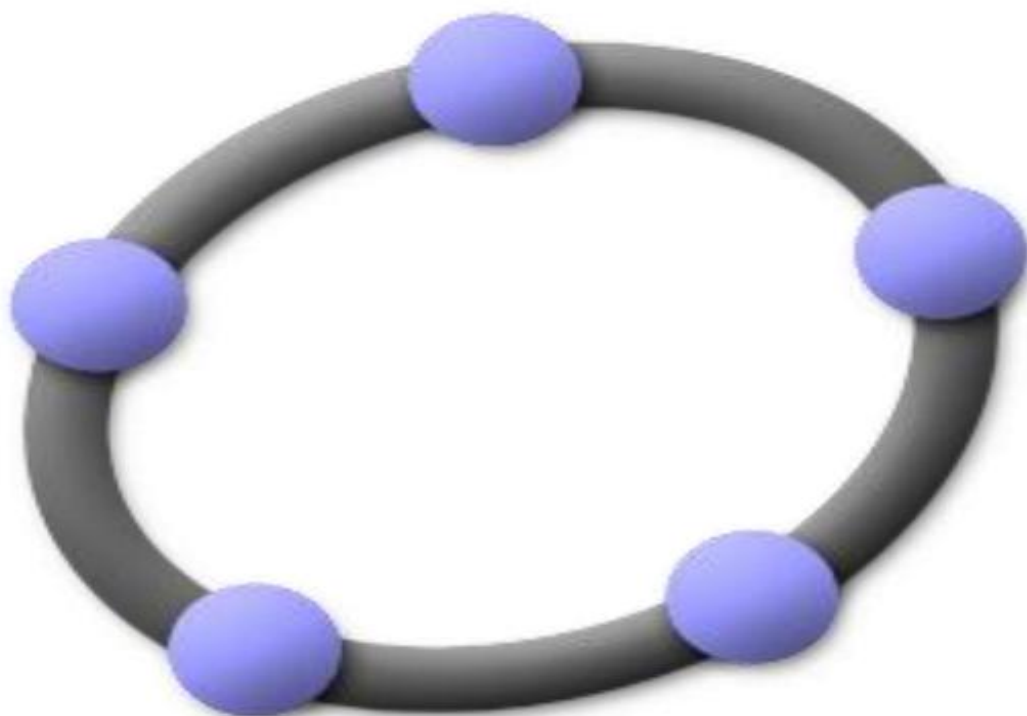


# GeoGebra指令汇编

简体中文5.0.527.0



唐家军 编著

2019年3月6日

# 前言

GeoGebra 的开源性使得它的指令和功能更新很快，本文的指令名（包括英文和中文名称）指令，基于 GeoGebra 5.0.527.0 版本，按照功能分块并以中文拼音顺序排列。

GeoGebra 指令包括函数和命令两部分，在中文界面使用过程中，输入英文指令或者中文指令名称，只要符合格式规范，都可以得到需要的结果。

函数都是使用英文名称简写后边携带圆括弧的形式，对圆括弧内的数值、表达式或者变量进行计算，返回一个结果。函数没有中文格式。

如：“ $\text{acos}()$  或  $\text{arccos}()$ ：Arccosine. 反余弦函数。”中“ $\text{acos}()$  或  $\text{arccos}()$ ”为函数名称；“Arccosine”为函数的英语释义；“反余弦函数”为函数的中文释义。

在 GeoGebra “指令栏”或“运算区”输入“ $\text{acos}(1)$ ”，则返回一个值，显示在“代数区”中；输入“ $\text{acos}(x)$ ”，则返回一个函数，其解析式显示在“代数区”中，图象同时显示在“绘图区”中。

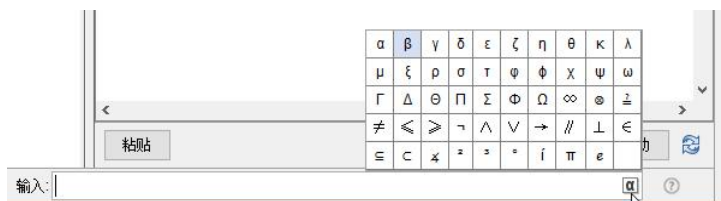
在早期版本中，命令都是使用英文或者中文名称简写后边携带方括弧的形式，对方括弧内的数值、表达式或者变量进行计算，返回一个结果。输入指令时方括弧必须输入，尖括弧不用输入，直接输入尖括弧内的对象名称即可调用对象。在较新的版本中，命令后边携带方括弧和圆括弧都符合格式规范。本指令中描述的命令，都使用圆括弧了。

如：“CircularArc. 圆弧过圆心与两点”指令可以输入为：“CircularArc(<圆心>, <点 1>, <点 2>)”和“圆弧(<圆心>, <点 1>, <点 2>)”两种格式。尖括弧内为对象名称或点坐标。

假设绘图区域中有点 O 和 A、B。在 GeoGebra “指令栏”或“运算区”输入“CircularArc(O, A, B)”或者“圆弧(O, A, B)”，则在“代数区”显示一个圆锥曲线标签和解析式（取决于代数区“分类方式”设置），在“绘图区”显示构造的圆弧。此圆弧以点 O 为圆心，OA 为半径，自点 A 开始，逆时针到弧与射线 OB 相交处结束。点 B 控制弧长，但不一定非要位于弧线之上。

指令中使用的英文字母和符号，都需要使用半角，中文指令名称可以使用全角。多数指令可以同时适用于 GeoGebra “指令栏”和“运算区”，但有时格式有特殊要求。

在 GeoGebra 的“视图”-“布局”中，勾选“指令栏”和选择指令栏“显示”方式后，画板中会出现“指令栏”，在“指令栏”中可以输入指令，如下图。在系统菜单“视图”中勾选“运算区”，会出现运算区域。在 GeoGebra 中，指令输入有自动补齐功能，输入了指令的前边字符，会自动提示后边字符，移动光标可选择需要的格式。指令编辑框的右端，有“指令帮助”的激活按钮（问号按钮），在“指令帮助”中，可以找到需要的指令，且显示其基本的语法格式提示。双击“指令帮助”中的指令名称，其指令主体进入到指令栏“输入：”后的编辑区域内，可以根据“指令帮助”下方显示的“指令说明”编辑自己需要的指令具体内容。“α”按钮可以辅助输入指令内容。



本指令汇编自 5.0 版本开始收集，官网信息与软件偶有不同，以个人软件列示的指令帮助为准。

本文指令尽管尽力包括本版本可用的所有指令，但也是一时之“大全”。GeoGebra 其他版本的指令中文名、格式及用法亦略有不同，请参照各自版本的指令集甄别使用。因为英文指令拼写很少改变，中文名称时有变化，故本汇编涉及的指令同时包含英文和中文两种名称，且依据中文版指令分类和拼音顺序排列，但每个指令都使用英文指令名称在前，中文名称在后的形式列放。

本文案例描述的双引号中的内容，可以复制到指令栏或运算区运算格或脚本中，个人做复制验证案例时，注意不要忘记复制的指令中间的某些空格经常是代表乘法运算。

# 目 录

前 言.....	1
目 录.....	2
1 函数与运算.....	14
1.1 基本运算.....	14
1.2 函数.....	14
1.3 布尔运算.....	16
2 命令.....	18
2.1 3D.三维.....	19
2.1.1 Side. 侧面.....	19
2.1.2 PerpendicularPlane. 垂直平面.....	19
2.1.3 Ends. 底面.....	20
2.1.4 Height. 高度.....	20
2.1.5 Prism. 棱柱.....	20
2.1.6 Pyramid. 棱锥.....	21
2.1.7 Plane. 平面.....	22
2.1.8 Sphere. 球面.....	23
2.1.9 Surface. 曲面.....	23
2.1.10 Top. 上底.....	24
2.1.11 Volume. 体积.....	24
2.1.12 InfiniteCylinder. 无限长圆柱.....	25
2.1.13 InfiniteCone. 无限长圆锥.....	25
2.1.14 Bottom. 下底.....	25
2.1.15 IntersectConic. 相交曲线.....	26
2.1.16 Cylinder. 圆柱.....	26
2.1.17 Cone. 圆锥.....	27
2.1.18 Net. 展开图.....	27
2.1.19 Octahedron. 正八面体.....	28
2.1.20 Icosahedron. 正二十面体.....	29
2.1.21 Cube. 正六面体.....	29
2.1.22 Dodecahedron. 正十二面体.....	30
2.1.23 Tetrahedron. 正四面体.....	31
2.1.24 PlaneBisector. 中垂面.....	31
2.2 GeoGebraCommands.GeoGebra 指令.....	32
2.2.1 AxisStepX. X 轴步长.....	32
2.2.2 AxisStepY. Y 轴步长.....	32
2.2.3 DynamicCoordinates. 动态坐标.....	32
2.2.4 Object. 对象.....	34
2.2.5 ToolImage. 工具图标.....	34
2.2.6 SlowPlot. 缓慢绘制.....	37
2.2.7 Corner. 角点.....	37
2.2.8 Name. 名称.....	39

2.2.9	SetConstructionStep. 设置作图步骤.....	39
2.2.10	ToPoint. 转换为点.....	39
2.2.11	ToComplex. 转换为复数.....	39
2.2.12	ToPolar. 转换为极坐标形式.....	40
2.2.13	ConstructionStep. 作图步序.....	40
2.3	Spreadsheet. 表格区.....	41
2.3.1	Cell. 单元格.....	41
2.3.2	CellRange. 单元格区域数字列表.....	41
2.3.3	ColumnName. 列名称.....	41
2.3.4	Column. 列序.....	41
2.3.5	FillCells. 填充单元格.....	42
2.3.6	FillColumn. 填充列.....	42
2.3.7	FillRow. 填充行.....	42
2.3.8	Row. 行序.....	43
2.4	Financial. 财务.....	43
2.4.1	Rate. 利率.....	43
2.4.2	Payment. 每期付款额.....	43
2.4.3	Periods. 期数.....	44
2.4.4	FutureValue. 未来值.....	44
2.4.5	PresentValue. 现值.....	44
2.5	Algebra. 代数.....	45
2.5.1	Cross. 叉积.....	45
2.5.2	Division. 除法.....	45
2.5.3	Dot. 点积.....	46
2.5.4	CommonDenominator. 公分母.....	46
2.5.5	NextPrime. 后一质数.....	47
2.5.6	Simplify. 化简.....	47
2.5.7	Solutions. 解集.....	47
2.5.8	NSolve. 近似解.....	48
2.5.9	NSolutions. 近似解集.....	49
2.5.10	CompleteSquare. 配方式.....	50
2.5.11	PreviousPrime. 前一质数.....	50
2.5.12	Solve. 精确解.....	50
2.5.13	Mod. 取余.....	51
2.5.14	Div. 取整.....	52
2.5.15	IFactor. 实数域因式分解.....	53
2.5.16	IsPrime. 是否为质数.....	53
2.5.17	Factor. 因式分解.....	53
2.5.18	Divisors. 因数个数.....	54
2.5.19	DivisorsSum. 因数和.....	54
2.5.20	DivisorsList. 因数列表.....	55
2.5.21	RightSide. 右边.....	55
2.5.22	Expand. 展开.....	56
2.5.23	PrimeFactors. 质因数.....	56

2.5.24 ToBase. 转换进制.....	56
2.5.25 FromBase. 转换为十进制.....	57
2.5.26 GCD. 最大公约数.....	57
2.5.27 Max. 最大值.....	58
2.5.28 LCM. 最小公倍数.....	59
2.5.29 Min. 最小值.....	60
2.5.30 LeftSide. 左边.....	61
2.6 Probability. 概率.....	61
2.6.1 FDistribution. F 分布.....	61
2.6.2 TDistribution. t 分布.....	62
2.6.3 Erlang. 爱尔朗分布.....	63
2.6.4 Bernoulli. 伯努利分布.....	64
2.6.5 Poisson. 泊松分布.....	64
2.6.6 RandomPoisson. 泊松分布随机数.....	65
2.6.7 HyperGeometric. 超几何分布.....	65
2.6.8 LogNormal. 对数正态分布.....	67
2.6.9 BinomialDist. 二项分布.....	67
2.6.10 Binomial (BinomialCoefficient). 二项式系数.....	69
2.6.11 Uniform. 均匀分布.....	69
2.6.12 RandomUniform. 均匀分布随机数.....	70
2.6.13 ChiSquared. 卡方分布.....	70
2.6.14 Cauchy. 柯西分布.....	71
2.6.15 RandomDiscrete. 离散随机数.....	72
2.6.16 Logistic. 逻辑分布.....	73
2.6.17 InverseFDistribution. 逆 F 分布.....	74
2.6.18 InverseTDistribution. 逆 T 分布.....	74
2.6.19 InversePoisson. 逆泊松分布.....	74
2.6.20 InverseHyperGeometric. 逆超几何分布.....	74
2.6.21 InverseLogNormal. 逆对数正态分布.....	75
2.6.22 InverseBinomial. 逆二项分布.....	75
2.6.23 InverseChiSquared. 逆卡方分布.....	75
2.6.24 InverseCauchy. 逆柯西分布.....	75
2.6.25 InverseLogistic. 逆逻辑分布.....	76
2.6.26 InversePascal. 逆帕斯卡分布.....	76
2.6.27 InverseZipf. 逆齐普夫分布.....	76
2.6.28 InverseWeibull. 逆威布尔分布.....	76
2.6.29 InverseNormal. 逆正态分布.....	77
2.6.30 InverseExponential. 逆指数分布.....	77
2.6.31 InverseGamma. 逆伽玛分布.....	77
2.6.32 Pascal. 帕斯卡分布.....	77
2.6.33 Zipf. 齐普夫分布.....	78
2.6.34 RandomBetween. 区间随机数.....	79
2.6.35 Triangular. 三角形分布.....	80
2.6.36 RandomPolynomial. 随机多项式.....	81

2.6.37	RandomBinomial. 随机二项分布数.....	81
2.6.38	RandomPointIn. 随机内点.....	82
2.6.39	Weibull. 威布尔分布.....	82
2.6.40	Normal. 正态分布.....	83
2.6.41	RandomNormal. 正态分布随机数.....	83
2.6.42	Exponential. 指数分布.....	84
2.6.43	nPr. 组合数.....	84
2.6.44	Gamma. 伽玛分布.....	85
2.7	Function. 函数与微积分.....	86
2.7.1	PartialFractions. 部分分式.....	86
2.7.2	ParametricDerivative. 参数导数.....	86
2.7.3	Degree. 次数.....	86
2.7.4	Derivative. 导数.....	87
2.7.5	Iteration. 迭代.....	88
2.7.6	IterationList. 迭代列表.....	89
2.7.7	NIntegral. 定积分.....	90
2.7.8	Polynomial. 多项式函数.....	90
2.7.9	NInvert. 反函数.....	91
2.7.10	Denominator. 分母.....	91
2.7.11	Numerator. 分子.....	92
2.7.12	ComplexRoot. 复数根.....	92
2.7.13	InflectionPoint (TurningPoint). 拐点.....	93
2.7.14	Normalize. 归一化.....	93
2.7.15	Function. 函数.....	93
2.7.16	Integral. 积分.....	95
2.7.17	IntegralBetween. 积分介于.....	96
2.7.18	Limit. 极限.....	97
2.7.19	Extremum. 极值点.....	97
2.7.20	Asymptote. 渐近线.....	98
2.7.21	SolveODE. 解常微分方程.....	98
2.7.22	NSolveODE. 解常微分方程组.....	100
2.7.23	RectangleSum. 矩形法则.....	102
2.7.24	Root. 零点.....	102
2.7.25	Roots. 零值点.....	103
2.7.26	RootList. 零值点列.....	104
2.7.27	PathParameter. 路径值.....	104
2.7.28	OsculatingCircle. 密切圆.....	105
2.7.29	SVD. 奇异值分解.....	106
2.7.30	Curvature. 曲率.....	107
2.7.31	CurvatureVector. 曲率向量.....	107
2.7.32	Curve (CurveCartesian). 曲线.....	108
2.7.33	TrigCombine. 三角式合并.....	108
2.7.34	TrigSimplify. 三角式化简.....	109
2.7.35	TrigExpand. 三角式展开.....	110

2.7.36	UpperSum. 上和.....	111
2.7.37	NDerivative. 数值导数.....	112
2.7.38	TaylorPolynomial. 泰勒公式.....	112
2.7.39	TrapezoidalSum. 梯形法则.....	112
2.7.40	Coefficients. 系数列表.....	113
2.7.41	LowerSum. 下和.....	113
2.7.42	SlopeField. 斜率场.....	114
2.7.43	Spline. 样条曲线.....	115
2.7.44	Factors. 因式.....	115
2.7.45	ImplicitCurve. 隐式曲线.....	116
2.7.46	ImplicitDerivative. 隐式微分.....	117
2.7.47	LimitAbove. 右极限.....	117
2.7.48	LeftSum. 左和.....	118
2.7.49	LimitBelow. 左极限.....	118
2.8	Geometry. 几何.....	118
2.8.1	Radius. 半径.....	118
2.8.2	envelope. 包络.....	119
2.8.3	Difference. 差异.....	119
2.8.4	Length. 长度.....	119
2.8.5	PerpendicularLine (OrthogonalLine). 垂线.....	121
2.8.6	Vertex. 顶点.....	123
2.8.7	Polygon. 多边形.....	124
2.8.8	Direction. 方向向量.....	125
2.8.9	AffineRatio. 仿射比 $\lambda$ .....	126
2.8.10	RigidPolygon. 刚体多边形.....	126
2.8.11	Locus. 轨迹.....	126
2.8.12	LocusEquation. 轨迹方程.....	127
2.8.13	Arc. 弧线.....	128
2.8.14	CrossRatio. 交比.....	129
2.8.15	Intersect. 交点.....	129
2.8.16	AngleBisector. 角平分线.....	131
2.8.17	Distance. 距离.....	132
2.8.18	Angle. 角度.....	133
2.8.19	Area. 面积.....	135
2.8.20	Point. 描点.....	135
2.8.21	PointIn. 内点.....	136
2.8.22	InteriorAngles. 内角.....	137
2.8.23	Tangent. 切线.....	137
2.8.24	ClosestPointRegion. 区域内最近点.....	139
2.8.25	Cubic. 三次曲线.....	139
2.8.26	CircumcircularSector. 三点扇形.....	140
2.8.27	CircumcircularArc. 三点圆弧.....	140
2.8.28	TriangleCurve. 三角曲线.....	140
2.8.29	TriangleCenter. 三角形中心.....	141

2.8.30	Trilinear. 三线坐标点.....	142
2.8.31	Sector. 扇形.....	142
2.8.32	Ray. 射线.....	143
2.8.33	ArePerpendicular. 是否垂直.....	143
2.8.34	AreConcurrent. 是否共点.....	144
2.8.35	AreCollinear. 是否共线.....	144
2.8.36	AreConcyclic. 是否共圆.....	144
2.8.37	AreParallel. 是否平行.....	145
2.8.38	AreCongruent. 是否全等.....	145
2.8.39	AreEqual. 是否相等.....	145
2.8.40	IsTangent. 是否相切.....	146
2.8.41	Segment. 线段.....	146
2.8.42	IntersectPath. 相交路径.....	146
2.8.43	Slope. 斜率.....	147
2.8.44	Centroid. 形心.....	147
2.8.45	CircularArc. 圆弧.....	148
2.8.46	CircularSector. 圆扇形.....	148
2.8.47	Circumference. 圆周长.....	148
2.8.48	PolyLine. 折线.....	148
2.8.49	Prove. 证明.....	149
2.8.50	ProveDetails. 证明过程.....	149
2.8.51	Line. 直线.....	150
2.8.52	PerpendicularBisector (LineBisector). 中垂线.....	150
2.8.53	Midpoint. 中点.....	151
2.8.54	Barycenter. 重心.....	151
2.8.55	Perimeter. 周长.....	152
2.8.56	ClosestPoint. 最近点.....	152
2.9	Transformation. 几何变换.....	153
2.9.1	Reflect (Mirror). 对称.....	153
2.9.2	Translate. 平移.....	154
2.9.3	Shear. 切变.....	154
2.9.4	Stretch. 伸缩.....	155
2.9.5	Dilate (Enlarge). 位似.....	155
2.9.6	Rotate. 旋转.....	155
2.10	ScriptingCommands. 脚本指令.....	156
2.10.1	Button. 按钮.....	156
2.10.2	PlaySound. 播放声音.....	157
2.10.3	ExportImage. 导出图片.....	162
2.10.4	ZoomIn. 放大.....	163
2.10.5	SetValue. 赋值.....	164
2.10.6	Checkbox. 复选框.....	164
2.10.7	CopyFreeObject. 复制自由对象.....	165
2.10.8	AttachCopyToView. 附加副本.....	165
2.10.9	UpdateConstruction. 更新作图.....	166



2. 10. 10 Turtle. 海龟.....	166
2. 10. 11 TurtleBack. 后退.....	167
2. 10. 12 Slider. 滑动条.....	167
2. 10. 13 ParseToFunction. 解析为函数.....	167
2. 10. 14 ParseToNumber. 解析为数.....	168
2. 10. 15 StartRecord. 开始记录.....	168
2. 10. 16 TurtleDown. 落笔.....	168
2. 10. 17 Pan. 平移视图.....	168
2. 10. 18 StartAnimation. 启动动画.....	168
2. 10. 19 StartLogging. 启动日志.....	169
2. 10. 20 TurtleForward. 前进.....	169
2. 10. 21 Delete. 删除.....	169
2. 10. 22 SetBackgroundColor. 设置背景颜色.....	169
2. 10. 23 SetDecoration. 设置标记.....	170
2. 10. 24 SetLabelMode. 设置标签模式.....	171
2. 10. 25 SetCaption. 设置标题.....	171
2. 10. 26 SetPointSize. 设置点径.....	171
2. 10. 27 SetPointStyle. 设置点型.....	172
2. 10. 28 SetDynamicColor. 设置动态颜色.....	172
2. 10. 29 SetFixed. 设置对象锁定.....	173
2. 10. 30 SetPerspective. 设置格局.....	173
2. 10. 31 SetTrace. 设置跟踪.....	174
2. 10. 32 SetTooltipMode. 设置工具提示模式.....	174
2. 10. 33 SetActiveView. 设置活动视图.....	175
2. 10. 34 SetVisibleInView. 设置可见性.....	175
2. 10. 35 SetViewDirection. 设置视图方向.....	175
2. 10. 36 SetFilling. 设置填充.....	176
2. 10. 37 SetLayer. 设置图层.....	176
2. 10. 38 SetLevelOfDetail. 设置细节级别.....	176
2. 10. 39 SetConditionToShowObject. 设置显示条件.....	176
2. 10. 40 SetLineThickness. 设置线径.....	176
2. 10. 41 SetLineStyle. 设置线型.....	177
2. 10. 42 SetColor. 设置颜色.....	177
2. 10. 43 SetSeed. 设置种子.....	178
2. 10. 44 SetSpinSpeed. 设置转速.....	178
2. 10. 45 SetCoords. 设置坐标.....	178
2. 10. 46 SetAxesRatio. 设置坐标轴比例.....	178
2. 10. 47 InputBox (Textfield). 输入框.....	179
2. 10. 48 DataFunction. 数据函数.....	179
2. 10. 49 ZoomOut. 缩小.....	179
2. 10. 50 TurtleUp. 抬笔.....	179
2. 10. 51 StopLogging. 停止日志.....	180
2. 10. 52 GetTime. 系统时间.....	180
2. 10. 53 ShowLabel. 显示标签.....	180

2. 10. 54 ShowLayer. 显示图层.....	181
2. 10. 55 ShowGrid. 显示网格.....	181
2. 10. 56 ShowAxes. 显示坐标轴.....	181
2. 10. 57 SelectObjects. 选择.....	181
2. 10. 58 HideLayer. 隐藏图层.....	182
2. 10. 59 TurtleRight. 右转.....	182
2. 10. 60 ReadText. 阅读文本.....	182
2. 10. 61 RunClickScript. 运行单击脚本.....	183
2. 10. 62 RunUpdateScript. 运行更新脚本.....	183
2. 10. 63 Execute. 执行.....	183
2. 10. 64 CenterView. 中心定位.....	184
2. 10. 65 Repeat. 重复.....	184
2. 10. 66 Rename. 重命名.....	184
2. 10. 67 TurtleLeft. 左转.....	184
2. 11 DiscreteMath. 离散数学.....	185
2. 11. 1 DelaunayTriangulation. Delaunay 三角网.....	185
2. 11. 2 Voronoi. Voronoi 图.....	185
2. 11. 3 TravelingSalesman. 旅行商问题.....	185
2. 11. 4 ConvexHull. 凸包.....	186
2. 11. 5 ShortestDistance. 最短距离.....	186
2. 11. 6 MinimumSpanningTree. 最小生成树.....	186
2. 12 List. 列表.....	187
2. 12. 1 Flatten. 扁平列表.....	187
2. 12. 2 Union. 并集.....	188
2. 12. 3 Insert. 插入.....	188
2. 12. 4 Product. 乘积.....	188
2. 12. 5 PointList. 点列.....	189
2. 12. 6 Join. 合并.....	190
2. 12. 7 Unique. 互异.....	190
2. 12. 8 Intersection. 交集.....	190
2. 12. 9 Reverse. 逆序排列.....	191
2. 12. 10 Frequency. 频数列表.....	191
2. 12. 11 TiedRank. 平秩列表.....	193
2. 12. 12 Remove. 去除.....	193
2. 12. 13 RemoveUndefined. 去除未定义对象.....	193
2. 12. 14 Sort. 升序排列.....	194
2. 12. 15 RandomElement. 随机元素.....	195
2. 12. 16 IndexOf. 索引.....	195
2. 12. 17 Take. 提取.....	196
2. 12. 18 Sequence. 序列.....	197
2. 12. 19 OrdinalRank. 序数列表.....	200
2. 12. 20 SelectedIndex. 选定索引.....	200
2. 12. 21 SelectedElement. 选定元素.....	200
2. 12. 22 Zip. 映射.....	201

