圆和椭圆的关系-割线

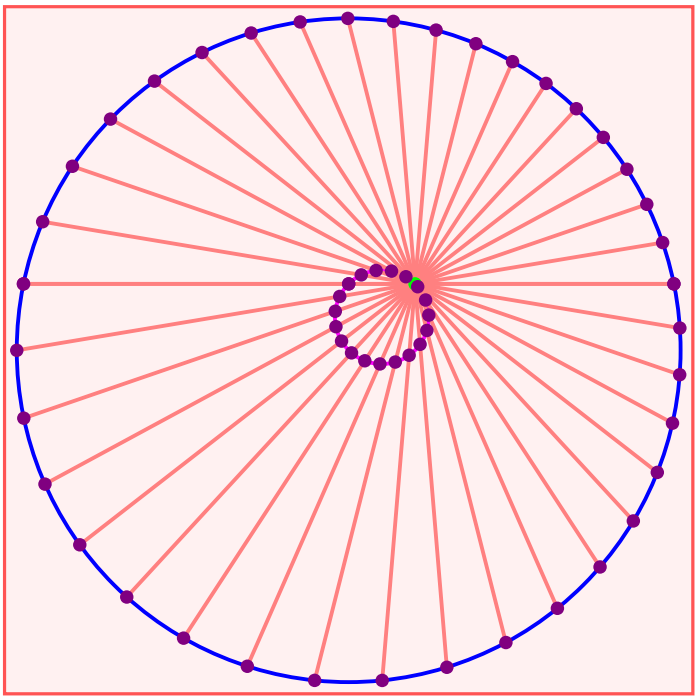
圆内过一点的所有割线的中点，连接起来还是圆

In[ ]:=
```
style = {Axes → True,
          坐标轴  真
   ImageSize → Large,
   图像尺寸      大
   GridLines → Automatic,
   网格线        自动
   GridLinesStyle → Directive[Gray, Dashed]
   网格线样式          指令       灰色   虚线
  };
```

版本一 (性能低)

In[ ]:=
```
DynamicModule[{pt = {1, 1}, r = 5, num = 19},
 LocatorPane[
  Dynamic[pt, (pt = If[#[[1]]^2 + #[[2]]^2 > r^2, r # / EuclideanDistance[#, {0, 0}], #]) &],
  Dynamic[Graphics[
    endpoints = ({x, y} /. NSolve[{#, x^2 + y^2 == r^2}, {x, y}]) & /@
      Table[y == (x - pt[[1]]) Tan[θ] + pt[[2]], {θ, 0, π, π/num}];
    centers = Map[(#1 + #2)/2 & @@ # &, endpoints];
    {Thick, Blue, Circle[{0, 0}, r],
     Magenta, Circumsphere[Take[centers, 3]],
     Pink, Line@endpoints,
     Green, PointSize[Large], Point[pt],
     Purple, Point[Join[Flatten[endpoints, 1], centers]]
    },
    style
   ]],
  Appearance → None
 ]]
```

Out[ ]=



上面的代码有两个问题：

1、因为使用了正切，故θ不能取$\pi/2$，也就是线段数num不能为偶数。
2、性能低，当num=19时，出现明显卡顿

解决问题

1，改用直线的点向式方程
{m,n}是圆内动点的坐标，即定位器位置
{j,k}是直线的方向向量，需要对a=0的情况分别处理
2，通过方程组的符号解，得到所有交点的参数形式

NSolve函数使用第三个参数Real，会有问题，暂时不懂为啥。

In[ ]:=
```
r = 5;
dot = {x, y} /. NSolve[{x^2 + y^2 == r^2, (y - n) j == (x - m) k}, {x, y}]
                      数值求解
dota0 = {x, y} /. NSolve[{x^2 + y^2 == r^2, x == m}, {x, y}]
                        数值求解
{m, n} = {1, 1};
vs = Table[AngleVector[θ], {θ, π/4, π, π/4}]
     表格   角度向量
endpoints = ({j, k} = #; If[j == 0, dota0, dot]) & /@ vs
                          如果
Clear[m, n, j, k, r, x, y]
清除
```

