

- [GeoGebra 命令表速查](#)
 - [1. 一般命令](#)
 - [绘图步骤 ConstructionStep](#)
 - [删除 Delete](#)
 - [关系 Relation](#)
 - [2. 代数 \(Algebra\) 或数值](#)
 - [叉积 Cross](#)
 - [点积 Dot](#)
 - [除法 Division](#)
 - [顶点式 CompleteSquare](#)
 - [公分母 CommonDenominator](#)
 - [是否为质数 IsPrime](#)
 - [后一质数 NextPrime](#)
 - [前一质数 PreviousPrime](#)
 - [质因数 PrimeFactors](#)
 - [化简 Simplify](#)
 - [解集 Solutions](#)
 - [近似解 Nsolve](#)
 - [近似解集 NSolutions](#)
 - [复数解集 CSolutions](#)
 - [复数根 ComplexRoot](#)
 - [精确解 Solve](#)
 - [取余 Mod](#)
 - [取整 Div](#)
 - [实数域因式分解 IFactor](#)
 - [因式分解 Factor](#)
 - [因数个数 Divisors](#)
 - [因数列表 DivisorsList](#)
 - [因数和 DivisorsSum](#)
 - [左边 LeftSide](#)
 - [右边 RightSide](#)
 - [展开 Expand](#)
 - [转换为十进制 FromBase](#)
 - [转换进制 ToBase](#)
 - [最大公约数 GCD](#)
 - [最大值 Max](#)
 - [最小值 Min](#)
 - [最小公倍数 LCM](#)
 - [仿射比 AffineRatio](#)
 - [面积 Area](#)
 - [坐标轴 AxisStepX, AxisStepY](#)
 - [二项式系数 BinomialCoefficient](#)
 - [圆周长 Circumference](#)
 - [交比 CrossRatio](#)
 - [曲率 Curvature](#)
 - [距离 Distance](#)

- 轴长 FirstAxisLength, SecondAxisLength
- 积分 Integral
- 迭代 Iteration
- 长度 Length
- 离心率 LinearEccentricity
- 下和 LowerSum
- 上和 UpperSum
- 梯形和 TrapezoidalSum
- 参数 Parameter
- 周长 Perimeter
- 半径 Radius
- 随机数 RandomBetween
- 斜率 Slope
- 3. 角 Angle
- 4. 点 Point
 - 中心点 Center
 - 形心 Centroid
 - 重心 Barycenter
 - 体中心
 - 极值点 Extremum
 - 焦点 Focus
 - 拐点 InflectionPoint
 - 交点 Intersect
 - 中点 Midpoint
 - 点 Point
 - 根 Root
 - 顶点 Vertex
- 5. 线 Line
 - 直线 Line
 - 线段 Segment
 - 射线 Ray
 - 垂线 OrthogonalLine
 - 中垂线 LineBisector
 - 中垂面 PlaneBiSector
 - 垂直平面 OrthogonalPlane
 - 角平分线 AngleBisector
 - 渐近线 Asymptote
 - 系数列表 Coefficients
 - 复数根 ComplexRoot
 - 对称轴 Axes
 - 径 Diameter
 - 准线 Directrix
 - 极线 Polar
 - 切线 Tangent
- 6. 多边形 Polygon
- 7. 向量和矩阵 Vector and Matrices
 - 单位法向量 UnitPerpendicularVector or UnitOrthogonalVector

- 单位矩阵 Identity
- 单位向量 UnitVector
- 法向量 PerpendicularVector or OrthogonalVector
- 简化行梯阵式 ReducedRowEchelonForm
- 矩阵的秩 MatrixRank
- 逆反 Invert
- 维度 Dimension
- 向量 Vector
- 行列式 Determinant
- 应用矩阵 ApplyMatrix
- 转置 Transpose
- 矩阵图 MatrixPlot
- 曲率向量 CurvatureVector
- 方向向量 Direction
- 8. 函数与微积分 Function and Calculus
 - 函数 Function
 - 条件函数 If
 - 次数 Degree
 - 分母 Denominator
 - 导数 Derivative
 - 积分 Integral
 - 多项式 Polynomial
 - 展开 Expand
 - 因式分解 Factor
 - 转换为十进制 FromBase
 - 转换进制 ToBase
 - 化简 Simplify
 - 泰勒展开式 TaylorPolynomial
- 9. 圆锥曲线 Conic
 - 半焦距 LinearEccentricity
 - 半圆 Semicircle
 - 副半轴长 SemiMinorAxisLength (SecondAxisLength)
 - 主半轴长 SemiMajorAxisLength (FirstAxisLength)
 - 主轴 MajorAxis (FirstAxis)
 - 副轴 MinorAxis (SecondAxis)
 - 轴线 Axes
 - 共轭直径 ConjugateDiameter/Diameter
 - 极线 Polar
 - 焦参数 Parameter
 - 焦点 Focus
 - 离心率 Eccentricity
 - 内切圆 Incircle
 - 抛物线 Parabola
 - 双曲线 Hyperbola
 - 椭圆 Ellipse
 - 圆周 Circle
 - 圆锥曲线 Conic

- 中心 Center
- 准线 Directrix
- 密切圆 OsculatingCircle
- 10. 参数曲线 Parameter Curve
 - 曲线 Curve, CurveCartesian
 - 参数曲线相关指令
- 11. 圆弧和扇形 Arc and Sector
 - 弧 Arc
 - 圆弧 CircularArc
 - 扇形 CircularSector
 - 半圆 Semicircle
- 12. 文字
 - 分数文字 FractionText
 - LaTeX文本 LaTeX
 - 字符与Unicode互转 LetterToUnicode
 - 名称 Name
 - 对象 Object
 - 文字 Text
 - 表格文本 TableText
- 13. 轨迹 Locus
- 14. 列表 List
 - 附加 追加 Append
 - 符合条件的数目 CountIf
 - 元素 Element
 - First
 - Last
 - 插入 Insert
 - 交集 Intersection
 - 并集 Union
 - 迭代数列 IterationList
 - 合并 Join
 - 筛选 KeepIf
 - 长度 Length
 - 最小值 Min
 - 最大值 Max
 - 内积 Product
 - 移除未定义对象 RemoveUndefined
 - 排序 Sort
 - 反序 Reverse
 - 求和 Sum
 - 提取 Take
 - 序列 Sequence
- 15. 几何变换 Transformation
 - 位似 Dilate
 - 对称, 镜像, 反射 Reflect、Mirror
 - 旋转 Rotate
 - 平移 Translate

- 切变 Shear
- 伸压、伸缩 Stretch
- 16. 统计
 - 求和 SigmaXX, SigmaXY, SigmaYY
 - 中位数 Median
 - 众数 Mode
 - 平均数 Mean
 - 四分位数 Q1, Q2, Q3
 - 方差 Variance
 - 标准差 SD
 - 相关系数 CorrelationCoefficient
 - 协方差 Covariance
 - 正态分布函数 Normal
 - 反正态函数 InverseNormal
 - 条形图 BarChart
 - 直方图 Histogram
 - 盒形图 BoxPlot
 - 线性回归 FitLine
 - 其他回归
 - 统计量指令 Sxx, Sxy, Syy
- 17. 表格 Table
 - 行序 Row
 - 列序 Column
 - 列名称 ColumnName
 - 单元格区域数值列表 CellRange
 - 单元格 Cell
 - 填充单元格 FillCells
 - 填充列 FillColumn
 - 填充行 FillRow
- 18. 逻辑命令
 - If
 - IsDefined
 - IsInteger
- 19. 优化指令 Optimization
 - 最大值点 Maximize
 - 最小值点 Minimize
- 20. 三维指令 3D
 - 上底 Top
 - 下底 Bottom
 - 侧面 QuadricSide
 - 底面 Ends
 - 高度 Height
 - 棱柱 Prism
 - 棱锥 Pyramid
 - 平面 Plane
 - 垂直平面 OrthogonalPlane
 - 中垂面 PlaneBiSector

- 球面 [Sphere](#)
- 曲面 [Surface](#)
- 体积 [Volume](#)
- 无限长圆柱 [InfiniteCylinder](#)
- 无限长圆锥 [InfiniteCone](#)
- 相交曲线 [IntersectConic](#)
- 圆柱 [Cylinder](#)
- 圆锥 [Cone](#)
- 正四面体 [Tetrahedron](#)
- 正六面体 [Cube](#)
- 正八面体 [Octahedron](#)
- 正十二面体 [Dodecahedron](#)
- 正二十面体 [Icosahedron](#)
- 展开图 [Net](#)

GeoGebra 命令表速查

Version 5.0.579

使用命令可以生成新对象或者修改已有对象。

当在 GeoGebra 的命令框中输入命令，软件会尝试着自动补齐命令，也就是说，在命令框中只要输入命令的前两个字符，GeoGebra 会显示最相近的命令（类似于输入法的联想功能）：

如果提示的命令刚好是想输入的，只需按下 Enter 键，即可接受建议将提示的命令输入命令框；

如果提示的命令并不是想输入的，可以继续输入，GeoGebra 会再提示其他相近的命令。

以下命令速查表基于：

- GeoGebra Classic 5.0.579.0-d
- 唐家军 《GeoGebra指令汇编-唐家军.pdf》

说明：由于版本不同，中文翻译有很多不一样，以5.0.579的中文翻译为准。

版本	日期	参与人员
V 5.0.579	2020-04-28	

1. 一般命令

绘图步骤 [ConstructionStep](#)

[ConstructionStep\[\]](#)：返回目前的步骤数

[ConstructionStep\[Object\]](#)：返回指定对象目前的步骤数

删除 [Delete](#)

[Delete\[Object\]](#)：删除某一对象以及所有与之相关的对象

关系 Relation

Relation[Object a, Object b]: 显示一个窗口让我们得知 Object a 和 Object b 之间的关系。

如：该命令可以知道是否一点在一条直线上（或一条圆锥曲线上）、一直线是与一圆锥曲线相切、一直线是否与一圆锥曲线相交或两对象是否相等。

2. 代数（Algebra）或数值

叉积 Cross

叉积(<向量1>, <向量2>): 得到一个右手系的向量

点积 Dot

点积(<向量1>, <向量2>): 得到一个数值

除法 Division

除法[<被除数>,<除数>]: 给出两个数值的商（得数结果的整数部分）以及余数。

如：除法[16,3]得出{5,1}。

除法[<被除式>,<除式>]: 给出两个多项式的商式以及余式，都取整式。

如：除法[x^2+3x+1 , $x-1$]得出{ $x+4$,5}。

除法(<被除数 @整数>, <除数 @整数>) 除法(<被除式 @整式>, <除式 @整式>)

顶点式 CompleteSquare

顶点式[<二次函数>]: 返回二次函数的完全平方形式: $a(x-h)^2 + k$ 。

如：CompleteSquare[x^2-4x+7] 得出 $1(x-2)^2 + 3$ 。

原来叫“配方式”