2. 12. 23 Element. 元素.....	201
2. 12. 24 Append. 追加.....	202
2. 12. 25 Classes. 组限.....	203
2. 12. 26 Last. 最后元素.....	203
2. 12. 27 First. 最前元素.....	204
2. 13 Logical. 逻辑.....	205
2. 13. 1 Relation. 关系.....	205
2. 13. 2 If. 如果.....	205
2. 13. 3 IsInteger. 是否为整数.....	208
2. 13. 4 IsDefined (Defined). 是否已定义.....	208
2. 13. 5 IsInRegion. 是否在区域内.....	208
2. 13. 6 CountIf. 条件计数.....	209
2. 13. 7 KeepIf. 条件子列.....	209
2. 14 Statistics. 统计.....	210
2. 14. 1 Mad. Mad.....	210
2. 14. 2 Mean. Mean.....	210
2. 14. 3 SigmaXX. 横坐标平方和.....	210
2. 14. 4 SigmaXY. 横纵坐标乘积和.....	211
2. 14. 5 SigmaYY. 纵坐标平方和.....	211
2. 14. 6 Spearman. 秩相关系数.....	212
2. 14. 7 Stdev. Sstdev.....	212
2. 14. 8 Stdevp. Stdevp.....	212
2. 14. 9 Sxx. Sxx.....	212
2. 14. 10 Sxy. Sxy.....	213
2. 14. 11 Syy. Syy.....	213
2. 14. 12 TTest. T 检验.....	213
2. 14. 13 Percentile. 百分位数.....	214
2. 14. 14 SD. 标准差.....	214
2. 14. 15 ZProportionEstimate. 单比例 Z 估计.....	215
2. 14. 16 ZProportionTest. 单比例 Z 检验.....	215
2. 14. 17 TMeanEstimate. 单均值 T 估计.....	215
2. 14. 18 ZMeanEstimate. 单均值 Z 估计.....	216
2. 14. 19 ZMeanTest. 单均值 Z 检验.....	216
2. 14. 20 Q3. 第三四分位数.....	216
2. 14. 21 Q1. 第一四分位数.....	217
2. 14. 22 HarmonicMean. 调和平均数.....	217
2. 14. 23 FitLog. 对数拟合.....	217
2. 14. 24 FitPoly. 多项式拟合.....	218
2. 14. 25 Variance. 方差.....	218
2. 14. 26 ANOVA. 方差分析.....	219
2. 14. 27 SDX. 横坐标标准差.....	219
2. 14. 28 MeanX. 横坐标平均数.....	219
2. 14. 29 GeometricMean. 几何平均数.....	219
2. 14. 30 RootMeanSquare. 均方根.....	219

2. 14. 31	ChiSquaredTest. 卡方检验.....	220
2. 14. 32	RSquare. 可决系数 R 方.....	220
2. 14. 33	FitLogistic. 逻辑斯蒂曲线拟合.....	221
2. 14. 34	FitPow. 幂函数拟合.....	221
2. 14. 35	Fit. 拟合曲线.....	221
2. 14. 36	FitLineX. 拟合直线 X.....	222
2. 14. 37	FitLine (FitLineY). 拟合直线 Y.....	222
2. 14. 38	TTestPaired. 配对 T 检验.....	223
2. 14. 39	Mad. 平均绝对偏差.....	223
2. 14. 40	Mean. 平均数.....	223
2. 14. 41	FitGrowth. 生长曲线拟合.....	224
2. 14. 42	ZProportion2Estimate. 双样本比例 Z 估计.....	224
2. 14. 43	ZProportion2Test. 双样本比例 Z 检验.....	225
2. 14. 44	TMean2Estimate. 双样本均值 T 估计.....	225
2. 14. 45	ZMean2Estimate. 双样本均值 Z 估计.....	225
2. 14. 46	ZMean2Test. 双样本均值 Z 检验.....	226
2. 14. 47	TTest2. 双总体 T 检验.....	226
2. 14. 48	Shuffle. 随机排列.....	226
2. 14. 49	SumSquaredErrors. 误差平方和.....	227
2. 14. 50	CorrelationCoefficient (PMCC). 相关系数.....	227
2. 14. 51	Covariance. 协方差.....	227
2. 14. 52	Sample. 样本.....	228
2. 14. 53	SampleSD. 样本标准差.....	229
2. 14. 54	SampleSDX. 样本点横坐标标准差.....	229
2. 14. 55	SampleSDY. 样本点纵坐标标准差.....	229
2. 14. 56	SampleVariance. 样本方差.....	230
2. 14. 57	FitImplicit. 隐函数拟合.....	230
2. 14. 58	FitSin. 正弦拟合.....	230
2. 14. 59	FitExp. 指数拟合.....	231
2. 14. 60	Median. 中位数.....	231
2. 14. 61	Mode. 众数.....	232
2. 14. 62	Sum. 总和.....	232
2. 14. 63	SDY. 纵坐标标准差.....	233
2. 14. 64	MeanY. 纵坐标平均数.....	233
2. 15	Chart. 图表.....	234
2. 15. 1	StickGraph. 棒图.....	234
2. 15. 2	ResidualPlot. 残差图.....	235
2. 15. 3	DotPlot. 点阵图.....	235
2. 15. 4	StepGraph. 阶梯图.....	236
2. 15. 5	StemPlot. 茎叶图.....	237
2. 15. 6	ContingencyTable. 列联表.....	238
2. 15. 7	FrequencyTable. 频数表.....	240
2. 15. 8	FrequencyPolygon. 频数多边形.....	242
2. 15. 9	BarChart. 条形图.....	243

2. 15. 10	BoxPlot. 箱线图.....	244
2. 15. 11	NormalQuantilePlot. 正态分位数图.....	245
2. 15. 12	Histogram. 直方图.....	246
2. 15. 13	HistogramRight. 直方图右.....	247
2. 16	Text. 文本.....	248
2. 16. 1	TableText. 表格文本.....	248
2. 16. 2	FractionText. 分数文本.....	251
2. 16. 3	SurdText. 根式文本.....	251
2. 16. 4	FormulaText (LaTeX). 公式文本.....	252
2. 16. 5	ScientificText. 科学计数法.....	253
2. 16. 6	ContinuedFraction. 连分式.....	253
2. 16. 7	VerticalText. 竖排文本.....	254
2. 16. 8	UnicodeToText. 统一码转换为文本.....	254
2. 16. 9	UnicodeToLetter. 统一码转换为字母.....	255
2. 16. 10	Text. 文本.....	255
2. 16. 11	TextToUnicode. 文本转换为统一码.....	256
2. 16. 12	Ordinal. 序数.....	256
2. 16. 13	RotateText. 旋转文本.....	257
2. 16. 14	LetterToUnicode. 字母转换为统一码.....	257
2. 17	Vector&Matrix. 向量与矩阵.....	257
2. 17. 1	UnitPerpendicularVector (UnitOrthogonalVector). 单位法向量.....	257
2. 17. 2	Identity. 单位矩阵.....	258
2. 17. 3	UnitVector. 单位向量.....	259
2. 17. 4	PerpendicularVector (OrthogonalVector). 法向量.....	259
2. 17. 5	ReducedRowEchelonForm. 简化行梯阵式.....	261
2. 17. 6	MatrixRank. 矩阵的秩.....	261
2. 17. 7	Invert. 逆反.....	261
2. 17. 8	Dimension. 维度.....	262
2. 17. 9	Vector. 向量.....	263
2. 17. 10	Determinant. 行列式.....	263
2. 17. 11	ApplyMatrix. 应用矩阵.....	264
2. 17. 12	Transpose. 转置.....	264
2. 18	Optimization. 优化指令.....	265
2. 18. 1	Maximize. 最大值点.....	265
2. 18. 2	Minimize. 最小值点.....	265
2. 19	Conic. 圆锥曲线.....	266
2. 19. 1	LinearEccentricity. 半焦距.....	266
2. 19. 2	Semicircle. 半圆.....	266
2. 19. 3	SemiMinorAxisLength (SecondAxisLength). 副半轴长.....	267
2. 19. 4	MinorAxis (SecondAxis). 副轴.....	267
2. 19. 5	ConjugateDiameter (Diameter). 共轭直径.....	267
2. 19. 6	Polar. 极线.....	268
2. 19. 7	Parameter. 焦参数.....	268
2. 19. 8	Focus. 焦点.....	268

2. 19. 9 Eccentricity. 离心率.....	269
2. 19. 10 Incircle. 内切圆.....	269
2. 19. 11 Parabola. 抛物线.....	269
2. 19. 12 Hyperbola. 双曲线.....	269
2. 19. 13 Ellipse. 椭圆.....	270
2. 19. 14 Circle. 圆周.....	271
2. 19. 15 Conic. 圆锥曲线.....	272
2. 19. 16 Center (Centre). 中心.....	272
2. 19. 17 Axes. 轴线.....	273
2. 18. 18 SemiMajorAxisLength (FirstAxisLength). 主半轴长.....	273
2. 19. 19 MajorAxis (FirstAxis). 主轴.....	273
2. 19. 20 Directrix. 准线.....	274
2. 20 CASSpecific. 运算区专属指令.....	274
2. 20. 1 MixedNumber. 带分数.....	275
2. 20. 2 GroebnerDegRevLex. 分次反字典序 Groebner 基.....	275
2. 20. 3 GroebnerLexDeg. 分次字典序 Groebner 基.....	275
2. 20. 4 CSolve. 复数解.....	276
2. 20. 5 CSolutions. 复数解集.....	276
2. 20. 6 CFactor. 复数域因式分解.....	276
2. 20. 7 CIFactor. 复无理数域因式分解.....	277
2. 20. 8 SolveCubic. 解三次多项式.....	277
2. 20. 9 Numeric. 近似数.....	277
2. 20. 10 Laplace. 拉普拉斯变换.....	278
2. 20. 11 InverseLaplace. 拉普拉斯逆变换.....	278
2. 20. 12 Substitute. 替换.....	279
2. 20. 13 Eliminate. 消元.....	279
2. 20. 14 Rationalize. 有理化.....	279
2. 20. 15 ToExponential. 转换为指数形式.....	280
2. 20. 16 GroebnerLex. 字典序 Groebner 基.....	280
3 中文指令名速查 (中文名称; 英文名称; 页码) .....	281
4 英文指令名速查 (英文名称; 中文名称; 页码) .....	287
5 软件指令列表截图.....	293

# 1 函数与运算

GeoGebra 是以数学计算为基础，进而引申到几何等其他领域。在“指令栏”可以使用下面的内置函数和操作创建数值、坐标和方程等。逻辑运算符和函数见“1.3 布朗运算”。

注：函数在指令栏中输入需要使用小括弧，且一定不能在函数名称和括弧间插入空格。

## 1.1 基本运算

$e$ ：欧拉数。键盘输入：(Alt)+(e)。

$i$ ：虚数单位。键盘输入：(Alt)+(i)。

$\pi$ ：圆周率。键盘输入：(Alt)+(p)或输入“pi”。

$^\circ$ ：度数单位符号。键盘输入：(Alt)+(o)或者使用中文输入法输入“°”。

$+$ ：Addition，相加。

$/$ ：Division，相除。

$^$ 或上标数值 ( $x^2$  或  $x^2$ )：Exponentiation，取幂（指数运算）。

$!$ ：Factorial，阶乘。

$-$ ：Subtraction，相减。

$()$ ：Parentheses 圆括弧插值（插入）。

$*$ 或键盘输入(Space)：Multiplication，相乘。

$\cdot$ 或键盘输入(Space)：Scalar product，标量（实量）积。

$\otimes$ ：Vector product，向量积。

以上基本计算操作符号，都是英文半角，可以使用键盘输入，也可以点击指令编辑栏后边的“ $\alpha$ ”按钮辅助输入。还可以使用软件虚拟键盘输入。

注：“ $2^3$ ”与“ $2^3$ ”的结果相同。“ $x*y$ ”等同于“ $x y$ ”（空格表示乘法运算）。

## 1.2 函数

GeoGebra 的外显函数（指令帮助中）有 54 个，还有几个内置函数或计算。以下列出函数都是计算圆括弧内数值、计算表达式的函数值。函数只能使用英文半角格式。双击指令帮助中的函数，函数名自动添加到指令栏。以下函数以英文字母顺序排列，包括其语法格式、英文含义和中文含义。

1. **abs(x)**：Absolute value，绝对值。

2. **acos(x) 或 arccos(x)**：Arc cosine，反余弦函数。

3. **acosd(x) 或 arccosd(x)**：Arc cosine，反余弦( $\cos^{-1}$ ) (以度为单位)。

4. **acosh(x) 或 arccosh(x)**：Antihyperbolic cosine，反双曲余弦函数。

5. **arg(x)**：Argument，复数的辐角： $z=r*(\cos \theta +i\sin \theta)$ ， $r$  是  $z$  的模，即： $r=|z|$ ； $\theta$  是  $z$  的辐角，记作： $\theta =\arg(z)$ 。

6. **asin(x) 或 arcsin(x)**：Arc sine，反正弦函数。

7. **asind(x) 或 arcsind(x)**：Arc sine，反正弦( $\sin^{-1}$ ) (以度为单位)。

8. **asinh(x) 或 arcsinh(x)**：Antihyperbolic sine，反双曲正弦函数。

9. **atan(x) 或 arctan(x)**：Arc tangent，反正切函数（返回值在 $-\pi/2$ 与 $\pi/2$ 之间）。

10. **atand(x) 或 arctand(x)**：Arc tangent，反正切( $\tan^{-1}$ ) (以度为单位)。

11. **atanh(x) 或 arctanh(x)**: Antihyperbolic tangent, 反双曲正切函数。
12. **atan2(y, x)**: Arctangent, 反正切函数 (返回值在  $-\pi$  与  $\pi$  之间)。对于任意不同时等于 0 的实参数  $x$  和  $y$ ,  $\text{atan2}(y, x)$  所表达的意思是坐标原点为起点, 指向  $(x, y)$  的射线在坐标平面上与  $x$  轴正方向之间的角的角度。当  $y > 0$  时, 射线与  $x$  轴正方向所得的角的角度指的是  $x$  轴正方向绕逆时针方向到达射线旋转的角的角度; 而当  $y < 0$  时, 射线与  $x$  轴正方向所得的角的角度指的是  $x$  轴正方向绕顺时针方向达到射线旋转的角的角度。
13. **beta(a, b)**:  $B(a, b)$ , 贝塔函数 ( $\beta$  函数)。
14. **beta(a, b, x)**:  $B(x; a, b)$ , 不完全贝塔函数。
15. **betaRegularized(a, b, x)**:  $I(x; a, b)$ , 正则化不完全贝塔函数。
16. **cbrt(x)**: Cubicroot, 三次方根、立方根。
17. **ceil(x)**: Least integer greater than or equal, “向上取整”, 返回大于或者等于指定表达式的最小整数。
18. **conjugate(x)**: Conjugate, 共轭函数。
19. **cos(x)**: 余弦函数。
20. **cosec(x) 或 csc(x)**: Cosecant, 余割函数。  $\text{cosec} \angle A = c/a$  (斜边/对边)。
21. **cosech(x) 或 csch(x)**: Hyperbolic cosecant, 双曲余割函数。
22. **cosh(x)**: Hyperbolic cosin, 双曲余弦函数。  $\cosh(x) = \frac{e^x + e^{-x}}{2}$ 。
23. **cosIntegral(x)**: Cosine Integral, 余弦积分。
24. **cot(x)**: Cotangent, 余切函数。  $\cot \angle A = c/a$  (邻边/对边)。
25. **coth(x)**: Hyperbolic cotangent, 双曲余切函数。
26. **exp(x) 或  $e^x$** : Exponential function, 指数函数。
27. **erf(x)**: Gaussian Error Function, 高斯误差函数。
28. **expIntegral(x)**: Exponential Integral, 指数积分。  $E_1(z) = \int e^{-zt}/t dt (1 \sim +\infty)$ 。
29. **fractionalPart(x)**: Fractional Part, 小数函数。如:  $\text{fractionalPart}(6/5) = 0.2$ 。
30. **floor(x)**: Greatest integer less than or equal, “向下取整”, 或者说“向下舍入”, 即取不大于  $x$  的最大整数。例如:  $x = 3.14$ ,  $\text{floor}(x) = 3$ ;  $y = 9.99999$ ,  $\text{floor}(y) = 9$ 。
31. **gamma(x)**:  $\Gamma(x)$ , 伽玛函数 (Gamma 函数), 也叫欧拉第二积分, 是阶乘函数在实数与复数上扩展的一类函数。
32. **gamma(a, x)**:  $\gamma(a, x)$ , (低阶) 不完全伽玛函数。
33. **gammaRegularized(a, x)**:  $P(a, x) = \gamma(a, x) / \Gamma(a)$ , (低阶) 正则化不完全伽玛函数。
34. **imaginary(x)**: Imaginary, 虚值函数 (参见实值函数)。
35. **ld(x)**: Logarithm to base2, 以 2 为底的对数函数。
36. **lg(x)**: Logarithm to base10, 以 10 为底的对数函数。
37. **ln(x) 或 log(x)**: Logarithm, 自然对数, 底数为  $e$ 。
38. **log(b, x)**: Logarithm of  $x$  to base  $b$ , 以  $b$  为底  $x$  的对数。
39. **nroot(x, n)**: 求  $x$  的  $n$  次方根。
40. **polygamma(m, x)**: Polygamma function, 多项伽玛函数。
41. **psi(x)**: Digamma function, Digamma 函数 (伽玛函数的对数的导数),  $\Psi(x)$ 。
42. **real(x)**: 实值函数 (函数值是“实数”, 不可以取虚数或  $\pm\infty$ ), 复数的实部数值,  $\text{Real}(2+3*i)$  的结果是 2。
43. **random(a, b)**: Random number between  $a$  and  $b$ ,  $a$  与  $b$  之间的随机数。
44. **round(x)**: Round, 四舍五入。
45. **sec(x)**: Secant, 正割函数。  $\sec \angle A = c/b$  (斜边/邻边)。



46. **sech(x)**: Hyperbolic secant, 双曲正割函数。

47. **sgn(x)** 或 **sign(x)**: Sign, 符号 (x 为正数返回 1, 负数返回-1, 零返回 0)。

48. **sin(x)**: Sine, 正弦函数。

49. **sinh(x)**: Hyperbolic sine, 双曲正弦函数。  $\sinh(x) = \frac{e^x - e^{-x}}{2}$ 。

50. **sinIntegral(x)**: Sine Integral, 正弦积分。缩写 Si(z),  $Si(z) = \int \sin(t)/t dt$ , 区间(0, z) 上的定积分。

51. **sqrt(x)**: Square root, 平方 (二次方) 根。

52. **tan(x)**: Tangent, 正切函数。

53. **tanh(x)**: Hyperbolic tangent, 双曲正切函数。  $\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$ 。

54. **zeta(x)**:  $\zeta(x)$ , 黎曼 zeta 函数。  $ZETA(z) = \sum(1/k^z, k, 1, \infty)$ 。

以下几个非严格的函数, 是 GeoGebra 内部约定的计算或变量。

55. **x(x)**: x-coordinate, 点对象 x 的横坐标值。x(A) 返回点 A 的横坐标值。

56. **xAxis**: x 轴。非严格意义的函数, 不需要变量, 相当于直线 y=0。

57. **y(x)**: y-coordinate, 点对象 x 的纵坐标值。Y(A) 返回点 A 的纵坐标。

58. **yAxis**: y 轴。非严格意义的函数, 不需要变量, 相当于直线 x=0。

59. **z(x)**: z-coordinate, 点对象 x 的 z 坐标值。Z(A) 返回点 A 的 z 坐标值。

60. **zAxis**: z 轴。非严格意义的函数, 不需要变量, 相当于 z 轴直线。

另外, 设有列表“表 1={1, 2, 3, a, D}”, “表 1(3)” 返回列表的第三号元素“3”。此为内定的特殊计算。

**案例**: “Conjugate(17+3\*i)” 得到“17+3\*i”的共轭复数“-3i+17”。

详见“复数”指令。

### 1.3 布尔运算

布尔值就是“true”和“false”, 对于产生这两个值的运算, 就是布尔运算, 也叫逻辑运算或布尔运算。“布尔值”也译作“布朗值”。

在输入栏中键入“a=true”或者“b=false”后回车, 就能得到两个可以用到 ggb 中的布尔值了。

在绘图区, 可以把布尔值放入到复选框内 (参见复选框工具), 在代数区选定一个布尔值以后, 可以使用键盘箭头键改变值 (参见手动动画)。

**注**: 还可以使用诸如 0 和 1 这样的数字作为布尔值, 可以在复选框中用 0 或 1 来停止或启动动画。在这种情况下, 如果还有其他动画滑块, 则动画按钮仅显示在绘图区。

可以使用指令栏右侧下拉选单或者键盘输入并使用以下布尔值。

在“指令栏”输入时, “菜单选择”是按 α “按钮”后选择。“键盘输入”是半角英文输入。

**案例**: 假设绘图区域中有两条直线 a 和 b, 在指令栏输入“a//b”, 就会在代数区返回一个值携带标签的布尔值 (true 或 false)

“运算与函数”中的运算符和函数, 可以嵌套组成复合运算, 在“指令栏”和“运算区”均可用。在对象的“脚本”中也可以使用。绝大多数指令同时适用于各版 GeoGebra 中。

### 布朗运算符号

意义	菜单选择	键盘输入	例子	适于对象类型
等于	$\doteq$	<code>==</code>	$a \doteq b$ 或 $a == b$	数值、点、直线、圆锥曲线
不等于	$\neq$	<code>!=</code>	$a \neq b$ 或 $a != b$	数值、点、直线、圆锥曲线
小于	$<$	<code>&lt;</code>	$a < b$	数值
大于	$>$	<code>&gt;</code>	$a > b$	数值
小于等于	$\leq$	<code>&lt;=</code>	$a \leq b$ 或 $a <= b$	数值
大于等于	$\geq$	<code>&gt;=</code>	$a \geq b$ 或 $a >= b$	数值
且	$\wedge$	<code>&amp;&amp;</code>	$a \wedge b$	布朗函数
或	$\vee$	<code>  </code>	$a \vee b$	布朗函数
非	$\neg$	<code>!</code>	$\neg a$ 或 $!a$	布朗函数
属于	$\in$		$a \in \langle \text{列表 } 1 \rangle$	数值 a, 列表<列表 1>
平行于	$//$		$a // b$	直线
垂直于	$\perp$		$a \perp b$	直线

## 2 命令

GeoGebra 的指令包括函数和命令两部分，可用于“指令栏”、“运算区”和“脚本”中。指令在不同的地点有时有不同的使用语法与适用性。某些指令只能用于“运算区”（参见“运算区指令”）。某些指令有系统工具图标，点击工具栏工具图标就执行该指令。而“脚本”指令，多数也未必是“脚本”中专属的。

GeoGebra 支持使用不同的语言（目前大约 81 种），指令帮助中列出的指令都按照本地语言排序。本指令汇编按照中文版顺序编辑，方便中文使用者查阅。在简体中文版中，输入英文和中文名称指令都可以被识别，在“运算区”输入的英文指令，只要格式正确，会自动变为本地化语言。指令的输入有提示功能，输入前 2 个字符就有其语法提示。中文名称经常因为改进而变动，只有符合规范的中文指令，才有输入提示。英文指令固化的比较好，且更适于国际化交流软件案例作品，故本汇编每个中文命令都同时携带了英文名称。

一个命令的运行结果可能会使用一个后跟等号(=)的标签作为名称。在下边的案例中，点的名称为 S。

**案例：**想得到两条直线  $g$  和  $h$  的交点 S，可以输入“S=交点( $g, h$ )”（参见“交点”指令）。

**注：**也能在对象的名称中使用下标：“ $A_1$ ”是输入“ $A_1$ ”，而“ $S_{AB}$ ”是使用“ $S_{\{AB\}}$ ”创建。这是 LaTeX 语法的一种运用。

官网的指令汇编经常是比较低的版本，3D 指令在 5.0 以上版本中开始使用，但官网帮助页总是滞后于软件。只有运算区被激活后，“运算区”指令才会在指令帮助中列出，并列在最后。

许多指令同时适用于指令栏和运算区，只是语法和得到的结果有所不同。

## 2.1 3D. 三维

在 GeoGebra 中，许多二维平面内的指令同样适用与三维空间，比如在空间的任意一个平面内构建点、线、圆以及与此类对象相关指令，在“3D 绘图区”也适用。本节以下指令只针对“3D 绘图区”专门指令，是 GeoGebra5.0 以上版本支持的指令，会随时变化更新。其名称就代表了其定义。

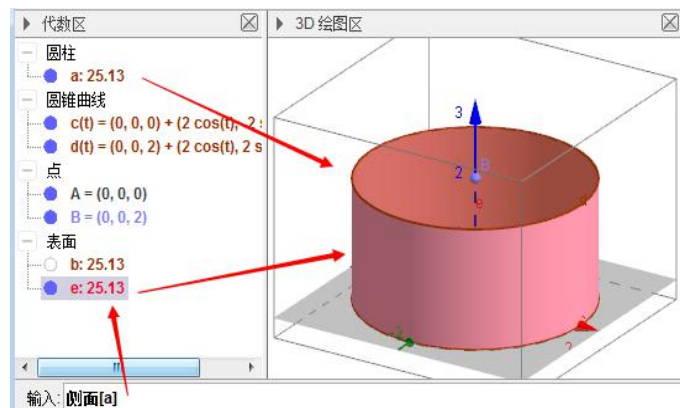
### 2.1.1 Side. 侧面

**Side**(**<Quadric>**); **侧面**(**<二次曲面>**)。

创建有限二次曲面的侧面。

**案例：**“侧面(圆柱)”创建已有圆柱 a 的曲面（侧面）。在代数区出现圆柱侧面面积，在 3D 区域绘制圆柱的侧面。

**注：**参见“上底”指令、“下底”指令和“底面”指令。




### 2.1.2 PerpendicularPlane. 垂直平面

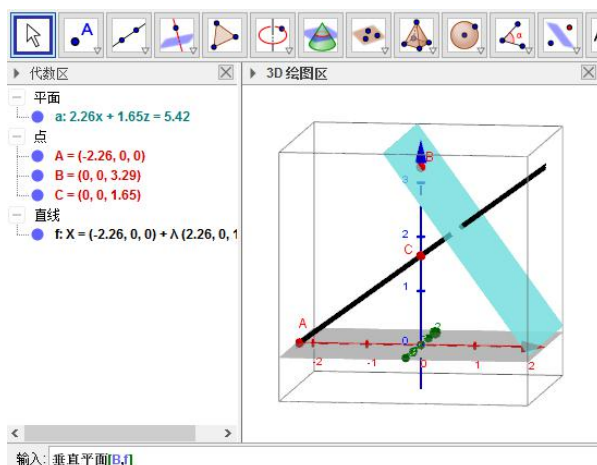
**PerpendicularPlane**(**<Point>**, **<Line>**); **垂直平面**(**<点>**, **<直线>**)。

创建过给定点垂直于给定直线的面。

**PerpendicularPlane**(**<Point>**, **<Vector>**); **垂直平面**(**<点>**, **<向量>**)。

创建过给定点垂直于给定向量的面。

**注：**参见  “垂直平面”工具。

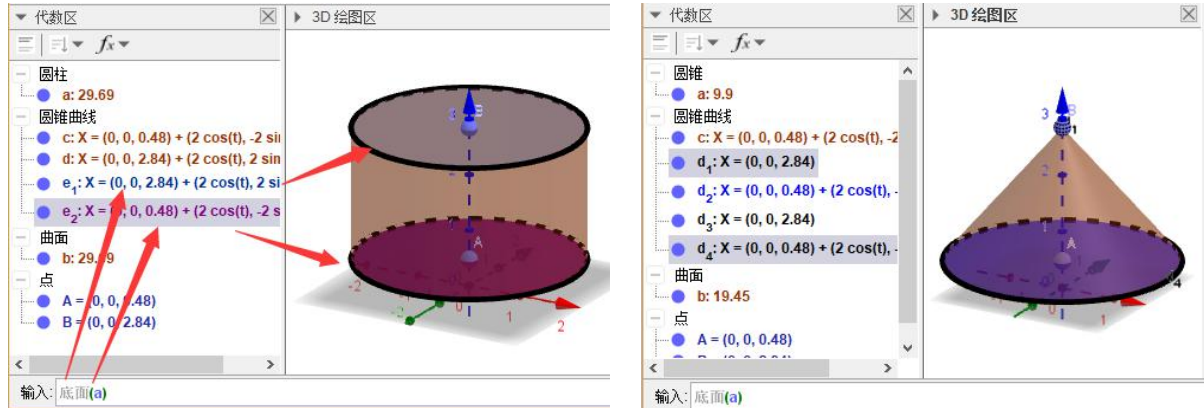


### 2.1.3 Ends. 底面

Ends(<Quadric>); 底面(<二次曲面>).