Out[ ]=
$$\left\{\left\{\frac{k^2 m - 1.\, j k n - 1.\, \sqrt{25.\, j^4 + 25.\, j^2 k^2 - 1.\, j^2 k^2 m^2 + 2.\, j^3 k m n - 1.\, j^4 n^2}}{j^2 + k^2},\ \frac{-1.\, k m + \frac{k^3 m}{j^2+k^2} + j n - \frac{1.\, j k^2 n}{j^2+k^2} - \frac{1.\, k \sqrt{-1.\, j^2 \left(-25.\, j^2 - 25.\, k^2 + k^2 m^2 - 2.\, j k m n + j^2 n^2\right)}}{j^2+k^2}}{j}\right\},\right.$$
$$\left.\left\{\frac{k^2 m - 1.\, j k n + \sqrt{25.\, j^4 + 25.\, j^2 k^2 - 1.\, j^2 k^2 m^2 + 2.\, j^3 k m n - 1.\, j^4 n^2}}{j^2 + k^2},\ \frac{-1.\, k m + \frac{k^3 m}{j^2+k^2} + j n - \frac{1.\, j k^2 n}{j^2+k^2} + \frac{k \sqrt{-1.\, j^2 \left(-25.\, j^2 - 25.\, k^2 + k^2 m^2 - 2.\, j k m n + j^2 n^2\right)}}{j^2+k^2}}{j}\right\}\right\}$$

Out[ ]=
$$\left\{\left\{m, -1.\, \sqrt{25. - 1.\, m^2}\right\}, \left\{m, \sqrt{25. - 1.\, m^2}\right\}\right\}$$

Out[ ]=
$$\left\{\left\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}, \{0, 1\}, \left\{-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}, \{-1, 0\}\right\}$$

Out[ ]=
```
{{{-3.53553, -3.53553}, {3.53553, 3.53553}}, {{1, -4.89898}, {1, 4.89898}},
 {{-2.39116, 4.39116}, {4.39116, -2.39116}}, {{-4.89898, 1.}, {4.89898, 1.}}}
```

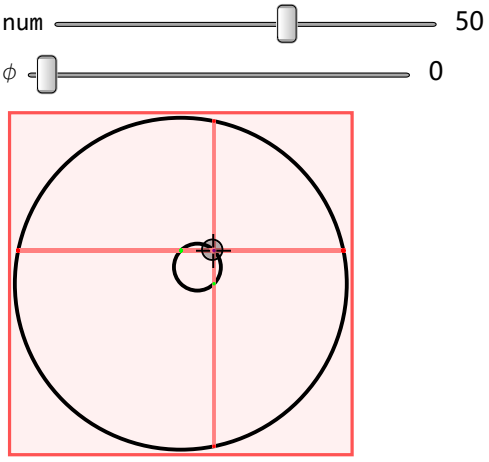
版本二

线段数num可以为任意整数，num = 50 时也是很丝滑。

In[ ]:=

```
DynamicModule[{m, n, j, k, x, y, dot, dota0, pt, num, r, ϕ},
 pt = {1, 1}; num = 50; r = 5; ϕ = 0;
 dot = {x, y} /. NSolve[{(y - n) j == (x - m) k, x^2 + y^2 == r^2}, {x, y}];
 dota0 = {x, y} /. NSolve[{x == m, x^2 + y^2 == r^2}, {x, y}];
 Column[{
   Labeled[Slider[Dynamic[num], {1, 80, 1}, Appearance → "Labeled"], "num", Left],
   Labeled[Slider[Dynamic[ϕ], {0, π}, Appearance → "Labeled"], "ϕ", Left],
   LocatorPane[
    Dynamic[pt, (pt = If[#[[1]]^2 + #[[2]]^2 > r^2, r # / EuclideanDistance[#, {0, 0}], #]) &],
    Dynamic[Graphics[{m, n} = pt;
      vectors = Table[AngleVector[θ], {θ, π/num + ϕ, π + ϕ, π/num}];
      endpoints = ({j, k} = #;
          If[j == 0, dota0, dot]) & /@ vectors;
      centers = (#1 + #2)/2 & @@ # & /@ endpoints;
      {Thick, Circle[{0, 0}, r],
       Circle[pt / 2, EuclideanDistance[{0, 0}, pt] / 2],
       Pink, Line@endpoints,
       Purple, PointSize[0.01], Point[pt],
       Red, PointSize[0.01], Point[Flatten[endpoints, 1]],
       Green, PointSize[0.01], Point[centers]
      },
      style
     ]]
   ]
 }]]
```

