

Nama :Rahma Amalia

NIM :24030130028

Kelas :Pendidikan Matematika C 2024

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.
```

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 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 membentuk

suatu segitiga

doublespread(sa): Spread sudut rangkap Spread 2ϕ , dengan

$sa = \sin(\phi)^2 \text{ spread } a.$

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

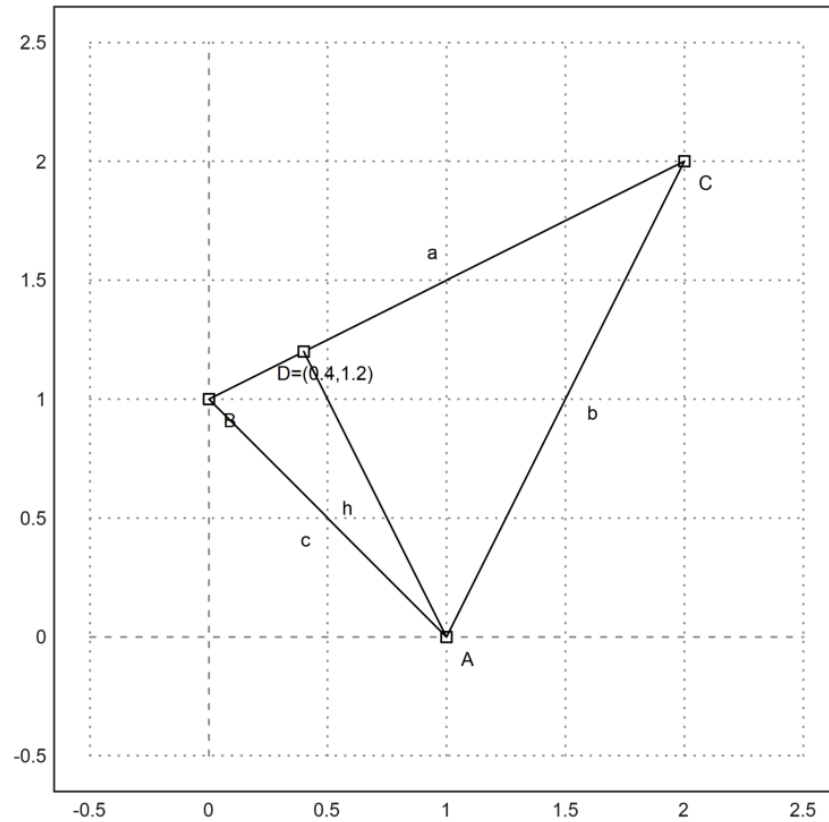
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

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 segitiga 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))
```

$36^{\circ}52'11.63''$

b. Sudut B

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

$71^{\circ}33'54.18''$

c. Sudut A

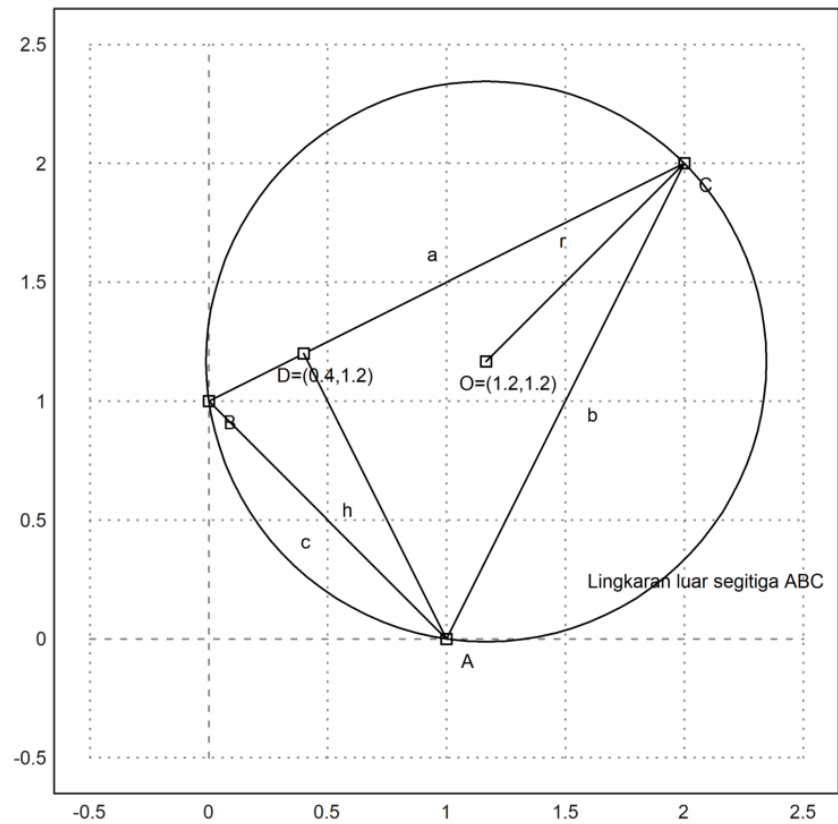
```
>degprint(computeAngle(C,A,B))
```

$71^{\circ}33'54.18''$

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

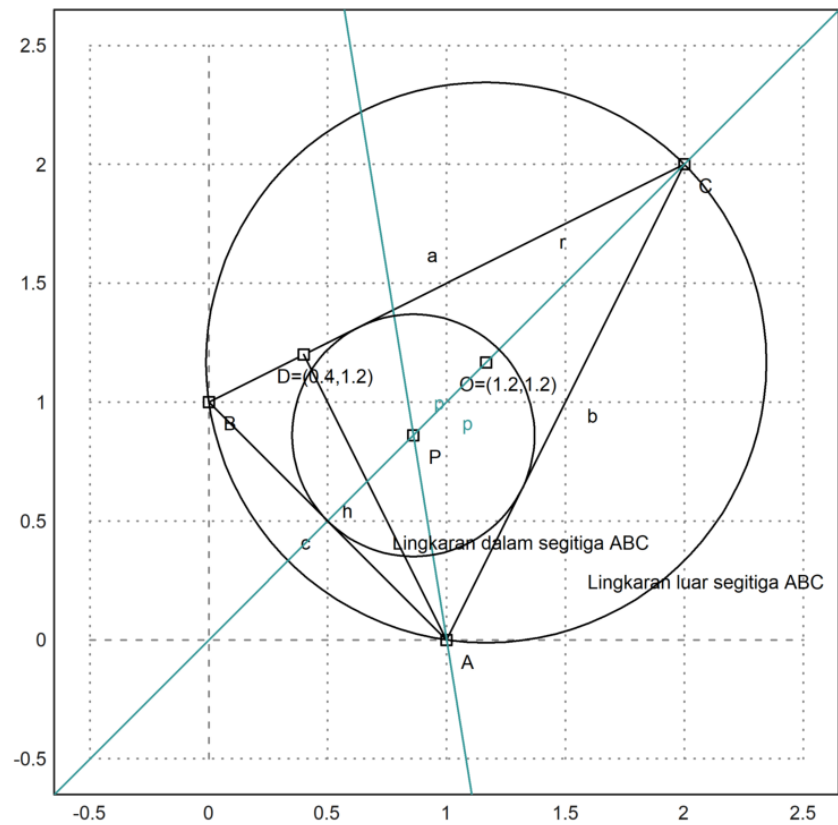
```
>color(5); plotLine(l); plotLine(g); color(1);
```

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
```

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

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

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)
```

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

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

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
```

$$1.178511301977579$$

```
>$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);
```

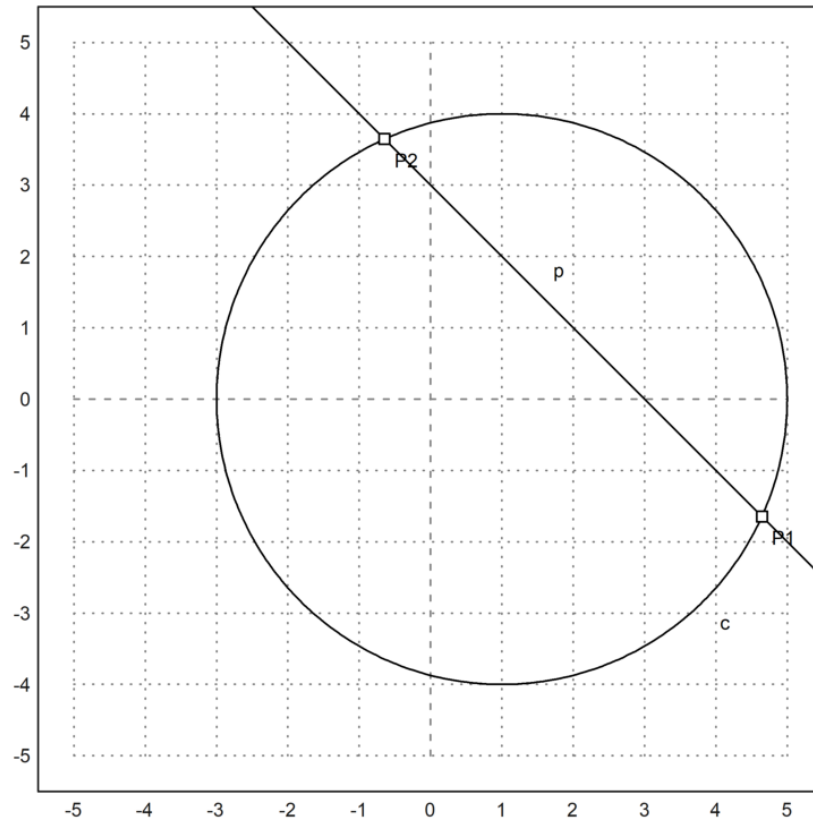
Perpotongan garis dengan lingkaran menghasilkan dua titik dan jumlah titik perpotongan.

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

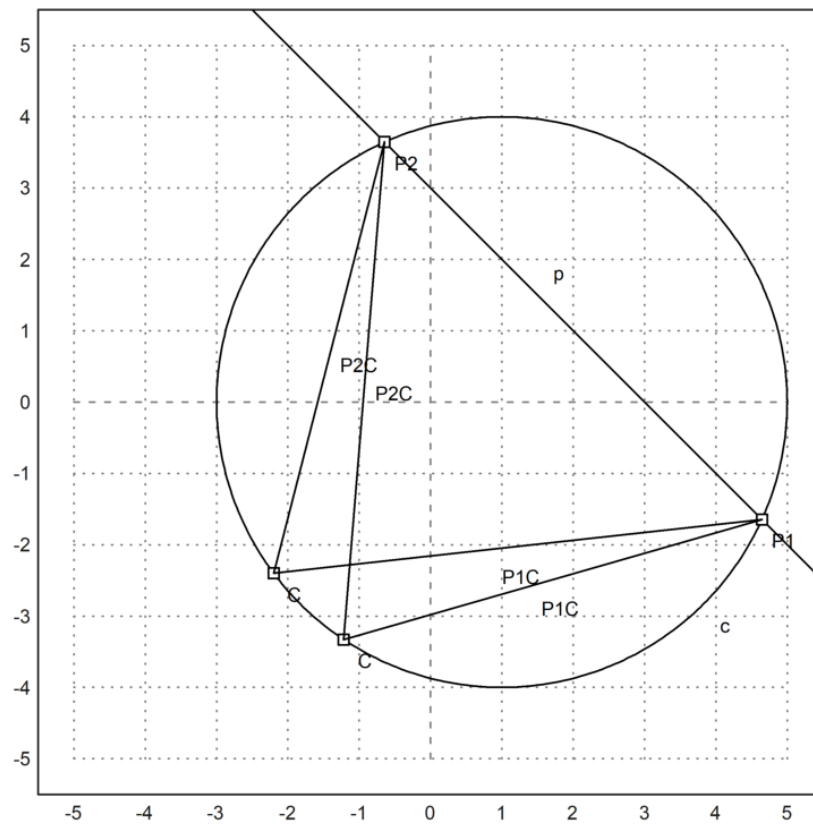
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);  
>degprint(computeAngle(P1,C,P2))
```

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

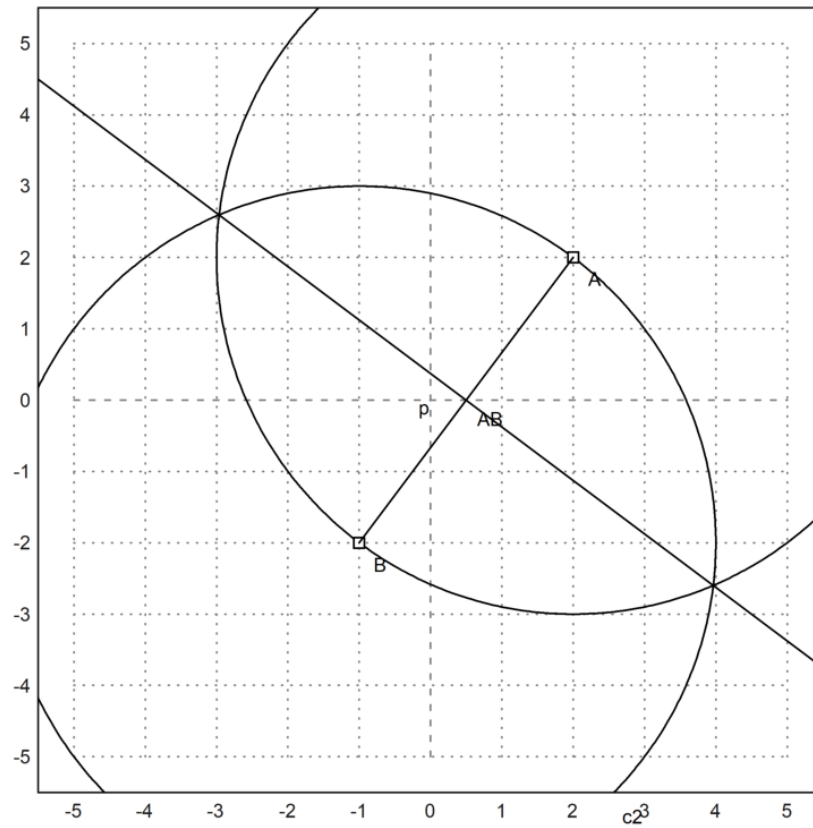
```
>l=lineThrough(P1,P2);
```

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

DEFINISI 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:

