [a=3^2,b=4^2]
```

$$\frac{1}{10}$$

5. Sudut dapat ditampilkan di Euler setelah mengubah sebaran menjadi radian.

```
>wa2 := arcsin(sqrt(1/10)); degprint(wa2)
```

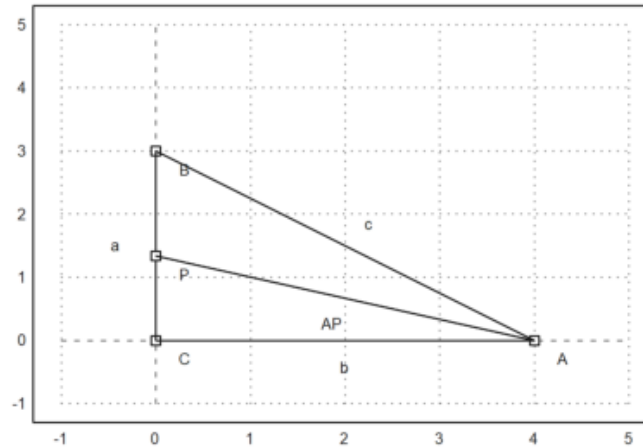
18°26'5.82''

Titik P merupakan titik potong pembagi sudut dengan sumbu y.

```
>P := [0,tan(wa2)*4]
```

[0, 1.33333]

```
>plotPoint(P,"P"); plotSegment(A,P):
```



6. Lakukan pemeriksaan sudut-sudut pada contoh spesifik.

```
>computeAngle(C,A,P), computeAngle(P,A,B)
```

```
0.321750554397
```

```
0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.

Kita gunakan teorema sinus pada segitiga APC. Teorema ini menyatakan bahwa

$$\frac{BC}{\sin(w_a)} = \frac{AC}{\sin(w_b)} = \frac{AB}{\sin(w_c)}$$

berlaku di setiap segitiga. Kuadratkan, dan rumus ini menghasilkan apa yang disebut "hukum sebaran".

$$\frac{a}{s_a} = \frac{b}{s_b} = \frac{c}{s_c}$$

di mana a, b, c menunjukkan kuadran.

Karena CPA sebaran adalah $1-sa^2$, kita peroleh darinya bisa $1 = b / (1-sa^2)$ dan dapat menghitung bisa (kuadran dari garis bagi sudut).

```
>factor(ratsimp(b/(1-sa^2))); bisa &= %; $bisa
```

$$\frac{2b(b+a)}{\sqrt{b(b+a)} + b + a}$$

Mari kita periksa rumus ini untuk nilai-nilai Mesir kita.

```
>sqrt(mxmeval("at(bisa,[a=3^2,b=4^2])")), distance(A,P)
```

```
4.21637021356
```

```
4.21637021356
```

Kita juga dapat menghitung P menggunakan rumus penyebaran.

```
>py:=factor(ratsimp(sa2*bisa)); $py
```

$$-\frac{b \left(\sqrt{b(b+a)} - b - a \right)}{\sqrt{b(b+a)} + b + a}$$

Nilai tersebut sama dengan yang kita dapatkan menggunakan rumus trigonometri.

```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

1.33333333333

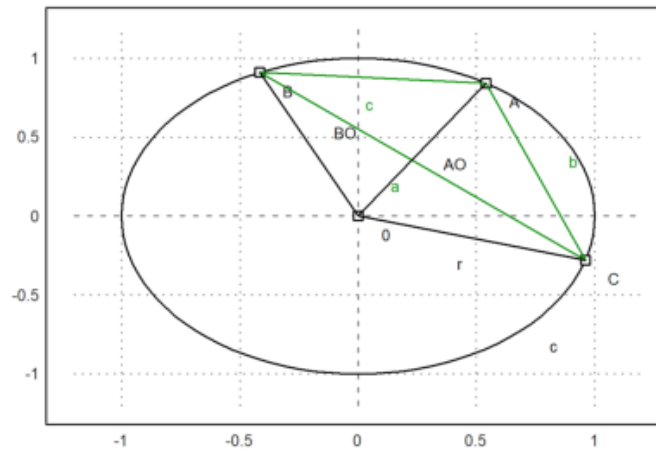
Sudut Tali Busur (The Chord Angle)

1. Pertama, diperhatikan suatu keadaan atau konfigurasi yang menunjukkan hubungan antara dua titik pada lingkaran dan tali busur yang menghubungkannya.

```

>setPlotRange(1.2); ...
>color(1); plotCircle(circleWithCenter([0,0],1)); ...
>A:=[cos(1),sin(1)]; B:=[cos(2),sin(2)]; C:=[cos(6),sin(6)]; ...
>plotPoint(A,"A"); plotPoint(B,"B"); plotPoint(C,"C"); ...
>color(3); plotSegment(A,B,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...
>color(1); O:=[0,0]; plotPoint(O,"O"); ...
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
>insimg;

```



2. Dapat digunakan Maxima untuk menyelesaikan rumus sebaran rangkap tiga pada sudut-sudut di pusat O terhadap r .

Dengan demikian, diperoleh rumus jari-jari kuadrat lingkaran luar (pericircle) yang dinyatakan dalam kuadrans sisi-sisinya.

Pada proses ini, Maxima menghasilkan beberapa akar kompleks yang diabaikan.

```
>remvalue(a,b,c,r); // hapus nilai-nilai sebelumnya untuk perhitungan baru
>rabc &= rhs(solve(triplespread(spread(b,r,r),spread(a,r,r),spread(c,r,r)),r)[4]); $rabc
```

$$-\frac{a b c}{c^2 - 2 b c + a (-2 c - 2 b) + b^2 + a^2}$$

3. Hal tersebut dapat dibuat sebagai fungsi dalam Euler.

```
>function periradius(a,b,c) &= rabc;
```

Selanjutnya, dilakukan pemeriksaan hasil untuk titik-titik A, B, dan C.

```
>a:=quadrance(B,C); b:=quadrance(A,C); c:=quadrance(A,B);
```

5. periksa hasil untuk titik A, B, dan C. Jari-jari tersebut memang bernilai 1.

```
>periradius(a,b,c)
```

Faktanya, sebaran $\angle CBA$ hanya bergantung pada b dan c .
Hal ini dikenal sebagai teorema sudut tali busur (chord angle theorem).

```
>$spread(b,a,c)*rabc | ratsimp
```

$$\frac{b}{4}$$

Faktanya, sebaran bernilai $b / (4r)$, dan terlihat bahwa sudut tali busur pada tali b adalah setengah dari sudut di pusat.

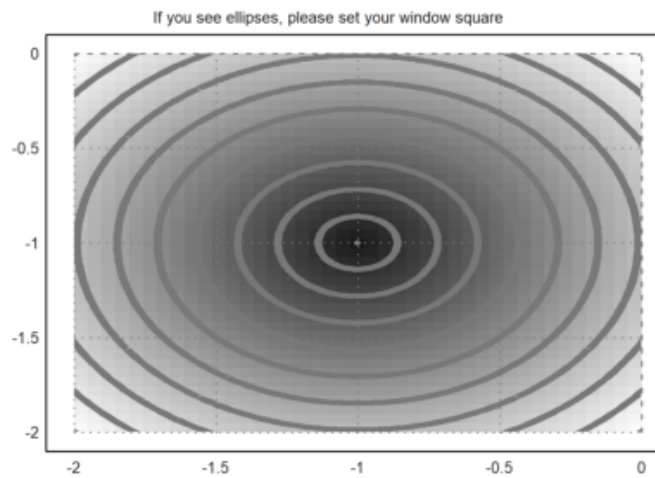
```
>$doublespread(b/(4*r))-spread(b,r,r) | ratsimp
```

Contoh 6: Jarak Minimal pada Bidang

Catatan pendahuluan

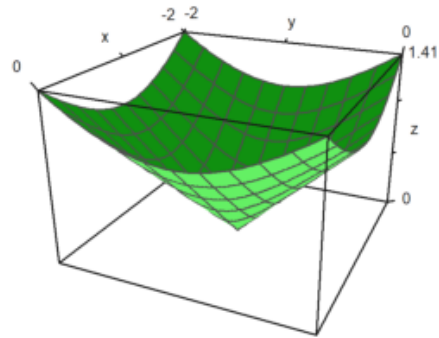
Fungsi yang, untuk suatu titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis tingkat yang cukup sederhana: lingkaran berpusat di A.