Out[ ]=



版本二（优化）

In[ ]:=
```
dot = {x, y} /. NSolve[{(y - n) j == (x - m) k, x^2 + y^2 == r^2}, {x, y}];
```
数值求解

```
dota0 = {x, y} /. NSolve[{x == m, x^2 + y^2 == r^2}, {x, y}];
```
数值求解

In[ ]:=
```
r = 5;
```

计算所有直线与圆的交点

In[ ]:=
```
calcPoints[pt_, num_] := (vectors = Table[AngleVector[θ], {θ, π/num, π, π/num}];
```
表格 角度向量

```
  endpoints =
   (If[#[[1]] == 0, dota0, dot] /. {j → #[[1]], k → #[[2]], m → pt[[1]], n → pt[[2]]} &) /@ vectors;
```
如果

```
  endpoints)
```

In[ ]:=
```
calcPoints[{1, 1}, 4]
```

Out[ ]=
```
{{{-3.53553, -3.53553}, {3.53553, 3.53553}}, {{1, -4.89898}, {1, 4.89898}},
 {{-2.39116, 4.39116}, {4.39116, -2.39116}}, {{-4.89898, 1.}, {4.89898, 1.}}}
```

计算上一步交点的中点

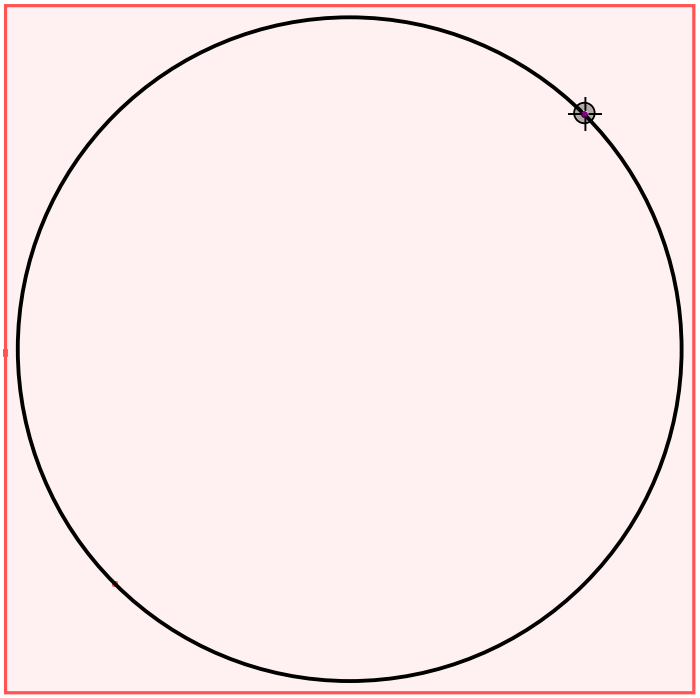
In[ ]:=
```
calcCenters[points_] := (#1 + #2)/2 & @@ # & /@ points;
```

In[ ]:=
```
calcCenters[calcPoints[{1, 1}, 4]]
```

Out[ ]=
$$\{\{0., 2.22045 \times 10^{-16}\}, \{1, 0.\}, \{1., 1.\}, \{0., 1.\}\}$$