创建有限二次曲面的上底和下底。

案例：“底面(圆柱)”给出两个圆。代数区有解析式，绘图区有图形。“底面(圆锥)”给出一个圆和锥端(点)



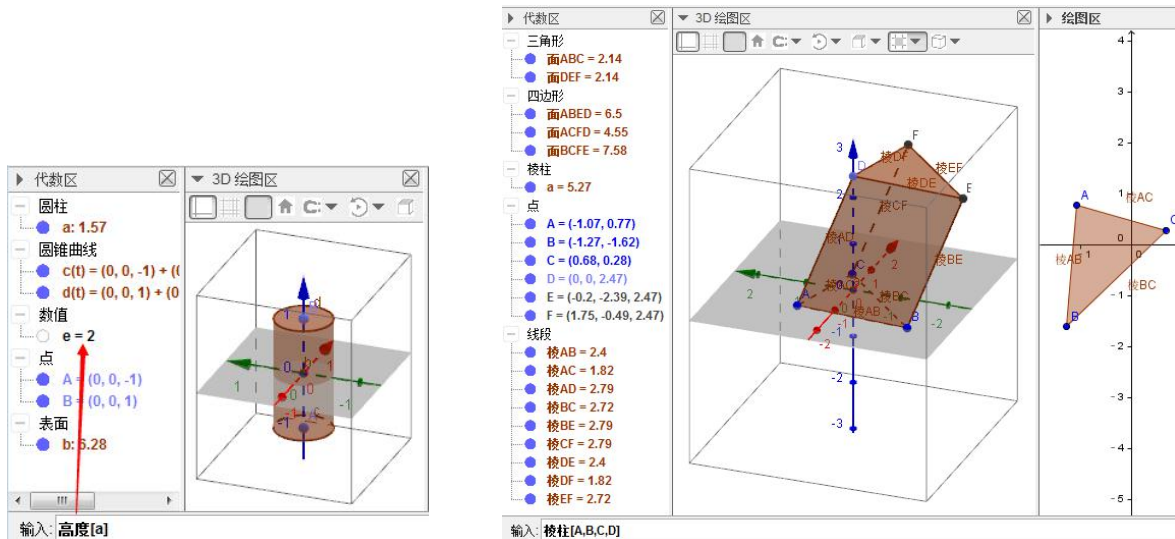
注：参见“上底”指令、“下底”指令和“侧面”指令。

### 2.1.4 Height. 高度

Height(<Solid>); 高度(<立体图形>).

计算给定立体图形的“定向”高度

案例：“高度(圆锥)”计算给定圆锥的“定向”高度；“高度(圆柱)”计算给定圆柱的“定向”高度；“高度(多面体)”计算给定多面体的“定向”高度。



### 2.1.5 Prism. 棱柱

Prism(<Point>, <Point>, ...); 棱柱(<点 1>, <点 2>, ...).

返回有给定点定义的棱柱。

案例：“棱柱(A, B, C, D)”创建下底为三角形 ABC，上底为 DEF 的棱柱，向量 AD、BE 和 CF 相等。

尽管是构造棱柱，但在实际操作中，大量与构造棱柱有关的对象都构造出来，所有构造出来的对象，都列在代数区，同时在绘图区和 3D 绘图区显示对象图形。

**Prism(<Polygon>, <Point>); 棱柱(<多边形>, <最高点>).**

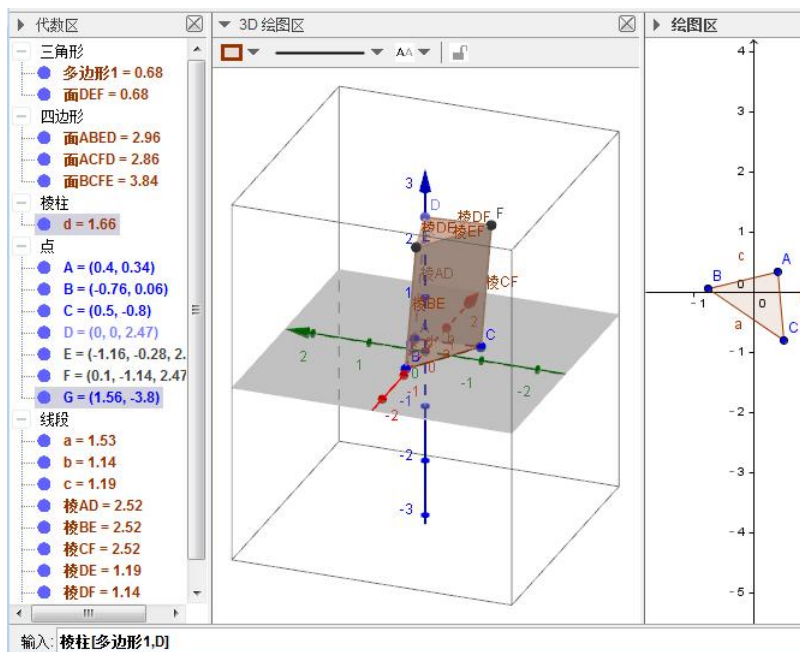
创建给定的多边形为下底且给定点为上底的第一个点的棱柱。

以上两个案例构造的棱柱都不一定是正棱柱。

**案例:** “棱柱(多边形 1, A)” 创建一个以 “多边形 1” 为底, 点 A 为上底点的棱柱。

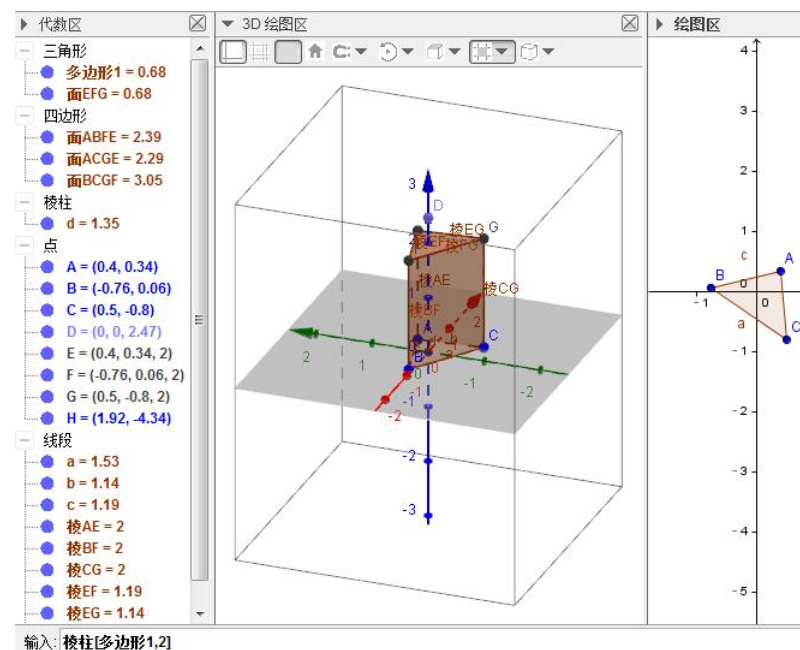
**Prism(<Polygon>, <Height value>), 棱柱(<多边形>, <高度>).**

创建一个多边形为底且给定高度的正棱柱。



**案例:** “棱柱(多边形 1, 2)” 创建底为 “多边形 1” 高度为 2 的正棱柱。

**注:** 参见 “棱柱” 和 “拉成柱体” 工具。

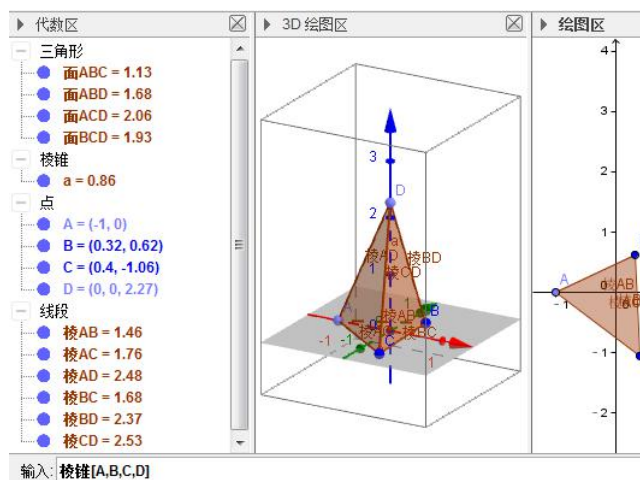


## 2.1.6 Pyramid. 棱锥

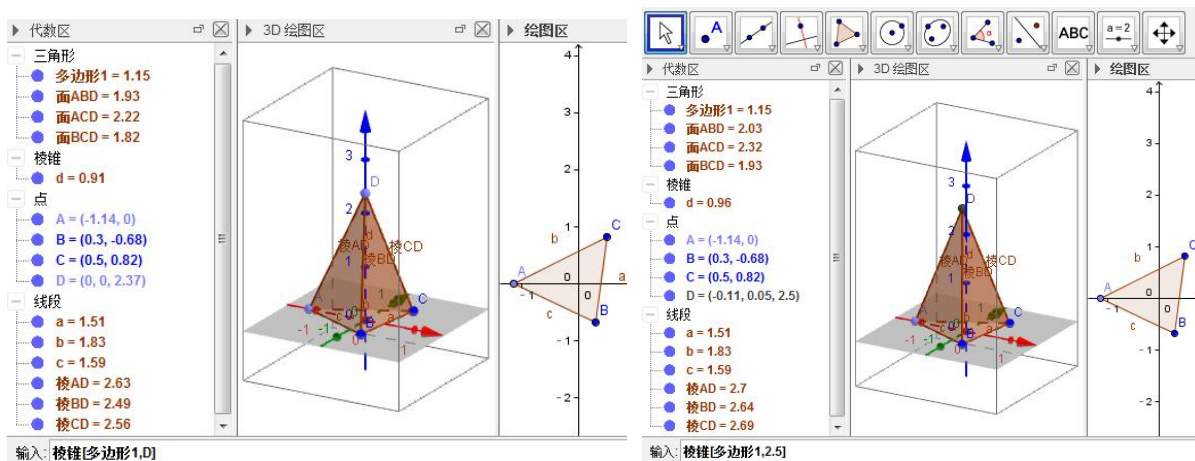
**Pyramid(<Point>, <Point>, ...); 棱锥(<点 1>, <点 2>, <点 3>, <点 4>, ...).**

返回由给定点定义的棱锥。

**案例：**“棱锥(A, B, C, D)”创建以三角形 ABC 为底，顶点为 D 的棱锥。  
**Pyramid(<Polygon>, <Point>); 棱锥(<多边形>, <顶点>).**  
 创建以多边形为底给定点为顶点的棱锥。



**案例：**“棱锥(多边形 1, A)”创建以“多边形 1”为底，顶点为 A 的棱锥。  
**Pyramid(<Polygon>, <Height>); 棱锥(<多边形>, <高度>).**  
 返回由多边形为底和给定高度定义的正棱锥。



**案例：**“棱锥(多边形 1, 2)”创建底为“多边形 1”高为 2 的正棱锥。  
**注：**参见 “棱锥”和 “拉成锥体”工具。

## 2.1.7 Plane. 平面

**Plane(<Polygon>); 平面(<多边形>).**

创建过多边形的平面。

**Plane(<Conic>); 平面(<圆锥曲线>).**

创建过圆锥曲线的平面。

**Plane(<Point>, <Plane>); 平面(<点>, <平行的平面>).**

创建过给定点且与已知平面平行的面。

**Plane(<Point>, <Line>); 平面(<点>, <经过的直线>).**

创建过给定点和直线的平面。

**Plane(<Line>, <Line>); 平面(<直线 1>, <直线 2>).**

创建过给定直线的平面(当两条直线在同一平面内时)。

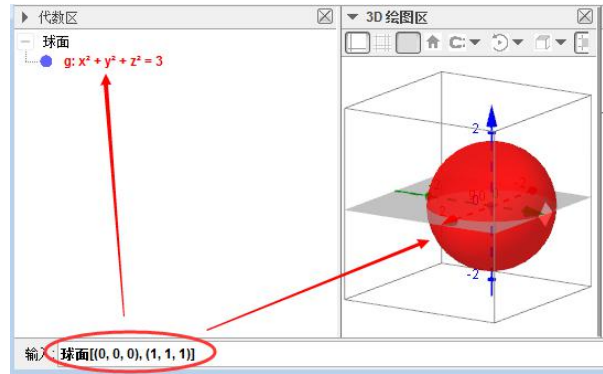
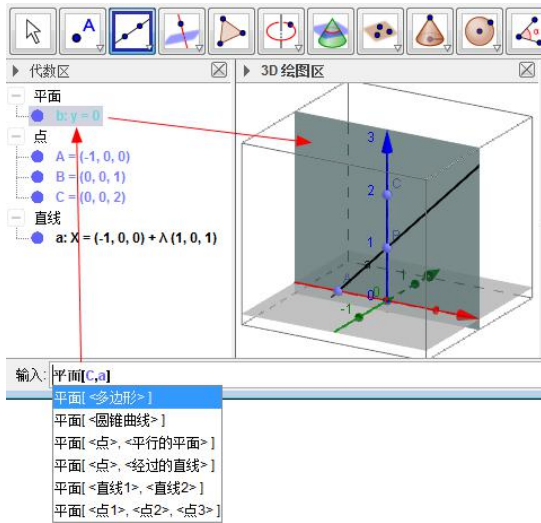
**Plane(<Point>, <Point>, <Point>); 平面(<点 1>, <点 2>, <点 3>).**

创建过三点的平面。

Plane(<Point>, <Vector1>, <Vector2>); 平面(<点>, <向量 1>, <向量 2>)。

编者注: 官网没有说明, 但功能已经实现。

注: 参见 “三点平面” 和 “平面” 工具。



### 2.1.8 Sphere. 球面

Sphere(<Point>, <Radius>); 球面(<球心>, <半径>)。

依据球心和半径创建球面。

Sphere(<Point>, <Point>); 球面(<球心>, <球面上一点>)。

创建第一个点为球心、过第二个点的球面。

案例: “球面((0, 0, 0), (1, 1, 1))” 得出 “ $x^2 + y^2 + z^2 = 3$ ”。在代数区出现球面的表达式, 在绘图区出现球面。

注: 参见 “球面(球心与一点)” 工具和 “球面(球心与半径)” 工具。

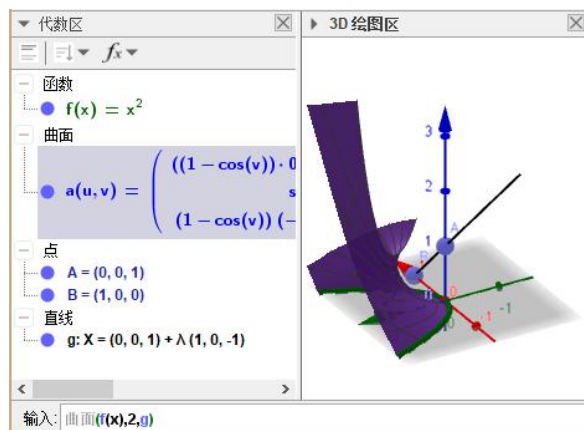
### 2.1.9 Surface. 曲面

Surface(<Function>, <Angle>); 曲面(<函数>, <角度|弧度>)。

创建以 x 轴为轴心旋转给定角度的旋转曲面。

Surface(<Curve>, <Angle>, <Line>); 曲面(<曲线>, <角度|弧度>, <轴线>)

创建以给定直线为轴把曲线旋转指定角度的旋转曲面。



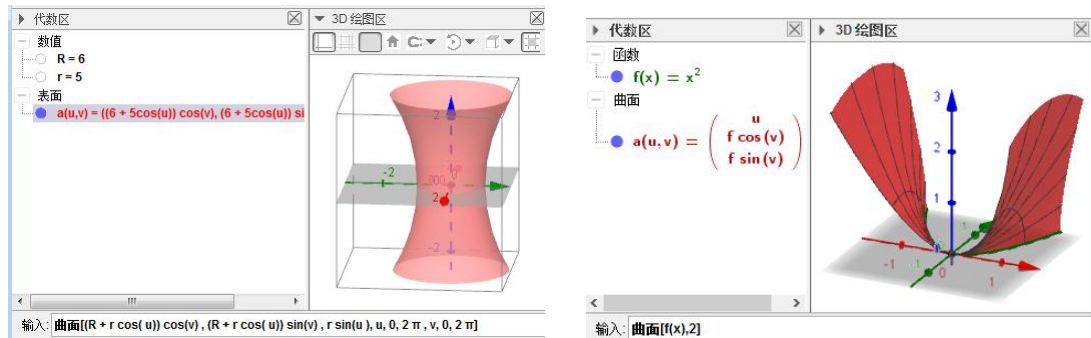


Surface(<Expression>, <Expression>, <Expression>, <Parameter Variable 1>, <Start Value>, <End Value>, <Parameter Variable 2>, <Start Value>, <End Value>); 曲面(<x 表达式>, <y 表达式>, <z 表达式>, <参数变量 1>, <初始值>, <终止值>, <参数变量 2>, <初始值>, <终止值>).

得出使用两个参数决定其区间的给定 x 表达式 (第一个表达式)、y 表达式 (第二个表达式) 和 z 表达式 (第三个表达式)。

**案例:** 设 r 和 R 是两个正实数, “曲面((R+cos(u))\*cos(v), (R+r\*cos(u)) sin(v), r\*sin(u), u, 0, 2π, v, 0, 2π)” 创建一个半径 r 和围绕 z 轴距离为 R 旋转的圆环。改变 R 和 r 的值, 圆环随变。

**注:** 终止值必须大于或者等于初始值且它们必须有限的。x、y 和 z 不允许作为参数变量。



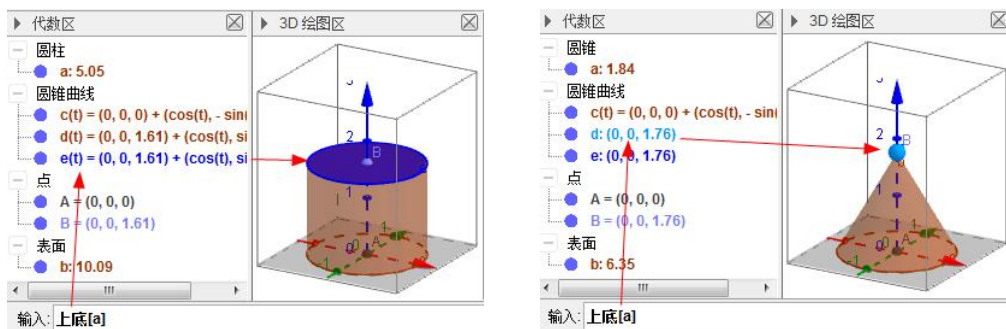
### 2.1.10 Top. 上底

Top(<Quadratic>); 上底(<二次曲面>).

创建有限二次曲面的上底。

**案例:** “上底(圆柱)” 得出一个圆。“上底(圆锥)” 得出锥顶(点)。

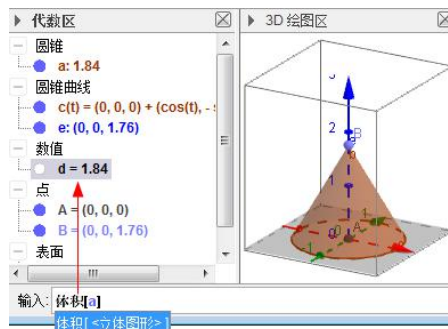
**注:** 参见“下底”指令、“底面”指令和“侧面”指令。




### 2.1.11 Volume. 体积

Volume(<Solid>); 体积(<立体图形>).

计算给定立体图形的体积。



**案例：**“体积(<棱锥>)”计算给定棱锥的体积；“体积(<棱柱>)”计算给定棱柱的体积；“体积(<圆柱>)”计算给定圆柱的体积；“体积(<圆锥>)”计算给定圆锥的体积。

**注：**参见  “体积”工具。

### 2.1.12 InfiniteCylinder. 无限长圆柱

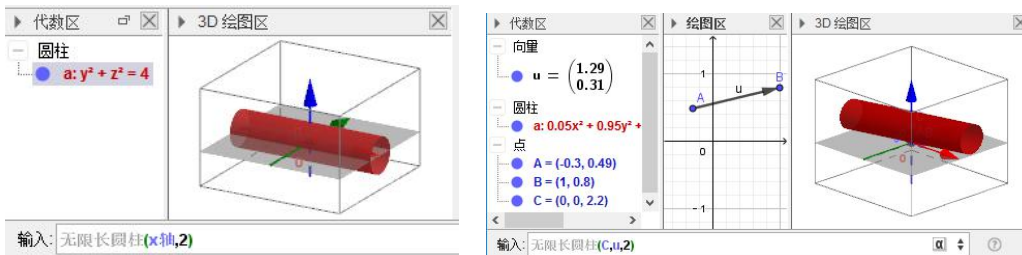
**InfiniteCylinder(<Line>, <Radius>);** 无限长圆柱(<轴线>, <半径>)。

使用给定半径和轴线创建无限圆柱。

**案例：**“无限长圆柱(x轴, 2)”创建以x轴为对称轴半径为2的圆柱。


**InfiniteCylinder(<Point>, <Vector>, <Radius>);** 无限长圆柱(<轴线上点>, <向量>, <半径>)。

创建给定半径和过给定点平行于已知向量的线为对称轴的无限圆柱。



**InfiniteCylinder(<Point>, <Point>, <Radius>);** 无限长圆柱(<轴线上点 1>, <轴线上点 2>, <半径>)。

创建给定半径和过给定点线为对称轴的无限圆柱。

**注：**参见“圆柱”指令和  “圆柱”工具。

### 2.1.13 InfiniteCone. 无限长圆锥

**InfiniteCone(<Point>, <Vector>, <Angle α>);** 无限长圆锥(<顶点>, <轴向量>, <半顶角角度|弧度>)。

构建以给定点为顶点，过顶点且平行于给定向量的直线为对称轴，顶角为  $2\alpha$  的无限圆锥。


**InfiniteCone(<Point>, <Point>, <Angle α>);** 无限长圆锥(<顶点>, <轴上点>, <半顶角角度|弧度>)。

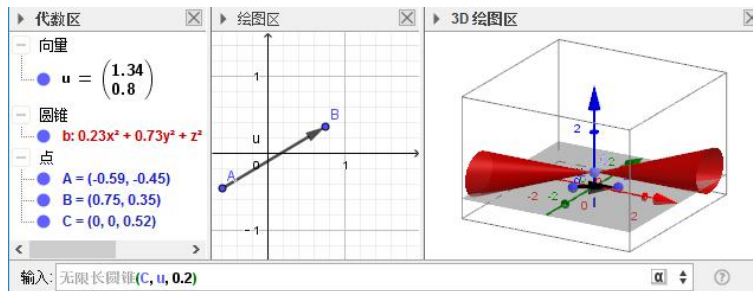
构建以给定点为顶点，过给定两点的直线为对称轴，顶角为  $2\alpha$  的无限圆锥。

**InfiniteCone(<Point>, <Line>, <Angle α>);** 无限长圆锥(<顶点>, <直线>, <半顶角角度>)。

构建以给定点为顶点，过顶点平行于给定直线的直线为对称轴，顶角为  $2\alpha$  的无限圆锥。

**注：**如果输入的角度没有单位符号，顶角默认为弧度制。

参见“圆锥”指令、“无限长圆柱”指令和  “圆锥”工具。



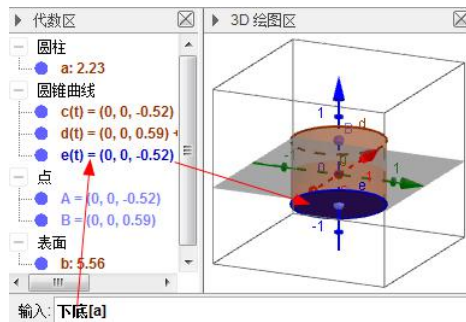
### 2.1.14 Bottom. 下底

Bottom(<Quadric>); 下底(<二次曲面>)。

构建有限二次曲面的下底。

案例：“下底(圆柱)”得出一个圆。

注：参见“上底”指令、“底面”指令和“侧面”指令。



### 2.1.15 IntersectConic. 相交曲线

IntersectConic(<Plane>, <Quadric>); 相交曲线(<平面>, <二次曲面>)。

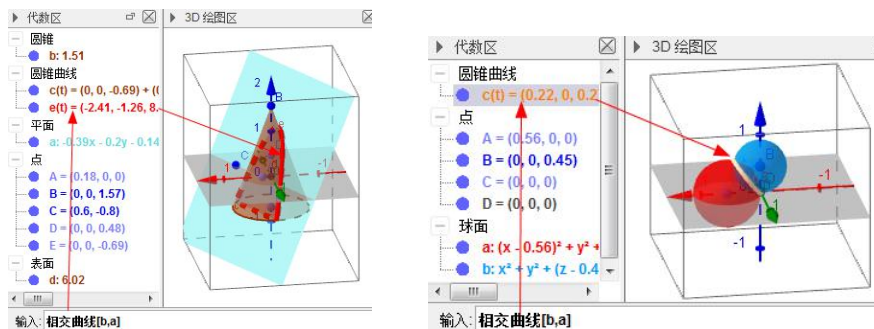
二次曲面与平面相交。

IntersectConic(<Quadric>, <Quadric>); 相交曲线(<二次曲面 1>, <二次曲面 2>)。

当相交曲线是圆锥曲线，返回一个曲线。

案例：“相交曲线(球面 1, 球面 2)”创建两个球面的相交曲线。

注：参见“交点”和“相交路径”指令。



### 2.1.16 Cylinder. 圆柱

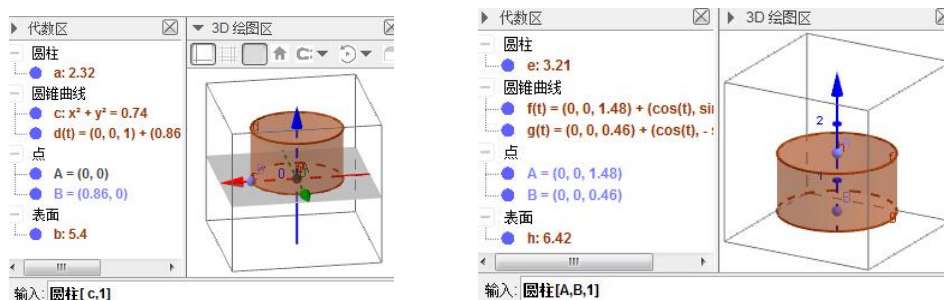
Cylinder(<Circle>, <Height>); 圆柱(<圆锥曲线底面>, <高度>)。