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

Berikut adalah bentuk untuk mencari luas segitiga dengan Rumus Heron

Perintah yang digunakan untuk mencari x dan y

```
>assume(a>0)
```

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

Memasukkan rumus simbolik tinggi segitiga (y) dalam bentuk akar ke fungsi ysol

```
>ysol &= y with sol[2][2]
```

Membuat fungsi luas segitiga (Rumus Heron) dari hasil simbolik tinggi segitiga (ysol)

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

Mencari luas segitiga dengan rumus heron

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

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

$$\begin{aligned} & \frac{-c + b + a}{2a}, y = \frac{\sqrt{-c^4 + 2b^2c^2 + 2a^2c^2 - b^4 + 2a^2b^2 - a^4}}{2a} \\ & [x = \frac{-c^2 + b^2 + a^2}{2a}, y = \frac{\sqrt{-c^4 + 2b^2c^2 + 2a^2c^2 - b^4 + 2a^2b^2 - a^4}}{2a}] \end{aligned}$$

```
>yso1 &= y with so1[2][2]; $'y=sqrt(factor(yso1^2))
```

$$y = \frac{\sqrt{(-c+b+a)(c-b+a)(c+b-a)(c+b+a)}}{2a}$$

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

$$H(a, b, c) = \frac{\sqrt{(-c + b + a)(c - b + a)(c + b - a)(c + b + a)}}{4}$$

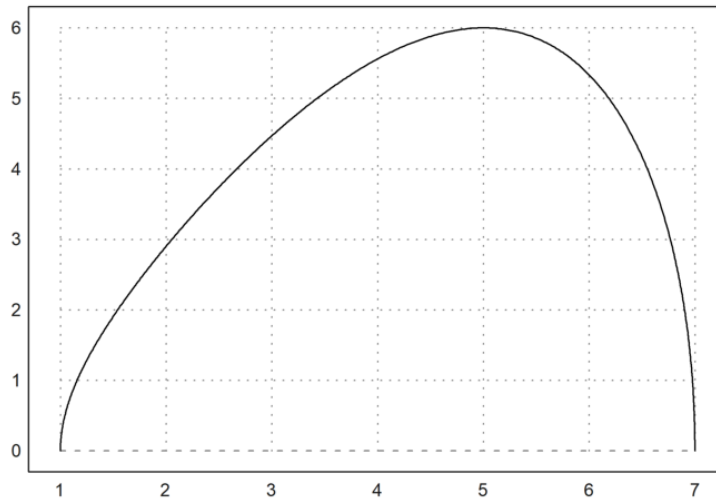
Contoh 1

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

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

6

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



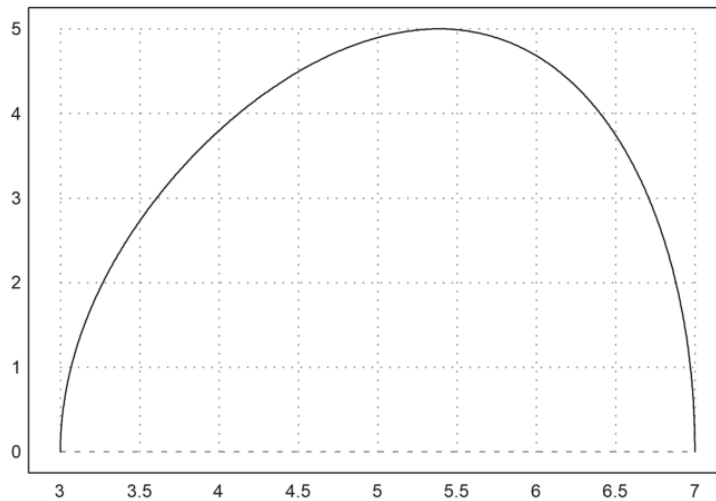
Contoh 2

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

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

4.6837484988

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



Contoh 4: Garis Euler dan Parabola

DEFINISI GARIS EULER

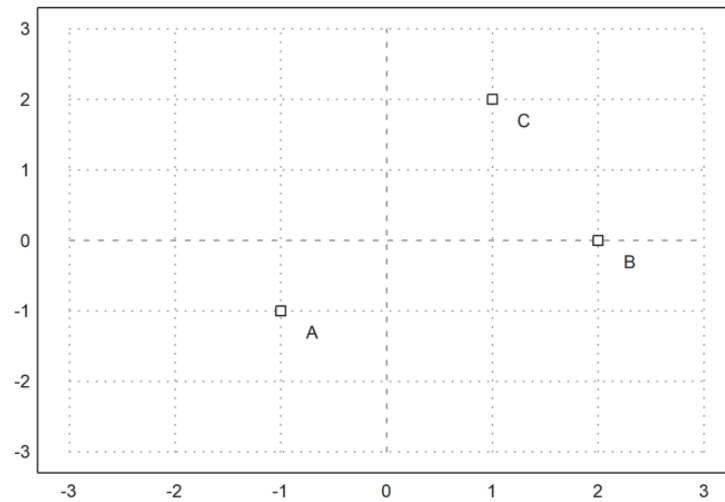
Garis Euler adalah garis yang ditentukan dari segitiga apa pun yang bukan sama sisi. Garis ini merupakan garis pusat segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga tersebut, termasuk orthocenter, circumcenter, centroid, titik Exeter, dan pusat lingkaran sembilan titik pada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

1. Mendefinisikan titik-titik segitiga

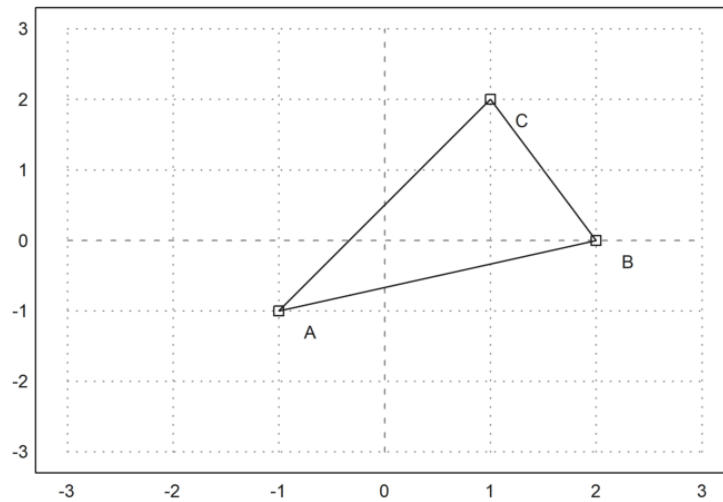
```
>A:=[-1,-1]; B:=[2,0]; C:=[1,2];
```

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

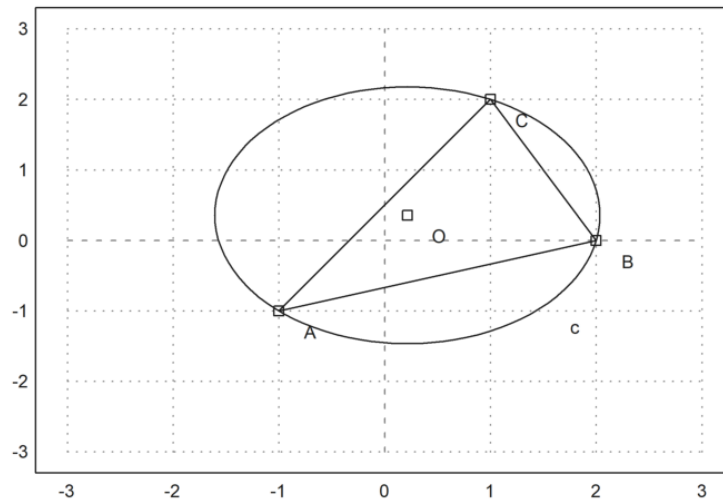
```
>//lingkaran yang melalui A, B, C  
>LL &= circleThrough(A,B,C); $getCircleEquation(LL,x,y)
```

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

```
>//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
```

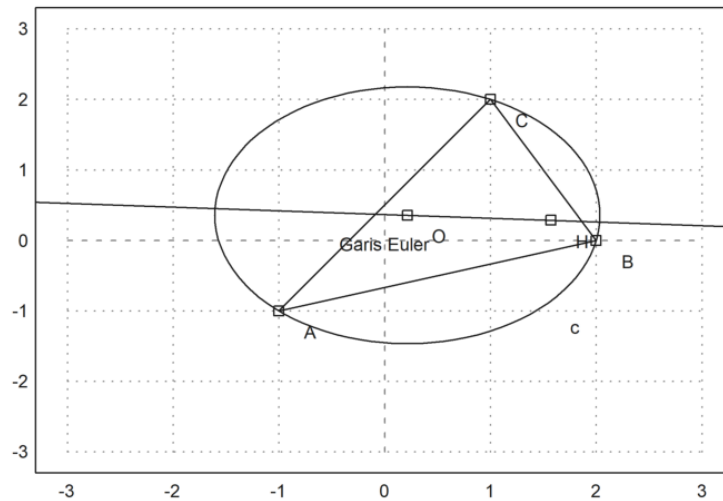
$$\left[\frac{11}{7}, \frac{2}{7} \right]$$

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"):
```

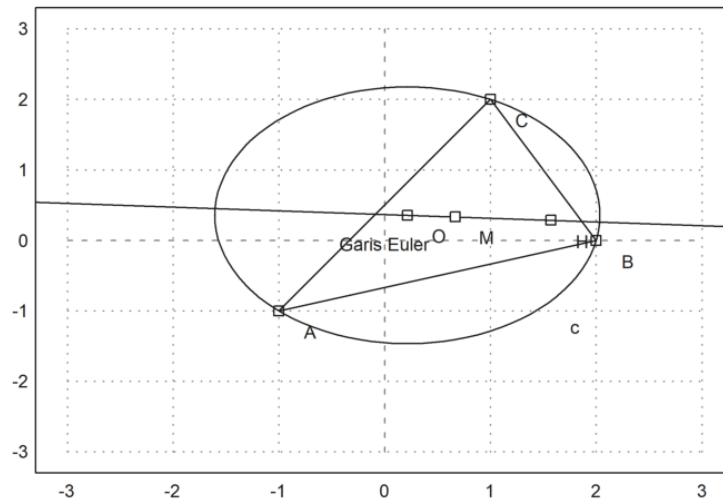


7. Menentukan Titik Berat Segitiga

```
>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(%())
```

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

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

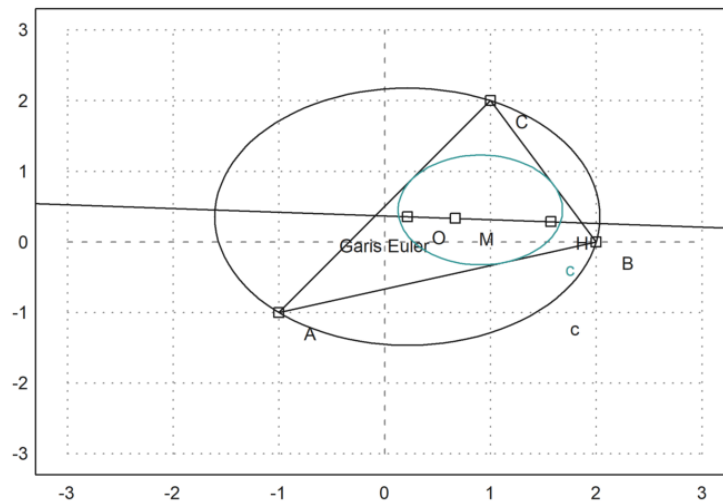
```
>Q &= lineIntersection(angleBisector(A,C,B),angleBisector(C,B,A))|radcan; $Q
```

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

```
>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());
```



1. Menentukan titik dan garis

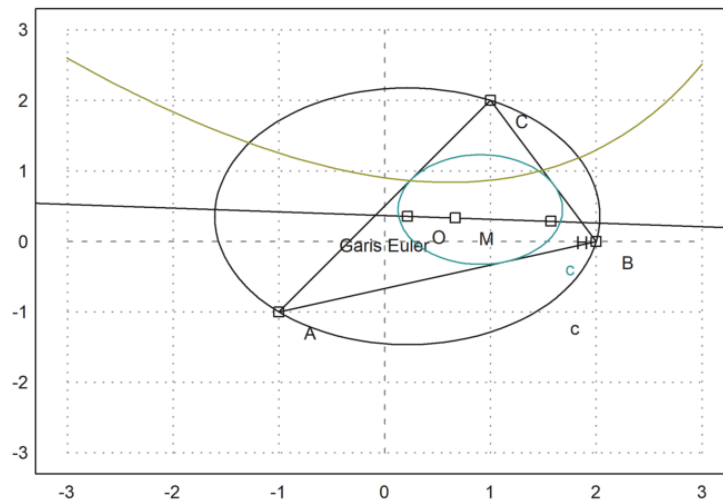
Mencari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

```
>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

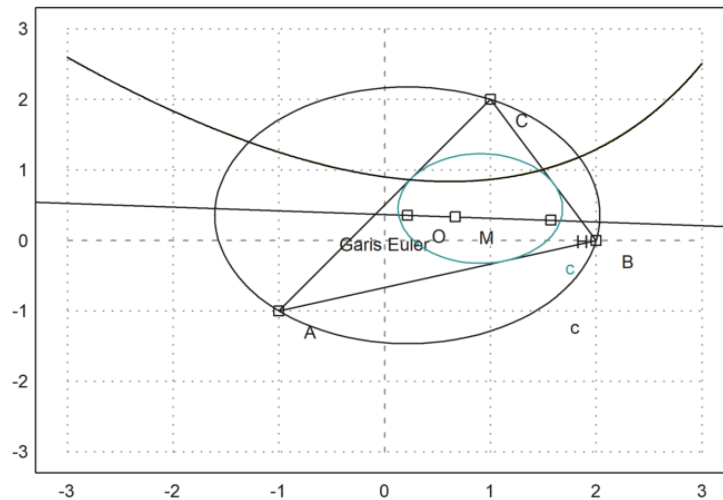
Dikarenakan Maxima kesulitan dengan akar, maka persamaan dikuadratkan:

```
>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):
```



Contoh 5: Trigonometri

Trigonometri rasional diperkenalkan oleh N.J. Wildberger untuk menghindari penggunaan fungsi trigonometri klasik (seperti $\sin$, $\cos$, $\tan$) yang menghasilkan bilangan irasional dan pengukuran sudut dalam derajat/radian yang kompleks.