公分母 CommonDenominator

公分母[<分式 1>,<分式 2>]: 返回两个表达式的最小公分母方程式函数。

如：CommonDenominator[$3/(2x+1)$, $3/(4x^2+4x+1)$] 得出 $f(x) = 4x^2 + 4x + 1$ 。

是否为质数 IsPrime

IsPrime[<数值>]: 根据数值是否为质数得出 true 或 false。

如：IsPrime[10] 得出 false，IsPrime[11]得出 true。

是否为质数(<数值>)

后一质数 NextPrime

后一质数[<数值>]: 返回比输入数值更大的最小质数。

如：后一质数[10000]得出 10007。

前一质数 PreviousPrime

前一质数[<数值>]: 返回比输入数值更小的最大质数。

如: 前一质数[10000]得出 9973。

质因数 PrimeFactors

质因数[<数值>]: 返回质数集合, 其中元素的乘积等于指定数值。

如: PrimeFactors[1024] 得出 {2,2,2,2,2,2,2,2};

质因数[42] 得出 {2,3,7}。

化简 Simplify

Simplify[Function]: 化简给定的函数, 尽可能简化所给函数的项(合并同类项)。如:

- Simplify[x + x + x]得到 $f(x) = 3x$
- Simplify[sin(x) / cos(x)]得到 $f(x) = \tan(x)$
- Simplify[-2 sin(x) cos(x)]得到 $f(x) = \sin(-2x)$

解集 Solutions

解集(<方程>): 给出方程的解集

如: 解集($x^2 - 2x + 1 = 0$): {1}

运算:

解集(<方程>)

解集(<方程>, <变量>)

解集(<方程列表>, <变量列表>)

近似解 Nsolve

近似解(<方程>)

如: 近似解($\ln(x) = 2$): {x=7.3891}

运算:

近似解(<方程>)

近似解(<方程>, <变量>)

近似解(<方程>, <变量 = 初值>)

近似解(<方程列表>, <变量列表>)

近似解集 NSolutions

近似解集(<方程>)

如: NSolutions($\ln(x^2)=2$): {-2.7183,2.7183}

运算:

近似解集(<方程>)

近似解集(<方程>, <变量>)

近似解集(<方程>, <变量 = 初值>)

近似解集(<方程列表>, <变量列表>)

复数解集 CSolutions

等待开发

复数解集(<方程>)

如: $\text{CSolutions}(x^2+1=0)$: $\{-i,i\}$

复数根 ComplexRoot

复数根(<多项式>): 计算多项式在 x 上的复数根, 绘图区创建对应的点。

如: 复数根 $[x^2+4]$ 得出两个复数根: $(0+2i)$ 和 $(0-2i)$ 。

精确解 Solve

精确解(<方程>)

如: $\text{solve}(x^2-4=0)$: $\{x=-2,x=2\}$

运算:

精确解(<关于 x 的方程>)

精确解(<方程>, <变量>)

精确解(<方程列表>, <变量列表>)

精确解(<参数方程列表>, <变量列表>)

精确解(<方程>, <变量>, <假设列表>)

取余 Mod

$\text{Mod}[\text{Integer } a, \text{Integer } b]$: 计算 a 除以 b 的余数。

取余(<被除数 @整数>, <除数 @整数>)

取余(<被除式 @整式>, <除式 @整式>)

取整 Div

$\text{Div}[\text{Number } a, \text{Number } b]$: 计算数字 a 除以 b 的商(取整数)。返回两个数值的整数商 (得数结果的整数部分)。

如: $\text{Div}[16,3]$ 得出 5。

取整(<被除数 @整数>, <除数 @整数>) 取整(<被除式 @整式>, <除式 @整式>): 返回两个多项式的商。如: 取整 $[x^2+3x+1, x-1]$ 得出 $f(x)=x+4$ 。

实数域因式分解 IFactor

实数域因式分解(<多项式>) 如: 实数域因式分解($x^2 - 2$) 得到 $(x - \sqrt{2})(x + \sqrt{2})$

运算:

实数域因式分解(<表达式>)

实数域因式分解(<表达式>, <变量>)

因式分解 Factor

$\text{Factor}[\text{Polynomial}]$: 将多项式转换成因式乘法形式。这个指令在有理数上因式分解表达式。

若要在复数上进行因式分解, 参见复数因式分解指令。

如：因式分解[x^2+x-6] 得出 $(x+3)(x-2)$ 。

运算:

因式分解(<整数>)

因式分解(<多项式>)

因式分解(<表达式>, <变量>)

因数个数 Divisors

因数个数(<正整数>): 计算全部正因数的数量, 包括指定数值本身。

如：因数个数[15] 得出 4, 15 全部正因数的数量, 包括 15。

因数列表 DivisorsList

因数列表(<正整数>): 给出全部正因数的集合, 包括指定数值本身。

如：因数列表[15] 得出 {1,3,5,15}, 15 的全部正因数集合, 包括 15。

因数和 DivisorsSum

因数和[<正整数>]: 计算全部的正因数之和, 包括指定数值本身。

如：因数和[15] 得出 24, $1+3+5+15$ 之和。

左边 LeftSide

左边[<方程>]: 给出简化方程的左边。

如：左边[$x+2=3x+1$]得出 x 。

运算:

左边(<方程>)

左边(<方程组列表>)

左边(<方程组列表>, <列表索引>)

右边 RightSide

右边[<方程>]: 给出简化方程的右边。 如：右边[$x+2=3x+1$]得出 0.5。

运算:

右边(<方程>)

右边(<方程组列表>)

右边(<方程组列表>, <列表索引>)

展开 Expand

Expand[Function]: 将式子按乘法展开。

展开[<表达式>]: 展开表达式。

如：展开[$(2x-1)^2+2x+3$] 得出 $4x^2-2x+4$ 。

转换为十进制 FromBase

转换为十进制["<数值作为文本>",<2-36 间的底数>]: 以给定的底数（进制）将指定数值转换为十进制。底数必须在 2 与 36 之间。数值必须是一个整数。

如: 转换为十进制["FF",16] 返回 255。

转换为十进制["100000000",2] 返回 256。

转换为十进制("<指定进制型数值>",<进制(基数) 2~36>)

转换进制 ToBase

转换进制[<整数>,<进制数_2-36>]: 将指定数值转化为不同的进制（底数），在 2 与 36 之间，数值必须是一整数。参见十进制指令。

如: 转换进制[255,16]="FF";

转换进制[256,2]="100000000"。

以前叫“指定进制”

最大公约数 GCD

GCD[Number a, Number b]: 计算两个数 a 和 b 的最大公因数。

GCD[List of numbers]: 计算整数集合中所有元素的最大公因数。

最大公约数(<整数列表>)

最大公约数(<整数1>, <整数2>)

运算:

最大公约数(<整数列表>)

最大公约数(<多项式列表>): 计算多项式集合的最大公约因式。

如: 最大公约数[{ x^2+4x+4 }, { x^2-x-6 }, { $x^3-4x^2-3x+18$ }]得出 $x+2$ 。

最大公约数(<整数1>, <整数2>) 最大公约数(<多项式1>, <多项式2>): 计算两个多项式的最大公约因式。

如: 最大公约数[x^2+4x+4 , x^2-x-6]得出 $x+2$ 。

最大值 Max

最大值(<区间 如: $2 < x < 3$): 求指定区间的最大值 最大值(<数值列表>): 求指定列表的最大值

最大值(<数值1>, <数值2>): 求 a 和 b 中的大者。 最大值(<数据列表>, <频数列表>) 最大值(<函数>, <x-起始值>, <x-终止值>): 求函数在指定定义域中的最大值，返回一个点。

Max(x^2 ,0,1): 得到点(1,1)

运算: 最大值(<数值列表>) 最大值(<数值1>, <数值2>)

最小值 Min

最小值(<区间 如: $2 < x < 3$): 求指定区间的最小值 最小值(<数值列表>): 求指定列表的最小值

最小值(<数值1>, <数值2>): 求 a 和 b 中的小者。 最小值(<数据列表>, <频数列表>) 最小值(<函数>, <x-起始值>, <x-终止值>): 求函数在指定定义域中的最小值，返回一个点。

Max(x^2 ,0,1): 得到点(0,0)

运算: 最小值(<数值列表>)

最小值(<数值1>, <数值2>)

最小公倍数 LCM

最小公倍数(<整数1>, <整数2>) 或 LCM[Number a, Number b]: 计算两个数的最小公倍数。

如: LCM[12,15] 得出 60。

最小公倍数(<整数列表>) 或 LCM[List of numbers]: 计算数列的最小公倍数。 如:

LCM[{12,30,18}] 得出 180。

最小公倍数[<多项式 1>,<多项式 2>]: 计算两个多项式的最小公倍数。

如: LCM[x^2+4x+4 , x^2-x-6] 得出 $x^3 + x^2 - 8x - 12$ 。 最小公倍数[<多项式集合>]: 计算集合中所有多项式的最小公倍数。

如: LCM[{ x^2+4x+4 , x^2-x-6 , $x^3-4x^2-3x+18$ }] 得出 $x^4 - 2x^3 - 11x^2 + 12x + 36$ 。

运算:

最小公倍数(<整数列表>)

最小公倍数(<多项式列表>)

最小公倍数(<整数1>, <整数2>)

最小公倍数(<多项式1>, <多项式2>)

仿射比 AffineRatio

AffineRatio[Point A, Point B, Point C]: 得出共线的三点 A, B, C 的比值 λ , 其中 $C = A + \lambda * AB$ 。

面积 Area

Area[Point A, Point B, Point C, ...]: 计算点 A, B, C, ... 所围成的多边形面积。

Area[Conic c]: 计算圆锥曲线 c 的面积。

坐标轴 AxisStepX, AxisStepY

AxisStepX[]: 得到当前 X 轴单位和数值。

AxisStepY[]: 得到当前 Y 轴单位和数值。

二项式系数 BinomialCoefficient

BinomialCoefficient[Number n, Number r]: 计算“n 选 r”的二项式系数 $\binom{n}{r} = \frac{n!}{r!(n-r)!}$ 。

圆周长 Circumference

Circumference[Conic]: 计算圆锥曲线的周长。

交比 CrossRatio

CrossRatio[Point A, Point B, Point C, Point D]: 计算共线的四点 A, B, C, D 的交比为 λ 。 $\lambda = \text{AffineRatio}[B, C, D] / \text{AffineRatio}[A, C, D]$ 。

曲率 Curvature

Curvature[Point, Function]: 计算函数在指定点的曲率。如: 曲率[(0,0), x^2]得出 2。

Curvature[Point, Curve]: 计算曲线在指定点的曲率。

曲率(<点>, <对象>)