以给定的底和高构建圆柱。

Cylinder(<Point>, <Point>, <Radius>); 圆柱(<下底圆心>, <上底圆心>, <半径>)。

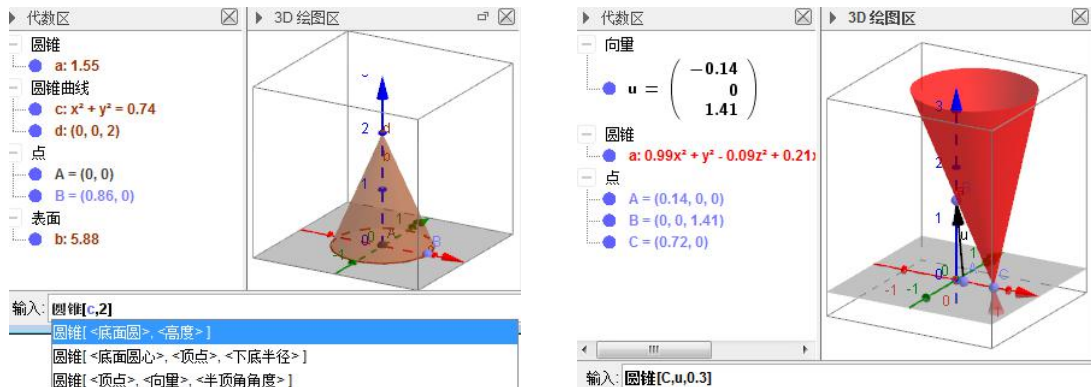
以给定的点为上下底圆心和给定半径构建圆柱。

注：参见“无限长圆柱”指令和“圆柱”及“拉成柱体”工具。



## 2.1.17 Cone. 圆锥

Cone(<Circle>, <Height>); 圆锥(<圆锥曲线底面>, <高度>)。  
用给定的底和高构建圆锥。



Cone(<Point>, <Point>, <Radius>); 圆锥(<底面圆心>, <顶点>, <底面半径>)。

以第一个点为圆心，第二点为顶点用给定的半径构建圆锥。

Cone(<Point>, <Vector>, <Angle  $\alpha$ >); 圆锥(<顶点>, <向量>, <半顶角角度|弧度>)。

以给定点为顶点，过顶点平行于向量的直线为对称轴，构建顶角为  $2\alpha$  的无限圆锥。

注：当角度大于  $\frac{\pi}{2}$  时，得出“未定义”

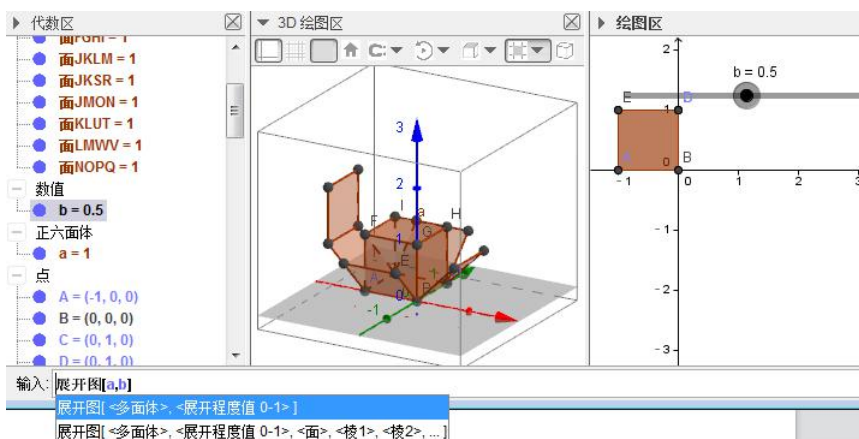
注：参见“无限长圆锥”指令， “圆锥”工具和 “拉成柱体”及 “拉成椎体”工具，这些操作能通过拖拽或选择圆且输入高度构建正圆锥。

## 2.1.18 Net. 展开图

Net(<Polyhedron>, <Number>); 展开图(<多面体>, <展开程度值 0-1>)。

构建凸多面体表面的平面展开图。数值须在 0 到 1 之间，用来确定展开的程度。当数值是 1 时，完全展开。

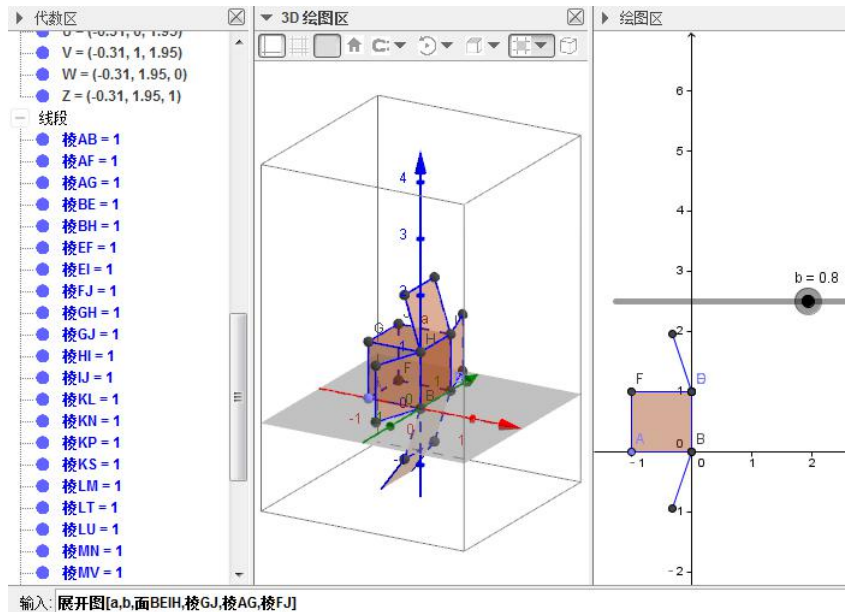
注：正六面体的展开图显示为拉丁十字。



上图中参数  $b$  在 0 到 1 之间，改变其大小能改变展开图的幅度。

Net(<Polyhedron>, <Number>, <Face>, <Edge>, <Edge>, ...); 展开图(<多面体>, <展开程度值 0-1>, <面>, <棱 1>, <棱 2>, ...)

仅适用于正六面体(目前)，允许通过指定基础面和作为切口棱创建正六面体的不同展开图。



将多面体 a 以面 BEIH 为基础面，按照指定的棱切开，各面围绕基础面展开。其展开程度由参数 b 控制。

编者注：官网描述滞后于软件功能，在较新的版本中，“展开图”指令已经适于正 n 面体了。

## 2.1.19 Octahedron. 正八面体

**Octahedron(<Point>,<Point>); 正八面体(<点 1>,<点 2>)**。拟修改为“正八面体[<等边三角形>]”  
创建由两个点确定第一个面的正八面体，第三个点自动建立到圆上，使得八面体可以围绕第一个棱旋转。

**Octahedron(<Point>,<Point>,<Point>); 正八面体(<点 1>,<点 2>,<点 3>)**。

由三个点确定正八面体的第一个面。这些点确定一个正三角形进而定义一个正八面体。

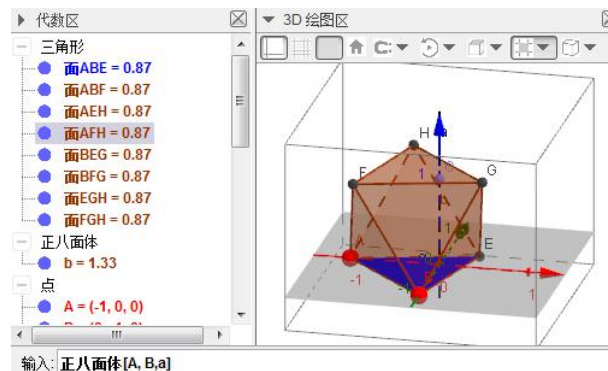
**Octahedron(<Point>,<Point>,<Direction>); 正八面体(<点 1>,<点 2>,<垂直于线段“点 1 点 2”的向量|线段|射线|直线; 或者平行于线段“点 1 点 2”的多边形|平面>)**。

创建以两个点间线段为棱的正八面体。

其他顶点由给定的向量确定，一般为：

垂直于点 1 和点 2 间线段的“向量/线段/直线/射线”，或者是平行于点 1 和点 2 间线段的“多边形/平面”。

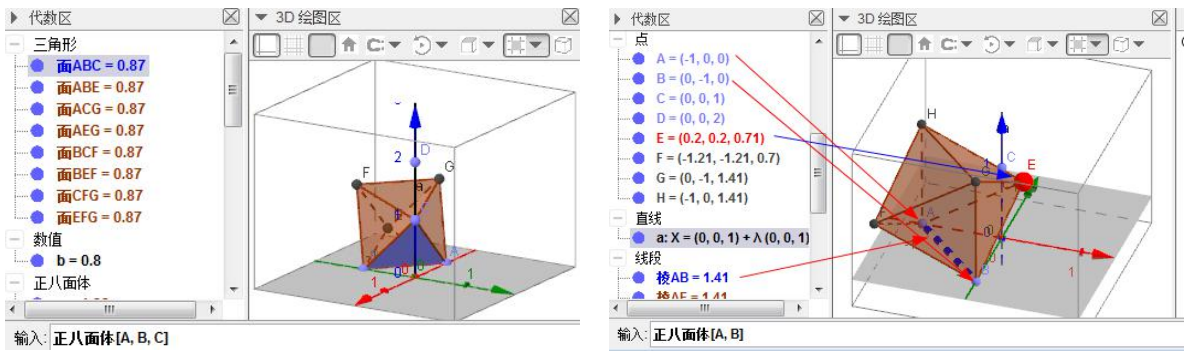
创建的正八面体为：一个表面内作为棱的线段均垂直于给定的向量/线段/直线/射线。或者，一个表面内作为棱的线段平行于多边形/平面。



直线 a 在 z 轴上，选定点 AB 构造的平面 ABE 与直线 a 垂直。

注：“正八面体(A,B)”是当“C=描点(圆周(中点(A,B),距离(A,B)  $\sqrt{3}/2$ , 线段(A,B)))”时，指

令“正八面体(A,B,C)”的快捷键。



注：参见“正六面体”、“正四面体”、“正十二面体”和“正二十面体”指令。

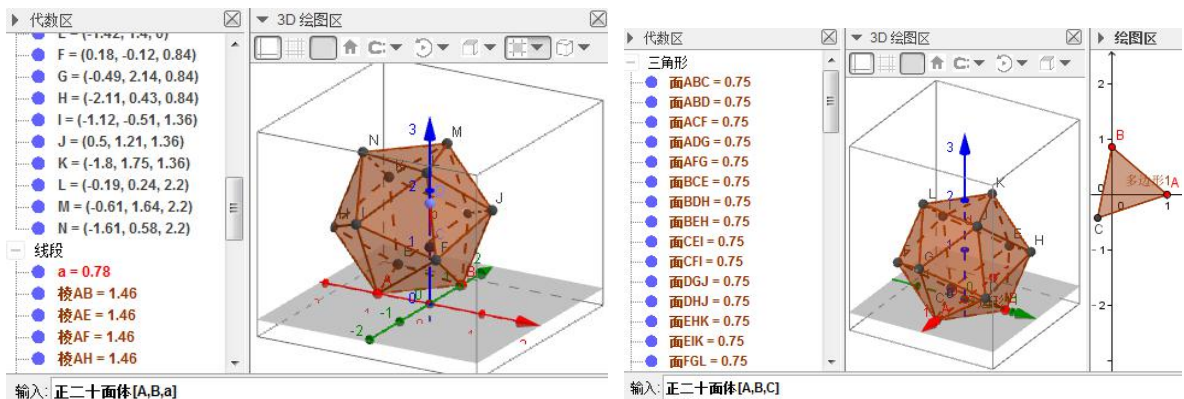
## 2.1.20 Icosahedron. 正二十面体

**Icosahedron(<Point>,<Point>); 正二十面体(<点 1>,<点 2>)**。拟修改为“正二十面体[<等边三角形>]”

创建由两个点确定第一个面的正二十面体，第三个点自动建立到圆上，使得二十面体可以围绕第一个棱旋转。

**Icosahedron(<Point>,<Point>,<Point>); 正二十面体(<点 1>,<点 2>,<点 3>)**。

创建由三点确定第一个面的正二十面体。这些点确定一个正三角形进而定义正二十面体。



选择正三角形三个点顺序以右手定则（右手虚握，四指指向三角形的构造方向，拇指指向面的正方向）决定面的正方向，同时决定正二十面体在面的正侧。

**Icosahedron(<Point>,<Point>,<Direction>); 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>)**。

创建以两个点间线段为棱的正二十面体。

其他顶点由给定的向量确定，一般为：

垂直于点 1 和点 2 间线段的“向量/线段/直线/射线”，或者是平行于点 1 和点 2 间线段的“多边形/平面”。

创建的正二十面体为：一个表面内作为棱的线段均垂直于给定的向量/线段/直线/射线。或者，一个表面内作为棱的线段平行于多边形/平面。

注：“正二十面体(A,B)”是当“C=描点(圆周(中点(A,B),距离(A,B)\*sqrt(3)/2,线段(A,B)))”时，“正二十面体(A,B,C)”的快捷键。

注：参见“正十二面体”、“正四面体”、“正八面体”和“正六面体”指令。

## 2.1.21 Cube. 正六面体

**Cube(<Point>,<Point>); 正六面体(<点 1>,<点 2>)。拟修改为“正六面体[<正方形>]”**

创建由两个点（相邻）确定第一个面的正六面体，第三个点自动建立到圆上，使得六面体可以围绕第一个棱旋转。

**Cube(<Point>,<Point>,<Point>); 正六面体(<点 1>,<点 2>,<点 3>)。**

创建由三点（相邻）确定第一个面的正六面体。这些点确定一个正方形进而定义正六面体。

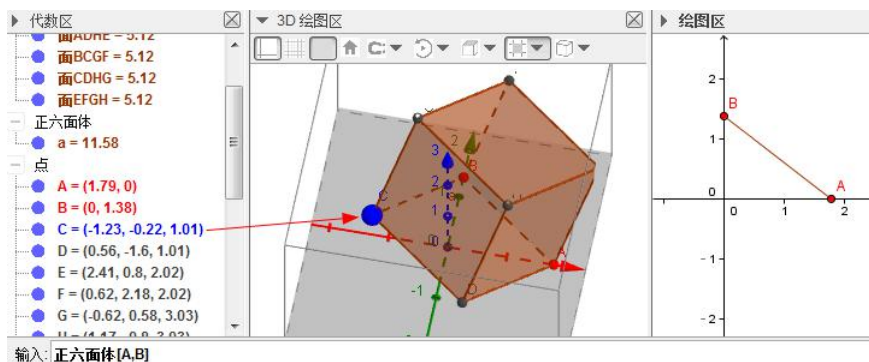
**Cube(<Point>,<Point>,<Direction>); 正六面体(<点 1>,<点 2>,<垂直于线段“点 1 点 2”的向量|线段|射线|直线; 或者平行于线段“点 1 点 2”的多边形|平面>。**

创建以两个点间线段为棱的正六面体。

其他顶点由给定的向量确定，一般为：

**垂直于点 1 和点 2 间线段的“向量/线段/直线/射线”，或者是平行于点 1 和点 2 间线段的“多边形/平面”。**

创建的正六面体为：一个表面内作为棱的线段垂直于给定的向量/线段/直线/射线。或者，一个表面内作为棱的线段平行于多边形/平面。



**注：**“正六面体(A,B)”是当“C=描点(圆周(B,距离(A,B),线段(A,B)))”时，“正六面体(A,B,C)”的快捷键。

**注：**参见“正十二面体”、“正四面体”、“正八面体”和“正二十面体”指令。

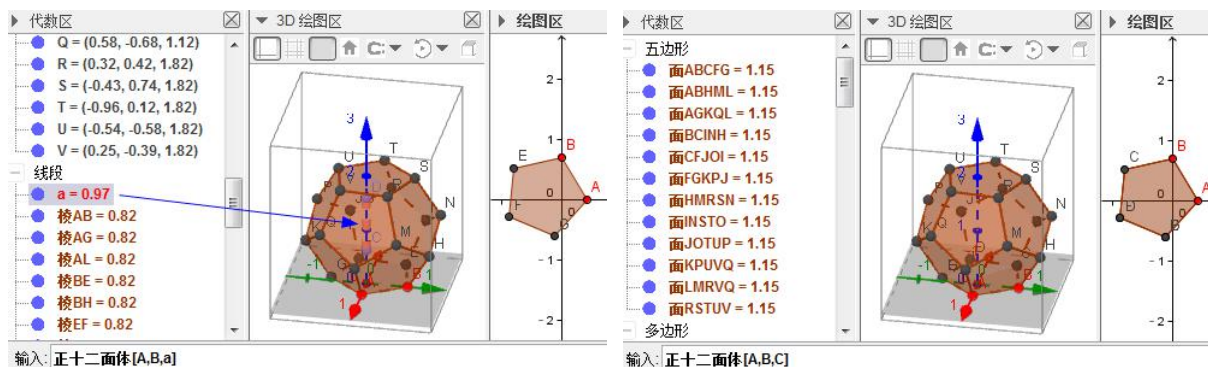
## 2.1.22 Dodecahedron. 正十二面体

**Dodecahedron(<Point>,<Point>); 正十二面体(<点 1>,<点 2>)。拟修改为“正十二面体[<正五边形>]”**

创建由两个点（相邻）确定第一个面的正十二面体，第三个点自动建立到圆上，使得十二面体可以围绕第一个棱旋转。

**Dodecahedron(<Point>,<Point>,<Point>); 正十二面体(<点 1>,<点 2>,<点 3>)。**

创建由三点（相邻）确定第一个面的正十二面体。这些点确定一个正五边形进而定义正十二面体。



**Dodecahedron(<Point>,<Point>,<Direction>); 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>。**

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>。

创建以两个点间线段为棱的正十二面体。

其他顶点由给定的向量确定，一般为：

垂直于点1和点2间线段的“向量/线段/直线/射线”，或者是平行于点1和点2间线段的“多边形/平面”。

创建的正二十面体为：一个表面内作为棱的线段均垂直于给定的向量/线段/直线/射线。或者，一个表面内作为棱的线段均平行于多边形/平面。

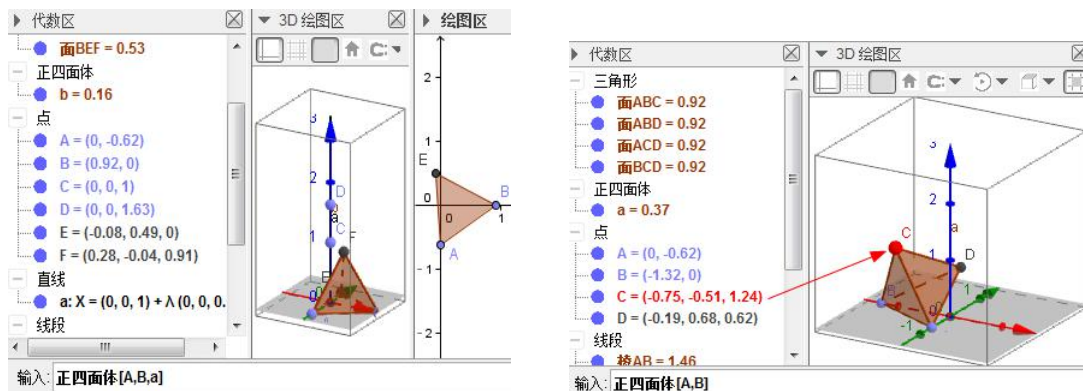
注：“正十二面体(A,B)”是当“C=描点(圆周(((1-sqrt(5))\*A+(3+sqrt(5))\*B)/4,距离(A,B)\*sqrt(10+2sqrt(5))/4,线段(A,B)))”时，指令“正十二面体(A,B,C)”的快捷键。

注：参见“正六面体”、“正四面体”、“正八面体”和“正二十面体”指令。

### 2.1.23 Tetrahedron. 正四面体

**Tetrahedron(<Point>,<Point>); 正四面体(<点1>,<点2>)。**拟修改为“正四面体[<等边三角形>]”

创建由两个点确定第一个面的正四面体，第三个点自动建立到圆上，使得四面体可以围绕第一个棱旋转。



**Tetrahedron(<Point>,<Point>,<Point>); 正四面体(<点1>,<点2>,<点3>)。**

创建由三点确定第一个面的正四面体。这些点确定一个正三角形进而定义正四面体。

**Tetrahedron (<Point>,<Point>,<Direction>); 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>)。**

创建以两个点间线段为棱的正四面体。

其他顶点由给定的向量确定，一般为：

垂直于点1和点2间线段的“向量/线段/直线/射线”，或者是平行于点1和点2间线段的“多边形/平面”。

创建的正四面体为：一个表面内作为棱的线段均垂直于给定的向量/线段/直线/射线。或者，一个表面内作为棱的线段平行于多边形/平面。

注：“正四面体(A,B)”是当“C=描点(圆周(中点(A,B),距离(A,B)\*sqrt(3)/2,线段(A,B)))”时，“正四面体(A,B,C)”的快捷键。

注：参见“正十二面体”、“正二十面体”、“正八面体”和“正六面体”指令。

### 2.1.24 PlaneBisector. 中垂面

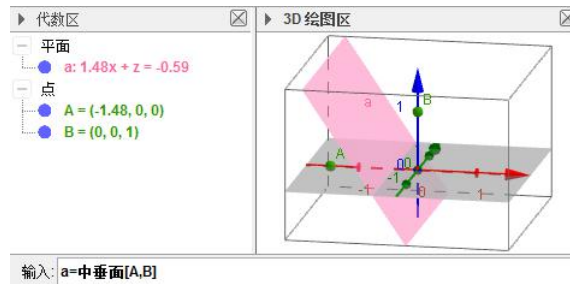
**PlaneBisector(<Point>,<Point>); 中垂面(<点1>,<点2>)。**

创建两点间的中垂面。

**PlaneBisector(<Segment>); 中垂面(<线段>)。**



创建线段的中垂面。



## 2.2 GeoGebraCommands. GeoGebra 指令

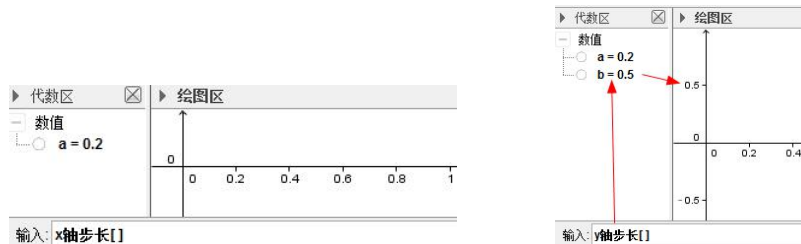
GeoGebra 指令是软件的一些特有指令，这些指令能够使 GeoGebra 软件有更好的操作或者展示，不属于某个专门的数学领域。

### 2.2.1 AxisStepX. X 轴步长

**AxisStepX(); X 轴步长()**。

返回当前 x 轴的步长幅度。不是单位刻度值，是刻度的最小精度值。

**注：**与“角点”和“序列”指令一起使用，“x 轴步长”和“y 轴步长”指令允许创建自定义坐标系（参见自定义坐标轴和网格）；参见“y 轴步长()”指令。



### 2.2.2 AxisStepY. Y 轴步长

**Y 轴步长()**

返回当前 y 轴的步长幅度，创建一个数值对象。

**注：**与“角点”和“序列”指令一起使用，“x 轴步长”和“y 轴步长”指令允许创建自定义坐标系；参见“x 轴步长()”指令。

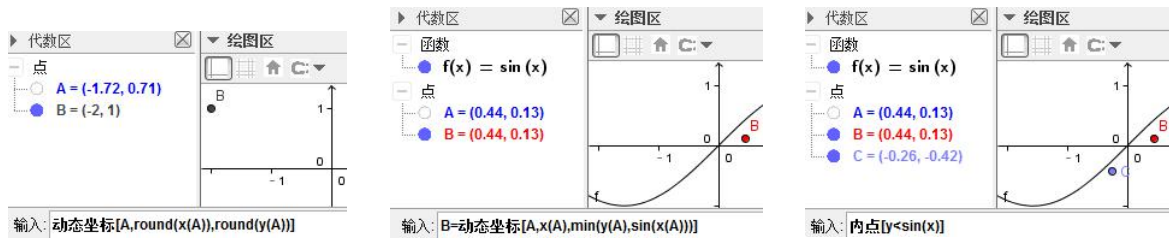
### 2.2.3 DynamicCoordinates. 动态坐标

**DynamicCoordinates(<Point A>, <Number X>, <Number Y>); 动态设置坐标(<点>, <横坐标>, <纵坐标>)**。

创建一个坐标为 (X, Y) 的点 B，这个点是从属点，但可以被移动。当试图移动这个新点 B 到设置坐标 (x, y)，实际上被移动的是点 A，并且根据 A(x, y) 重新计算点 B 的坐标。将点 A 隐藏并用鼠标拖动此点时的效果更好。新点 B 的横坐标 X 与纵坐标 Y 中至少有一个要取决于点 A。

**案例：**设 A 为一个自由点，而“B=动态设置坐标(A, round(x(A)), round(y(A)))”。当试图使用移动工具将点 B 移动到设置坐标 (1.3, 2.1) 时，事实上是点 A 的值变成 (1.3, 2.1) 而点 B 出现在 (1, 2)。

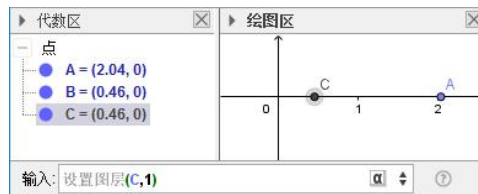
“B=动态设置坐标(A, x(A), min(y(A), sin(x(A))))” 创建在  $\sin(x)$  函数图象下方的点 B。