Konsep ini menggunakan:

- Quadrance (kuadrat jarak) menggantikan jarak.
- Spread (kuadrat sinus sudut) menggantikan sudut.

Quadrance (Kuadrat Jarak) Quadrance adalah kuadrat dari jarak

antara dua titik.

Jarak antara dua titik A dan B didefinisikan sebagai:

$$Q(A, B) = (x_2 - x_1)^2 + (y_2 - y_1)^2$$

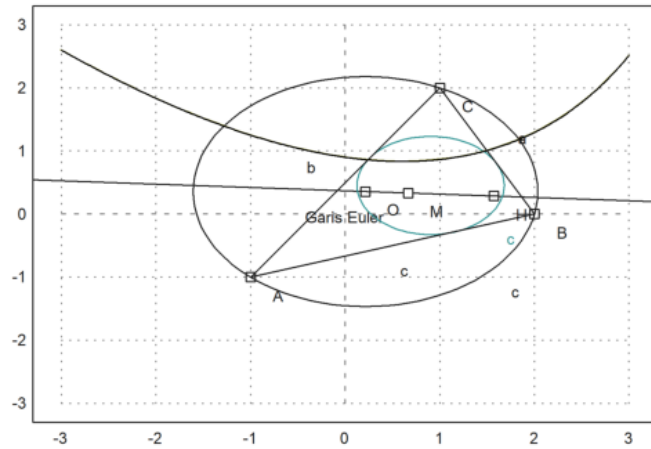
Teorema Pythagoras dalam bentuk quadrance:

$$Q_1 + Q_2 = Q_3$$

Contoh:

Definisikan titik sudut segitampilkan plotnya.

```
>load geometry;  
>plotSegment(B,A,"c"); plotSegment(A,C,"b"); plotSegment(C,B,"a"); ...  
>insimg(30);
```



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

9

16

25

```
>& a+b=c
```

25 = 25

Spread adalah kuadrat dari sinus sudut antara dua garis.

Definisi:

$$s = \sin(\theta)^2$$

Sifat-sifatnya:

$$0 \leq s \leq 1$$

$$s = 0 \rightarrow \text{garis sejajar}$$

$$s = 1 \rightarrow \text{garis tegak lurus}$$

Spread dalam segitiga siku-siku:

Pada segitiga siku-siku dengan sisi siku-siku a , sisi siku-siku b , dan sisi miring c :

$$s = \frac{Q_a}{Q_c} = \frac{a^2}{c^2}$$

Contoh (Segitiga 3-4-5):

```
>a = 3^2; b = 4^2; c = 5^2;  
>sA &= a/c; $sA
```

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

```
>sB &= b/c; $sB
```

$$\frac{16}{25}$$

```
>sC = 1    // spread di sudut C (siku-siku)
```

1

Cross Law (Rumus Silang)

Cross Law adalah versi rasional dari Hukum Cosinus.

Rumus Cross Law:

Untuk segitiga dengan quadrance sisi Q_1, Q_2, Q_3 dan spread s di hadapan sisi Q_1 :

$$(Q_2 + Q_3 - Q_1)^2 = 4Q_2Q_3(1 - s_1)$$

Hubungan dengan Hukum Cosinus:

-Klasik:

$$c^2 = a^2 + b^2 - 2ab \cos \gamma$$

-Rasional:

$$(Q_a + Q_b - Q_c)^2 = 4Q_aQ_b(1 - s_c)$$

```
>Qa=9; Qb=16; Qc=25;  
>sc=1;  
>LHS=(Qa+Qb-Qc)^2
```

0

```
>RHS=4*Qa*Qb*(1-sc)
```

0

```
>LHS==RHS //Cek kebenaran
```

1

```
>$crosslaw(25,16,9,1)
```

$$0 = 0$$

Spread Law (Teorema Perbandingan)

Spread Law adalah versi rasional dari Hukum Sinus.

Rumus:

$$\frac{Q_1}{s_1} = \frac{Q_2}{s_2} = \frac{Q_3}{s_3}$$

Contoh:

```
>//Segitiga dengan Q1=9, Q2=16, Q3=25  
>s1=9/25; s2=16/25; s3=1;  
>9/s1
```

25

```
>16/s2
```

25

```
>25/s3
```

25

Triple Spread Formula (Identitas Tiga Kuadrat Sinus)

Triple Spread Formula adalah identitas/hubungan antara tiga kuadrat sinus dalam sebuah segitiga.
Triple Spread Formula menggantikan rumus jumlah sudut

$$\alpha + \beta + \gamma = 180^\circ.$$

Rumus:

Untuk tiga spread s_1 , s_2 , s_3 dalam segitiga:

$$(s_1 + s_2 + s_3)^2 = 2(s_1^2 + s_2^2 + s_3^2) + 4s_1s_2s_3$$

Contoh (Segitiga 3-4-5):

```
>s1=9/25; s2=16/25; s3=1;  
>LHS=(s1+s2+s3)^2
```

4

```
>RHS=2*(s1^2+s2^2+s3^2)+4*s1*s2*s3
```

4

```
>LHS==RHS
```

```
>$triplespread(9/25,16/25,1)
```

$$4 = 4$$

Double Spread Formula (Rumus Sinus Rangkap)

Double Spread Formula adalah rumus untuk sudut rangkap dalam trigonometri rasional.
Rumus:

Jika $s = \sin^2(\theta)$, maka:

$$s(2\theta) = 4s(1 - s)$$

Contoh:

```
>s=1/2;  
>s2=4*s*(1-s)
```

1

```
>$doublespread(1/2)
```

1

Chord Angle (Sudut Tali Busur)

Sudut tali busur adalah sudut yang dibentuk oleh dua tali busur dalam lingkaran.

Teorema sudut keliling:

Besarnya sudut keliling sama dengan setengah sudut pusat yang menghadap busur yang sama.

$$s_{\text{keliling}} = \frac{s_{\text{tali busur}}}{4R^2}$$
$$\text{atau } s_{\text{keliling}} = \frac{Q_{\text{tali busur}}}{4R^2}$$

Jika dua tali busur memiliki panjang sama, maka:

s_{keliling} = sama untuk semua titik pada busur yang sama

Contoh:

```
>R=1; $R
```

1

```
>L=1; $L
```

1

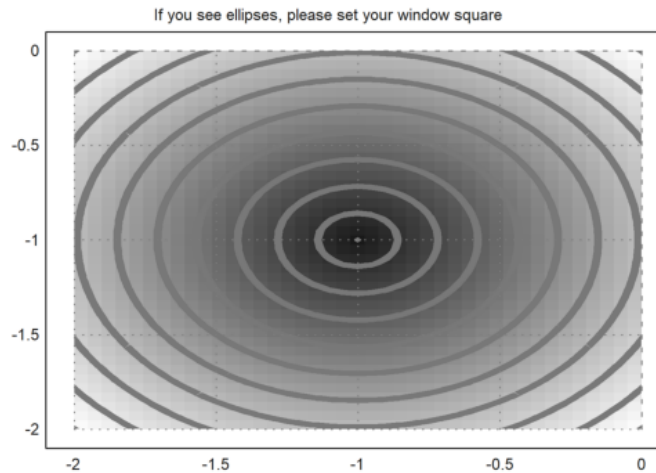
```
>Qtalibusur=L^2;  
>skeliling=Qtalibusur
```

1

Contoh 6: Jarak Minimal pada Bidang Fungsi yang, pada sebuah titik

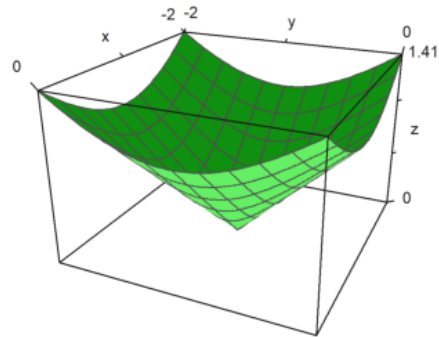
M pada bidang, memberikan jarak AM antara titik tetap A dan M, memiliki garis-garis tingkat yang cukup sederhana: lingkaran yang berpusat di A.

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



dan grafiknya juga cukup sederhana: bagian atas kerucut:

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

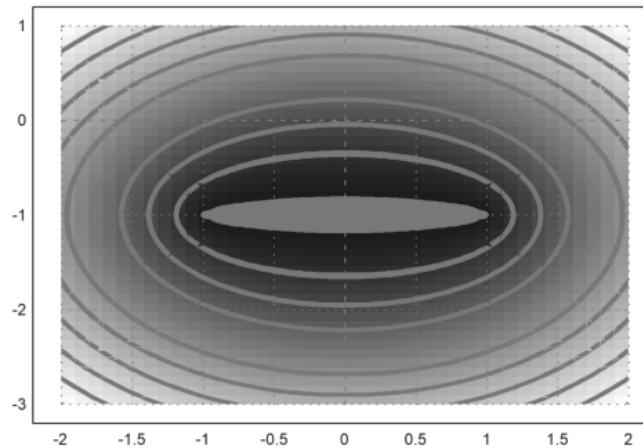


Tentu saja nilai minimum 0 dicapai di A.

Dua titik

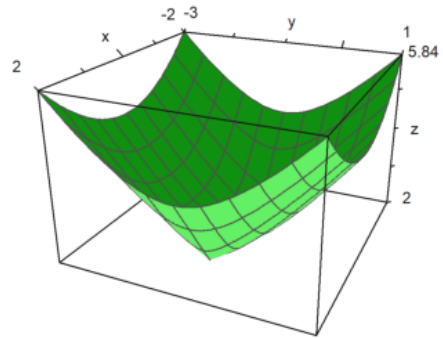
Sekarang kita lihat fungsi $MA+MB$ di mana A dan B adalah dua titik (tetap). Ini adalah "fakta yang terkenal" bahwa kurva tingkat adalah elips, titik fokusnya adalah A dan B ; kecuali AB minimum yang konstan pada segmen $[AB]$:

```
>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):
```



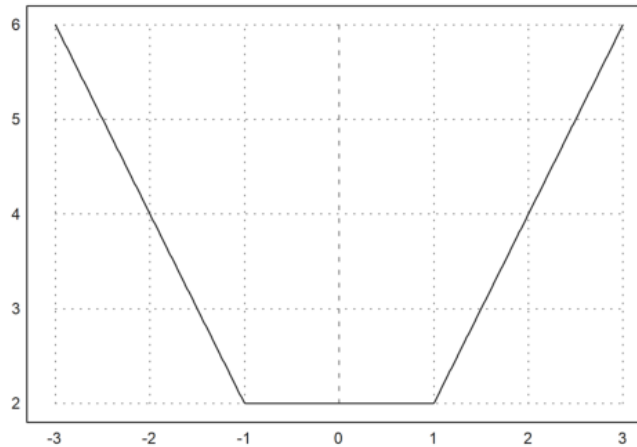
Grafiknya lebih menarik:

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



Pembatasan pada garis (AB) lebih terkenal:

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



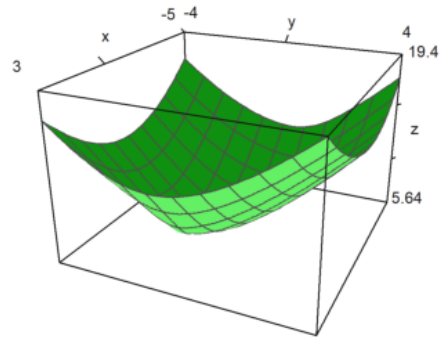
Tiga titik

Sekarang, hal-hal menjadi tidak terlalu sederhana: Hal ini sedikit kurang dikenal bahwa $MA+MB+MC$ mencapai minimum pada satu titik bidang, tetapi untuk menentukannya kurang sederhana:

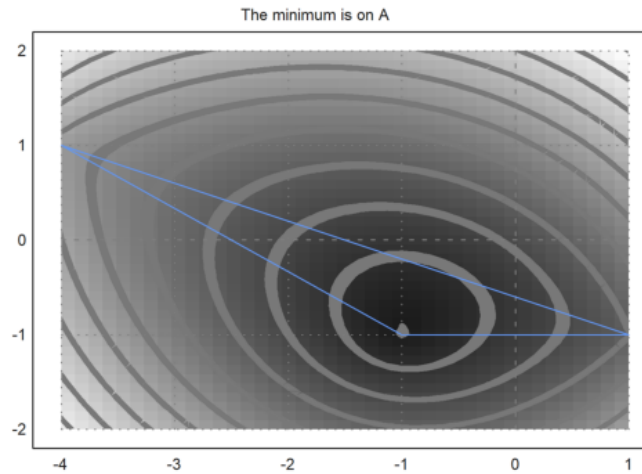
1) Jika salah satu sudut segitiga ABC lebih dari 120° (katakanlah di A), maka minimum dicapai pada titik ini (katakanlah AB+AC).

Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
>insimg;
```

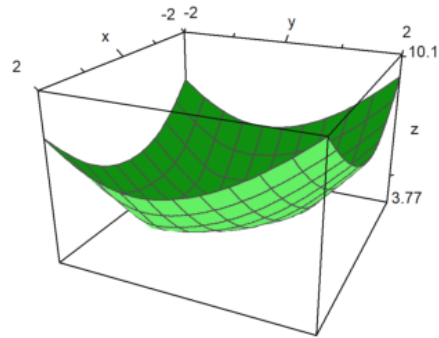


```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");  
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);  
>insimg;
```



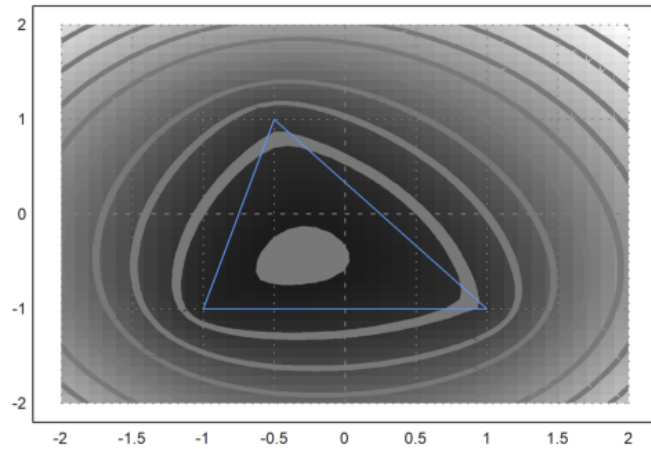
2) Tetapi jika semua sudut segitiga ABC kurang dari 120° , minimumnya adalah pada titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (masing-masing 120°):

```
>C=[-0.5,1];  
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):
```

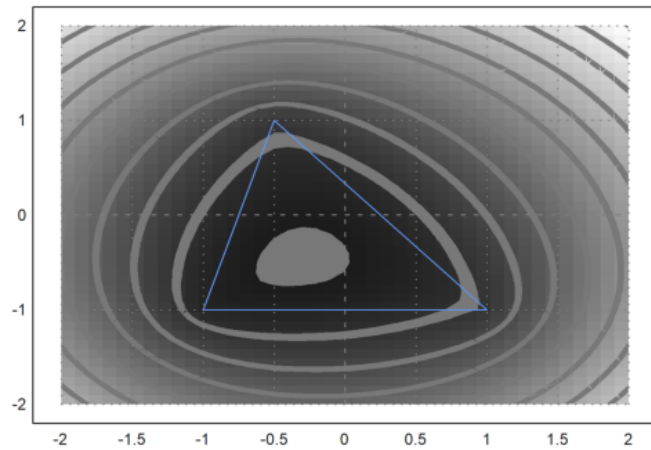


```
>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");  
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);  
>insimg:
```

The Fermat point

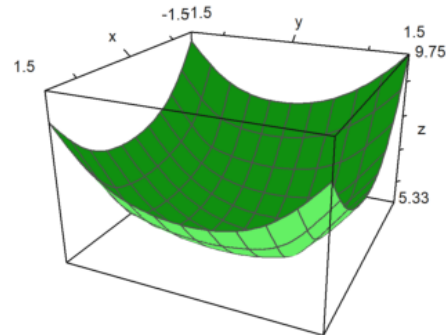


The Fermat point

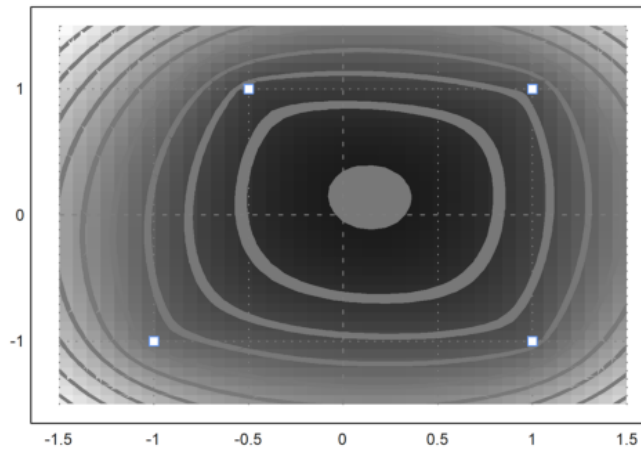


Langkah selanjutnya adalah menambahkan titik ke-4 D dan mencoba meminimalkan $MA+MB+MC+MD$; misalkan Anda adalah operator TV kabel dan ingin menemukan di bidang mana Anda harus meletakkan antena Anda sehingga Anda dapat memberi ke empat desa dan menggunakan panjang kabel sesedikit mungkin!

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



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



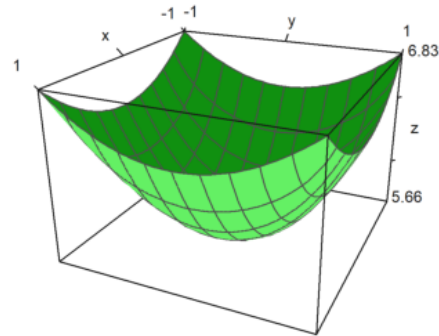
Masih ada nilai minimum dan tidak ada simpul A, B, C atau D yang tercapai:

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

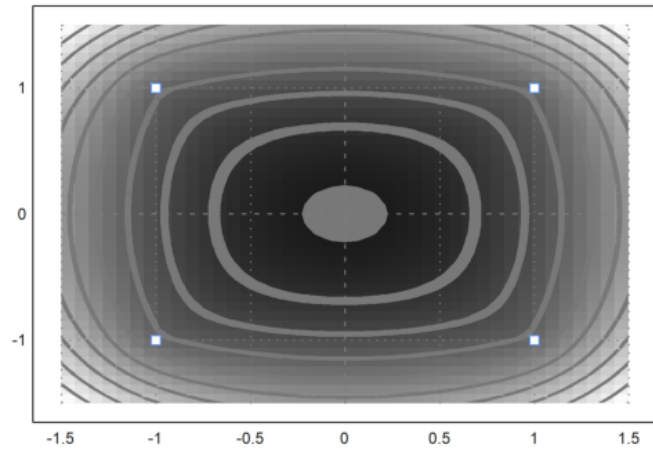
[0.142858, 0.142857]

Tampaknya dalam kasus ini, koordinat titik optimal adalah rasional atau mendekati rasional... Karena ABCD adalah sebuah bujur sangkar, maka kita berharap bahwa titik optimalnya adalah pusat dari 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

Bola Dandelin adalah satu/dua bola yang bersinggungan dengan suatu bidang dan kerucut yang memotong bidang tersebut. Perpotongan antara bidang dan kerucut tersebut merupakan "irisan" kerucut, dan terdapat titik di dalamnya yang menyentuh bola-bola tersebut, titik-titik ini disebut fokus irisan kerucut. karena itulah bola dandelin juga terkadang disebut sebagai bola fokus.

Menggambar Kerucut

Pada gambar yang terbentuk di bawah ini, akan terlihat bahwa dibutuhkan dua lingkaran yang menyentuh dua garis yang membentuk sebuah kerucut, dan akan terdapat suatu garis yang membentuk bidang yang memotong kerucut.

Pertama-tama, panggil berkas geometry.e

```
>load geometry;
```

Setiap kerucut, umumnya bila dilihat dari depan (alasnya sebagai axis) akan terlihat seperti segitiga sama kaki dengan puncaknya sebagai titik puncak segitiganya. Pada segitiga sama kaki, garis tinggi titik puncak dan garis sumbu terhadap sisi alasnya sama. Sehingga untuk mempermudah visualisasinya, kita dapat menjadikan sumbu tegak(Y) sebagai garis tingginya dan titik (0,0) sebagai titik puncaknya.

1. Membentuk dua garis pembentuk kerucut(dua garis tersebut harus saling berpotongan)

```
>g1 &= lineThrough([0,0],[1,q])
```

$[-q, 1, 0]$

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

$[-q, -1, 0]$

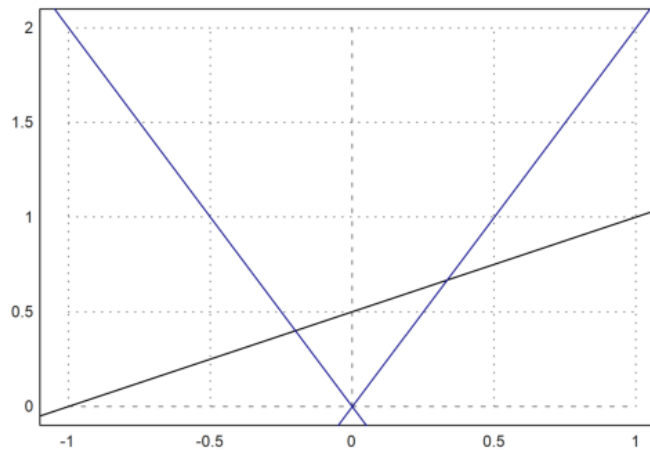
2. Bentuk garis ketiga yang memotong dua garis tersebut

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

$[-1, 2, 1]$

3. Buat plot garis (g1),(g2), dan (g)

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



4. Menggambar 2 lingkaran yang bersinggungan dengan garis (g1), (g2), dan (g). Ambil titik umum pada sumbu-Y, misal $P(0,u)$

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

[0, u]

Hitung jarak P terhadap garis (g1)

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

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

Hitung jarak P dengan garis (g)

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

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

Menentukan nilai u, dengan menggunakan sifat bahwa jarak antara P dengan (g1) sama dengan jaraknya ke(g)

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

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

Terdapat dua nilai u (misal u_1 & u_2)

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

```
[0.333333, 1]
```

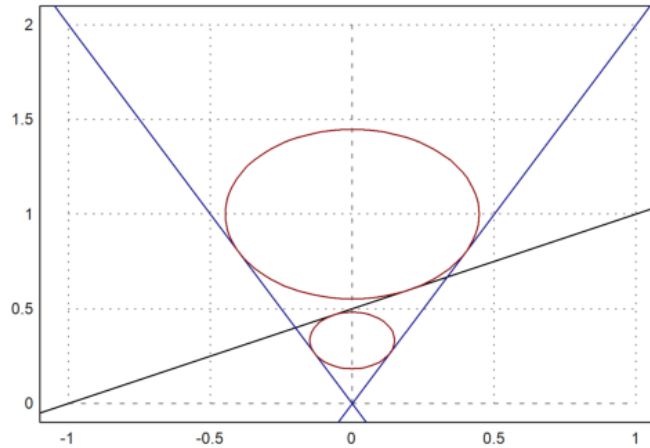
Didapati titik pusat lingkaran 1(0,3333) dan lingkaran 2(0,1)
Selanjutnya mencari nilai jari-jari lingkaran 1&2

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

```
[0.149071, 0.447214]
```

Menggambar di dimensi 2

```
>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 ini, dan jalankan 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

Persiapkan ruang model 3D

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

Selanjutnya Render 2 bola(bola 1 & bola 2) yang bersesuaian dengan lingkaran 1 & 2

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

Render kerucut yang bersesuaian dengan segitiga, agar objek-objek di dalamnya terlihat maka akan dibuat sedikit transparan

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

Render bidang yang sesuai dengan garis (g)

```
>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));
```

Render titik singgung antara bola dengan 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)));
```

Render titik singgung antara bola dengan bidang (g)

```
>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)));
```

Menghitung Persimpangan antara P1 & P2 terhadap bidang (g)

```
>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);
>writeln(povpoint(P5,povlook(yellow)));
```

Hubungkan semua titik-titik (singgung dan potong)

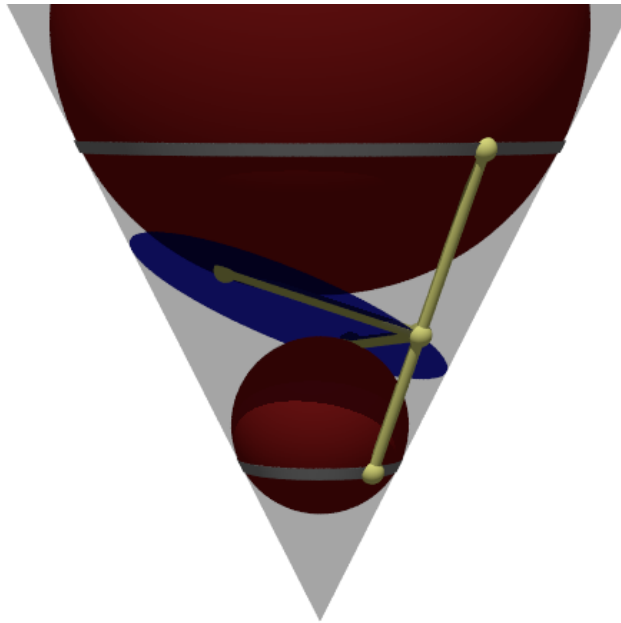
```
>writeln(povsegment(P1,P2,povlook(yellow)));  
>writeln(povsegment(P5,P3,povlook(yellow)));  
>writeln(povsegment(P5,P4,povlook(yellow)));
```

Render "band" pada bola 1&2 tempat bola-bola tersebut 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)));
```

Tampilkan hasil program Povray

```
>povend();
```



Untuk mendapatkan Anaglyph ini, kita perlu menempatkan semuanya ke dalam fungsi scene. Fungsi ini akan digunakan dua kali nanti.

Contoh 8: Geometri Bumi

Geometri Bumi adalah cabang dari geometri yang mempelajari bentuk, ukuran, dan sifat permukaan Bumi, serta bagaimana titik-titik di permukaannya diukur dan dihubungkan satu sama lain.

Dalam konteks ini, Bumi tidak dianggap sebagai bidang datar, melainkan sebagai benda bulat (sfera) atau elipsoid, sehingga perhitungan jarak, sudut, dan luas harus memperhitungkan kelengkungan Bumi.

Pada materi ini, kita akan melakukan beberapa komputasi bola. Fungsi-fungsi tersebut terdapat pada file "spherical.e" pada folder contoh. Kita perlu memuat file tersebut terlebih dahulu.