```
>remvalue();  
>A=[-1,-1];  
>function d1(x,y):=sqrt((x-A[1])^2+(y-A[2])^2)  
>fcontour("d1",xmin=-2,xmax=0,ymin=-2,ymax=0,hue=1, ...  
>title="If you see ellipses, please set your window square");
```



dan grafiknya juga cukup sederhana: bagian atas kerucut:

```
>plot3d("d1",xmin=-2,xmax=0,ymin=-2,ymax=0):
```

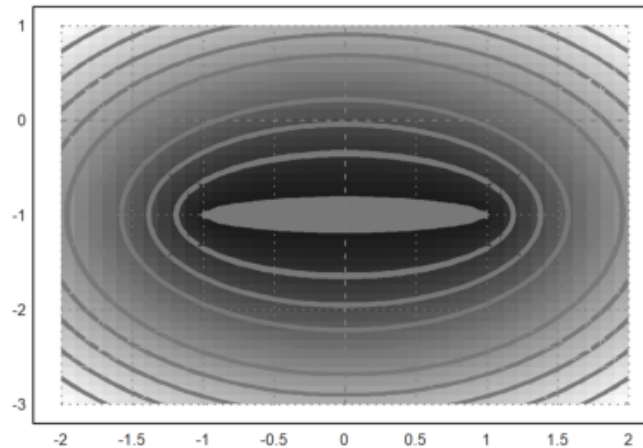


Tentu saja, titik minimum 0 tercapai di A.

Dua titik

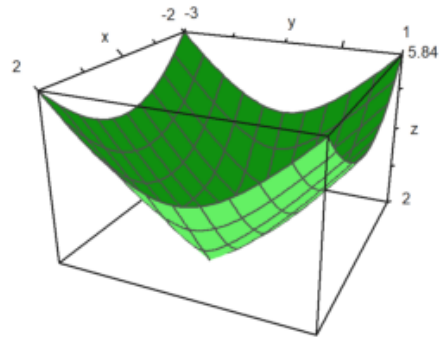
Sekarang kita lihat fungsi $MA+MB$ di mana A dan B adalah dua titik (tetap). Sudah menjadi "fakta umum" bahwa kurva level berbentuk elips, dengan titik fokus A dan B; kecuali untuk titik minimum AB yang konstan pada segmen $[AB]$:

```
>B=[1,-1];  
>function d2(x,y):=d1(x,y)+sqrt((x-B[1])^2+(y-B[2])^2)  
>fcontour("d2",xmin=-2,xmax=2,ymin=-3,ymax=1,hue=1):
```



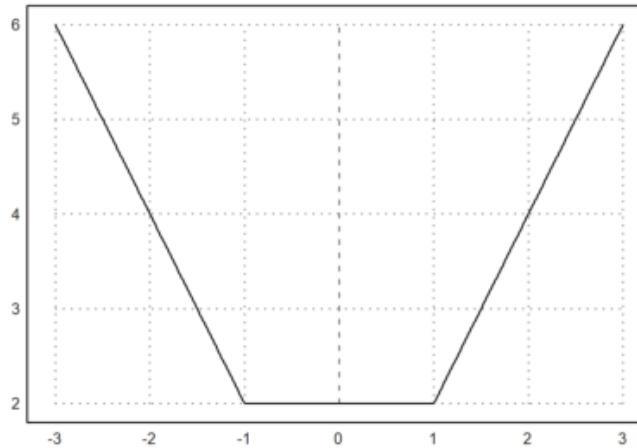
Grafiknya lebih menarik:

```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



Pembatasan pada garis (AB) lebih terkenal:

```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```



Tiga poin

Sekarang, semuanya menjadi kurang sederhana: Memang kurang diketahui bahwa $MA+MB+MC$ mencapai nilai minimumnya di salah satu titik bidang, tetapi menentukannya tidaklah sesederhana itu:

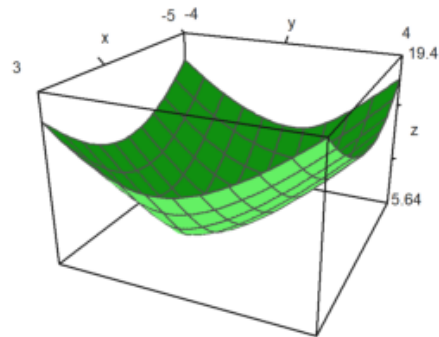
1) Jika salah satu sudut segitiga ABC lebih dari 120° (misalnya di A), maka nilai minimumnya tercapai di titik tersebut (misalnya $AB+AC$).

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;

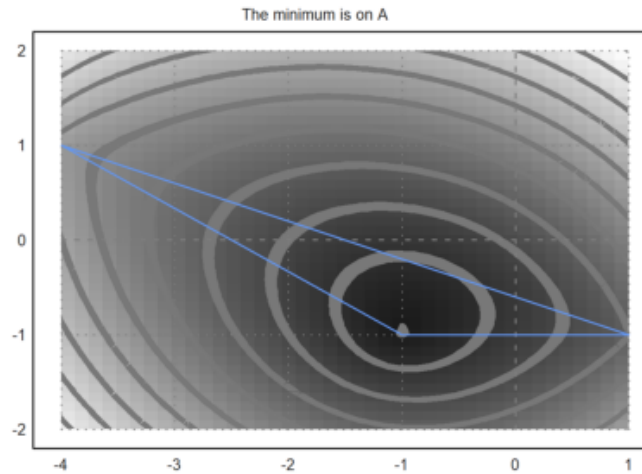
```



```

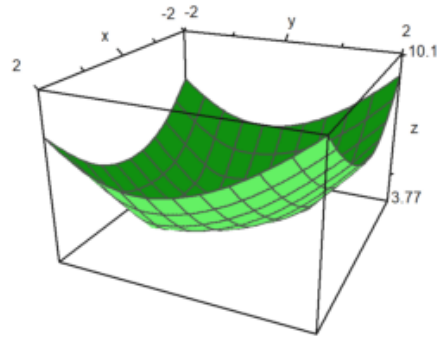
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;

```

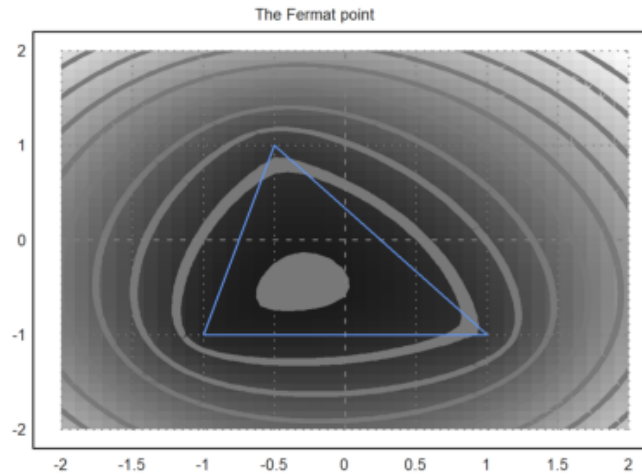


2) Namun jika semua sudut segitiga ABC kurang dari 120° , maka sudut minimumnya ada di titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (masing-masing 120°):

```
>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;
```



Kegiatan yang menarik adalah merealisasikan gambar tersebut menggunakan perangkat lunak geometri; misalnya, terdapat perangkat lunak berbasis Java yang memiliki perintah “garis kontur”.

Penemuan ini dilakukan oleh seorang hakim Prancis bernama Pierre de Fermat, yang menulis surat kepada para diletan seperti pendeta Marin Mersenne dan Blaise Pascal, yang bekerja pada bidang pajak penghasilan.

Titik unik F, sehingga $FA + FB + FC$ minimum, disebut titik Fermat segitiga.

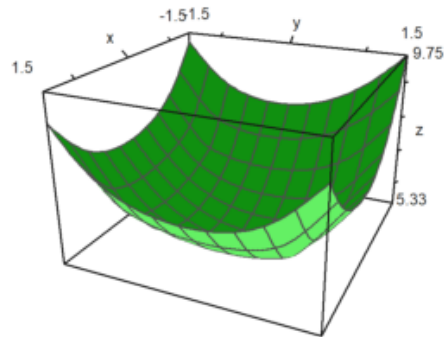
amun, tampaknya beberapa tahun sebelumnya, orang Italia Torricelli telah menemukan titik ini sebelum Fermat.

Tetap menggunakan notasi F untuk titik tersebut.