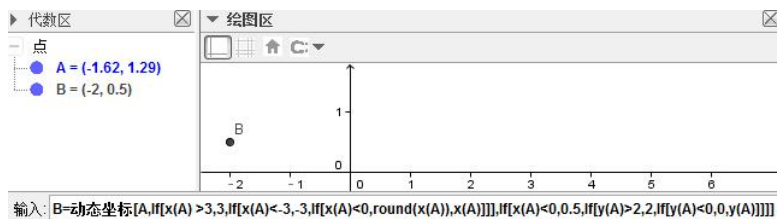
注：“内点( $y < \sin(x)$ )”更容易实现本例。如果先构造区域“ $y < \sin(x)$ ”，然后使用“对象上的点”工具在区域内创建一个新点，更有把握构造一个正弦曲线下方的一个点。

如下例子显示其它约束点 C 位置的方法：

案例 1：设“A=描点(xAxis)”且“B=描点(xAxis)”，在指令栏键入：“动态坐标(B, Min(x(B), x(A)), 0)”后按 Enter 键；“设置显示条件(B, 1, false)”后按 Enter 键；“设置图层(C, 1)”后按 Enter 键。现在，点 C 在第一图层中，不能移动到点 A 的右侧（点 A 向左移动不能越过点 C，点 C 可以在点 A 的左侧移动）。

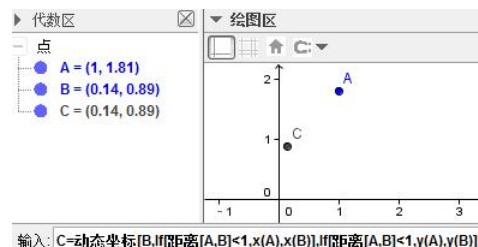


案例 2：定义点“A=(1, 2)”；在指令栏键入：“设置显示条件(A, 1, false)”且按 Enter 键；“B=动态设置坐标(A, 如果(x(A) > 3, 3, 如果(x(A) < -3, -3, 如果(x(A) < 0, round(x(A)), x(A))), 如果(x(A) < 0, 0.5, 如果(y(A) > 2, 2, 如果(y(A) < 0, 0, y(A))))”且按 Enter 键。



如此，点 B 的位置被限定为 x-负半轴的跳点 ( $y=0.5$ ) 和 x-正半轴的矩形自由区域（从点(0, 0)到点(3, 2)的矩形区域内）。

案例 3：这个案例当拖拽点 C 靠近点 A 时，点 A 变为粘滞（点 C 跳向 A）。定义点“A=(1, 2)”和“B=(2, 3)”。在指令栏键入：“设置显示条件(B, 1, false)”且按 Enter 键。“C=动态设置坐标(B, 如果(距离(A, B) < 1, x(A), x(B)), 如果(距离(A, B) < 1, y(A), y(B)))”。



DynamicCoordinates(<Point A>, <Number X>, <Number Y>, <Number Z>); 动态设置坐标(<点>, <横坐标>, <纵坐标>, <竖坐标 z> )。

创建一个三维坐标新点，这个点是从属点，但可以被移动。当试图移动这个新点到设置坐标(x, y, z)，实际上被移动的是给定点，并且根据给定点重新计算新点的坐标。将给定点隐藏并用鼠标拖动效果更好。新点的至少一个轴向坐标要取决于给定点。

## 2.2.4 Object. 对象

**Object(<Name of Object as Text>); 对象(<对象名称文本>)。**

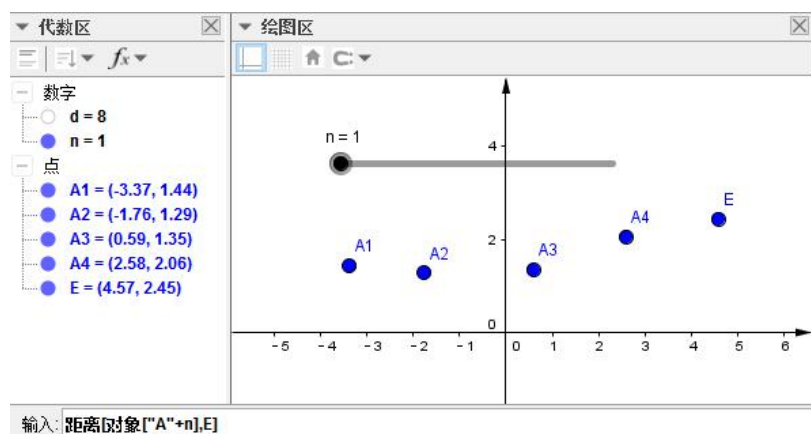
返回给定文本格式名称的对象。结果是从属对象。

**注：**“对象”指令与“名称”指令相反。

**案例：**假设一些点  $A_1, A_2, \dots, A_{20}$  以及一个参数  $n=2$ ，“对象( $"A"+n$ )”创建一个对象 B，其为点  $A_2$  的拷贝。改变  $n$  的大小在 1 到 20 之间的整数变化，创建的对象会自动改变为指定的  $A_n$  点。下图中的案例，当  $n$  变化时，距离会变化为点  $A_n$  和点 E 的距离。

**注：**必须确保在作图过程中，所引用对象作图过程步骤在本指令的前面。


**警告：**对象指令不能用在自定义工具中。

















## 2.2.5 ToolImage. 工具图标

**ToolImage(<Number>); 工具图标(<序号 0——73 及 5XX 和 20XX>)。**

创建一个图片对象，其为指定数值对应工具的图标，大小为 32x32 像素左下角在坐标系原点处。

**案例：**“工具图标(0)”，会得到图片 。

目前可以调用的工具图标参见下表，有的图标因为在不同的视区工作，有重号但图相同：

序号	英文	中文	图标
0	Move	移动	
1	Point	描点	
2	Line	直线	
3	Parallel Line	平行线	
4	Perpendicular Line	垂线	
5	Intersect	交点	
6	Delete	删除	
7	Vector	向量	
8	Perpendicular Bisector	中垂线	
9	Angle Bisector	角平分线	
10	Circle with Center through Point	圆(圆心与一点)	
11	Circle through 3 Points	圆(过三点)	
12	Conic through 5 Points	圆锥曲线	
13	Tangents	切线	

14	Relation	关系判断	
15	Segment	线段	
16	Polygon	多边形	
17	Insert Text	文本	
18	Ray	射线	
19	Midpoint or Center	中点/中心	
20	Circular Arc	圆弧	
21	Circular Sector	扇形	
22	Circumcircular Arc	圆弧(过三点)	
23	Circumcircular Sector	扇形(过三点)	
24	Semicircle through 2 Points	半圆	
25	Slider	滑动条	
26	Image	图像	
27	Show / Hide Object	显示/隐藏对象	
28	Show / Hide Label	显示/隐藏标签	
29	Reflect about Point	中心对称	
30	Reflect about Line	轴对称	
31	Translate by Vector	平移	
32	Rotate around Point	旋转	
33	Dilate from Point	位似	
34	Circle with Center and Radius	圆(圆心与半径)	
35	Copy Visual Style	复制样式	
36	Angle	角度	
37	Vector from Point	相等向量	
38	Distance or Length	距离/长度	
39	Move around Point	转动	
40	Move Graphics View	平移视图	
41	Zoom In	放大	
42	Zoom Out	缩小	
44	Polar or Diameter Line	极线/径线	
45	Segment with Given Length	定长线段	
46	Angle with Given Size	定值角度	
47	Locus	轨迹	
49	Area	面积	
50	Slope	斜率	
51	Regular Polygon	正多边形	
52	Check Box	复选框	
53	Compass	圆规(半径与圆心)	
54	Reflect Object about Circle	反演	

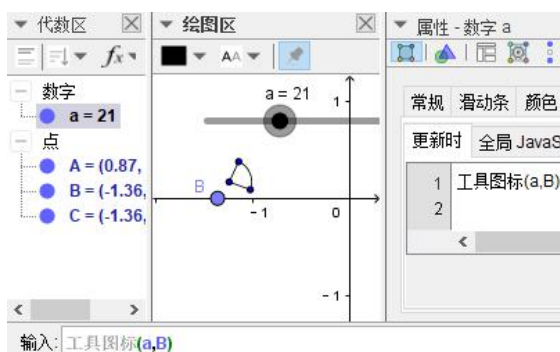
55	Ellipse	椭圆	
56	Hyperbola	双曲线	
57	Parabola	抛物线	
58	Best Fit Line	最佳拟合直线	
60	Button	按钮	
61	Input Box	输入框	
62	Pen	自由绘图	
64	Rigid Polygon	刚体多边形	
65	PolyLine	折线	
66	Probability Calculator	概率统计	
67	Attach/Detach Point	附着/脱离点	
68	Function Inspector	函数检视	
70	Vector Polygon	向量多边形	
71	Create List	列表	
72	Complex Number	复数	
73	Freehand Shape	智能绘图	
501	Point on Object	对象上的点	
1001	Evaluate	精确运算	
1002	Numeric	近似运算	
1003	Keep Input	检查	
1004	Expand	展开	
1005	Factor	分解	
1006	Substitute	替换	
1007	Solve	精确解	
1008	Derivative	导数	
1009	Integral	积分	
1010	Solve Numerically	近似解	
2001	Create List	列表	
2002	Create Matrix	矩阵	
2003	Create List of Points	点列	
2004	Create Table	表格	
2005	Create Polyline	折线	
2020	One Variable Analysis	单变量分析	
2021	Two Variable Regression Analysis	双变量回归分析	
2022	Multiple Variable Analysis	多变量分析	
2040	Sum	求和	
2041	Mean	平均数	
2042	Count	计数	
2043	Minimum	最小值	

2044	Maximum	最大值	123
------	---------	-----	-----

`ToolImage( <Number>, <Point> )`; 工具图标( <数字>, <点> )

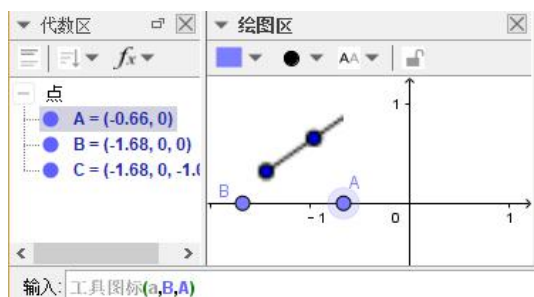
创建一个图片对象，其为指定数值对应工具的图标，大小为 32x32 像素，图片的左下角固定在指定点。

**案例：**建立一个参数 a 滑动条（整数 1 到 73），在其属性中的“更新时”的脚本追加“工具图标(a,B)”，改变 a 的大小，图片随变。



`ToolImage( <Number>, <Point>, <Point> )`; 工具图标( <数字>, <点 1>, <点 2> )

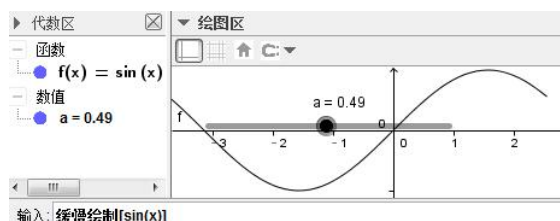
创建一个图片对象，其为指定数值对应工具的图标，大小为 32x32 像素，图片的左下角固定在第一个指定点，第二个点根据与第一个点位置，图片进行缩放。



## 2.2.6 SlowPlot. 缓慢绘制

`SlowPlot( <Function> )`; 缓慢绘制( <函数> )。

创建一个函数图象的动画：函数自左向右画出。动画由一个参数控制，这个参数随指令同时构造。



**案例：**“缓慢绘制(sin(x))”输出一个函数 sin(x) 的动画图象和一个控制参数 a（默认开启 a 动画）。动画过程由参数 a 控制，可以停止 a 的动画，手动拖动 a 观察缓慢绘制的过程。a 的值在 0 到 1 之间，控制图象在当前绘图区域的左边框起到右边框结束。

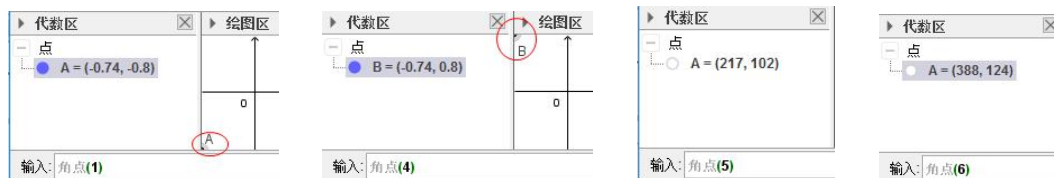
`SlowPlot( <Function>, true|false )`; 缓慢绘制( <函数>, <是否重复?true|false> )

**编者注：**官网没有帮助，且看不出与上一条指令有何不同。

## 2.2.7 Corner. 角点

**Corner**(**<Number of Corner>**); **角点**(**<角点数字 1\_左下角|2\_右下角|3\_右上角|4\_左上角|5\_绘图区分辨率|6\_Geogebra 视窗的宽度与高度>**)。

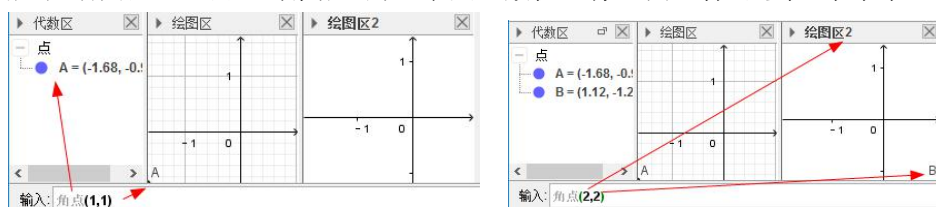
当数值  $n=1, 2, 3, 4$  时，在绘图区 1 的对应角落创建一个点（并不显示）；当  $n=5$  时，返回点  $(w, h)$ ， $w$  和  $h$  分别是绘图区 1 的像素宽度和高度；当  $n=6$  时，返回点为 GeoGebra 视窗的像素宽度和高度。即便绘图区 2 被激活，这个指令也只能用于绘图区 1。



**注：**当绘图区或者 GeoGebra 视窗的大小改变时，这些点的坐标值随动。角点(6)包含表格区，但是不包含工作区外框、菜单、工具栏、指令栏和指令帮助区。

**Corner**(**<Graphics View>**, **<Number of Corner>**); **角点**(**<绘图区>**, **<角点数字 1—4>**)。

创建固定在绘图区（1 或 2）角落上的一个点。数值  $n$  像上面一样可取值 1, 2, 3, 4。



**注：**在输入指令前选定哪个绘图区，构造的点就在哪个绘图区内。

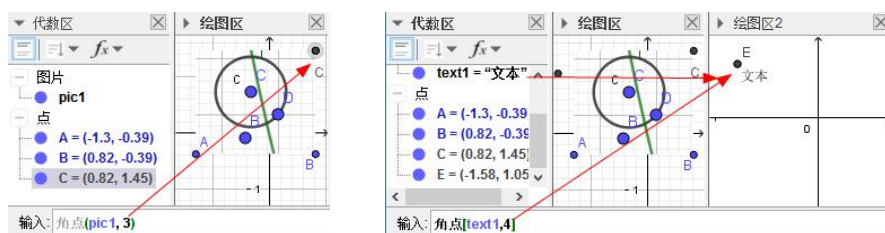
**Corner**(**<Image>**, **<Number of Corner>**); **角点**(**<图片>**, **<角点数字 1—4>**)。

创建固定在图片角落的一个点（数值  $n=1, 2, 3, 4$ ）。

**注：** $n=5$  不能用于图片对象上。有时，在不同的电脑上打开 GeoGebra 文件，可能会遇到找不到某些对象的麻烦（对象在可视区域之外）。这通常是由于别的电脑显示分辨率不同，可以用一些替代的处理方法避免对象从可视区域消失。

1. 将对象放在靠近左上角的位置。这是因为 GeoGebra 将这个角作为可视绘图区的位置标记而存入文件，因此对象应始终可见。

2. 利用“角点”指令计算对象的位置相对于绘图区角落的关系。



**Corner**(**<Text>**, **<Number of Corner>**); **角点**(**<文本>**, **<角点数字 1—4>**)。

创建固定在文本角落的一个点（数值  $n=1, 2, 3, 4$ ）。

**注：**“角点(**<文本>**, **<角点数字 1—4>**)”指令不能在“序列”或“映射”指令中作为参数使用。

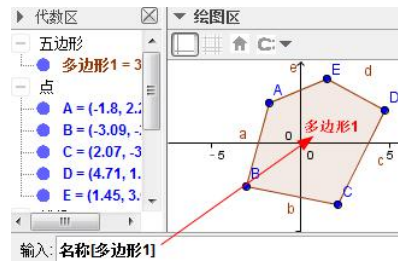
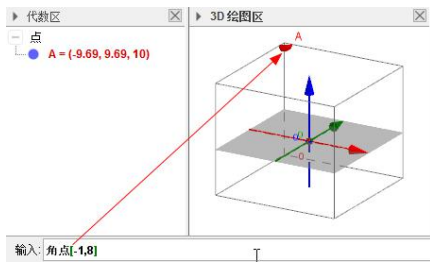
**注：**角落的编号按逆时针方向排序，起点是左下角。

**Corner**(**-1**, **<Number>**); **角点**(**-1**, **<数值 1—8>**)。

创建 3D 绘图区的一个角点(可用数值为 1 到 8，指向 8 个卦限远离坐标原点的顶点)。

**注：**3D 绘图区的角点使用 -1（数值从 1 到 8）；对于  $n=9$ ，返回点  $(w, h, 0)$ ，其中  $w$  和  $h$  是图形视图的宽度和高度，以像素为单位；对于  $n=10$ ，返回点  $(w, h, 0)$ ，其中  $w$  和  $h$  是主窗口的宽度和高度，以像素为单位；对于  $n=11$ ，返回视图方向（对于平行投影）或眼睛位置（例如透视投影）。

角点的编号按逆时针方向排序，起点是左下角。



### 2.2.8 Name. 名称

返回一个对象的名称，在绘图区生成一个文本对象。

**案例：**当绘图区中有“多边形 1”时，“名称(多边形 1)”返回一个文本，其内容就是多边形名称。

**注：**在动态文本内使用本命令时，对象可能会被重新命名。“名称”指令与“对象”指令相反。

### 2.2.9 SetConstructionStep. 设置作图步骤

**SetConstructionStep(<Number>); 设置作图步骤(<数字>)。**

为作图步骤赋予给定的值。可以使用这个指令建立按钮替代或者改善导航栏。

**案例：**假设某个构件由 10 步骤构成，“设置作图步骤(5)”使得绘图区域中，只有前 5 步骤构建的对象出现。

如：建立滑动条  $n$ ，整数，自 2 到 30 区间；建立按钮，在脚本中，输入“作图步骤( $n$ )”，下一行输入“ $n=n+1$ ”。继续构造构件。点击按钮就会逐步显示构件绘图区内的作图过程。

### 2.2.10 ToPoint. 转换为点

**ToPoint(<Complex Number>); 转换为点(<复数>)**

将给定的一个复数转换为点。

**案例：**“转换为点( $3+2i$ )”得出一个点“(3, 2)”。



**注：** $i$  由键盘“alt+i”输入。参见“转换为复数”、“转换为指数形式”和“转换为极坐标形式”指令。

运算区语法相同。但“转换为指数形式”只能用于运算区。

### 2.2.11 ToComplex. 转换为复数



### ToComplex(<Vector>); 转换为复数(<向量或点>)

把一个向量或点转换为代数形式复数，点（向量）改变跟随改变。

案例：“转换为复数((3,2))”得到复数“ $3+2i$ ”。

注：i 由键盘“alt+i”输入。参见“转换为点形式”、“转换为指数形式”和“转换为极坐标形式”指令。



运算区语法相同。只是结果不在代数区显示，且不在绘图区绘制复数点。

## 2.2.12 ToPolar. 转换为极坐标形式

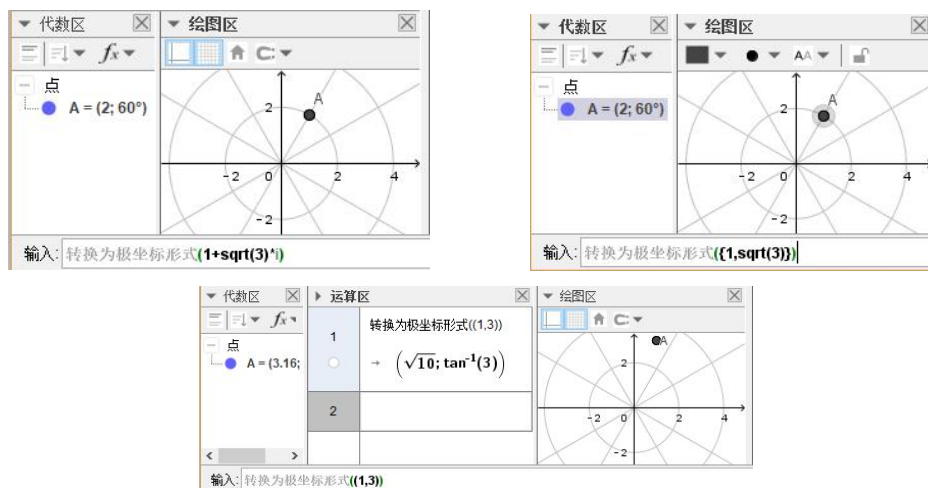
### ToPolar(<Complex Number>); 转换为极坐标形式(<复数>)

将给定的一个向量（点、复数）转换为极坐标值。

案例：“转换为极坐标形式( $1+\sqrt{3}i$ )”得到点( $2; 60^\circ$ )。

### ToPolar(<Vector>); 转换为极坐标形式(<向量>)

案例：“转换为极坐标形式( $\{1, \sqrt{3}\}$ )”得到点( $2; 60^\circ$ )；“转换为极坐标形式( $(1, 3)$ )”得到点( $\sqrt{10}; \text{atan}(3)$ )。



注：i 由键盘“alt+i”输入。参见“转换为点”、“转换为指数形式”和“转换为复数”指令。运算区语法相同。

## 2.2.13 ConstructionStep. 作图步序

### ConstructionStep(); 作图步序()

构件需要多个步骤完成，返回当前已完成的操作步骤数。

### ConstructionStep(<Object>); 作图步序(<对象>)

构件由许多对象构成，返回指定对象在构件中的步骤序数。

注：如果在对象文本 1 的显示条件栏中输入“作图步序()  $\neq$  作图步序(文本 1)”，只有当导航

栏进行播放并运行到一个特定的步骤时，这个文本对象才会出现。如此，一些注释文字可以一个接一个地追加在程序架构中特定的步骤位置。如果某个对象在文本以后建立，就可以让文本与这个对象作图过程同时出现。

## 2.3 Spreadsheet. 表格区

### 2.3.1 Cell. 单元格

**Cell(<Column>, <Row>); 单元格(<列序>, <行序>)。**

返回表格区中指定列和行的单元格内容（文本格式）的复制品。

**案例：**“单元格(2, 1)”返回“B1”单元格的复制品。更改“B1”内容，其跟随着更新改变。

**注：**默认情况下表格区的单元格是辅助对象，故这个命令返回的对象也是辅助对象。

需要确保调用的单元格在作图过程中早于这个命令。



### 2.3.2 CellRange. 单元格区域数字列表

**CellRange(<Start Cell>, <End Cell>); 单元格区域数字列表(<起始单元格>, <终止单元格>)。**

创建包括单元格区域内单元格数值的列表。

**案例：**设 A1=1、A2=4、A3=9 都是表格区单元格数值。然后在指令栏输入“单元格区域数字列表(A1, A3)”返回“列表{1, 4, 9}”。

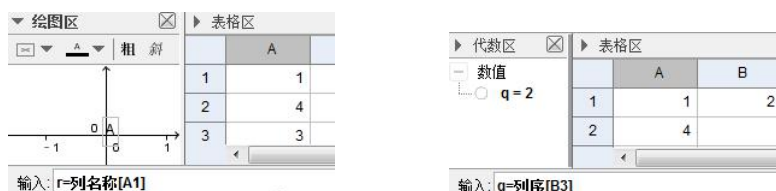
**注：**矩形区域可以是多行和多列，列表中元素顺序都是先列后行。可以简洁输入“A1:B3”返回列表。

### 2.3.3 ColumnName. 列名称

**ColumnName(<Spreadsheet Cell>); 列名称(<表格区单元格>)。**

返回单元格一个文本格式的列名称。

**案例：**“r=列名称(A1)”创建“文本 r=A”且在绘图区显示文本“A”（初始位置(0, 0)）。



### 2.3.4 Column. 列序

**Column(<Spreadsheet Cell>); 列序(<表格区单元格>)。**

返回单元格数值型列序号(从 1 起)。

**案例：**“q=列序(B3)”返回“q=2”，因为 B 列是表格区第二列。

### 2.3.5 FillCells. 填充单元格

**FillCells(<CellRange>, <Object>); 填充单元格(<单元格区域>, <对象>)**。

从选择的单元格区域的第一个单元格开始，向右向下复制填充值或计算式，目标单元格必须是自由对象，将从属于复制的对象。

**注：**单元格区域可以输入诸如 B2:D5 格式；对象可以是任意诸如“3”、“区间随机数(0, 10)”、“圆周(A, B)”等对象；单元格已经使用列和行如“B2”的标签。

**案例：**“填充单元格(B2:D5, 区间随机数(0, 10))”，在工作表“B2:D5”区间产生 0—10 之间的随机数。



**FillCells(<Cell>, <List>); 填充单元格(<单元格>, <列表>)**。

在给定单元格中从第一个单元格向右复制列表中的值。目标单元格必须是自由对象，将从属于复制的对象。

**案例：**“填充单元格(B1, {3, 7, 2, 1})”，得出自 B1 单元格开始向右填充的数字。



**FillCells(<Cell>, <Matrix>); 填充单元格(<单元格>, <矩阵>)**。

从矩阵中复制数据。矩阵的左上角匹配给定单元格。目标单元格必须是自由对象，将从属于矩阵。

**注：**参见“填充行”和“填充列”指令。

### 2.3.6 FillColumn. 填充列

**FillColumn(<Column>, <List>); 填充列(<列序>, <列表>)**。

从列序指定的列（1 是 A，2 是 B，等等）第一个单元格开始复制列表中的值。目标单元格必须是自由对象，将从属于列表。

**注：**参见“填充行”和“填充单元格”指令。

### 2.3.7 FillRow. 填充行

**FillRow(<Row>, <List>); 填充行(<行序>, <列表>)**。

从行序指定的行第一个单元格开始复制列表中的值。目标单元格必须是自由对象，将从属于列表。

**注：**参见“填充列”和“填充单元格”指令。

### 2.3.8 Row. 行序

Row(<Spreadsheet Cell>); 行序(<表格区单元格>)。

返回表格区单元格的行数值(自1起)。

案例：“r=行序(B3)”得出“r=3”。

## 2.4 Financial. 财务

“财务”指令只有几个指令，都是关于一个定期存款的每期付款额、现值、未来值、利率和期数的计算。

### 2.4.1 Rate. 利率

Rate(<Number of Periods>, <Payment>, <Present Value>, <Future Value (optional)>, <Type (optional)>, <Guess (optional)>); 利率(<期数>, <每期付款额>, <现值>, <未来值(可选)>, <类型(可选)1\_期初|0\_期末>, <预期利率(可选)\_范围为0——1间的数>)。

