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 plotkan.

>A=[1,0]; plotPoint(A,“A”); // definisi dan gambar tiga titik

>B=[0,1]; plotPoint(B,“B”);

>C=[2,2]; plotPoint(C,“C”);

Kemudian tiga segmen.

>plotSegment(A,B,“c”); // c=AB

>plotSegment(B,C,“a”); // a=BC

>plotSegment(A,C,“b”); // b=AC

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

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

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

Hitung garis tegak lurus yang melalui A 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) mencari panjang AD

```
1.5
```

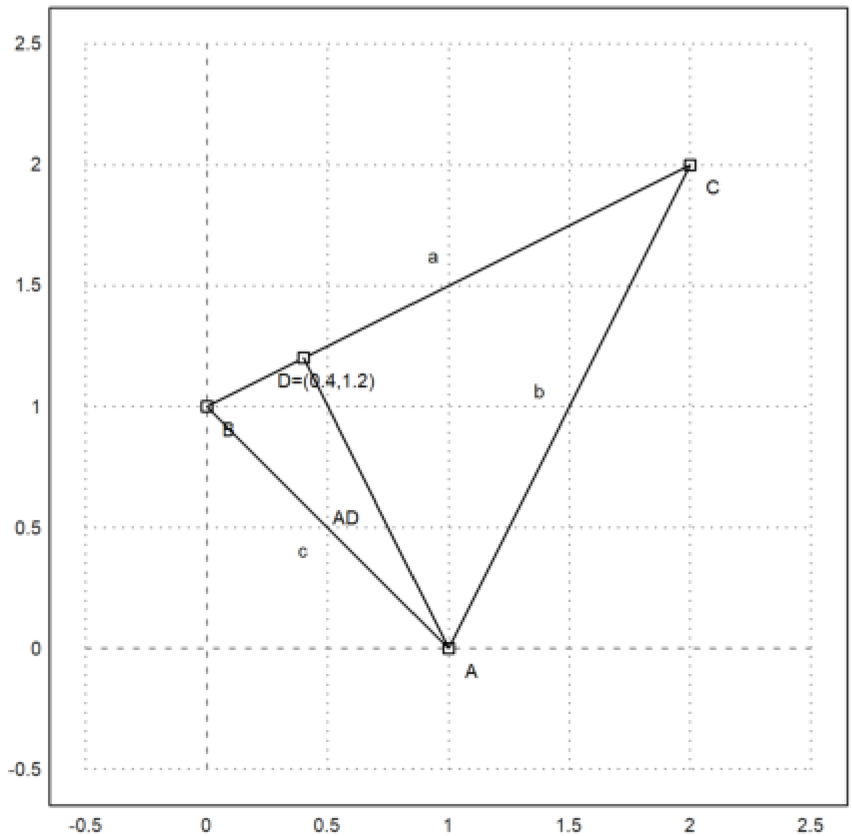

Figure 1: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-001.png

Bandingkan dengan rumus determinan.

menggunakan determinan dari 3 titik yang sudah diketahui A(1,0), B(0,1), C(2,2), dengan rumus:

$$\frac{1}{2}$$

>areaTriangle(A,B,C) // hitung luas segitiga langusng dengan fungsi menggunakan rumus determinan

```
1.5
```

Cara lain menghitung luas segitigas ABC:

>distance(A,D)*distance(B,C)/2 //jarak dari 2 titik

```
1.5
```

Menghitung besar sudut

Sudut pada C.

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

```
36°52'11.63''
```

Menggambar Lingkaran Luar Segitiga

Sekarang, lingkarilah segitiga tersebut.

>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”):

Dari gambar tersebut kita bisa menghitung koordinat titik pusat dan jari-jari lingkaran luar dengan perintah tersebut. Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

>O, R

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

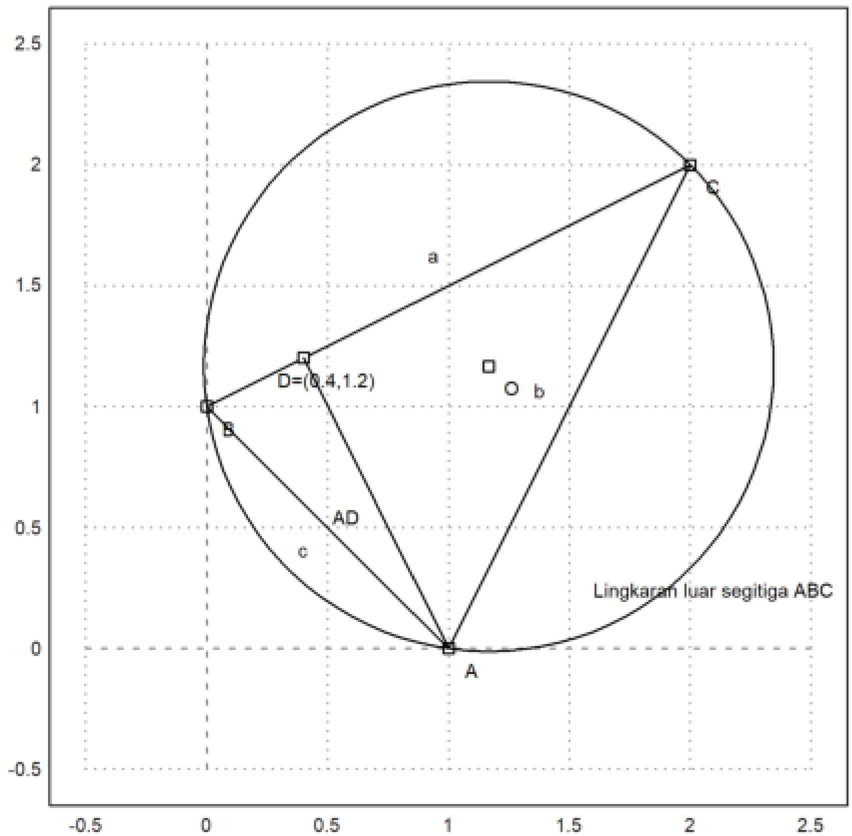

Figure 2: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-004.png

Menggambar Lingkaran Dalam Segitiga

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]
```

Setelah didapat titik koordinat P, tambahkan semuanya ke plot.

>color(5); plotLine(l); plotLine(g); color(1); // gambar kedua garis bagi sudut

>plotPoint(P,“P”); // gambar titik potongnya

>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam

```
Function lineThrough needs a vector for B
Error in:
r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingk ...
                                          ^
```

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

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

The functions for lines and circle work just like the Euler functions, but provide symbolic computations.

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

$$[-1, 2, 2]$$

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?

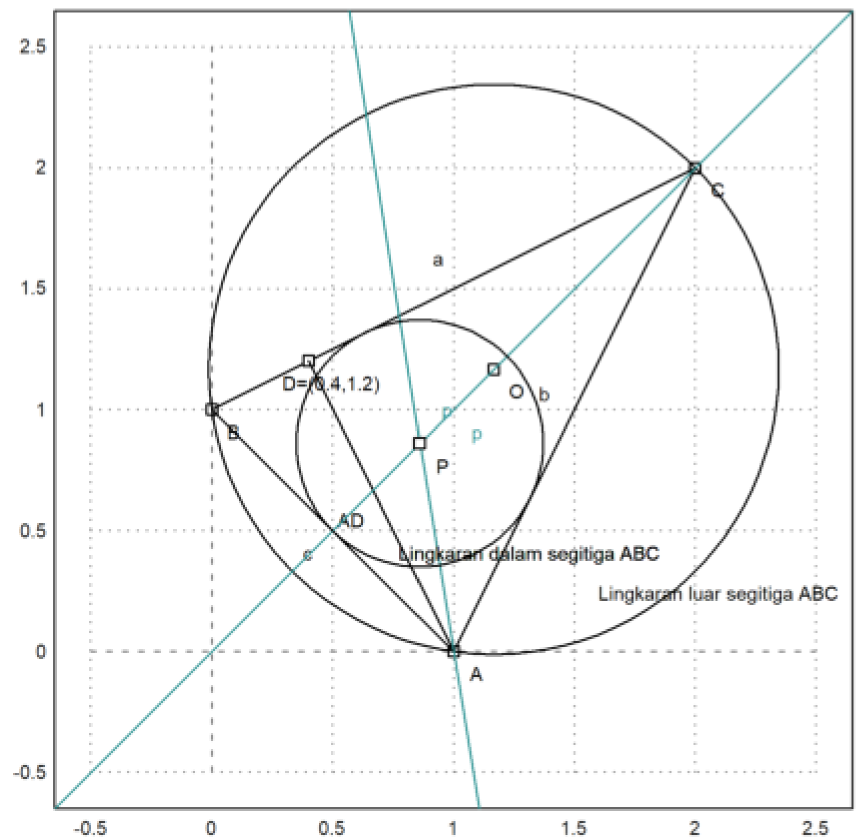

Figure 3: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-005.png

>load geometry

```
Numerical and symbolic geometry.
```

>plotCircle(circleWithCenter(P,r),“Lingkaran dalam segitiga ABC”);

>K = lineCircleIntersections(lineThrough(A,B),circleWithCenter(P,r))

```
[1.23607,  -3]
```

>L = lineCircleIntersections(lineThrough(C,B),circleWithCenter(P,r))

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

>M = lineCircleIntersections(lineThrough(C,A),circleWithCenter(P,r))

```
[-3.12132,  -3.12132]
```

>plotPoint(K,“K”); plotPoint(M,“M”); plotPoint(L,“L”):

Bentuk dari segitiga ABC yaitu segitiga sama kaki

>plotSegment(K,L,“m”);

>plotSegment(L,M,“n”);

>plotSegment(M,K,“l”):

>distance(P,lineIntersection(lineThrough(A,B),l))

```
4.472135955
```

>getCircleRadius(circleWithCenter(P,r))

```
3
```

Menghitung Luas Segitiga

>areaTriangle(A,B,C)

```
18
```

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

```
18
```

Mencari persamaan garis menggunakan metode simbolik

Kita bisa mendapatkan persamaan untuk sebuah garis dengan mudah.

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

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

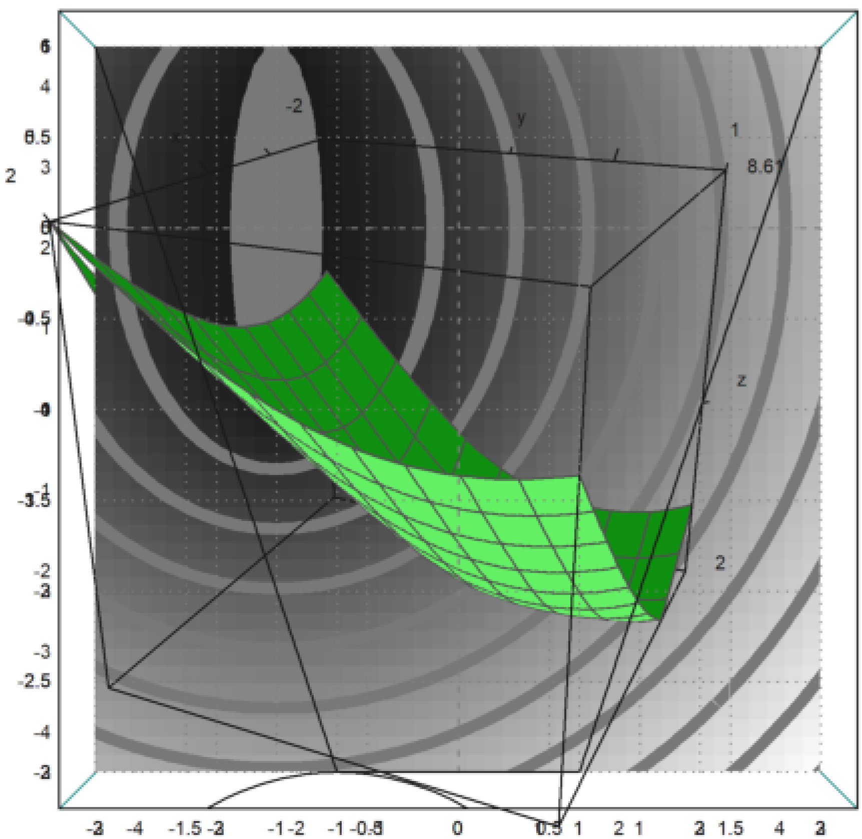

Figure 4: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-006.png

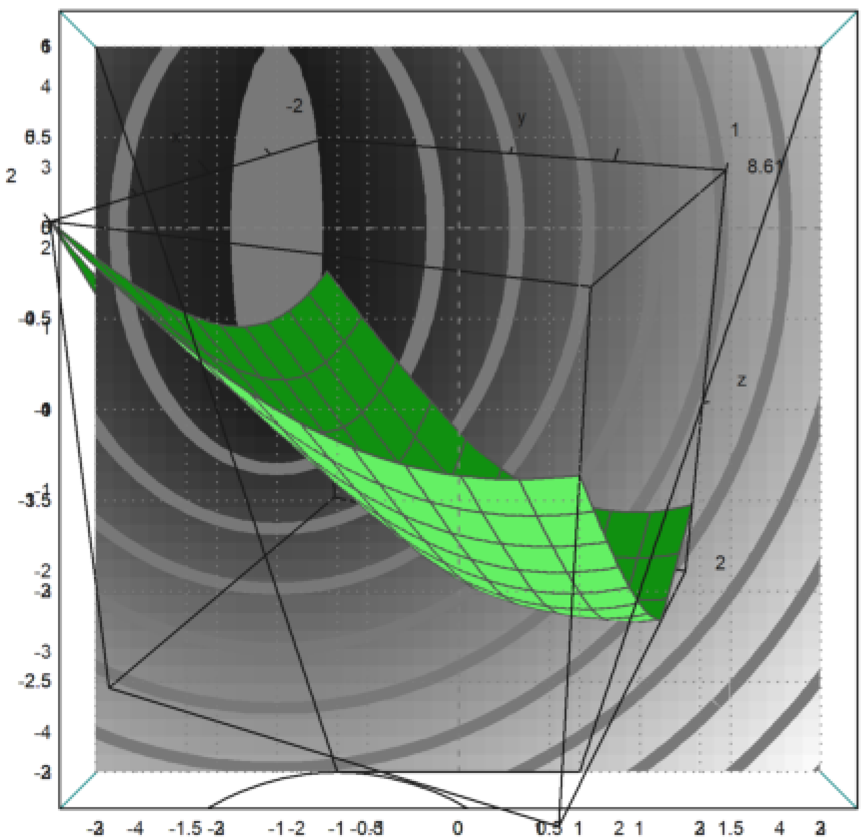

Figure 5: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-007.png

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

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

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

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

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

$$x\ (A-y_1)+y\ (x_1-A)=[A-y_1,x_1-A]\cdot A$$

Selanjutnya, kita mencari garis yang tegak lurus dengan BC dan melalui titik A

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

