Visualisasi dan Perhitungan Geometri dengan EMT

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

```
>load geometry
```

```
Numerical and symbolic geometry.
```

Fungsi-fungsi Geometri

Fungsi-fungsi untuk Menggambar Objek Geometri:

```
defaultd:=textheight()*1.5: nilai asli untuk parameter d
setPlotrange(x1,x2,y1,y2): menentukan rentang x dan y pada bidang koordinat
setPlotRange(r): pusat bidang koordinat (0,0) dan batas-batas sumbu-x dan y adalah -r sd r
plotPoint (P, "P"): menggambar titik P dan diberi label "P"
plotSegment (A,B, "AB", d): menggambar ruas garis AB, diberi label "AB" sejauh d
plotLine (g, "g", d): menggambar garis g diberi label "g" sejauh d
plotCircle (c,"c",v,d): Menggambar lingkaran c dan diberi label "c"
plotLabel (label, P, V, d): menuliskan label pada posisi P
```

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

```
turn(v, phi): memutar vektor v sejauh phi
turnLeft(v):   memutar vektor v ke kiri
turnRight(v):  memutar vektor v ke kanan
normalize(v): normal vektor v
crossProduct(v, w): hasil kali silang vektorv dan w.
lineThrough(A, B): garis melalui A dan B, hasilnya [a,b,c] sdh. ax+by=c.
lineWithDirection(A,v): garis melalui A searah vektor v
getLineDirection(g): vektor arah (gradien) garis g
getNormal(g): vektor normal (tegak lurus) garis g
getPointOnLine(g):  titik pada garis g
perpendicular(A, g):  garis melalui A tegak lurus garis g
parallel (A, g):  garis melalui A sejajar garis g
lineIntersection(g, h):  titik potong garis g dan h
projectToLine(A, g):   proyeksi titik A pada garis g
distance(A, B):  jarak titik A dan B
distanceSquared(A, B):  kuadrat jarak A dan B
quadrance(A, B): kuadrat jarak A dan B
areaTriangle(A, B, C):  luas segitiga ABC
computeAngle(A, B, C):   besar sudut <ABC
angleBisector(A, B, C): garis bagi sudut <ABC
circleWithCenter (A, r): lingkaran dengan pusat A dan jari-jari r
getCircleCenter(c):  pusat lingkaran c
getCircleRadius(c):  jari-jari lingkaran c
circleThrough(A,B,C):  lingkaran melalui A, B, C
middlePerpendicular(A, B): titik tengah AB
lineCircleIntersections(g, c): titik potong garis g dan lingkran c
circleCircleIntersections (c1, c2):  titik potong lingkaran c1 dan c2
planeThrough(A, B, C):  bidang melalui titik A, B, C
```

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

```
getLineEquation (g,x,y): persamaan garis g dinyatakan dalam x dan y
getHesseForm (g,x,y,A): bentuk Hesse garis g dinyatakan dalam x dan y dengan titik A pada
sisi positif (kanan/atas) garis
quad(A,B): kuadrat jarak AB
spread(a,b,c): Spread segitiga dengan panjang sisi-sisi a,b,c, yakni sin(alpha)^2 dengan
alpha sudut yang menghadap sisi a.
crosslaw(a,b,c,sa): persamaan 3 quads dan 1 spread pada segitiga dengan panjang sisi a, b, c.
triplespread(sa,sb,sc): persamaan 3 spread sa,sb,sc yang memebntuk suatu segitiga
doublespread(sa): Spread sudut rangkap Spread 2*phi, dengan sa=sin(phi)^2 spread a.
```

Contoh 1: Luas, Lingkaran Luar, Lingkaran Dalam Segitiga

Untuk menggambar objek-objek geometri, langkah pertama adalah menentukan rentang sumbu-sumbu koordinat. Semua objek geometri akan digambar pada satu bidang koordinat, sampai didefinisikan bidang koordinat yang baru.

```
>setPlotRange(-0.5,2.5,-0.5,2.5); // mendefinisikan bidang koordinat baru
```

Sekarang tetapkan tiga titik dan plotlah.

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

Lalu tiga segmen.

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

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

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

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

Hitunglah garis tegak lurus melalui A pada BC.

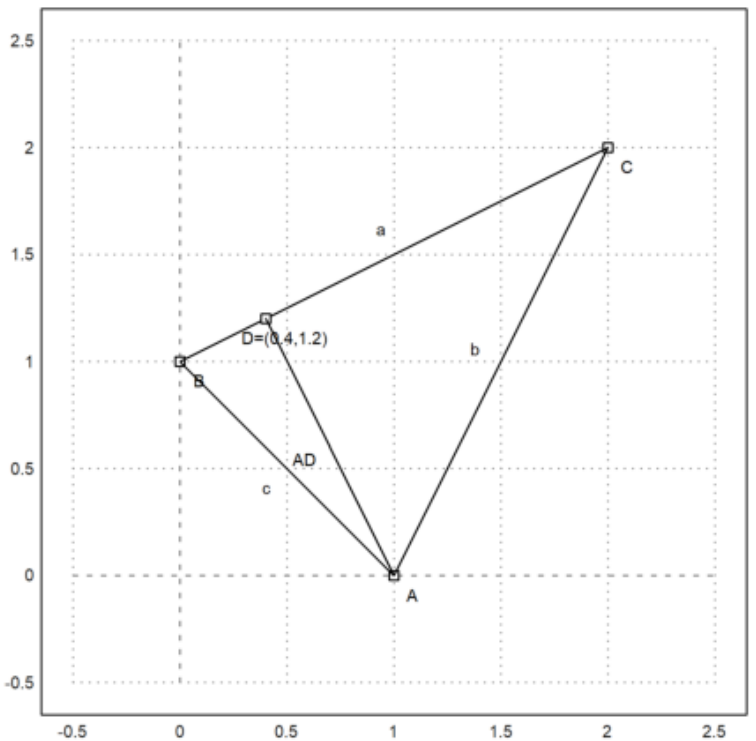
```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
```

Dan persimpangannya dengan BC.

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

Rencanakan itu.

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



Hitung luas ABC:

$$L_{\triangle ABC} = \frac{1}{2} AD.BC.$$

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

```
  1.5
```

Bandingkan dengan rumus determinan.

```
>areaTriangle(A,B,C) // hitung luas segitiga langusng dengan fungsi
```

```
  1.5
```

Cara lain menghitung luas segitigas ABC:

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

```
  1.5
```

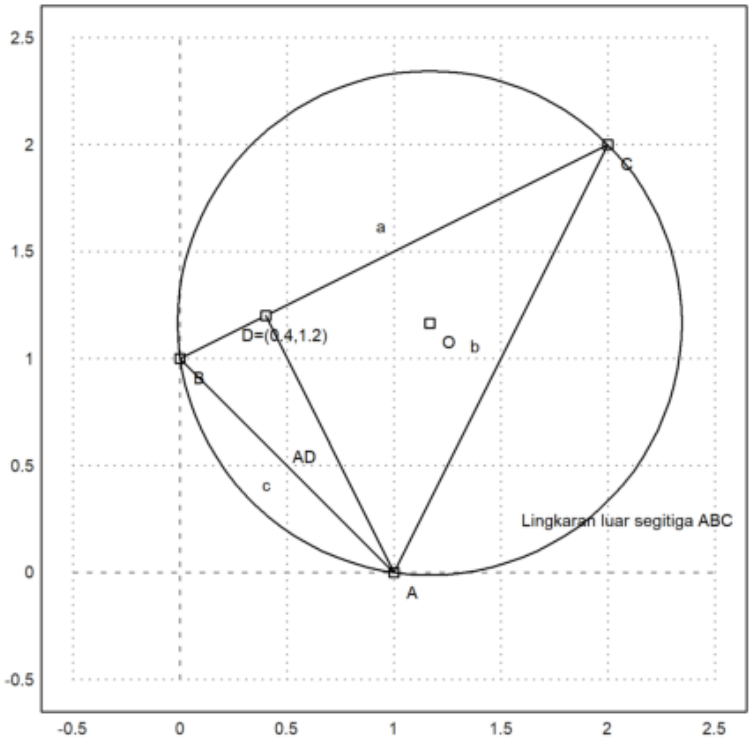
Sudut di C.

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

```
  36°52’11.63’’
```

sekarang lingkaran luar segitiga.

```
>c=circleThrough(A,B,C); // lingkaran luar segitiga ABC
>R=getCircleRadius(c); // jari2 lingkaran luar
>O=getCircleCenter(c); // titik pusat lingkaran c
>plotPoint(O,"O"); // gambar titik "O"
>plotCircle(c,"Lingkaran luar segitiga ABC"):
```



Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>O, R
```

```
  [1.16667,  1.16667]
1.17851130198
```

Sekarang akan digambar lingkaran dalam segitiga ABC. Titik pusat lingkaran dalam adalah titik potong garis-garis bagi sudut.

```
>l=angleBisector(A,C,B); // garis bagi <ACB
>g=angleBisector(C,A,B); // garis bagi <CAB
>P=lineIntersection(l,g) // titik potong kedua garis bagi sudut
```

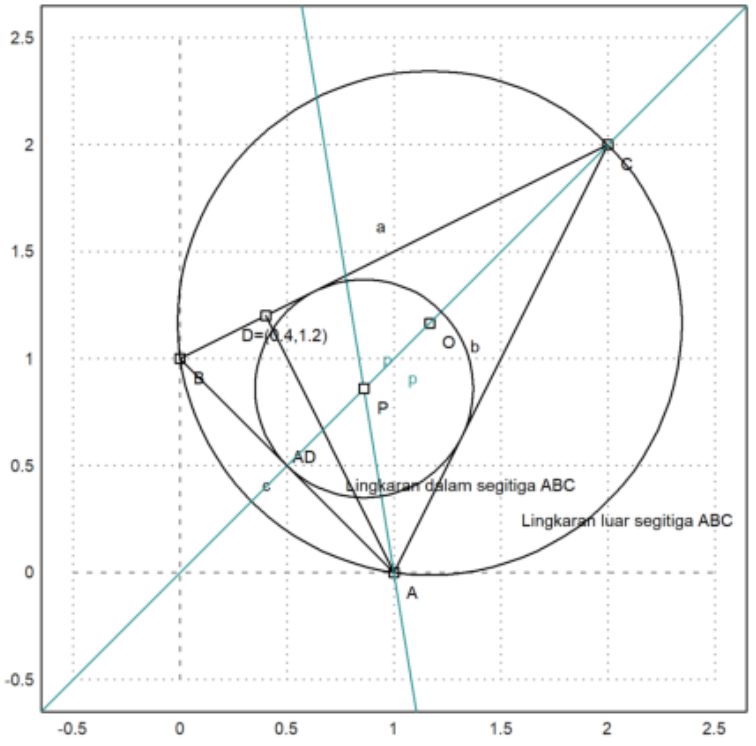
```
  [0.86038,  0.86038]
```

Tambahkan semuanya ke dalam alur cerita.

```
>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut
>plotPoint(P,"P"); // gambar titik potongnya
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam
```

```
  0.509653732104
```

```
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC"): // gambar lingkaran dalam
```


2.5
2
1.5
1
0.5
0
-0.5
-0.5
0
0.5
1
1.5
2
2.5
C
a
D=(0.4,1.2)
O
b
B
P
AD
c
Lingkaran dalam segitiga ABC
Lingkaran luar segitiga ABC
A

Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.
2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. Merupakan segitiga apakah itu?
3. Hitung luas segitiga tersebut.
4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.
5. Gambar jari-jari lingkaran dalam.
6. Hitung luas lingkaran luar dan luas lingkaran dalam segitiga ABC. Adakah hubungan antara luas kedua lingkaran tersebut dengan luas segitiga ABC?

Contoh 2: Geometri Smbolik

Kita dapat menghitung geometri eksak dan simbolik menggunakan Maxima.

Berkas geometry.e menyediakan fungsi yang sama (dan lebih banyak lagi) di Maxima. Namun, sekarang kita dapat menggunakan komputasi simbolik.

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

Fungsi untuk garis dan lingkaran bekerja seperti fungsi Euler, tetapi menyediakan perhitungan simbolis.

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

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

Kita dapat memperoleh persamaan garis dengan mudah.

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

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

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

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

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

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

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

$$\left(x_1-1\right)\ y-x\ y_1=-y_1$$

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

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

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

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

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

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

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

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

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

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

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

$$1.178511301977579$$

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

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

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

$$y = x$$

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

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

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