距离 Distance

Distance[Point A, Point B]: 计算两点 A 和 B 之间的距离。

Distance[Point, Line]: 计算点和线之间的距离。

Distance[Line g, Line h]: 计算两直线 g 和 h 之间的距离。

轴长 FirstAxisLength, SecondAxisLength

FirstAxisLength[Conic]: 计算圆锥曲线主轴的长度。如: 椭圆中是长轴的一半。

SecondAxisLength[Conic]: 计算圆锥曲线副轴的长度。如: 椭圆中是短轴的一半。

积分 Integral

Integral[Function, Number a, Number b]: 计算某函数在 $[a, b]$ 的积分。

Integral[Function f, Function g, Number a, Number b]: 计算在区间 $[a, b]$ 上, $f(x) - g(x)$ 的积分。

迭代 Iteration

Iteration[Function, Number x0, Number n]: 给定起始值 x0, 重复带入 f 函数 n 次。如: 定义 $f(x) = x^2$, 输入命令 **Iteration[f, 3, 2]** 就可以得到结果 81。

长度 Length

Length[Vector]: 计算向量的长度。

Length[Point A]: 计算原点 (0, 0) 到点 A 的长度。

Length[Function, Number x1, Number x2]: 计算函数 f 从 x1 到 x2 之间的长度。

Length[Function, Point A, Point B]: 计算函数 f 从点 A 到点 B 之间的长度。如果给定的点不在函数图像上, 那么将以点的 x 坐标作为区间。

Length[Curve, Number t1, Number t2]: 计算曲线在参数 t1 到 t2 之间的长度。

Length[Curve c, Point A, Point B]: 计算曲线 c 在点 A 和 B 之间的曲线长度。

Length[List]: 计算列表的长度 (列表的元素个数)。

离心率 LinearEccentricity

LinearEccentricity[Conic]: 计算圆锥曲线的离心率。

下和 LowerSum

LowerSum[Function, Number a, Number b, Number n]: 函数 f 在 $[a, b]$ 区间以 n 个长条区求出的面积下和。

註: 此指令會同時繪出這些下和的長方形。

上和 UpperSum

UpperSum[Function, Number a, Number b, Number n]: 計算函數 f 在 $[a,b]$ 區間以 n 個長條區求出的面積上和。

註: 此指令亦可繪出這些上和的長方形。

梯形和 TrapezoidalSum

TrapezoidalSum[Function, Number a, Number b, Number n of trapezoids]: 計算函數 f 在區間 $[a, b]$ 分成 n 個梯形的總和。

註: 此指令會同時畫出函數下方的梯形。

参数 Parameter

Parameter[Parabola]: 計算拋物線 p 的参数（准线和焦点间的距离）。

周长 Perimeter

Perimeter[Polygon]: 計算多邊形周长。周长(<多邊形>)

周长(<圓錐曲線>)

周长(<軌迹>)

半径 Radius

Radius[Circle]: 計算圓 c 的半径。

随机数 RandomBetween

RandomBetween[Min integer, Max integer]: 在最大值和最小值直接产生随机数。

RandomBinomial[Number n of trials, Probability p]: 使用二项分布产生随机数。

RandomNormal[Mean, Standard deviation]: 使用正态分布产生随机数。

RandomPoisson[Mean]: 使用泊松分布产生随机数。

斜率 Slope

Slope[Line]: 計算给定直线的斜率。

3. 角 Angle

Angle[Vector v1, Vector v2]: 計算两向量 $v1$ 和 $v2$ 之间的夹角。

Angle[Line g, Line h]: 計算两直线 g 和 h 方向向量之间的夹角。

Angle[Point A, Point B, Point C]: 以 B 为顶点, 計算 BA 和 BC 之间的夹角。

Angle[Point A, Point B, Angle α]: 以线段 AB 为始边, 画出大小为 α 的角。

Angle[Conic]: 計算圓錐曲線 c 主軸的转角。

Angle[Vector]: 計算 x 轴到向量 v 之间的夹角。

Angle[Point]: 計算 x 轴到点 A 之间的夹角。

Angle[Number]: 将一个数字 n 转换成弧度（介于 0 到 2π 之间）。

Angle[Polygon]: 建立多邊形 p 的所有内角。如果多邊形是以逆时针方向选取, 可以得到内角和。如果多邊形是顺时针方向选取, 可以得到外角和。

角度(<几何对象 圆锥曲线|向量|点|数值|多边形>)

角度(<向量1>, <向量2>)

角度(<直线1>, <直线2>)

角度(<直线>, <平面>)

角度(<平面1>, <平面2>)

角度(<点>, <顶点>, <点>)

角度(<点>, <顶点>, <度|弧度>)

角度(<点1>, <点2>, <点3>, <方向>)

4. 点 Point

中心点 Center

Center[Conic]: 计算圆锥曲线的中心点。

形心 Centroid

Centroid[Polygon]: 计算多边形 p 的重心。

重心 Barycenter

Barycenter(点列, 权重列表): 计算点列的重心。

重心(<点列>, <权重列表>)

体中心

等待实现

极值点 Extremum

Extremum[Polynomial]: 计算多项式函数图像上的局部极值。

焦点 Focus

Focus[Conic]: 得到圆锥曲线 c 的焦点。

拐点 InflectionPoint

InflectionPoint[Polynomial]: 计算多项式 f 的所有拐点。

交点 Intersect

Intersect[Line g, Line h]: 计算直线 g 和 h 的交点。

Intersect[Line, Conic]: 计算直线 g 和圆锥曲线 c 的所有交点。

Intersect[Line, Conic, Number n]: 计算直线 g 和圆锥曲线 c 的第 n 个交点。

Intersect[Conic c1, Conic c2]: 计算两圆锥曲线 c 和 d 的所有交点。

Intersect[Conic c1, Conic c2, Number n]: 计算两圆锥曲线 c 和 d 的第 n 个交点。

Intersect[Polynomial f1, Polynomial f2]: 计算多项式 f1 和多项式 f2 的所有交点。

Intersect[Polynomial f1, Polynomial f2, Number n]: 计算多项式 f1 和多项式 f2 的第 n

个交点。

`Intersect[Polynomial, Line]`: 计算多项式 f 和直线 g 的所有交点。

`Intersect[Polynomial, Line, Number n]`: 计算多项式 f 和直线 g 的第 n 个交点。

`Intersect[Function f, Function g, Point A]`: 计算函数 f 和 g 在起始点 A 的所有交点。

`Intersect[Function, Line, Point A]`: 计算函数 f 和直线 g 在起始点 A 的所有交点。

中点 Midpoint

`Midpoint[Point A, Point B]`: 计算点 A 和 B 的中点。

`Midpoint[Segment]`: 计算线段 s 的中点。

点 Point

`Point[Line]`: 得到直线上一点。

`Point[Conic]`: 得到圆锥曲线上一点。

`Point[Function]`: 得到函数 f 上一点。

`Point[Polygon]`: 得到多边形 p 上一点。

`Point[Vector]`: 得到向量 v 上一点。

`Point[Point, Vector]`: 得到从 P 点平移向量 v 之后的点。

根 Root

`Root[Polynomial]`: 得到多项式的所有根。

`Root[Function, Number a]`: 得到函数 f 在起始值 a 的一个根。

`Root[Function, Number a, Number b]`: 得到函数 f 在区间 $[a, b]$ 上的根。

顶点 Vertex

`Vertex[Conic]`: 生成圆锥曲线的全部顶点。

5. 线 Line

直线 Line

`Line[Point A, Point B]`: 建立过点 A 和 B 的直线。

`Line[Point, Line]`: 建立过点 A 且平行于直线 g 的直线。

`Line[Point, Vector v]`: 建立过点 A 且方向为 v 的直线。

线段 Segment

`Segment[Point A, Point B]`: 建立点 A 和点 B 之间的线段。

`Segment[Point A, Number a]`: 建立以点 A 为起点, 线段长为 a 的线段。

射线 Ray

`Ray[Point A, Point B]`: 建立起点为 A 过 B 点的射线。

`Ray[Point, Vector v]`: 建立起点为 A 且方向向量为 v 的射线。

垂线 OrthogonalLine

OrthogonalLine[Point, Line]: 建立过点 A 且垂直于直线 g 的直线。

OrthogonalLine[Point, Vector]: 建立过点 A 且垂直于向量 v 的直线。

垂线(<点>, <直线>)

垂线(<点>, <线段>)

垂线(<点>, <向量>)

垂线(<点>, <平面>)

垂线(<直线1>, <直线2>)

垂线(<点>, <向量1>, <向量2>)

垂线(<点>, <直线>, <平面xOy | 3D 空间>)

中垂线 LineBisector

LineBisector[Point A, Point B]: 建立线段 AB 的垂直平分线。

LineBisector[Segment]: 建立线段的垂直平分线。

中垂线(<线段>)

中垂线(<点1>, <点2>)

中垂线(<点1>, <点2>, <方向向量>)

中垂面 PlaneBiSector

PlaneBiSector(Segment)

PlaneBiSector(Point A, Point B)

中垂面(<线段>)

中垂面(<点1>, <点2>)

垂直平面 OrthogonalPlane

OrthogonalPlane(Point, Line)

OrthogonalPlane(Point, Vector)

垂直平面(<点>, <直线>)

垂直平面(<点>, <向量>)

角平分线 AngleBisector

AngleBisector[Point A, Point B, Point C]: 生成角 ABC 的角平分线。

AngleBisector[Line g, Line h]: 生成直线 g 和 h 的角平分线。角平分线(<直线1>, <直线2>)

角平分线(<点1>, <顶点2>, <点3>)

渐近线 Asymptote

Asymptote[Hyperbola]: 建立圆锥（二次）曲线的全部渐近线。

如：渐近线[$x^2 - y^2/4 = 1$] 返回直线 $-2x + y = 0$ 和直线 $-2x - y = 0$ 。

渐近线(<双曲线>): 得出圆锥（二次）曲线（双曲线）的全部渐近线。

渐近线(<函数>): 创建函数的渐近线, 并将它们返回在一个集合中。有可能不会全部找出来, 例如类似于 $\ln(x)$ 这样的非-有理函数的垂直渐近线。

如 **asymptote(($x^3 - 2x^2 - x + 4$)/($2x^2 - 2$))** 返回集合对象 $\{y = 0.5x - 1, x = 1, x = -1\}$ 。

渐近线(<隐式曲线>): 得出一个包含隐式曲线(高次曲线)全部渐近线的集合。

如: 渐近线 $[x^3+y^3+y^2-3x=0]$ 返回集合对象 $\{x+y=-0.33\}$ 。

系数列表 Coefficients

(1)系数列表[<多项式>]: 得出多项式 $a_k x^k + a_{k-1} x^{k-1} + \dots + a_1 x + a_0$ 全部系数的集合 $\{a_k, a_{k-1}, \dots, a_1, a_0\}$ 。如: 系数列表 $[x^3-3x^2+3x]$ 得出 $\{1, -3, 3, 0\}$, x^3-3x^2+3x 所有系数的集合。