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

Untuk memasukkan posisi geografis, kita dapat menggunakan vektor dengan dua koordinat dalam radian (untuk utara dan timur bernilai positif, sedangkan untuk selatan dan barat bernilai negatif). Contoh: 7°28'51.4"S 110°10'47"E menjadi -7,-28.51 dan 110,10.47. Untuk mendapatkan titik koordinat suatu tempat, kita bisa menggunakan Google Earth

Berikut adalah koordinat Rumah A

```
>Rumah=[rad(-7,-28.51),rad(110,10.47)]
```

```
[-0.130466, 1.92291]
```

Kita dapat mencetak posisi ini dengan sposprint (cetak posisi bulat). Perintah sposprint digunakan untuk mengonversi titik pada bola ke posisi [lintang, bujur]

```
>sposprint(Rumah)
```

```
S 7°28.510' E 110°10.470'
```

Mari kita tambahkan dua kota lagi, misal Magelang dan Semarang.

```
>Magelang=[rad(-7,-28.47),rad(13,3.69)]; Semarang=[rad(-6,-59.05),rad(110,24.533)];  
>sposprint(Magelang), sposprint(Semarang),
```

```
S 7°28.470' E 13°3.690'  
S 6°59.050' E 110°24.533'
```

Pertama, kita menghitung vektor dari satu titik ke titik lainnya pada bola ideal. Vektor ini adalah [heading, jarak] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang 7° .

```
>br=svector(Magelang,Rumah); degprint(br[1]), br[2]*rearth(7°)->km,
```

```
98°22'53.67''  
10687.9289607
```

Ini merupakan sebuah pendekatan (aproksimasi) yang cukup baik. Rutin-rutin (fungsi/prosedur) berikut menggunakan pendekatan yang lebih baik lagi. Pada jarak yang begitu pendek, hasilnya hampir sama, yaitu perintah esdist. esdist digunakan untuk menghitung jarak antara dua titik di permukaan Bumi berdasarkan koordinat lintang (latitude) dan bujur (longitude). Perhitungannya mengasumsikan Bumi sebagai bola (spherical), tetapi menggunakan rata-rata jari-jari Bumi di antara kedua titik

```
>esdist(Magelang,Rumah)->" km",
```

```
10687.8548467 km
```

Ada sebuah fungsi untuk arah (heading) yang memperhitungkan bentuk elips Bumi. Sekali lagi, kita mencetak (hasilnya) dengan cara yang lebih cangguh/lanjutan.

```
>sdegprint(esdir(Rumah,Magelang))
```

```
261.62°
```

Sudut-sudut sebuah segitiga melebihi 180° pada permukaan bola. Dalam geometri bola, sebuah segitiga dibentuk oleh tiga busur lingkaran besar yang terletak di permukaan bola, bukan oleh tiga garis lurus seperti pada bidang datar. Karena permukaan bola melengkung, maka sifat-sifat segitiganya berbeda dengan segitiga dalam geometri Euclid(datar). Salah satu perbedaan paling mencolok adalah bahwa jumlah ketiga sudut dalam segitiga bola selalu lebih besar dari 180° . Hal ini terjadi karena setiap sisi segitiga bola melengkung ke arah pusat bola, sehingga pertemuan antarbusur menghasilkan sudut yang lebih besar dibandingkan segitiga datar.

```
>asum=sangle(Rumah,Magelang,Semarang)+sangle(Rumah,Magelang,Semarang)+sangle(Rumah,Magelang,Semarang)
```

```
1°21'49.09''
```

Perintah sangle digunakan untuk menghitung besar sudut ABC di titik B pada sebuah bola sempurna. Ini dapat digunakan untuk menghitung luas segitiga. Catatan: Untuk segitiga kecil, cara ini tidak akurat karena kesalahan pengurangan dalam asum-pi.

```
>esarea(Rumah,Magelang,Semarang)->" km^2"
```

```
357363.037841 km^2
```

Perintah esares akan menghitung luas segitiga di permukaan Bumi. Namun untuk segitiga yang kecil, hasil perhitungannya tidak terlalu akurat.

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

```
>v=svector(Rumah,Magelang); sposprint(saddvector(Rumah,v)), sposprint(Magelang),
```

```
S 7°28.470' E 13°3.690'
```

```
S 7°28.470' E 13°3.690'
```

Fungsi-fungsi ini mengasumsikan bola yang ideal. Hal yang sama di bumi.

```
>sposprint(esadd(Rumah,esdir(Rumah,Semarang),esdist(Rumah,Semarang))), sposprint(Semarang),
```

```
S 6°59.050' E 110°24.533'
```

```
S 6°59.050' E 110°24.533'
```

Misal jarak dari Tugu Jogja dan Monas

```
>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 sebuah aproksimasi yang bagus.

```
>esdist(Tugu,Monas)->" km",
```

```
431.565659488 km
```

Arah (heading) yang dihasilkan sama dengan arah yang dihitung di Google Earth.

```
>degprint(esdir(Tugu,Monas))
```

```
294°17'2.85''
```

Namun, kita tidak lagi memperoleh posisi target yang tepat jika kita menambahkan arah (heading) dan jarak ke posisi asal.

Hal ini terjadi karena kita tidak menghitung fungsi inversnya secara tepat, melainkan menggunakan pendekatan terhadap jari-jari Bumi sepanjang lintasan.

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

S 6°10.500' E 106°48.717'

Namun demikian, kesalahannya tidak besar.

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

S 6°10.500' E 106°48.717'

Tentu saja, kita tidak dapat berlayar dengan arah (heading) yang sama dari satu tujuan ke tujuan lainnya jika kita ingin menempuh jalur terpendek. Bayangkan, kita terbang ke arah timur laut (NE) mulai dari suatu titik di permukaan Bumi — maka kita akan berputar spiral menuju Kutub Utara. Lingkaran besar (great circle) tidak mengikuti arah yang tetap(konstan)!

Perhitungan berikut menunjukkan bahwa kita akan sangat melenceng dari tujuan yang sebenarnya jika kita menggunakan arah yang sama sepanjang perjalanan.

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

Sekarang kita menambahkan sepuluh kali jarak sepersepuluhnya, dengan menggunakan arah (heading) menuju Monas yang kita peroleh di Tugu.

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

Hasilnya jauh berbeda.

```
>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 lintang 30°, tetapi jalur yang lebih pendek yang dimulai 10° lebih jauh ke utara di P1.

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

79.69°

Tetapi, jika kita mengikuti arah kompas ini, kita akan berputar spiral menuju Kutub Utara! Karena itu, kita harus menyesuaikan arah (heading) kita sepanjang perjalanan. Untuk keperluan perkiraan kasar, kita menyesuaikannya setiap menempuh 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-jarak tersebut tidak akurat, karena kita akan menambah sedikit galat jika mengikuti arah (heading) yang sama terlalu lama.

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

0.203km

Kita mendapatkan perkiraan yang baik, jika kita menyesuaikan arah setiap 1/100 dari total jarak dari Tugu ke Monas.

```
>p=Tugu; dist=esdist(Tugu,Monas); ...  
> loop 1 to 100; p=esadd(p,esdir(p,Monas),dist/100); end;  
>skmprint(esdist(p,Monas))
```

0.000km

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS sepanjang lingkaran besar menuju Monas menggunakan fungsi navigate.

```
>load spherical; v=navigate(Tugu,Monas,10); ...  
> loop 1 to rows(v); sposprint(v[#]), end;
```

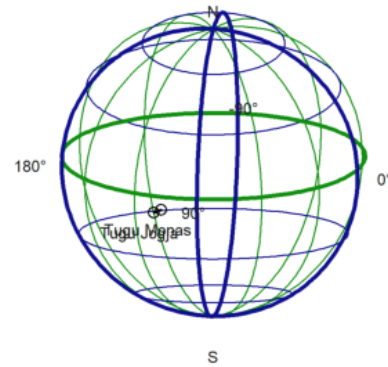
```
S 7°46.998' E 110°21.966'  
S 7°37.422' E 110°0.573'  
S 7°27.829' E 109°39.196'  
S 7°18.219' E 109°17.834'  
S 7°8.592' E 108°56.488'  
S 6°58.948' E 108°35.157'  
S 6°49.289' E 108°13.841'  
S 6°39.614' E 107°52.539'  
S 6°29.924' E 107°31.251'  
S 6°20.219' E 107°9.977'  
S 6°10.500' E 106°48.717'
```

Kami menulis sebuah fungsi, yang memetakan bumi, dua posisi, dan posisi di antaranya.

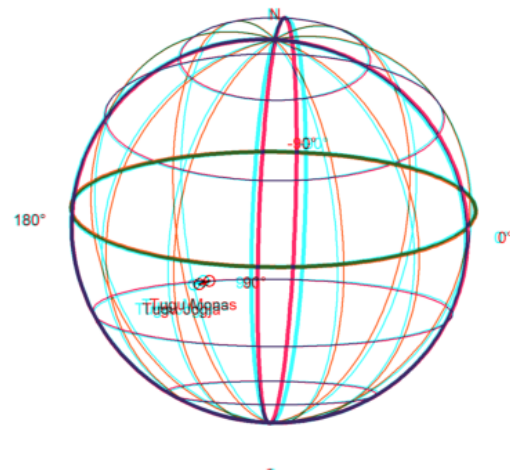
```
>function testplot ...
```

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

```
>plot3d("testplot",angle=25, height=6,>own,>user,zoom=3):
```



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



```
>reset();
```

1. Gambarlah segi-n beraturan jika diketahui titik pusat O, n, dan jarak titik pusat ke titik-titik sudut segi-n tersebut (jari-jari lingkaran luar segi-n), r.

Petunjuk:

- Besar sudut pusat yang menghadap masing-masing sisi segi-n adalah $(360/n)$.
- Titik-titik sudut segi-n merupakan perpotongan lingkaran luar segi-n dan garis-garis yang melalui pusat dan saling membentuk sudut sebesar kelipatan $(360/n)$.
- Untuk n ganjil, pilih salah satu titik sudut adalah di atas.
- Untuk n genap, pilih 2 titik di kanan dan kiri lurus dengan titik pusat.
- Anda dapat menggambar segi-3, 4, 5, 6, 7, dst beraturan.

2. Gambarlah suatu parabola yang melalui 3 titik yang diketahui.

Petunjuk:

- Misalkan persamaan parabolanya $y = ax^2 + bx + c$.
- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.
- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.

- Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung

(sisinya-sisintya merupakan garis singgung lingkaran yang sama yakni lingkaran dalam segi-4 tersebut).

- Suatu segi-4 merupakan segi-4 garis singgung apabila keempat

garis bagi sudutnya bertemu di satu titik.

- Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar

lingkaran dalamnya.

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

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

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

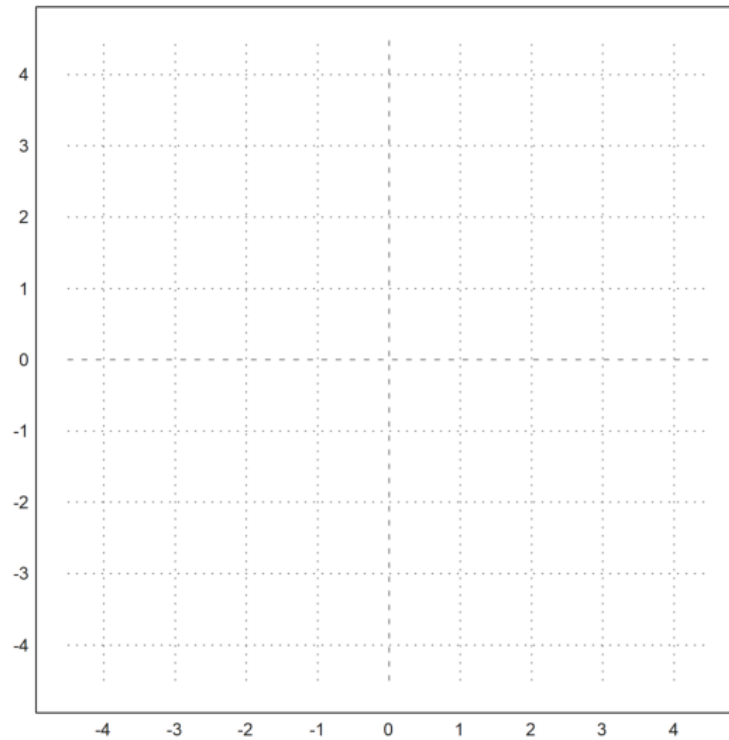
5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

```
>load geometry
```

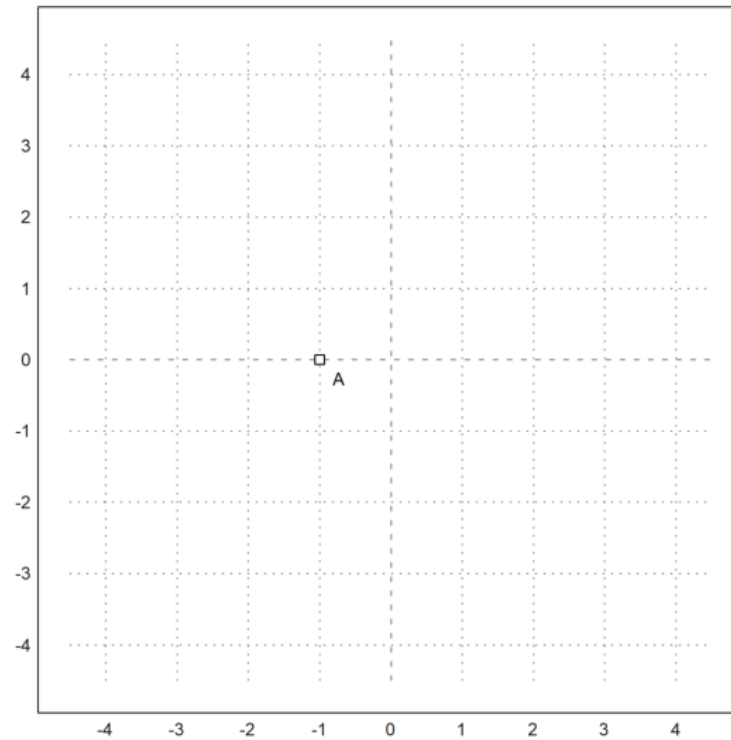
Numerical and symbolic geometry.

