

# 二维计算几何基础

我们将需要解决的几何问题的范围限制在二维平面内，这样就涉及到二维计算几何的基本知识。在研究时，通常把图形放在平面直角坐标系或极坐标系下，这样解决问题就会方便很多。



## 前置技能

如并不了解：

- 几何基础
- 平面直角坐标系
- 向量（包括向量积）
- 极坐标与极坐标系

请先阅读 [向量](#) 和 [极坐标](#)。

## 图形的记录

### 点

在平面直角坐标系下，点用横纵坐标表示，比如点  $(5, 2)$ ，点  $(-1, 0)$  等。由于坐标是有序数对，因此可以利用 `std::pair` 存储，也可以定义结构体存储。

在极坐标系下，点用极径与极角表示，使用类似于直角坐标系的存储方式进行存储即可。

### 向量

由于向量的坐标表示与点相同，所以只需要像点一样存储向量即可。

在极坐标系下，与点同理。

### 线

在极坐标系中表示线是较为困难的，因此我们仅讨论直角坐标系中的情况。

#### 直线与射线

一般在求解数学问题时，我们用解析式表示一条直线。直线的解析式有一般式  $Ax + By + C = 0$ 、斜截式  $y = kx + b$ 、截距式  $\frac{x}{a} + \frac{y}{b} = 1$  等多种形式，但其形式最终服务于方程求解析解。在解决一些计算几何问题中，使用解析式表示直线会增加解决问题的难度。

在计算几何中，通常使用直线上一点与直线的方向向量表示一条直线，用射线的端点与射线的方向向量表示一条射线。方向向量是一个与直线平行的非零向量，通常情况下取单位向量。与一条直线平行的单位向量有两个，可以根据题目选择任意一个；而对于射线，方向向量通常取方向与射线延伸方向相同的那个。

## 线段

记录一条线段只需要记录其左右端点即可。

## 多边形

使用数组按一定顺序记录多边形的每个顶点即可。对于简单多边形，一般按顺时针或逆时针顺序记录各个顶点。

特殊地，记录矩形时，如果矩形的各边均与坐标轴平行，只需记录其对角线上的两个顶点。

## 曲线图形

一些特殊曲线，如函数图像等一般记录其解析式。对于圆，直接记录其圆心和半径即可。

## 基本公式

### 正弦定理

在  $\triangle ABC$  中，若角  $A, B, C$  所对边分别为  $a, b, c$ ，则有：

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C} = 2R$$

其中， $R$  为  $\triangle ABC$  的外接圆半径。

### 余弦定理

在  $\triangle ABC$  中，若角  $A, B, C$  所对边分别为  $a, b, c$ ，则有：

$$\begin{aligned}a^2 &= b^2 + c^2 - 2bc \cos A \\b^2 &= a^2 + c^2 - 2ac \cos B \\c^2 &= a^2 + b^2 - 2ab \cos C\end{aligned}$$

上述公式的证明略。均为人教版高中数学 A 版必修二内容（旧教材为必修五）。

## 基本操作

### 判断一个点在直线的哪边

我们有直线上的一点  $P$  的直线的方向向量  $\mathbf{v}$ ，想知道某个点  $Q$  在直线的哪边。这里所指的「哪边」是相对于方向向量的，也就是说，判断时我们面朝方向向量所指的方向进行判断。

我们利用向量积的性质，算出  $\mathbf{v} \times \overrightarrow{PQ}$ 。根据右手定则，如果向量积为负，则  $Q$  在直线右侧；如果向量积为 0，则  $Q$  在直线上；如果向量积为正，则  $Q$  在直线左侧。

