

1 交点

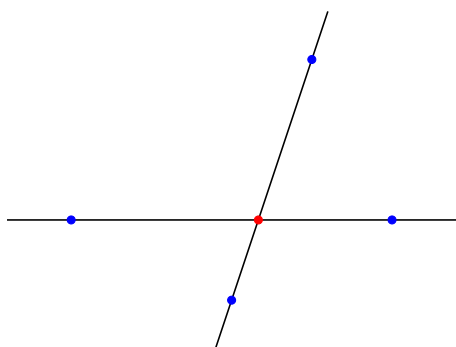
可以求得两条直线、一条直线与一个圆及两个圆之间的交点。
求交点的相关命令没有可选参数，用户必须确保交点存在。

1.1 `\tkzInterLL`命令：求两条直线的交点

`\tkzInterLL($\langle A,B \rangle$)($\langle C,D \rangle$)`

求直线 (AB) 和 (CD) 的交点，并保存于`\tkzPointResult`命令中，两条直线分别由两个圆括号中的点对定义。可以通过`\tkzDefPoint`保存并命令交点。

1.1.1 直线交点示例



```
\begin{tikzpicture}[rotate=-45,scale=.75]
\tkzDefPoint(2,1){A}
\tkzDefPoint(6,5){B}
\tkzDefPoint(3,6){C}
\tkzDefPoint(5,2){D}
\tkzDrawLines(A,B C,D)
\tkzInterLL(A,B)(C,D)
\tkzGetPoint{I}
\tkzDrawPoints[color=blue](A,B,C,D)
\tkzDrawPoint[color=red](I)
\end{tikzpicture}
```

1.2 `\tkzInterLC`命令：定义一条直线和一个圆的交点

直线可以由两个点定义，圆可以按如下方式进行定义：

- (O,C) O 是圆心， C 是圆上的一个点。
- (O,r) O 是圆心， r 半径，单位可以是 *cm* 可 *pt*。

`\tkzInterLC[\langle 命令选项 \rangle]($\langle A,B \rangle$)($\langle O,C \rangle$) 或 ($\langle O,r \rangle$) 或 ($\langle O,C,D \rangle$)`

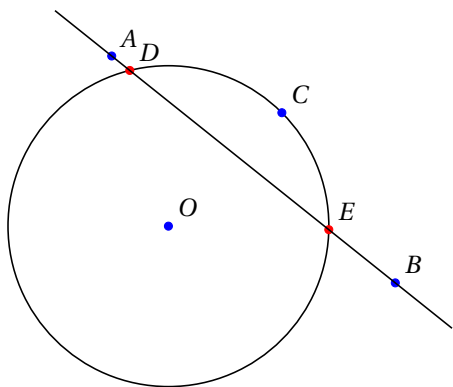
参数必须是一条直线和一个圆。

选项	默认值	含义
N	N	(O,C)
R	N	$(O, 1 \text{ cm})$ 或 $(O, 120 \text{ pt})$
with nodes	N	(O,C,D) CD 是半径

定义直线与由圆心 O 和半径 r 定义的圆的交点 I 和 J ，如果出现错误，则记录在 `.log` 日志文件中。

1.2.1 直线与圆的交点示例

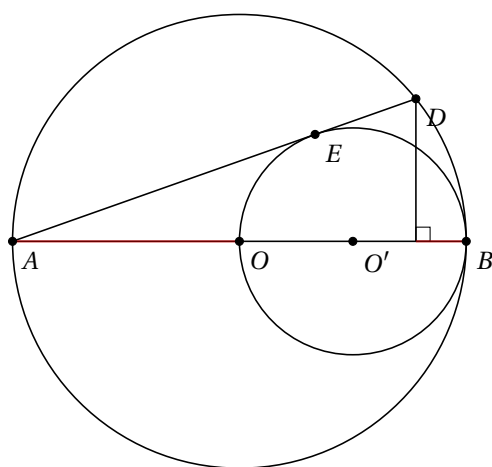
在下面示例代码中，圆用两个点表示，直线与圆的有两个交点。



```
\begin{tikzpicture}[scale=.75]
  \tkzInit[xmax=5,ymax=4]
  \tkzDefPoint(1,1){O}
  \tkzDefPoint(0,4){A}
  \tkzDefPoint(5,0){B}
  \tkzDefPoint(3,3){C}
  \tkzInterLC(A,B)(O,C) \tkzGetPoints{D}{E}
  \tkzDrawCircle(O,C)
  \tkzDrawPoints[color=blue](O,A,B,C)
  \tkzDrawPoints[color=red](D,E)
  \tkzDrawLine(A,B)
  \tkzLabelPoints[above right](O,A,B,C,D,E)
\end{tikzpicture}
```