In[ ]:=
```
DynamicModule[{pt, num},
 动态模块
 pt = {1, 1}; num = 20;
 LocatorPane[
 定位器窗格
  Dynamic[pt, (pt = If[#[[1]]^2 + #[[2]]^2 > r^2, r # / EuclideanDistance[#, {0, 0}], #]) &],
  动态                 如果                              欧几里得距离
  Dynamic[Graphics[
  动态      图形
    endpoints = calcPoints[pt, num];
    centers = calcCenters[endpoints];
    {Thick, Circle[{0, 0}, r],
     粗     圆
     Circle[pt / 2, EuclideanDistance[pt, {0, 0}] / 2],
     圆             欧几里得距离
     Pink, Line@endpoints,
     粉色  线段
     Purple, PointSize[0.01], Point[pt],
     紫色    点的大小          点
     Red, PointSize[0.01], Point[Flatten[endpoints, 1]],
     红色 点的大小          点    压平
     Green, PointSize[0.01], Point[centers]
     绿色   点的大小          点
    },
    style
   ]]
 ]]
```

Out[ ]=



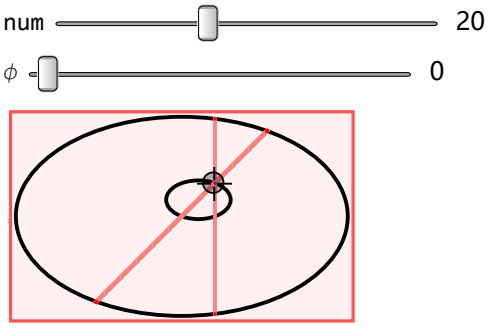
定位器可以在圆内的任何位置，

椭圆

```
In[ ]:= DynamicModule[{m, n, j, k, a, b, x, y, dot, dota0, pt, num, ϕ},
        动态模块

  a = 5; b = 3; pt = {1, 1}; num = 20; ϕ = 0;

  dot = {x, y} /. NSolve[{(y - n) j == (x - m) k, x^2/a^2 + y^2/b^2 == 1}, {x, y}];
                         数值求解

  dota0 = {x, y} /. NSolve[{x == m, x^2/a^2 + y^2/b^2 == 1}, {x, y}];
                           数值求解

  Column[{
  列

    Labeled[Slider[Dynamic[num], {1, 50, 1}, Appearance → "Labeled"], "num", Left],
    标记      滑动条  动态                   外观                    标记              左
    Labeled[Slider[Dynamic[ϕ], {0, π}, Appearance → "Labeled"], "ϕ", Left],
    标记      滑动条  动态               外观                    标记              左
    LocatorPane[ Dynamic[pt, (pt = If[#[[1]]^2/a^2 + #[[2]]^2/b^2 > 1, {j, k, m, n} = Join[#, #];
    定位器窗格   动态          如果                                                  连接

          Nearest[If[j == 0, dota0, dot], #][[1]] 0.99, #]) &],
          最接近   如果

    Dynamic[Graphics[{m, n} = pt;
    动态    图形

      vectors = Table[AngleVector[θ], {θ, π/num + ϕ, π + ϕ, π/num}];
                表格   角度向量

      endpoints = ({j, k} = #;
           If[j == 0, dota0, dot]) & /@ vectors;
           如果
      centers = calcCenters@endpoints;

      {Thick, Circle[{0, 0}, {a, b}],
      粗      圆

       Circle[pt / 2, {j, k} = pt;
       圆

         {a, b} EuclideanDistance[{0, 0}, pt] / EuclideanDistance @@ If[j == 0, dota0, dot]],

       Pink, Line@endpoints,
       粉色  线段

       Purple, PointSize[0.01], Point[pt],
       紫色    点的大小          点

       Red, PointSize[0.01], Point[Flatten[endpoints, 1]],
       红色 点的大小          点    压平

       Green, PointSize[0.01], Point[centers]
       绿色   点的大小          点
```

```
        },
        style
      ]]]
  }]]
```