```
  [0.86038,  0.86038]
```

Garis dan Lingkaran yang Berpotongan

Tentu saja, kita juga bisa berpotongan antara garis dengan lingkaran, dan lingkaran dengan lingkaran.

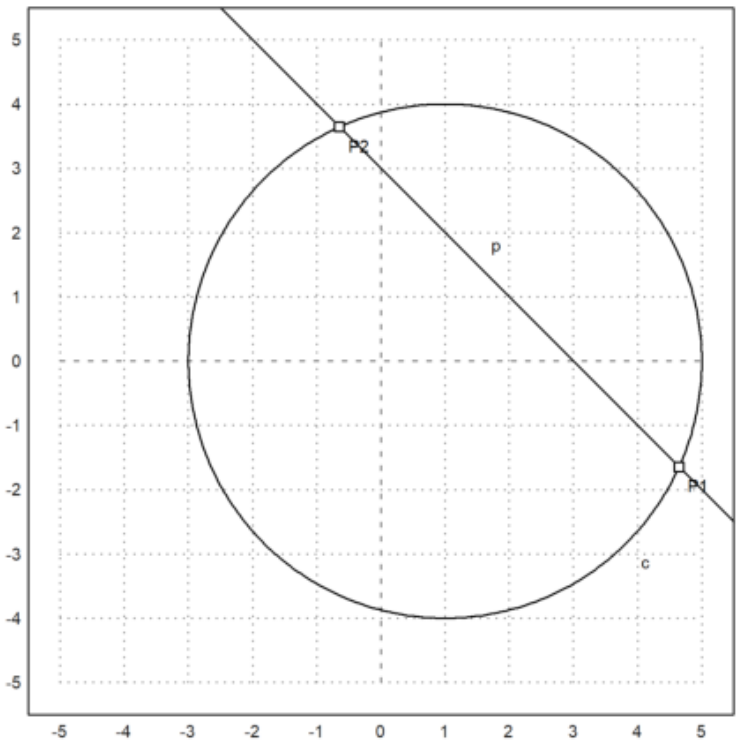
```
>A &:= [1,0]; c=circleWithCenter(A,4);
>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C);
>setPlotRange(5); plotCircle(c); plotLine(l);
```

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

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

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

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



Sama halnya di Maxima.

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

[1, 0, 4]

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

[1, 1, 3]

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

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

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

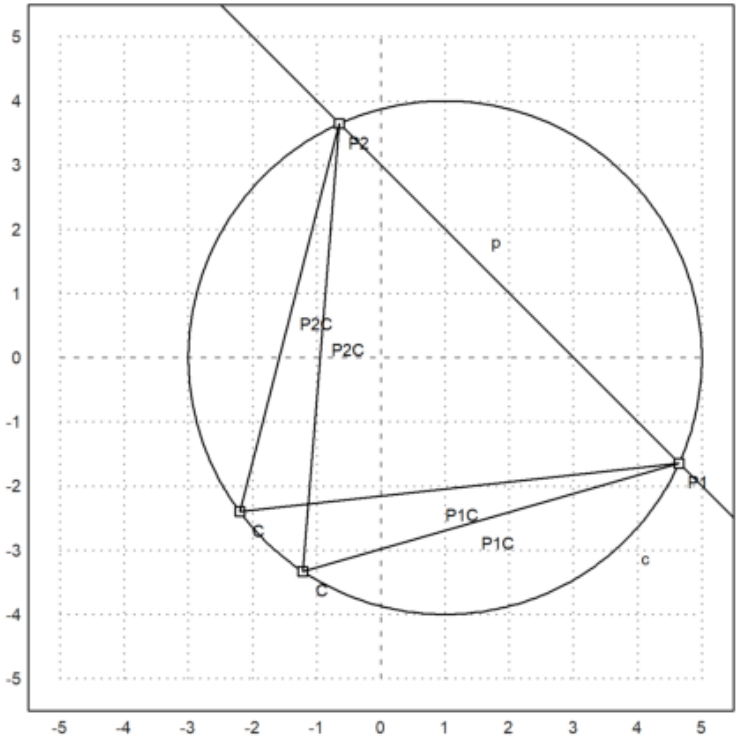
```
>C=A+normalize([-2,-3])*4; plotPoint(C); plotSegment(P1,C); plotSegment(P2,C);
>degprint(computeAngle(P1,C,P2))
```

```
  69°17’42.68’’
```

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

```
  69°17’42.68’’
```

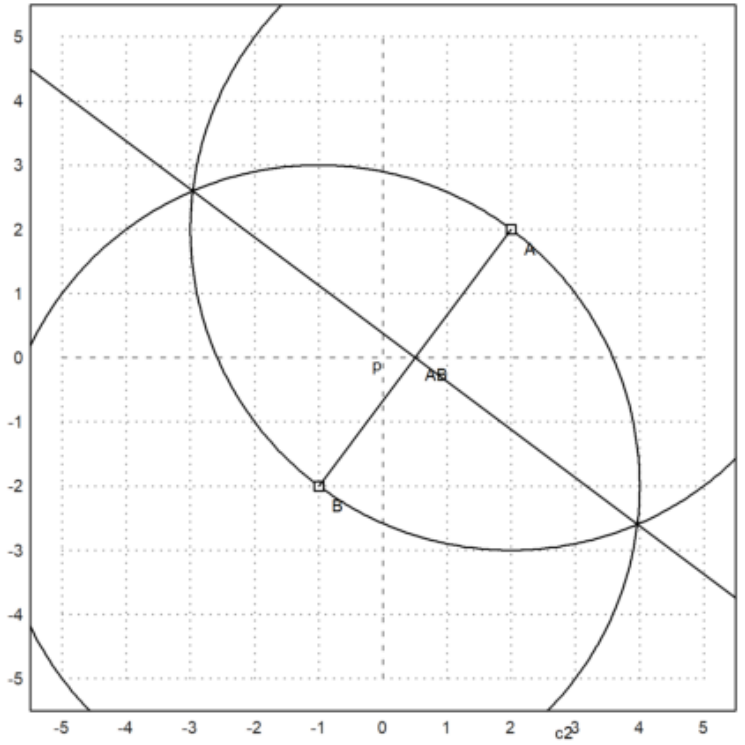
```
>insimg;
```


P2
p
P2C
P2C
P1
P1C
P1C
C
C
c

Berikut adalah langkah-langkah menggambar garis sumbu ruas garis AB:

1. Gambar lingkaran dengan pusat A melalui B.
2. Gambar lingkaran dengan pusat B melalui A.
3. Tarik garis melallui kedua titik potong kedua lingkaran tersebut. Garis ini merupakan garis sumbu (melalui titik tengah dan tegak lurus) AB.

```
>A=[2,2]; B=[-1,-2];
>c1=circleWithCenter(A,distance(A,B));
>c2=circleWithCenter(B,distance(A,B));
>{P1,P2,f}=circleCircleIntersections(c1,c2);
>l=lineThrough(P1,P2);
>setPlotRange(5); plotCircle(c1); plotCircle(c2);
>plotPoint(A); plotPoint(B); plotSegment(A,B); plotLine(l):
```



Selanjutnya, kita melakukan hal yang sama di Maxima dengan koordinat umum.

```
>A &= [a1,a2]; B &= [b1,b2];
>c1 &= circleWithCenter(A,distance(A,B));
>c2 &= circleWithCenter(B,distance(A,B));
>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2];
```

Persamaan untuk perpotongan tersebut cukup rumit. Namun, kita dapat menyederhanakannya dengan mencari nilai y.

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

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

Ini memang sama dengan tegak lurus tengah, yang dihitung dengan cara yang sepenuhnya berbeda.

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

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

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

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

Perhatikan hasil kali gradien garis g dan h adalah:

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

Artinya kedua garis tegak lurus.

Contoh 3: Rumus Heron

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

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

atau bisa ditulis dalam bentuk lain:

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

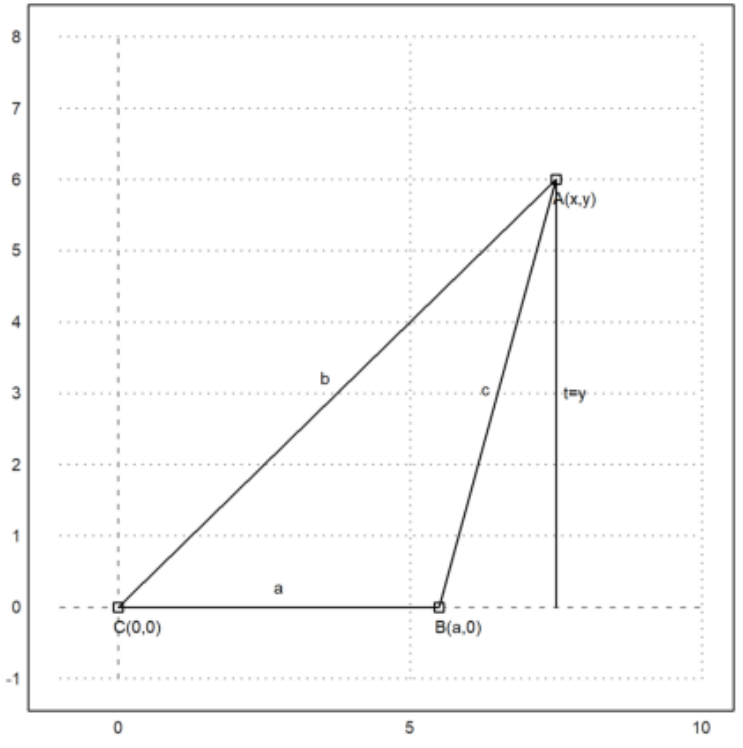
Untuk membuktikan hal ini kita misalkan C(0,0), B(a,0) dan A(x,y), b=AC, c=AB. Luas segitiga ABC adalah

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

Nilai y didapat dengan menyelesaikan sistem persamaan:

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

```
>setPlotRange(-1,10,-1,8); plotPoint([0,0], "C(0,0)"); plotPoint([5.5,0], "B(a,0)");  ...
> plotPoint([7.5,6], "A(x,y)");
>plotSegment([0,0],[5.5,0], "a",25); plotSegment([5.5,0],[7.5,6],"c",15);  ...
>plotSegment([0,0],[7.5,6],"b",25);
>plotSegment([7.5,6],[7.5,0],"t=y",25):
```




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

[]

Ekstrak solusi y.

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

```
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)) ...
                          ^
```

Kita mendapatkan rumus Heron.

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

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

```
>$'Luas=H(2,5,6) // luas segitiga dengan panjang sisi-sisi 2, 5, 6
```

$$Luas=|ysol|$$

Tentu saja, setiap segitiga siku-siku adalah kasus yang terkenal.

```
>H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5
```

```
Variable or function ysol not found.
Try "trace errors" to inspect local variables after errors.
H:
    useglobal; return a*abs(ysol)/2
Error in:
H(3,4,5) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...
        ^
```

Dan jelaslah pula, bahwa ini adalah segitiga dengan luas maksimal dan dua sisinya 3 dan 4.

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

```
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());
```

Kasus umum juga berfungsi.

```
>$solve(diff(H(a,b,c)^2,c)=0,c)
```

```
Maxima said:
diff: second argument must be a variable; found [1,0,4]
 -- an error. To debug this try: debugmode(true);

Error in:
 $solve(diff(H(a,b,c)^2,c)=0,c) ...
                             ^
```

Sekarang mari kita cari himpunan semua titik di mana b+c=d untuk suatu konstanta d. Diketahui bahwa himpunan ini adalah elips.

```
>s1 &= subst(d-c,b,sol[2]); $s1
```

```
Maxima said:
part: invalid index of list or matrix.
 -- an error. To debug this try: debugmode(true);

Error in:
s1 &= subst(d-c,b,sol[2]); $s1 ...
                       ^
```

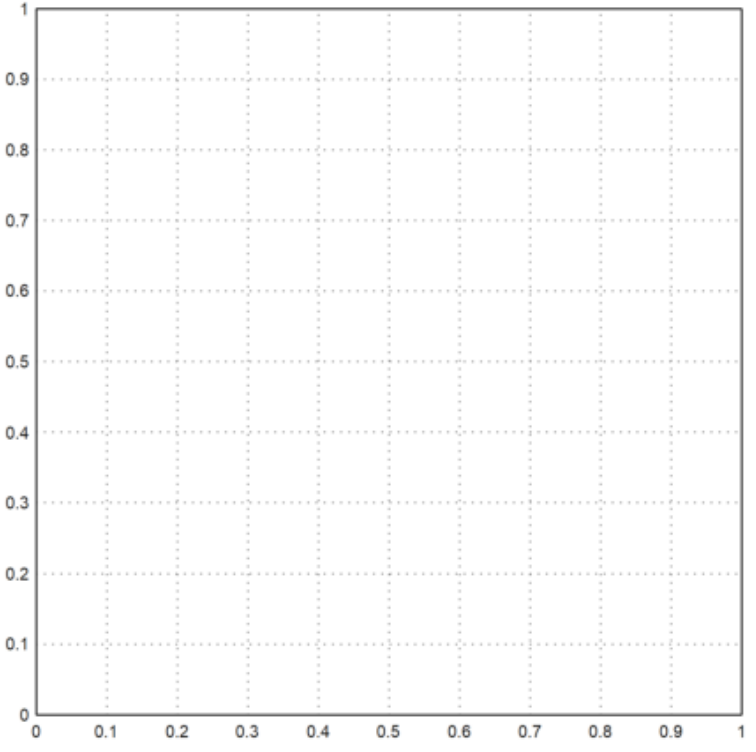
Dan buatlah fungsi dari ini.

```
>function fx(a,c,d) &= rhs(s1[1]); $fx(a,c,d), function fy(a,c,d) &= rhs(s1[2]); $fy(a,c,d)
```

0

Sekarang kita bisa menggambar himpunannya. Sisi b bervariasi dari 1 hingga 4. Diketahui bahwa kita mendapatkan elips.

```
>aspect(1); plot2d(&fx(3,x,5),&fy(3,x,5),xmin=1,xmax=4,square=1):
```



Kita dapat memeriksa persamaan umum untuk elips ini, yaitu:

$$\frac{(x - x_m)^2}{u^2} + \frac{(y - y_m)}{v^2} = 1,$$

di mana (xm, ym) adalah pusatnya, dan u dan v adalah sumbu setengahnya.

```
>$ratsimp((fx(a,c,d)-a/2)^2/u^2+fy(a,c,d)^2/v^2 with [u=d/2,v=sqrt(d^2-a^2)/2])
```

$$\frac{a^2}{d^2}$$

Kita melihat bahwa tinggi dan luas segitiga tersebut maksimum untuk x=0. Dengan demikian, luas segitiga dengan a+b+c=d maksimum jika segitiga tersebut sama sisi. Kita ingin memperoleh nilai ini secara analitis.

```
>eqns &= [diff(H(a,b,d-(a+b))^2,a)=0,diff(H(a,b,d-(a+b))^2,b)=0]; $eqns
```

$$\left[\frac{a\ ysol^2}{2}=0, 0=0\right]$$

Kita memperoleh beberapa nilai minimum, yang dimiliki oleh segitiga dengan satu sisi 0, dan penyelesaiannya a=b=c=d/3.

```
>$solve(eqns,[a,b])
```

$$[[a=0, b=\%r_1]]$$

Ada juga metode Lagrange, yang memaksimalkan H(a,b,c)^2 terhadap a+b+d=d.