1. Menggambar 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 .

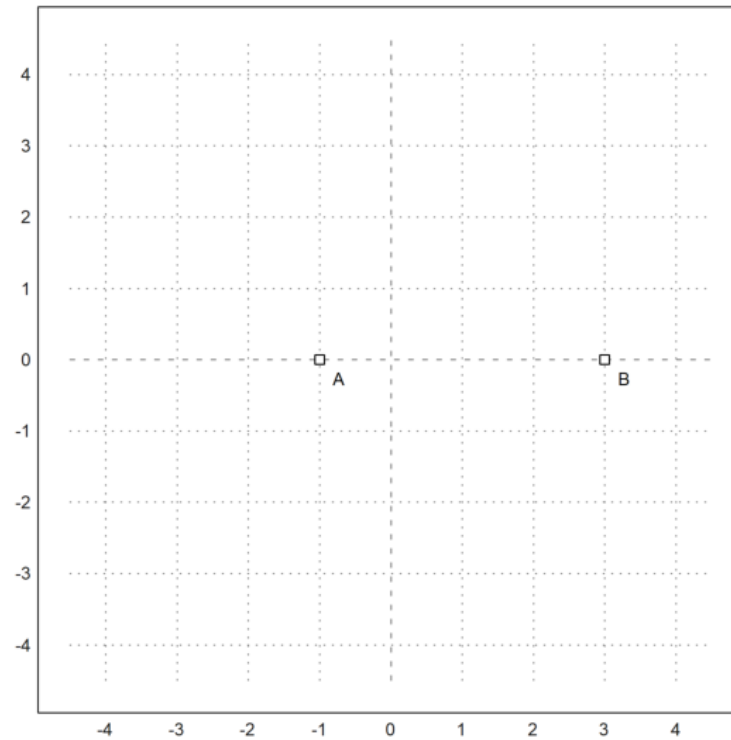
```
>setPlotRange(-4.5,4.5,-4.5,4.5):
```



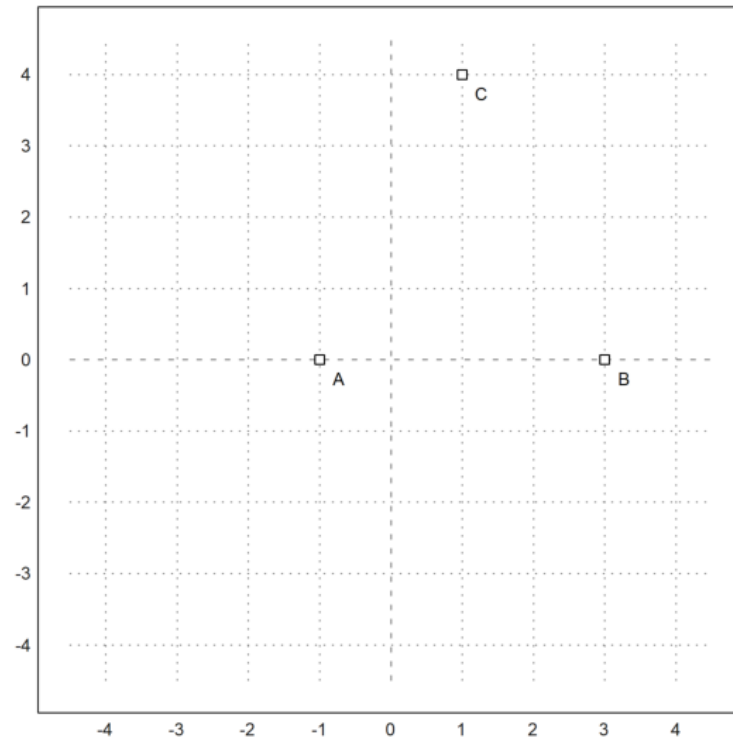
```
>A=[-1,0]; plotPoint(A,"A"):
```



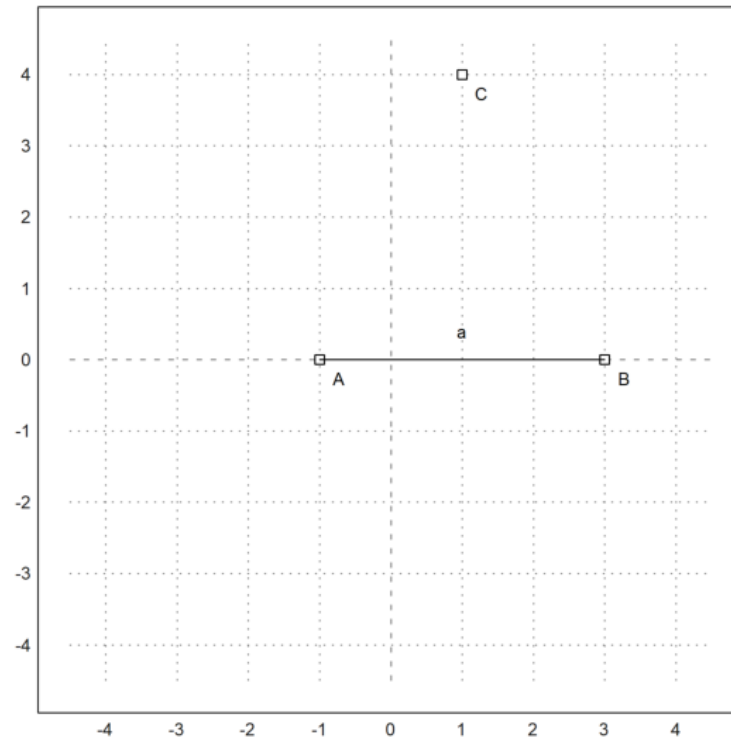
```
>B=[3,0]; plotPoint(B,"B"):
```



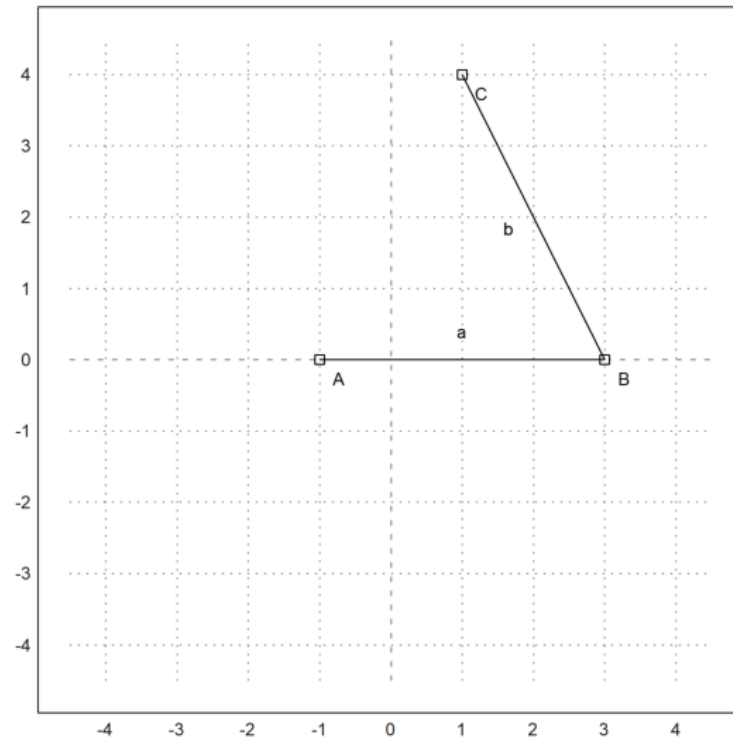
```
>C=[1,4]; plotPoint(C,"C"):
```



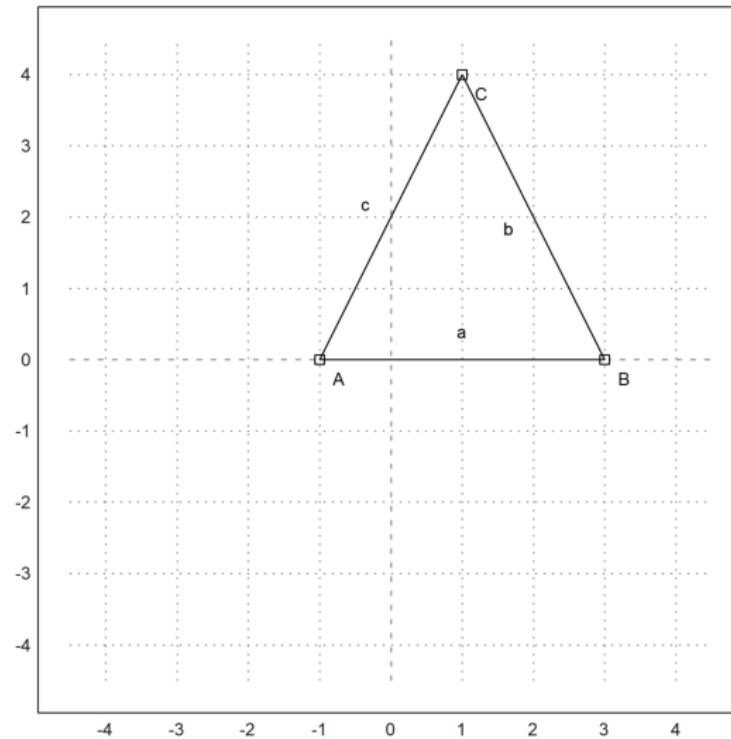
```
>plotSegment(A,B,"a"):
```



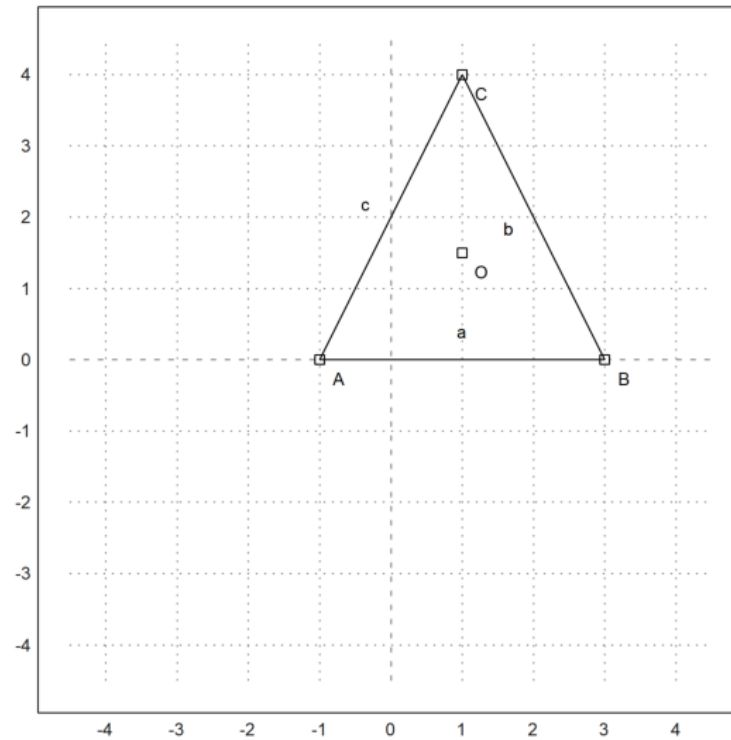
```
>plotSegment(B,C,"b"):
```



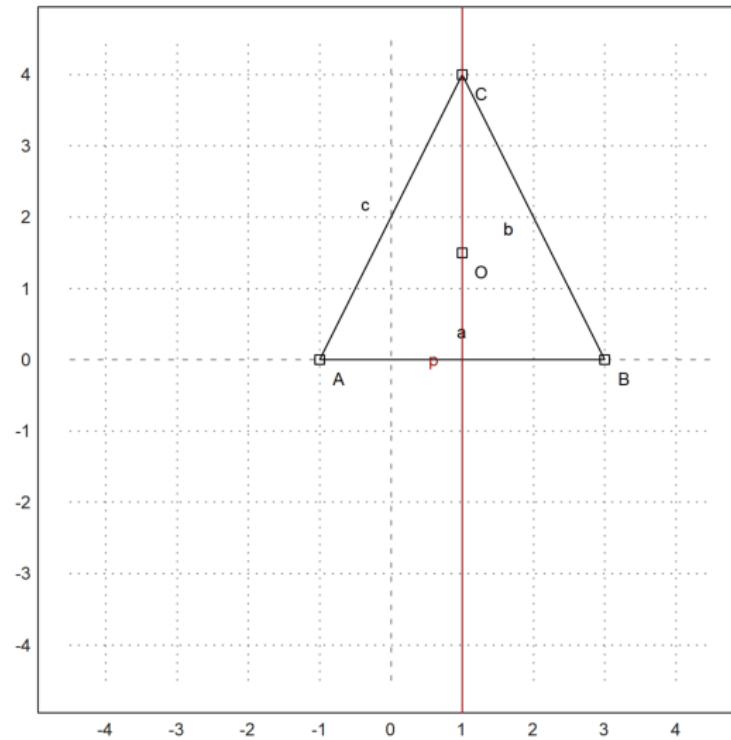
```
>plotSegment(A,C,"c"):
```



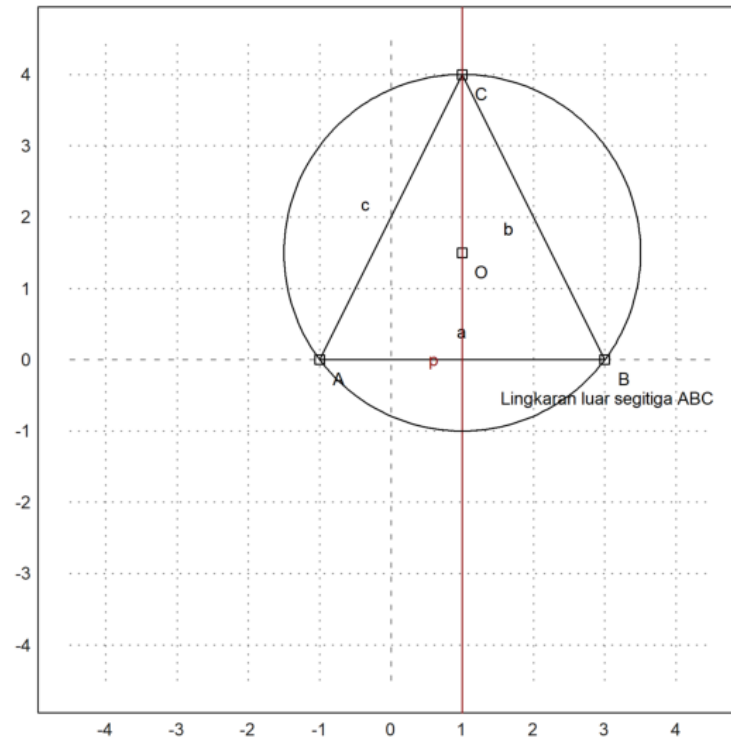
```
>c=circleThrough(A,B,C);  
>R=getCircleRadius(c);  
>O=getCircleCenter(c);  
>plotPoint(O,"O"):
```



```
>l=angleBisector(A,C,B);  
>color(2); plotLine(l); color(1):
```

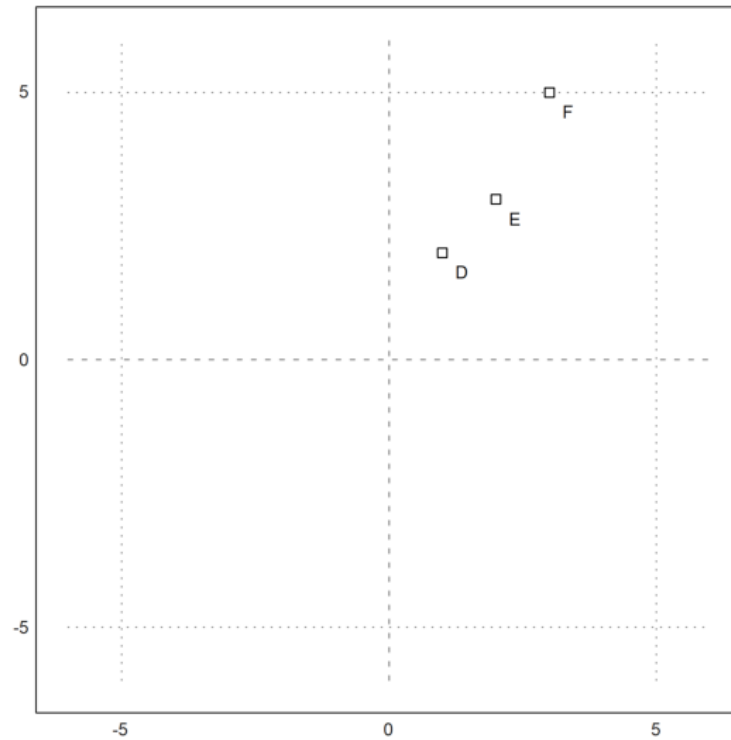


```
>plotCircle(c,"Lingkaran luar segitiga ABC"):
```



2. Menggambar suatu parabola yang melalui 3 titik yang diketahui.

```
>load geometry;
>setPlotRange(6); D=[1,2]; E=[2,3]; F=[3,5];
>plotPoint(D,"D"); plotPoint(E,"E"); plotPoint(F,"F"):
```



```
>sol &= solve([a+b+c=2,4*a+2*b+c=3,9*a+3*b+c=5],[a,b,c])
```

Maxima said:

solve: all variables must not be numbers.