(2)系数列表[<圆锥曲线>]: 返回列表 $\{a, b, c, d, e, f\}$, 其由圆锥(二次)曲线的标准形式得出: $ax^2 + by^2 + c + dxy + ex + fy = 0$ 。

给定(隐式直线) $l: 3x + 2y - 2 = 0$, 则 $x(l)$ 返回 3, $y(l)$ 返回 2, 以及 $z(l)$ 返回-2。

对于由 $l: ax + by + c = 0$ 所代表一条隐式直线, 可以使用语句 $x(l)$ 、 $y(l)$ 、 $z(l)$ 来获得它的系数。

复数根 ComplexRoot

复数根[<多项式>]: 计算多项式在 x 上的复数根, 绘图区创建对应的点。

如: 复数根 $[x^2+4]$ 得出 $(0+2i)$ 和 $(0-2i)$ 。

对称轴 Axes

Axes[Conic]: 建立圆锥曲线 c 的对称轴。

径 Diameter

Diameter[Line, Conic]: 圆锥曲线 c 平行于直线 g 的径。

Diameter[Vector, Conic]: 圆锥曲线 c 平行于向量 v 的径。

准线 Directrix

准线[<抛物线>]

Directrix[Parabola]: 建立抛物线 p 的准线。

得出圆锥(二次)曲线(抛物线)的准线。准线 $[x^2-3x+3y=9]$ 得出直线 $y=4.5$ 。

极线 Polar

Polar[Point, Conic]: 建立相对于圆锥曲线 c 的极线。

切线 Tangent

Tangent[Point, Conic]: 建立圆锥曲线 c 过点 A 的所有切线。

Tangent[Line, Conic]: 建立圆锥曲线 c 的所有平行于直线 g 的切线。

Tangent[Number a, Function]: 建立 f(x)在 x=a 的切线。

Tangent[Point A, Function]: 建立 f(x)在 $x=x(A)$ 的切线。

6. 多边形 Polygon

Polygon[Point A, Point B, Point C,...]: 生成由给定点 A, B, C.....所围成的多边形。

Polygon[Point A, Point B, Number n]: 以线段 AB 为边长, 生成由 n 个顶点的正多边形。

多边形(<点列>)

多边形(<点1>, ..., <点n>)

多边形(<点1>, <点2>, <顶点数>)

多边形(<点1>, <点2>, <顶点数>, <方向向量>)

7. 向量和矩阵 Vector and Matrices

单位法向量 UnitPerpendicularVector or UnitOrthogonalVector

UnitPerpendicularVector[Line]: 求直线 g 的单位法向量。

UnitPerpendicularVector[Vector]: 求向量 v 的单位法向量

单位法向量(<直线|射线>)

单位法向量(<线段>)

单位法向量(<向量>)

单位法向量(<平面>)

运算:

单位法向量(<向量>)

单位矩阵 Identity

单位矩阵(<数值>)

单位向量 UnitVector

UnitVector[Line]: 直线 g 的单位方向向量。

UnitVector[Vector]: 与向量 v 同方向的单位向量。

单位向量(<几何对象>)

运算:

单位向量(<向量>)

法向量 PerpendicularVector or OrthogonalVector

PerpendicularVector[Line]: 求直线 g 的法向量。如: 直线 $ax + by = c$ 的法向量为 (a, b) 。

PerpendicularVector[Vector v]: 求向量 v 的法向量。

法向量(<直线>)

法向量(<线段>)

法向量(<向量>)

法向量(<平面>)

运算:

法向量(<向量>)

简化行梯阵式 ReducedRowEchelonForm

简化行梯阵式(<矩阵>)

矩阵的秩 MatrixRank

矩阵的秩(<矩阵>)

逆反 Invert

Invert[Matrix]: 返回 逆矩阵

例: **Invert**[[{1, 2}, {3, 4}]] 得到逆矩阵 {{-2, 1}, {1.5, -0.5}}.

逆反(<矩阵>)

逆反(<函数>)

维度 Dimension

维度(<点|向量|矩阵>)

向量 Vector

Vector[Point A, Point B]: 从点 A 到点 B 的向量。

Vector[Point]: 点 A 的位置向量，即起点为原点，终点为指定点。

向量(<终点(原点为起点)>)

向量(<起点>, <终点>)

行列式 Determinant

Determinant[Matrix]: 得到方阵的行列式值

例: **Determinant**[[{1, 2}, {3, 4}]] 得到数值.

应用矩阵 ApplyMatrix

应用矩阵(<矩阵>, <对象>)

转置 Transpose

Transpose[Matrix]: 矩阵转置。

例: **Transpose**[[{1, 2}, {3, 4}]] 可以得到转置矩阵 {{1, 3}, {2, 4}}.

转置(<矩阵>)

矩阵图 MatrixPlot

等待开发

曲率向量 CurvatureVector

CurvatureVector[Point, Function]: 求函数 f 在点 A 的曲率向量。如: 曲率向量[(0,0),x^2] 得出向量(0,2)。

CurvatureVector[Point, Curve]: 求曲线 c 在点 A 的曲率向量。

曲率向量(<点>, <对象>)

方向向量 Direction

Direction[Line]: 直线的方向向量

注意: 直线方程式 $ax + by = c$ 的方向向量为 $(b, -a)$ 。

方向向量(<直线|射线|线段>)

8. 函数与微积分 Function and Calculus

函数 Function

Function[Function, Number a, Number b]: 生成函数 f 在 $[a, b]$ 区间上函数图像。如:
 $f(x) = \text{Function}[x^2, -1, 1]$ 得到函数 x^2 在区间 $[-1, 1]$ 上的函数图像。如果输入 $g(x) = 2f(x)$, 将会得到 $g(x) = 2x^2$, 但这个函数的定义域并不限制在 $[-1, 1]$ 区间上。

条件函数 If

如果要建立条件函数可以使用布朗命令 **If**。如:

$$f(x) = \text{If}[x < 3, \sin(x), x^2] \text{ 相当于: } f(x) = \begin{cases} \sin(x), & \text{当 } x < 3 \\ x^2, & \text{当 } x \geq 3 \end{cases}$$

次数 Degree

次数[<多项式>]: 给出一个多项式主变量的最高次数。

如: 次数 $[x^4 + 2x^2]$ 得出 4。

运算区指令

(1) 次数[<多项式>];

(2) 次数[<多项式>, <变量>]: 给出一个多项式中指定变量的最高次数。

如: 次数 $[x^4y^3 + 2x^2y^3, x]$ 得出 4。 次数 $[x^4y^3 + 2x^2y^3, y]$ 得出 3。

分母 Denominator

(1) 分母[<函数>]: 返回一个函数的分母。 如: 分母 $[5/(x^2+2)]$ 得出 $f(x) = (x^2+2)$ 。

(2) 分母[<表达式>]: 返回一个有理数或表达式的分母。 如: 分母 $[2/3 + 1/15]$ 得出 15

导数 Derivative

Derivative[Function]: 求函数 $f(x)$ 的导数 (微分) $f'(x)$ 。

Derivative[Function, Number n]: 求函数 $f(x)$ 的 n 阶导数 (n 次微分) $\frac{d^n f}{dx^n}$ 。

导数(<函数>): 返回函数关于主变量(x)的导数。可以用 $f'(x)$ 代替导数[f], 或 $f''(x)$ 代替导数[f,2], 并以此类推。 如: 导数 $[x^3 + x^2 + x]$ 得出 $3x^2 + 2x + 1$ 。

导数(<曲线>): 返回曲线的微分, 只能用于参数曲线。 如: 导数[曲线[cos(t), tsin(t), t, 0, π]] 得出曲线 $x = -\sin(t), y = \sin(t) + t\cos(t)$ 。

导数(<函数>, <阶数>): 返回函数关于主变量(x)的 n 阶导数 (n 次微分)。 如: 导数 $[x^3 + x^2 + x, 2]$ 得出 $6x + 2$ 。

导数(<函数>, <变量>): 返回函数关于指定变量的偏导数。 如: 导数 $[x^3y^2 + y^2 + xy, y]$ 得出 $2x^3y + x + 2y$

导数(<曲线>, <阶数>): 返回曲线的 n 次微分 (n 阶导数), 只能用于参数曲线。 如: 导数[曲线[cos(t), tsin(t), t, 0, π], 2] 得出曲线 $x = -\cos(t), y = 2\cos(t) - tsin(t)$ 。

导数(<函数>, <变量>, <阶数>): 返回函数关于指定变量的 n 阶偏导数。如: 导数 $[x^3+3xy,x,2]$ 得出 $6x$ 。

运算:

导数(<表达式>)

导数(<表达式>, <变量>)

导数(<表达式>, <变量>, <阶数>)

积分 Integral

Integral[Function]: 求函数 $f(x)$ 的不定积分。

多项式 Polynomial

Polynomial[Function]: 求函数 f 的展开式。

Polynomial[List of n points]: 建立经过 n 个点的 $n-1$ 次多项式。

展开 Expand

Expand[Function]: 将式子按乘法展开。

展开[<表达式>]: 展开表达式。

如: 展开 $[(2x-1)^2+2x+3]$ 得出 $4x^2-2x+4$ 。

因式分解 Factor

Factor[Polynomial]: 将多项式转换成因式乘法形式。这个指令在有理数上因式分解表达式。

若要在复数上进行因式分解, 参见复数因式分解指令。

如: 因式分解 $[x^2+x-6]$ 得出 $(x+3)(x-2)$ 。

转换为十进制 FromBase

转换为十进制["<数值作为文本>",<2-36 间的底数>]: 以给定的底数(进制)将指定数值转换为十进制。底数必须在 2 与 36 之间。数值必须是一个整数。

如: 转换为十进制["FF",16] 返回 255。

转换为十进制["100000000",2] 返回 256。

转换为十进制("<指定进制型数值>",<进制(基数) 2~36>)

转换进制 ToBase

转换进制[<整数>,<进制数_2-36>]: 将指定数值转化为不同的进制(底数), 在 2 与 36 之间, 数值必须是一整数。参见十进制指令。

如: 转换进制[255,16]="FF";

转换进制[256,2]="100000000"。

以前叫“指定进制”

化简 Simplify

Simplify[Function]: 化简给定的函数, 尽可能简化所给函数的项(合并同类项)。如:

- Simplify[$x + x + x$]得到 $f(x) = 3x$
- Simplify[sin(x) / cos(x)]得到 $f(x) = \tan(x)$
- Simplify[-2 sin(x) cos(x)]得到 $f(x) = \sin(-2x)$

泰勒展开式 TaylorPolynomial

TaylorPolynomial[Function, Number a, Number n]: 建立函数 $f(x)$ 在点 $x = a$ 的 n 次泰勒展开式。