Out[ ]=

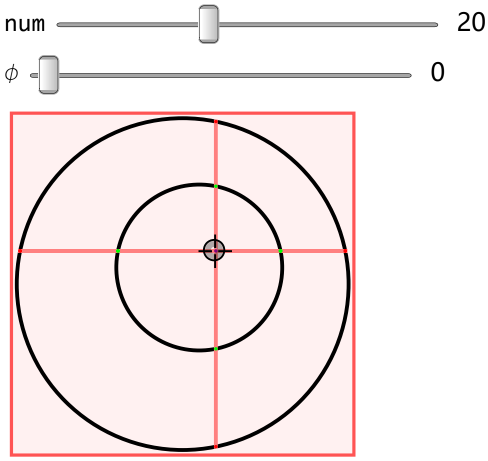


圆和椭圆

In[ ]:=

```
DynamicModule[{m, n, j, k, r, x, y, dot, dota0, pt, num, ϕ},
 动态模块

 r = 5; pt = {1, 1}; num = 20; ϕ = 0;
 dot = {x, y} /. NSolve[{(y - n) j == (x - m) k, x^2 + y^2 == r^2}, {x, y}];
                        数值求解
 dota0 = {x, y} /. NSolve[{x == m, x^2 + y^2 == r^2}, {x, y}];
                          数值求解
 Column[{
 列
   Labeled[Slider[Dynamic[num], {1, 50, 1}, Appearance → "Labeled"], "num", Left],
   标记    滑动条  动态                    外观              标记                 左
   Labeled[Slider[Dynamic[ϕ], {0, π}, Appearance → "Labeled"], "ϕ", Left],
   标记    滑动条  动态                外观              标记               左
   LocatorPane[
   定位器窗格
    Dynamic[pt, (pt = If[#[[1]]^2 + #[[2]]^2 > r^2, r # / EuclideanDistance[#, {0, 0}], #]) &],
    动态          如果                              欧几里得距离
    Dynamic[Graphics[{m, n} = pt;
    动态    图形
      vectors = Table[AngleVector[θ], {θ, π/num + ϕ, π + ϕ, π/num}];
                表格  角度向量

      endpoints = ({j, k} = #;
          If[j == 0, dota0, dot]) & /@ vectors;
          如果
      (*所有交点与动点的中点*)
      centers = (pt + #)/2 & /@ Flatten[endpoints, 1];
                                压平

      {Thick, Circle[{0, 0}, r],
       粗     圆
       Circle[pt / 2, r / 2],
       圆
       Pink, Line@endpoints,
       粉色  线段
       Purple, PointSize[0.01], Point[pt],
       紫色    点的大小          点
       Red, PointSize[0.01], Point[Flatten[endpoints, 1]],
       红色 点的大小          点    压平
       Green, PointSize[0.01], Point[centers]
       绿色   点的大小          点
      },
      style
    ]]]
 }]]
```

Out[ ]=



可以看到小圆始终是大圆的一半，圆心是动点和原点的中点。

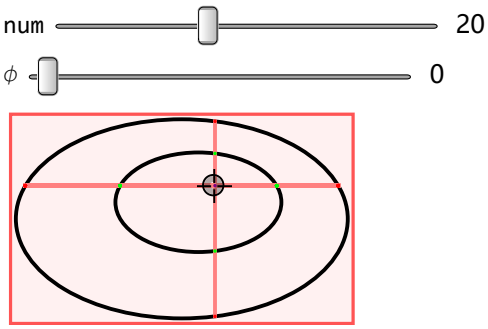
In[ ]:=

```
DynamicModule[{m, n, j, k, a, b, x, y, dot, dota0, pt, num, ϕ},
 a = 5; b = 3; pt = {1, 1}; num = 20; ϕ = 0;
 dot = {x, y} /. NSolve[{(y - n) j == (x - m) k, x^2/a^2 + y^2/b^2 == 1}, {x, y}];

 dota0 = {x, y} /. NSolve[{x == m, x^2/a^2 + y^2/b^2 == 1}, {x, y}];

 Column[{
   Labeled[Slider[Dynamic[num], {1, 50, 1}, Appearance → "Labeled"], "num", Left],
   Labeled[Slider[Dynamic[ϕ], {0, π}, Appearance → "Labeled"], "ϕ", Left],
   LocatorPane[ Dynamic[pt, (pt = If[#[[1]]^2/a^2 + #[[2]]^2/b^2 > 1, {j, k, m, n} = Join[#, #];

        Nearest[If[j == 0, dota0, dot], #][[1]] 0.99, #]) &],

    Dynamic[Graphics[{m, n} = pt;

      vectors = Table[AngleVector[θ], {θ, π/num + ϕ, π + ϕ, π/num}];

      endpoints = ({j, k} = #;
          If[j == 0, dota0, dot]) & /@ vectors;
      (*所有交点与动点的中点*)
      centers = (pt + #)/2 & /@ Flatten[endpoints, 1];
```

