

Najwa Ayu Lisa Nabila

* 24030130026

Visualisasi dan Perhitungan Geometri dengan EMT

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

```
>load geometry
```

```
incorrect syntax: geometry is not an infix operator
load geometry;
^
```

Fungsi-fungsi Geometri

Fungsi-fungsi untuk Menggambar Objek Geometri:

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

Fungsi-fungsi Geometri Analitik (numerik maupun simbolik):

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

`circleThrough(A,B,C)`: lingkaran melalui A, B, C
`middlePerpendicular(A, B)`: titik tengah AB
`lineCircleIntersections(g, c)`: titik potong garis g dan lingkaran c
`circleCircleIntersections(c1, c2)`: titik potong lingkaran c1 dan c2
`planeThrough(A, B, C)`: bidang melalui titik A, B, C

Fungsi-fungsi Khusus Untuk Geometri Simbolik:

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

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

Kemudian tiga segmen.

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

Fungsi geometri merangkumi fungsi untuk membuat garis dan bulatan. Format untuk garis adalah [a,b,c], yang mewakili garis dengan persamaan $ax+by=c$.

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

```
lineThrough(B, C)
```

Hitung garis tegak lurus Melalui A pada BC.

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

Dan persimpangan dengan BC.

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

Plot itu.

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

```
incorrect syntax: Premature termination of input at ;.  
plotSegment(A,D):;  
^
```

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)
```

```
(sqrt((B - C) . (transpose(B) - transpose(C)))  
sqrt((A - D) . (transpose(A) - transpose(D))))/2
```

Compare with determinant formula.

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

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

Cara lain menghitung luas segitiga ABC:

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

$$\frac{\text{distance}(A, D) \text{ distance}(B, C)}{2}$$

The angle at C.

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

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


Now the circumcircle of the triangle.

```
>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"):
```

```
incorrect syntax: Premature termination of input at ;.
uar segitiga ABC");;
^
```

Tampilkan koordinat titik pusat dan jari-jari lingkaran luar.

```
>O, R
```

O

R

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
```

```
P = lineIntersection(l, g)
```

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
```

```
    r = sqrt((P - projectToLine(P, lineThrough(A, B)))  
    . (transpose(P) - transpose(projectToLine(P, lineThrough(A, B)))))
```

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

```
incorrect syntax: Premature termination of input at ;.  
lam segitiga ABC");;  
^
```

Latihan

1. Tentukan ketiga titik singgung lingkaran dalam dengan sisi-sisi segitiga ABC.

```
>setPlotRange(1,10,1,10); //(batas sumbu x, batas sumbu y)  
>A=[-3,-1/2]; plotPoint(A,"A"); // definisi dan gambar titik  
>B=[1/2,3]; plotPoint(B,"B");  
>C=[3/2,-5/2]; plotPoint(C,"C");  
>plotSegment(A,B,"c"); // c=AB  
>plotSegment(B,C,"a"); // a=BC  
>plotSegment(C,A,"b"); // b=AC
```

2. Gambar segitiga dengan titik-titik sudut ketiga titik singgung tersebut. Merupakan segitiga apakah itu?

```
>h=perpendicular(A,lineThrough(B,C)); // garis h tegak lurus BC melalui A
>D=lineIntersection(h,lineThrough(B,C)); // D adalah titik potong h dan BC
>plotPoint(D,value=1); // koordinat D ditampilkan
>aspect(1); plotSegment(A,D,"h"): // tampilkan semua gambar hasil plot...()
```

```
incorrect syntax: Premature termination of input at ;.
otSegment(A,D,"h");;
      ^
```

3. Hitung luas segitiga tersebut.

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

4. Tunjukkan bahwa garis bagi sudut yang ke tiga juga melalui titik pusat lingkaran dalam.

```
>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  
>plotPoint(P,"P"); // gambar titik potongnya  
>color(5); plotLine(l); plotLine(g); color(1);
```

5. Gambar jari-jari lingkaran dalam.