9. 圆锥曲线 Conic

半焦距 LinearEccentricity

半焦距[<圆锥曲线>]: 计算平面圆锥（二次）曲线的偏离率（半焦距，线型离心率），即圆锥曲线的中心与其焦点（或它两个焦点中的一个）之间的距离。

半焦距[$4x^2 - y^2 + 16x + 20 = 0$] 返回 2.24。

半圆 Semicircle

半圆[<点 1>, <点 2>]: 以给定两点为直径新建一个半圆弧线（逆时针方向）。

半圆（半圆弧线过两点）工具：选定两个点 A 和 B，创建一条在线段（或间隔）AB 之上的半圆弧线。

副半轴长 SemiMinorAxisLength (SecondAxisLength)

副半轴长(<圆锥曲线>): 返回平面圆锥（二次）曲线半短轴的长度（短轴的一半）。如：SemiMinorAxisLength[$x^2 + 2y^2 - 2x - 4y = 5$] 得出 2。

主半轴长 SemiMajorAxisLength (FirstAxisLength)

主半轴长(<圆锥曲线>): 返回平面圆锥（二次）曲线半长轴的长度（长轴的一半）。如：SemiMajorAxisLength[$(x-1)^2 + (y-2)^2 = 4$] 得出 2。

主轴 MajorAxis (FirstAxis)

主轴[<圆锥曲线>]: 返回一条圆锥曲线的长轴直线方程。如：MajorAxis[$x^2/9 + y^2/4 = 1$] 返回 $y=0$ 。

副轴 MinorAxis (SecondAxis)

副轴[<圆锥曲线>]: 返回一条圆锥曲线的短轴直线方程。如：MinorAxis[$x^2/9 + y^2/4 = 1$] 返回 $x=0$ 。

轴线 Axes

轴线(<圆锥曲线>): 返回一条圆锥曲线的长轴和短轴的直线方程。参见“长轴”和“短轴”指令。
如：轴线[$x^2 + 4y^2 + 2x - 8y + 1 = 0$] 返回直线 $a:y=1$ 和直线 $b:x=-1$ 。
轴线(<二次曲面>):

共轭直径 ConjugateDiameter/Diameter

共轭直径(<向量>, <圆锥曲线>): 创建已知向量相对于圆锥曲线的共轭直径。

如: 设 $u=(4,1)$ 为一个向量。共轭直径 $[u, x^2+4y^2+2x-8y+1=0]$ 得出直线 $x+y=0$ 。

共轭直径(<直线>, <圆锥曲线>): 创建平面圆锥曲线平行于指定直线的共轭直径。

如: 共轭直径 $[-4x+5y=-2, x^2+4y^2+2x-8y+1=0]$ 得出直线 $5x+16y=11$ 。

释义: 连结椭圆上任意两点的线段叫弦。过椭圆中心的弦叫直径, 平行于直径 DE 的弦的中点的轨迹 AB 和直径 DE 互为共轭直径。类似地可定义双曲线的共轭直径。由于上述 DE 直径是任意取的, 因此椭圆的共轭直径有无数对。当一对共轭直径互相垂直时, 即为椭圆的长轴和短轴。

极线 Polar

极线[<点>,<圆锥曲线>]: 新建给定点关于平面圆锥(二次)曲线的极线。如: 极线 $[(0,2), y=x^2-3x+5]$ 新建直线 $1.5x+0.5y=4$ 。

极线(<直线>, <圆锥曲线>)

释义: 如果圆锥曲线(适用于圆)切于 A、B 两点的切线相交于 P 点, 那么 P 点称为直线 AB 关于该曲线的极点(pole), 直线 AB 称为 P 点的极线(polar)。P 点在圆锥曲线内部时也可以定义极线, 可以认为极线是过 P 点做此圆锥曲线两条虚切线切点的连线。

焦参数 Parameter

焦参数(<抛物线>): 返回抛物线的参数(焦距), 即准线与焦点的距离。如: 焦参数 $[y=x^2-3x+5]$ 返回 0.5。

焦点 Focus

焦点[<圆锥曲线>]: 得出平面圆锥(二次)曲线的(全部)焦点。

如: 焦点 $[4x^2-y^2+16x+20=0]$ 返回 $A=(-2, -2.24)$ 和 $B=(-2, 2.24)$ 。

离心率 Eccentricity

离心率[<圆锥曲线>]: 计算圆锥(二次)曲线的离心率。

如: 离心率 $[x^2/9+y^2/4=1]$ 返回 $a=0.75$ 。

内切圆 Incircle

内切圆[<点 1>,<点 2>,<点 3>]: 返回由三个点所构成三角形的内切圆。

抛物线 Parabola

Parabola[Point F, Line g]: 建立焦点为 F, 准线为 g 的抛物线。如: 设 $a=\text{直线}[(0,1),(2,1)]$, 抛物线 $[(3,3),a]$ 得出 $x^2-6x-4y=-17$ 。

双曲线 Hyperbola

Hyperbola[Point F, Point G, Number a]: 建立焦点为 F 和 G 且半长轴长为 a 的双曲线。当焦点之间的距离 $<2a$, 将会得到一个椭圆。

如: 建立参数 a, 双曲线 $[(0, -4), (2, 4), a]$ 当 $a=1$ 时得出 $-8xy - 15y^2 + 8y = -16$ 。当

$a=5$ 时, 得到的是一个椭圆 $24x^2 - 8xy + 9y^2 - 48x + 8y = 176$ 。

Hyperbola[Point F, Point G, Segment]: 建立焦点为 F 和 G 且半长轴长等于线段 s 长度的双曲线。

Hyperbola[Point A, Point B, Point C]: 建立焦点为 A 和 B 且过点 C 的双曲线, 三点双曲线。

如: 双曲线[(1,1),(2,1),(-2,-4)] 得出 $-2.69x^2 + 1.30y^2 + 8.07x - 2.62y = 4.52$ 。

双曲线工具

双曲线(<焦点1>, <焦点2>, <主半轴长>)

双曲线(<焦点1>, <焦点2>, <主半轴线段>)

双曲线(<焦点1>, <焦点2>, <双曲线上一点>)

椭圆 Ellipse

Ellipse[Point F, Point G, Number a]: 建立焦点为 F 和 G 且半长轴长为 a 的椭圆。如果条件: $2 \times \text{半长轴长度} > \text{焦点之间的距离}$ 不能满足, 将会得到一个双曲线。

如: 椭圆[(0,1),(1,1),1] 得出 $12x^2 + 16y^2 - 12x - 32y = -7$ 。

Ellipse[Point F, Point G, Segment]: 建立焦点为 F 和 G 且 半长轴长等于线段 s 长度的椭圆。如: 设 s=线段[(0,1),(2,1)]; 椭圆[(0,1),(2,1),s] 得出 $3x^2 + 4y^2 - 6x - 8y = 5$ 。

Ellipse[Point A, Point B, Point C]: 建立焦点为 A 和 B 且过点 C 的椭圆, 三点椭圆。

如: 椭圆[(0,1),(2,1),(1,2)] 得出 $1x^2 + 2y^2 - 2x - 4y = -1$ 。

椭圆(<焦点1>, <焦点2>, <主半轴长>)

椭圆(<焦点1>, <焦点2>, <主半轴线段>)

椭圆(<焦点1>, <焦点2>, <椭圆上一点>)

圆周 Circle

Circle[Point M, Number r]: 建立圆心为 M 半径为 r 的圆, 圆心和半径。

Circle[Point M, Segment]: 建立圆心为 M 半径为 s 的圆 (参考先确定半径再画圆的绘图工具), 圆心与半径长。

Circle[Point M, Point A]: 建立圆心为 M 且过点 A 的圆, 圆心与一点。

Circle[Point A, Point B, Point C]: 建立过三点 A, B, C 的圆, 也称过三点圆。

圆周(<圆心>, <半径长度>)

圆周(<圆心>, <半径>)

圆周(<圆心>, <圆上一点>)

圆周(<轴线>, <圆上一点>)

圆周(<点1>, <点2>, <点3>)

圆周(<圆心>, <半径>, <轴向量>)

圆周(<圆心>, <圆上一点>, <轴向量>)

圆锥曲线 Conic

圆锥曲线(<点1>, <点2>, <点3>, <点4>, <点5>)

Conic[Point A, Point B, Point C, Point D, Point E]: 得到过五点 A, B, C, D, E 的圆锥曲线, 过五点圆锥曲线。

圆锥曲线(<x平方系数>, <y平方系数>, <常数项>, <xy系数>, <x系数>, <y系数>): 返回一个平面圆锥（二次）曲线 $ax^2 + dxy + by^2 + ex + fy + c = 0$ 。如：圆锥曲线[2,3,-1,4,2,-3] 得出 $2x^2 + 4xy + 3y^2 + 2x - 3y = 1$ 。还可以结合其他指令，如圆锥曲线 [随机排列 [{1,2,3,4,5,6}]]，每次输入创建不同的曲线。

中心 Center

中心(<二次曲线>): 返回一个圆、椭圆或是双曲线的中心。

如：中心[x^2+4y^2+2x-8y+1=0]返回点 A=(-1,1)。

中心(<二次曲面>)

中点或中心工具可用。你还可以在一个平面圆锥曲线（圆 或椭圆）上点击从而创建它的中心点。

准线 Directrix

准线[<抛物线>]: 得出圆锥（二次）曲线（抛物线）的准线。准线[x^2-3x+3y=9] 得出直线 y=4.5。

密切圆 OsculatingCircle

OsculatingCircle[Point, Function]: 建立函数 f 在点 A 的密切圆。

OsculatingCircle[Point, Curve]: 建立曲线 c 在点 A 的密切圆。

10. 参数曲线 Parameter Curve

曲线 Curve, CurveCartesian

曲线[<x(t)>, <y(t)>, <参数值 t>, <t-起始值>, <t-终止值>], 以指定的 x-表达式（x(t)）和 y-表达式（y(t)），以及（用名称为 t（或其他）的可变参数）限定的闭区间[<t-起始值>, <t-终止值>]:

Curve[Expression e1, Expression e2, Parameter t, Number a, Number b]: 生成坐标为（e1,e2）的曲线，其中 e1, e2 为参数式。如：曲线 c:

Curve[2cos(t), 2sin(t), t, 0, 2pi] 新建了一个以坐标系原点为圆心、半径为 2 的圆;