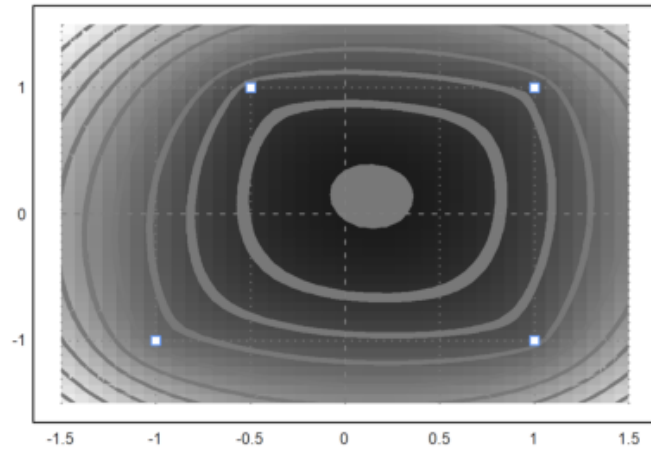
Empat Titik (Four Points)

Tambahkan titik keempat D dan dicoba meminimalkan jumlah $MA + MB + MC + MD$.
isalnya, sebagai operator TV kabel, tujuan adalah menentukan posisi antenna agar dapat menjangkau empat desa dengan panjang kabel seminimal mungkin.

```
>D=[1,1];  
>function d4(x,y):=d3(x,y)+sqrt((x-D[1])^2+(y-D[2])^2)  
>plot3d("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5):
```



```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);  
>P=(A_B_C_D)'; plot2d(P[1],P[2],points=1,add=1,color=12);  
>insimg;
```



Masih ada nilai minimum, dan nilai tersebut tidak tercapai di salah satu dari titik sudut A, B, C, atau D:

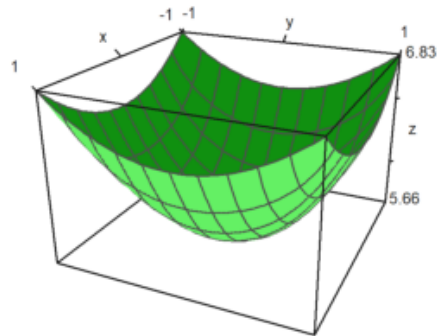
```
>function f(x):=d4(x[1],x[2])  
>neldermin("f",[0.2,0.2])
```

```
[0.142858, 0.142857]
```

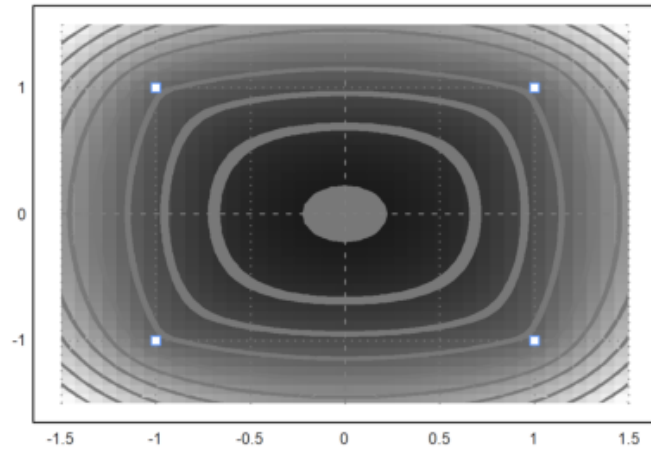
Sepertinya dalam kasus ini, koordinat titik optimal adalah rasional atau hampir rasional...

Sekarang ABCD adalah persegi, kita mengharapkan bahwa titik optimal akan menjadi pusat ABCD:

```
>C=[-1,1];  
>plot3d("d4",xmin=-1,xmax=1,ymin=-1,ymax=1):
```



```
>fcontour("d4",xmin=-1.5,xmax=1.5,ymin=-1.5,ymax=1.5,hue=1);  
>P=(A_B_C_D)'; plot2d(P[1],P[2],add=1,color=12,points=1);  
>insimg;
```



```
>load geometry;  
>g1 &= lineThrough([0,0],[1,a])
```

$[-a, 1, 0]$

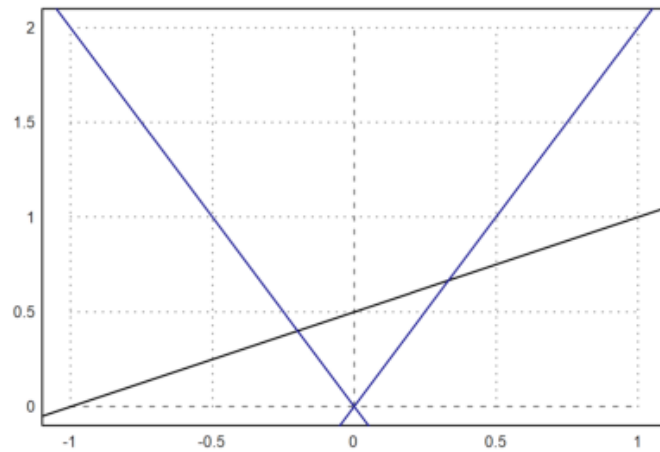
```
>g2 &= lineThrough([0,0],[-1,a])
```

$[-a, -1, 0]$

```
>g &= lineThrough([-1,0],[1,1])
```

[- 1, 2, 1]

```
>setPlotRange(-1,1,0,2);  
>color(black); plotLine(g(),"")  
>a:=2; color(blue); plotLine(g1(),""), plotLine(g2(),""):
```



```
>P &= [0,u]
```

$[0, u]$

```
>d1 &= distance(P,projectToLine(P,g1)); $d1
```

$$\sqrt{\left(\frac{a^2 u}{a^2 + 1} - u\right)^2 + \frac{a^2 u^2}{(a^2 + 1)^2}}$$

```
>d &= distance(P,projectToLine(P,g)); $d
```

$$\sqrt{\left(\frac{u + 2}{5} - u\right)^2 + \frac{(2u - 1)^2}{25}}$$

```
>sol &= solve(d1^2=d^2,u); $sol
```

$$\left[u = \frac{-\sqrt{5}\sqrt{a^2+1}+2a^2+2}{4a^2-1}, u = \frac{\sqrt{5}\sqrt{a^2+1}+2a^2+2}{4a^2-1} \right]$$

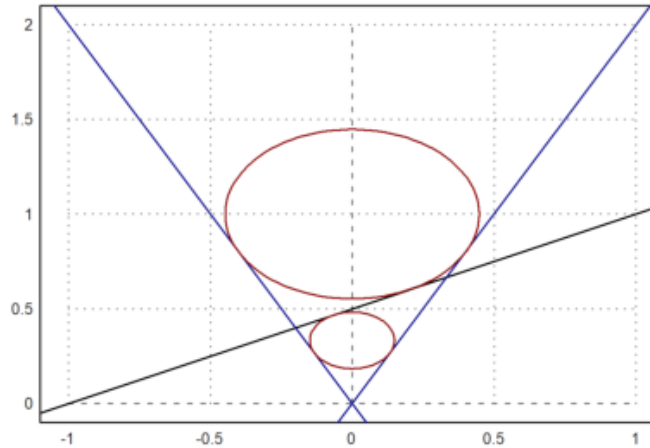
```
>u := sol()
```

```
[0.333333, 1]
```

```
>dd := d()
```

```
[0.149071, 0.447214]
```

```
>color(red);  
>plotCircle(circleWithCenter([0,u[1]],dd[1]),"");  
>plotCircle(circleWithCenter([0,u[2]],dd[2]),"");  
>insimg;
```



Plot dengan Povray

Selanjutnya, kita plot semuanya dengan Povray. Perhatikan bahwa Anda mengubah perintah apa pun dalam urutan perintah Povray berikut, dan menjalankan kembali semua perintah dengan Shift-Return.

Pertama, kita memuat fungsi povray.

```
>load povray;  
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

Kami menyiapkan adegan dengan tepat.

```
>povstart(zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```

Selanjutnya, kita menuliskan kedua bola tersebut ke dalam berkas Povray.

```
>writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));  
>writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
```

Dan kerucut, transparan.

```
>writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
```

Kami menghasilkan sebuah bidang yang dibatasi oleh 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 menentukan dua titik pada lingkaran, di mana bola-bola tersebut menyentuh kerucut.

```
>function turnz(v) := return [-v[2],v[1],v[3]]
>P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
>writeln(povpoint(P1,povlook(yellow)));
>P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
>writeln(povpoint(P2,povlook(yellow)));
```

Kemudian kita menentukan dua titik di mana bola-bola tersebut menyentuh bidang. Titik-titik ini merupakan fokus-fokus elips.

```
>P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
>writeln(povpoint(P3,povlook(yellow)));
>P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
>writeln(povpoint(P4,povlook(yellow)));
```

Selanjutnya, kita menghitung irisan 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.