```
>r=norm(P-projectToLine(P,lineThrough(A,B))) // jari-jari lingkaran dalam  
>plotCircle(circleWithCenter(P,r),"Lingkaran dalam segitiga ABC");
```

6. Hitung luas lingkaran luar dan luas lingkaran dalam segitiga ABC. Adakah hubungan antara luas kedua lingkaran tersebut dengan luas segitiga ABC?

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

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

Fungsi untuk garis dan bulatan berfungsi seperti fungsi Euler, tetapi menyediakan pengiraan simbolik.

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

```
incorrect syntax: & is not an infix operator  
c &=  
^
```

Kita boleh mendapatkan persamaan untuk garis dengan mudah.

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

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

```
[getLineEquation(c, x, y) = 0]
```

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

```
getLineEquation(lineThrough([x1, y1], [x2, y2]), x, y)
```

```
[getLineEquation(lineThrough([x1, y1], [x2, y2]), x, y) = 0]
```

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

```
getLineEquation(lineThrough(A, [x1, y1]), x, y)
```

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

```
incorrect syntax: & is not an infix operator
```

```
h &=
```

```
^
```

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

```
incorrect syntax: & is not an infix operator  
Q &=  
^
```

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

```
projectToLine(A, lineThrough(B, C))
```

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

```
distance(A, Q)
```

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


cc

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

r

r

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

computeAngle(A, C, B)

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

getLineEquation(angleBisector(A, C, B), x, y) = 0

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

P

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

P()

Garis dan bulatan yang bersilang

Sudah tentu, kita juga boleh memotong garis dengan bulatan, dan bulatan dengan bulatan.

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

Persimpangan garis dengan bulatan mengembalikan dua titik dan bilangan titik persimpangan.

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

```
incorrect syntax: Missing }  
{P1;  
  ^
```

P2

```
>P1, P2, f
```

P1

P2

f

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

```
incorrect syntax: Premature termination of input at ;.  
plotPoint(P2);;  
^
```

Begitu juga dengan Maxima.

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

```
incorrect syntax: & is not an infix operator  
c &=  
^
```

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

```
incorrect syntax: & is not an infix operator  
l &=  
^
```

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

```
lineCircleIntersections(l, c)
```

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''

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

69°17'42.68''

```
>insimg;
```

Garis Sumbu

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

1. Gambar lingkaran dengan pusat A melalui B.
2. Gambar lingkaran dengan pusat B melalui A.
3. Tarik garis melalui 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):
```

Seterusnya, kami melakukan perkara yang sama dalam Maxima dengan koordinat umum.

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

Persamaan untuk persimpangan agak terlibat. Tetapi kita boleh mempermudah, jika kita menyelesaikan untuk y.

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

Ini memang sama dengan tegak lurus tengah, yang dikira dengan cara yang sama sekali berbeda.

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

Perhatikan hasil kali gradien garis g dan h adalah:

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

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])
```

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

Ekstrak solusinya y.

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

Kita dapat Formula Heron.

```
>function H(a,b,c) &= sqrt(factor((ysol*a/2)^2)); $'H(a,b,c)=H(a,b,c)
>$'Luas=H(2,5,6) // luas segitiga dengan panjang sisi-sisi 2, 5, 6
```

Tentu saja, setiap segitiga persegi panjang adalah kasus yang terkenal.

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

6

Dan juga jelas, bahwa ini adalah segitiga dengan luas maksimal dan kedua 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)
```

Kasus umum juga berhasil.

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

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

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

Dan membuat 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)
```

Sekarang kita bisa menggambar set. 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.

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

dimana (x_m, y_m) adalah pusatnya, dan u dan v adalah setengah sumbunya.

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

Kita melihat bahwa tinggi dan luas segitiga adalah maksimal untuk $x=0$. Jadi luas segitiga dengan $a+b+c=d$ adalah maksimal, jika sama sisi. Kami ingin mendapatkan ini secara analitis.

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

Kami mendapatkan beberapa minima, yang termasuk dalam segitiga dengan satu sisi 0, dan solusi $a=b=c=d/3$.

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

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

```
>&solve([diff(H(a,b,c)^2,a)=1a,diff(H(a,b,c)^2,b)=1a, ...
> diff(H(a,b,c)^2,c)=1a,a+b+c=d],[a,b,c,1a])
```

$$[[a = 0, b = \frac{d}{2}, c = \frac{d}{2}, 1a = 0], [a = \frac{d}{2}, b = 0, c = \frac{d}{2}, 1a = 0],$$

$$[a = \frac{d}{2}, b = \frac{d}{2}, c = 0, 1a = 0], [a = \frac{d}{3}, b = \frac{d}{3}, c = \frac{d}{3}, 1a = \frac{d}{108}]]$$

Kita dapat membuat plot dari situasi tersebut

Pertama-tama atur poin dalam Maxima.