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

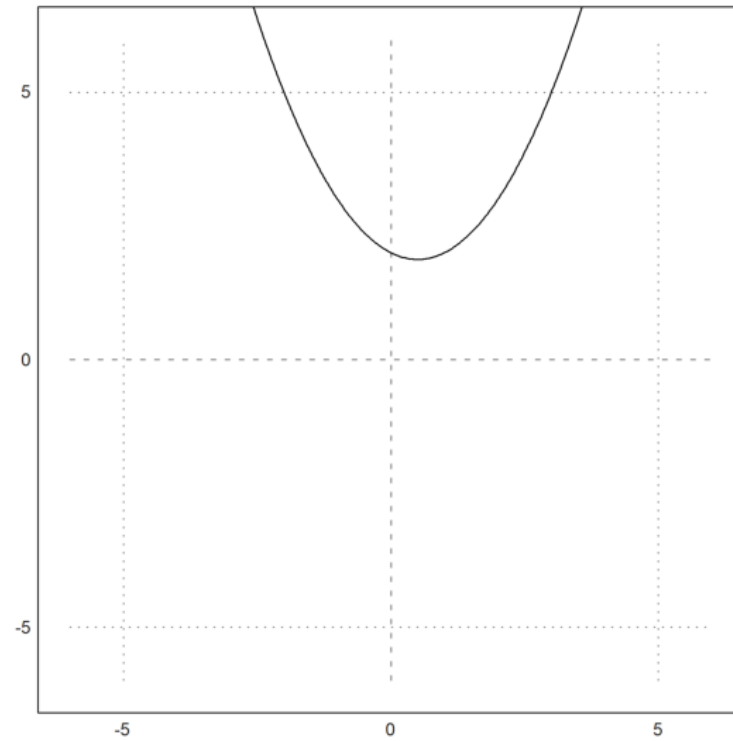
Error in:

```
... solve([a+b+c=2,4*a+2*b+c=3,9*a+3*b+c=5],[a,b,c]) ...
```

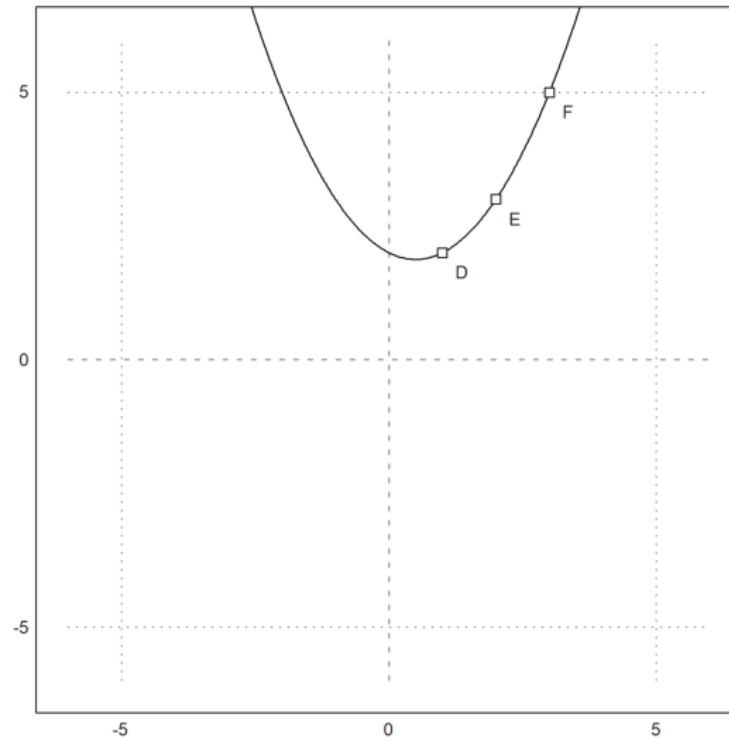
```
>function y&=(1/2)*x^2+(-1/2)*x+2
```

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

```
>plot2d("(1/2)*x^2+(-1/2)*x+2",-6,6,-6,6):
```



```
>plotPoint(D,"D"); plotPoint(E,"E"); plotPoint(F,"F");
```



3. Menggambar 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.

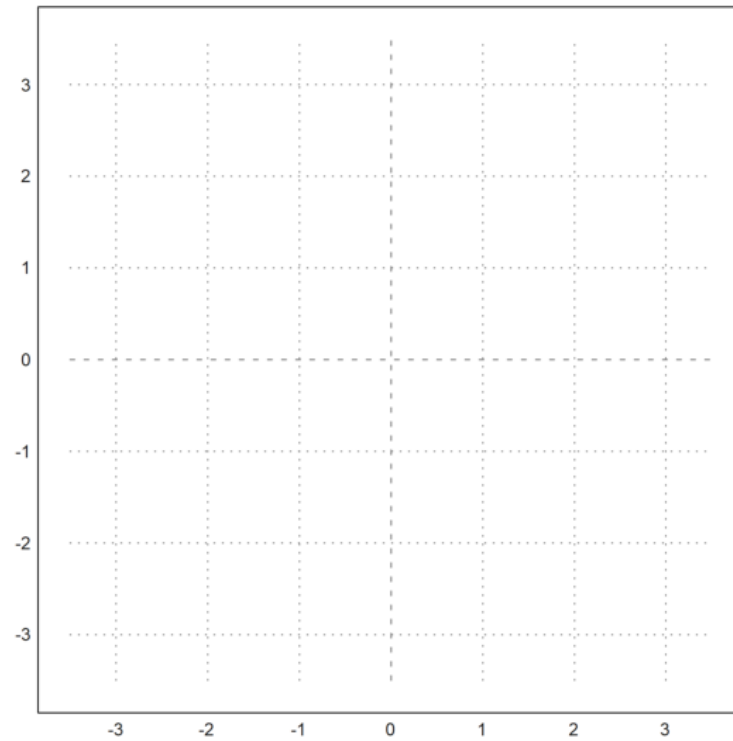
-Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung

apabila hasil kali panjang sisi-sisi yang berhadapan sama.

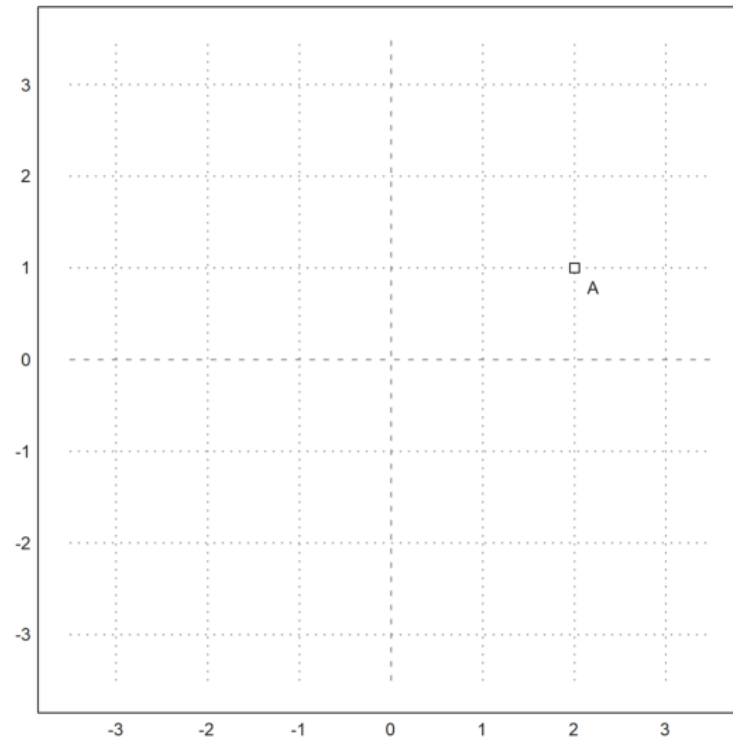
```
>load geometry
```

Numerical and symbolic geometry.

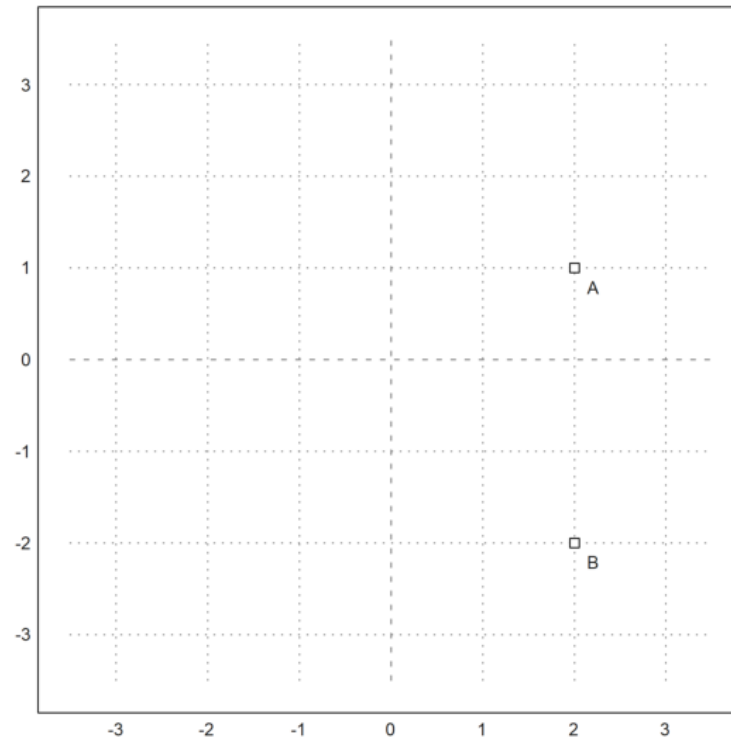
```
>setPlotRange(-3.5,3.5,-3.5,3.5):
```