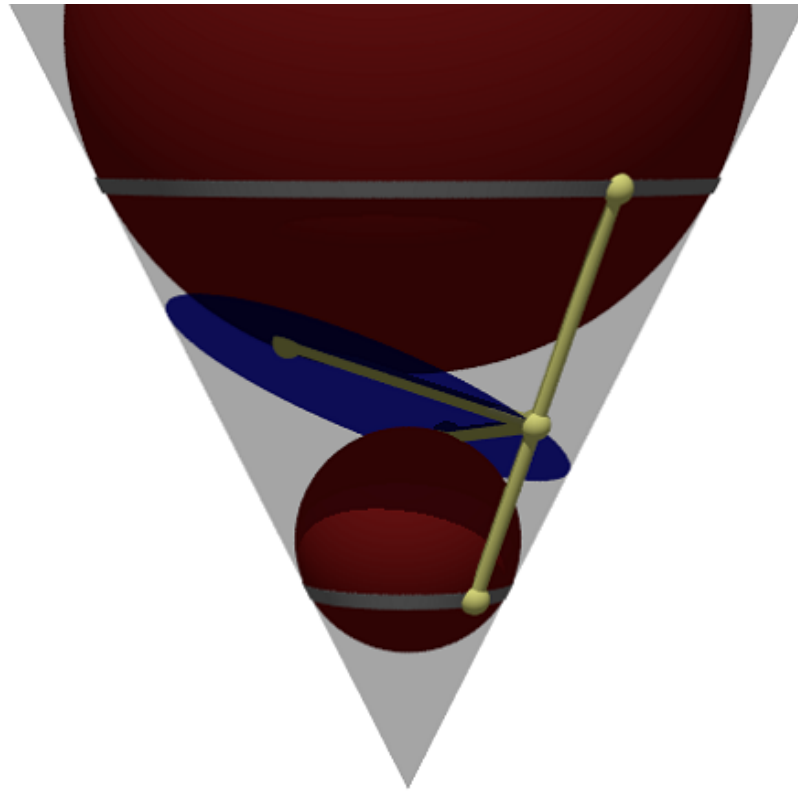
```
>writeln(povsegment(P1,P2,povlook(yellow)));  
>writeln(povsegment(P5,P3,povlook(yellow)));  
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Sekarang kita menghasilkan pita abu-abu di mana bola-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)));
```

Jalankan program Povray.

```
>povend();
```



Untuk mendapatkan anaglyph dari ini, kita perlu memasukkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali nanti.

```
>function scene () ...
```

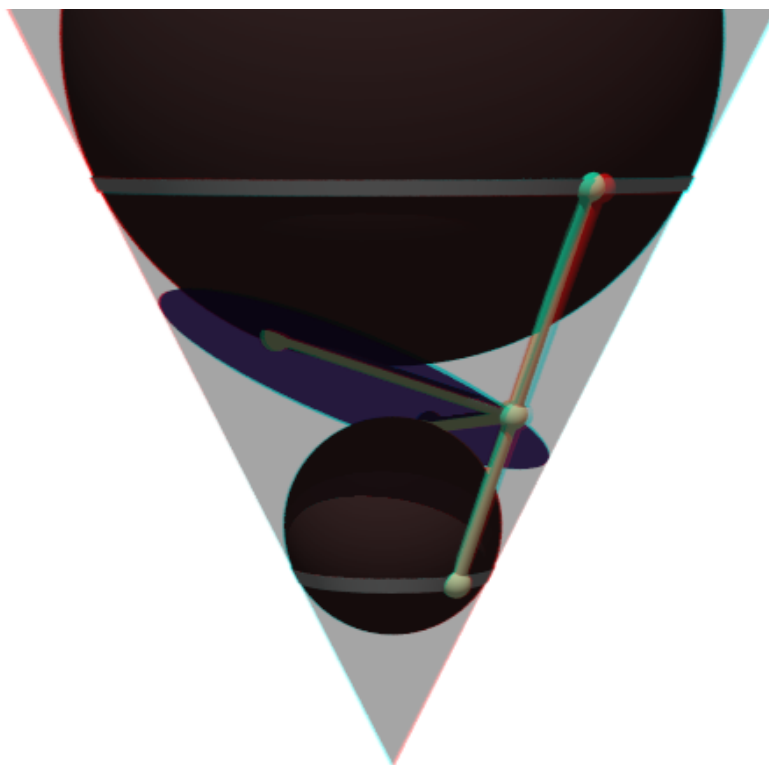
```

global a,u,dd,g,g1,defaultpointsize;
writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));
writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));
writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));
gp=g();
pc=povcone([0,0,0],0,[0,0,a],1,"");
vp=[gp[1],0,gp[2]]; dp=gp[3];
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
P1=projectToLine([0,u[1]],g1()); P1=turnz([P1[1],0,P1[2]]);
writeln(povpoint(P1,povlook(yellow)));
P2=projectToLine([0,u[2]],g1()); P2=turnz([P2[1],0,P2[2]]);
writeln(povpoint(P2,povlook(yellow)));
P3=projectToLine([0,u[1]],g()); P3=[P3[1],0,P3[2]];
writeln(povpoint(P3,povlook(yellow)));
P4=projectToLine([0,u[2]],g()); P4=[P4[1],0,P4[2]];
writeln(povpoint(P4,povlook(yellow)));
t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
writeln(povpoint(P5,povlook(yellow)));
writeln(povsegment(P1,P2,povlook(yellow)));
writeln(povsegment(P5,P3,povlook(yellow)));
writeln(povsegment(P5,P4,povlook(yellow)));
pcw=povcone([0,0,0],0,[0,0,a],1.01);
pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc1],povlook(gray)));
pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
writeln(povintersection([pcw,pc2],povlook(gray)));
endfunction

```

Anda memerlukan kacamata merah/biru untuk menikmati efek berikut ini.

```
>povanaglyph("scene",zoom=11,center=[0,0,0.5],height=10°,angle=140°);
```



Contoh 8: Geometri Bumi

Di buku catatan ini, kita ingin melakukan beberapa komputasi sferis. Fungsi-fungsinya terdapat dalam berkas "spherical.e" di folder contoh. Kita perlu memuat berkas tersebut terlebih dahulu.

```
>load "spherical.e";
```

Untuk memasukkan posisi geografis, kita menggunakan vektor dengan dua koordinat dalam satuan radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut adalah koordinat untuk Kampus FMIPA UNY.

```
>FMIPA=[rad(-7,-46.467),rad(110,23.05)]
```

```
[-0.13569, 1.92657]
```

Anda dapat mencetak posisi ini menggunakan `sposprint` (cetak posisi bola).

```
>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 titik ke titik lain pada bola ideal. Vektor ini berupa [arah, jarak] dalam radian. Untuk menghitung jarak di Bumi, kita mengalikan dengan jari-jari Bumi pada 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. Rutin-rutin berikut menggunakan perkiraan yang lebih baik lagi. Pada jarak yang begitu pendek, hasilnya hampir sama.

```
>esdist(FMIPA,Semarang)->km // perkiraan jarak FMIPA-Semarang
```

88.0114026318

Ada fungsi untuk judul, yang memperhitungkan 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(
```

180°0'10.77''

Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, perhitungan ini tidak akurat karena adanya kesalahan pengurangan dalam asumsi pi.

```
>(asum-pi)*rearth(48°)^2->km^2 // perkiraan luas segitiga FMIPA-Solo-Semarang
```

2116.02948749

Ada fungsi untuk ini, yang menggunakan rata-rata lintang segitiga untuk menghitung jari-jari Bumi, dan menangani kesalahan pembulatan untuk segitiga yang sangat kecil.

```
>esarea(Solo,FMIPA,Semarang)->" km^2", //perkiraan yang sama dengan fungsi esarea()
```

```
2123.64310526 km^2
```

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi arah dan jarak, keduanya dalam radian. Untuk mendapatkan vektor, kita menggunakan svector. Untuk menambahkan vektor ke posisi, kita menggunakan saddvector.

```
>v=svector(FMIPA,Solo); sposprint(saddvector(FMIPA,v)), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'
```

```
S 7°34.333' E 110°49.683'
```

Fungsi-fungsi ini mengasumsikan bola ideal. Hal yang sama berlaku 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'
```

```
>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. Kita mendapatkan perkiraan yang cukup akurat.

```
>esdist(Tugu,Monas)->km // perkiraan jarak Tugu Jogja - Monas Jakarta
```

```
431.565659488
```

Judulnya sama dengan 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 arah dan jarak ke posisi asli. Hal ini terjadi karena kita tidak menghitung fungsi invers secara tepat, melainkan menggunakan perkiraan jari-jari bumi sepanjang jalur tersebut.

```
>sposprint(esadd(Tugu,esdir(Tugu,Monas),esdist(Tugu,Monas)))
```

S 6°10.500' E 106°48.717'

Kesalahan tersebut tidak terlalu besar, namun.

```
>sposprint(Monas),
```

S 6°10.500' E 106°48.717'

Tentu saja, kita tidak bisa berlayar dengan arah yang sama dari satu tujuan ke tujuan lain jika kita ingin mengambil jalur terpendek. Bayangkan, Anda terbang ke arah timur laut dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti arah yang konstan!

Perhitungan berikut menunjukkan bahwa kita jauh dari tujuan yang benar jika kita menggunakan arah yang sama selama perjalanan kita.

```
>dist=esdist(Tugu,Monas); hd=esdir(Tugu,Monas);
```

Sekarang kita tambahkan 10 kali sepersepuluh jarak, menggunakan arah ke Monas, kita sampai di Tugu.

```
>p=Tugu; loop 1 to 10; p=esadd(p,hd,dist/10); end;
```

Hasilnya masih 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 lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran lintang 30°, melainkan jalur yang lebih pendek yang dimulai 10° lebih utara di P1.

```
>sdegprint(esdir(P1,P2))
```

79.69°

Namun, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke Kutub Utara! Oleh karena itu, kita harus menyesuaikan arah kita sepanjang perjalanan. Untuk perkiraan kasar, kita menyesuaikannya pada 1/10 dari jarak total.

```
>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°

Jarak-jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan jika kita terus mengikuti arah yang sama terlalu lama.