```
>A &= at([x,y],sol[2]); $A
>B &= [0,0]; $B, C &= [a,0]; $C
```

Kemudian atur rentang plot, dan plot poinnya.

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

Plot segmennya.

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

Hitung tegak lurus tengah secara Maksimal.

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

Dan pusat kelilingnya.

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

Kami mendapatkan rumus untuk jari-jari lingkaran.

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

Mari kita tambahkan ini ke plot.

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

Dengan menggunakan geometri, kami menurunkan rumus sederhana

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

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

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

Contoh 4: Garis Euler dan Parabola

Garis euler adalah garis yang ditentukan dari setiap segitiga yang tidak sama sisi. Ini adalah garis tengah segitiga, dan melewati beberapa titik penting yang ditentukan dari segitiga, termasuk orthocenter, circumcenter, centroid, titik Exeter, dan pusat lingkaran sembilan titik segitiga.

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

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

```
>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 menjumlahkan sisi-sisi segitiga.

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


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

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

Kita dapat menghitung koefisien sisi c.

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

$[-1, 3, -2]$

Dan juga dapatkan rumus untuk baris ini.

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

Untuk bentuk Hesse, kita perlu menentukan suatu titik, sehingga titik tersebut berada di sisi positif dari Bentuk Hess. Memasukkan titik menghasilkan jarak positif ke garis.

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

Sekarang kita menghitung lingkaran ABC.

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

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

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

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

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

Sekarang kita dapat menghitung garis Euler dari segitiga tersebut.

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

Menambahkannya ke plot.

```
>plotPoint(H(),"H"); plotLine(e1(),"Garis Euler"):
```

Pusat gravitasi harus berada pada garis ini.

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

Teori tersebut memberi tahu kita $MH=2*MO$. Kita perlu menyederhanakan dengan radcan untuk mencapai ini.

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

Fungsinya termasuk fungsi untuk sudut juga.

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

60°15'18.43''

Persamaan untuk pusat lingkaran tidak terlalu bagus.

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

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

```
>r &= distance(Q,projectToLine(Q,lineThrough(A,B)))|ratsimp; $r  
>LD &= circleWithCenter(Q,r); // Lingkaran dalam
```

Mari kita tambahkan ini ke plot.

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

Parabola

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

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

Persamaan tersebut dapat digambar menjadi satu dengan gambar sebelumnya.

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

Ini seharusnya beberapa fungsi, tetapi pemecah default Maxima hanya dapat menemukan solusinya, jika kita kuadratkan persamaannya. Akibatnya, kami mendapatkan solusi palsu.

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

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

Solusi pertama adalah

$$akar_1$$

Menambahkan solusi pertama ke plot menunjukkan, bahwa itu memang jalan yang kita cari. Teori tersebut memberi tahu kita bahwa itu adalah parabola yang diputar.

```
>plot2d(&rhs(akar[1]),add=1):  
>function g(x) &= rhs(akar[1]); $'g(x)= g(x)// fungsi yang mendefinisikan kurva di atas  
>T &=[-1, g(-1)]; // ambil sebarang titik pada kurva tersebut  
>dTC &= distance(T,C); $fullratsimp(dTC), $float(%) // jarak T ke C  
>U &= projectToLine(T,lineThrough(A,B)); $U // proyeksi T pada garis AB  
>dU2AB &= distance(T,U); $fullratsimp(dU2AB), $float(%) // jarak T ke AB
```

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 "Proporsi Ilahi", Wildberger mengusulkan untuk menggantikan pengertian klasik tentang jarak dan sudut dengan segi empat dan penyebaran. Dengan menggunakan ini, memang mungkin untuk menghindari fungsi trigonometri dalam banyak contoh, dan tetap "rasional".

Berikut ini, saya memperkenalkan konsepnya, dan memecahkan beberapa masalah. Saya menggunakan perhitungan simbolik Maxima di sini, yang menyembunyikan keuntungan utama trigonometri rasional bahwa perhitungan hanya dapat dilakukan dengan kertas dan pensil. Anda diundang untuk memeriksa hasilnya tanpa komputer.