曲线[(1-cos(t))*cos(t), (1-cos(t))*sin(t), t, 0, 2pi] 生成心形曲线。

曲线(<x(t)>, <y(t)>, <参变量t>, <t-起始值>, <t-终止值>): 2D 参数曲线

曲线(<x(t)>, <y(t)>, <z(t)>, <参变量t>, <t-起始值>, <t-终止值>): 3D 参数曲线

参数曲线相关指令

Curvature[Point, Curve]: 计算曲线上一点的曲率。

CurvatureVector[Point, Curve]: 生成曲线上一点的曲率向量。

Derivative[Function]: 计算函数 f(x) 的微分。

Derivative[Function, Number n]: 计算函数 f(x) 的 n 次微分。

Length[Curve, Number t1, Number t2]: 计算曲线在参数 t1 和 t2 之间的长度。

Length[Curve c, Point A, Point B]: 计算曲线 c 在点 A 和 B 之间的曲线长度。

OsculatingCircle[Point, Curve]: 曲线 c 在点 A 的密切圆。

Tangent[Point, Curve]: 建立曲线 c 过点 A 的切线。

11. 圆弧和扇形 Arc and Sector

圆弧的代数值为其长度，扇形的代数值为其面积。

弧 Arc

`Arc[Conic, Point A, Point B]`: 得到介于圆锥曲线 c 上两点 A 和 B 之间的弧。

`Arc[Conic, Number t1, Number t2]`: 得到介于圆锥曲线 c 上在参数 $t1$ 和 $t2$ 之间的弧。

- 圆: $(\cos(t), \sin(t))$ 表示半径为 r 的圆
- 椭圆: $(a\cos(t), b\sin(t))$ 其中 a 和 b 为长轴和短轴半长

圆弧 CircularArc

`CircularArc[Point M, Point A, Point B]`: 建立以 M 点为圆心，起点为 A ，终点为 B 的圆弧。

`CircumcircularArc[Point A, Point B, Point C]`: 建立通过 A , B , C 三点的圆弧, 三点圆弧。

扇形 CircularSector

`CircularSector[Point M, Point A, Point B]`: 建立 M 点为圆心，起点为 A ，终点为 B 的扇形。

`CircumcircularSector[Point A, Point B, Point C]`: 建立通过 A , B , C 三点的扇形, 三点扇形。

`Sector[Conic, Point A, Point B]`: 建立介于圆锥曲线 c 上的两点 A 和 B 之间的圆锥曲线扇形区域。

`Sector[Conic, Number t1, Number t2]`: 建立介于圆锥曲线 c 上的两参数 $t1$ 和 $t2$ 之间的圆锥曲线扇形区域。

半圆 Semicircle

`Semicircle[Point A, Point B]`: 建立线段 AB 上的半圆。

12. 文字

分数文字 FractionText

`FractionText[Number]`: 将数值转换成分数形式。

LaTeX文本 LaTeX

`LaTeX[Object]`: 得到用 $LaTeX$ 表达的对象文字。如: 若 $a = 2$ 则 $f(x) = ax^2$, 那么 $LaTeX[f]$ 得到 $2x^2$ 。

`LaTeX[Object, Boolean]`: 通过逻辑判断得到用 $LaTeX$ 表达的对象文字。如果为真, 那么用数值代替代数, 否则显示代数式。

解释: $LaTeX$ 是一种基于 TeX 的排版系统, 利用这种格式即使使用者没有排版和程序设计的知识也可以充分发挥由 TeX 所提供的强大功能, 能在几天, 甚至几小时内生成很多具有书籍质量的印刷品。对于生成复杂表格和数学公式, 这一点表现得尤为突出。因此它非常适用于生成高印刷质量的科技和数学类文档。

字符与Unicode互转 LetterToUnicode

LetterToUnicode["Letter"]: 将'字符'转化成 Unicode（统一码）。如：

LetterToUnicode["a"] 得到 97。

UnicodeToLetter[Integer]: 將整數的萬國碼轉換回 Graphics View 中的字母

範例: **UnicodeToText[97]** 會得到 text "a".

TextToUnicode["Text"]: 将'文字'转化成 Unicode（统一码）。如: **TextToUnicode["Some text"]** 得到{83, 111, 109, 101, 32, 116, 101, 120, 116}。

UnicodeToText[List of Integers]: 将 Unicode 转化成文字。如: **UnicodeToText[{104, 101, 108, 108, 111}]** 会得到文字“hello”。

解释: Unicode（统一码、万国码、单一码）是一种在计算机上使用的字符编码。它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求。

名称 Name

Name[Object]: 得到绘图区的对象名称。

对象 Object

Object[Name of object as text]: 得到对象的名称。改命令与“名称”相反。

文字 Text

Text[Object]: 将对象的方程式转化成文字。最终出现的数值或字符结果。

Text[Object, Boolean]: 通过逻辑判断得到对象的文字。参考 **LaTeX[Object, Boolean]**, **LaTeX** 命令得到的文字形式使用 **LaTeX** 语法表达的。

表格文本 TableText

TableText[List 1, List 2, List 3,...]: 產生串列物件的表格

註: 電腦預設每一串列產生表格中的一列。

範例:

TableText[{x^2, x^3, x^4}] 產生串列的表格。每列是由左到右排列

TableText[Sequence[i^2, i, 1, 10]] 產生序列的表格。每列是由左到右排列

TableText[List 1, List 2, List 3,..., "Alignment of text"]: 產生一個 text 包含表格內所列的物件。選擇的 text 即 **Alignment of text** 控制表格 text 的方向和排列。

註: 可能的值為 "vl", "vc", "vr", "v", "h", "hl", "hc", "hr". 預設值為 "hl".

"v" = 垂直, i.e. lists are columns

"h" = 水平, i.e. lists are rows

"l" = 從左到右排列

"r" = 從右到左排列

"c" = 中間

範例:

TableText[{1,2,3,4},{1,4,9,16},"v"] 產生兩行四列的 text，其中每個元素都是從左到右排列

TableText[{1,2,3,4},{1,4,9,16},"h"] 產生四行兩列的 text，其中每個元素都是從從左到右

排列

`TableText[{11.2, 123.1, 32423.9, "234.0"}, "r"]` 產生一系列的 `text`，其中每個元素都是從右到左排列

表格文本(<列表1>, <列表2>, ...)

表格文本(<列表1>, <列表2>, ..., <对齐方式 "v" #垂直|"h" #水平|"l" #左对齐|"r" #右对齐|"c" #居中|...>)

13. 轨迹 Locus

`Locus[Point Q, Point P]`: 求点 Q 的轨迹线 (P 为控制点)。

轨迹(<构造轨迹的点>, <控制点>)

轨迹(<构造轨迹的点>, <滑动条>)

轨迹(<斜率场>, <点>)

轨迹(<f(x, y)>, <点>)

14. 列表 List

附加 追加 Append

`Append[List, Object]`: 将对象加入列表中成为新的列表的最后一个元素。

`Append[Object, List]`: 将对象加入列表中成为新的列表的第一个元素。

符合条件的数目 CountIf

`CountIf[Condition, List]`: 计算符合条件的元素个数。如: `CountIf[x < 3, {1, 2, 3, 4, 5}]`得到 2。

元素 Element

`Element[List, Number n]`: 调用列表中的第 n 个元素。

First

`First[List]`: 得到列表中的第一个元素。

`First[List, Number n of elements]`: 得到一个包含给定列表中前 n 个元素的新列表。

Last

`Last[List]`: 得到列表中的最后一个元素。

`Last[List, Number n of elements]`: 得到一个包含给定列表中后 n 个元素的新列表。

插入 Insert

`Insert[Object, List, Position]`: 将对象插入到列表的指定位置。 `Insert[List 1, List 2, Position]`: 将 list1 的所有元素插入到 list2 中指定的位置。

例: `Insert[{11, 12}, {1, 2, 3, 4, 5}, 3]` 将 list1 所有元素插入到list2 的第三个 (與隨後的) 位置並得到新串列 {1, 2, 11, 12, 3, 4, 5}

註: 若指定的位置為負值, 則位置由右邊算起。

例: `Insert[{11, 12}, {1, 2, 3, 4, 5}, -2]` 將list1 所有元素插入到list2 的倒數第二個位置並得到串列 {1, 2, 3, 4, 11, 12, 5}.

交集 Intersection

`Intersection[List 1, List 2]`: 將兩個列表的公共部分生成一個新的列表。

并集 Union

`Union[List 1, List 2]`: 合併兩列表並去除重複的元素。

迭代數列 IterationList

`IterationList[Function, Number x0, Number n]`: 可以得到 $n+1$ 個元素的列表, 其中元素是由函數帶入 x_0 經過多次迭代生成的。如: 定義 $f(x) = x^2$, 輸入 $L =$

`IterationList[f, 3, 2]` 可以得到 $L = \{3, 9, 81\}$

合併 Join

`Join[List 1, List 2, ...]`: 將多個列表合併成一個列表。這種合併保留相同元素, 不會重新排序。

`Join[List of lists]`: 將子列表合併得到一個更大的列表。新列表包含所有元素, 並保留相同元素, 不會重新排序。

篩選 KeepIf

`KeepIf[Condition, List]`: 篩選列表中符合條件的元素。如: `KeepIf[x<3, {1, 2, 3, 4, 1, 5, 6}]` 可得到新列表{1, 2, 1}。

長度 Length

`Length[List]`: 計算列表的長度, 也就是元素個數。

最小值 Min

`Min[List]`: 得到列表中最小的元素。

最大值 Max

`Max[List]`: 得到列表中最大的元素。

內積 Product

`Product[List of numbers]`: 計算列表中所有數字的乘積。

移除未定義對象 RemoveUndefined

`RemoveUndefined[List]`: 移除列表中没有定義的對象。

排序 Sort

Sort[List]: 对列表中的数值、文字或点做排序。如: **Sort[{3, 2, 1}]**可得到列表{1, 2, 3}。

反序 Reverse

Reverse[List]: 倒序排列列表。

求和 Sum

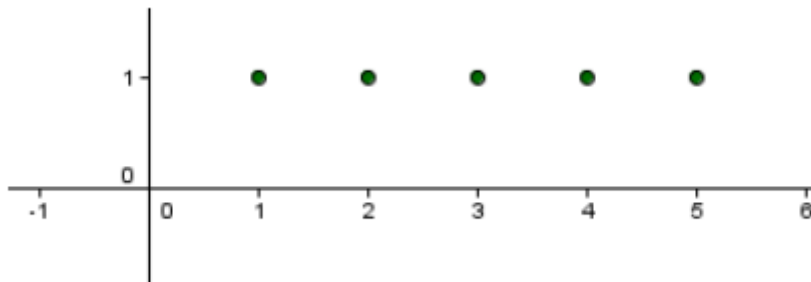
Sum[List]: 计算列表中所有元素之和。该命令可以用于数值、点、向量、文字和函数。

提取 Take

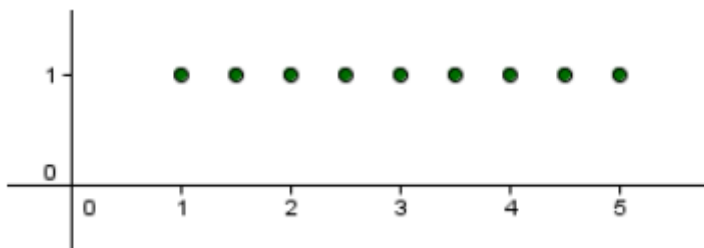
Take[List, Start position m, End position n]: 提取列表中第 m 个元素到第 n 个元素组成新的列表。