1.2.2 直线与圆的交点复杂示例

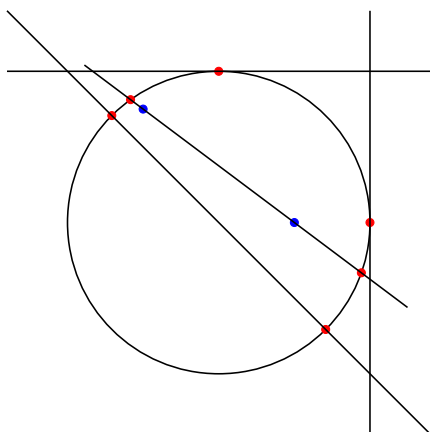
可参阅http://www.gogeometry.com/problem/p190_tangent_circle_diameter_perpendicular.htm



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){A} \tkzDefPoint(8,0){B}
  \tkzDefMidPoint(A,B) \tkzGetPoint{O}
  \tkzDrawCircle(O,B)
  \tkzDefMidPoint(O,B) \tkzGetPoint{O'}
  \tkzDrawCircle(O',B)
  \tkzDefTangent[from=A](O',B)
  \tkzGetSecondPoint{E}
  \tkzInterLC(A,E)(O,B)
  \tkzGetSecondPoint{D}
  \tkzDefPointBy[projection=onto A--B](D)
  \tkzGetPoint{F}
  \tkzMarkRightAngle(D,F,B)
  \tkzDrawSegments(A,D A,B D,F)
  \tkzDrawSegments[color=red,line width=1pt,
    opacity=.4](A,O F,B)
  \tkzDrawPoints(A,B,O,O',E,D)
  \tkzLabelPoints(A,B,O,O',E,D)
\end{tikzpicture}
```


1.2.3 由圆心和半径定义圆示例

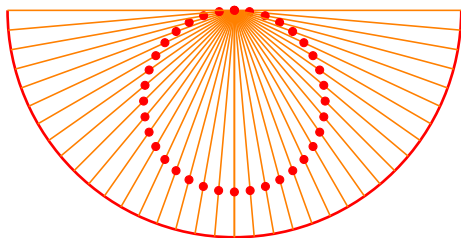
直线与圆相切的特例：



```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(0,8){A} \tkzDefPoint(8,0){B}
\tkzDefPoint(8,8){C} \tkzDefPoint(4,4){I}
\tkzDefPoint(2,7){E} \tkzDefPoint(6,4){F}
\tkzDrawCircle[R](I,4 cm)
\tkzInterLC[R](A,C)(I,4 cm) \tkzGetPoints{I1}{I2}
\tkzInterLC[R](B,C)(I,4 cm) \tkzGetPoints{J1}{J2}
\tkzInterLC[R](A,B)(I,4 cm) \tkzGetPoints{K1}{K2}
\tkzDrawPoints[color=red](I1,J1,K1,K2)
\tkzDrawLines(A,B B,C A,C)
\tkzInterLC[R](E,F)(I,4 cm) \tkzGetPoints{I2}{J2}
\tkzDrawPoints[color=blue](E,F)
\tkzDrawPoints[color=red](I2,J2)
\tkzDrawLine(I2,J2)
\end{tikzpicture}
```

1.2.4 更为复杂的示例

 注意语法细节：首先，在传递参数的同时，可以计算点的坐标，但是必须嵌套`xfp`语法。由于`xfp`宏包能够使用弧度，如使用`pi`，当然，也可以使用度，但需要使用类似`sind`或`cosd`命令进行计算。其次，当计算中需要圆括号时，需要使用分组命令：`...TeX{ ...}`。