```
>&solve([diff(H(a,b,c)^2,a)=la,diff(H(a,b,c)^2,b)=la, ...
>   diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la])
```

```
Maxima said:
diff: second argument must be a variable; found [1,0,4]
 -- an error. To debug this try: debugmode(true);

Error in:
... la,    diff(H(a,b,c)^2,c)=la,a+b+c=d],[a,b,c,la]) ...
                                                  ^
```

Kita bisa membuat plot dari situasinya

Pertama-tama atur titik-titik di Maxima.

```
>A &= at([x,y],sol[2]); $A
```

```
Maxima said:
part: invalid index of list or matrix.
 -- an error. To debug this try: debugmode(true);

Error in:
A &= at([x,y],sol[2]); $A ...
                      ^
```

```
>B &= [0,0]; $B, C &= [a,0]; $C
```

$$[a, 0]$$

Lalu atur rentang plot dan plot titik-titiknya.

```
>setPlotRange(0,5,-2,3); ...
>a=4; b=3; c=2; ...
>plotPoint(mxmeval("B"),"B"); plotPoint(mxmeval("C"),"C"); ...
>plotPoint(mxmeval("A"),"A"):
```

```
Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
    return evaluate(mxm(s));
Error in:
... otPoint(mxmeval("C"),"C"); plotPoint(mxmeval("A"),"A"): ...
                                                          ^
```

Plot segmennya.

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

```
Variable a1 not found!
Use global variables or parameters for string evaluation.
Error in Evaluate, superfluous characters found.
Try "trace errors" to inspect local variables after errors.
mxmeval:
    return evaluate(mxm(s));
Error in:
plotSegment(mxmeval("A"),mxmeval("C")); plotSegment(mxmeval("B ...
                        ^
```

Hitunglah garis tegak lurus tengah di Maxima.

```
>h &= middlePerpendicular(A,B); g &= middlePerpendicular(B,C);
```

Dan pusat kelilingnya.

```
>U &= lineIntersection(h,g);
```

Kita mendapatkan rumus untuk jari-jari lingkaran luar.

```
>&assume(a>0,b>0,c>0); $distance(U,B) | radcan
```

$$\frac{\sqrt{{a_2}^2+{a_1}^2}\,\sqrt{{a_2}^2+{a_1}^2-2\,a\,a_1+a^2}}{2\,\left|a_2\right|}$$

Mari kita tambahkan ini ke dalam alur cerita.

```
>plotPoint(U()); ...
>plotCircle(circleWithCenter(mxmeval("U"),mxmeval("distance(U,C)"))):
```

```
Variable a2 not found!
Use global variables or parameters for string evaluation.
Error in ^
Error in expression: [a/2,(a2^2+a1^2-a*a1)/(2*a2)]
Error in:
plotPoint(U()); plotCircle(circleWithCenter(mxmeval("U"),mxmev ...
                ^
```

Dengan menggunakan geometri, kita mendapatkan rumus sederhana

$$\frac{a}{\sin(\alpha)}=2r$$

untuk jari-jari. Kita dapat memeriksa apakah ini benar dengan Maxima. Maxima akan memfaktorkan ini hanya jika kita mengkuadratkannya.

```
>$c^2/sin(computeAngle(A,B,C))^2  | factor
```

$$\left[\frac{{a_2}^2+{a_1}^2}{{a_2}^2},0,\frac{16\left({a_2}^2+{a_1}^2\right)}{{a_2}^2}\right]$$

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari 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.

Sebagai demonstrasi, kami menghitung dan memplot garis Euler dalam sebuah segitiga.

Pertama, kami mendefinisikan sudut-sudut segitiga dalam Euler. Kami menggunakan definisi yang terlihat dalam ekspresi simbolik.

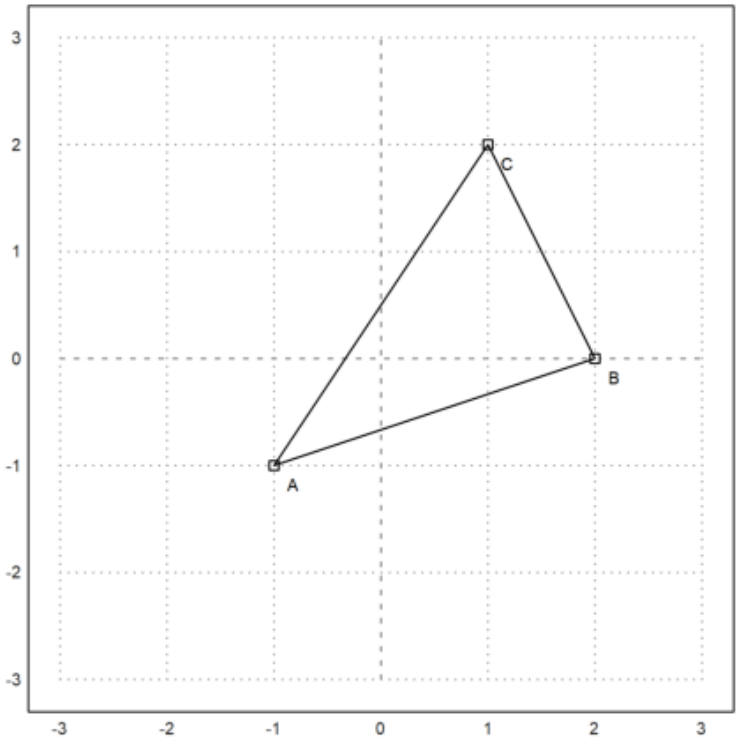
```
>A::=[-1,-1]; B::=[2,0]; C::=[1,2];
```

Untuk memplot objek geometris, kita siapkan area plot dan tambahkan titik-titik ke dalamnya. Semua plot objek geometris ditambahkan ke plot saat ini.

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

Kita juga dapat menambahkan sisi-sisi segitiga.

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



Berikut adalah luas segitiga, menggunakan rumus determinan. Tentu saja, kita harus mengambil nilai absolut dari hasil ini.

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

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

Kita dapat menghitung koefisien sisi c.

```
>c &= lineThrough(A,B)
```

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

Dan dapatkan juga rumus untuk baris ini.

```
>$getLineEquation(c,x,y)
```

$$3\,y - x = -2$$

Untuk bentuk Hesse, kita perlu menentukan sebuah titik, sehingga titik tersebut berada di sisi positif bentuk Hesse. Memasukkan titik tersebut akan menghasilkan jarak positif ke garis.

```
>$getHesseForm(c,x,y,C), $at(%,[x=C[1],y=C[2]])
```

$$\frac{7}{\sqrt{10}}$$

$$\frac{7}{\sqrt{10}}$$

Sekarang kita hitung lingkaran luar ABC.

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

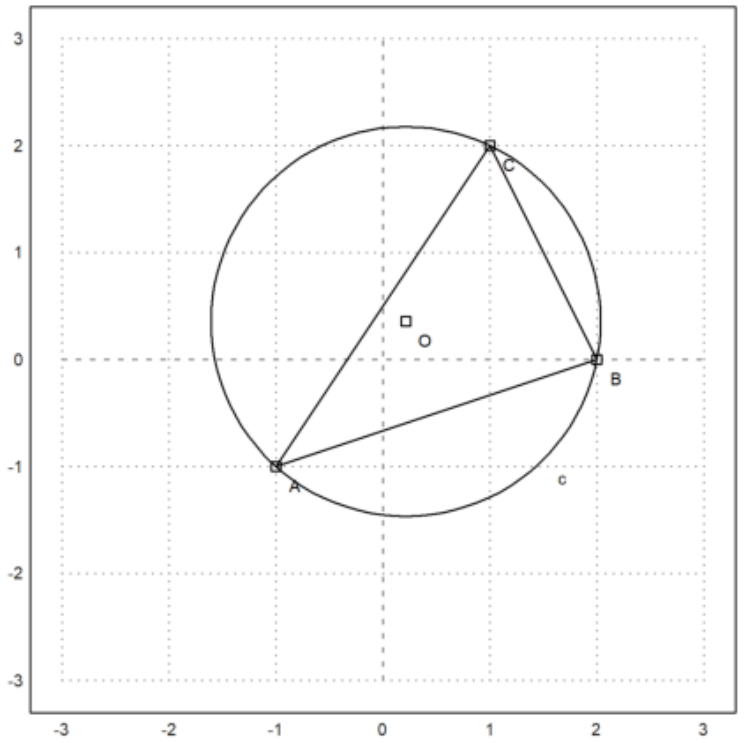
$$\left(y-\frac{5}{14}\right)^2+\left(x-\frac{3}{14}\right)^2=\frac{325}{98}$$

```
>O &= getCircleCenter(LL); $O
```

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

Gambarkan lingkaran dan titik pusatnya. Cu dan U adalah simbol. Kita evaluasi ekspresi ini untuk Euler.

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



Kita dapat menghitung perpotongan tinggi di ABC (orthocenter) secara numerik dengan perintah berikut.

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

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

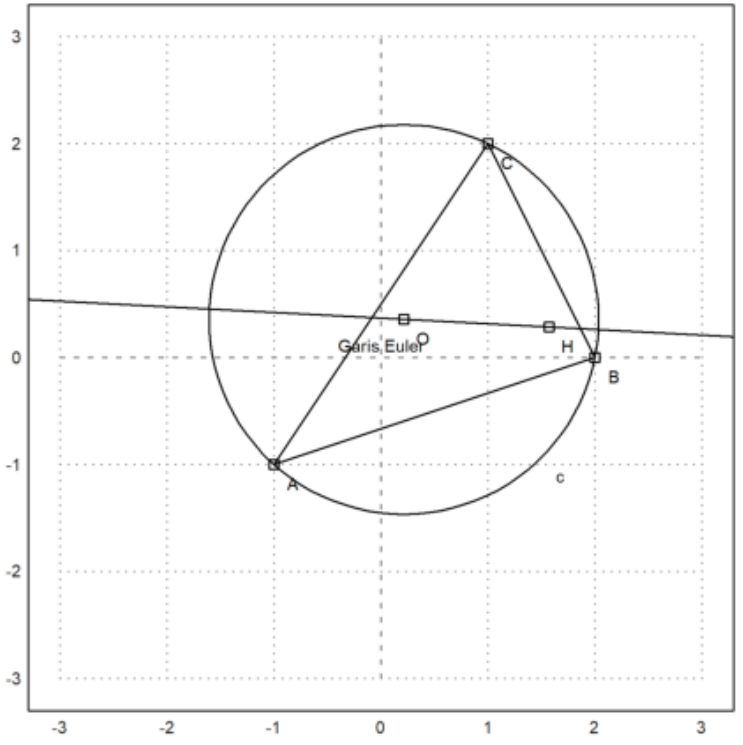
Sekarang kita dapat menghitung garis Euler dari segitiga tersebut.

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

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

Tambahkan ke plot kita.

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

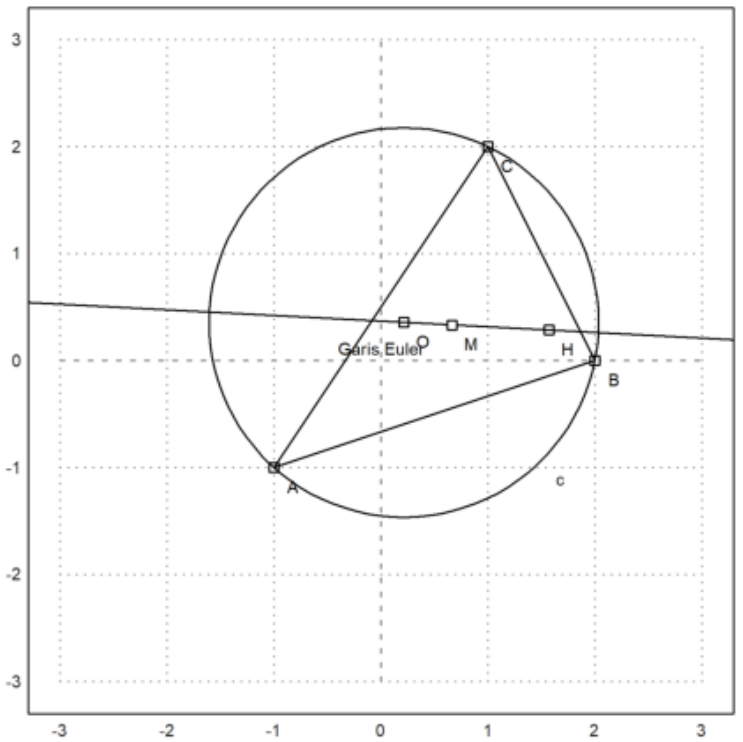


Pusat gravitasi seharusnya berada pada garis ini.

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

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

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



Teori ini menyatakan MH=2*MO. Kita perlu menyederhanakannya dengan radcan untuk mencapai hal ini.

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

$$2$$

Fungsinya termasuk fungsi untuk sudut juga.

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

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

```
60°15’18.43’’
```

Persamaan untuk pusat lingkaran dalam kurang bagus.

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

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

Mari kita hitung juga ekspresi untuk jari-jari lingkaran tertulis.

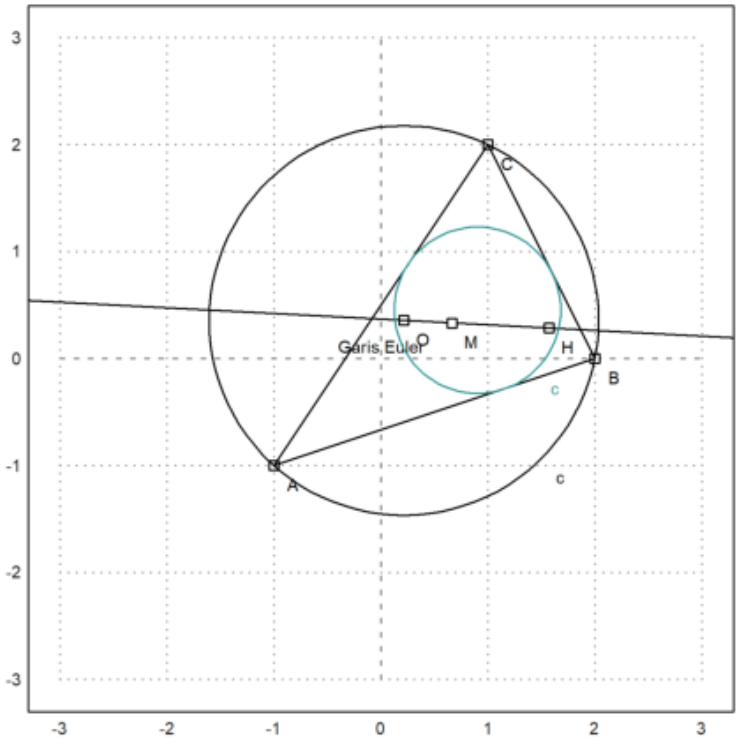
```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r
```

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

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

Mari kita tambahkan ini ke dalam alur cerita.