```
[2, 1, [2, 1] . A]
```

2. Mencari titik potong garis h dengan garis BC

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

```
 2  6
[-, -]
 5  5
```

Mencari proyeksi daroi titik A ke titik D. MAksudnya, kita akan menarik garis dari titik A menuju titik D. dan akan menghasilkan koordinat yang sama

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

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

3. Mencari panjang garis AD

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

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

4. Menentukan titik pusat dan jari-jari

menggunakan fungsi geometri simbolik

>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

$$1.178511301977579$$

1.178511301977579

5. Mencari nilai sudut

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

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

6. Mencari persamaan garis bagi dan titik potong sudut ACB

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

$$y = x$$

mencari titik potong 2 garis bagi tersebut

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

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

>P() // hasilnya sama dengan perhitungan sebelumnya

```
[0.86038,  0.86038]
```

Perpotongan Garis dan Lingkaran

Tentu saja, kita juga bisa memotong garis dengan lingkaran, dan lingkaran dengan lingkaran.

>A &:= [1,0]; c=circleWithCenter(A,4); //lingkaran dengan pusat A dan jari-jari 4

>B &:= [1,2]; C &:= [2,1]; l=lineThrough(B,C); //garis yang melalui titik B dan C

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

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

Mencari titik koordinat P1 dan P2 dalam (x,y)

>{P1,P2,f}=lineCircleIntersections(l,c); // titik potong garis l dan lingkaran c

>P1, P2, f

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

>plotPoint(P1); plotPoint(P2):

Hal yang sama pada Maxima.

>c &= circleWithCenter(A,4) // lingkaran dengan pusat A(1,0) dan jari-jari 4

```
[1, 0, 4]
```

menampilkan persamaan garis l yang melalui garis BC

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

```
[1, 1, 3]
```

Hasil [1,1,3] mempresentasikan garis dengan persamaan x+y=3

Mencari titik potong antara

>$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.

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

>degprint(computeAngle(P1,C,P2))

```
69°17'42.68''
```

dengan perintah yang sama

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

>degprint(computeAngle(P1,C,P2))

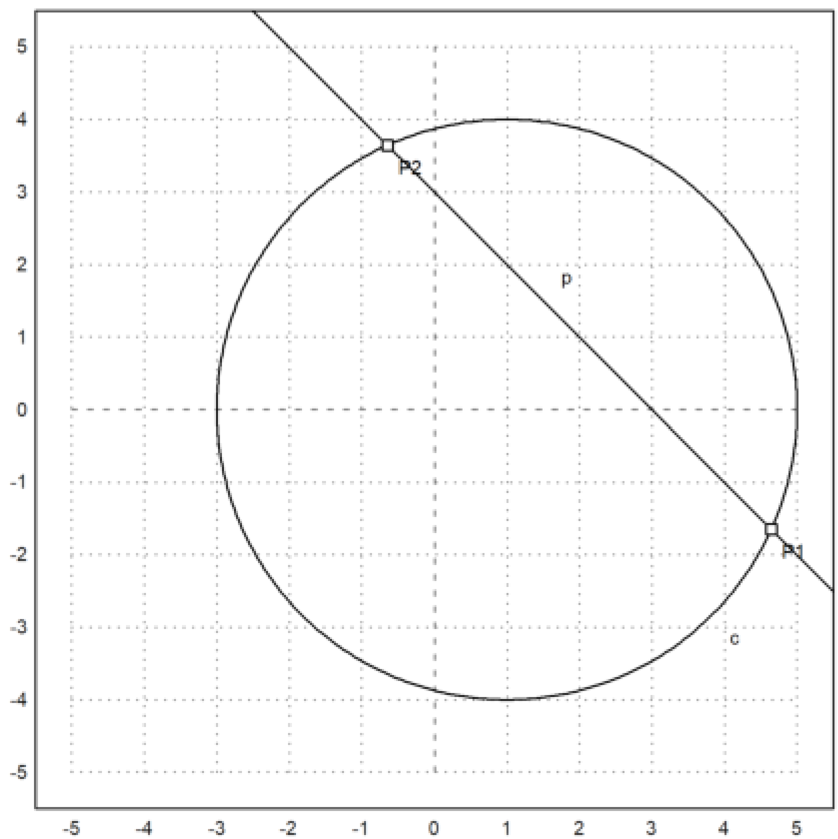

Figure 6: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-021.png

```
69°17'42.68''
```

>insimg;

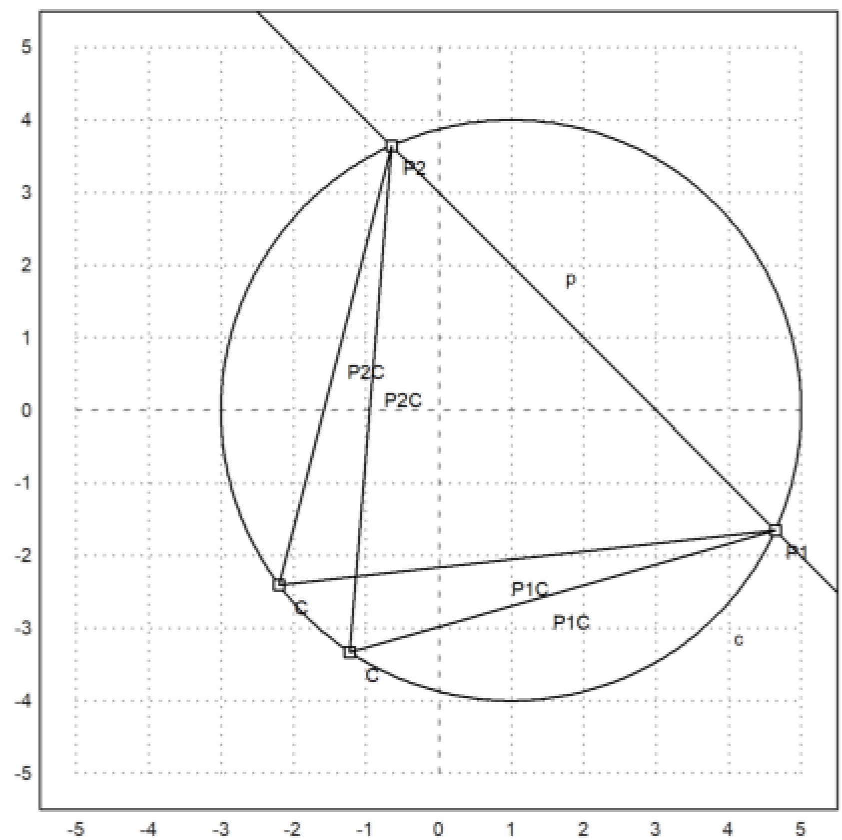

Figure 7: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-023.png

Garis Sumbu

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

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

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

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

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

Mencari titik potong antar dua lingkaran

>{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):

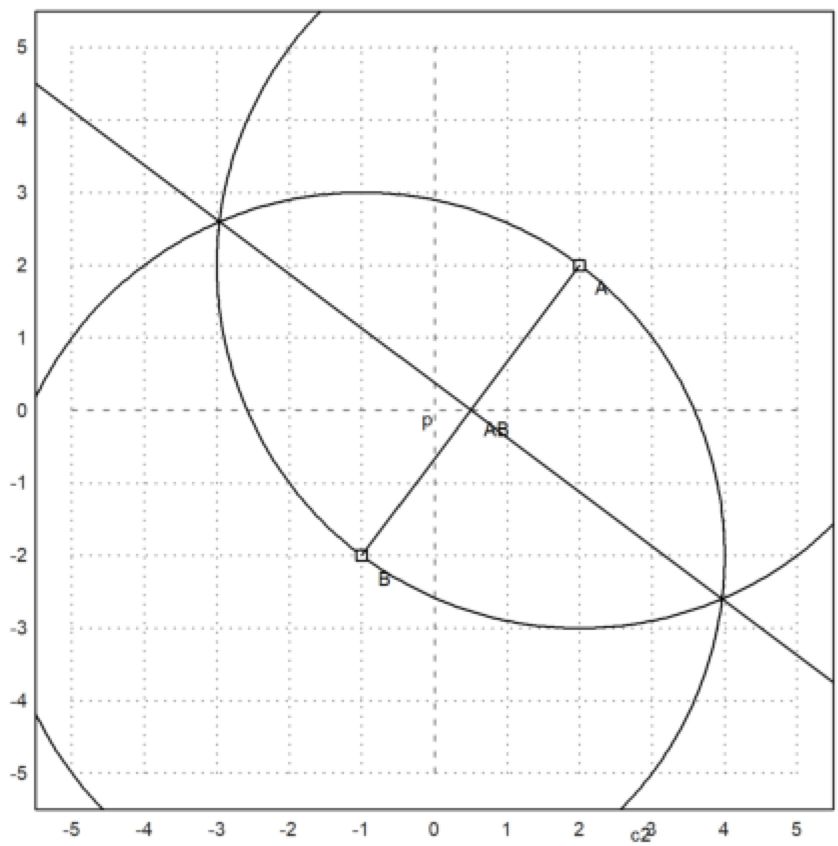

Figure 8: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-024.png

•

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

-Membuat lingkaran dan mencari titik potong

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

>c1 &= circleWithCenter(A,distance(A,B)); // Lingkaran dgn titik pusat A

>c2 &= circleWithCenter(B,distance(A,B)); // Lingkaran dgn titik pusat B

>P &= circleCircleIntersections(c1,c2); P1 &= P[1]; P2 &= P[2]; // titik potong kedua lingkaran

>function f(x)

Persamaan untuk persimpangan cukup rumit. Tetapi kita dapat menyederhanakannya, jika kita menyelesaikan untuk 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 sama sekali 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 larutan y.

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

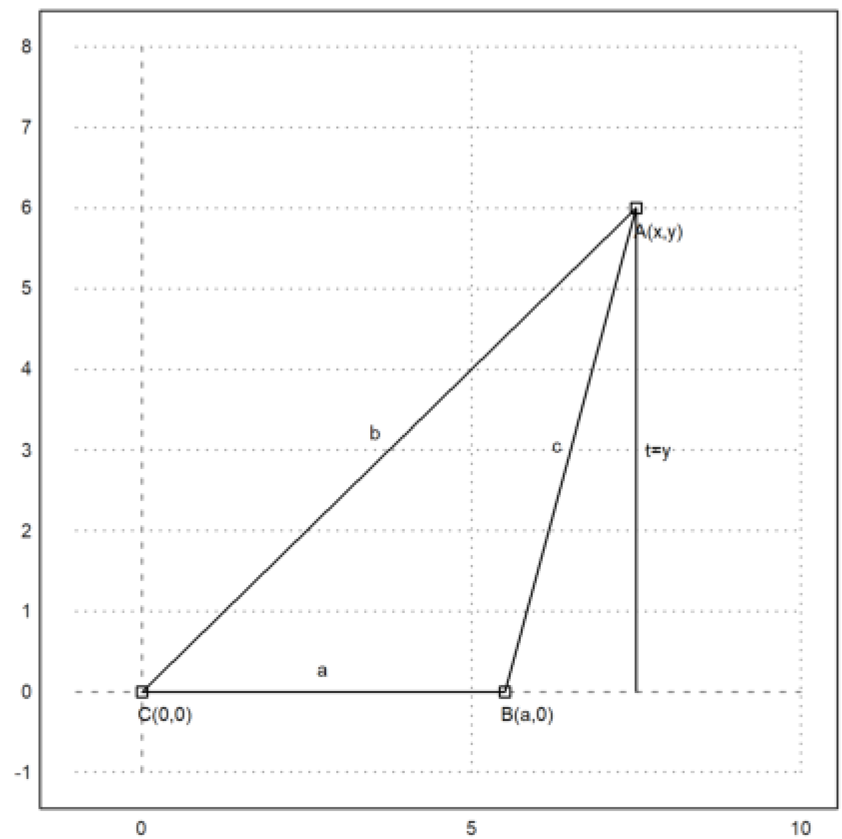

Figure 9: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-033.png

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

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

Kami 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 persegi panjang adalah kasus yang terkenal.

>H(4,7,9) //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(4,7,9) //luas segitiga siku-siku dengan panjang sisi 3, 4, 5 ...
        ^
```

Dan juga jelas, bahwa ini adalah segitiga dengan luas maksimal dan kedua sisi 3 dan 4.