```
\begin{tikzpicture}[scale=1.2]
\tkzDefPoint(0,1){J} \tkzDefPoint(0,0){O}
\tkzDrawArc[R,line width=1pt,color=red](J,2.5 cm)(180,0)
\foreach \i in {0,-5,-10,...,-85,-90}{
\tkzDefPoint({2.5*cosd(\i)},{1+2.5*sind(\i)}){P}
\tkzDrawSegment[color=orange](J,P)
\tkzInterLC[R](P,J)(O,1 cm)
\tkzGetPoints{M}{N}
\tkzDrawPoints[red](N)
}
\foreach \i in {-90,-95,...,-175,-180}{
\tkzDefPoint({2.5*cosd(\i)},{1+2.5*sind(\i)}){P}
\tkzDrawSegment[color=orange](J,P)
\tkzInterLC[R](P,J)(O,1 cm)
\tkzGetPoints{M}{N}
\tkzDrawPoints[red](M)
}
\end{tikzpicture}
```

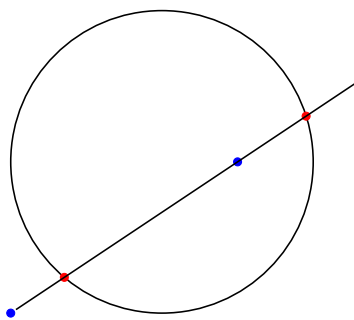
1.3 半径计算

使用`pgfmath`模块的`\pgfmathsetmacro`命令进行计算。

半径的计算是提前完成的，不是在交点计算命令中计算的。可以有多种方式计算长度，一种方式是使用`pgfmath`模块的`\pgfmathsetmacro`命令进行计算。某些情况下，其精度不足，如 $0.0002 \div 0.0001$ 的结果是 1.98，但如使用`xfp`宏包进行计算，则其结果为 2。

1.3.1 半径计算示例 1

使用`xfp`宏包的`\fpeval`命令：

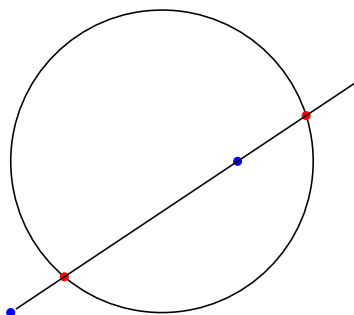


```
\begin{tikzpicture}
\tkzDefPoint(2,2){A}
\tkzDefPoint(5,4){B}
\tkzDefPoint(4,4){O}
\edef\tkzLen{\fpeval{0.0002/0.0001}}
\tkzDrawCircle[R](O,\tkzLen cm)
\tkzInterLC[R](A,B)(O,\tkzLen cm)
\tkzGetPoints{I}{J}
\tkzDrawPoints[color=blue](A,B)
\tkzDrawPoints[color=red](I,J)
\tkzDrawLine(I,J)
\end{tikzpicture}
```

1.3.2 半径计算示例 2

使用 $\text{T}_\text{E}\text{X}$ 的 `\tkzLength` 命令计算。

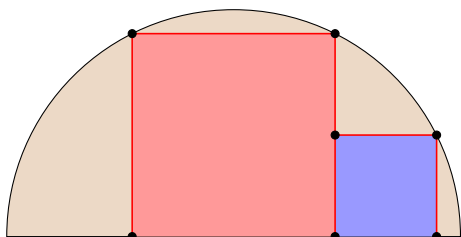
用 `\newdimen` 命令定义一尺寸。当然，用的是 $\text{T}_\text{E}\text{X}$ 进行计算。