```
>color(5); plotCircle(LD()):
```



Parabola

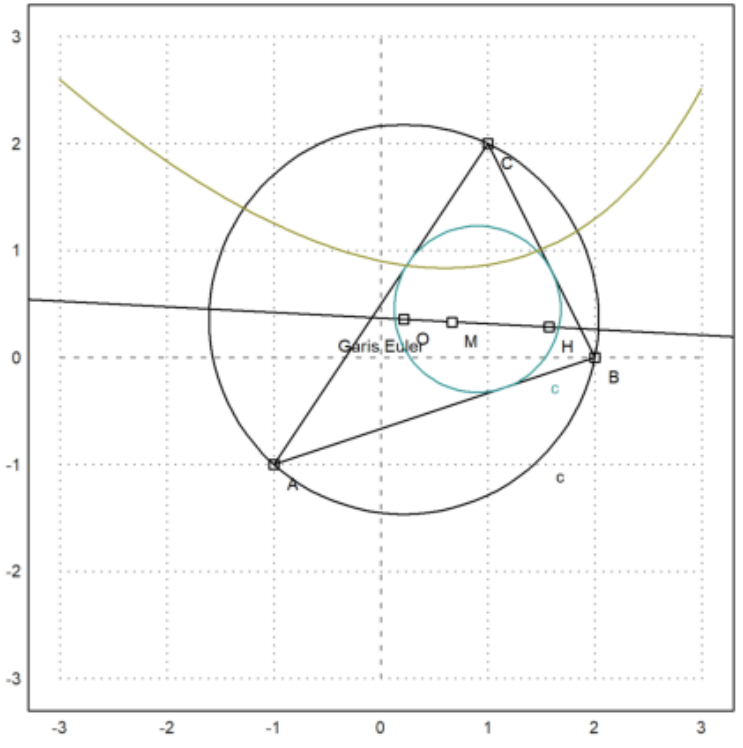
Selanjutnya akan dicari persamaan tempat kedudukan titik-titik yang berjarak sama ke titik C dan ke garis AB.

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

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

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

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



Seharusnya ini adalah suatu fungsi, tetapi pemecah standar Maxima hanya dapat menemukan solusinya jika kita mengkuadratkan persamaannya. Akibatnya, kita mendapatkan solusi palsu.

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

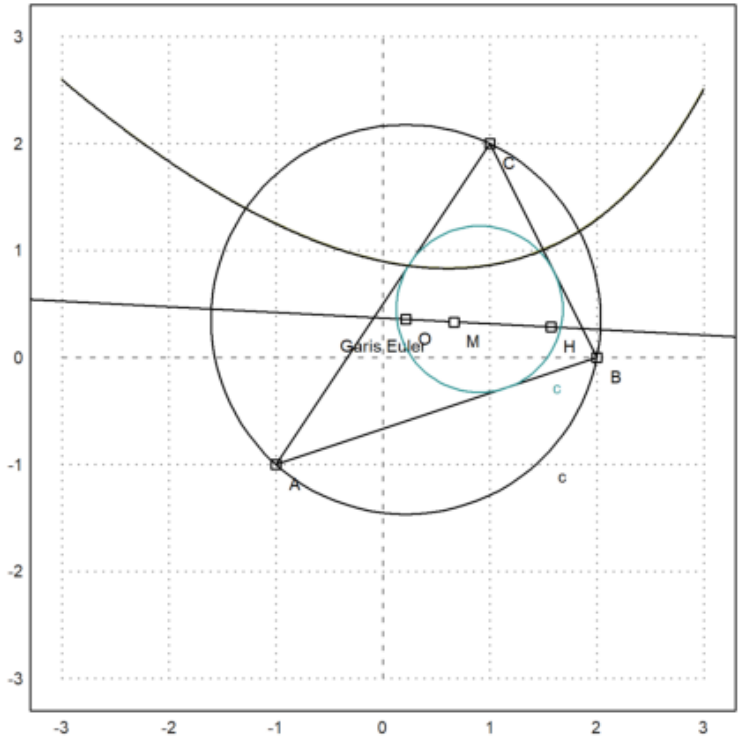
```
[y = - 3 x - sqrt(70) sqrt(9 - 2 x) + 26,
                    y = - 3 x + sqrt(70) sqrt(9 - 2 x) + 26]
```

Solusi pertama adalah

maxima: akar[1]

Menambahkan solusi pertama ke dalam plot menunjukkan bahwa memang jalur tersebut yang kita cari. Teorinya memberi tahu kita bahwa jalur tersebut adalah parabola berputar.

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




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

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

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

2.135605779339061

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

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

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

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 oleh ceramah N.J.Wildberger. Dalam bukunya "Divine Proportions", Wildberger mengusulkan untuk mengganti konsep klasik jarak dan sudut dengan kuadran dan sebaran. Dengan menggunakan konsep ini, memang dimungkinkan untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

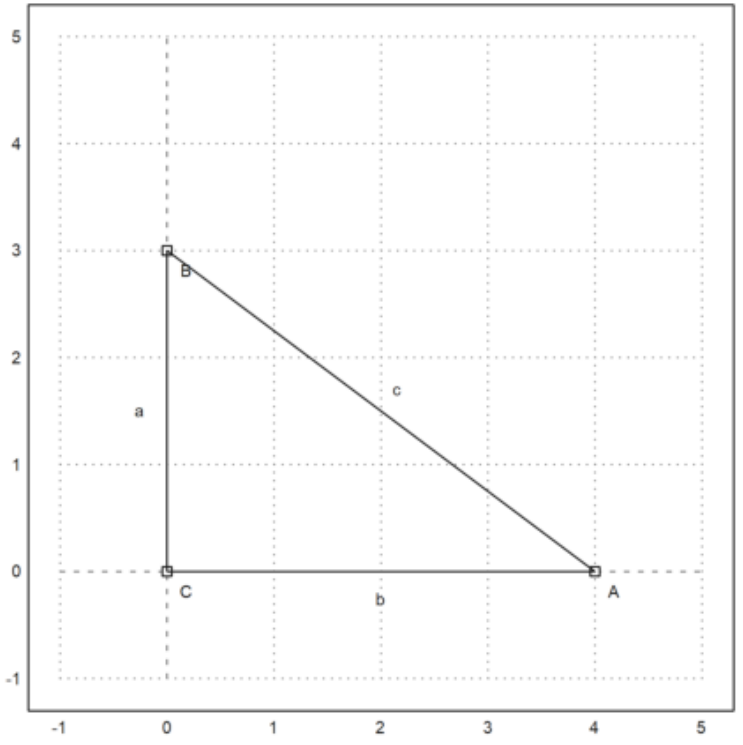
Berikutnya, saya memperkenalkan konsep-konsep tersebut, dan menyelesaikan beberapa soal. Saya menggunakan perhitungan simbolis Maxima di sini, yang menyembunyikan keuntungan utama trigonometri rasional, yaitu perhitungannya hanya dapat dilakukan dengan kertas dan pensil. Anda dipersilakan untuk memeriksa hasilnya tanpa komputer.

Intinya adalah bahwa perhitungan simbolis rasional seringkali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

```
>load geometry;
```

Sebagai pengantar pertama, kami menggunakan segitiga siku-siku dengan proporsi Mesir yang terkenal, yaitu 3, 4, dan 5. Perintah berikut adalah perintah Euler untuk memplot geometri bidang yang terdapat dalam berkas Euler "geometry.e".

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



Tentu saja,

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

di mana wa adalah sudut di A. Cara umum untuk menghitung sudut ini adalah dengan mengambil invers dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak secara perkiraan.

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

```
  36°52’11.63’’
```

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadran, yang menggantikan jarak. Sebenarnya, itu hanyalah kuadrat jarak. Selanjutnya, a, b, dan c menunjukkan kuadran sisi-sisinya.

Teorema Pythogoras menjadi a+b=c.

```
>a &= 3^2; b &= 4^2; c &= 5^2; &a+b=c
```

$$25 = 25$$

Konsep kedua trigonometri rasional adalah sebaran. Sebaran mengukur celah antar garis. Nilainya 0 jika garis-garisnya sejajar, dan 1 jika garis-garisnya persegi panjang. Nilai sebaran adalah kuadrat sinus sudut antara kedua garis tersebut.

Sebaran garis AB dan AC pada gambar di atas didefinisikan sebagai

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

di mana a dan c adalah kuadran dari setiap segitiga siku-siku dengan salah satu sudutnya berada di A.

```
>sa &= a/c; $sa
```

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

Ini tentu saja lebih mudah dihitung daripada sudutnya. Namun, Anda kehilangan sifat bahwa sudut dapat dijumlahkan dengan mudah.

Tentu saja, kita dapat mengonversi nilai perkiraan untuk sudut wa ke sprad, dan mencetaknya sebagai pecahan.

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

```
  9/25
```

Hukum kosinus dalam trgonometri klasik diterjemahkan menjadi "hukum silang" berikut.

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

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

Hukum-hukum ini diimplementasikan dalam berkas geometry.e yang kami muat 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 mencari sebaran di A. Untuk melakukannya, kita buat hukum silang untuk kuadran a, b, dan c, lalu selesaikan untuk sebaran sa yang belum diketahui.

Anda dapat melakukannya dengan mudah secara manual, tetapi saya menggunakan Maxima. Tentu saja, kita mendapatkan hasilnya, yang sudah kita miliki.

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

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

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

Kita sudah tahu ini. Definisi sebaran merupakan kasus khusus dari hukum silang.

Kita juga dapat menyelesaikan ini untuk persamaan umum a, b, c. Hasilnya adalah rumus yang menghitung sebaran sudut suatu segitiga jika diketahui kuadran ketiga sisinya.

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

$$\left[\left[\frac{168\,bb\,x+36\,bb^2+(-72\,aa-84)\;bb+36\,aa^2-84\,aa+49}{36},\frac{168\,bb\,x+36\,bb^2+(-72\,aa-84)\;bb+36\,aa^2-84\,aa+4}{36}\right.\right.$$

Kita bisa membuat fungsi dari hasil tersebut. Fungsi tersebut sudah didefinisikan dalam berkas geometry.e milik Euler.

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

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

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

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

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

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

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

Ini adalah sudut dalam derajat.

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

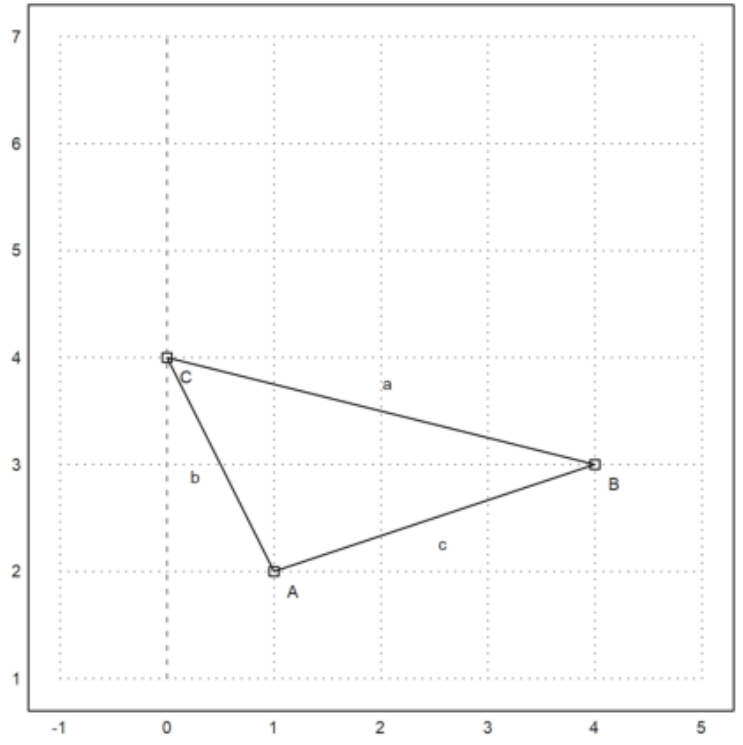
```
  67°47'32.44''
```

Contoh Lain

Sekarang, mari kita coba contoh yang lebih lanjut.

Kita tentukan tiga sudut segitiga sebagai berikut.

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



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

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

$$\sqrt{10}$$

Euler juga memuat fungsi untuk kuadran antara dua titik.

Dalam contoh berikut, karena c+b bukan a, segitiga tersebut bukan persegi panjang.

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


Pertama, mari kita hitung sudut tradisional. Fungsi computeAngle menggunakan metode umum berdasarkan perkalian titik dua vektor. Hasilnya adalah beberapa aproksimasi floating point.

$$A =< 1,2 > \quad B =< 4,3 >, \quad C =< 0,4 >$$

$$\mathbf{a} = C - B =< -4,1 >, \quad \mathbf{c} = A - B =< -3,-1 >, \quad \beta = ABC$$

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

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

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

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