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

```
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 bisa digunakan.

>$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. Sudah diketahui bahwa ini adalah sebuah elips.

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

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

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

Dan membuat fungsi-fungsi 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)

$$0$$

0

Sekarang kita dapat menggambar himpunan tersebut. Sisi b bervariasi dari 1 hingga 4. Sudah diketahui bahwa kita mendapatkan sebuah 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 pusat, dan u dan v adalah setengah sumbu.

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

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

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

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

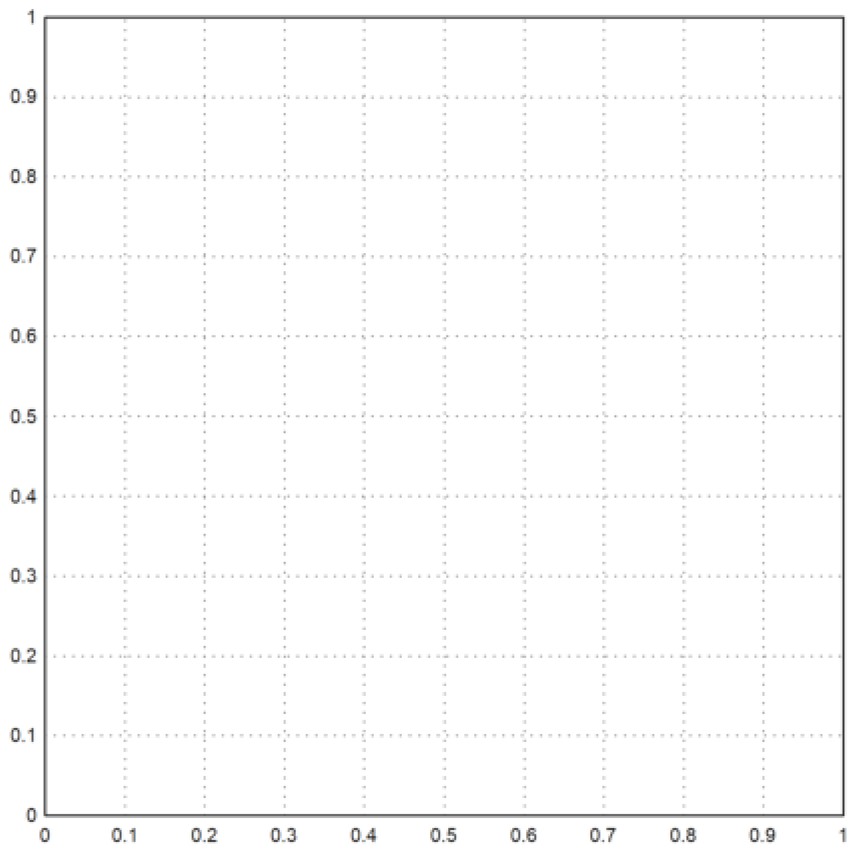

Figure 10: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-038.png

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

Kita mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusi 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 sehubungan dengan 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 situasi

Pertama-tama, tetapkan 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 &amp;= at([x,y],sol[2]); $A ...
                     ^
```

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

$$[a, 0]$$

$[a, 0]$

Kemudian, tetapkan kisaran plot, dan plot titik-titiknya.

>setPlotRange(0,10,-2,10); . . .
> a=5; b=5; c=5; . . .
> plotPoint(mxmeval("B"),"B"); plotPoint(mxmeval("C"),"C"); . . .
> plotPoint(mxmeval("A"),"A"):

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

Plot segmen-segmen tersebut.

>plotSegment(mxmeval(“A”),mxmeval(“C”)); . . .
> plotSegment(mxmeval(“B”),mxmeval(“C”)); . . .
> plotSegment(mxmeval(“B”),mxmeval(“A”)):

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

Hitung garis tegak lurus tengah dalam Maxima.

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

Dan bagian tengah keliling.

>p &= lineIntersection(h,g);

```
Maxima said:
part: invalid index of list or matrix.
#0: lineIntersection(g=(b1-a1)*y+(a2-b2)*x = a1*(a2-b2)+a2*(b1-a1),h=2*(b2-a2)*sqrt(-((b2-a2)^2+(
 -- an error. To debug this try: debugmode(true);

Error in:
p &amp;= lineIntersection(h,g); ...
                          ^
```

Kita mendapatkan rumus untuk jari-jari lingkaran.

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

$$\sqrt{2}\,|U|$$

Mari kita tambahkan ini ke dalam plot.

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

```
Function U not found.
Try list ... to find functions!
Error in:
plotPoint(U()); plotCircle(circleWithCenter(mxmeval("U"),mxmev ...
                ^
```

Dengan menggunakan geometri, kami memperoleh rumus sederhana

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

untuk jari-jari. Kita bisa mengeceknya dengan Maxima, apakah ini benar dengan Maxima. Maxima akan memperhitungkannya hanya jika kita mengkuadratkannya.

>\$c$^{2/\sin(\text{computeAngle(A,B,C)})}$2 | factor

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

Contoh 4: Garis Euler dan Parabola

Garis Euler adalah garis yang ditentukan dari segitiga apa pun yang tidak sama sisi. Garis ini merupakan garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk ortosentrum, circumcentrum, centroid, titik Exeter, dan pusat lingkaran sembilan titik segitiga.

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

Pertama, kita mendefinisikan sudut-sudut segitiga dalam Euler.Kami menggunakan definisi, yang terlihat dalam ekspresi simbolis.

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

Untuk memplot objek geometris, kita menyiapkan area plot, dan menambahkan titik-titiknya. Semua plot objek geometris ditambahkan ke plot saat ini.

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

Kita juga bisa menambahkan sisi-sisi segitiga.

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

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

>\$areaTriangle(A,B,C)

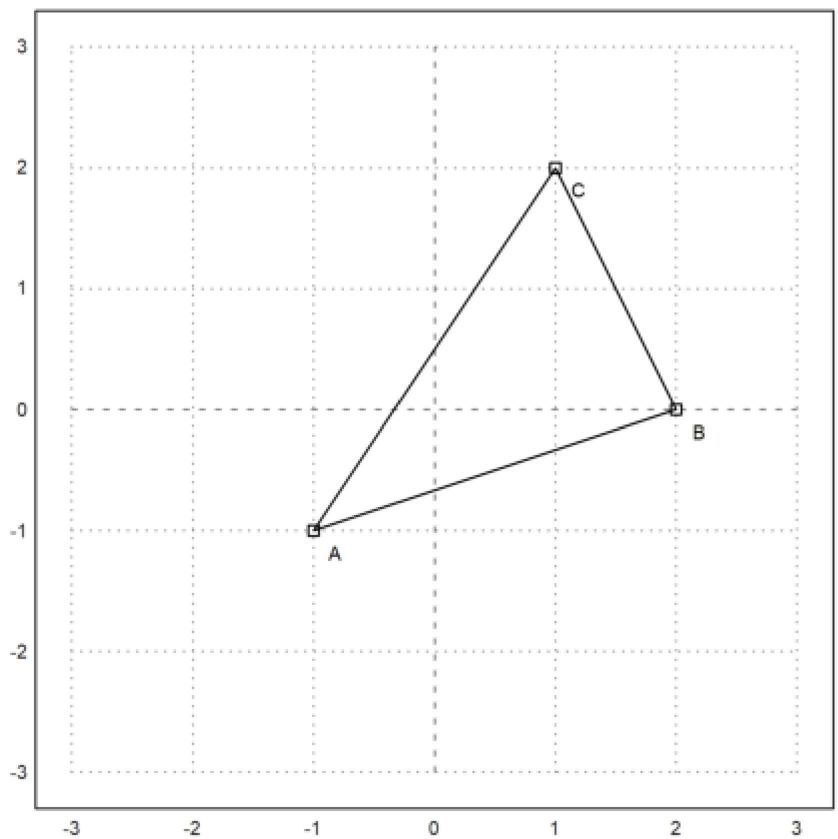

Figure 11: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-048.png

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

Kita dapat menghitung koefisien dari sisi c.

>c &= lineThrough(A,B)

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

Dan juga mendapatkan formula 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 dari bentuk Hesse. Memasukkan titik tersebut akan menghasilkan jarak positif ke garis.

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

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

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

Lingkaran luar dan Dalam

Sekarang kita menghitung keliling ABC.

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

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

>O &= getCircleCenter(LL); $O

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

Plot lingkaran dan pusatnya. Cu dan U adalah simbolik. Kami mengevaluasi ekspresi ini untuk Euler.

>plotCircle(LL()); plotPoint(O(),“O”):

Kita dapat menghitung perpotongan ketinggian di ABC (pusat ortosentrum) secara numerik dengan perintah berikut ini.

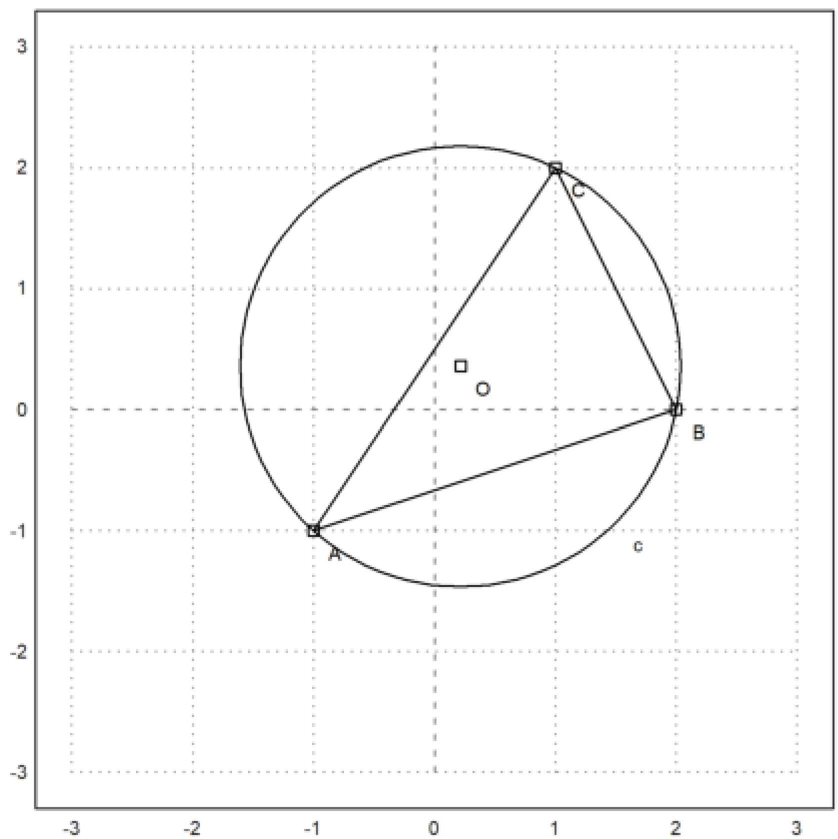

Figure 12: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-055.png

>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 kami.

>plotPoint(H(),“H”); plotLine(el(),“Garis Euler”):

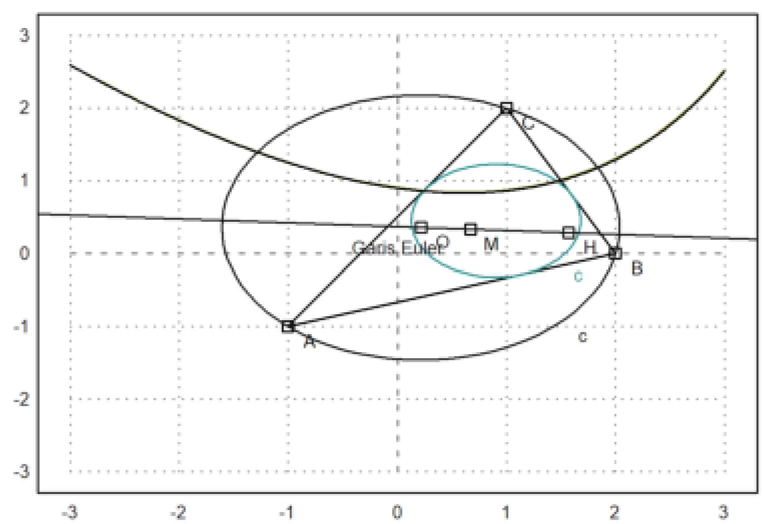

Figure 13: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-058.png

Pusat gravitasi harus berada pada garis ini.

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

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

>plotPoint(M(),“M”): // titik berat

Teori mengatakan bahwa MH = 2*MO. Kita perlu menyederhanakan dengan radcan untuk mencapai hal ini.

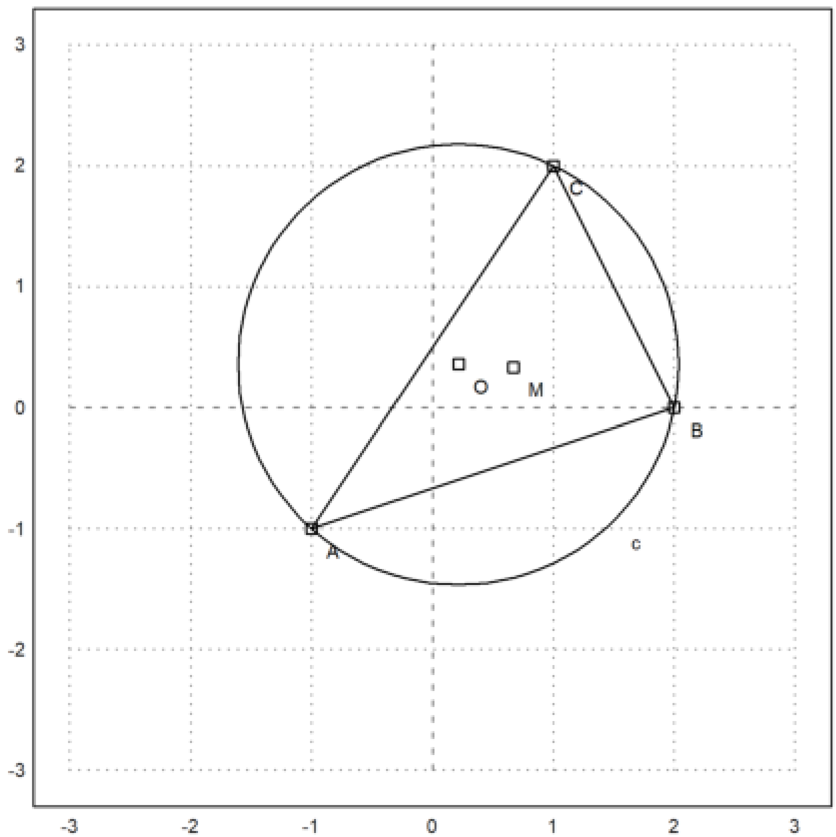

Figure 14: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-060.png

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

$$2$$

Fungsi-fungsi ini juga mencakup fungsi untuk sudut.