```
>skmprint(esdist(p,P2))
```

0.203km

Kita mendapatkan perkiraan yang cukup akurat jika kita menyesuaikan arah kita setelah setiap 1/100 dari jarak total antara Tugu dan 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 tujuan navigasi, kita dapat memperoleh urutan posisi GPS sepanjang lingkaran besar menuju Monas menggunakan fungsi navigate.

```
>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'
```

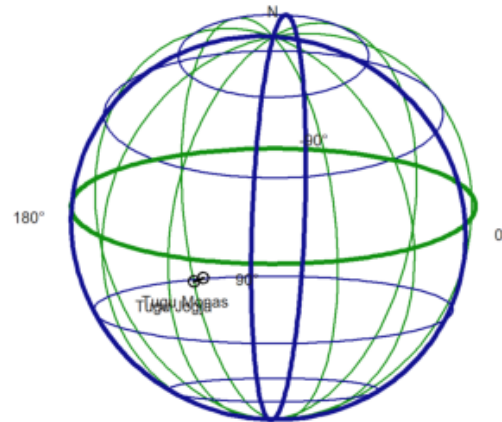
Kami menulis sebuah fungsi yang menggambar Bumi, dua posisi, dan posisi di antara keduanya.

```
>function testplot ...
```

```
    useglobal;  
    plotearth;  
    plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");  
    plotposline(v);  
endfunction
```

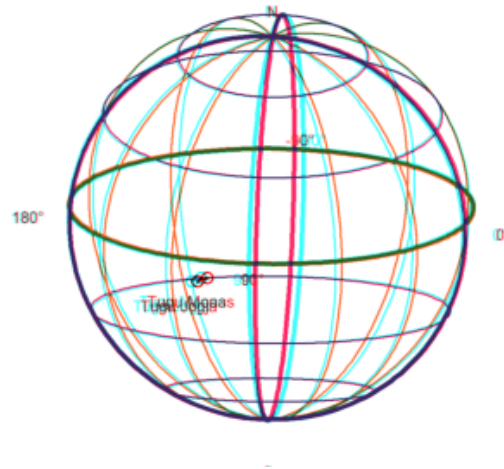
Sekarang gambarkan semuanya.

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=4):
```



Atau gunakan plot3d untuk mendapatkan tampilan anaglyph-nya. Tampilan ini terlihat sangat bagus dengan kacamata merah/biru.

```
>plot3d("testplot",angle=25,height=6,distance=5,own=1,anaglyph=1,zoom=4):
```



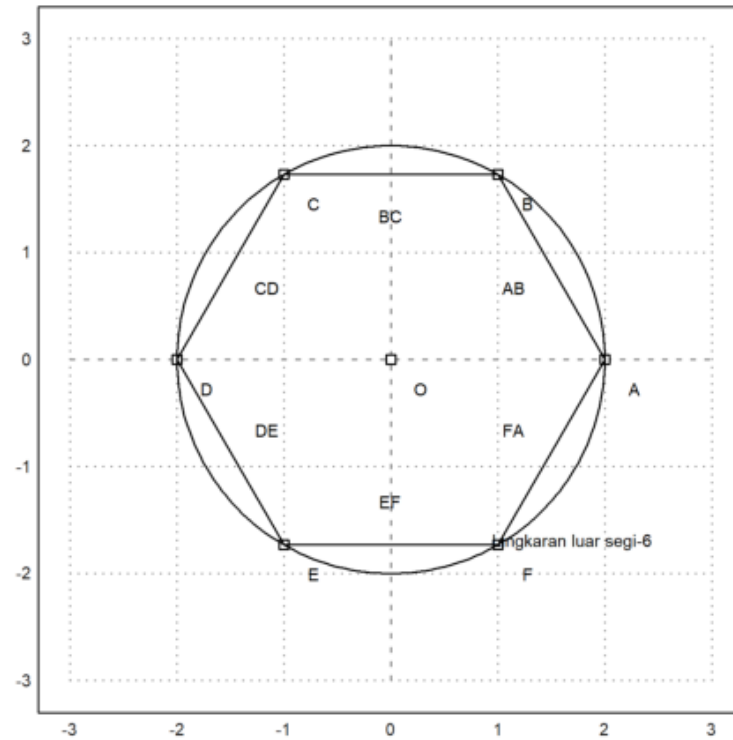
>

1. Gambarkanlah 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.

```
>setPlotRange(-3,3,-3,3); // sesuaikan dengan jari-jari
>O=[0,0]; plotPoint(0,"O"); // titik pusat
>n=6; r=2; theta=360/n; // ganti n dan r sesuai soal
>A=O + r*[cos(0*pi/180), sin(0*pi/180)]; plotPoint(A,"A");
>B=O + r*[cos(theta*pi/180), sin(theta*pi/180)]; plotPoint(B,"B");
>C=O + r*[cos(2*theta*pi/180), sin(2*theta*pi/180)]; plotPoint(C,"C");
>D=O + r*[cos(3*theta*pi/180), sin(3*theta*pi/180)]; plotPoint(D,"D");
>E=O + r*[cos(4*theta*pi/180), sin(4*theta*pi/180)]; plotPoint(E,"E");
>F=O + r*[cos(5*theta*pi/180), sin(5*theta*pi/180)]; plotPoint(F,"F");
>plotSegment(A,B);
>plotSegment(B,C);
>plotSegment(C,D);
>plotSegment(D,E);
>plotSegment(E,F);
>plotSegment(F,A);
>plotCircle(circleWithCenter(O,r),"Lingkaran luar segi-6");
```



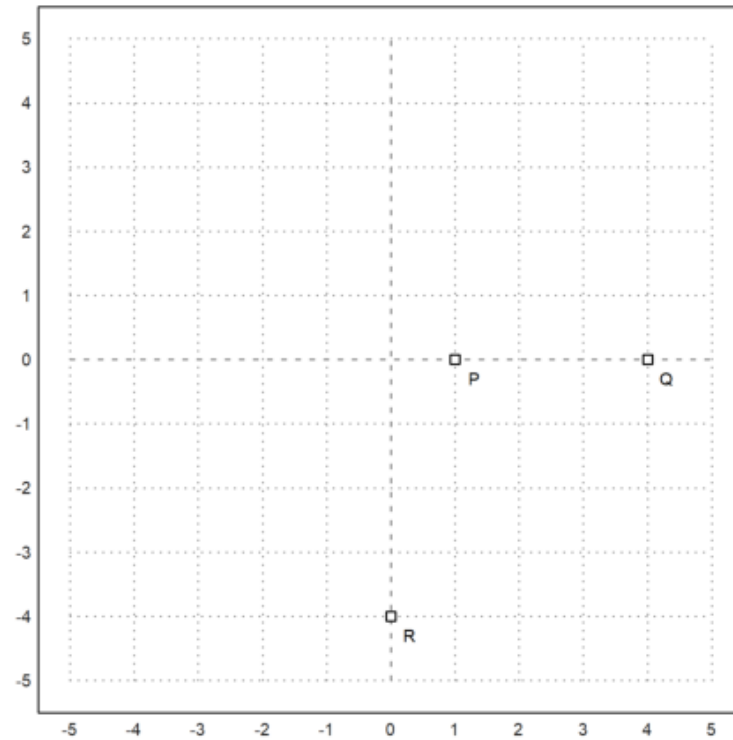
```
>aspect(1);
```

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 .

```
>load geometry;  
>setPlotRange(5); P=[1,0]; Q=[4,0]; R=[0,-4];  
>plotPoint(P,"P"); plotPoint(Q,"Q"); plotPoint(R,"R");
```