```
32.4711922908
```

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

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

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

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

Yaitu, apa yang dilakukan fungsi sebaran yang didefinisikan dalam "geometry.e".

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

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

Maxima mendapatkan hasil yang sama menggunakan trigonometri biasa, jika kita memaksakannya. Maxima memang menyelesaikan suku sin(arccos(...)) menjadi hasil pecahan. Kebanyakan siswa tidak dapat melakukan ini.

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

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

Setelah kita memiliki sebaran di B, kita dapat menghitung tinggi ha pada sisi a. Ingat bahwa

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

menurut definisi.

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

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

Gambar berikut dihasilkan dengan program geometri C.a.R., yang dapat menggambar kuadran dan sebaran.

gambar: (20) Rational_Geometry_CaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadrannya.

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

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

Sekarang kita bisa menghitung luas segitiga. Jangan lupa, kita sedang membahas tentang kuadran!

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

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

Rumus determinan yang biasa menghasilkan hasil yang sama.

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

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

Rumus Heron

Sekarang, mari kita selesaikan masalah ini secara umum!

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

Pertama-tama, kita hitung sebaran di titik B untuk segitiga dengan sisi a, b, dan c. Kemudian, kita hitung luas kuadrat ("quadrea"?), faktorkan dengan Maxima, dan kita dapatkan rumus Heron yang terkenal.

Memang, ini sulit dilakukan dengan pensil dan kertas.

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

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

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

Aturan Sebaran Tiga Kali Lipat

Kerugian sebaran adalah tidak lagi hanya menambahkan sudut-sudut yang sama. Namun, tiga sebaran segitiga memenuhi aturan "sebaran tiga kali lipat" berikut.

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

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

Aturan ini berlaku untuk tiga sudut yang jika dijumlahkan menjadi 180°.

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

Karena sebaran

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

sama, aturan sebaran rangkap tiga juga berlaku, jika

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

Karena sebaran sudut negatif sama, aturan sebaran rangkap tiga juga berlaku, jika

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

Misalnya, kita dapat menghitung sebaran sudut 60°. Hasilnya adalah 3/4. Persamaan tersebut memiliki solusi kedua, di mana semua sebarannya adalah 0.

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

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

Sebaran sudut 90° jelas merupakan 1. Jika dua sudut berjumlah 90°, sebarannya menyelesaikan persamaan sebaran rangkap tiga dengan a, b, 1. Dengan perhitungan berikut, kita mendapatkan a+b=1.

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

$$[x=1-y]$$

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

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

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

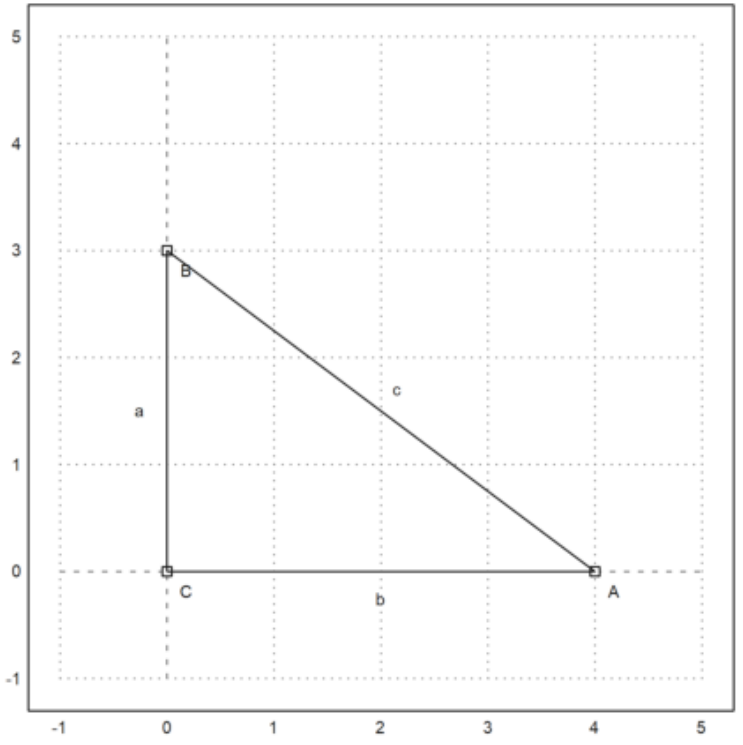
$$[x = 4\,a - 4\,a^2, x = 0]$$

```
- 4 (a - 1) a
```

Garis Bagi Sudut

Kita sudah tahu situasinya seperti ini.

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



Mari kita hitung panjang garis bagi sudut di A. Namun kita ingin menyelesaikannya untuk a,b,c umum.

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

Jadi, pertama-tama kita hitung sebaran sudut yang dibagi dua di A, menggunakan rumus sebaran rangkap tiga.

Masalah dengan rumus ini muncul lagi. Rumus ini memiliki dua solusi. Kita harus memilih solusi yang tepat. Solusi lainnya mengacu pada sudut yang dibagi dua 180°-wa.

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

$$\frac{-\sqrt{b}\,\sqrt{b+a}+b+a}{2\,b+2\,a}$$

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

$$\frac{-\sqrt{b}\,\sqrt{b+a}+b+a}{2\,b+2\,a}$$

Mari kita periksa persegi panjang Mesir.

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

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

Kita dapat mencetak sudut dalam Euler, setelah mentransfer sebarannya ke radian.

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

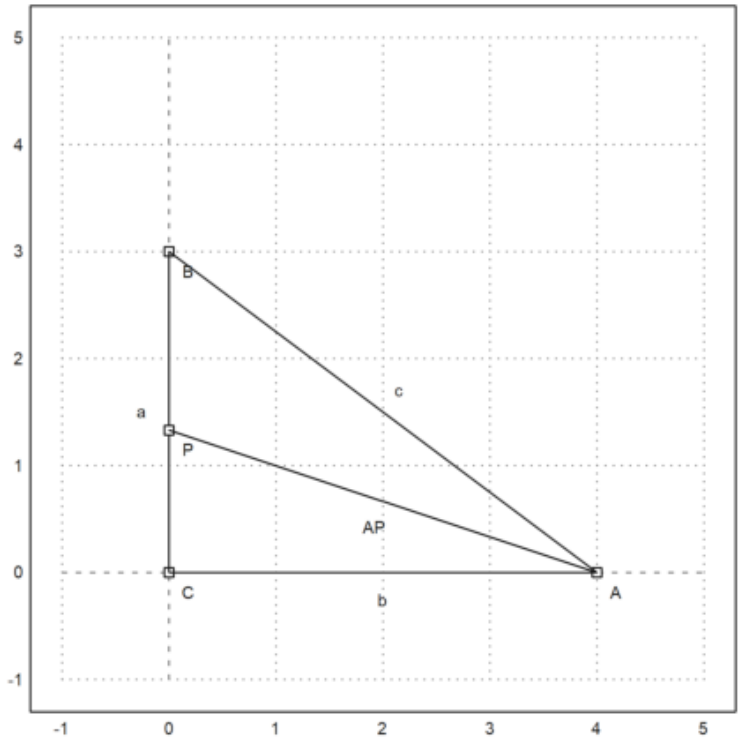
```
  18°26’5.82’’
```

Titik P merupakan perpotongan garis bagi sudut dengan sumbu y.

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

```
  [0,  1.33333]
```

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



Mari kita periksa sudut dalam contoh spesifik kita.

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

```
0.321750554397
0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.

Kita gunakan teorema sinus 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".

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

di mana a, b, c menunjukkan kuadran.

Karena CPA sebaran adalah 1-sa2, kita peroleh darinya bisa/1 = b / (1-sa2) dan dapat menghitung bisa (kuadran dari garis bagi sudut).

```
>&factor(ratsimp(b/(1-sa2))); bisa &= %; $bisa
```

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

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

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

```
  4.21637021356
  4.21637021356
```

Kita juga dapat menghitung P menggunakan rumus penyebaran.

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

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

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

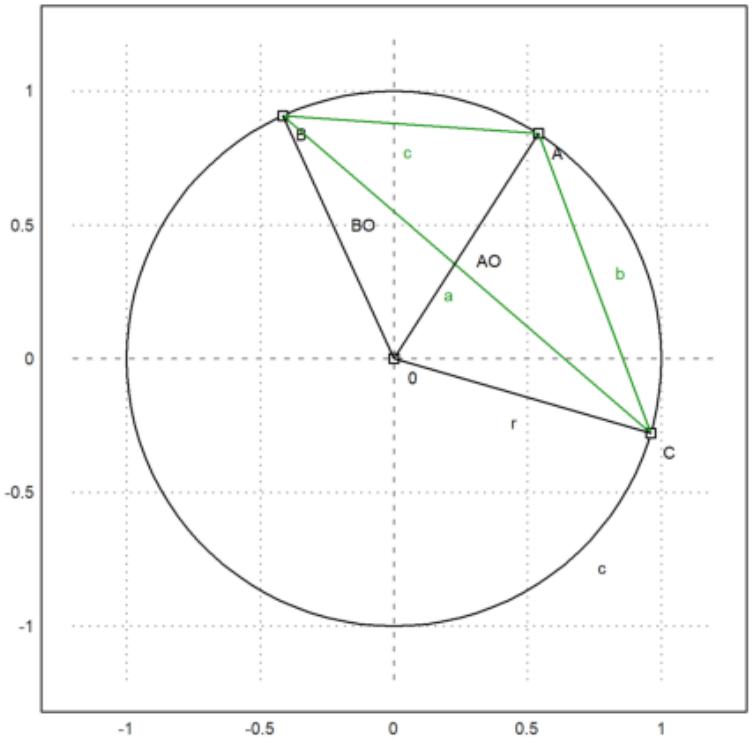
```
>sqrt(mxmeval("at(py,[a=3^2,b=4^2])"))
```

```
  1.33333333333
```

Sudut Kord

Perhatikan situasi berikut.

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



Kita dapat menggunakan Maxima untuk menyelesaikan rumus sebaran rangkap tiga untuk sudut-sudut di pusat O untuk r. Dengan demikian, kita mendapatkan rumus untuk jari-jari kuadrat perilingkaran dalam bentuk kuadran sisi-sisinya.

Kali ini, Maxima menghasilkan beberapa nol kompleks, yang kita abaikan.

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

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

Kita dapat menjadikannya fungsi Euler.

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

Mari kita periksa hasilnya untuk titik A, B, C.

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

Jari-jarinya memang 1.

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

Faktanya, sebaran CBA hanya bergantung pada b dan c. Ini adalah teorema sudut tali busur.

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

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

Faktanya, sebarannya adalah b/(4r), dan kita melihat bahwa sudut tali busur b adalah setengah sudut pusat.

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

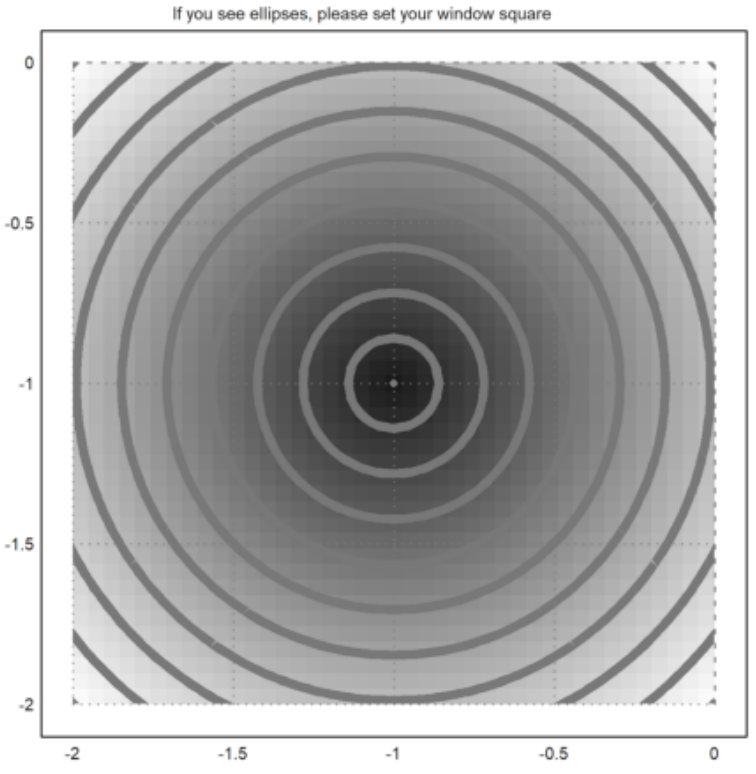
$$0$$

Contoh 6: Jarak Minimum pada Bidang

Catatan Awal

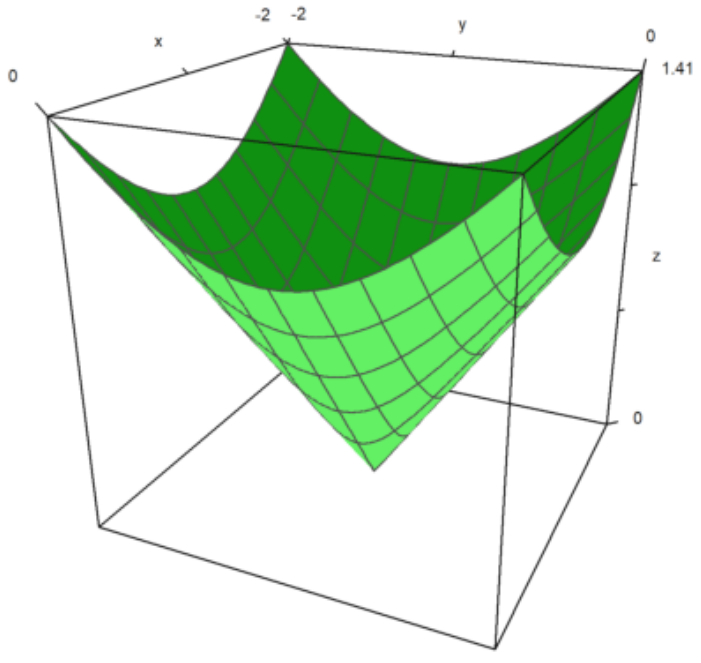
Fungsi yang, pada suatu titik M di bidang, menetapkan jarak AM antara suatu titik tetap A dan M, memiliki garis datar yang cukup sederhana: lingkaran yang berpusat di A.