## 快速排斥实验与跨立实验

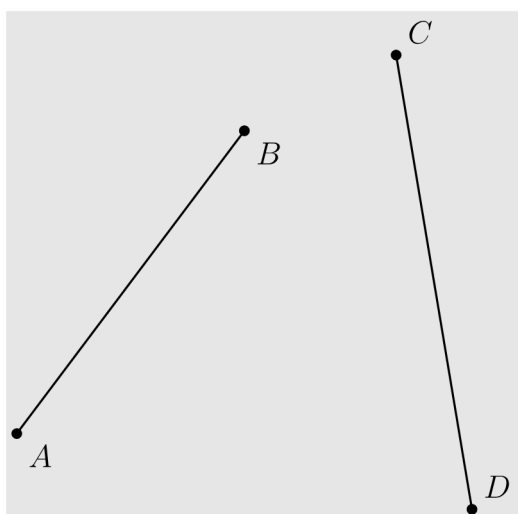
我们现在想判断两条线段是否相交。

首先特判一些特殊情况。如果两线段平行，自然不能相交。这种情况通过判断线段所在直线的斜率是否相等即可。

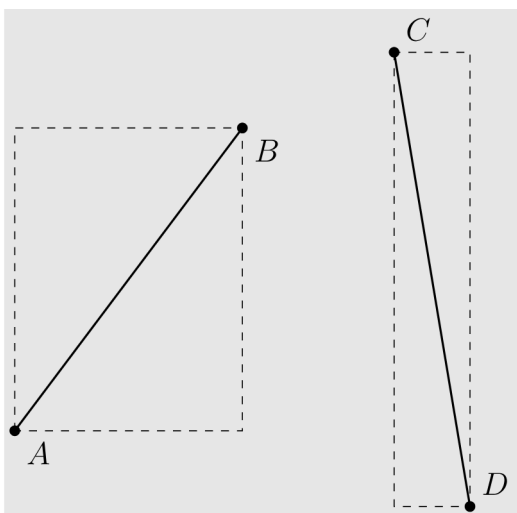
当然，如果两线段重合或部分重合，或者两线段的交点为其中一条线段的端点，只需要判断是否有三点共线的情况即可。

还有些显然不相交的情况，我们口头上称之为「两条线段离着太远了」。规定「一条线段的区域」为以这条线段为对角线的，各边均与某一坐标轴平行的矩形所占的区域，那么可以发现，如果两条线段没有公共区域，则这两条线段一定不相交。

比如有以下两条线段：



它们占用的区域是这样的：



于是可以快速判断出来这两条线段不相交。

这就是 **快速排斥实验**。上述情况称作 **未通过快速排斥实验**。

未通过快速排斥实验是两线段无交点的 **充分不必要条件**，我们还需要进一步判断。

因为两线段  $a, b$  相交， $b$  线段的两个端点一定分布在  $a$  线段所在直线两侧；同理， $a$  线段的两个端点一定分布在  $b$  线段所在直线两侧。我们可以直接判断一条线段的两个端点相对于另一线段所在直线的位置关系，如果不同，则两线段相交，反之则不相交。如上一节所说，直线与点的位置关系我们可以利用向量积判断。

这就是 **跨立实验**，如果对于两线段  $a, b$ ， $b$  线段的两个端点分布在  $a$  线段所在直线的两侧，且  $a$  线段的两个端点分布在  $b$  线段所在直线的两侧，我们就说  $a, b$  两线段 **通过了跨立实验**，即两线段相交。

注意到当两条线段共线但不相交时也可以通过跨立实验，因此想要准确判断还需要与快速排斥实验结合。

## 判断一点是否在多边形内部

在计算几何中，这个问题被称为 **PIP 问题**，已经有一些成熟的解决方法，下面依次介绍。

### 光线投射算法 (Ray casting algorithm)

在 [这里](#) 可以看到最原始的思路。

我们先特判一些特殊情况，比如「这个点离多边形太远了」。类似判断两条线段相交，考虑一个能够完全覆盖该多边形的最小矩形，如果这个点不在这个矩形范围内，那么这个点一定不在多边形内。这样的矩形很好求，只需要知道多边形横坐标与纵坐标的最小值和最大值，坐标两两组合成四个点，就是这个矩形的四个顶点了。

还有点在多边形的某一边或某顶点上，同样十分容易判断。对于每条边，检查点是否在边上或顶点上即可。

我们考虑以该点为端点引出一条射线，如果这条射线与多边形有奇数个交点，则该点在多边形内部，否则该点在多边形外部，我们简记为 **奇内偶外**。这个算法同样被称为奇偶规则 (Even-odd rule)。