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

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

60°15’18.43’’

Persamaan untuk bagian tengah lingkaran tidak terlalu bagus.

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

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

Mari kita hitung juga ekspresi untuk jari-jari lingkaran yang 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 plot.

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

Parabola

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

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

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

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

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

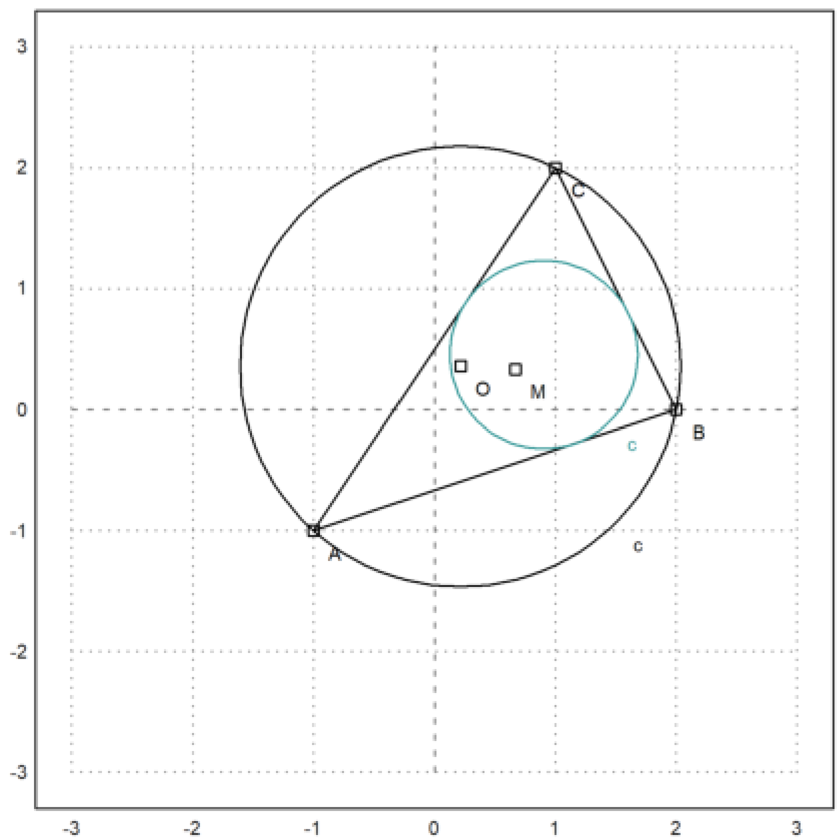

Figure 15: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-065.png

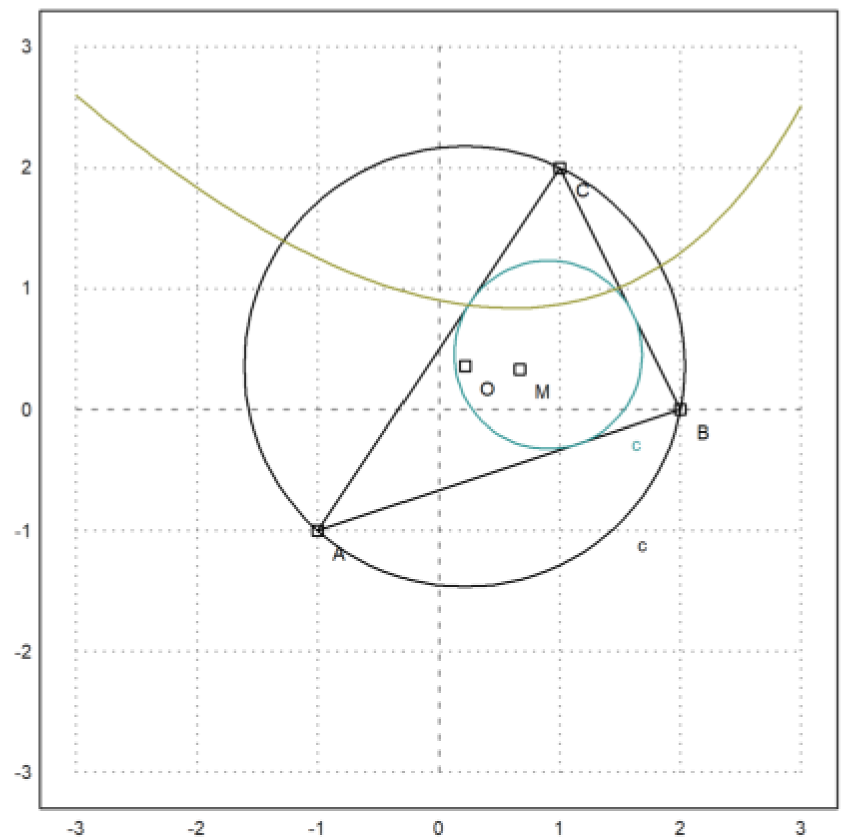

Figure 16: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-067.png

Ini seharusnya merupakan suatu fungsi, tetapi solver default Maxima hanya bisa menemukan solusinya jika kita mengkuadratkan persamaannya. Akibatnya, kita mendapatkan solusi palsu.

>akar &= solve(getHesseForm(lineThrough(A,B),x,y,C)$^{\text{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

$$y=-3\,x-\sqrt{70}\,\sqrt{9-2\,x}+26$$

Menambahkan solusi pertama ke dalam plot menunjukkan, bahwa ini memang jalur yang kita cari. Teori mengatakan bahwa ini adalah sebuah parabola yang diputar.

>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

$$2.135605779339061$$

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

$$2.135605779339061$$

2.135605779339061

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

Contoh 5: Trigonometri Rasional

Ini terinspirasi dari sebuah ceramah N.J. Wildberger. Dalam bukunya “Proporsi Ilahi”, Wildberger mengusulkan untuk mengganti gagasan klasik tentang jarak

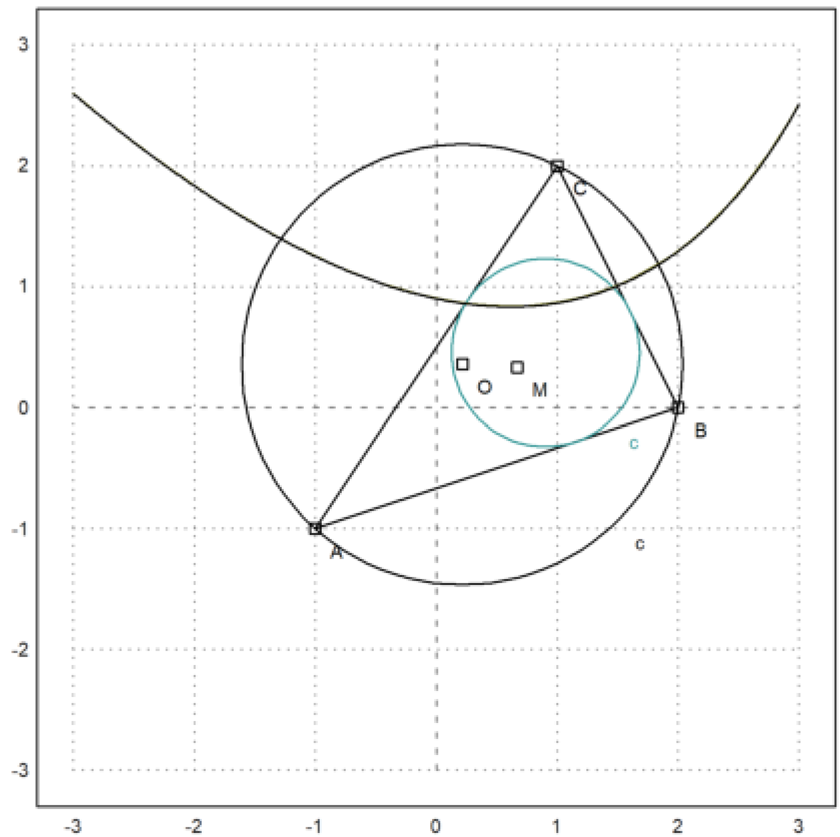

Figure 17: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-069.png

dan sudut dengan kuadransi dan penyebaran.Dengan menggunakan ini, memang memungkinkan untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

Berikut ini, saya akan memperkenalkan konsep-konsep tersebut, dan memecahkan beberapa masalah. Saya menggunakan komputasi simbolik Maxima di sini, yang menyembunyikan keuntungan utama dari trigonometri rasional yaitu komputasi dapat dilakukan dengan kertas dan pensil saja. Anda dipersilakan untuk memeriksa hasilnya tanpa komputer.

Intinya adalah bahwa komputasi rasional simbolik sering kali memberikan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang dievaluasi dengan pendekatan numerik saja.

>load geometry;

Untuk pengenalan pertama, kita menggunakan segitiga persegi panjang dengan proporsi Mesir yang terkenal 3, 4, dan 5. Perintah berikut ini adalah perintah Euler untuk memplot geometri bidang yang terdapat pada file Euler "geometry.e".

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

Tentu saja,

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

di mana wa adalah sudut di A. Cara biasa untuk menghitung sudut ini, adalah dengan mengambil kebalikan dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak kira-kira.

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

```
36°52'11.63''
```

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah kuadrat, yang menggantikan jarak. Sebenarnya, ini hanyalah jarak yang dikuadratkan. Berikut ini, 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
```

Gagasan kedua dari trigonometri rasional adalah penyebaran. Penyebaran mengukur bukaan di antara garis-garis. Ini adalah 0, jika garis-garisnya sejajar,

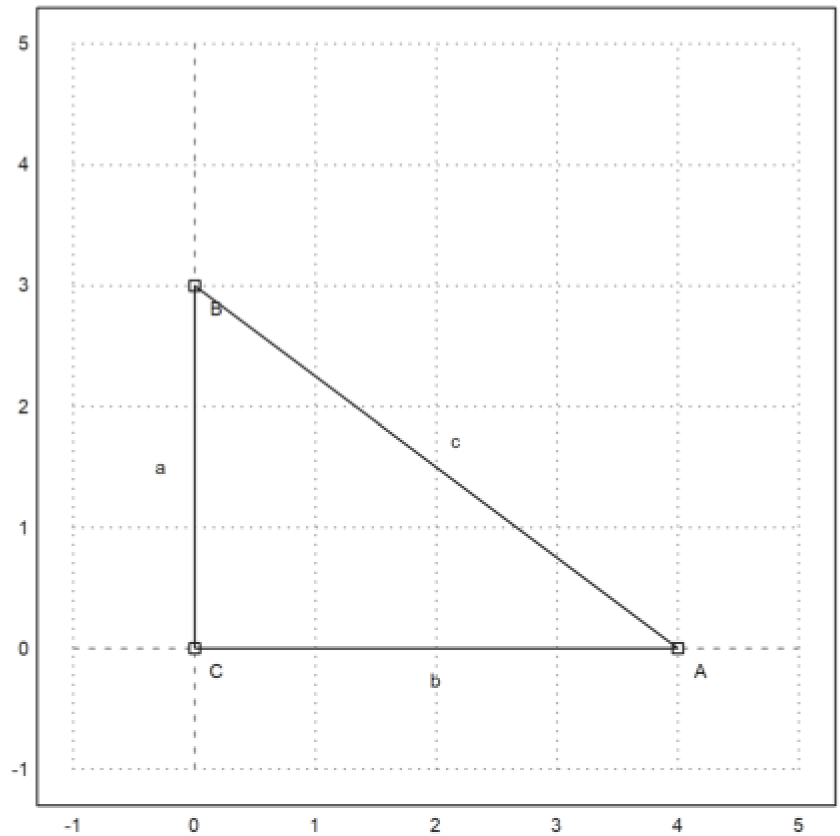

Figure 18: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-076.png

dan 1, jika garis-garisnya persegi panjang. Ini adalah kuadrat dari sinus sudut antara dua garis.

Penyebaran 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 segitiga persegi panjang dengan satu sudut di A.

>sa &= a/c; $sa

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

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

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

>fracprint(sin(wa)^2)