返回年金的每期利率。

<期数>贷款的总期数。

<每期付款额>每期付款总额。

<现值>几期付款总额的现值。如果不输入值，假定为0。

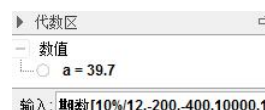
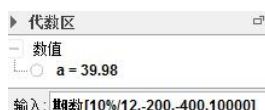
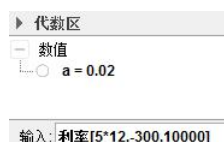
<未来值(可选)>最后一期付款后想达到的款余额。如果不输入未来值，假定为0。

<类型(可选)>投资到期的显示。如果没有输入值或者输入0投资在周期末结束，如果输入1其在周期的开始时结束。

<预期利率(可选)>预期的一个利率。

案例：“利率(5\*12, -300, 10000)”得出每月利率0.02(2%)。

注：如果月付一个5年贷款，期数使用5\*12；对所有参数，现金支付使用负数而现金收入使用正数；参见“每期付款额”、“期数”、“现值”和“未来值”指令。



### 2.4.2 Payment. 每期付款额

Payment(<Rate>, <Number of Periods>, <Present Value>, <Future Value (optional)>, <Type (optional)>); 每期付款额(<利率>, <期数>, <现值>, <未来值(可选)>, <类型(可选)1\_期初|0\_期末>)。

返回固定每期付款额和固定利率的投资期数。

<利率>每期利率。

<期数>贷款的总期数。

<现值>几期付款总额的现值。如果不输入值，假定为0。

<未来值(可选)>最后一期付款后想达到的款余额。如果不输入未来值，假定为0。

<类型(可选)>投资到期的显示。如果没有输入值或者输入0投资在周期末结束，如果输入1其在周期的开始时结束。

**案例：**“每期付款额(6%/12, 10, 10000, 0, 1)”得出一个贷款月付 1022.59。

**注：**确保使用的<利率>和<期数>单位保持一致。如果逐月投资一个年度利率为 6%的 4 年期贷款，利率使用 6%/12 而期数使用 4\*12；对所有参数，现金支付使用负数而现金收入使用正数；参见“利率”、“期数”、“现值”和“未来值”指令。



### 2.4.3 Periods. 期数

**Periods(<Rate>, <Payment>, <Present Value>, <Future Value (optional)>, <Type (optional)>); 期数(<利率>, <每期付款额>, <现值>, <未来值(可选)>, <类型(可选)>1\_期初|0\_期末)**

返回基于固定每期付款额和固定利率的定期投资期数。

<利率>每期利率。

<期数>贷款的总期数。

<现值>几期付款总额的现值。如果不输入值，假定为 0。

<未来值(可选)>最后一期付款后想达到的款余额。如果不输入未来值，假定为 0。

<类型(可选)>投资到期的显示。如果没有输入值或者输入 0 投资在周期末结束，如果输入 1 其在周期的开始时结束。

**案例：**

“期数(10%/12, -200, -400, 10000)”得出贷款的总期数为 39.98。

“期数(10%/12, -200, -400, 10000, 1)”得出贷款的总期数为 39.7。

**注：**如果逐月每期付款投资的年利率是 10%，利率使用 10%/12；对所有参数，现金支付使用负数而现金收入使用正数；参见“每期付款额”、“利率”、“现值”和“未来值”指令。

### 2.4.4 FutureValue. 未来值

**FutureValue(<Rate>, <Number of Periods>, <Payment>, <Present Value(optional)>, <Type (optional)>); 未来值(<利率>, <期数>, <每期付款额>, <现值(可选)>, <类型(可选)>1\_期初|0\_期末)**

返回基于恒定的每期付款额和恒定利率的定期投资未来值。

<利率>每期利率。

<期数>一个年金付款的周期总数。

<每期付款额>每期付款总额。

<现值(可选)>几期付款总额的现值。如果不输入值，假定为 0。

<类型(可选)>投资到期的显示。如果没有输入值或者输入 0 投资在周期末结束，如果输入 1 其在周期的开始时结束。

**案例：**“未来值(10%/12, 15, -200, 0, 1)”得出未来值为 3207.99。

**注：**确保使用的<利率>和<期数>单位保持一致。如果逐月投资一个年度利率为 10%的 4 年期贷款，利率使用 10%/12 而期数使用 4\*12；对所有参数，现金支付使用负数而现金收入使用正数；参见“每期付款额”、“利率”、“现值”和“期数”指令。

### 2.4.5 PresentValue. 现值

**PresentValue(<Rate>, <Number of Periods>, <Payment>, <Future Value (optional)>, <Type**

(optional)>); 现值(<利率>, <期数>, <每期付款额>, <未来值(可选)>, <类型(可选) 1\_期初|0\_期末> )。

返回一个投资的付款总计。

<利率>每期利率。

<期数>贷款的总期数。

<每期付款额>每期付款总额。

<未来值(可选)>最后一期付款后想达到的款余额。如果不输入未来值，假定为 0。

<类型(可选)>投资到期的显示。如果没有输入值或者输入 0 投资在周期末结束，如果输入 1 其在周期的开始时结束。

**案例：**

“现值(12%/12, 4\*12, -100, 5000, 0)”得出现值为 696.06。

“现值(12%/12, 4\*12, -100, 5000, 1)”得出现值为 734.07。

**注：**确保使用的<利率>和<期数>单位保持一致。如果逐月投资一个年度利率为 10%的 4 年期贷款，利率使用 10%/12 而期数使用 4\*12；对所有参数，现金支付使用负数而现金收入使用正数；参见每期付款额、期数、利率和未来值指令。

**运算区语法：**财务指令均可以用于运算区。

## 2.5 Algebra. 代数

软件中代数指令帮助中的第一个指令是“ShowSteps”，是待开发指令，没有实际意义。

### 2.5.1 Cross. 叉积

**Cross(<Vector u>, <Vector v>); 叉积(<向量 1>, <向量 2>)**

计算向量 u 和 v 的向量积。

**案例：**“叉积({1, 3, 2}, {0, 3, -2})”得出“{-12, 2, 3}”。



如果向量包含有未定义参数，在向量积中会给出格式代码。仅适于运算区。

**案例：**“叉积({a, b, c}, {d, e, f})”得出“{b f-c e, -a f+c d, a e-b d}”。

**注：**在指令栏可以使用“ $u \otimes v$ ”；参见“点积”指令。

**释义：**叉积又名叉乘、向量的外积、向量积。源自于三维向量空间的运算，两个三维向量的叉积等于一个新的向量，该向量与前两者垂直，且长度为前两者张成的平行四边形面积，其方向按照右手螺旋决定。在三维向量空间中，假设 a 和 b 是两个向量，那么它们的叉积“ $c = a \times b$ ”可如下严格定义。(1)  $|c| = |a \times b| = |a| |b| \sin \langle a, b \rangle$ ; (2)  $c \perp a$ , 且  $c \perp b$ ; (3) c 的方向要用“右手法则”判断（用右手的四指先表示向量 a 的方向，然后手指朝着手心的方向摆动到向量 b 的方向，大拇指所指的方向就是向量 c 的方向）。

### 2.5.2 Division. 除法

**Division(<Dividend Number>, <Divisor Number>); 除法(<被除数\_整数>, <除数\_整数>)**

给出两个数值的商（得数结果的整数部分）以及两个数值相除的余数。

案例：“除法(16,3)”输出“列表1={5,1}”。

Division(<Dividend Polynomial>, <Divisor Polynomial>); 除法(<被除式\_整式>, <除式\_整式>)

给出两个多项式相除的商和余式。

案例：“除法( $x^2+3x+1, x-1$ )”输出“列表1= $\{x+4, 5\}$ ”，并绘制  $f(x)=5$  和  $f(x)=x+4$  的图象。



CAS Syntax (运算区语法): 本指令适于运算区。



### 2.5.3 Dot. 点积

Dot(<Vector>, <Vector>); 点积(<向量 1>, <向量 2>)

返回两个向量的点积（内积）。

案例：“点积( $\{1, 3, 2\}, \{0, 3, -2\}$ )”得到 5。



在运算区中，如果向量包含有未定义参数，在向量积中会给出格式代码。

案例：“点积( $\{a, b, c\}, \{d, e, f\}$ )”得出“ $\{ad+be+cf\}$ ”。

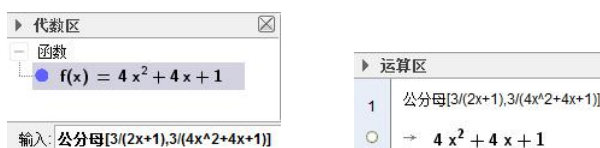
释义：点积又称数量积或内积。向量  $u$ 、 $v$  的点积是一个标量，用  $u \cdot v$  表示。在三维空间中它被定义为： $u_x v_x + u_y v_y + u_z v_z$ 。点积的值由以下三个值确定： $u$  的大小、 $v$  的大小、 $u$  和  $v$  夹角的余弦。在  $u$  和  $v$  非零的前提下，点积如果为负，则  $u$  和  $v$  形成的角大于 90 度；如果为零，那么  $u$  和  $v$  垂直；如果为正，那么  $u$  和  $v$  形成的角为锐角。

### 2.5.4 CommonDenominator. 公分母

CommonDenominator(<Expression>, <Expression>); 公分母(<分式 1>, <分式 2>)

返回两个表达式最小公分母的方程格式函数。

案例：“公分母( $3/(2x+1), 3/(4x^2+4x+1)$ )”输出“ $f(x)=4x^2+4x+1$ ”并绘制出函数图象。



CAS Syntax (运算区语法)

CommonDenominator(<Expression>, <Expression>); 公分母(<分式 1>, <分式 2>)

返回两个表达式最小公分母。

案例：“公分母( $3/(2x+1)$ ,  $3/(4x^2+4x+1)$ )”输出“ $4x^2+4x+1$ ”。

### 2.5.5 NextPrime. 后一质数

**NextPrime(<Number>); 后一质数(<数值>)。**

返回比输入数值更大的最小质数。

案例：“后一质数(10000)”输出“a=10007”。



注：运算区通用；参见“前一质数”指令。

### 2.5.6 Simplify. 化简

**Simplify(<Function>); 化简(<函数>)。**

尽可能简化所给函数的项（合并同类项）。

案例：“化简( $x+x+x$ )”输出函数“ $f(x)=3x$ ”。

案例：对于  $a=b=c=-1$  “化简( $f(x)="+a+"x^2"+"b+"x"+"c$ )”在绘图区输出文本“ $f(x)=-x^2-x-1$ ”。

注：公式文本指令通常会得出更好的效果而且更简便。

这个指令会加载代数计算系统，某些电脑会变慢。尝试使用多项式函数指令替代。



**CAS Syntax (运算区语法)**

**Simplify(<Text>); 化简("<文本>")。**

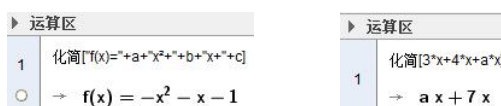
通过清除重复的符号而整理表达式文本等等。

案例：对于  $a=b=c=-1$  “化简( $f(x)="+a+"x^2"+"b+"x"+"c$ )”输出文本“ $f(x)=-x^2-x-1$ ”。

**Simplify(<Function>); 化简(<函数>)**

尽可能简化所给函数的项。未定义的变量也可以包括在内。

案例：“化简( $3*x+4*x+a*x$ )”输出“ $ax+7x$ ”。



注：参见“因式分解”指令。

### 2.5.7 Solutions. 解集

**Solutions(<Equation>); 解集(<方程>)。**

解给定主变量方程且返回解列表。

案例：“解集( $x^2=4x$ )”得出“ $x^2=4x$ ”的解列表“ $\{4, 0\}$ ”。





**Solutions**(**<Equation>**, **<Variable>**); **解集**(**<方程>**, **<变量>**)。

解给定未知变量方程且返回解列表。

**案例**：“解集( $x*a^2=4a, a$ )”得出“ $xa^2=4a$ ”的解列表。

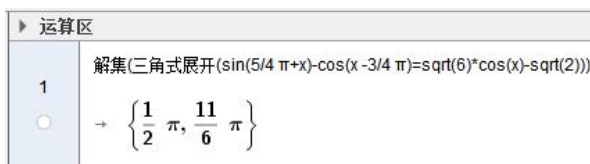
**Solutions**(**<List of Equations>**, **<List of Variables>**); **解集**(**<方程列表>**, **<变量列表>**)。

解给定未知变量方程组且返回所有解列表。

**案例**：“解集( $\{x=4x+y, y+x=2\}, \{x, y\}$ )”得出方程组“ $x=4x+y$ ”和“ $y+x=2$ ”的解集“ $(-1 \ 3)$ ”。“解集( $\{2a^2+5a+3=b, a+b=3\}, \{a, b\}$ )”得出“ $\{\{0, 3\}, \{-3, 6\}\}$ ”。



**注**：有时需要做一些操作以允许计算器工作，如“解集(三角式展开( $\sin(5/4 \pi+x)-\cos(x-3/4 \pi)=\sqrt{6}*\cos(x)-\sqrt{2}$ )))”



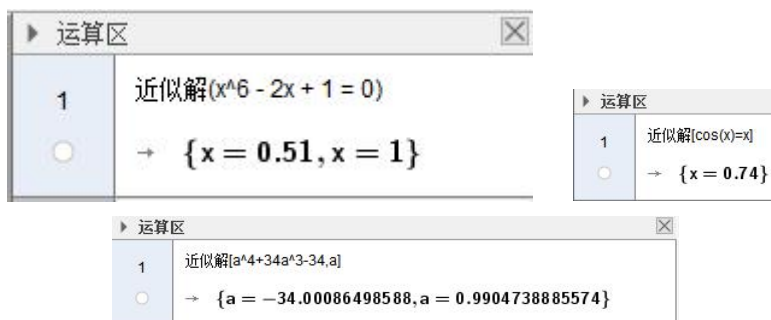
参见“精确解”指令。

## 2.5.8 NSolve. 近似解

**NSolve**(**<Equation>**); **近似解**(**<方程>**)。

找出给定变量  $x$  的方程的近似解。

**案例**：“近似解( $\cos(x)=x$ )”得出“ $\{x=0.7390851332151606\}$ ”



这个指令以下用法仅适用于运算区。

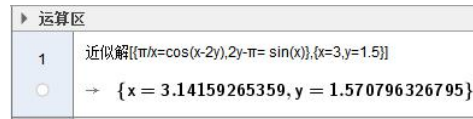
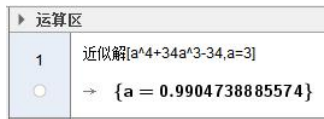
**NSolve**(**<Equation>**, **<Variable>**); **近似解**(**<方程>**, **<变量>**)。

找出给定未知变量的方程的近似解。

**案例**：“近似解( $a^4+34a^3-34, a$ )”得出列表结果为“ $\{a=-34.000864985883736479458, a=0.99047388855741781044881\}$ ”。

**NSolve**(**<Equation>**, **<Variable=starting value>**); **近似解**(**<方程>**, **<变量=初值>**)

找到给定未知变量方程未知变量初值的近似解。



**NSolve**(**<List of Equations>**,**<List of Variables>**); **近似解**(**<方程(组)列表>**,**<变量列表>**)。找出给定未知变量的方程组的近似解

**案例**：“近似解( $\{\pi/x=\cos(x-2y), 2y-\pi=\sin(x)\}, \{x=3, y=1.5\}$ )”得出“ $\{x=3.14159265359, y=1.570796326795\}$ ”。

**注**：可选起始值点如  $\{x=3, y=1.5\}$ ；可以键入“Alt+p”输入“ $\pi$ ”；

**注**：如果没有给出  $a=3$  或  $\{x=3, y=1.5\}$  这样的起点，那么数值算法可能会发现很难找到近似解（给出一个起点也不能保证有近似解）。结果小数位数取决于选项中设置的精确度。近似解不适用于渐近  $x$  轴或其他极端例子的函数，它们通常可以重新定义。近似解只对连续函数有效。

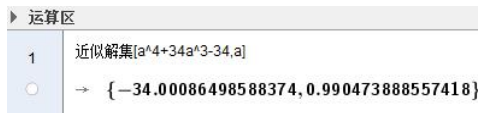
参见“精确解”指令和“近似解集”指令。

### 2.5.9 NSolutions. 近似解集

**NSolutions**(**<Equation>**); **近似解集**(**<方程>**)。

找到给定变量的方程的数值近似解集。

**案例**：“近似解集( $\cos(x)=x$ )”得出“ $\{0.7390851332151606\}$ ”。



这个指令以下用法仅适用于运算区。

**NSolutions**(**<Equation>**,**<Variable>**); **近似解集**(**<方程>**,**<变量>**)。

找到给定未知变量方程的近似解集。

**案例**：“近似解集( $a^4+34a^3-34, a$ )”得出结果为列表为“ $\{-34.000864985883736479458, 0.99047388855741781044881\}$ ”。

**NSolutions**(**<Equation>**,**<Variable=starting value>**); **近似解集**(**<方程>**,**<变量=初值>**)

找到给定未知变量方程未知变量初值的近似解集。

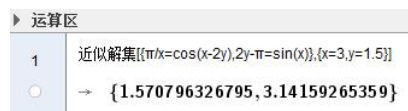
**案例**：“近似解集( $\cos(x)=x, x=0$ )”得出“ $\{0.74\}$ ”；“近似解集( $a^4+34a^3=34, a=3$ )”得出“ $\{0.99\}$ ”。



**NSolutions**(**<List of Equations>**,**<List of Variables>**); **近似解集**(**<方程组>**,**<变量列表>**)。

找到给定未知变量方程组的近似解集。

**案例**：“近似解集( $\{\pi/x=\cos(x-2y), 2y-\pi=\sin(x)\}, \{x=3, y=1.5\}$ )”得出“ $\{3.141592651686591, 1.570796327746508\}$ ”。



**注**：可选起始值如  $\{x=3, y=1.5\}$ ；可以按“Alt+p”得到“ $\pi$ ”；

**注**：如果没有给出  $a=3$  或  $\{x=3, y=1.5\}$  这样的起点，那么数值算法可能会发现很难找到近似解集（给出一个起点也不能保证有近似解）。结果小数位数取决于选项中设置的精确度。近似解集不适用于渐近  $x$  轴或其他极端例子的函数，它们通常可以重新定义。近似解集只对连续函数有效。参见“解集”指令和“近似解”指令。

## 2.5.10 CompleteSquare. 配方式

CompleteSquare(<Quadratic Function>); 配方式(<二次函数>)。

返回二次函数的完全平方形式:  $a(x-h)^2+k$ 。

案例: “配方式( $x^2-4x+7$ )” 输出 “ $g(x)=1(x-2)^2+3$ ”, 并绘制出函数图象。



CAS Syntax (运算区语法)

CompleteSquare(<Quadratic Function>); 配方式(<二次函数>)。

返回二次函数的完全平方形式:  $a(x-h)^2+k$ 。

案例: “配方式( $x^2-4x+7$ )” 输出 “ $(x-2)^2+3$ ”。

注: 两处区别在于前边的系数 a 为 1 时, 运算区会化简。

## 2.5.11 PreviousPrime. 前一质数

PreviousPrime(<Number>); 前一质数(<数值>)。

返回比输入数值小的最大质数。

案例: “前一质数(10000)” 输出 “a=9973”。



注: 运算区通用; 参见 “后一质数” 指令。

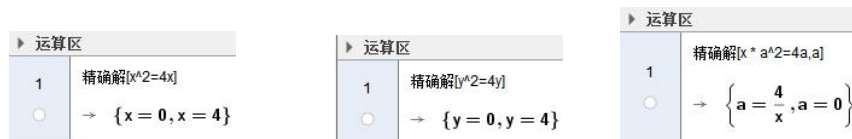
## 2.5.12 Solve. 精确解

注: “精确解” 和 “解集” 指令都是在实数范围内解方程或方程组。求解数值方程, 请用 “近似解” 指令。解复数方程, 使用 “复数解” 指令。

Solve(<Equation in x>); 精确解(<关于 x 的方程>)。

解指定主变量 x 的方程且给出解集列表。

案例: “精确解( $x^2=4x$ )” 得出方程 “ $x^2=4x$ ” 的解集 “ $\{x=0, x=4\}$ ”。



这个指令以下用法仅适用于运算区。

Solve(<Equation>, <Variable>); 精确解(<方程>, <变量>)。

解未知主变量的方程且返回解集列表。

案例: “精确解( $x * a^2=4a, a$ )” 得出方程 “ $xa^2=4a$ ” 的解集 “ $\{a=\frac{4}{x}, a=0\}$ ”

Solve(<List of Equations>, <List of Variables>); 精确解(<方程组>, <变量列表>)。

解给定未知变量列表的方程组且返回解列表。

**案例：**“精确解( $\{x=4x+y, y+x=2\}, \{x, y\}$ )”得出方程组“ $x=4x+y$ ”和“ $y+x=2$ ”的解列表“ $(x=-1, y=3)$ ”。“精确解( $\{2a^2+5a+3=b, a+b=3\}, \{a, b\}$ )”得出“ $\{a=0, b=3\}, \{a=-3, b=6\}$ ”。



**注：**方程的右边可以省略（如上述某些语句）。如果右边省略了，视为等于0。

**Solve(<Equations>, <Variables>, <List of assumptions>); 精确解(<方程>, <变量表>, <假设列表>)**。

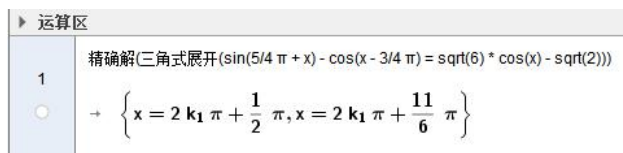
按照假设列表条件解给定未知变量的方程且返回解列表。

**案例：**“精确解( $u*x < a, x, u > 0$ )”得出当“ $u > 0$ ”时，方程“ $u*x < a$ ”的解“ $x < a/u$ ”。

“精确解( $u*x < a, x, \{u < 0, a < 0\}$ )”得出“ $x > a/u$ ”



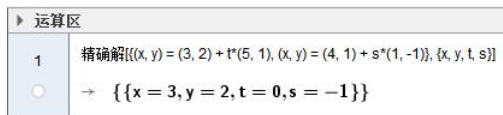
有时需要做一些操作以允许计算器工作，如“精确解(三角式展开( $\sin(5/4 \pi + x) - \cos(x - 3/4 \pi) = \sqrt{6} * \cos(x) - \sqrt{2}$ )))”



**Solve(<List of Parametric Equations>, <List of Variables>); 精确解(<参数方程组>, <变量列表>)**。

解给定未知变量列表的参数方程组且返回解列表。

**案例：**“精确解( $\{(x, y) = (3, 2) + t*(5, 1), (x, y) = (4, 1) + s*(1, -1)\}, \{x, y, t, s\}$ )”得出列表结果“ $\{x=3, y=2, t=0, s=-1\}$ ”。



**注：**如果方程的右边空着，默认右边为0。有时，需要对方程原式变通一下，才能求解。

对于分段函数，需要使用“近似解”指令。

**注：**自 GeoGebra 4.4 开始，可以解参数方程；解分段定义的函数，需要使用“近似解”；参见“解集”、“近似解”和“复数解”指令。

### 2.5.13 Mod. 取余

**Mod(<Dividend Number>, <Divisor Number>); 取余(<被除数>, <除数>)**。

输出被除数除以除数所得的余数。

**案例：**“取余(9, 4)”输出“a=1”。

**Mod(<Dividend Polynomial>, <Divisor Polynomial>); 取余(<被除式\_整式>, <除式\_整式>)**。

输出被除式除以除式得到的余数。

**案例：**“取余( $x^3+x^2+x+6, x^2-3$ )”输出“ $f(x)=4x+9$ ”，并在坐标系中绘制此函数。

### CAS Syntax (运算区语法)

$\text{Mod}(\langle \text{Dividend Number} \rangle, \langle \text{Divisor Number} \rangle)$ ; 取余(⟨被除数⟩, ⟨除数⟩)。

输出被除数除以除数所得的余数。

案例：“取余(9, 4)”输出“1”。

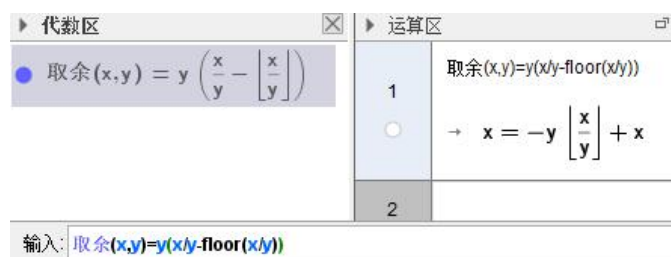
$\text{Mod}(\langle \text{Dividend Polynomial} \rangle, \langle \text{Divisor Polynomial} \rangle)$ ; 取余(⟨被除式\_整式⟩, ⟨除式\_整式⟩)。

输出被除式除以除式得到的余数。

案例：“取余( $x^3+x^2+x+6, x^2-3$ )”输出“ $4x+9$ ”。



注：如果想对一个函数求余，可以自定义，如“取余( $x, y$ )= $y(x/y - \text{floor}(x/y))$ ”。



### 2.5.14 Div. 取整

$\text{Div}(\langle \text{Dividend Number} \rangle, \langle \text{Divisor Number} \rangle)$ ; 取整(⟨被除数\_整数⟩, ⟨除数\_整数⟩)。

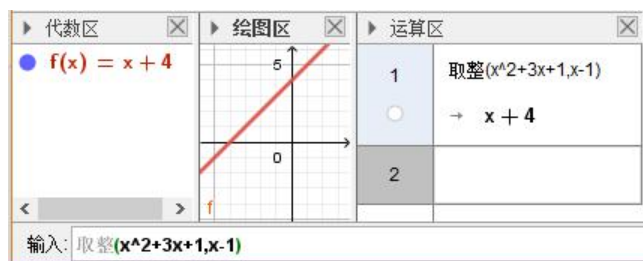
返回两个数值的整数商(得数结果的整数部分)。

案例：“取整(16, 3)”输出“a=5”。

$\text{Div}(\langle \text{Dividend Polynomial} \rangle, \langle \text{Divisor Polynomial} \rangle)$ ; 取整(⟨被除式\_整式⟩, ⟨除式\_整式⟩)。