由于 [Jordan curve theorem](#)，我们知道，这条射线每次与多边形的一条边相交，就切换一次与多边形的内外关系，所以统计交点数的奇偶即可。

对于选取的射线，可以随机选取这条射线所在直线的斜率，建议为无理数以避免出现射线与多边形某边重合的情况。

在原版代码中，使用的是记录多边形的数组中最后一个点作为射线上一点，这样统计时，如果出现射线过多边形某边或某顶点时，可以规定射线经过的点同在射线一侧，进而做跨立实验即可。

#### 注意

光线投射算法只适用于简单多边形。对于边自交的多边形并不适用 [Jordan curve theorem](#)，因此无法判断。

### 回转数算法 (Winding number algorithm)

回转数是数学上的概念，是平面内闭合曲线逆时针绕过该点的总次数。很容易发现，当回转数等于 0 的时候，点在曲线外部。这个算法同样被称为非零规则 (Nonzero-rule)。

如何计算呢？我们把该点与多边形的所有顶点连接起来，计算相邻两边夹角的和。注意这里的夹角是 **有方向的**。如果夹角和为 0，则这个点在外，否则在内。

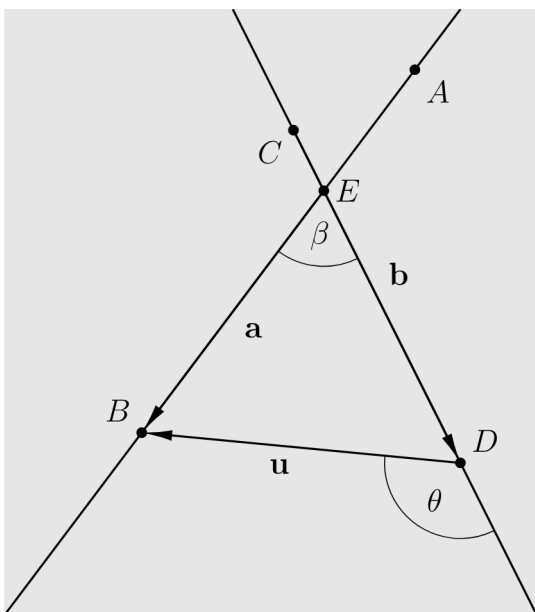
### 求两条直线的交点

首先，我们需要确定两条直线相交，只需判断一下两条直线的方向向量是否平行即可。如果方向向量平行，则两条直线平行，交点个数为 0。进一步地，若两条直线平行且过同一点，则两直线重合。

那么，问题简化为我们有直线  $AB, CD$  交于一点，想求出交点  $E$ 。

如果两直线相交，则交点只有一个，我们记录了直线上的一个点和直线的方向向量，所以我们只需要知道这个点与交点的距离  $l$ ，再将这个点沿方向向量平移  $l$  个单位长度即可。

考虑构造三角形，利用正弦定理求解  $l$ ，可以利用向量积构造出正弦定理。



由上图可知， $|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}||\mathbf{b}|\sin\beta$ ， $|\mathbf{u} \times \mathbf{b}| = |\mathbf{u}||\mathbf{b}|\sin\theta$ 。

作商得：

$$T = \frac{|\mathbf{u} \times \mathbf{b}|}{|\mathbf{a} \times \mathbf{b}|} = \frac{|\mathbf{u}|\sin\theta}{|\mathbf{a}|\sin\beta}$$

可以看出， $\left| \frac{|\mathbf{u}|\sin\theta}{\sin\beta} \right| = l$ 。若绝对值内部式子取值为正，代表沿  $\mathbf{a}$  方向平移，反之则为反方向。

同时，我们将  $T$  直接乘上  $\mathbf{a}$ ，就自动出现了直线的单位向量，不需要进行其他消去操作了。

于是，只需要将点  $B$  减去  $T\mathbf{a}$  即可得出交点。

## 求任意多边形的周长和面积

### 求任意多边形的周长

直接计算即可，简洁即美德。

### 求任意多边形的面积

考虑向量积的模的几何意义，我们可以利用向量积完成。

将多边形上的点逆时针标记为  $p_1, p_2, \dots, p_n$ ，再任选一个辅助点  $O$ ，记向量  $\mathbf{v}_i = p_i - O$ ，那么这个多边形面积  $S$  可以表示为：

$$S = \frac{1}{2} \left| \sum_{i=1}^n \mathbf{v}_i \times \mathbf{v}_{(i \bmod n)+1} \right|$$