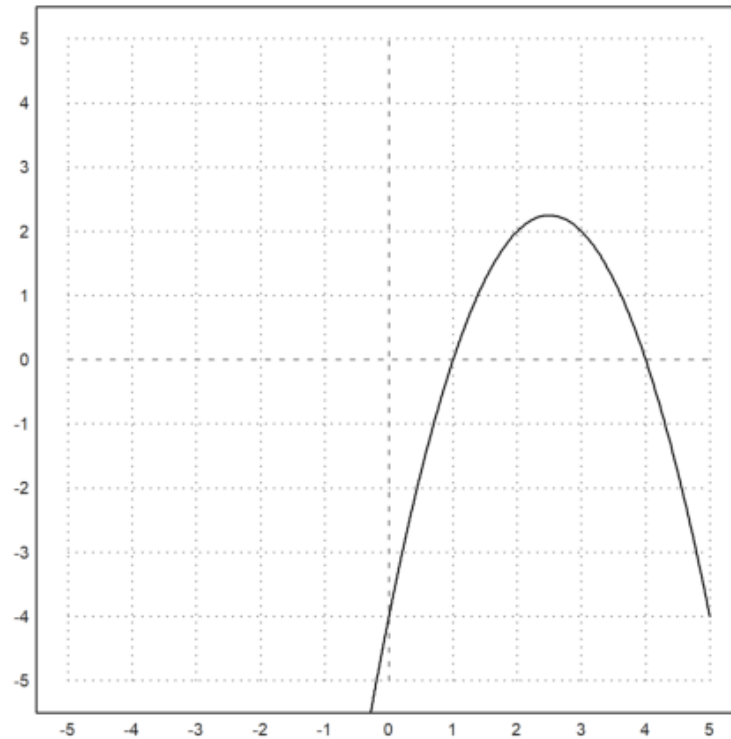
```
>sol &= solve([a+b=-c,16*a+4*b=-c,c=-4],[a,b,c])
```

```
[[a = - 1, b = 5, c = - 4]]
```

```
>function y&=-x^2+5*x-4
```

$$-x^2 + 5x - 4$$

```
>plot2d("-x^2+5*x-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.

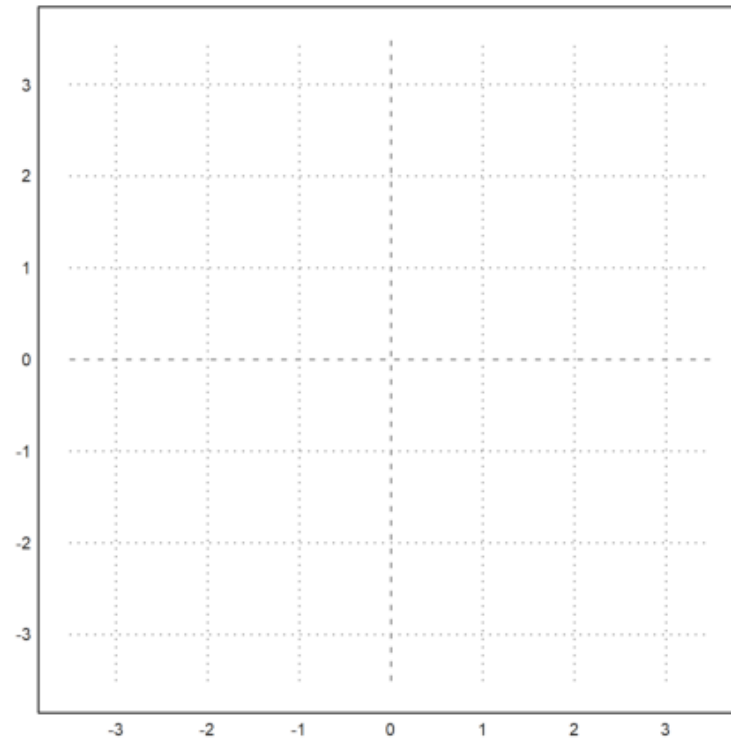
- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis

singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

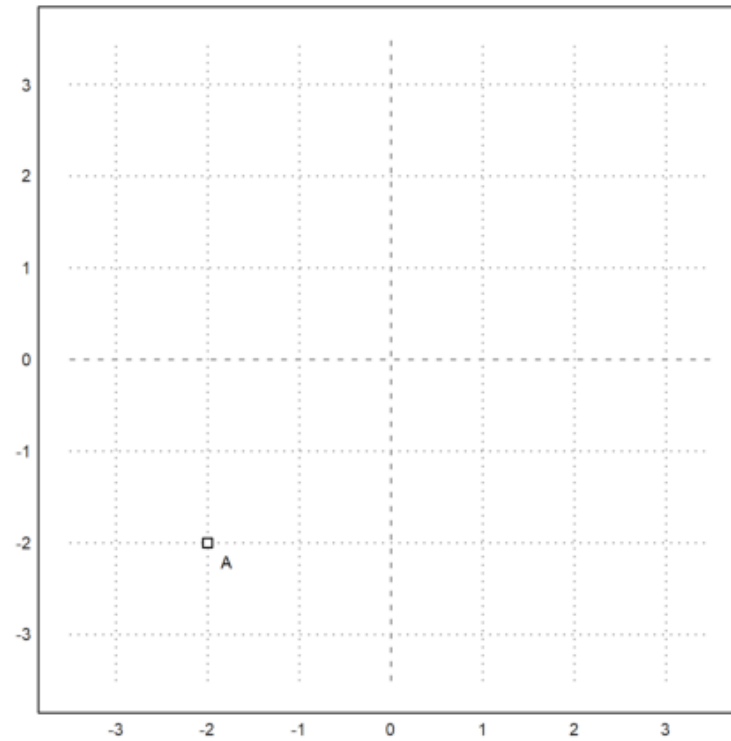
```
>load geometry
```

Numerical and symbolic geometry.

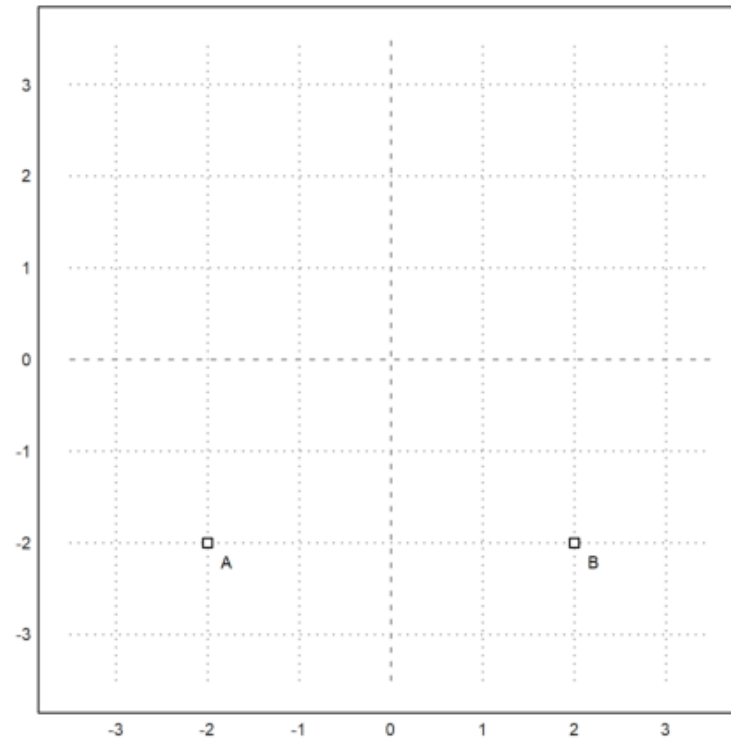