返回两个多项式的商。

案例：“取整( $x^2+3x+1, x-1$ )”输出“ $f(x)=x+4$ ”并绘制出函数图象。



### CAS Syntax (运算区语法)

$\text{Div}(\langle \text{Dividend Number} \rangle, \langle \text{Divisor Number} \rangle)$ ; 取整(⟨被除数\_整数⟩, ⟨除数\_整数⟩)。

返回两个数值的商(得数结果的整数部分)。

案例：“取整(16, 3)”输出“5”。

$\text{Div}(\langle \text{Dividend Polynomial} \rangle, \langle \text{Divisor Polynomial} \rangle)$ ; 取整(⟨被除式\_整式⟩, ⟨除式\_整式⟩)。

返回两个多项式的商。

**案例：**“取整( $x^2+3x+1, x-1$ )”输出“ $x+4$ ”。

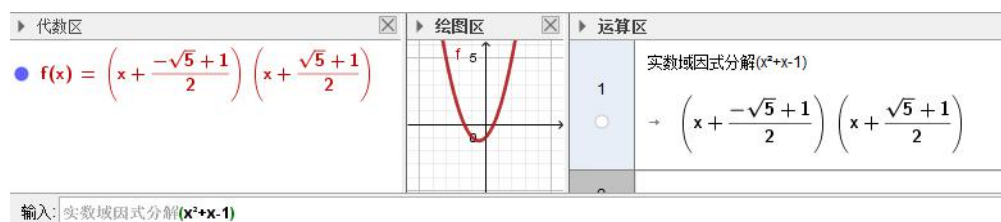
**注：**在运算区得到的多项式整数商不是方程格式的函数，是多项式。

### 2.5.15 IFactor. 实数域因式分解

**IFactor(<Polynomial>); 实数域因式分解(<多项式>)。**

在可能的情况下，分解出多项式含有无理数的因式。

**案例：**“实数域因式分解( $x^2+x-1$ )”，输出“ $f(x) = \left(x + \frac{-\sqrt{5}+1}{2}\right) \left(x + \frac{\sqrt{5}+1}{2}\right)$ ”



**CAS Syntax (运算区语法)**

**IFactor(<Expression>); 实数域因式分解(<表达式>)。**

在可能的情况下，分解出多项式含有无理数的因式。

**案例：**实数域因式分解( $x^2+x-1$ )，输出“ $\left(x + \frac{-\sqrt{5}+1}{2}\right) \left(x + \frac{\sqrt{5}+1}{2}\right)$ ”

**IFactor(<Expression>, <Variable>); 实数域因式分解(<表达式>, <变量>)。**

表达式关于指定变量含有无理数的因式。

**案例：**实数域因式分解( $a^2+a-1, a$ )，输出“ $\left(a + \frac{-\sqrt{5}+1}{2}\right) \left(a + \frac{\sqrt{5}+1}{2}\right)$ ”

**注：**参见“因式分解”指令。

### 2.5.16 IsPrime. 是否为质数

**IsPrime(<Number>); 是否为质数(<数字>)**

根据数值是否为质数得出 true 或 false。

**案例：**“是否为质数(10)”输出“a=false”；“是否为质数(11)”输出“a=true”。

**CAS Syntax (运算区语法)**

**IsPrime(<Number>); 是否为质数(<数值>)**

根据数值是否为质数得出 true 或 false。

**案例：**“是否为质数(10)”输出“false”；“是否为质数(11)”输出“true”。

### 2.5.17 Factor. 因式分解

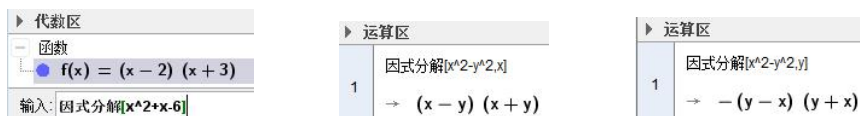
这个指令因英语变体而异：Factor(US)、Factorise(UK)、Factorise(Aus)。

**Factor(<Polynomial>); 因式分解(<多项式>)。**

分解出多项式的因式。

**案例：**“因式分解( $x^2+x-6$ )”输出“ $f(x)=(x+3)(x-2)$ ”且绘制出函数图象。

**注：**这个指令需要加载代数计算系统，故某些电脑会变慢。



### CAS Syntax (运算区语法)

**Factor(<Polynomial>); 因式分解(<多项式>)**。

分解出多项式的因式。

**案例：**“因式分解( $x^2-y^2$ )”输出“ $(x+y)(x-y)$ ”。

**Factor(<Expression>, <Variable>); 因式分解(<表达式>, <变量>)**。

表达式关于指定变量分解的因式。

**案例：**“因式分解( $x^2-y^2, x$ )”输出“ $(x+y)(x-y)$ ”，即“ $x^2-y^2$ ”关于  $x$  的分解因式；“因式分解( $x^2-y^2, y$ )”输出“ $-(y-x)(y+x)$ ”，即“ $x^2-y^2$ ”关于  $y$  的分解因式。

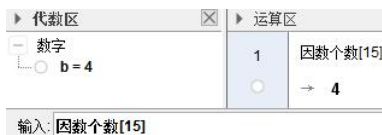
**注：**这个指令能超越有理数分解表达式。在无理数范围内分解因式，参见“实数域因式分解”指令。在复数范围内因式分解，参见“复数因数分解”指令和“复无理数域因式分解”指令。

## 2.5.18 Divisors. 因数个数

**Divisors(<Number>); 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。

**案例：**“因数个数(15)”输出“4”，数值 15 的所有正因数个数，包括 15。



### CAS Syntax (运算区语法)

**Divisors(<Number>); 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。

**案例：**“因数个数(15)”输出“4”，数值 15 的所有正因数个数，包括 15。

**注：**参见“因数列表”指令和“因数和”指令。

## 2.5.19 DivisorsSum. 因数和

**DivisorsSum(<Number>); 因数和(正整数)**。

计算全部的正因数之和，包括指定数值本身。

**案例：**“因数和(15)”输出“a=24”，为“1+3+5+15”之和。



CAS Syntax (运算区语法)

**DivisorsSum(<Number>); 因数和(正整数)。**

计算全部的正因数之和, 包括指定数值本身。

**案例:** “因数和(15)” 输出 “24”, 为 “1+3+5+15” 之和。

**注:** 参见 “因数个数” 指令和 “因数列表” 指令。

## 2.5.20 DivisorsList. 因数列表

**DivisorsList(<Number>); 因数列表(<正整数>)。**

给出全部正因数的列表, 包括指定数值本身。

**案例:** “因数列表(15)” 输出 “列表 1={1, 3, 5, 15}”, 15 的全部正因数列表, 包括 15。



CAS Syntax (运算区语法)

**DivisorsList(<Number>); 因数列表(<正整数>)。**

给出全部正因数的列表, 包括指定数值本身。

**案例:** “因数列表(15)” 输出 “{1, 3, 5, 15}”, 15 的全部正因数集合, 包括 15。

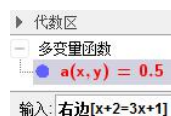
**注:** 参见 “因数个数” 指令和 “因数和” 指令。

## 2.5.21 RightSide. 右边

**RightSide(<Equation>); 右边(<方程>)。**

给出化简后方程式的右边。

**案例:** “右边(x+2=3x+1)” 输出 “a(x, y)=0.5”。



CAS Syntax (运算区语法)

**RightSide(<Equation>); 右边(<方程>)。**

给出方程式的右边。

**案例:** “右边(x+3=3x+1)” 输出 “3x+1”。

**RightSide(<List of Equations>); 右边(<方程列表>)。**

给出方程式列表中各个方程的右边列表。

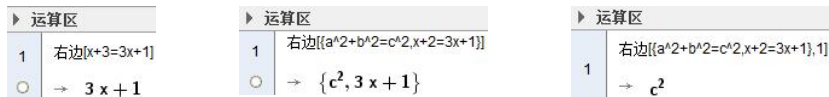
**案例:** “右边({a^2+b^2=c^2, x+2=3x+1})” 输出 “{c^2, 3x+1}”。

**RightSide(<List of Equations>, <Index>); 右边(<方程列表>, <索引>)。**

给出列表中索引指定方程式的右边。

**案例:** “右边({a^2+b^2=c^2, x+2=3x+1}, 1)” 输出 “c^2”。





注：参见“左边”指令。

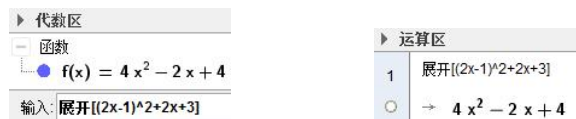
## 2.5.22 Expand. 展开

**Expand(<Expression>); 展开(表达式)**

展开表达式。

**案例：**“展开 $((2x-1)^2+2x+3)$ ”输出“ $f(x)=4x^2-2x+4$ ”并绘制出函数图象。

**注：**这个指令需要加载代数计算系统，故某些电脑会变慢。请尝试使用多项式指令替代。



**CAS Syntax (运算区语法)**

**Expand(<Expression>)**

展开表达式。

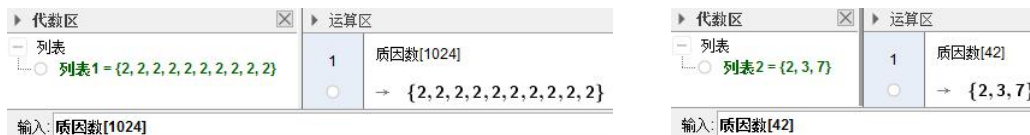
**案例：**“展开 $((2x-1)^2+2x+3)$ ”输出“ $4x^2-2x+4$ ”。

## 2.5.23 PrimeFactors. 质因数

**PrimeFactors(<Number>); 质因数(<数字>)。**

返回质因数列表，其中元素的乘积等于指定数值。

**案例：**“质因数(1024)”输出“列表1={2, 2, 2, 2, 2, 2, 2, 2, 2}”；“质因数(42)”输出“列表1={2, 3, 7}”。



**CAS Syntax (运算区语法)**

**PrimeFactors(<Number>); 质因数(<数字>)。**

返回数值的质因数列表，其中元素的乘积等于指定数值。

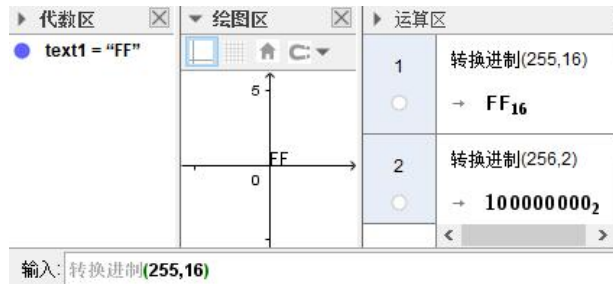
**案例：**“质因数(1024)”输出“{2, 2, 2, 2, 2, 2, 2, 2, 2}”；“质因数(42)”输出“{2, 3, 7}”。

## 2.5.24 ToBase. 转换进制

**ToBase(<Number>, <Base>); 转换进制(<十进制数>, <目标进制\_2-36>)。**

将指定数值转化为不同的进制（底数），在2与36之间，数值必须是整数。

**案例：**“转换进制(255, 16)”在绘图区返回文本“FF”；“转换进制(256, 2)”在绘图区返回文本“100000000”。



注：参见“转换为十进制”指令。

### 2.5.25 FromBase. 转换为十进制

**FromBase**("<Number as Text>", <Base>); 转换为十进制 ("<指定进制数值>", <进制数 2-36>)。  
以给定的基数（进制）将指定数值转换为十进制。基数必须在 2 与 36 之间。数值必须是一个整数。

案例：“转换为十进制("FF", 16)”返回“a=255”；“转换为十进制("100000000", 2)”返回“a=256”。



注：参见“进制形式”指令。运算区通用。

### 2.5.26 GCD. 最大公约数

这个指令因英语变体而异：：GCD Command (US) 和 HCF Command (UK + Aus)。

**GCD**(<Number>, <Number>); 最大公约数 (<整数 1>, <整数 2>)。

计算两个数值的最大公约数。

案例：“最大公约数(12, 15)”输出“a=3”。

**GCD**(<List of Numbers>); 最大公约数 (<整数列表>)。

计算数集中全部元素的最高公约数。

案例：“最大公约数({12, 30, 18})”输出“a=6”。



**CAS Syntax (运算区语法)**

**GCD**(<Number>, <Number>); 最大公约数 (<整数 1>, <整数 2>)。

计算两个数值的最大公约数。

案例：“最大公约数(12, 15)”输出“3”。

**GCD**(<List of Numbers>); 最大公约数 (<整数列表>)。

计算数列表中全部元素的最高公约数。

案例：“最大公约数({12, 30, 18})”输出“6”。

**GCD**(<Polynomial>, <Polynomial>); 最大公约数 (<多项式 1>, <多项式 2>)。

案例：“最大公约数( $x^2+4x+4, x^2-x-6$ )”输出“ $x+2$ ”。

GCD(<List of Polynomials>); 最大公约数(<多项式列表>)。

计算多项式列表中各多项式的最大公约式。

案例：“最大公约数( $\{x^2+4x+4, x^2-x-6, x^3-4x^2-3x+18\}$ )”输出“ $x+2$ ”。



## 2.5.27 Max. 最大值

Max(<Interval>); 最大值(<区间 如:2<x<3>)

返回区间上界。

案例：“最大值(2<x<3)”输出“a=3”。

注：开闭区间同样对待。

Max(<List>); 最大值(<数字列表>)。

返回列表中数值的最大值。

案例：“最大值( $\{-2, 12, -23, 17, 15\}$ )”输出“a=17”。

注：若输入的元素包含非数值对象，指令则处理这些对象所关联的数值。例如元素为线段，最大值指令将得出线段的最大长度。

Max(<Number>, <Number>); 最大值(<数字 1>, <数字 2>)

返回两个给定数值的最大者。

案例：“最大值(12, 15)”输出“a=15”。



Max(<List of Data>, <List of Frequencies>); 最大值(<数据列表>, <频数列表>)。

返回数据列表中与频数大于 0 的最大值。

案例：“最大值( $\{1, 2, 3, 4, 5\}, \{5, 3, 4, 2, 0\}$ )”输出“a=4”。原理是：列表中的 1 出现 5 次，2 出现 3 次，3 出现 4 次，4 出现 2 次，5 出现 0 次。所有出现的数值中，最大的就是 4。

Max(<Function>, <Start x-Value>, <End x-Value>); 最大值(<函数>, <x-起始值>, <x-终止值>)。

计算函数在指定区间上最大值所对应的点。函数应在区间上连续并且只有一个局部极大值所对应的点。

案例：“最大值( $x^3+2x^2-1, -2, 0$ )”创建点“A=(-1.33, 0.19)”并在坐标系中绘制出点 A。



注：参见“极值点”指令、“最小值”指令和“函数检视”工具。

CAS Syntax (运算区语法)

**Max(<List>); 最大值(<数字列表>)**。

返回列表中数值的最大值。

**案例：**“最大值({-2, 12, -23, 17, 15})”输出“17”。

**注：**参见“极值点”指令和“最小值”指令。

**Max(<Number>, <Number>); 最大值(<数字 1>, <数字 2>)**

返回两个给定数值的最大者。

**案例：**“最大值(12, 15)”输出“15”。



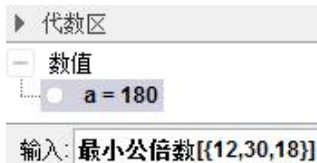
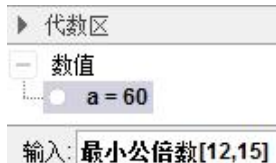
## 2.5.28 LCM. 最小公倍数

UK 英语: LCM=lowest common multiple

**LCM(<List of Numbers>); 最小公倍数(<整数列表>)**。

计算数列表中全部元素的最小公倍数。

**案例：**“最小公倍数({12, 30, 18})”输出“a=180”。



**LCM(<Number>, <Number>); 最小公倍数(<整数 1>, <整数 2>)**

计算两个数值的最小公倍数。

**案例：**“最小公倍数(12, 15)”输出“a=60”。

**CAS Syntax (运算区语法)**

**LCM(<List of Numbers>); 最小公倍数(<整数列表>)**。

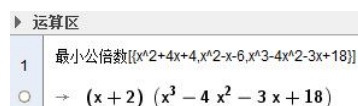
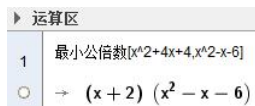
计算数列表中全部元素的最小公倍数。

**案例：**“最小公倍数({12, 30, 18})”输出“180”。

**LCM(<List of Polynomials>); 最小公倍数(<多项式列表>)**。

计算多项式列表中各个多项式的最小公倍数。

**案例：**“最小公倍数({ $x^2+4x+4$ ,  $x^2-x-6$ ,  $x^3-4x^2-3x+18$ })”输出“ $x^4-2x^3-11x^2+12x+36$ ”。



**LCM(<Number>, <Number>); 最小公倍数(<整数 1>, <整数 2>)**

计算两个数值的最小公倍数。

**案例：**“最小公倍数(12, 15)”输出“60”。

**LCM(<Polynomial>, <Polynomial>); 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。

**案例：**“最小公倍数( $x^2+4x+4, x^2-x-6$ )”输出“ $x^3+x^2-8x-12$ ”。

**注：**实际操作中，多项式的最小公倍多项式并不是分解因式格式，如：输出结果是“ $(x+2)(x^3-4x^2-3x+18)$ ”。如果继续做因式分解是：“ $(x-3)^2(x+2)^2$ ”。

## 2.5.29 Min. 最小值

**Min(<Interval>); 最小值(<区间 如  $2<x<3$ >)**。

返回区间的下界值。

**案例：**“最小值( $2<x<3$ )”输出“ $a=2$ ”。

**注：**区间的开闭没有区别。

**Min(<List>); 最小值(<数字列表>)**。

返回列表中数值的最小值。

**案例：**“最小值( $\{-2, 12, -23, 17, 15\}$ )”输出“ $a=-23$ ”。

**注：**若输入的元素包含非数值对象，指令则处理这些对象所关联的数值。例如元素为线段，最大值指令将得出线段的最小长度。

**Min(<Number>, <Number>); 最小值(<数值 1>, <数值 2>)**。

返回两个给定数值的最小者。

**案例：**“最小值(12, 15)”输出“ $a=12$ ”。



**Min(<List of Data>, <List of Frequencies>); 最小值(<数据列表>, <频数列表>)**。

返回数据列表中与频数大于 0 的最小值。

**案例：**“最小值( $\{1, 2, 3, 4, 5\}, \{0, 3, 4, 2, 3\}$ )”输出“ $a=2$ ”。原理是：列表中 1 出现了 0 次，2 出现了 3 次，3 出现了 4 次，4 出现了 2 次，5 出现了 3 次，所有出现的数中，最小值是 2。



**Min(<Function>, <Start x-Value>, <End x-Value>); 最小值(<函数>, <x-起始值>, <x-终止值>)**。

计算(数字表示)给定函数区间的最小值点。函数连续且在本区间内只有一个最小值点。

**案例：**“最小值( $x^3+2x^2-1, -2, 0$ )”创建点“ $A(0, -1)$ ”并在坐标系中绘制点 A。

**CAS Syntax (运算区语法)**

**Min(<List>); 最小值(<列表>)**。

返回列表中数值的最小者。

**案例：**“最小值( $\{-2, 12, -23, 17, 15\}$ )”输出“ $-23$ ”。

**Min(<Number>, <Number>); 最小值(<数值 1>, <数值 2>)**。

返回两个给定值的最小者。

**案例：**“最小值(12, 15)”输出“12”。

注：参见“最大值”指令、“极值点”指令和“函数检视”工具。



### 2.5.30 LeftSide. 左边

LeftSide(<Equation>); 左边(<方程>).

给出最简方程的左边。

案例：“左边(x+2=3x+1)”输出“a(y, x)=x”。

CAS Syntax (运算区语法)

LeftSide(<Equation>); 左边(<方程>).

给出方程的左边。

案例：“左边(x+3=3x+1)”输出“x+3”。

LeftSide(<List of Equations>); 左边(<方程列表>).

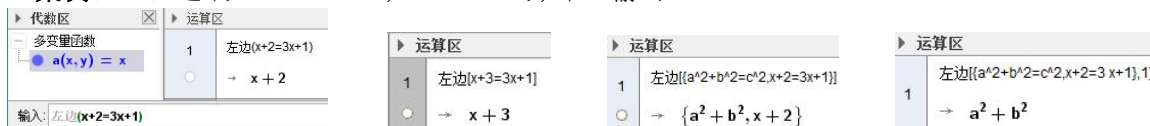
给出方程式列表中各个方程的左边列表。

案例：“左边({a^2+b^2=c^2, x+2=3x+1})”输出“{a^2+b^2, x+2}”。

LeftSide(<List of Equations>, <Index>); 左边(<方程列表>, <索引>)

给出列表中索引指定方程式的左边。

案例：“左边({a^2+b^2=c^2, x+2=3 x+1}, 1)”输出“a^2+b^2”。



注：参见“右边”指令。

## 2.6 Probability. 概率

### 2.6.1 FDistribution. F 分布

FDistribution(<Numerator Degrees of Freedom>, <Denominator Degrees of Freedom>, x );

F 分布(<分子自由度>, <分母自由度>, <变量值>).

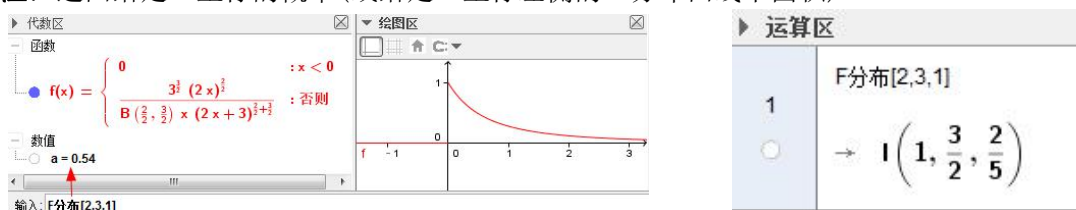
创建依据参数 n 和 d 的 F 分布密度概率函数 (n 是分子自由度, d 是分母自由度)。



FDistribution(<Numerator Degrees of Freedom>, <Denominator Degrees of Freedom>, <Variable Value>); F 分布(<分子自由度>, <分母自由度>, <变量值>).

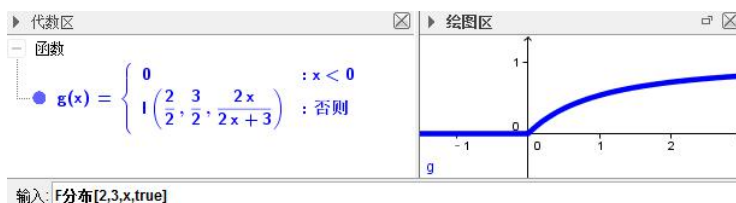
计算 F 分布的变量值  $v$  处的累积分布函数，自变量  $x$  为  $n$  和  $d$  ( $n$  是分子自由度， $d$  是分母自由度) 的 F 分布概率  $P(X \leq v)$ 。

**注：** 返回给定  $x$  坐标的概率 (或给定  $x$  坐标左侧的 F 分布曲线下面积)。



$\text{FDistribution}(\langle \text{Numerator Degrees of Freedom} \rangle, \langle \text{Denominator Degrees of Freedom} \rangle, x, \langle \text{Boolean Cumulative} \rangle)$ ; F 分布 ( $\langle \text{分子自由度} \rangle, \langle \text{分母自由度} \rangle, x, \langle \text{是否累积?true|false} \rangle$ )。

如果累积是 true，创建 F 分布的累积概率函数 (如下图)，否则创建 F 分布概率分布函数 (如上图)。



CAS Syntax (运算区语法)

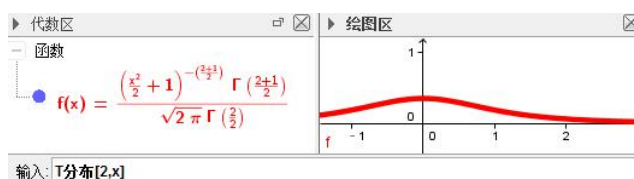
$\text{FDistribution}(\langle \text{Numerator Degrees of Freedom} \rangle, \langle \text{Denominator Degrees of Freedom} \rangle, \langle \text{Variable Value} \rangle)$ ; F 分布 ( $\langle \text{分子自由度} \rangle, \langle \text{分母自由度} \rangle, \langle \text{变量值} \rangle$ )。

计算 F 分布的变量值  $v$  处的累积分布函数，自变量  $x$  为  $n$  和  $d$  ( $n$  是分子自由度， $d$  是分母自由度) 的 F 分布概率  $P(X \leq v)$ 。

## 2.6.2 TDistribution. t 分布

$\text{TDistribution}(\langle \text{Degrees of Freedom} \rangle, x)$ ; t 分布 ( $\langle \text{自由度} \rangle, \langle \text{变量值} \rangle$ )。

由给定的自由度创建 T 分布密度概率函数 (pdf)。

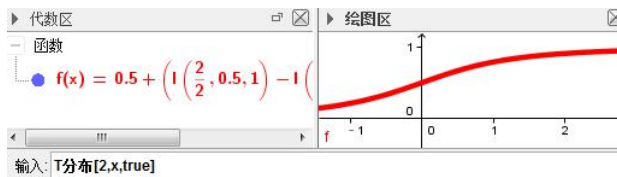


$\text{TDistribution}(\langle \text{Degrees of Freedom} \rangle, \langle \text{Variable Value} \rangle)$ ; t 分布 ( $\langle \text{自由度} \rangle, \langle \text{变量值} \rangle, \langle \text{是否累积?true|false} \rangle$ )。

计算 T 分布的变量值  $v$  处的累积分布函数值，如给定平均数  $\mu$  和标准差  $\sigma$  的自变量为  $x$  的 T 分布的概率  $P(X \leq v)$ 。

**案例：** “T 分布 (10, 0)” 得出 0.5。

**注：** 返回给定  $x$  坐标的概率值 (或给定  $x$  坐标左侧的 T 分布曲线下面积)。



$\text{TDistribution}(\langle \text{Degrees of Freedom} \rangle, x, \langle \text{Boolean Cumulative} \rangle)$ ; t 分布 ( $\langle \text{自由度} \rangle, x, \langle \text{是否累积?true|false} \rangle$ )。

如果累积是 true，创建累积 T 分布函数，否则创建 T 分布密度概率函数。

CAS Syntax (运算区语法)

$\text{TDistribution}(\langle \text{Degrees of Freedom} \rangle, \langle \text{Variable Value} \rangle)$ ; t 分布 ( $\langle \text{自由度} \rangle, \langle \text{变量值} \rangle$ )。