## 圆与直线相关

## 求直线与圆的交点

首先判断直线与圆的位置关系。如果直线与圆相离则无交点，若相切则可以利用切线求出切点与半径所在直线，之后转化为求两直线交点。

若有两交点，则可以利用勾股定理求出两交点的中点，然后沿直线方向加上半弦长即可。

## 求两圆交点

首先我们判断一下两个圆的位置关系，如果外离或内含则无交点，如果相切，可以算出两圆心连线的方向向量，然后利用两圆半径计算出平移距离，最后将圆心沿这个方向向量进行平移即可。

如果两圆相交，则必有两个交点，并且关于两圆心连线对称。因此下面只说明一个交点的求法，另一个交点可以用类似方法求出。

我们先将一圆圆心与交点相连，求出两圆心连线与该连线所成角。这样，将两圆心连线的方向向量旋转这个角度，就是圆心与交点相连形成的半径的方向向量了。沿方向向量方向将圆心平移半径长度即可得到一个交点。

## 极角序

一般来说，这类题需要先枚举一个极点，然后计算出其他点的极坐标，在极坐标系下按极角的顺序解决问题。



### 「JOISC 2014 Day4」两个人的星座



平面内有  $n$  个点，有三种颜色，每个点的颜色是三种中的一种。求不相交的三色三角形对数。  
 $6 \leq n \leq 3000$ 。

## 题解

如果两个三角形不相交，则一定可以做出两条内公切线，如果相交或内含是做不出内公切线的。三角形的公切线可以类比圆的公切线。

先枚举一个原点，记为  $O$ ，以这个点为极点，过这个点且与  $x$  轴平行的直线作为极轴，建立极坐标系，把剩余点按极角由小到大排序。然后统计出在极轴上方和下方的每种点的个数。

然后根据点枚举公切线，记枚举到的点为  $P$ ，初始时公切线为极轴。开始统计。那么一定存在一条公切线过点  $O$  和点  $P$ 。因为公切线与三角形不相交，所以一方选择公切线上方的点，另一方一定选择下方的点。然后利用乘法原理统计方案数即可。

统计完后旋转公切线，那么点  $P$  一定改变了相对于公切线的上下位置，而其他点不动，应该只将它的位置信息改变。这样，可以发现，同一对三角形最终被统计了 4 次，就是同一条公切线会被枚举两次，最后做出的答案应除以 4。

分析一下算法复杂度。我们枚举了一个原点，然后对于每一个原点将剩余点排序后线性统计，因此时间复杂度为  $O(n^2 \log n)$ 。

## 代码编写注意事项

由于计算几何经常进行 `double` 类型的浮点数计算，因此带来了精度问题和时间问题。

有些问题，例如求点坐标均为整数的三角形面积，可以利用其特殊性进行纯整数计算，避免用浮点数影响精度。

由于浮点数的读入与计算都比整数慢，所以需要注意程序的常数因子给时间带来的影响。

🔧 本页面最近更新：2025/8/23 16:21:13，[更新历史](#)

✎ 发现错误？想一起完善？[在 GitHub 上编辑此页！](#)

👤 本页面贡献者：[Ir1d](#), [HeRaNO](#), [sshwy](#), [H-J-Granger](#), [StudyingFather](#), [countercurrent-time](#), [Enter-tainer](#), [NachtgeistW](#), [OxDkXy](#), [AngelKitty](#), [CCXXI](#), [cjsoft](#), [diauweb](#), [Early0v0](#), [ezoixx130](#), [GekkaSaori](#), [Konano](#), [LovelyBuggies](#), [Makkiy](#), [mgt](#), [minghu6](#), [P-Y-Y](#), [PotassiumWings](#), [SamZhangQingChuan](#), [Suyun514](#), [weiyong1024](#), [Chrogeek](#), [GavinZhengOI](#), [Gesrua](#), [ksyx](#), [kxccc](#), [lychees](#), [Mathias Zhang](#), [mcendu](#), [Nanarikom](#), [ouuan](#), [Peanut-Tang](#), [SukkaW](#), [Tiphereth-A](#), [TSSstudio](#), [代建杉](#)

© 本页面的全部内容可在 [CC BY-SA 4.0](#) 和 [SATA](#) 协议之条款下提供，附加条款亦可能应用