动态模块 · 数值求解 · 列 · 标记 · 滑动条 · 动态 · 外观 · 左 · 定位器窗格 · 如果 · 连接 · 最接近 · 图形 · 表格 · 角度向量 · 压平

```
    {Thick, Circle[{0, 0}, {a, b}],
     粗    圆
     Circle[pt / 2, {a, b} / 2],
     圆
     Pink, Line@endpoints,
     粉色  线段
     Purple, PointSize[0.01], Point[pt],
     紫色    点的大小          点
     Red, PointSize[0.01], Point[Flatten[endpoints, 1]],
     红色 点的大小          点    压平
     Green, PointSize[0.01], Point[centers]
     绿色   点的大小          点
    },
    style
   ]]]
 }]]
```

Out[ ]=



可以看到小椭圆始终为大椭圆的一半，中心是动点和原点的中点。

圆里生椭圆

将先前的线段绕中点旋转90°，所围成的图形就是椭圆

In[ ]:=

```
DynamicModule[{m, n, j, k, r, x, y, dot, dota0, pt, num, ϕ, lineType},
动态模块
 r = 5; pt = {1, 1}; num = 20; ϕ = 0;
 dot = {x, y} /. NSolve[{(y - n) j == (x - m) k, x^2 + y^2 == r^2}, {x, y}];
                        数值求解
 dota0 = {x, y} /. NSolve[{x == m, x^2 + y^2 == r^2}, {x, y}];
                          数值求解
 Column[{
 列
   Labeled[Slider[Dynamic[num], {1, 50, 1}, Appearance → "Labeled"], "num", Left],
   标记    滑动条  动态                    外观                  标记              左
   Labeled[Slider[Dynamic[ϕ], {0, π}, Appearance → "Labeled"], "ϕ", Left],
   标记    滑动条  动态                外观                  标记          左
   Labeled[SetterBar[Dynamic[lineType], {InfiniteLine, Line}], "lineType", Left],
   标记    设置按钮条  动态                  无限长直线          线段                左
   LocatorPane[ Dynamic[pt,
   定位器窗格   动态
```

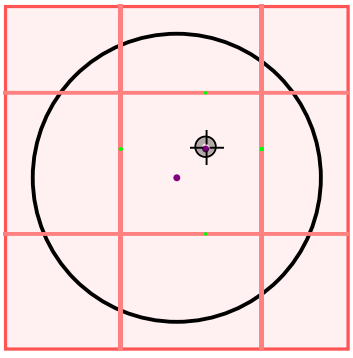
```
(pt = If[#[[1]]^2 + #[[2]]^2 > r^2, 0.99 r #/EuclideanDistance[#, {0, 0}], #]) &],
Dynamic[Graphics[{m, n} = pt;
  vectors = Table[AngleVector[θ], {θ, π/num + ϕ, π + ϕ, π/num}];
  endpoints = ({j, k} = #;
    If[j == 0, dota0, dot]) & /@ vectors;
  (*所有交点与动点的中点*)
  centers = (pt + #)/2 & /@ Flatten[endpoints, 1];
  {Thick, Circle[{0, 0}, r],
   (*Circle[pt/2,r/2],*)
   Pink, GeometricTransformation[lineType[{#, pt}],
    RotationTransform[π/2, (pt + #)/2]] & /@ Flatten[endpoints, 1],
   Purple, PointSize[0.02], Point[{pt, {0, 0}}],
   (*Red,PointSize[0.01],Point[Flatten[endpoints,1]],*)
   Green, PointSize[0.01], Point[centers]
  },
  style,
  PlotRange → r + 1
 ]]
}]]
```