```
>setPlotRange(-3.5,3.5,-3.5,3.5):
```



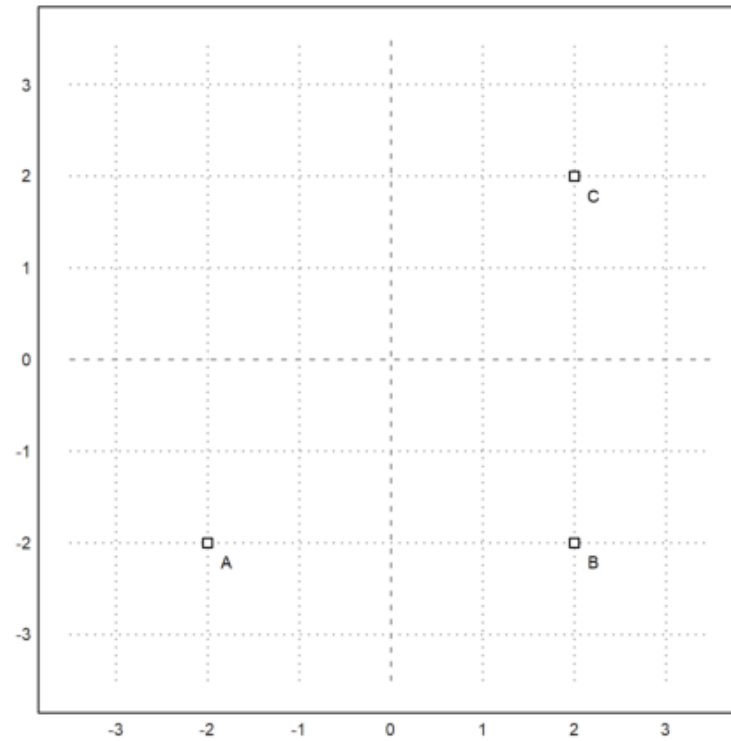
```
>A=[-2,-2]; plotPoint(A,"A"):
```



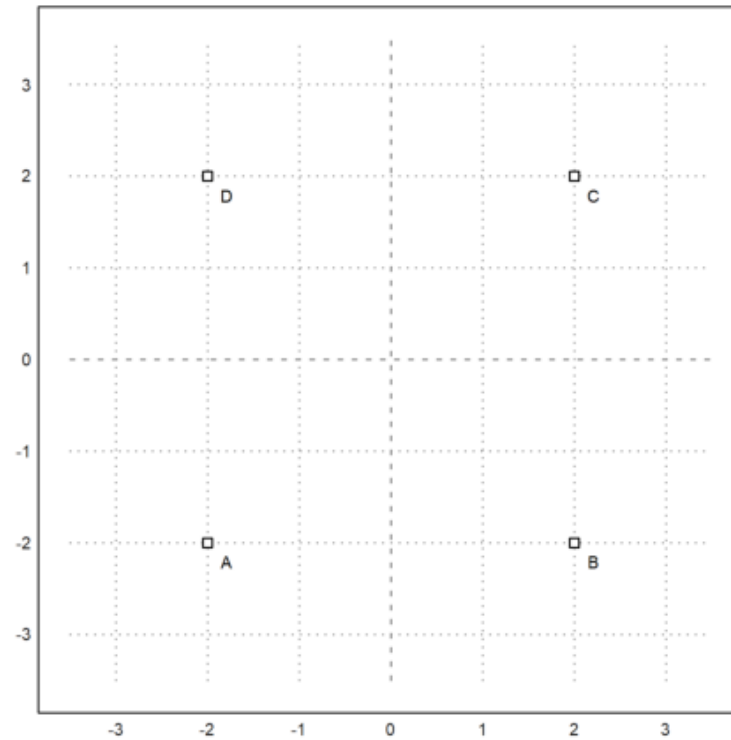
```
>B=[2,-2]; plotPoint(B,"B"):
```



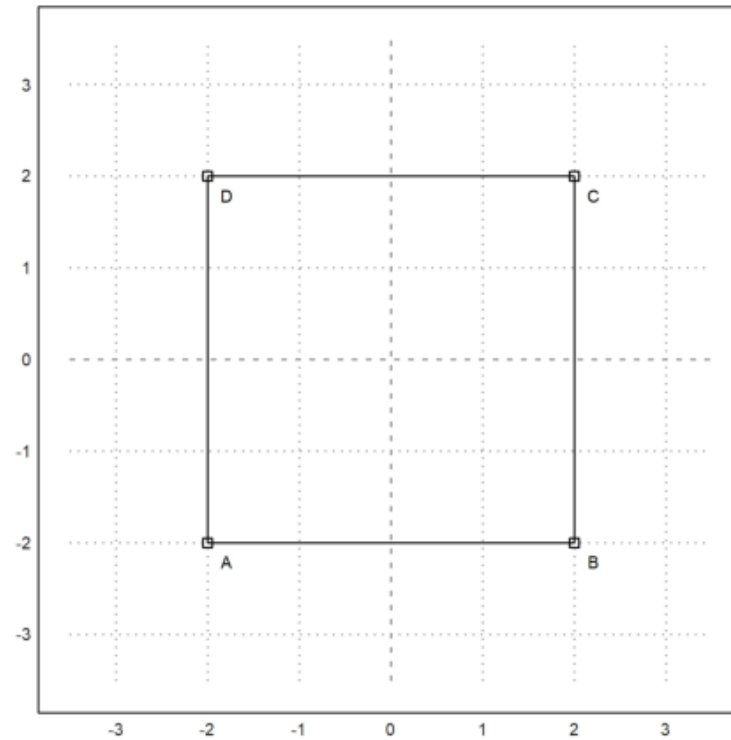
```
>C=[2,2]; plotPoint(C,"C"):
```



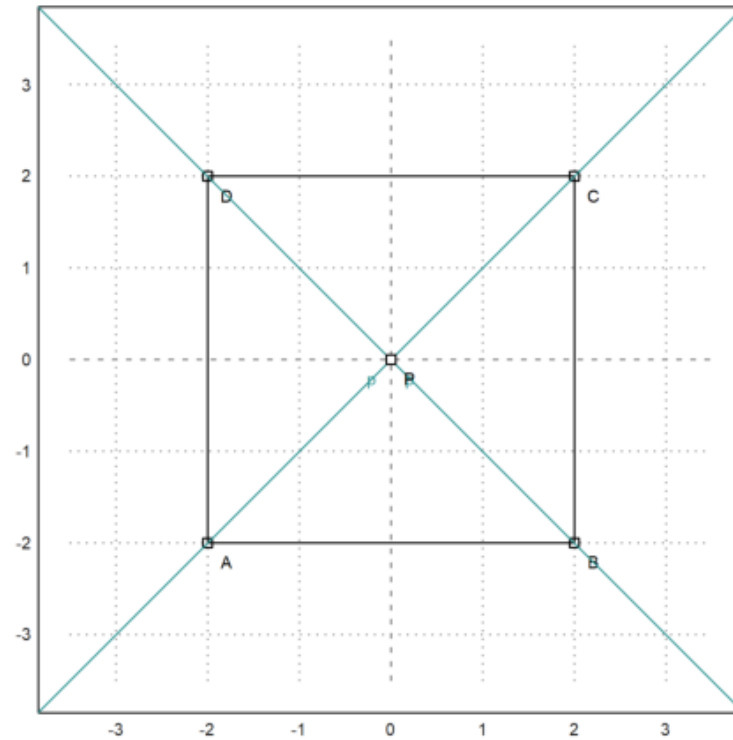
```
>D=[-2,2]; plotPoint(D,"D"):
```



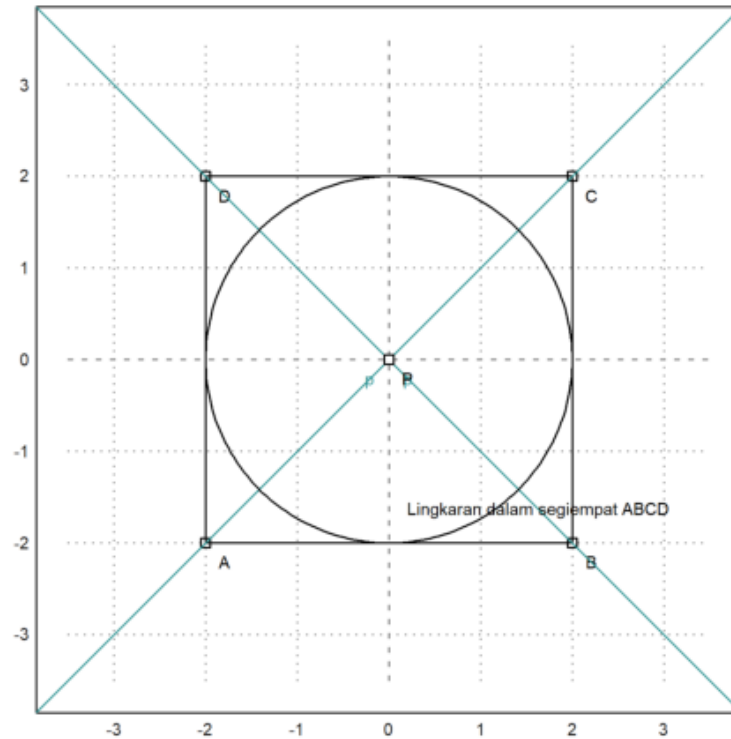
```
>plotSegment(A,B,"");  
>plotSegment(B,C,"");  
>plotSegment(C,D,"");  
>plotSegment(A,D,"");  
>aspect(1):
```