序列 Sequence

Sequence[Expression, Variable i, Number a, Number b]: 生成一个序列, 使用给定的表达式及变量 i 从 a 到 b 变化。如: **L = Sequence[(i, 1), i, 1, 5]**得到的点序列, x 坐标变化从 1 到 5, 如下图所示。



Sequence[Expression, Variable i, Number a, Number b, Number s]: 生成一个序列, 使用给定的表达式及变量 i 从 a 到 b 变化, 其中步长为 s。如: **L = Sequence[(i, 1), i, 1, 5, 0.5]**得到的点序列, x 坐标变化从 1 到 5, 步长为 0.5。如下图所示。



15. 几何变换 Transformation

位似 Dilate

Dilate[Point A, Number, Point S]: 点 A 关于点 S 做指定比例的位似变换。

Dilate[Line, Number, Point S]: 直线关于点 S 做指定比例的位似变换。

Dilate[Conic, Number, Point S]: 圆锥曲线关于点 S 做指定比例的位似变换。

Dilate[Polygon, Number, Point S]: 多边形关于点 S 做指定比例的位似变换。

Dilate[Image, Number, Point S]: 图片关于点 S 做指定比例的位似变换。

位似(<几何对象>, <位似比>)

位似(<几何对象>, <位似比>, <位似中心>)

对称, 镜像, 反射 Reflect、Mirror

Reflect[Point A, Point B]: 点 A 对点 B 做镜像。

Reflect[Line, Point]: 直线对指定点做镜像。

Reflect[Conic, Point]: 圆锥曲线对指定点做镜像。

Reflect[Polygon, Point]: 多边形对指定点做镜像。

Reflect[Image, Point]: 图片对指定点做镜像。

Reflect[Point, Line]: 点对指定直线做镜像。

Reflect[Line g, Line h]: 直线 g 对指定直线 h 做镜像。

Reflect[Conic, Line]: 圆锥曲线对指定直线做镜像。

Reflect[Polygon, Line]: 多边形对指定直线做镜像。

Reflect[Image, Line]: 图片对指定直线做镜像。

Reflect[Point, Circle]: 点对圆进行反演。

对称(<几何对象>, <对称中心点>): 中心对称工具

对称(<几何对象>, <对称轴 直线|射线|线段>): 指定轴的对称工具

对称(<几何对象>, <对称平面>)

对称(<几何对象>, <反演基圆>): 几何对象关于指定圆的反演图像（哈哈镜原理），有反演工具

Mirror(Object, Point/Plane/Circle/Line/Ray/Segment)

旋转 Rotate

Rotate[Point, Angle]: 点关于原点旋转指定的角度。

Rotate[Vector, Angle]: 向量关于原点旋转指定的角度。

Rotate[Line, Angle]: 直线关于原点旋转指定的角度。

Rotate[Conic, Angle]: 圆锥曲线关于原点旋转指定的角度。

Rotate[Polygon, Angle]: 多边形关于原点旋转指定的角度。

Rotate[Image, Angle]: 图片关于原点旋转指定的角度。

Rotate[Point A, Angle, Point B]: 点 A 关于点 B 旋转指定的角度。

Rotate[Line, Angle, Point]: 直线关于指定点旋转指定的角度。

Rotate[Vector, Angle, Point]: 向量关于指定点旋转指定的角度。

Rotate[Conic, Angle, Point]: 圆锥曲线关于指定点旋转指定的角度。

Rotate[Polygon, Angle, Point]: 多边形关于指定点旋转指定的角度。

Rotate[Image, Angle, Point]: 图片关于指定点旋转指定的角度。

旋转(<几何对象>, <度|弧度>): 绕坐标轴原点的旋转工具

旋转(<几何对象>, <度|弧度>, <旋转中心>): 绕指定旋转中心旋转的工具

旋转(<几何对象>, <度|弧度>, <旋转轴>): 绕指定轴旋转的工具

旋转(<几何对象>, <度|弧度>, <轴上的点>, <轴方向或平面>):

平移 Translate

Translate[Point, Vector]: 点沿指定向量平移。

Translate[Line, Vector]: 直线沿指定向量平移。

Translate[Conic, Vector]: 圆锥曲线沿指定向量平移。

Translate[Function, Vector]: 函数沿指定向量平移。

Translate[Polygon, Vector]: 多边形沿指定向量平移。

Translate[Image, Vector]: 图片沿指定向量平移。

Translate[Vector, Point]: 向量 v 平移 to 指定的点。

平移(<几何对象>, <向量>): 沿向量移动几何对象。平移一个多边形时, 会同时新建平移后的顶点和线段。平移工具

平移(<向量>, <起点>): 向量移动到起点

切变 Shear

Shear(Object, Line/Ray/Segment, Ratio)

切变(<几何对象>, <直线|射线|线段>, <比>)

切变一个对象将会: 直线上的点位置保持不变, 对象上的点进行转换, 与直线距离为 d 的点转换为与直线距离为 " $d \times$ 比率" 的点 (转换的方向是关于直线的不同的半平面)。

伸压、伸缩 Stretch

伸缩(<几何对象>, <向量>)

将对象在指定向量的平行方向上以向量的幅度 (模长) 为比率进行拉伸或压缩。对象的全部构成点中, 如果其位置在向量的起点垂线 (过向量起点并垂直于向量的直线) 上, 则伸压后位置不变;

其他的构成点与起点垂线之间的距离, 伸压后转换为原距离乘以给定比率。

伸缩(<几何对象>, <直线|射线|线段>, <比>)

将对象在指定直线的垂直方向上以指定比率进行拉伸或压缩。即对象的全部构成点中, 如果其位置在指定直线上, 则伸压后这个构成点的位置不变;

其他的构成点与指定直线之间的距离, 伸压 后转换为原距离乘以指定比率。

16. 统计

求和 SigmaXX, SigmaXY, SigmaYY

SigmaXX[List of numbers]: 计算给定数值的平方和。

SigmaXX[List of points]: 计算给定点列的 x 坐标的平方和。

SigmaXY[List of x-coordinates, List of y-coordinates]: 计算两点列的 x 与 y 坐标乘积的和。

SigmaXY[List of points]: 计算 x 坐标和 y 坐标的乘积和。

SigmaYY[List of Points]: 计算给定点列 y 坐标的平方和。

中位数 Median

Median[List of numbers]: 得到列表中元素的中位数。

众数 Mode

Mode[List of numbers]: 得到列表中元素的众数。

平均数 Mean

Mean[List of numbers]: 计算列表中元素的平均数。

MeanX[List of points]: 计算列表中元素 x 坐标的平均数。

MeanY[List of points]: 计算列表中元素 y 坐标的平均数。

四分位数 Q1, Q2, Q3

Q1[List of numbers]: 得到列表元素中的第一四分位数。

Q3[List of numbers]: 得到列表元素中的第三四分位数。

方差 Variance

Variance[List of numbers]: 计算列表元素的方差。

标准差 SD

SD[List of Numbers]: 计算列表中数值的标准差。

相关系数 CorrelationCoefficient

CorrelationCoefficient[List of x-coordinates, List of y-coordinates]: 使用给定的 x 坐标和 y 坐标计算相关系数的乘积。

CorrelationCoefficient[List of points]: 使用给定点的坐标计算相关系数的乘积。

协方差 Covariance

Covariance[List 1 of numbers , List 2 of numbers]: 计算两列表的协方差。

Covariance[List of points]: 计算 x 坐标和 y 坐标的协方差。

正态分布函数 Normal

Normal[Mean, Standard deviation, Variable value]: 计算函数 $(\Phi(x)-\text{mean})/(\text{standard deviation})$ 其中 $\Phi(x)$ 是符合 $N(0,1)$ 分布的概率密度函数。

反正态函数 InverseNormal

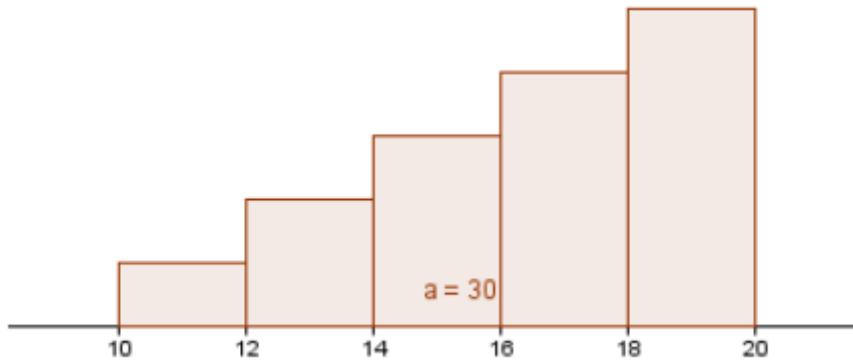
InverseNormal[Mean, Standard deviation, Probability]: 计算具有给定概率正态分布的区间点。计算函数 $\Phi^{-1}(P) * \sigma + \mu$, 其中 $\Phi(x)^{-1}$ 是 $\Phi(x)$ 概率密度函数佈於 $N(0,1)$ 的反函数。

註: 從給定的機率算出在常態分佈曲線下方 x 的座標, 回傳 x 座標。

条形图 BarChart

BarChart[Start value, End value, List of heights]: 建立指定区间内的条形图, 条形个数取决于表格长度, 高度取决于元素大小。如: **BarChart[10, 20, {1,2,3,4,5}]** 得到如下

图所示。

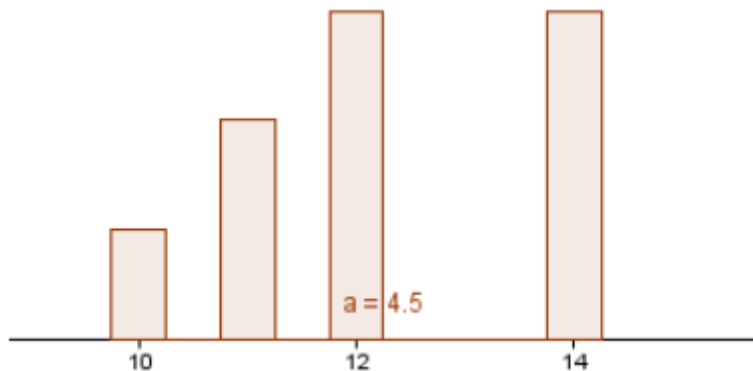


`BarChart[Start value a, End value b, Expression, Variable k, From number c, To number d]`: 建立 $[a, b]$ 之间的条形图, 用含变量 k 的表达式表示条形高度, 其中 k 从 c 到 d 变化。

`BarChart[List of raw data, Width of bars]`: 由给定数据生成条形图。可设定条宽。

`BarChart[List of data, List of frequencies]`: 建立位置长度的条形图。

`BarChart[List of data, List of frequencies, Width of bars w]`: 建立位置长度的条形图, 并指定条宽。如: `BarChart[{10,11,12,13,14}, {1,2,3,0,3}, 0.5]` 得到如下图所示。