Out[ ]=

num 20

ϕ 0

lineType InfiniteLine Line



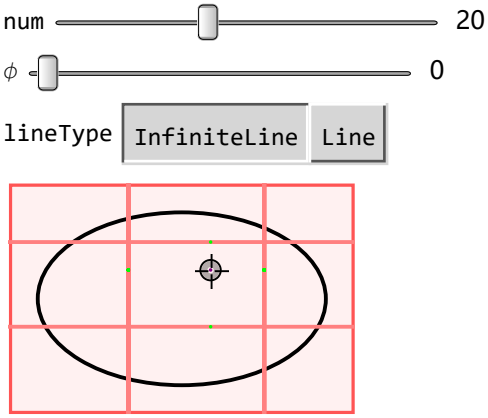
可以证明所有直线围成的图形是一个椭圆，焦点是圆心和动点

把上面的规则应用到椭圆

```
In[ ]:= DynamicModule[{m, n, j, k, a, b, x, y, dot, dota0, pt, num, ϕ, lineType},
        (动态模块)

  a = 5; b = 3; pt = {1, 1}; num = 20; ϕ = 0;

  dot = {x, y} /. NSolve[{(y - n) j == (x - m) k, x^2/a^2 + y^2/b^2 == 1}, {x, y}];
                        (数值求解)

  dota0 = {x, y} /. NSolve[{x == m, x^2/a^2 + y^2/b^2 == 1}, {x, y}];
                          (数值求解)

  Column[{
   (列)

    Labeled[Slider[Dynamic[num], {1, 50, 1}, Appearance → "Labeled"], "num", Left],
    (标记 滑动条 动态 外观 标记 左)
    Labeled[Slider[Dynamic[ϕ], {0, π}, Appearance → "Labeled"], "ϕ", Left],
    (标记 滑动条 动态 外观 标记 左)

    Labeled[SetterBar[Dynamic[lineType], {InfiniteLine, Line}], "lineType", Left],
    (标记 设置按钮条 动态 无限长直线 线段 左)

    LocatorPane[ Dynamic[pt, (pt = If[#[[1]]^2/a^2 + #[[2]]^2/b^2 > 1, {j, k, m, n} = Join[#, #];
    (定位器窗格 动态 如果 连接)

          Nearest[If[j == 0, dota0, dot], #][[1]] 0.99, #]) &],
          (最接近 如果)

    Dynamic[Graphics[
    (动态 图形)

      {m, n} = pt;
      vectors = Table[AngleVector[θ], {θ, π/num + ϕ, π + ϕ, π/num}];
                (表格 角度向量)

      endpoints = ({j, k} = #;
          If[j == 0, dota0, dot]) & /@ vectors;
          (如果)
      (*所有交点与动点的中点*)
      centers = (pt + #)/2 & /@ Flatten[endpoints, 1];
                                (压平)

      {Thick, Circle[{0, 0}, {a, b}],
       (粗 圆)
       (*Circle[pt/2,{a,b}/2],*)
        (圆)
       Pink, GeometricTransformation[lineType[{#, pt}],
       (粉色 几何变换)
          RotationTransform[π / 2, (pt + #) / 2]] & /@ Flatten[endpoints, 1],
          (旋转变换)                                  (压平)
```

```
旋转变换                                                    压平
 Purple, PointSize[0.01], Point[pt],
 紫色     点的大小              点
 (*Red,PointSize[0.01],Point[Flatten[endpoints,1]],*)
   红色 点的大小              点      压平
 Green, PointSize[0.01], Point[centers]
 绿色     点的大小              点
},
style,
PlotRange → ({-1, 1} # & /@ {a + 1, b + 1})
绘制范围
]]]
}]]
```

Out[ ]=



看起来很美，但我不知道该怎么叫它，有时像个水滴，有时像个眼睛。