```
9/25
```

Hukum kosinus trgonometri klasik diterjemahkan ke dalam "hukum silang" berikut ini.

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

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

Hukum-hukum ini diimplementasikan dalam file geometri.e yang kita masukkan ke dalam Euler.

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

$$\left(cc + bb - aa\right)^2 = 4\,bb\,cc\,\left(1 - saa\right)$$

Dalam kasus kami, kami mendapatkan

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

$$1024 = 1024$$

Mari kita gunakan crosslaw ini untuk mencari sebaran di A. Untuk melakukannya, kita buat crosslaw untuk kuadran a, b, dan c, dan selesaikan untuk sebaran sa yang tidak diketahui.

Anda bisa melakukan ini dengan tangan dengan mudah, tapi saya menggunakan Maxima. Tentu saja, kita mendapatkan hasil yang sudah kita dapatkan.

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

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

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

Kita sudah mengetahui hal ini. Definisi penyebaran adalah kasus khusus dari crosslaw.

Kita juga dapat menyelesaikannya untuk a, b, c secara umum. Hasilnya adalah sebuah rumus yang menghitung penyebaran sudut sebuah segitiga dengan kuadran ketiga sisinya.

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

$$\left[x = \frac{-cc^2 - (-2\,bb - 2\,aa)\;cc - bb^2 + 2\,aa\,bb - aa^2}{4\,bb\,cc}\right]$$

Kita dapat membuat sebuah fungsi dari hasil tersebut. Fungsi seperti itu sudah didefinisikan dalam file geometry.e dari Euler.

>$spread(a,b,c)

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

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

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

Hasilnya adalah 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;

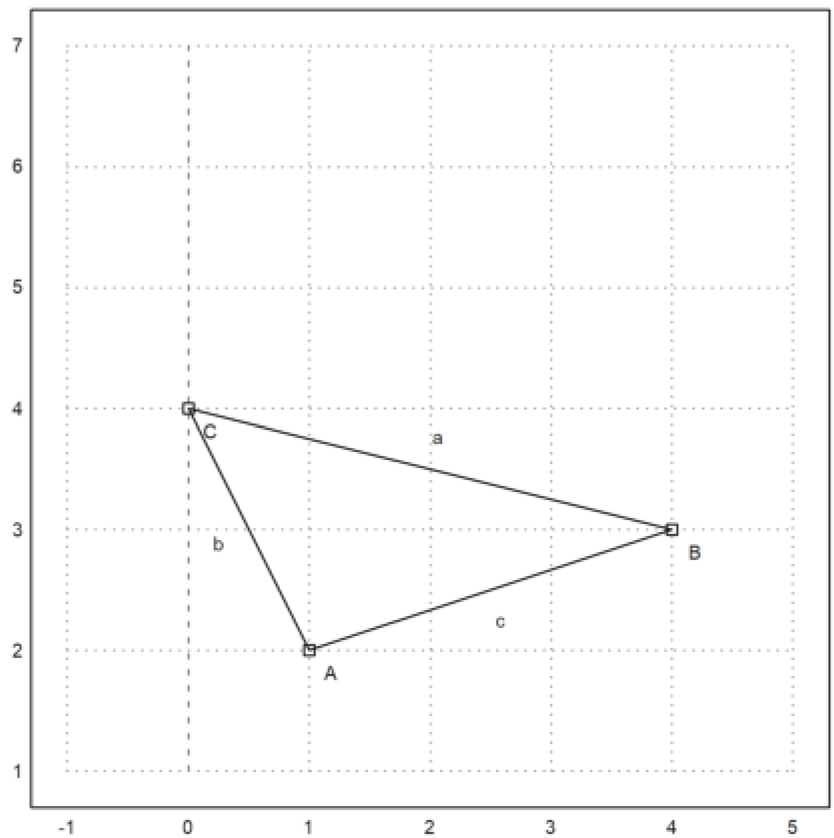

Figure 19: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-089.png

Dengan menggunakan Pythogoras, mudah untuk menghitung jarak antara dua titik. Pertama-tama saya menggunakan jarak fungsi dari file Euler untuk geometri. Jarak fungsi menggunakan geometri klasik.

>$distance(A,B)

$$\sqrt{10}$$

Euler juga memiliki fungsi untuk kuadranan antara dua titik.

Pada contoh berikut, karena c+b bukan a, maka segitiga tersebut tidak berbentuk persegi panjang.

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

$$17$$

5

$$17$$

Figure 20: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-093.png

Pertama, mari kita menghitung sudut tradisional. Fungsi computeAngle menggunakan metode yang biasa berdasarkan hasil kali titik dari dua vektor. Hasilnya adalah beberapa perkiraan titik mengambang.

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

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

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

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

>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)

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

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

Itulah yang dilakukan oleh fungsi spread yang didefinisikan dalam "geometry.e".

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

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

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

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

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

Setelah kita memiliki penyebaran di B, kita dapat menghitung tinggi ha di sisi a. Ingatlah bahwa

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

menurut definisi.

>ha &= c*sb; $ha

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

Gambar berikut ini dibuat dengan program geometri C.a.R., yang dapat menggambar kuadran dan penyebaran.

image: (20) Rational_Geometry_CaR.png

Menurut definisi, panjang ha adalah akar kuadrat dari kuadrannya.

>$sqrt(ha)

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

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita berurusan dengan kuadran!

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

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

Rumus penentu 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 menghitung penyebaran di B untuk segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat ("quadrea"?), memfaktorkannya dengan Maxima, dan kita mendapatkan rumus Heron yang terkenal.

Memang, hal ini sulit dilakukan dengan pensil dan kertas.

>\$spread(b^{2,c}2,aˆ2), \$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 Triple Spread

Kerugian dari spread adalah bahwa mereka tidak lagi hanya menambahkan sudut seperti.

Namun, tiga spread dari sebuah segitiga memenuhi aturan "triple spread" berikut ini.

>&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 berjumlah 180°.

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

Karena spread dari

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

adalah sama, aturan triple spread juga benar, jika

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

Karena penyebaran sudut negatif adalah sama, aturan triple spread juga berlaku, jika

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

Sebagai contoh, kita bisa menghitung penyebaran sudut 60°. Ini adalah 3/4. Namun, persamaan ini memiliki solusi kedua, di mana semua penyebarannya adalah 0.

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

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

Penyebaran 90° jelas adalah 1. Jika dua sudut ditambahkan ke 90°, penyebarannya akan menyelesaikan persamaan penyebaran tiga dengan a, b, 1. Dengan perhitungan berikut, kita mendapatkan a + b = 1.

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

$$[x = 1 - y]$$

$[x = 1 - y]$

Karena penyebaran 180°-t sama dengan penyebaran t, rumus penyebaran tiga kali lipat juga berlaku, jika satu sudut adalah jumlah atau selisih dari dua sudut lainnya.

Jadi kita dapat menemukan penyebaran sudut dua kali lipat. Perhatikan bahwa ada dua solusi lagi. Kita jadikan ini sebuah fungsi.

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

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

- 4 (a - 1) a

Pembagi Sudut

Ini adalah situasi yang sudah kita ketahui.

>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;

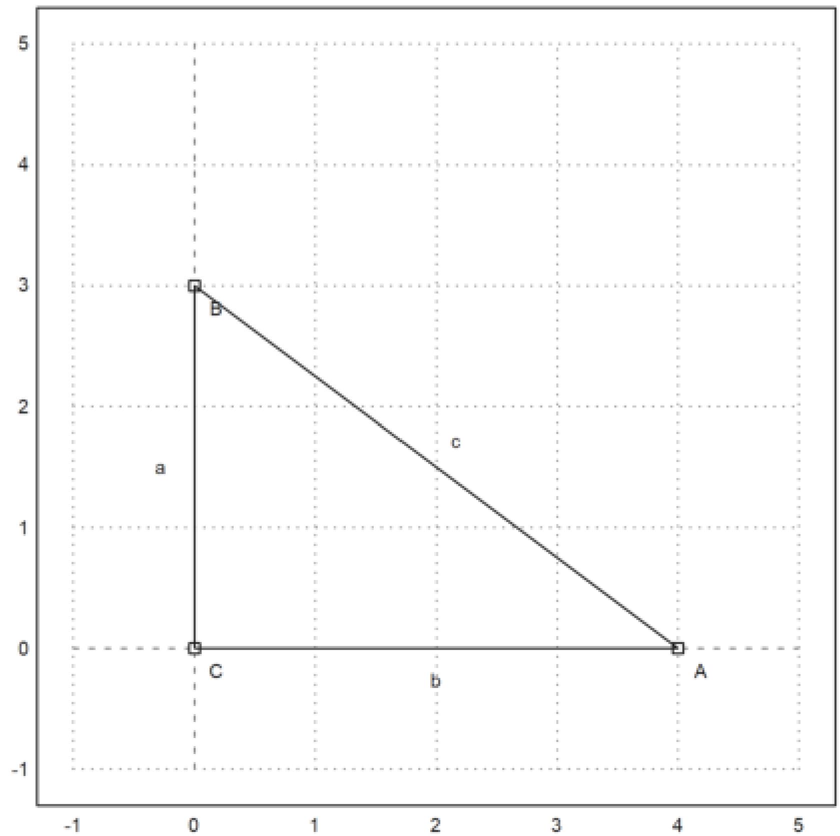

Figure 21: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-119.png

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

>&remvalue(a,b,c);

Jadi, pertama-tama kita menghitung penyebaran sudut yang dibelah dua di A, menggunakan rumus penyebaran tiga.

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

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

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

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

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

Figure 22: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-122.png

Mari kita periksa persegi panjang Mesir.

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

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

Kita bisa mencetak sudut dalam Euler, setelah mentransfer penyebaran ke radian.

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

```
18°26'5.82''
```

Titik P adalah perpotongan garis bagi sudut dengan sumbu y.

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

```
[0,  1.33333]
```

>plotPoint(P,“P”); plotSegment(A,P):

Mari kita periksa sudut-sudutnya dalam contoh spesifik kita.

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

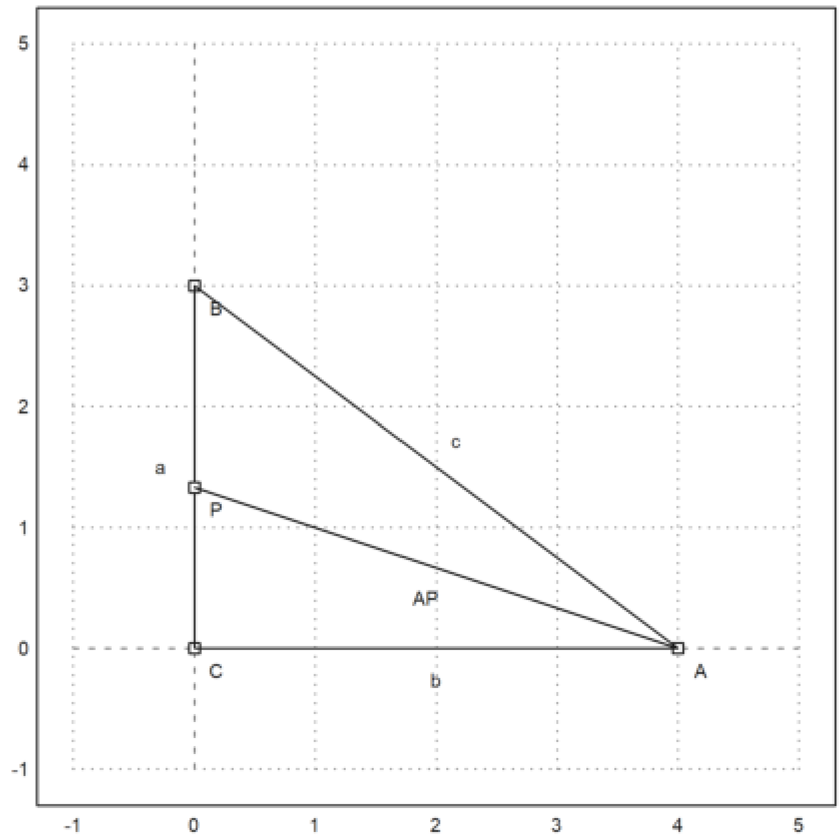

Figure 23: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-124.png

```
0.321750554397
0.321750554397
```

Sekarang kita hitung panjang garis bagi AP.

Kita menggunakan teorema sinus dalam segitiga APC. Teorema ini menyatakan bahwa

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

berlaku dalam segitiga apa pun. Kuadratkan, ini diterjemahkan ke dalam apa yang disebut “spread law” atau “hukum penyebaran”

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

di mana a, b, c menunjukkan kuadrannya.

Karena spread CPA adalah 1-sa2, kita mendapatkan bisa/1=b/(1-sa2) dan bisa menghitung bisa (kuadran dari pembagi sudut).

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

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

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 dengan 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 Akor

Lihatlah situasi berikut ini.

>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;

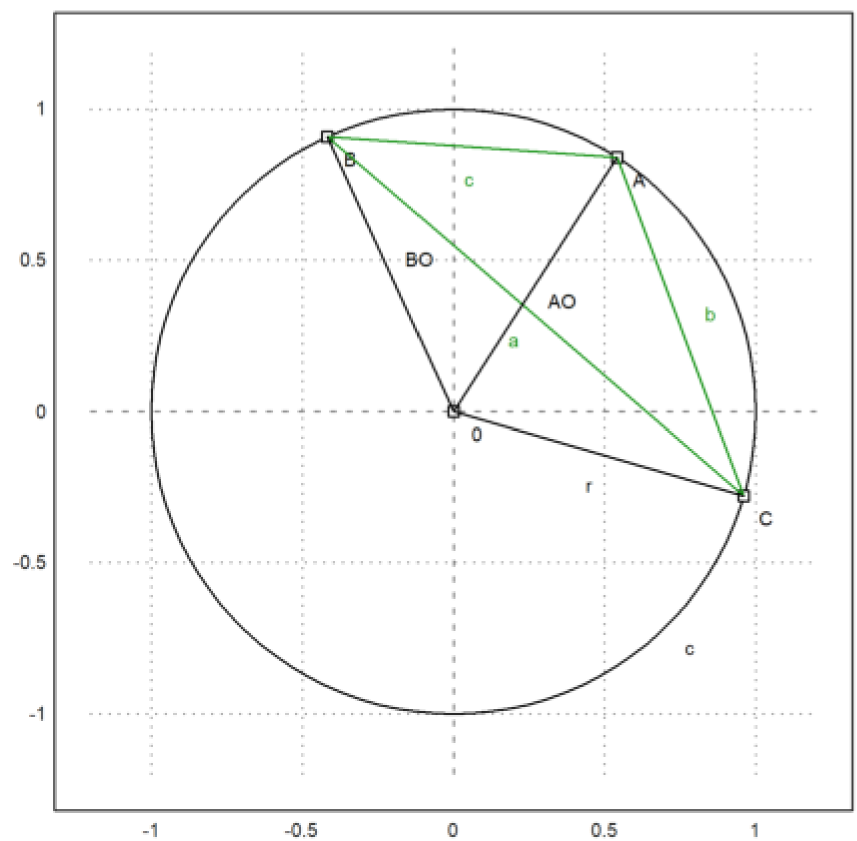

Figure 24: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-129.png

Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran tiga untuk sudut-sudut di pusat O untuk r. Dengan demikian kita mendapatkan rumus untuk jari-jari kuadrat dari pericircle dalam hal kuadran sisi-sisinya.

Kali ini, Maxima menghasilkan beberapa angka nol yang rumit, 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

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

Kita dapat menjadikannya sebuah fungsi Euler.

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

Mari kita periksa hasil untuk poin A, B, C. Mari kita periksa hasil untuk poin A, B, C.

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

Radiusnya memang 1.

>periradius(a,b,c)