Intinya adalah bahwa perhitungan rasional simbolik sering kali menghasilkan hasil yang sederhana. Sebaliknya, trigonometri klasik menghasilkan hasil trigonometri yang rumit, yang hanya mengevaluasi perkiraan numerik.

```
>load geometry;
```

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

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

dimana w_a adalah sudut pada A. Cara yang biasa untuk menghitung sudut ini, adalah dengan mengambil invers dari fungsi sinus. Hasilnya adalah sudut yang tidak dapat dicerna, yang hanya dapat dicetak kira-kira.

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

36°52'11.63''

Trigonometri rasional mencoba menghindari hal ini.

Gagasan pertama trigonometri rasional adalah segi empat, yang menggantikan jarak. Faktanya, itu hanya jarak yang dikuadratkan. Berikut ini, a, b, dan c menunjukkan segi empat sisi-sisinya.

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

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


Pengertian kedua dari trigonometri rasional adalah penyebarannya. Spread mengukur pembukaan antar garis. Ini adalah 0, jika garisnya sejajar, dan 1, jika 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 segi empat dari setiap segitiga persegi panjang dengan salah satu sudutnya adalah A.

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

Ini lebih mudah dihitung daripada sudutnya, tentu saja. Tetapi Anda kehilangan properti bahwa sudut dapat ditambahkan dengan mudah.

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

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

9/25

Hukum kosinus trigonometri klasik diterjemahkan menjadi "hukum silang" berikut.

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

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

Hukum-hukum ini diterapkan dalam geometri.e file yang kami muat ke Euler.

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

Di kasus kita, kita dapat

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

Mari kita gunakan hukum silang ini untuk menemukan sebaran pada A . Untuk melakukan ini, kita menghasilkan hukum silang untuk segi empat a , b , dan c , dan menyelesaikannya untuk sebaran sa yang tidak diketahui.

Anda dapat melakukannya dengan tangan dengan mudah, tetapi saya menggunakan Maxima. Tentu saja, kami mendapatkan hasilnya, kami sudah memilikinya.

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

Kita sudah tahu ini. Definisi spread adalah kasus khusus dari crosslaw.

Kita juga dapat menyelesaikan ini untuk umum a , b , c . Hasilnya adalah rumus yang menghitung luas suatu sudut segitiga dengan kuadrat dari ketiga sisinya.

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

Kita bisa membuat fungsi dari hasilnya. Fungsi seperti itu sudah didefinisikan dalam geometri.e file dari Euler.

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

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

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

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

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

Ini adalah sudut dalam derajat.

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

67°47'32.44''

Contoh lain

Sekarang, mari kita coba contoh yang lebih maju.

Kami menetapkan tiga sudut segitiga sebagai berikut.

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

Menggunakan Pythagoras, mudah untuk menghitung jarak antara dua titik. Saya pertama kali menggunakan jarak fungsi dari file Euler untuk geometri. Jarak fungsi menggunakan geometri klasik.

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

Euler juga mengandung fungsi untuk segi empat antara dua titik.

Dalam contoh berikut, karena $c+b$ bukan a , segitiga tidak berbentuk persegi panjang.

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

Pertama, mari kita hitung sudut tradisionalnya. Fungsi `computeAngle` menggunakan metode biasa berdasarkan perkalian titik dari dua vektor. Hasilnya adalah beberapa pendekatan floating point.

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

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

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

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

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

32.4711922908

Dengan menggunakan pensil dan kertas, kita dapat melakukan hal yang sama dengan hukum salib. Kami memasukkan segi empat a , b , dan c ke dalam hukum silang dan menyelesaikannya untuk x .

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

Artinya, apa yang didefinisikan oleh penyebaran fungsi dalam ” geometri.e ” tidak.

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

Maxima mendapatkan hasil yang sama dengan menggunakan trigonometri biasa, jika kita memaksakan-nya. Itu menyelesaikan dosa (arccos (...)) istilah untuk hasil pecahan. Sebagian besar siswa tidak dapat melakukan ini.

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

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

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

menurut definisi.

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

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

gambar: (20) Rational_Geometry_CaR.png

Menurut definisi, panjang h_a adalah akar kuadrat dari kuadratnya.

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

Sekarang kita dapat menghitung luas segitiga. Jangan lupa, bahwa kita sedang berhadapan dengan segi empat!