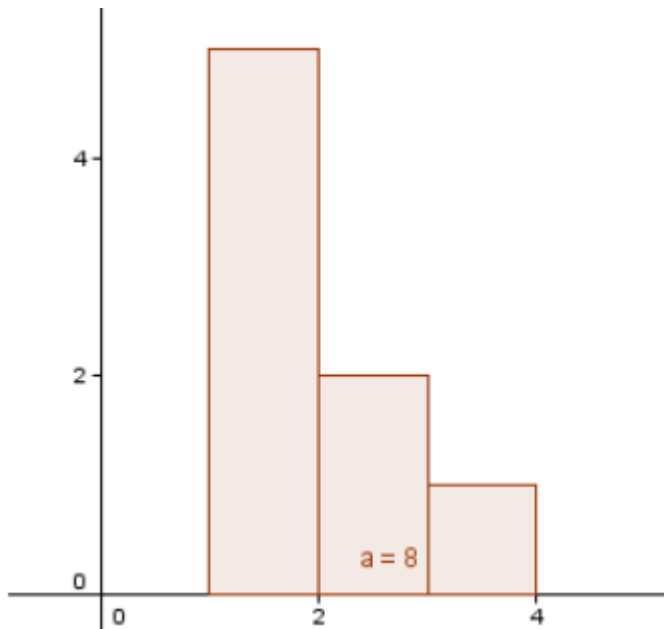


直方图 Histogram

`Histogram[List of class boundaries, List of heights]`: 由给定的高度建立直方图。

`Histogram[List of class boundaries, List of raw data]`: 使用给定的数据建立直方图。

如: `Histogram[{1, 2, 3, 4}, {1.0, 1.1, 1.1, 1.2, 1.7, 2.2, 2.5, 4.0}]` 得到如下图所示。



盒形图 BoxPlot

BoxPlot[yOffset, yScale, List of raw data]: 建立给定数据的盒状图，并在坐标系的垂直方向是由 y 的偏移量控制的，高度受 y 的度量影响。

BoxPlot[yOffset, yScale, Start value, Q1, Median, Q3, End value]: 在区间上建立盒形图。

线性回归 FitLine

FitLine[List of points]: 计算 xy 坐标上的线性回归。

其他回归

FitExp[List of points]: 计算指数回归曲线。

FitLineX[List of points]: 计算 xy 坐标平面上的线性回归。

FitLog[List of points]: 计算对数回归曲线。

FitLogistic[List of points]: 计算 Logistic 回归曲线。

FitPoly[List of points, Degree n of polynomial]: 计算 n 次多项式的回归方程。

FitPow[List of points]: 计算 axb 型的回归曲线。

FitSin[List of points]: 计算 $a + b \sin(cx + d)$ 型的回归曲线。

统计量指令 Sxx, Sxy, Syy

Sxx[List of numbers, List of numbers]: 计算 $\Sigma(x^2) - \Sigma(x) \times \Sigma(x)/n$

Sxx[List of points]: 计算 $\Sigma(x^2) - \Sigma(x) \times \Sigma(x)/n$

Sxy[List of numbers, List of numbers]: 计算 $\Sigma(xy) - \Sigma(x) \times \Sigma(y)/n$

Sxy[List of points]: 计算 $\Sigma(xy) - \Sigma(x) \times \Sigma(y)/n$

Syy[List of numbers, List of numbers]: 计算 $\Sigma(y^2) - \Sigma(y) \times \Sigma(y)/n$

Syy[List of points]: 计算 $\Sigma(y^2) - \Sigma(y) \times \Sigma(y)/n$

註: 這些统计量是非常態形式的 XY 的方差及协方差。 $Sxx = N \text{ var}(X)$, $Syy = N \text{ var}(Y)$, and $Sxy = N \text{ cov}(X, Y)$

範例: 可以使用 $\text{Sxy}[\text{list}] / \text{sqrt}(\text{Sxx}[\text{list}] \text{ Syy}[\text{list}])$ 來算出點串列中的協方差係數。

17. 表格 Table

行序 Row

Row[Spreadsheet cell]: 得到非空单元格的行号。如: 若 B3 非空, 那么 Row[B3] 可以得到数值 $a=3$, 因为 B3 在第三行。

行序(<表格区单元格>)

列序 Column

Column[Spreadsheet cell]: 得到非空单元格的列号。如: 若 B3 非空, 那么 Column[B3] 可以得到数值 $a=2$, 因为 B 是第二列。

列序(<表格区单元格>)

列名称 ColumnName

ColumnName[Spreadsheet cell]: 回傳非空格的欄名當作文字

例: 如果 A1 格是非空, 則 ColumnName[A1] 可在幾何視區得到文字 "A"。

列名称(<表格区单元格>)

单元格区域数值列表 CellRange

CellRange[Start cell, End cell]: 由指定范围的单元格生成列表, 先列后行排列。

单元格区域数值列表(<起始单元格>, <终止单元格>)

单元格 Cell

Cell(<列序>, <行序>): 单元格(<列序>, <行序>)

填充单元格 FillCells

填充单元格(<单元格区域>, <对象>)

填充单元格(<单元格>, <列表>)

填充单元格(<单元格>, <矩阵>)

FillCells(B5,{55}): B5 = 55

FillCells(A4:B6, CellRange(A1:B3)): 复制

填充列 FillColumn

填充列(<列序>, <列表>)

填充行 FillRow

填充行(<行序>, <列表>)

18. 逻辑命令

If

If[Condition, Object]: 如果条件式为真, 可以得到 **Object**。如果条件式为假时, 为 **undefined Object**。

If[Condition, Object a, Object b]: 如果条件式为真, 可以得到 **Object a**。如果条件式为假时, 为 **Object b**。

IsDefined

IsDefined[Object]: 根据对象是否有定义, 返回真假值。

IsInteger

IsInteger[Number]: 根据对象是否为整数, 返回真假值。

19. 优化指令 Optimization

最大值点 Maximize

最大值点[<因变量（派生的数值对象） >,<自变量（自由的数值对象） >]:

计算自变量（自由的数值对象）使其给出因变量（派生的数值对象）的最大值。自变量（自由的数值对象）必须是一个滑杆, 这个滑杆的区间被用做搜寻的区间。如果是一个复杂的构造, 这个指令有可能失败。

最小值点 Minimize

最小值点[<因变量（派生的数值对象） >,<自变量（自由的数值对象） >]:

计算自变量（自由的数值对象）使其给出因变量（派生的数值对象）的最小值。

20. 三维指令 3D

上底 Top

上底(<二次曲面>)

下底 Bottom

下底(<二次曲面>)

侧面 QuadricSide

侧面(<二次曲面>)

底面 Ends

底面(<二次曲面>)

高度 Height

高度(<立体图形>)

注意：高度角 alt()

棱柱 Prism

棱柱(<多边形>, <最高点>)

棱柱(<多边形>, <高度>)

棱柱(<点1>, <点2>, ...)

棱锥 Pyramid

棱锥(<多边形>, <顶点>)

棱锥(<多边形>, <高度>)

棱锥(<点1>, <点2>, <点3>, <点4>, ...)

平面 Plane

平面(<多边形>)

平面(<圆锥曲线>)

平面(<点>, <平行的平面>)

平面(<点>, <经过的直线>)

平面(<直线1>, <直线2>)

平面(<点1>, <点2>, <点3>)

平面(<点>, <向量1>, <向量2>)

垂直平面 OrthogonalPlane

垂直平面(<点>, <直线>)

垂直平面(<点>, <向量>)

中垂面 PlaneBiSector

中垂面(<线段>)

中垂面(<点1>, <点2>)

球面 Sphere

球面(<球心>, <半径>)

球面(<球心>, <球面上一点>)

曲面 Surface

曲面(<函数>, <绕x轴旋转的角度-度|弧度>)

曲面(<曲线>, <绕轴线旋转的角度-度|弧度>, <旋转轴-线段|射线|直线>)

曲面(<x表达式>, <y表达式>, <z表达式>, <参变量1>, <起始值>, <终止值>, <参变量2>, <起始值>, <终止值>)

体积 Volume

体积(<立体图形>)

无限长圆柱 InfiniteCylinder

无限长圆柱(<轴线>, <半径>)

无限长圆柱(<轴线上的一点>, <轴向量>, <半径>)

无限长圆柱(<轴线上点1>, <轴线上点2>, <半径>)

无限长圆锥 InfiniteCone

无限长圆锥(<顶点>, <轴向量>, <半顶角大小 度|弧度>)

无限长圆锥(<顶点>, <轴线上的一点>, <半顶角大小 度|弧度>)

无限长圆锥(<顶点>, <轴线>, <半顶角大小 度|弧度>)

相交曲线 IntersectConic

相交曲线(<平面>, <二次曲面>)

相交曲线(<二次曲面1>, <二次曲面2>)

注意：区别于“相交路径” - IntersectPath

圆柱 Cylinder

圆柱(<圆锥曲线底面>, <高度>)

圆柱(<下底圆心>, <上底圆心>, <半径>)

圆锥 Cone

圆锥(<圆锥曲线底面>, <高度>)

圆锥(<底面圆心>, <顶点>, <底面半径>)

圆锥(<顶点>, <向量>, <半顶角大小 度|弧度>)

正四面体 Tetrahedron

正四面体(<等边三角形>)

正四面体(<点1>, <点2>, <点3>)

正四面体(<点1>, <点2>, <垂直于线段"点1点2"的向量|线段|射线|直线; 或者平行于线段"点1点2"的多边形|平面>)

正六面体 Cube

正六面体(<正方形>)

正六面体(<点1>, <点2>, <点3>)

正六面体(<点1>, <点2>, <垂直于线段"点1点2"的向量|线段|射线|直线; 或者平行于线段"点1点2"的多边形|平面>)

正八面体 Octahedron

正八面体(<等边三角形>)

正八面体(<点1>, <点2>, <点3>)

正八面体(<点1>, <点2>, <垂直于线段"点1点2"的向量|线段|射线|直线; 或者平行于线段"点1点2"的多边形|平面>)

正十二面体 Dodecahedron

正十二面体(<正五边形>)

正十二面体(<点1>, <点2>, <点3>)

正十二面体(<点1>, <点2>, <垂直于线段"点1点2"的向量|线段|射线|直线; 或者平行于线段"点1点2"的多边形|平面>)

正二十面体 Icosahedron

正二十面体(<等边三角形>)

正二十面体(<点1>, <点2>, <点3>)

正二十面体(<点1>, <点2>, <垂直于线段"点1点2"的向量|线段|射线|直线; 或者平行于线段"点1点2"的多边形|平面>)

展开图 Net

展开图(<多面体>, <展开程度值 0~1>)

展开图(<多面体>, <展开程度值 0~1>, <面>, <棱1>, <棱2>, ...)

注意：展开 Expand， 三角式展开 TrigExpand