```
1
```

Faktanya adalah, bahwa penyebaran CBA hanya bergantung pada b dan c. Ini adalah teorema sudut akor.

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

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

Faktanya, penyebarannya adalah b/(4r), dan kita melihat bahwa sudut chord b adalah setengah dari sudut tengah.

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

$$0$$

Contoh 6: Jarak Minimal pada Bidang

Preliminary remark

The function which, to a point M in the plane, assigns the distance AM between a fixed point A and M, has rather simple level lines: circles centered in 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"):

and the graph is rather simple too: the upper part of a cone:

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

Of course the minimum 0 is attained in A.

Two points

Now we look at the function MA+MB where A and B are two points (fixed). It is a "well-known fact" that the level curves are ellipses, the focal points being A and B; except for the minimum AB which is constant on the segment [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):

The graph is more interesting:

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

The restriction to line (AB) is more famous:

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

Three points

Now things are less simple: It is a little less well-known that MA+MB+MC attains its minimum at one point of the plane but to determine it is less simple:

1) If one of the angles of the triangle ABC is more than 120° (say in A), then the minimum is attained at this very point (say AB+AC).

Example:

>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);

>insim;

```
>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) But if all the angles of triangle ABC are less than 120°, the minimum is on a point F in the interior of the triangle, which is the only point which sees the sides of ABC with the same angles (then 120° each):

```
>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;
```

It is an interesting activity to realize the above figure with a geometry software; for example, I know a soft written in Java which has a “contour lines” instruction...

All of this above have been discovered by a french judge called Pierre de Fermat; he wrote letters to other dilettants like the priest Marin Mersenne and Blaise Pascal who worked at the income taxes. So the unique point F such that FA+FB+FC is minimal, is called the Fermat point of the triangle. But it seems that a few years before, the italian Torriccelli had found this point before Fermat did! Anyway the tradition is to note this point F...

Four points

The next step is to add a 4th point D and try and minimize MA+MB+MC+MD; say that you are a cable TV operator and want to find in which field you must put your antenna so that you can feed four villages and use as little cable length as possible!

```
>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;
```

There is still a minimum and it is attained at none of the vertices A, B, C nor D:

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

>neldermin("f",[0.2,0.2])

```
[0.142858,  0.142857]
```

It seems that in this case, the coordinates of the optimal point are rational or near-rational...

Now ABCD is a square we expect that the optimal point will be the center of 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

You can run this demonstration, if you have Povray installed, and pvengine.exe in the program path.

First we compute the radii of the spheres.

If you look at the figure below, you see that we need two circles touching the two lines which form the cone, and one line which forms the plane cutting the cone.

We use the geometry.e file of Euler for this.

>load geometry;

First the two lines forming the cone.

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

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

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

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

Thenm a third line.

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

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

We plot everything so far.

>setPlotRange(-1,1,0,2);

>color(black); plotLine(g(),“ “)

>a:=2; color(blue); plotLine(g1(),“ “), plotLine(g2(),” “):

Now we take a general point on the y-axis.

>P &= [0,u]

```
[0, u]
```

Compute the distance to g1.

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

Compute the distance to g.

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

And find the centers of the two circles, where the distances are equal.

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

There are two solutions.

We evaluate the symbolic solutions, and find both centers, and both distances.

>u := sol()

```
[0.333333,  1]
```

>dd := d()

```
[0.149071,  0.447214]
```

Plot the circles into the figure.

>color(red);

>plotCircle(circleWithCenter([0,u[1]],dd[1]),“ “);

>plotCircle(circleWithCenter([0,u[2]],dd[2]),“ “);

>insimg;

Plot with Povray

Next we plot everything with Povray. Note that you change any command in the following sequence of Povray commands, and rerun all commands with Shift-Return.

First we load the povray functions.

>load povray;

>defaultpovray=“C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe”

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

We setup the scene appropriately.

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

Next we write the two spheres to the Povray file.

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

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

And the cone, transparent.

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

We generate a plane restricted to the cone.

>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));

Now we generate two points on the circles, where the spheres touch the cone.

>function turnz(v) := return

>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)));

Then we generate the two points where the spheres touch the plane. These are the foci of the ellipse.

>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)));

Next we compute the intersection of P1P2 with the plane.

>t1=scalp(vp,P1)-dp; t2=scalp(vp,P2)-dp; P5=P1+t1/(t1-t2)*(P2-P1);

>writeln(povpoint(P5,povlook(yellow)));

We connect the points with line segments.

>writeln(povsegment(P1,P2,povlook(yellow)));

>writeln(povsegment(P5,P3,povlook(yellow)));

>writeln(povsegment(P5,P4,povlook(yellow)));

Now we generate a gray band, where the spheres touch the cone.

>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)));

Start the Povray program.

>povend();

To get an Anaglyph of this we need to put everything into a scene function. This function will be used twice later.

>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
```

You need red/cyan glasses to appreciate the following effect.

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

Contoh 8: Geometri Bumi

In this notebook, we want to do some spherical computations. The functions are contained in the file “spherical.e” in the examples folder. We need to load that file first.

>load “spherical.e”;

To enter a geographical position, we use a vector with two coordinates in radians (north and east, negative values for south and west). The following are the coordinates for the Campus of the 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'
```

Let us add two more towns, Solo and 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'
```

First we compute the vector from one to the other on an ideal ball. This vector is [heading,distance] in radians. To compute the distance on the earth, we multiply with the earth radius at a latitude of 7°.

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

```
65°20'26.60''
53.8945384608
```

This is a good approximation. The following routines use even better approximations. On such a short distance the result is almost the same.

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

```
88.0114026318 km
```

There is a function for the heading, taking the elliptical shape of the earth into account. Again, we print in an advanced way.

>sdegprint(esdir(FMIPA,Solo))

```
      65.34°
```

The angle of a triangle exceeds 180° on the sphere.

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

```
180°0'10.77''
```

This can be used to compute the area of the triangle. Note: For small triangles, this is not accurate due to the subtraction error in asum-pi.

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

```
2123.67056312 km^2
```

There is a function for this, which uses the mean latitude of the triangle to compute the earth radius, and takes care of rounding errors for very small triangles.

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

```
2123.64310526 km^2
```

We can also add vectors to positions. A vector contains the heading and the distance, both in radians. To get a vector, we use svector. To add a vector to a position, we use 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'
```

These functions assume an ideal ball. The same on the earth.

>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'
```

Let us turn to a larger example, 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'
```

According to Google Earth, the distance is 429.66km. We get a good approximation.

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

```
431.565659488 km
```

The heading is the same as the one computed in Google Earth.

>degprint(esdir(Tugu,Monas))

```
294°17'2.85''
```

However, we do no longer get the exact target position, if we add the heading and distance to the orginal position. This is so, since we do not compute the inverse function exactly, but take an approximation of the earth radius along the path.

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

```
S 6°10.500' E 106°48.717'
```

The error is not large, however.

>sposprint(Monas),

```
S 6°10.500' E 106°48.717'
```

Of course, we cannot sail with the same heading from one destination to another, if we want to take the shortest path. Imagine, you fly NE starting at any point on the earth. Then you will spiral to the north pole. Great circles do not follow a constant heading!

The following computation shows that we are way off the correct destination, if we use the same heading during our travel.

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

Now we add 10 times one-tenth of the distance, using the heading to Monas, we got in Tugu.

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

The result is far off.

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

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

As another example, let us take two points on the earth at the same lattitude.

>P1=[30°,10°]; P2=[30°,50°];

The shortest path from P1 to P2 is not the circle of lattitude 30°, but a shorter path starting 10° further north at P1.

>sdegprint(esdir(P1,P2))

```
79.69°
```

But, if we follow this compass reading, we will spiral to the north pole! So we must adjust our heading along the way. For rough purposes, we adjust it at 1/10 of the total distance.

>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°
```

The distances are not right, since we will add a bit off error, if we follow the same heading for too long.

>skmprint(esdist(p,P2))

```
0.203km
```

We get a good approximation, if we adjust out heading after each 1/100 of the total distance from Tugu to 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
```

For navigational purposes, we can get a sequence of GPS position along the great circle to Monas with the function 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'
```

We write a function, which plots the earth, the two positions, and the positions in between.

>function testplot ...

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

Now plot everything.

>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):

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.

>load geometry