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

Rumus determinan biasa menghasilkan hasil yang sama.

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

Rumus Bangau

Sekarang, mari kita selesaikan masalah ini secara umum!

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

Pertama-tama kita menghitung sebaran di B untuk sebuah segitiga dengan sisi a, b, dan c. Kemudian kita menghitung luas kuadrat ("kuadrea"?), faktorkan dengan Maxima, dan kita mendapatkan rumus Bangau yang terkenal.

Memang, ini sulit dilakukan dengan pensil dan kertas.

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

Aturan Penyebaran Tiga Kali Lipat

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

Namun, tiga penyebaran segitiga memenuhi aturan "penyebaran tiga kali lipat" berikut.

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


Aturan ini berlaku untuk tiga sudut yang ditambahkan ke 180° .

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

Karena penyebaran

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

sama, aturan penyebaran tiga kali lipat juga benar, jika

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

Karena penyebaran sudut negatifnya sama, aturan penyebaran rangkap tiga juga berlaku, jika

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

Misalnya, kita dapat menghitung penyebaran sudut 60° Ini adalah $3/4$. Persamaan memiliki solusi kedua, bagaimanapun, di mana semua spread adalah 0.

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

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

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

Sejak penyebaran 180° -t adalah sama seperti penyebaran t, triple menyebarkan formula juga memegang, jika salah satu sudut adalah jumlah atau selisih dua sudut lainnya.

Sehingga kita dapat menemukan penyebaran dari dua kali lipat sudut. Perhatikan bahwa ada dua solusi lagi. Kami menjadikan ini sebuah fungsi.

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

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

Garis Bagi Sudut

Ini situasinya, kita sudah tahu.

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

Garis Bagi Sudut

Ini situasinya, kita sudah tahu.

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

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

Masalah dengan rumus ini muncul lagi. Ini memiliki dua solusi. Kita harus memilih yang benar. Solusi lainnya mengacu pada sudut yang dibelah dua 180°

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

Mari kita periksa persegi panjang Mesir.

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

Kita dapat mencetak sudut dalam Euler, setelah mentransfer penyebarannya 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)
```

```
0.321750554397
```

```
0.321750554397
```

Sekarang kita menghitung panjang garis bagi AP.

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

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

tahan dalam segitiga apa pun. Kuadratkan, itu diterjemahkan menjadi apa yang disebut "hukum penyebaran"

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

dimana a,b, c menunjukkan qudrances.

Karena spread CPA adalah $1-sa^2$, kita dapatkan darinya $b^2 = 1 - (1-sa^2)$ dan dapat menghitung bisa (segiempat dari garis bagi sudut).

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

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

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

4.21637021356

4.21637021356

Kita juga dapat menghitung P menggunakan rumus sebaran.

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

Nilainya sama dengan yang kita dapatkan dengan rumus trigonometri.

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

1.33333333333

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,"O"); ...
>plotSegment(A,O); plotSegment(B,O); plotSegment(C,O,"r"); ...
>insimg;
```

Kita dapat menggunakan Maxima untuk menyelesaikan rumus penyebaran rangkap tiga untuk sudut-sudut di pusat O untuk r. Jadi kita mendapatkan rumus jari-jari kuadrat lingkaran dalam segi empat sisi-sisinya.

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

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

Kita bisa menjadikannya fungsi Euler.

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

Mari kita periksa hasilnya untuk poin kita 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
```

Sebenarnya penyebarannya adalah $b / (4r)$, dan kita melihat bahwa sudut akor dari akor b adalah setengah dari sudut tengah.

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


Contoh 6: Jangkauan Minimal pada Tawaran

Komentar awal

Fungsi yang, ke titik M di bidang, menetapkan jarak AM antara titik tetap A dan M, memiliki garis datar yang agak 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 agak sederhana: bagian atas kerucut:

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

Tentu saja minimum 0 dicapai dalam A.

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

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

Grafiknya lebih menarik:

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

Pembatasan garis (AB) lebih terkenal:

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

Sekarang semuanya menjadi kurang sederhana: Sedikit kurang diketahui bahwa $MA + MB + MC$ mencapai minimum pada satu titik bidang tetapi untuk menentukannya kurang sederhana:

1) Jika salah satu sudut segitiga ABC lebih dari 120°

Contoh:

```
>C=[-4,1];
>function d3(x,y):=d2(x,y)+sqrt((x-C[1])^2+(y-C[2])^2)
>plot3d("d3",xmin=-5,xmax=3,ymin=-4,ymax=4);
>insimg;
>fcontour("d3",xmin=-4,xmax=1,ymin=-2,ymax=2,hue=1,title="The minimum is on A");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

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

```
>C=[-0.5,1];
>plot3d("d3",xmin=-2,xmax=2,ymin=-2,ymax=2):
>fcontour("d3",xmin=-2,xmax=2,ymin=-2,ymax=2,hue=1,title="The Fermat point");
>P=(A_B_C_A)'; plot2d(P[1],P[2],add=1,color=12);
>insimg;
```

Merupakan kegiatan yang menarik untuk mewujudkan gambar di atas dengan perangkat lunak geometri; misalnya, saya tahu soft tertulis di Java yang memiliki instruksi "garis kontur"...

Semua hal di atas telah ditemukan oleh seorang hakim Prancis bernama Pierre de Fermat; dia menulis surat kepada dilettant lain seperti pendeta Marin Mersenne dan Blaise Pascal yang bekerja di bidang pajak penghasilan. Jadi titik unik F sedemikian rupa sehingga $FA + FB + FC$ minimal, disebut Titik Fermat segitiga. Tetapi tampaknya beberapa tahun sebelumnya, Torricelli dari Italia telah menemukan hal ini sebelum Fermat melakukannya! Pokoknya tradisinya adalah mencatat poin ini F...

Empat poin

Langkah selanjutnya adalah menambahkan titik ke-4 D dan mencoba meminimalkan $MA + MB + MC + MD$; katakan bahwa Anda adalah operator TV kabel dan ingin mengetahui di bidang mana Anda harus meletakkan antena sehingga Anda dapat memberi makan empat desa dan menggunakan 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 pada simpul A, B, C atau D:

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

[0.142858, 0.142857]

Tampaknya dalam hal ini, koordinat titik optimal adalah rasional atau mendekati rasional...
Sekarang ABCD adalah persegi kita berharap 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;
```

Contoh 7: Bola Dandelin dengan Povray

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

Pertama-tama kita menghitung jari-jari bola.

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

Kami menggunakan geometri.e file Euler untuk ini.

```
>load geometry;
```

Pertama dua garis yang membentuk kerucut.

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

$$[-a, 1, 0]$$

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

$[-a, -1, 0]$

Kemudian baris ketiga.

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

$[-1, 2, 1]$

Kami merencanakan semuanya sejauh ini.

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

Sekarang kita mengambil titik umum pada sumbu y.

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

$[0, u]$

Hitung jarak ke g1.

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

Hitung jarak ke g.

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

Dan menemukan pusat dari dua lingkaran, dimana jarak yang sama.

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


Ada dua solusi.

Kami mengevaluasi solusi simbolik, dan menemukan kedua pusat, dan kedua jarak.

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

```
[0.333333, 1]
```

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

```
[0.149071, 0.447214]
```

Plot lingkaran menjadi gambar.

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

Plot dengan Povray

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

Pertama kita memuat fungsi povray.

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

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

Kami mengatur adegan dengan tepat.

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

Selanjutnya kita menulis dua spheres ke file Povray.

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

Dan kerucutnya, transparan.

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

Kami menghasilkan bidang yang terbatas pada kerucut.

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

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

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

Kemudian kita menghasilkan dua titik di mana bola menyentuh bidang. Ini adalah fokus dari elips.

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

Selanjutnya kita menghitung persimpangan P1P2 dengan pesawat.

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

Kami menghubungkan titik-titik dengan segmen garis.

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

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

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

Mulai program Povray.