```
\begin{tikzpicture}
\tkzDefPoints{2/2/A,5/4/B,4/4/O}
\tkzLength=2cm
\tkzDrawCircle[R](O,\tkzLength)
\tkzInterLC[R](A,B)(O,\tkzLength)
\tkzGetPoints{I}{J}
\tkzDrawPoints[color=blue](A,B)
\tkzDrawPoints[color=red](I,J)
\tkzDrawLine(I,J)
\end{tikzpicture}
```

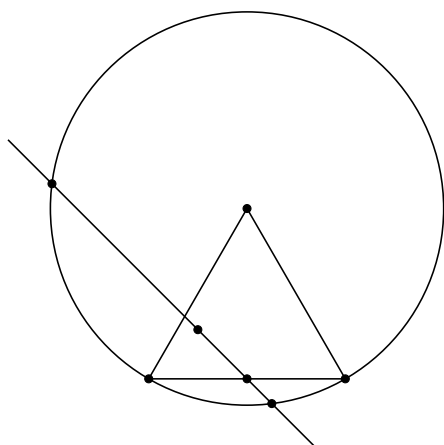
1.3.3 半圆中的矩形示例

在一个半圆内同时画两个正方形是比较困难的，需要通过圆的半径计算正方形 S 边长。



```
\begin{tikzpicture}[scale=.75]
\tkzDefPoints{0/0/A,8/0/B,4/0/I}
\tkzDefSquare(A,B) \tkzGetPoints{C}{D}
\tkzInterLC(I,C)(I,B) \tkzGetPoints{E'}{E}
\tkzInterLC(I,D)(I,B) \tkzGetPoints{F'}{F}
\tkzDefPointsBy[projection = onto A--B](E,F){H,G}
\tkzDefPointsBy[symmetry = center H](I){J}
\tkzDefSquare(H,J) \tkzGetPoints{K}{L}
\tkzDrawSector[fill=brown!30](I,B)(A)
\tkzFillPolygon[color=red!40](H,E,F,G)
\tkzFillPolygon[color=blue!40](H,J,K,L)
\tkzDrawPolySeg[color=red](H,E,F,G)
\tkzDrawPolySeg[color=red](J,K,L)
\tkzDrawPoints(E,G,H,F,J,K,L)
\end{tikzpicture}
```

1.3.4 "with nodes"选项示例