```
Numerical and symbolic geometry.
```

>setPlotRange(-3.5,3.5,-3.5,3.5);

>A=[-2,-2]; plotPoint(A,“A”);

>B=[2,-2]; plotPoint(B,“B”);

>plotSegment(A,B,“c”);

>plotSegment(B,C,“a”);

>plotSegment(A,C,“b”);

>aspect(1):

>c=circleThrough(A,B,C);

>R=getCircleRadius(c);

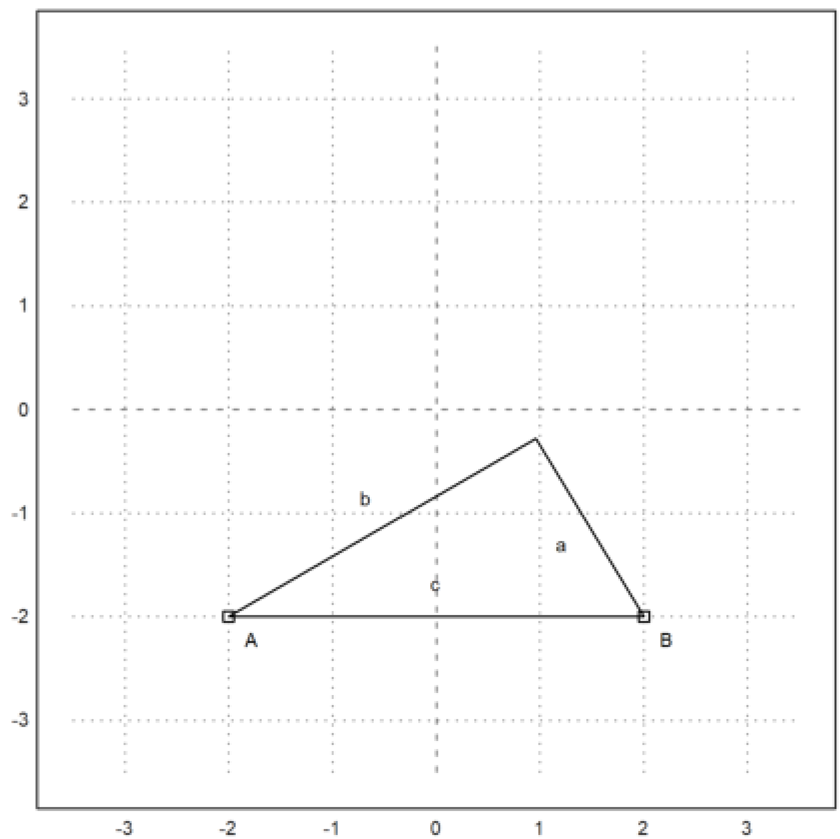

Figure 25: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-133.png

>O=getCircleCenter(c);

>plotPoint(O,“O”);

>l=angleBisector(A,C,B);

>color(2); plotLine(l); color(1);

>plotCircle(c,“Lingkaran luar segitiga ABC”):

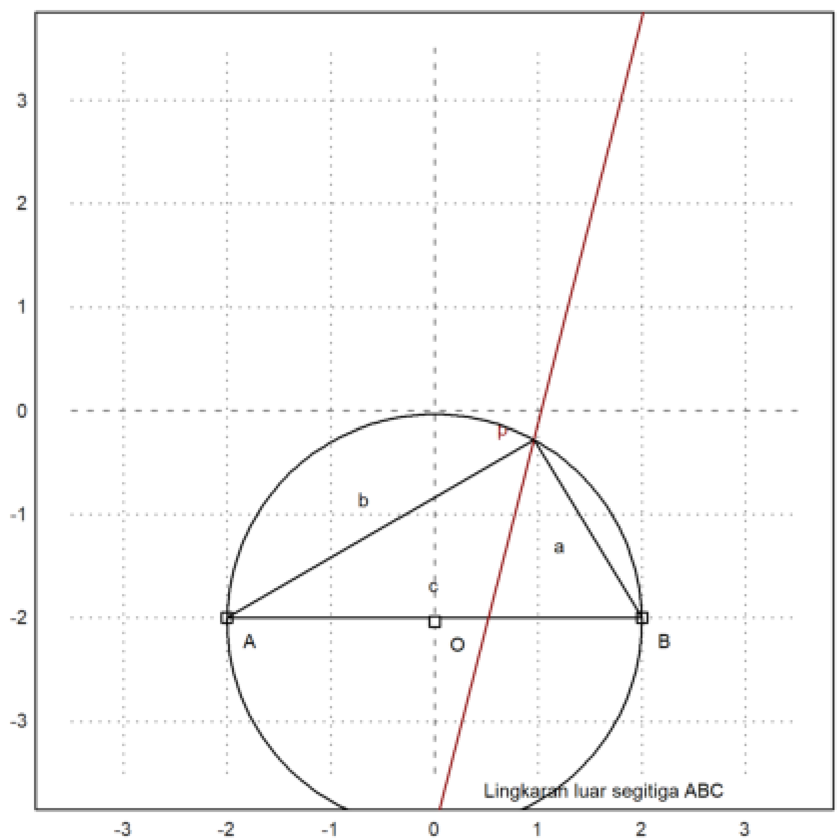

Figure 26: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-134.png

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

Petunjuk:

- Misalkan persamaan parabolanya y= ax^2+bx+c.
- Substitusikan koordinat titik-titik yang diketahui ke persamaan tersebut.

- Selesaikan SPL yang terbentuk untuk mendapatkan nilai-nilai a, b, c.

>load geometry;

>setPlotRange(5); P=[2,0]; Q=[4,0]; R=[0,-4];

>plotPoint(P,“P”); plotPoint(Q,“Q”); plotPoint(R,“R”):

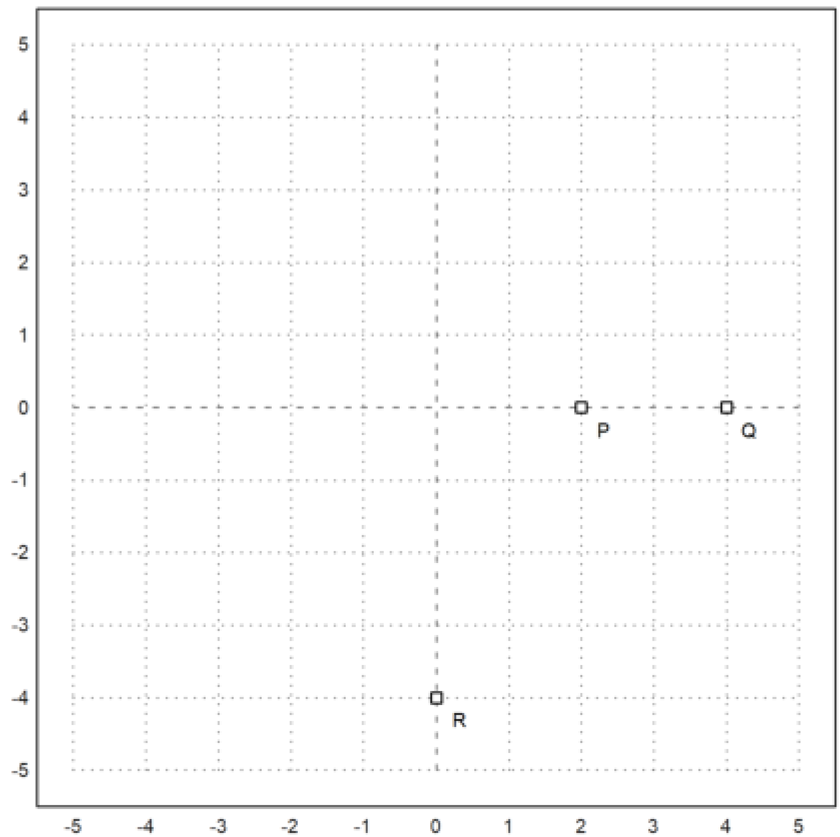

Figure 27: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-135.png

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

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

3. Gambarlah suatu segi-4 yang diketahui keempat titik sudutnya, misalnya A, B, C, D.
   - Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung (sisinya-sisintya merupakan garis singgung lingkaran yang sama yakni

lingkaran dalam segi-4 tersebut).

- Suatu segi-4 merupakan segi-4 garis singgung apabila keempat garis bagi sudutnya bertemu di satu titik.
- Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya.
- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali panjang sisi-sisi yang berhadapan sama.

>load geometry

```
Numerical and symbolic geometry.
```

>setPlotRange(-4.5,4.5,-4.5,4.5);

>A=[-3,-3]; plotPoint(A,“A”);

>B=[3,-3]; plotPoint(B,“B”);

>C=[3,3]; plotPoint(C,“C”);

>D=[-3,3]; plotPoint(D,“D”);

>plotSegment(A,B,“ “);

>plotSegment(B,C,“ “);

>plotSegment(C,D,“ “);

>plotSegment(A,D,“ “);

>aspect(1):

>l=angleBisector(A,B,C);

>m=angleBisector(B,C,D);

>P=lineIntersection(l,m);

>color(5); plotLine(l); plotLine(m); color(1);

>plotPoint(P,“P”):

Dari gambar diatas terlihat bahwa keempat garis bagi sudutnya bertemu di satu titik yaitu titik P.

>r=norm(P-projectToLine(P,lineThrough(A,B)));

>plotCircle(circleWithCenter(P,r),“Lingkaran dalam segiempat ABCD”):

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