```
>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"):
```



```
>r=norm(P-projectToLine(P,lineThrough(A,B)));  
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segiempat ABCD");
```



Dari gambar di atas, terlihat bahwa sisi-sisinya merupakan garis singgung lingkaran yang sama yaitu lingkaran dalam segiempat.
Kemudian, akan ditunjukkan bahwa hasil kali panjang sisi-sisi yang berhadapan sama

```
>AB=norm(A-B) //panjang sisi AB
```

4

```
>CD=norm(C-D) //panjang sisi CD
```

4

```
>AD=norm(A-D) //panjang sisi AD
```

4

```
>BC=norm(B-C) //panjang sisi BC
```

4

```
>AB.CD
```

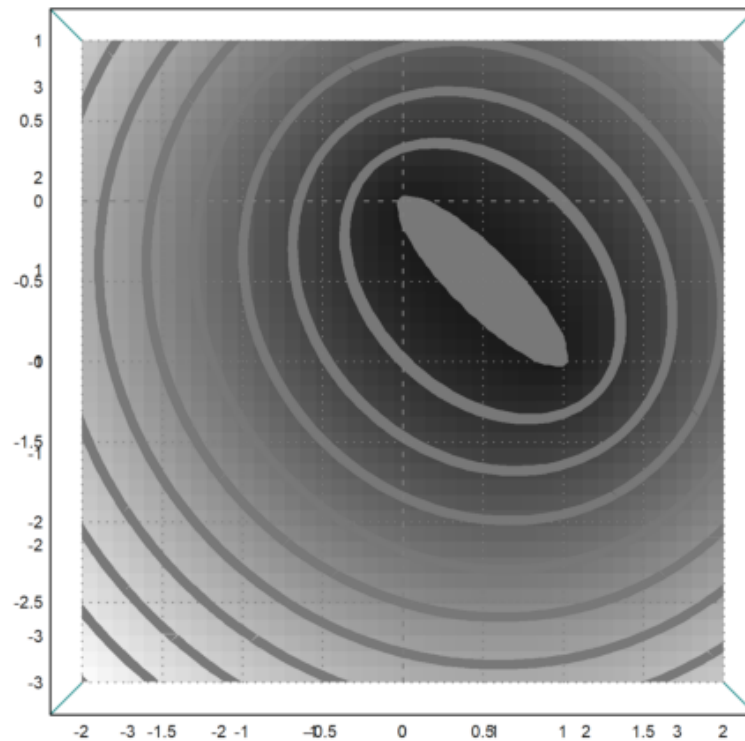
16

```
>AD.BC
```

16

```
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):
```



Terlihat bahwa,

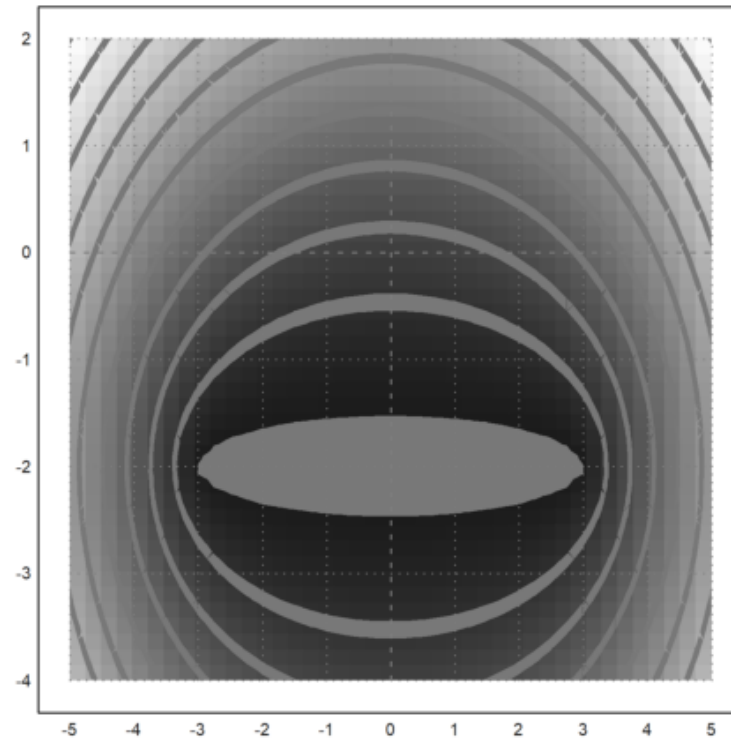
Sehingga terbukti bahwa
 $AB \times CD = 16$, $AD \times BC = 16$
 $AB \times CD, AD \times BC$

Karena semua syarat telah terpenuhi, maka dapat dipastikan bahwa segiempat tersebut merupakan segiempat garis singgung

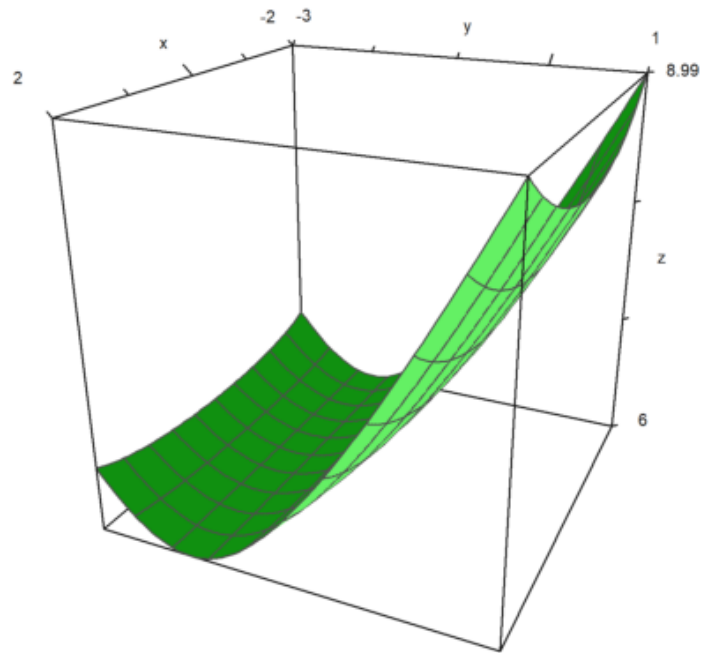
>

4. Gambarkanlah 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).

```
>
>P=[-3,-2]; Q=[3,-2];
>function d1(x,y):=sqrt((x-P[1])^2+(y-P[2])^2)
>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=-5,xmax=5,ymin=-4,ymax=2,hue=1):
```

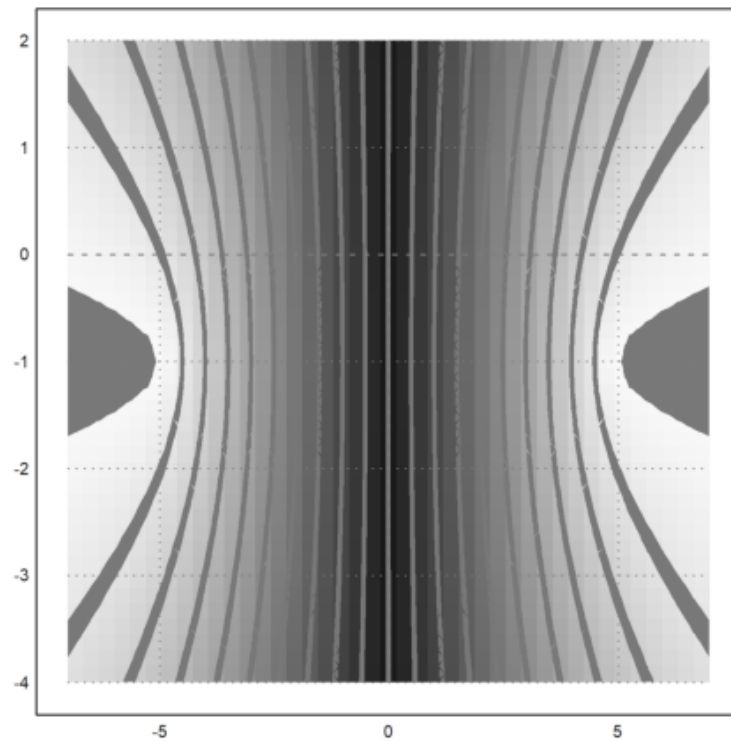


```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```

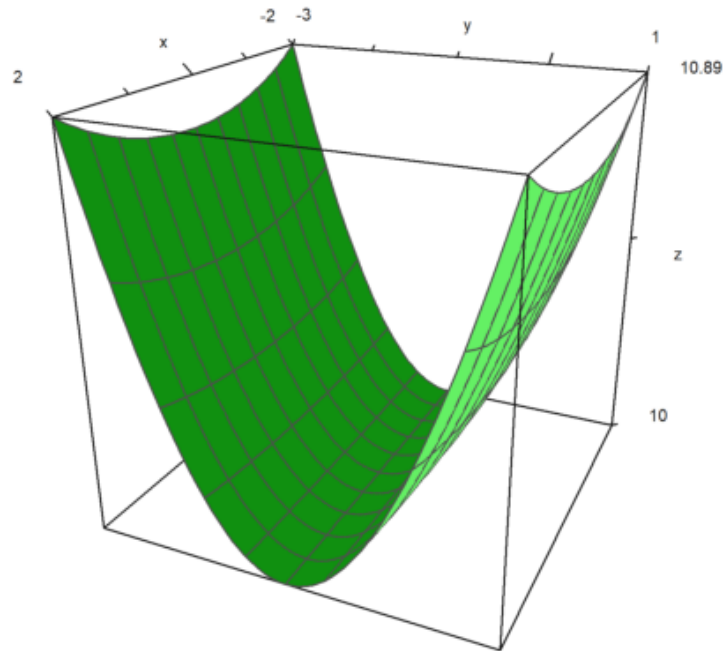


5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

```
>P=[-5,-1]; Q=[5,-1];  
>function d(x,y):=abs(sqrt((x-P[1])^2+(y-P[2])^2) - sqrt((x-Q[1])^2+(y-Q[2])^2))  
>fcontour("d",xmin=-7,xmax=7,ymin=-4,ymax=2,hue=2):
```



```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```



```
>plot2d("abs(x+1)+abs(x-1)",xmin=-3,xmax=3):
```