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



dan grafiknya juga cukup sederhana: bagian atas kerucut:

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

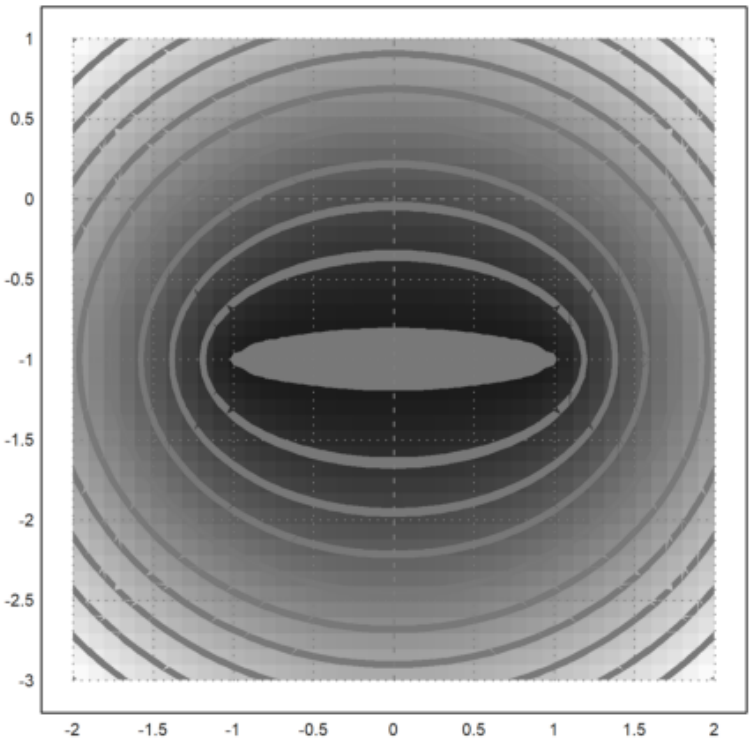


Tentu saja, titik minimum 0 tercapai di A.

Dua titik

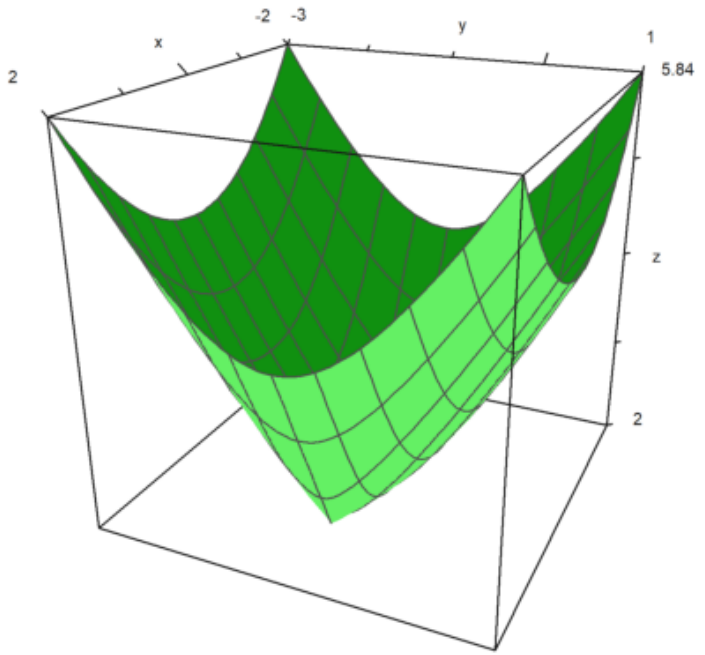
Sekarang kita lihat fungsi MA+MB di mana A dan B adalah dua titik (tetap). Sudah menjadi "fakta umum" bahwa kurva level berbentuk elips, dengan titik fokus A dan B; kecuali untuk titik minimum AB yang konstan pada segmen [AB]:

```
>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):
```



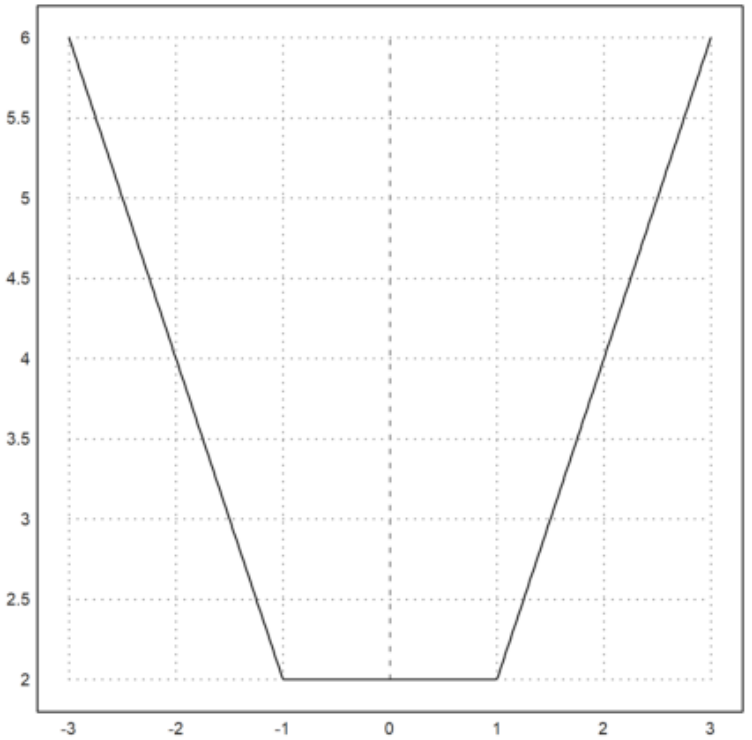
Grafiknya lebih menarik:

```
>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):
```

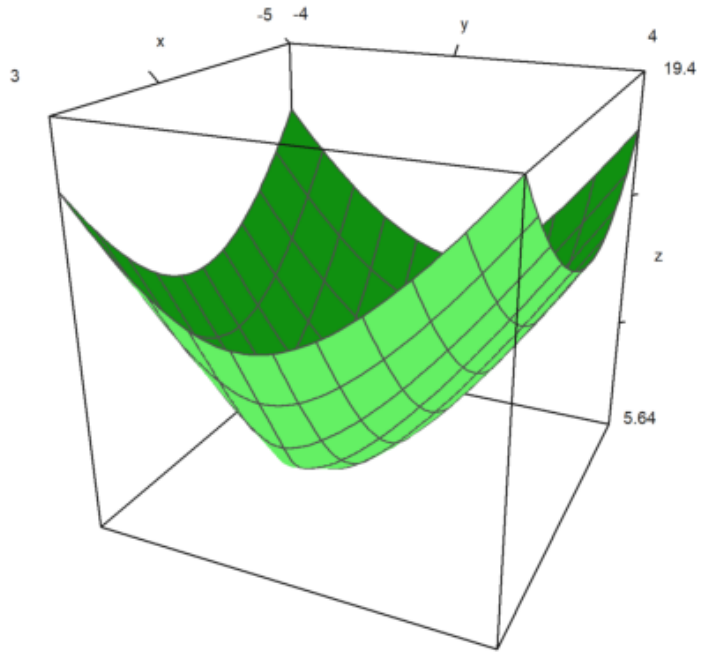


Sekarang, semuanya menjadi kurang sederhana: Memang kurang diketahui bahwa MA+MB+MC mencapai nilai minimumnya di salah satu titik bidang, tetapi menentukannya tidaklah sesederhana itu:

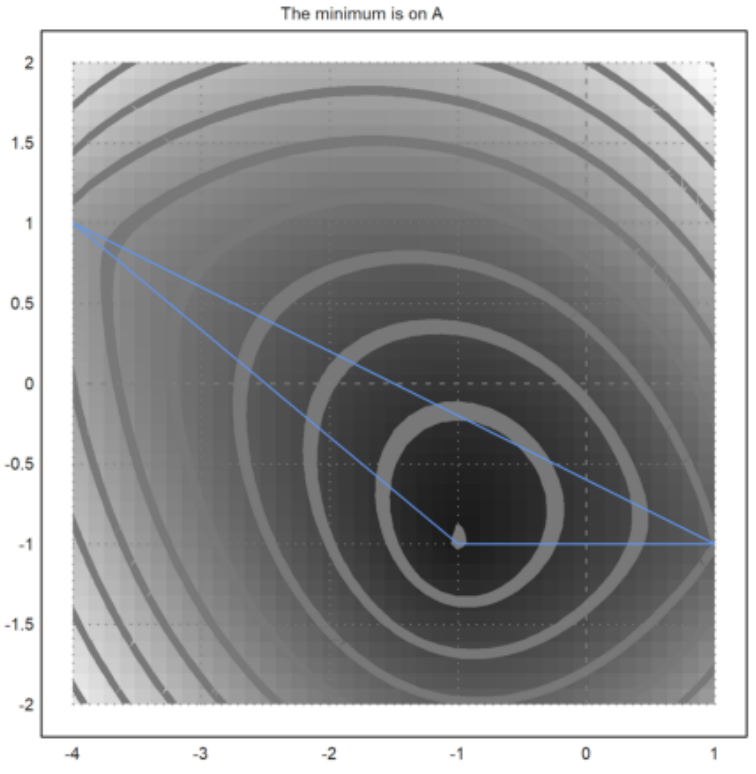
1) Jika salah satu sudut segitiga ABC lebih dari 120° (misalnya di A), maka nilai minimumnya tercapai di titik tersebut (misalnya AB+AC).

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;
```

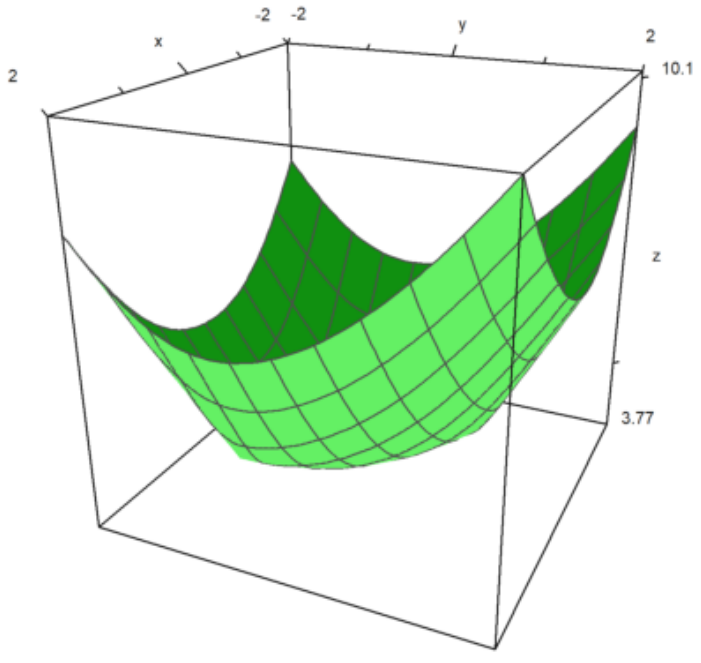



```
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

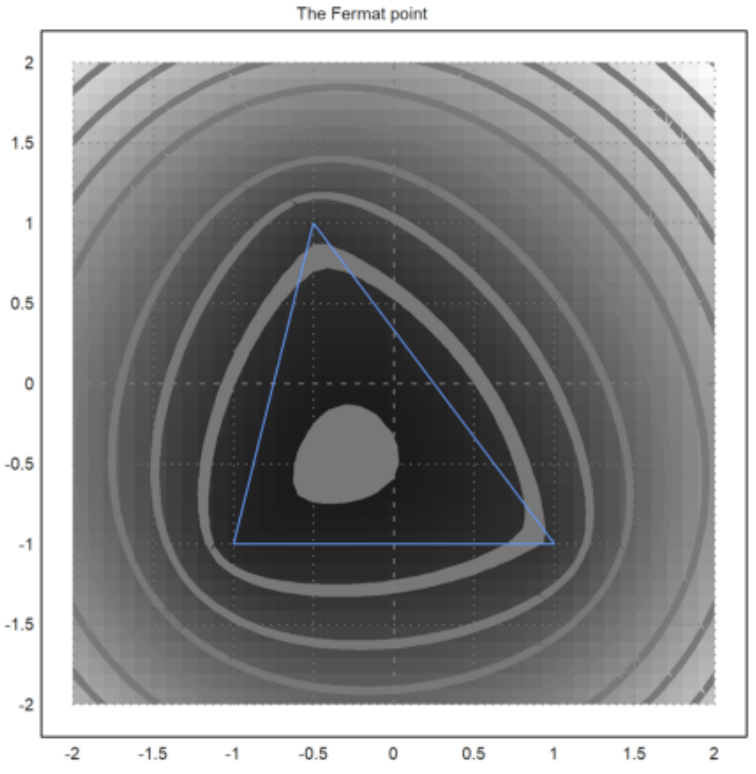


2) Namun jika semua sudut segitiga ABC kurang dari 120˚, maka sudut minimumnya ada di titik F di bagian dalam segitiga, yang merupakan satu-satunya titik yang melihat sisi-sisi ABC dengan sudut yang sama (masing-masing 120˚):

```
>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;
```



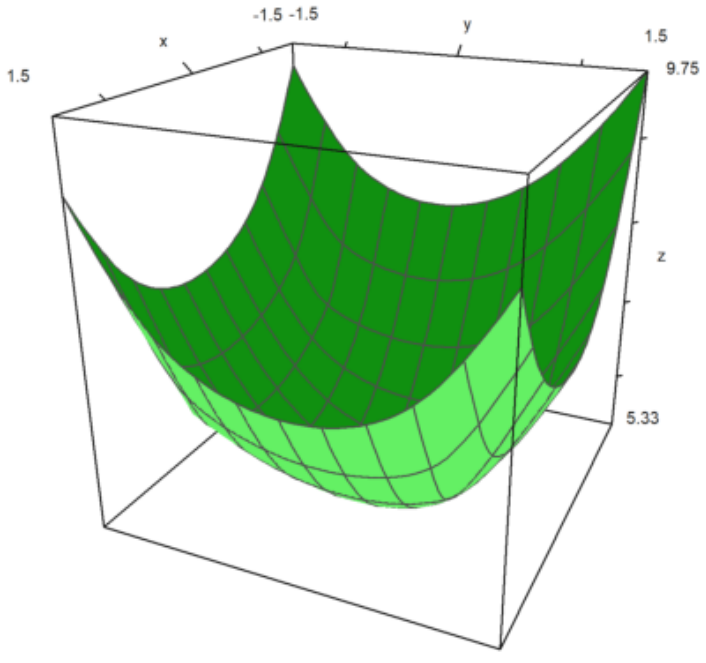
Merupakan aktivitas yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya tahu perangkat lunak yang ditulis dalam Java dengan instruksi ”garis kontur”...