```
6
```

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

```
6
```

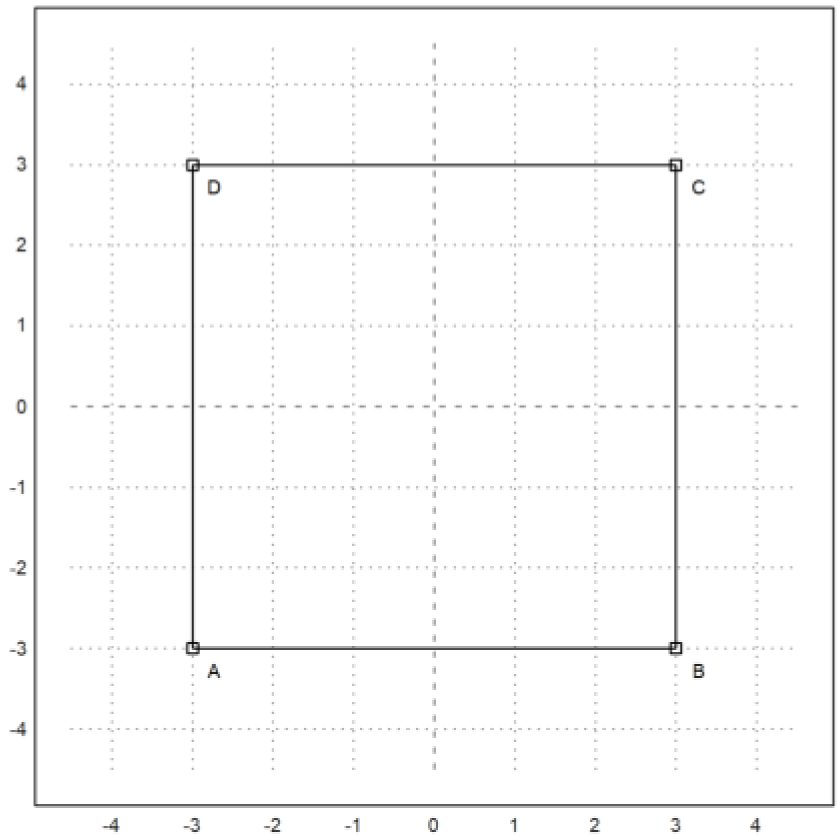

Figure 28: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-136.png

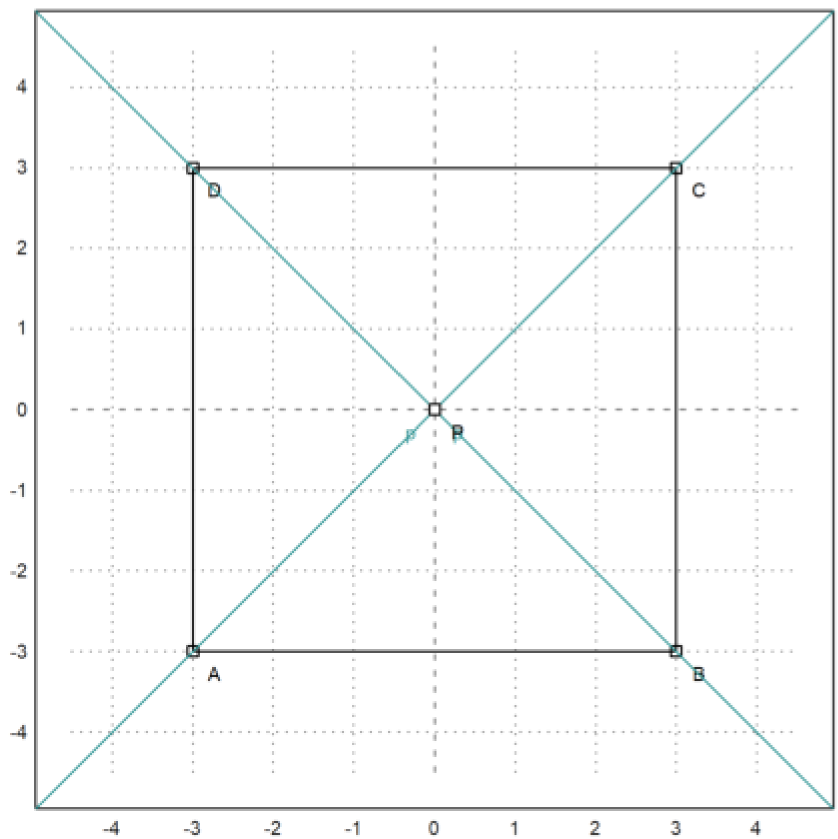

Figure 29: images/EMT4Geometry__23030630086__Annisa%20Nur%20Isnaini-137.png

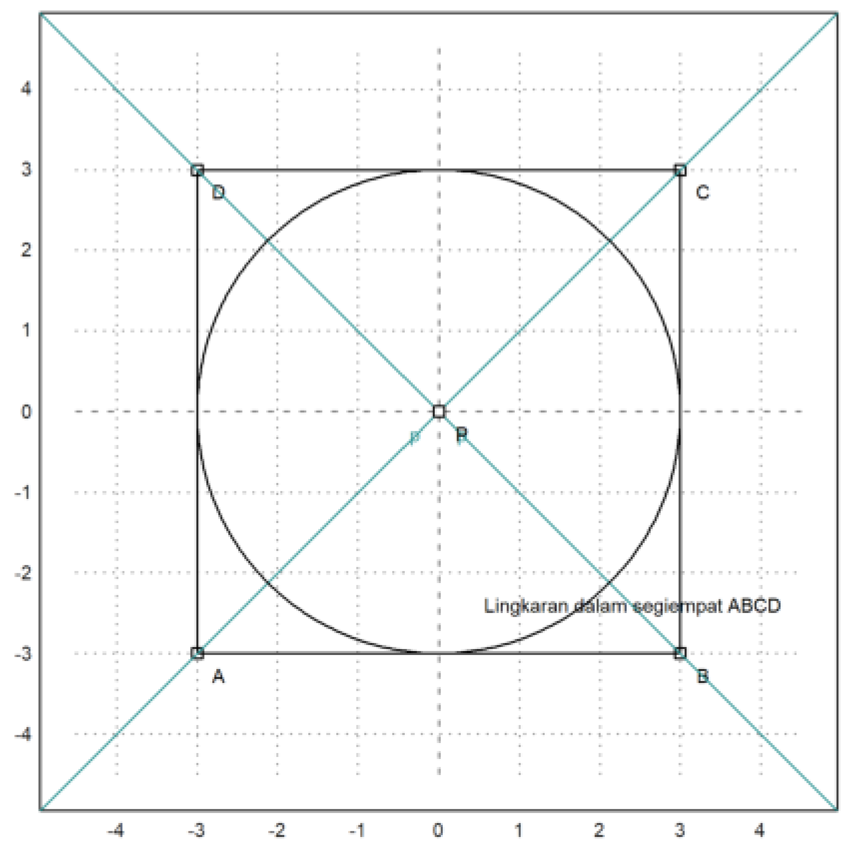

Figure 30: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-138.png

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

6

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

6

>AB.CD

36

>AD.BC

36

Maka terbukti bahwa hasil kali panjang sisi-sisi yang berhadapan sama yaitu 36. Jadi dapat dipastikan bahwa segiempat tersebut merupakan segiempat garis singgung.

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

Penyelesaian :

Diketahui kedua titik fokus P = [-1,-1] dan Q = [1,-1]

>

>P=[-1,-1]; Q=[1,-1];

>function d1(x,y):=sqrt((x-P[1])$^{2+(y-P[2])}$2)

>Q=[1,-1]; function d2(x,y):=sqrt((x-P[1])$^{2+(y-P[2])}$2)+sqrt((x-Q[1])$^{2+(y-Q[2])}$2)

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

Grafik yang lebih menarik

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

Batasan ke garis PQ

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

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

>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):

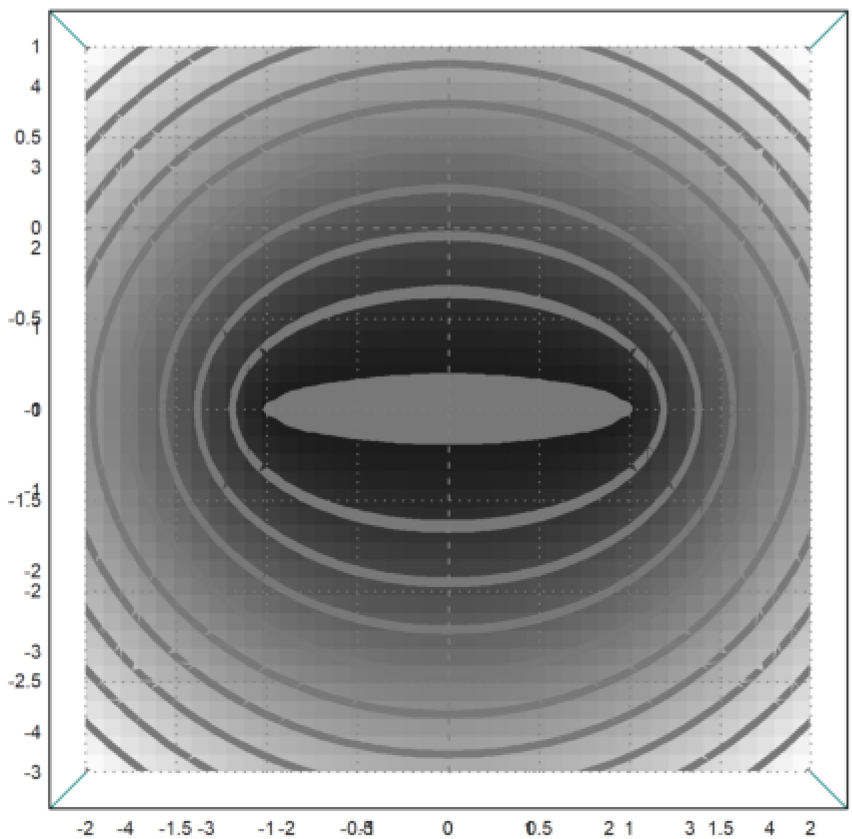

Figure 31: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-139.png

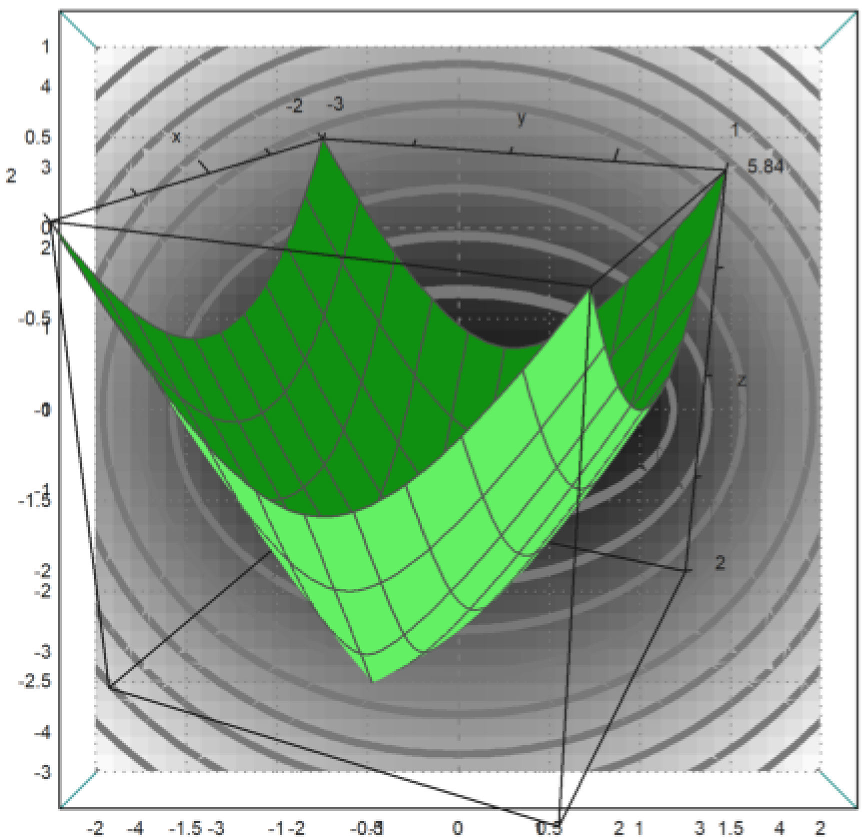

Figure 32: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-140.png

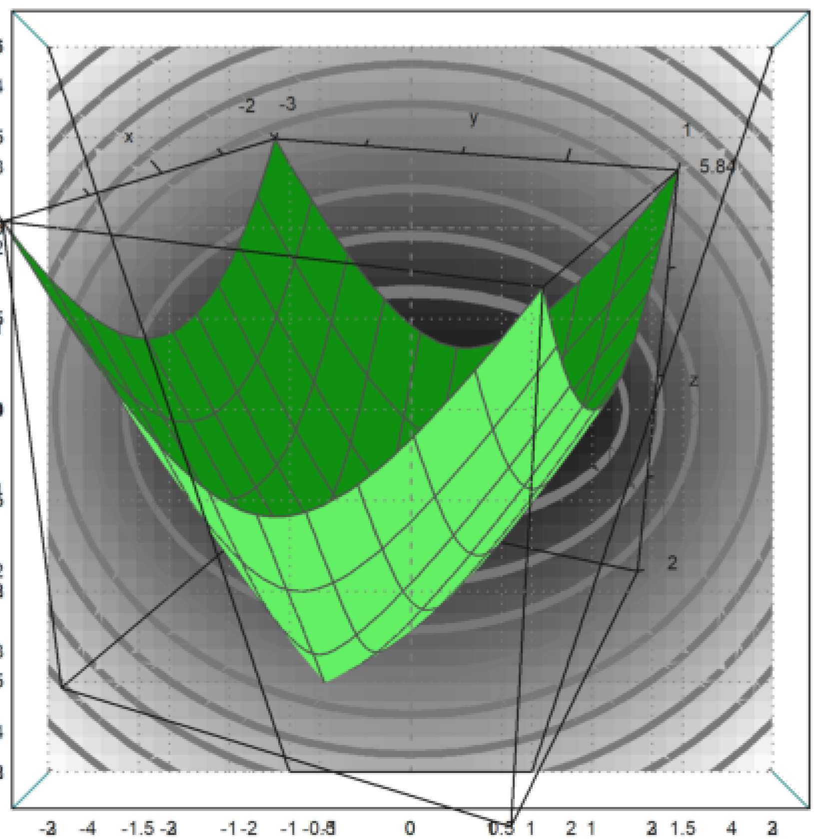

Figure 33: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-141.png

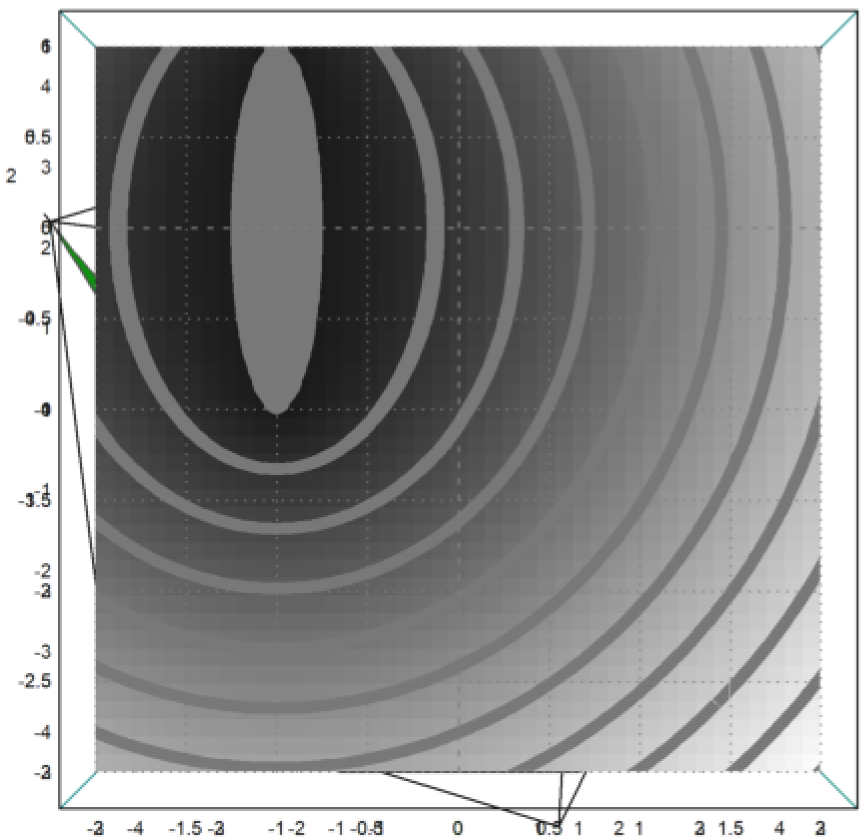

Figure 34: images/EMT4Geometry__23030630086__Annisa%20Nur%20Isnaini-142.png

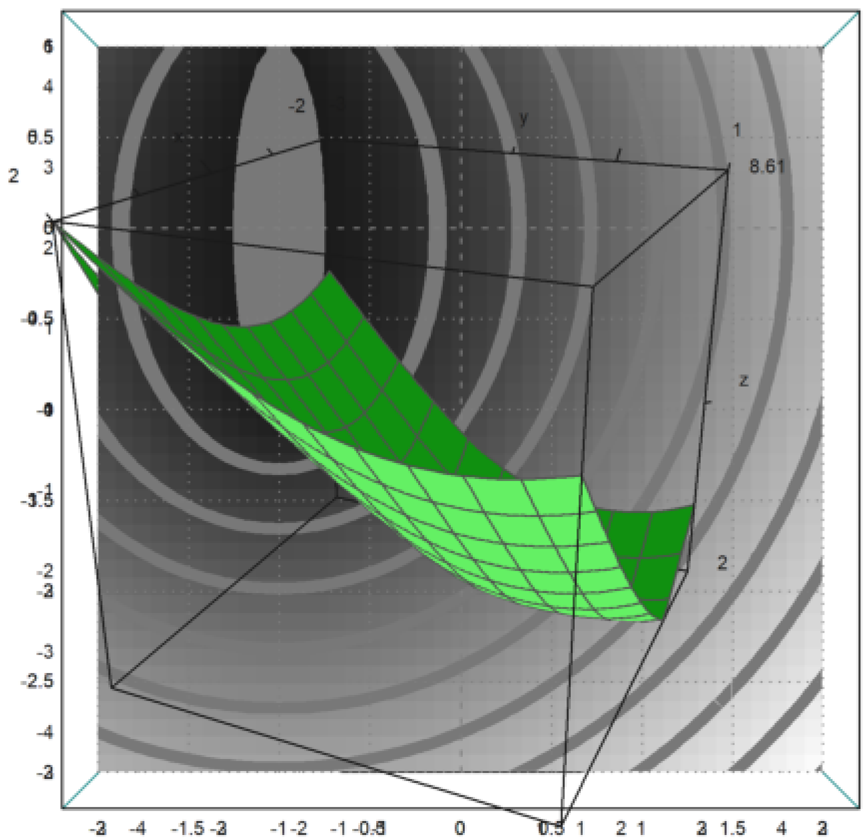

Figure 35: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-143.png

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

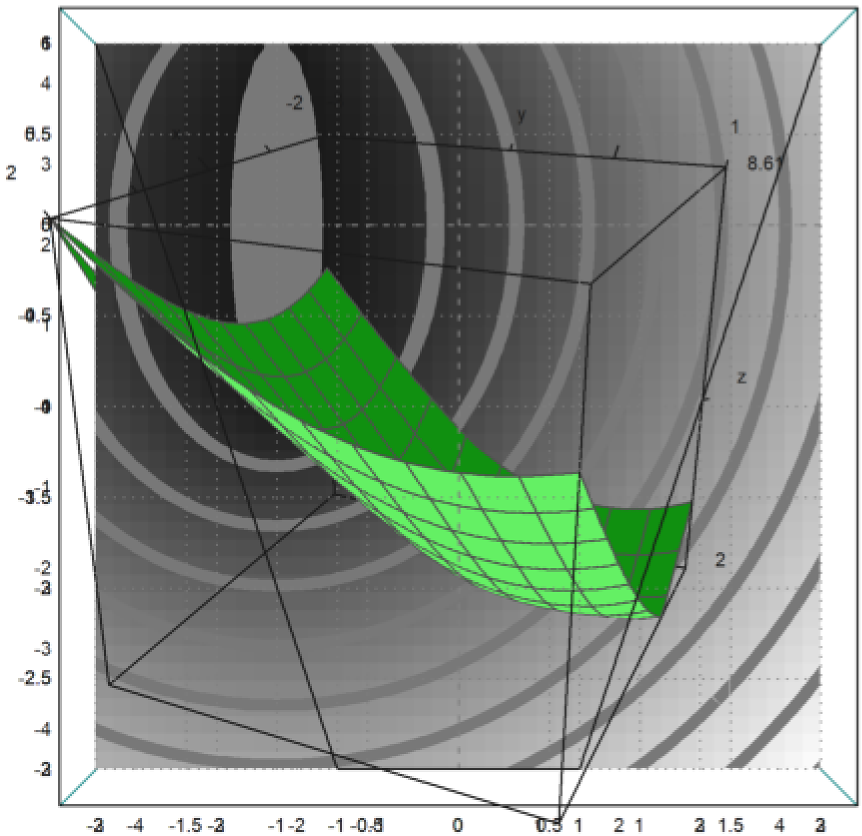

Figure 36: images/EMT4Geometry_23030630086_Annisa%20Nur%20Isnaini-144.png