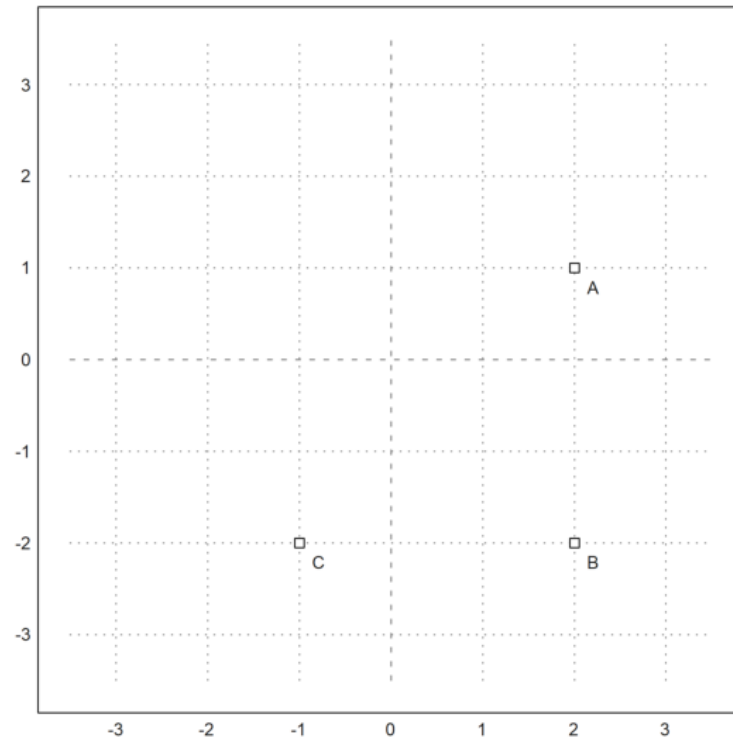
```
>A=[2,1]; plotPoint(A,"A"):
```



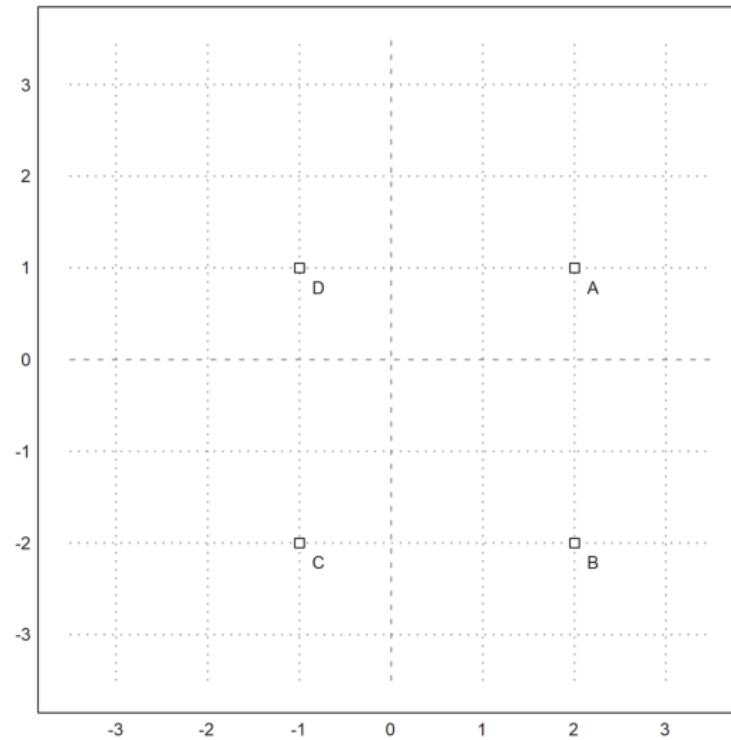
```
>B=[2,-2]; plotPoint(B,"B"):
```



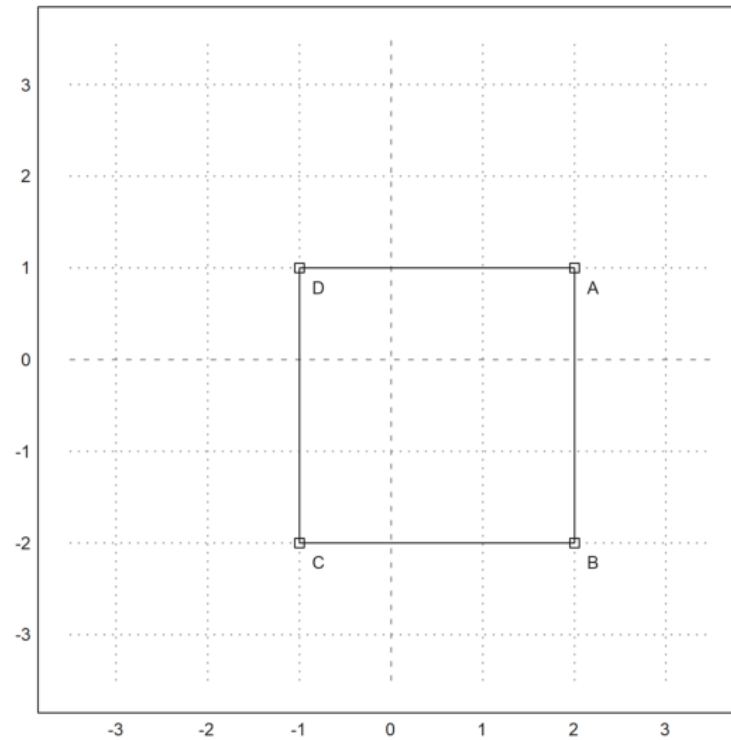
```
>C=[-1,-2]; plotPoint(C,"C"):
```



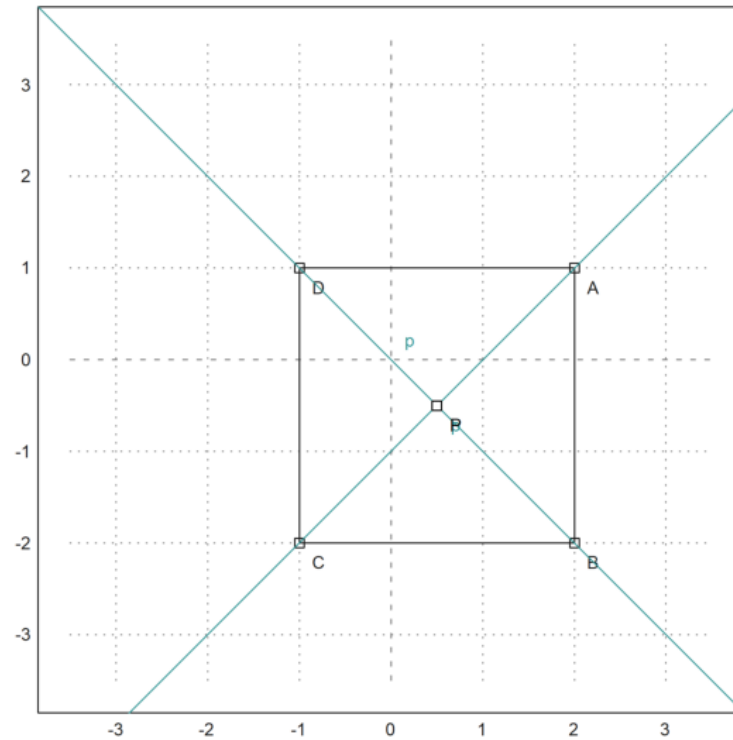
```
>D=[-1,1]; plotPoint(D,"D"):
```



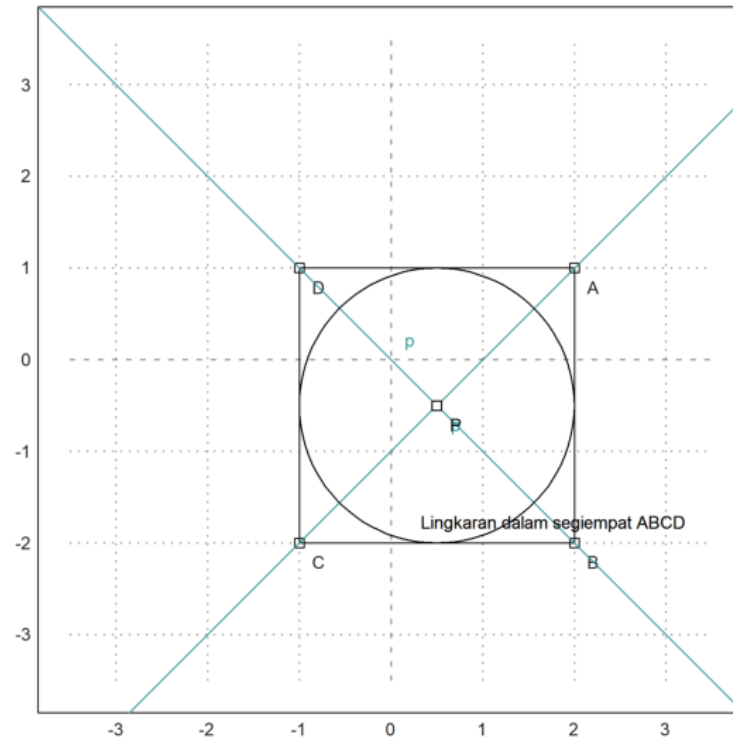
```
>plotSegment(A,B,""); plotSegment(B,C,""); plotSegment(C,D,""); plotSegment(A,D,"");  
>aspect(1):
```



```
>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");
```



```
>AB=distance(A,B) //panjang sisi AB
```

```
>CD=distance(C,D) //panjang sisi CD
```

3

```
>AD=distance(A,D) //panjang sisi AD
```

3

```
>BC=distance(B,C) //panjang sisi BC
```

3

```
>AB.CD
```

9

```
>AD.BC
```

9

Terdapat,

$$AB \times CD = 9, AD \times BC = 9$$

Sehingga terbukti bahwa

$$AB \times CD, AD \times BC$$

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

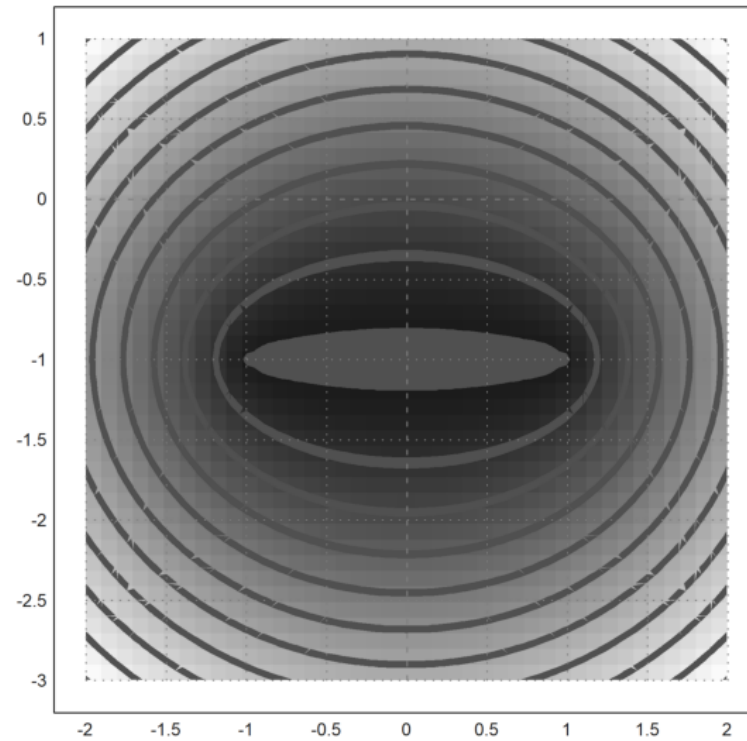
```
>reset();
```

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).

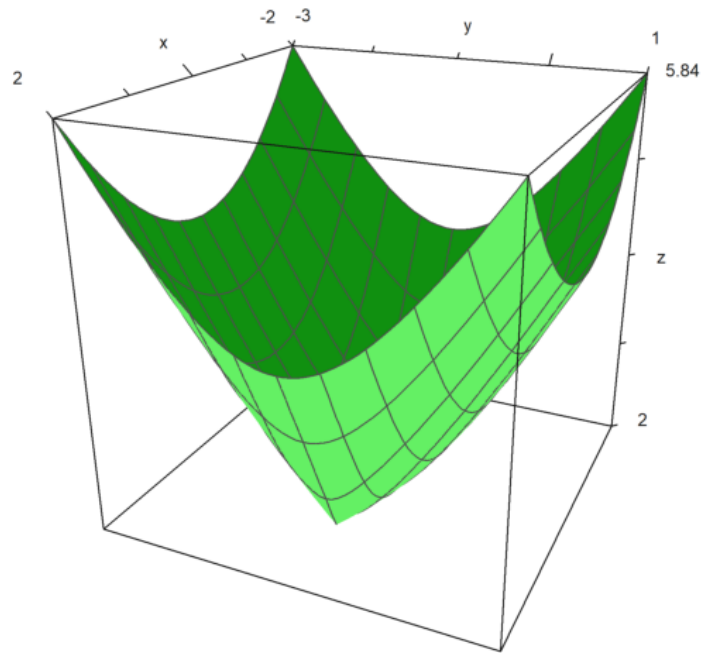
```
>load geometry
```

Numerical and symbolic geometry.

```
>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):
```

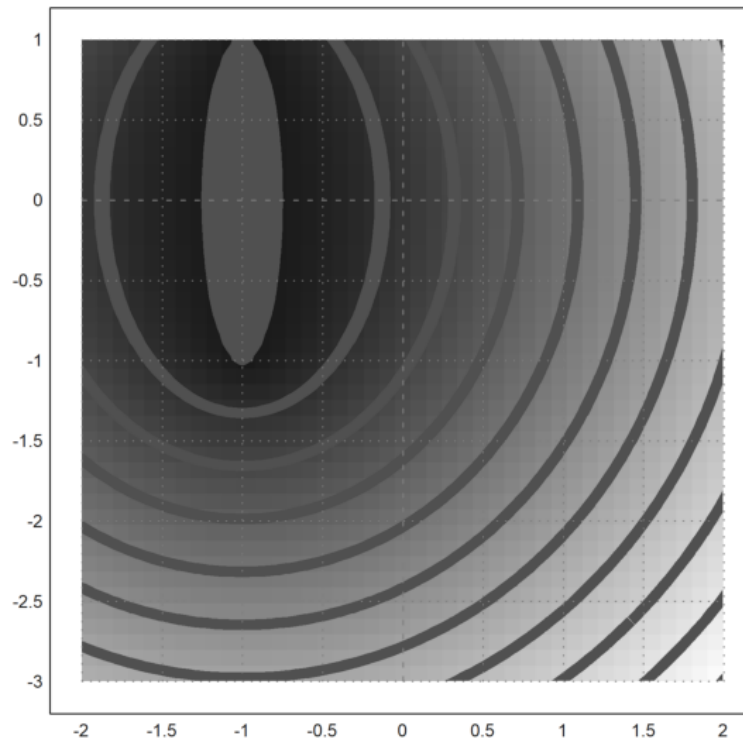


```
>plot3d("d2",xmin=-2,xmax=2,ymin=-3,ymax=1):
```

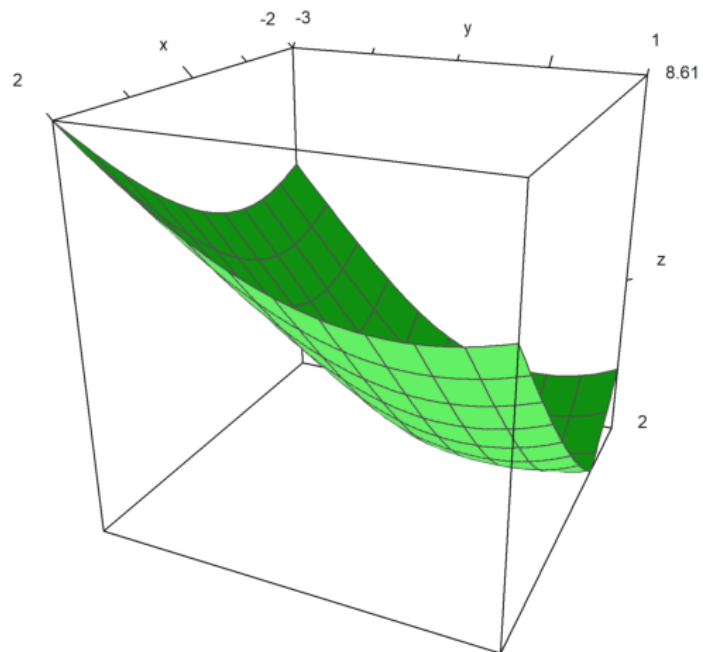


5. Gambarlah suatu hiperbola jika diketahui kedua titik fokusnya, misalnya P dan Q. Ingat ellips dengan fokus P dan Q adalah tempat kedudukan titik-titik yang selisih jarak ke P dan ke Q selalu sama (konstan).

```
>P=[-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):
```



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



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