```
\begin{tikzpicture}[scale=.65]
\tkzDefPoints{0/0/A,4/0/B,1/1/D,2/0/E}
\tkzDefTriangle[equilateral](A,B)
\tkzGetPoint{C}
\tkzDrawCircle(C,A)
\tkzInterLC[with nodes](D,E)(C,A,B)
\tkzGetPoints{F}{G}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,...,G)
\tkzDrawLine(F,G)
\end{tikzpicture}
```

1.4 \tkzInterCC命令：求两个圆的交点

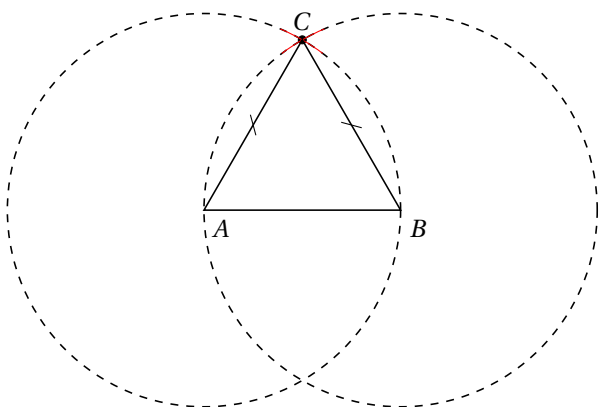
通常，两个圆是由圆心和另一个点确定的，但也可以用R选项后，在参数中指定半径。

\tkzInterCC[< 命令选项>](<O,A>)(<O',A'>) 或 (<O,r>)(<O',r'>) 或 (<O,A,B>)(<O',C,D>)

选项	默认值	含义
N	N	OA 和 O'A' 是半径，O 和 O' 是圆心
R	N	r 和 r' 是半径
with nodes	N	在 (A,A,C)(C,B,F) 中 AC 和 BF 是半径

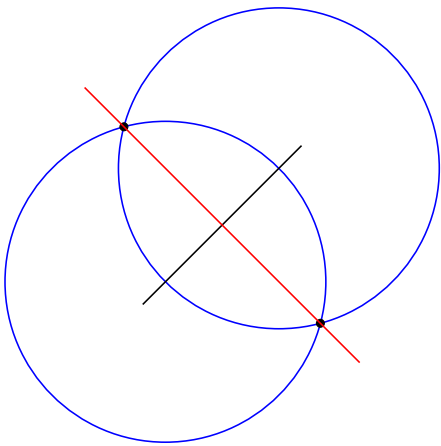
该命令定义了O和O'两个圆的交点I和J,如果两个圆没有交点,则返回错误。也可以直接使用\tkzInterCCN命令和\tkzInterCCR命令进行计算。

1.4.1 构造等边三角形示例



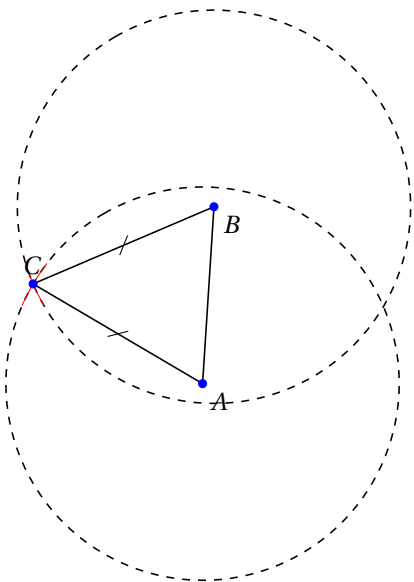
```
\begin{tikzpicture}[trim left=-1cm,scale=0.65]
\tkzDefPoint(1,1){A}
\tkzDefPoint(5,1){B}
\tkzInterCC(A,B)(B,A)\tkzGetPoints{C}{D}
\tkzDrawPoint[color=black](C)
\tkzDrawCircle[dashed](A,B)
\tkzDrawCircle[dashed](B,A)
\tkzCompass[color=red](A,C)
\tkzCompass[color=red](B,C)
\tkzDrawPolygon(A,B,C)
\tkzMarkSegments[mark=s|](A,C B,C)
\tkzLabelPoints[] (A,B)
\tkzLabelPoint[above](C){C}
\end{tikzpicture}
```

1.4.2 求中点示例



```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(2,2){B}
  \tkzDrawCircle[color=blue](B,A)
  \tkzDrawCircle[color=blue](A,B)
  \tkzInterCC(B,A)(A,B)\tkzGetPoints{M}{N}
  \tkzDrawLine(A,B)
  \tkzDrawPoints(M,N)
  \tkzDrawLine[color=red](M,N)
\end{tikzpicture}
```

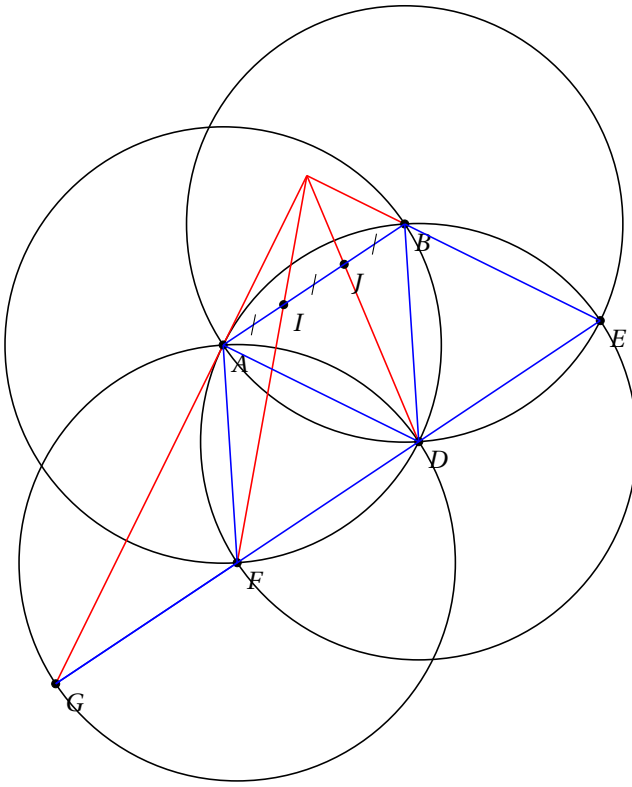
1.4.3 求等腰三角形示例