Semua hal di atas telah ditemukan oleh seorang hakim Prancis bernama Pierre de Fermat; ia menulis surat kepada dilettan lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di pajak penghasilan. Jadi, titik unik F sedemikian rupa sehingga FA+FB+FC minimal, disebut titik Fermat dari segitiga tersebut. Namun, tampaknya beberapa tahun sebelumnya, Torriccelli dari Italia telah menemukan titik ini sebelum Fermat! Bagaimanapun, tradisinya adalah mencatat titik F ini...

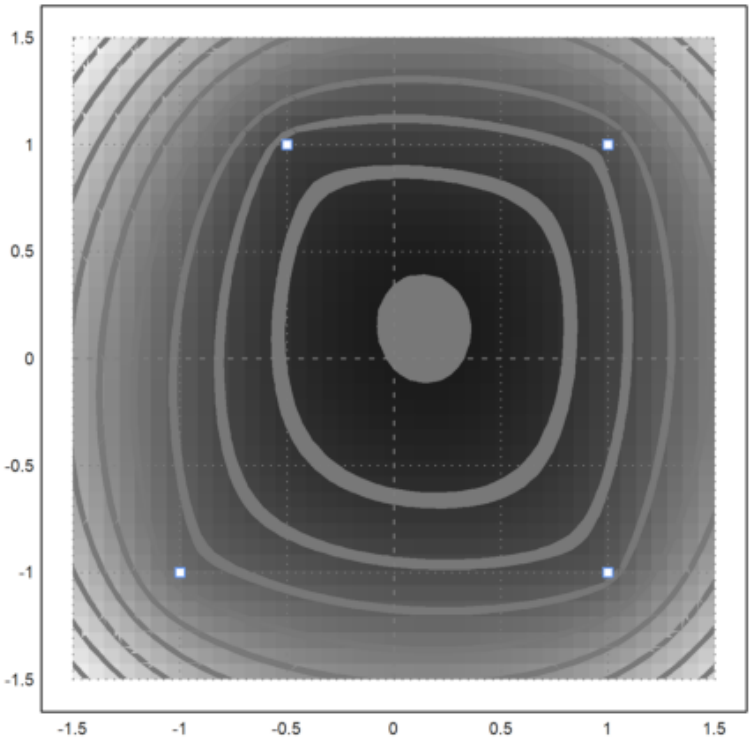
Empat titik

Langkah selanjutnya adalah menambahkan titik ke-4 D dan mencoba meminimalkan MA+MB+MC+MD; katakanlah Anda seorang operator TV kabel dan ingin mencari di bidang mana Anda harus menempatkan antena agar dapat mengudara empat desa dan menggunakan panjang kabel sesedikit mungkin!

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




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



Masih ada minimum dan tidak tercapai di titik A, B, C, maupun D:

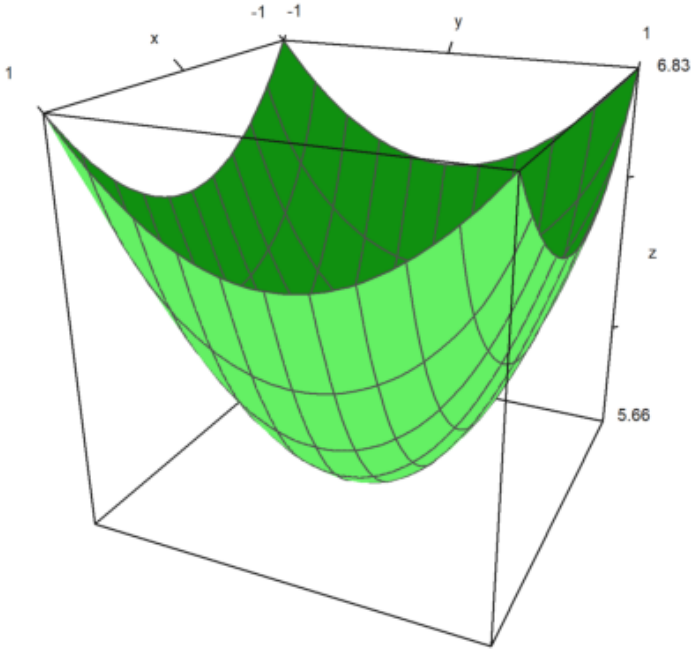
```
>function f(x):=d4(x[1],x[2])
>neldermin("f",[0.2,0.2])
```

```
  [0.142858,  0.142857]
```

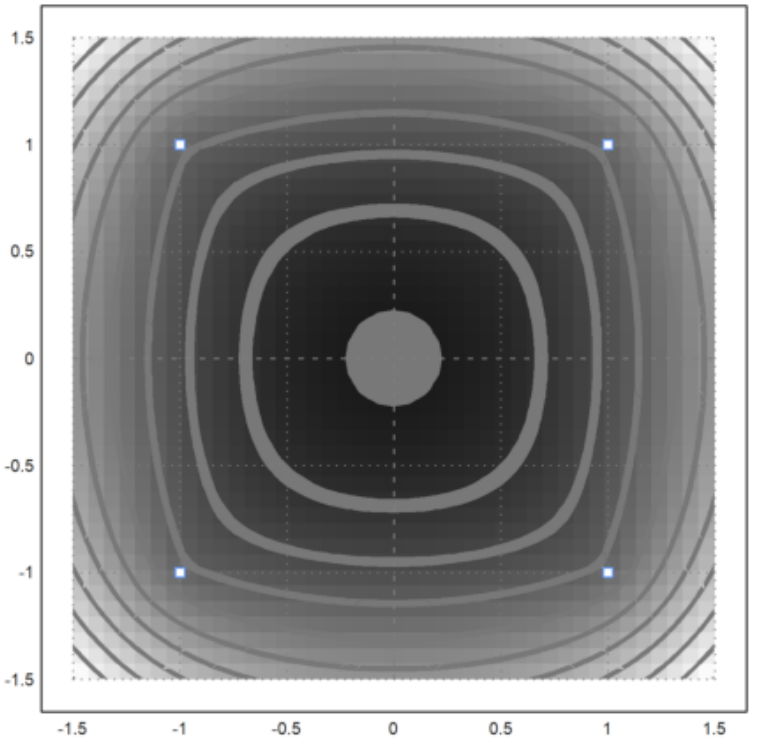
Tampaknya dalam kasus ini, koordinat titik optimal bersifat rasional atau mendekati rasional...

Karena ABCD adalah persegi, kita mengharapkan titik optimalnya adalah pusat ABCD:

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




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


1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

Contoh 7: Bola Dandelin dengan Povray

Anda dapat menjalankan demonstrasi ini jika Anda telah menginstal Povray dan pvengine.exe di jalur program.

Pertama, kita hitung jari-jari bola.

Jika Anda melihat gambar di bawah, Anda akan melihat bahwa kita membutuhkan dua lingkaran yang menyentuh dua garis yang membentuk kerucut, dan satu garis yang membentuk bidang yang memotong kerucut.

Kita menggunakan berkas geometry.e milik Euler untuk ini.

```
>load geometry;
```

Pertama dua garis membentuk kerucut.

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

```
[- a, 1, 0]
```

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

```
[- a, - 1, 0]
```

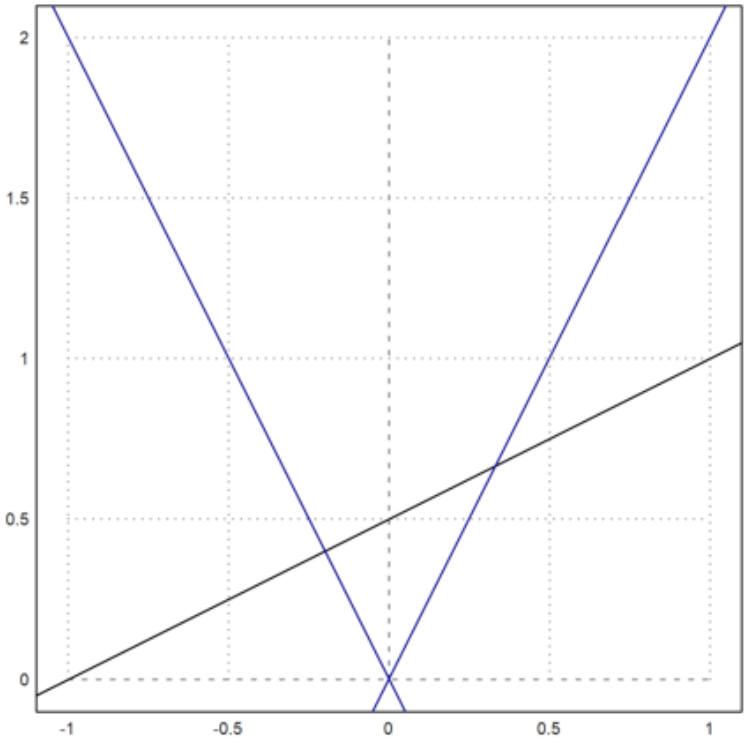
Lalu baris ketiga.

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

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

Kami merencanakan segalanya sejauh ini.

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



Sekarang kita ambil titik umum pada sumbu y.

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

```
[0, u]
```

Hitunglah jarak ke g1.

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

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

Hitunglah jarak ke g.

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

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

Dan temukan pusat kedua lingkaran, yang jaraknya sama.

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

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

Ada dua solusi.

Kita evaluasi solusi simbolisnya, lalu temukan kedua pusat dan kedua jaraknya.

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

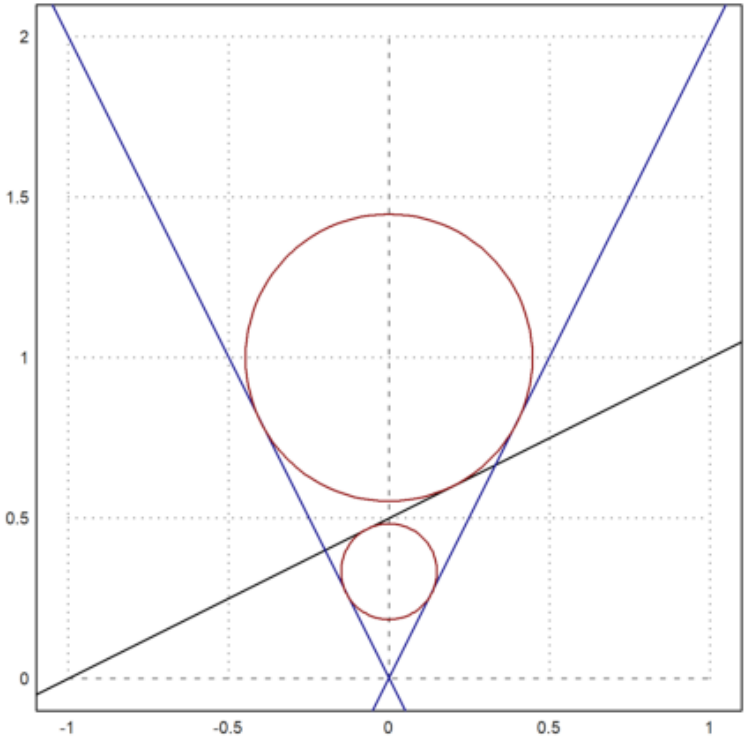
```
  [0.333333,  1]
```

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

```
  [0.149071,  0.447214]
```

Gambarkan lingkaran-lingkaran tersebut ke dalam gambar.

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


2
1.5
1
0.5
0
-1
-0.5
0
0.5
1

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:\Users\Audrey Sava\Documents\POV-Ray\v3.7\ini\pvengine.exe"
```

```
  C:\Users\Audrey Sava\Documents\POV-Ray\v3.7\ini\pvengine.exe
```

Kami menyiapkan suasananya dengan tepat.

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

Berikutnya kita menulis kedua bola tersebut ke berkas Povray.

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

Dan kerucutnya, transparan.

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

Kami membuat bidang yang dibatasi pada kerucut.

```
>gp=g();
>pc=povcone([0,0,0],0,[0,0,a],1,"");
>vp=[gp[1],0,gp[2]]; dp=gp[3];
>writeln(povplane(vp,dp,povlook(blue,0.5),pc));
```

Sekarang kita buat dua titik pada lingkaran, di mana bola menyentuh kerucut.

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

Kemudian, kita buat dua titik di mana bola-bola tersebut menyentuh bidang tersebut. Titik-titik inilah yang menjadi fokus elips.

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

Berikutnya kita hitung perpotongan P1P2 dengan bidang.

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

Kita menghubungkan titik-titik dengan segmen garis.

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

Sekarang kita buat pita abu-abu, di mana bola-bola menyentuh kerucut.

```
>pcw=povcone([0,0,0],0,[0,0,a],1.01);
>pc1=povcylinder([0,0,P1[3]-defaultpointsize/2],[0,0,P1[3]+defaultpointsize/2],1);
>writeln(povintersection([pcw,pc1],povlook(gray)));
>pc2=povcylinder([0,0,P2[3]-defaultpointsize/2],[0,0,P2[3]+defaultpointsize/2],1);
>writeln(povintersection([pcw,pc2],povlook(gray)));
```

```
Variable pcw not found!
Error in:
writeln(povintersection([pcw,pc2],povlook(gray))); ...
                            ^
```

Mulai program Povray.