```
>povend();
```

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

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

```
global a,u,dd,g,g1,defaultpointsize;  
writeln(povsphere([0,0,u[1]],dd[1],povlook(red)));  
writeln(povsphere([0,0,u[2]],dd[2],povlook(red)));  
writeln(povcone([0,0,0],0,[0,0,a],1,povlook(lightgray,1)));  
gp=g();  
pc=povcone([0,0,0],0,[0,0,a],1,"");  
vp=[gp[1],0,gp[2]]; dp=gp[3];  
writeln(povplane(vp,dp,povlook(blue,0.5),pc));
```

```

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

```

Anda membutuhkan kacamata merah / cyan untuk menghargai efek berikut.

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

Contoh 8: Geometri Bumi

Dalam buku catatan ini, kami ingin melakukan beberapa perhitungan bola. Fungsi-fungsi tersebut terdapat dalam file "spherical.e" di folder contoh. Kita perlu memuat file itu terlebih dahulu.

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

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

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

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

Anda dapat mencetak posisi ini dengan `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-tama kita menghitung vektor dari satu ke yang lain pada bola yang ideal. Vektor ini adalah [heading, distance] dalam radian. Untuk menghitung jarak di bumi, kita kalikan dengan jari-jari bumi pada garis lintang 7°

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

65°20'26.60''
53.8945384608

Ini adalah perkiraan yang bagus. Rutinitas berikut menggunakan perkiraan yang lebih baik. Pada jarak yang begitu dekat hasilnya hampir sama.

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

88.0114026318 km

Ada fungsi untuk heading, mengambil bentuk elips dari bumi ke rekening. Sekali lagi, kami mencetak dengan cara yang canggih.

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

65.34°

Sudut segitiga melebihi 180°

```
>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 pada asum-pi.

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

2123.67056312 km^2

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

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

```
2123.64310526 km^2
```

Kita juga dapat menambahkan vektor ke posisi. Sebuah vektor berisi heading 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 yang ideal. Hal yang sama di bumi.

```
>sposprint(esadd(FMIPA,esdir(FMIPA,Solo),esdist(FMIPA,Solo))), sposprint(Solo),
```

```
S 7°34.333' E 110°49.683'
```

```
S 7°34.333' E 110°49.683'
```

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''
```

Namun, kita tidak lagi mendapatkan posisi target yang tepat, jika kita menambahkan heading dan jarak ke posisi aslinya. Begitulah, karena kita tidak menghitung fungsi kebalikannya dengan tepat, tetapi mengambil perkiraan jari-jari bumi di sepanjang jalurnya.

```
>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 menempuh jalur terpendek. Bayangkan, Anda terbang NE mulai dari titik mana pun di bumi. Kemudian Anda akan berputar ke kutub utara. Lingkaran besar tidak mengikuti arah yang konstan!

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

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

Sekarang kita tambahkan 10 kali sepersepuluh jarak, dengan menggunakan pos menuju Monas, kita sampai di Tugu.

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

Hasilnya adalah jauh.

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

S 6°11.250' E 106°48.372'
1.529km

Sebagai contoh lain, mari kita mengambil dua poin di bumi pada saat yang sama lintang.

```
>P1=[30°,10°]; P2=[30°,50°];
```

Jalur terpendek dari P1 ke P2 bukanlah lingkaran latitude 30°

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

79.69°

Tapi, jika kita mengikuti pembacaan kompas ini, kita akan berputar ke kutub utara! Jadi kita harus menyesuaikan arah kita di sepanjang jalan. Untuk tujuan kasar, kami menyesuaikannya pada 1/10 dari total jarak.

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

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

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

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

```
0.203km
```

Kami mendapatkan perkiraan yang baik, jika kami menyesuaikan heading setelah setiap 1/100 dari total jarak dari Tugu ke Monas.

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

0.000km

Untuk keperluan navigasi, kita bisa mendapatkan urutan posisi GPS di sepanjang great circle menuju Monas dengan fungsi navigate.

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

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

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

```
>function testplot ...  
  
    useglobal;  
    plotearth;  
    plotpos(Tugu,"Tugu Jogja"); plotpos(Monas,"Tugu Monas");  
    plotposline(v);  
endfunction
```

Sekarang plot semuanya.

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

Atau gunakan plot3d untuk mendapatkan tampilan anaglyph. Ini terlihat sangat bagus dengan kacamata merah / cyan.

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


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

Petunjuk:

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

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

Petunjuk:

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

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

- Tentukan apakah segi-4 tersebut merupakan segi-4 garis singgung (sisinya-sisintya

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

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

bertemu di satu titik.

- Jika segi-4 tersebut merupakan segi-4 garis singgung, gambar lingkaran dalamnya.
- Tunjukkan bahwa syarat suatu segi-4 merupakan segi-4 garis singgung apabila hasil kali

panjang sisi-sisi yang berhadapan sama.

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