```
\begin{tikzpicture}[rotate=120,scale=0.65]
  \tkzDefPoint(1,2){A}
  \tkzDefPoint(4,0){B}
  \tkzInterCC[R](A,4cm)(B,4cm)
  \tkzGetPoints{C}{D}
  \tkzDrawCircle[R,dashed](A,4 cm)
  \tkzDrawCircle[R,dashed](B,4 cm)
  \tkzCompass[color=red](A,C)
  \tkzCompass[color=red](B,C)
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPoints[color=blue](A,B,C)
  \tkzMarkSegments[mark=s](A,C B,C)
  \tkzLabelPoints[] (A,B)
  \tkzLabelPoint[above](C){C}
\end{tikzpicture}
```

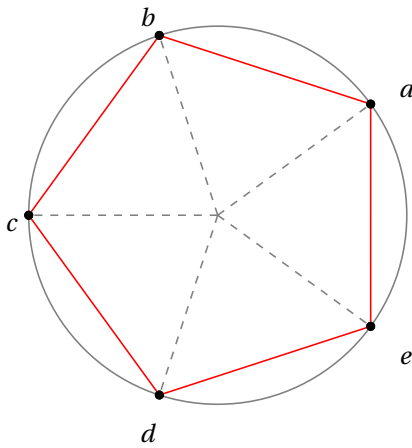
1.4.4 三等分线段示例

与尺规作图方式将线段三等分操作一致。



```
\begin{tikzpicture}[scale=0.80]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(3,2){B}
  \tkzInterCC(A,B)(B,A)
  \tkzGetPoints{C}{D}
  \tkzInterCC(D,B)(B,A)
  \tkzGetPoints{A}{E}
  \tkzInterCC(D,B)(A,B)
  \tkzGetPoints{F}{B}
  \tkzInterLC(E,F)(F,A)
  \tkzGetPoints{D}{G}
  \tkzInterLL(A,G)(B,E)
  \tkzGetPoint{O}
  \tkzInterLL(O,D)(A,B)
  \tkzGetPoint{J}
  \tkzInterLL(O,F)(A,B)
  \tkzGetPoint{I}
  \tkzDrawCircle(D,A)
  \tkzDrawCircle(A,B)
  \tkzDrawCircle(B,A)
  \tkzDrawCircle(F,A)
  \tkzDrawSegments[color=red](O,G
    O,B O,D O,F)
  \tkzDrawPoints(A,B,D,E,F,G,I,J)
  \tkzLabelPoints(A,B,D,E,F,G,I,J)
  \tkzDrawSegments[blue](A,B B,D A,D
    A,F F,G E,G B,E)
  \tkzMarkSegments[mark=s|](A,I I,J J,B)
\end{tikzpicture}
```

1.4.5 with nodes 选项示例



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{0/0/a,0/5/B,5/0/C}
  \tkzDefPoint(54:5){F}
  \tkzDrawCircle[color=gray](A,C)
  \tkzInterCC[with nodes](A,A,C)(C,B,F)
  \tkzGetPoints{a}{e}
  \tkzInterCC(A,C)(a,e) \tkzGetFirstPoint{b}
  \tkzInterCC(A,C)(b,a) \tkzGetFirstPoint{c}
  \tkzInterCC(A,C)(c,b) \tkzGetFirstPoint{d}
  \tkzDrawPoints(a,b,c,d,e)
  \tkzDrawPolygon[color=red](a,b,c,d,e)
  \foreach \vertex/\num in {a/36,b/108,c/180,
    d/252,e/324}{%
    \tkzDrawPoint(\vertex)
    \tkzLabelPoint[label=\num:$\vertex$](\vertex){}
    \tkzDrawSegment[color=gray,style=dashed](A,\vertex)
  }
\end{tikzpicture}
```