```
>povend();
```

```
Function povend not found.
Try list ... to find functions!
Error in:
povend(); ...
        ^
```

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

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

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

Anda memerlukan kacamata merah/cyan untuk menghargai efek berikut.

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

```
Function povanaglyph not found.
Try list ... to find functions!
Error in:
... ",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, kami menggunakan vektor dengan dua koordinat dalam radian (utara dan timur, nilai negatif untuk selatan dan barat). Berikut adalah koordinat Kampus FMIPA UNY.

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

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

You can print this position with sposprint (spherical position print).

```
>sposprint(FMIPA) // posisi garis lintang dan garis bujur FMIPA UNY
```

```
  S 7°46.467' E 110°23.050'
```

Mari kita tambahkan dua kota lagi, Solo dan Semarang.

```
>Solo=[rad(-7,-34.333),rad(110,49.683)]; Semarang=[rad(-6,-59.05),rad(110,24.533)];
>sposprint(Solo), sposprint(Semarang),
```

```
S 7°34.333' E 110°49.683'
S 6°59.050' E 110°24.533'
```

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

```
>br=svector(FMIPA,Solo); degprint(br[1]), br[2]*rearth(7°)->km // perkiraan jarak FMIPA-Solo
```

```
65°20'26.60''
53.8945384608
```

Ini perkiraan yang bagus. Rutin berikut bahkan menggunakan perkiraan yang lebih baik. Pada jarak sependek itu, hasilnya hampir sama.

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

```
Commands must be separated by semicolon or comma!
Found:  // perkiraan jarak FMIPA-Semarang (character 32)
You can disable this in the Options menu.
Error in:
esdist(FMIPA,Semarang)->" km" // perkiraan jarak FMIPA-Semaran ...
                         ^
```

Terdapat fungsi untuk heading, yang memperhitungkan bentuk elips bumi. Sekali lagi, kami mencetak dengan cara yang canggih.

```
>sdegprint(esdir(FMIPA,Solo))
```

```
      65.34°
```

Sudut suatu segitiga melebihi 180° pada bola.

```
>asum=sangle(Solo,FMIPA,Semarang)+sangle(FMIPA,Solo,Semarang)+sangle(FMIPA,Semarang,Solo); degprint(
```

```
  180°0’10.77’’
```

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

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

```
Commands must be separated by semicolon or comma!
Found:  // perkiraan luas segitiga FMIPA-Solo-Semarang (character 32)
You can disable this in the Options menu.
Error in:
(asum-pi)*rearth(48°)^2->" km^2" // perkiraan luas segitiga FM ...
                                 ^
```

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

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

```
2123.64310526 km^2
```

Kita juga dapat menambahkan vektor ke posisi. 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’
```

Mari kita lihat contoh yang lebih besar, Tugu Jogja dan Monas Jakarta (menggunakan Google Earth untuk mencari koordinatnya).

```
>Tugu=[-7.7833°,110.3661°]; Monas=[-6.175°,106.811944°];
>sposprint(Tugu), sposprint(Monas)
```

```
  S 7°46.998’ E 110°21.966’
  S 6°10.500’ E 106°48.717’
```

Menurut Google Earth, jaraknya adalah 429,66 km. Kami mendapatkan perkiraan yang baik.

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

```
Commands must be separated by semicolon or comma!
Found:  // perkiraan jarak Tugu Jogja - Monas Jakarta (character 32)
You can disable this in the Options menu.
Error in:
esdist(Tugu,Monas)->" km" // perkiraan jarak Tugu Jogja - Mona ...
                      ^
```

Judulnya sama dengan yang dihitung di Google Earth.

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

```
294°17’2.85’’
```

Namun, kita tidak lagi mendapatkan posisi target yang tepat jika kita menambahkan arah dan jarak ke posisi awal. Hal ini terjadi karena kita tidak menghitung fungsi invers secara tepat, melainkan mengambil perkiraan radius bumi di sepanjang jalur tersebut.

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

```
S 6°10.500’ E 106°48.717’
```

Namun, kesalahannya tidak besar.

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

```
  S 6°10.500' E 106°48.717'
```

Tentu saja, kita tidak 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 jaraknya, dengan memakai arah ke Monas, kita sampai di Tugu.

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

Hasilnya sangat jauh.

```
>sposprint(p), skmprint(esdist(p,Monas))
```

```
  S 6°11.250' E 106°48.372'
       1.529km
```

Sebagai contoh lain, mari kita ambil dua titik di bumi pada garis lintang yang sama.

```
>P1=[30°,10°]; P2=[30°,50°];
```

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°
```

Namun, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi, kita harus menyesuaikan arah kita di sepanjang jalan. Sebagai perkiraan, kita menyesuaikannya pada 1/10 dari total jarak.

```
>p=P1;  dist=esdist(P1,P2); ...
>  loop 1 to 10; dir=esdir(p,P2); sdegprint(dir), p=esadd(p,dir,dist/10); end;
```

```
79.69°
81.67°
83.71°
85.78°
87.89°
90.00°
92.12°
94.22°
96.29°
98.33°
```

Jaraknya tidak tepat, karena kita akan menambahkan sedikit kesalahan, jika kita mengikuti arah yang sama terlalu lama.

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

```
0.203km
```

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

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

```
       0.000km
```

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

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

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

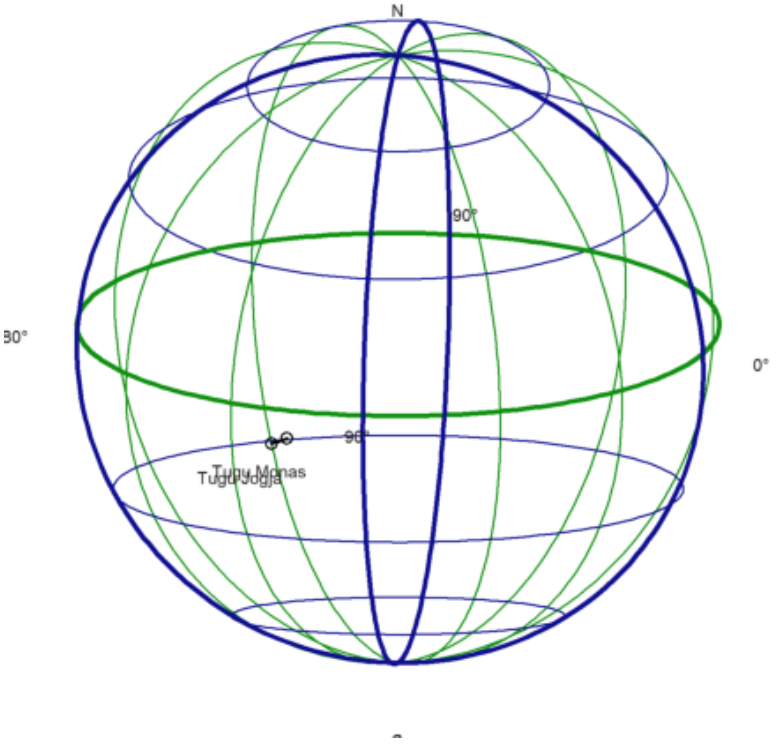
Kita menulis suatu fungsi yang memplot bumi, dua posisi, dan posisi di antaranya.

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

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

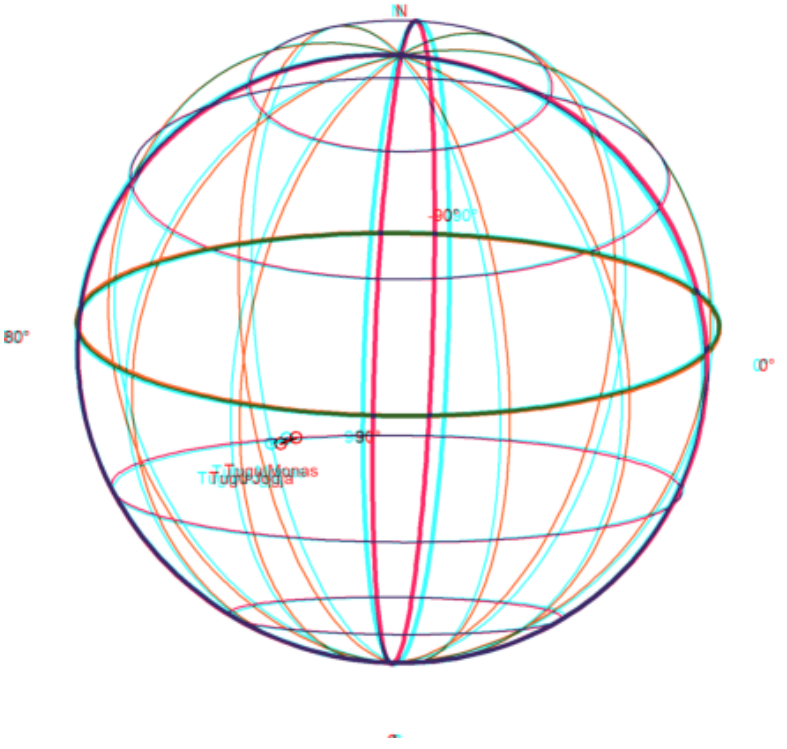
Sekarang rencanakan semuanya.

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



Or use plot3d to get an anaglyph view of it. This looks really great with red/cyan glasses.

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


N
90°
80°
0°
Tanay
Manas

Latihan

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

Petunjuk:

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

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

Petunjuk:
- Misalkan persamaan parabolanya y= axˆ2+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).

```
>n=5;
>r = 10;
>angle = 360/n;
>theta = (0:n-1)*angle;
>r=10;
>x=r*cos(theta)
```

```
  [10,  -9.67251,  8.71147,  -7.17985,  5.17796]
```

```
>y=r*sin(theta)
```

```
  [0,  2.53823,  -4.91022,  6.96058,  -8.55504]
```

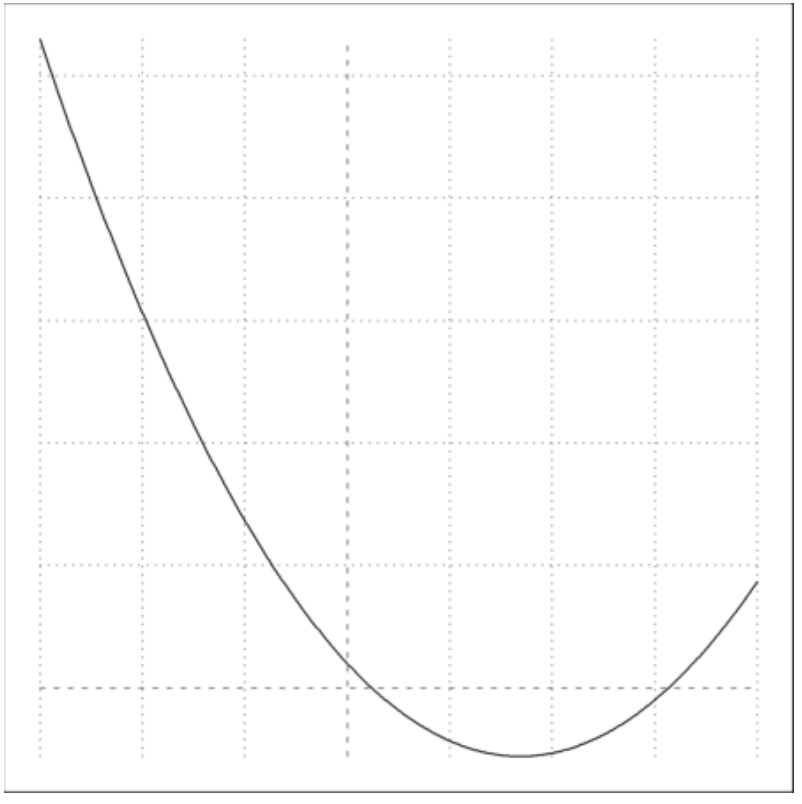
```
>x=x|x[1]
```

```
  [10,  -9.67251,  8.71147,  -7.17985,  5.17796,  10,  10]
```

```
>y=x|y[1]
```

```
  [10,  -9.67251,  8.71147,  -7.17985,  5.17796,  10,  10]
```

```
>plot2d(x,y,1);
>hold on;
>plot2d(x,y,-2);
>plot2d(0,0,style="points",color=red,add=true);
>("Segi-"+n+"beraturan dengan jari-jari"+r);
>x1:=-1; y1:=6;
>x2:=2; y2:=3;
>x3:=4; y3:=15;
>d1:=x2-x1;
>d2:=x3-x1;
>r1:=(y2-y1)/d1;
>r2:=(y3-y1)/d2;
>a:=(r2-r1)/(x3-x2);
>b:=r1-a*(x1+x2);
>c:=y1-a*x1^2-b*x1;
>fx:=y1-a*x1^2-(b*x1);
>function f(x):= a*x^2+b*x+c;
>plot2d("f(x)", x1-2, x3+2):
```



```
>circle([x1,y1],0.08);
```

```
Function circle not found.
Try list ... to find functions!
Error in:
circle([x1,y1],0.08); ...
                      ^
```

```
>x1:=1; y1:=2;
>x2:=5; y2:=6;
>dx:=x2-x1;
>dx
```

```
  4
```

```
>m:=(y2-y1)/dx;
>m
```

```
  1
```

```
>c:=y1- m*x1;
>c
```

```
  1
```

```
>res:=[m,c];
>res
```

```
  [1,  1]
```

```
>xmin :=x1-2
```

-1

```
>xmax :=x2+2
```

7

```
>dxplot:=0.5
```

0.5

```
>pts:=[];
>x:=xmin;
>xnext:=x+dx;
>y:=m*x+c;
>ynext:=m*xnext+c;
```