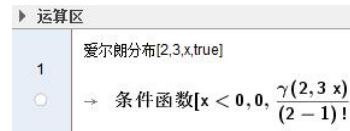
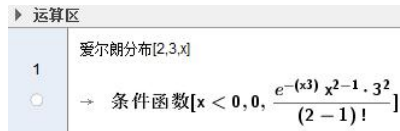
计算 T 分布的变量值  $v$  处的累积分布函数值，如给定平均数  $\mu$  和标准差  $\sigma$  的自变量为  $x$  的 T 分布的概率  $P(X \leq v)$ 。

案例：“T 分布(10, 0)” 得出 0.5。

### 2.6.3 Erlang. 爱尔朗分布

$\text{Erlang}(\langle \text{Shape} \rangle, \langle \text{Rate} \rangle, \langle \text{Variable Value} \rangle)$ ; 爱尔朗分布 ( $\langle \text{形状参数 } k \rangle, \langle \text{比率参数 } \lambda \rangle, \langle \text{变量值} \rangle$ )。

计算爱尔朗分布的变量值  $v$  处的累积分布函数，自变量  $x$  为适当自由度的爱尔朗分布概率  $P(X \leq v)$ 。

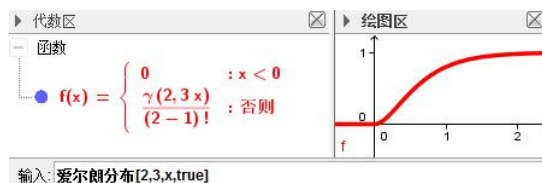
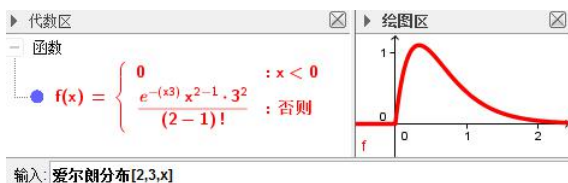


注：返回给定  $x$  坐标的概率(或给定  $x$  坐标左侧的爱尔朗分布曲线下面积)。

$\text{Erlang}(\langle \text{Shape} \rangle, \langle \text{Rate} \rangle, \langle \text{Variable Value} \rangle, \langle \text{Boolean Cumulative} \rangle)$ ; 爱尔朗分布 ( $\langle \text{形状参数 } k \rangle, \langle \text{比率参数 } \lambda \rangle, \langle \text{变量值} \rangle, \langle \text{是否累积?true|false} \rangle$ )。

$\text{Erlang}(\langle \text{Shape} \rangle, \langle \text{Rate} \rangle, x, \langle \text{Boolean Cumulative} \rangle)$ ; 爱尔朗分布 ( $\langle \text{形状参数 } k \rangle, \langle \text{比率参数 } \lambda \rangle, x, \langle \text{是否累积?true|false} \rangle$ )。

如果逻辑值是 true，创建爱尔朗分布累积分布函数（下右图），否则创建爱尔朗分布函数（下左图）。



CAS Syntax (运算区语法)

$\text{Erlang}(\langle \text{Shape} \rangle, \langle \text{Rate} \rangle, x)$ ; 爱尔朗分布 ( $\langle \text{形状参数 } k \rangle, \langle \text{比率参数 } \lambda \rangle, x$ )。

创建依据形状参数  $k$  和比率参数  $\lambda$  的爱尔朗分布密度概率函数。



**Erlang(<Shape>, <Rate>, x, <Boolean Cumulative>); 爱尔朗分布 (<形状参数 k>, <比率参数  $\lambda$ >, x, <是否累积?true|false>)**。

如果逻辑值是 true，创建爱尔朗分布累积分布函数，否则创建爱尔朗分布函数。

**Erlang(<Shape>, <Rate>, <Variable Value>); 爱尔朗分布 (<形状参数 k>, <比率参数  $\lambda$ >, <变量值>)**。

计算爱尔朗分布的变量值 v 处的累积分布函数，自变量 x 为适当自由度的爱尔朗分布概率  $P(X \leq v)$ 。

**注：**返回给定 x 坐标的概率(或给定 x 坐标左侧的爱尔朗分布曲线下面积)。

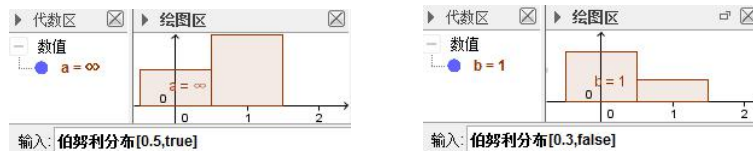
**释义：**爱尔朗分布 (ErlangDistribution) 是一种连续型概率分布，又称爱尔兰分布、埃朗分布、埃尔朗分布、爱尔朗分布、厄朗分布等等；它与指数分布一样多用来表示独立随机事件发生的时间间隔。相比于指数分布，它能更好地对现实数据进行拟合（更适用于多个串行过程，或无记忆性假设不显著的情况下）。除非退化为指数分布，它不具有无记忆性（或马尔可夫性质），因此对其进行分析相对困难一些。一般通过将爱尔朗过程分解为多个指数过程的技巧来对爱尔朗分布进行分析。遵循爱尔朗分布的随机变量可以被分解多个同参数指数分布随机变量之和，该性质使得爱尔朗分布被广泛用于排队论中。

### 2.6.4 Bernoulli. 伯努利分布

**Bernoulli(<Probability p>, <Boolean Cumulative>); 伯努利分布 (<概率>, <是否累积?true|false>)**。

如果累积为 “false”，返回成功概率等于 p 的伯努利分布条形图。

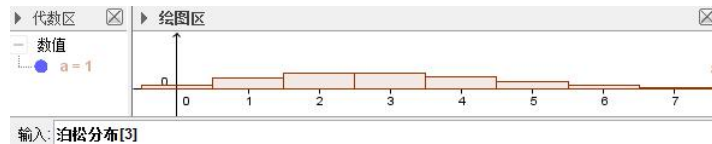
如果累积为 “true”，返回累积伯努利概率分布条形图。



### 2.6.5 Poisson. 泊松分布

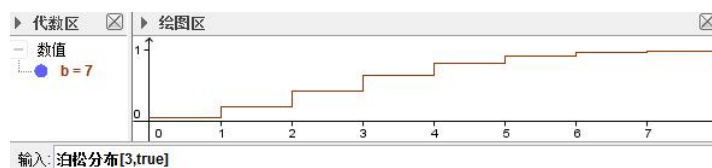
**Poisson(<Mean  $\lambda$ >); 泊松分布 (<平均数>)**

返回给定平均数的泊松分布条形图。



**Poisson(<Mean  $\lambda$ >, <Boolean Cumulative>); 泊松分布 (<平均数>, <是否累积?true|false>)**。

当累积为 “false”，返回一个泊松分布条形图。当累积为 “true”，返回一个累积泊松分布条形图（如上图）。第一个参数同上。

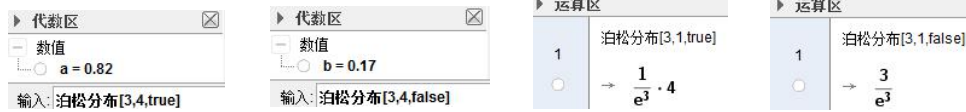


**Poisson(<Mean  $\lambda$ >, <Variable Value v>, <Boolean Cumulative>); 泊松分布 (<平均数>, <变**

量值), <是否累积?true|false>。

设  $X$  是泊松随机变量且  $v$  是变量值。

当累积是“false”，返回  $P(X=v)$ 。当累积是“true”，返回  $P(X \leq v)$ 。第一个参数同上。



CAS Syntax (运算区语法)

Poisson(<Mean  $\lambda$ >, <Variable Value  $v$ >, <Boolean Cumulative>); 泊松分布(<平均数>, <变量值>, <是否累积?true|false>)。

设  $X$  是泊松随机变量且  $v$  是变量值。

当累积是“false”，返回  $P(X=v)$ 。

当累积是“true”，返回  $P(X \leq v)$ 。

第一个参数同上。

案例：“泊松分布(3, 1, true)”得出  $\frac{4}{e^3}$ ；“泊松分布(3, 1, false)”得出  $\frac{3}{e^3}$ 。

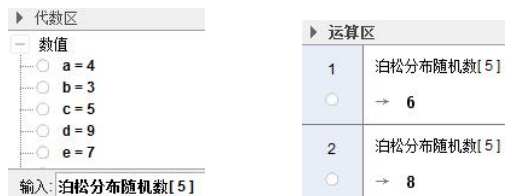
注：可以使用简化的语法来计算  $P(u \leq X \leq v)$ 。例如：泊松分布(1, 1..5)得到 0.63153，与泊松分布(1, {1, 2, 3, 4, 5})相同。

### 2.6.6 RandomPoisson. 泊松分布随机数

RandomPoisson(<Mean>); 泊松分布随机数(<平均数>)。

由一个指定平均数(均值)的泊松分布生成一个随机数。

案例：“泊松分布随机数(3)”由一个平均数(均值)为3的泊松分布得出一个随机数。



CAS Syntax (运算区语法)

RandomPoisson(<Mean>); 泊松分布随机数(<平均数>)。

由一个指定平均数(均值)的泊松分布生成一个随机数。

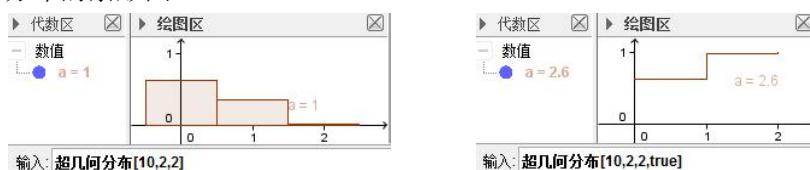
案例：“泊松分布随机数(3)”由一个平均数(均值)为3的泊松分布得出一个随机数。

注：每次运行都产生新的随机数；参见“设置种子”指令、“随机元素”指令、“区间随机数”指令、“随机二项分布数”指令、“正态分布随机数”指令和“均匀分布随机数”指令。

### 2.6.7 HyperGeometric. 超几何分布

HyperGeometric(<Population Size>, <Number of Successes>, <Sample Size>); 超几何分布(<总体容量>, <成功次数>, <样本容量>)。

返回超几何分布的条形图。



### 参数:

总体容量: 瓮中球的总数。

成功次数: 瓮中白球数。

样本容量: 从瓮中取球数。

条形图显示的是样本中白色球的数量概率函数。

**HyperGeometric(<Population Size>, <Number of Successes>, <Sample Size>, <Boolean Cumulative>); 超几何分布(<总体容量>, <成功次数>, <样本容量>, <是否累积?true|false>)**。

当累积为“false”时, 返回超几何分布概率条形图。

当累积为“true”时, 返回累积超几何分布概率条形图。

前三个参数同上。

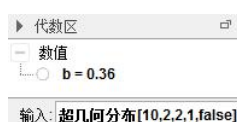
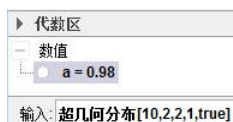
**HyperGeometric(<Population Size>, <Number of Successes>, <Sample Size>, <Variable Value>, <Boolean Cumulative>); 超几何分布(<总体容量>, <成功次数>, <样本容量>, <变量值>, <是否累积?true|false>)**

设  $X$  是超几何分布的变量且变量值为  $v$ 。

当累积为“false”时, 返回  $P(X=v)$ 。

当累积为“true”时, 返回  $P(X \leq v)$ 。

前三个参数同上。



### CAS Syntax (运算区语法)

**HyperGeometric(<Population Size>, <Number of Successes>, <Sample Size>, <Variable Value>, <Boolean Cumulative>); 超几何分布(<总体容量>, <成功次数>, <样本容量>, <变量值>, <是否累积?true|false>)**

设  $X$  是超几何分布的变量且变量值为  $v$ 。

当累积为“false”时, 返回  $P(X=v)$ 。

当累积为“true”时, 返回  $P(X \leq v)$ 。

前三个参数同上。

### 案例:

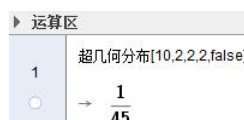
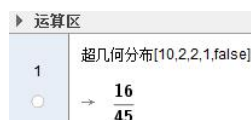
假定从 10 个球 (其中有 2 个是白色) 中取出 2 个, 不再放回, 各种情况的概率。

“超几何分布(10, 2, 2, 0, false)” 得出  $\frac{28}{45}$ , 零个白球的概率。

“超几何分布(10, 2, 2, 1, false)” 得出  $\frac{16}{45}$ , 一个白球的概率。

“超几何分布(10, 2, 2, 2, false)” 得出  $\frac{1}{45}$ , 两个都是白球的概率。

“超几何分布(10, 2, 2, 3, false)” 得出 0, 三个白球的概率。



“超几何分布(10, 2, 2, 0, true)” 得出  $\frac{28}{45}$ , 零个 (或更少) 白球的概率。

“超几何分布(10, 2, 2, 1, true)” 得出  $\frac{44}{45}$ , 一个及以下白球的概率。

“超几何分布(10, 2, 2, 2, ture)” 得出 1，二个及以下白球的概率。

“超几何分布(10, 2, 2, 3, ture)” 得出 1，三个及以下白球的概率。



## 2.6.8 LogNormal. 对数正态分布

**LogNormal(<Mean>, <Standard Deviation>, <Variable Value>); 对数正态分布(<平均数>, <标准差>, <变量值>)**。

计算对数正态分布的变量值  $v$  处的累积分布函数值，如给定平均数  $\mu$  和标准差  $\sigma$  的自变量为  $x$  的对数正态分布的概率  $P(X \leq v)$ 。

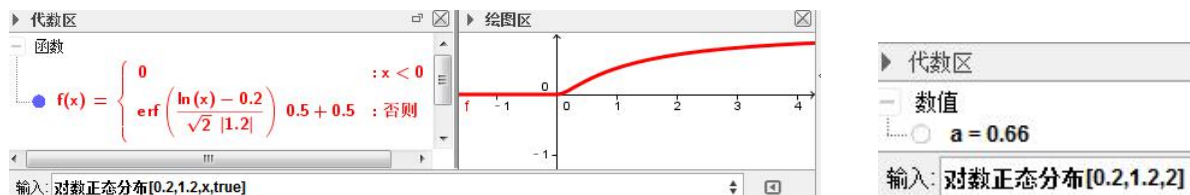
**注：**返回给定  $x$  坐标的概率值(或给定  $x$  坐标左侧的对数正态分布曲线下面积)。

**LogNormal(<Mean>, <Standard Deviation>, <Variable Value>, <Boolean Cumulative>); 对数正态分布(<平均数>, <标准差>, <变量值>, <是否累积?true|false>)**。

如果累积是“true”，创建变量值对应的对数正态分布累积概率函数，否则创建对数正态分布概率函数。

**LogNormal(<Mean>, <Standard Deviation>,  $x$ , <Boolean Cumulative>); 对数正态分布(<平均数>, <标准差>,  $x$ , <是否累积?true|false>)**。

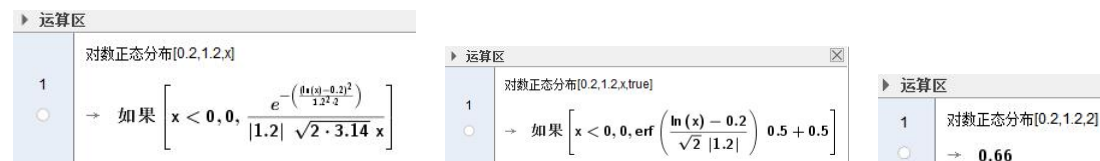
如果累积是 true，创建对数正态分布累积概率函数（如下左图），否则创建对数正态分布概率函数。



**CAS Syntax (运算区语法)**

**LogNormal(<Mean>, <Standard Deviation>,  $x$ ); 对数正态分布(<平均数>, <标准差>,  $x$ )**。

创建给定参数平均数  $\mu$  和标准差  $\sigma$  的对数正态分布密度概率函数。



**LogNormal(<Mean>, <Standard Deviation>,  $x$ , <Boolean Cumulative>); 对数正态分布(<平均数>, <标准差>,  $x$ , <是否累积?true|false>)**。

如果累积是“true”，创建对数正态分布累积概率函数，否则创建对数正态分布概率函数。

**LogNormal(<Mean>, <Standard Deviation>, <Variable Value>); 对数正态分布(<平均数>, <标准差>, <变量值>)**。

计算对数正态分布的变量值  $v$  处的累积分布函数值，如给定平均数  $\mu$  和标准差  $\sigma$  的自变量为  $x$  的对数正态分布的概率  $P(X \leq v)$ 。

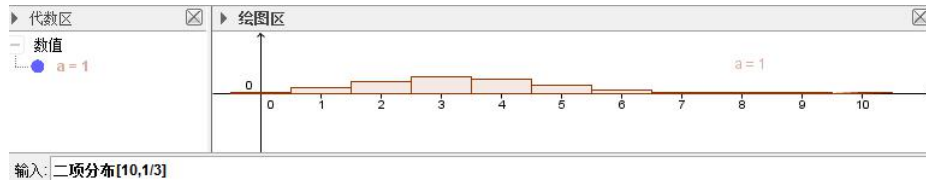
**注：**返回给定  $x$  坐标的概率值(或给定  $x$  坐标左侧的对数正态分布曲线下面积)。

## 2.6.9 BinomialDist. 二项分布

**BinomialDist(<Number of Trials>, <Probability of Success>); 二项分布(<试验次数>, <成**

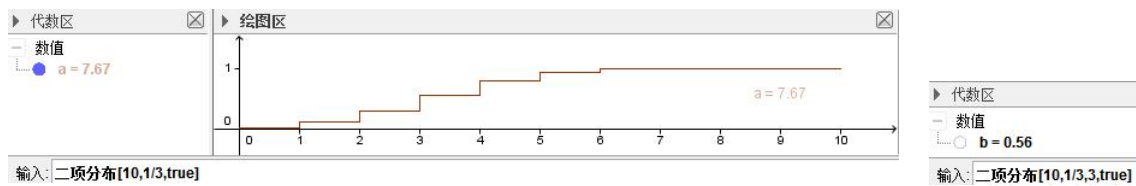
功概率> )。

返回二项分布的条形图。参数“试验次数”指定独立试验次数，参数“成功概率”是一次试验成功的概率。



**BinomialDist**(<Number of Trials>, <Probability of Success>, <Boolean Cumulative>); 二项分布(<试验次数>, <成功概率>, <是否累积?true|false> )。

当累积为“false”时，返回二项分布条形图（如上图）。当累积为“true”时，返回累积二项分布条形图。前两个参数同上。



**BinomialDist**(<Number of Trials>, <Probability of Success>, <Variable Value>, <Boolean Cumulative>); 二项分布(<试验次数>, <成功概率>, <变量值>, <是否累积?true|false> )。

设  $x$  是二项式的自变量， $v$  是变量值。

当累积是“false”，返回  $P(X=v)$ 。

当累积=true 返回  $P(X \leq v)$ 。

前两个参数同上。

#### CAS Specific Syntax (运算区特定语法)

在运算区只允许一种语法：

**BinomialDist**(<Number of Trials>, <Probability of Success>, <Variable Value>, <Boolean Cumulative>); 二项分布(<试验次数>, <成功概率>, <变量值>, <是否累积?true|false> )

设  $x$  是二项式的自变量， $v$  是变量值。

当累积是“false”，返回  $P(X=v)$ 。当累积是“true”返回  $P(X \leq v)$ 。前两个参数同上。



**案例：**假定转移三个数据包越过失误线。任意一个数据包未能移过失误线的概率是  $\frac{1}{10}$ ，因此任意一个数据包成功的概率是  $\frac{9}{10}$ 。

“二项分布 (3, 0.9, 0, false)” 得出  $\frac{1}{1000}$ ，三个包无一转移成功的概率。

“二项分布 (3, 0.9, 1, false)” 得出  $\frac{27}{1000}$ ，三个包恰好一个转移成功的概率。

“二项分布 (3, 0.9, 2, false)” 得出  $\frac{243}{1000}$ ，三个包恰好两个转移成功的概率。

“二项分布 (3, 0.9, 3, false)” 得出  $\frac{729}{1000}$ ，所有三个包都转移成功的概率。

“二项分布(3, 0.9, 4, false)” 得出 0, 三个包恰好四个转移成功的概率。



“二项分布(3, 0.9, 0, true)” 得出  $\frac{1}{1000}$ , 三个包无一转移成功的概率。

“二项分布(3, 0.9, 1, true)” 得出  $\frac{7}{250}$ , 三个包至多一个转移成功的概率。

“二项分布(3, 0.9, 2, true)” 得出  $\frac{271}{1000}$ , 三个包至多两个转移成功的概率。

“二项分布(3, 0.9, 3, true)” 得出 1, 三个包至多三个转移成功的概率。

“二项分布(3, 0.9, 4, true)” 得出 1, 三个包至多四个转移成功的概率。



### 2.6.10 Binomial (BinomialCoefficient) . 二项式系数

**BinomialCoefficient(<Number>, <Number>); 二项式系数(<数值 n>, <数值 r>).**

计算“n 选 r”的二项式系数。第一个数值代表所有元素 n, 第二个数值表示所选元素 r。

**案例:** “二项式系数(5, 3)” 得出 10。



**CAS Syntax (运算区语法)**

**BinomialCoefficient(<Number>, <Number>); 二项式系数(<数值 n>, <数值 r>).**

计算“n 选 r”的二项式系数。第一个数值代表所有元素 n, 第二个数值表示所选元素 r。如果键入了未定义的变量替代数值会得出二项式格式公式。

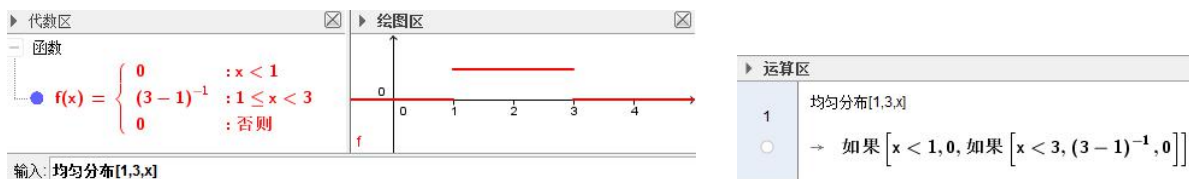
**案例:** “二项式系数(n, 3)” 得出  $\frac{n^3 - 3n^2 + 2n}{6}$ 。

**注:** 参见“组合数”指令。

### 2.6.11 Uniform. 均匀分布

**Uniform(<Lower Bound min>, <Upper Bound max>, <Variable Value>); 均匀分布(<下界>, <上界>, <变量值>).**

返回区间为(下界值, 上界值)的均匀分布密度概率函数。



**Uniform(<Lower Bound min>, <Upper Bound max>, <Variable Value v>); 均匀分布(<下界>, <**

上界>, <变量值>。

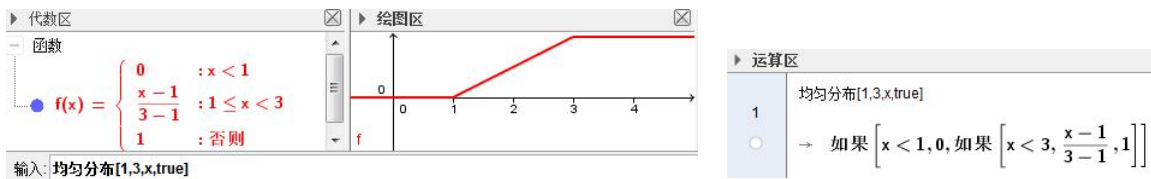
返回区间为(下界值, 上界值) 变量值  $v$  处的均匀分布累积分布函数值 ( $P(X \leq v)$ )。



Uniform(<Lower Bound min>, <Upper Bound max>,  $x$ , <Boolean Cumulative b>); 均匀分布(<下界>, <上界>,  $x$ , <是否累积?true|false>).

当 “b=false”, 返回区间为(下界值, 上界值)的均匀分布密度概率函数。

当 “b=true”, 返回区间为(下界值, 上界值)的累积均匀分布密度概率函数。



### CAS Syntax (运算区语法)

同于指令栏语法, 结果见以上图。

## 2.6.12 RandomUniform. 均匀分布随机数

RandomUniform(<Min>, <Max>); 均匀分布随机数(<最小值>, <最大值>).

返回在区间(最小值, 最大值)上的均匀分布所生成的随机实数。

案例: “均匀分布随机数(0, 1)” 返回在 0 与 1 之间的一个随机数。



RandomUniform(<Min>, <Max>, <Number of Samples n>); 均匀分布随机数(<最小值>, <最大值>, <样本数量>).

返回在区间(最小值, 最大值)上的均匀分布所生成的  $n$  个随机实数列表。

案例: “均匀分布随机数(0, 1, 3)” 返回在区间(0, 1)上的均匀分布所生成的含有 3 个随机实数的列表。

### CAS Syntax (运算区语法)

RandomUniform(<Min>, <Max>); 均匀分布随机数(<最小值>, <最大值>).

返回在区间(最小值, 最大值)上的均匀分布所生成的随机实数。

案例: “均匀分布随机数(0, 1)” 返回在 0 与 1 之间的一个随机数。

注: “均匀分布随机数(0, 1)” 等同于函数 random() (参见已定义函数和操作)。

注: 参见“设置种子”指令、“随机元素”指令、“区间随机数”指令、“随机二项分布数”指令、“正态分布随机数”指令、“泊松分布随机数”指令。

## 2.6.13 ChiSquared. 卡方分布

ChiSquared(<Degrees of Freedom>, <Variable Value>); 卡方分布(<自由度>, <变量值>).

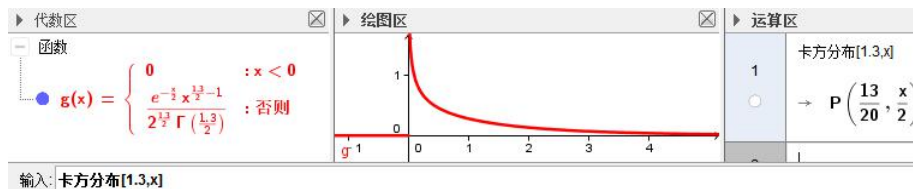
计算卡方分布的变量值  $v$  处的累积分布函数, 自变量  $x$  为适当自由度的卡方分布概率  $P(X \leq v)$ 。

注：返回给定 x 坐标的概率(或给定 x 坐标左侧的卡方分布曲线下面积)。



ChiSquared(<Degrees of Freedom>, x); 卡方分布(<自由度>, x)。

创建适当自由度的卡方密度概率函数(pdf)。



ChiSquared(<Degrees of Freedom>, <Variable Value>, <Boolean Cumulative>); 卡方分布(<自由度>, <变量值>, <是否累积?true|false>)。

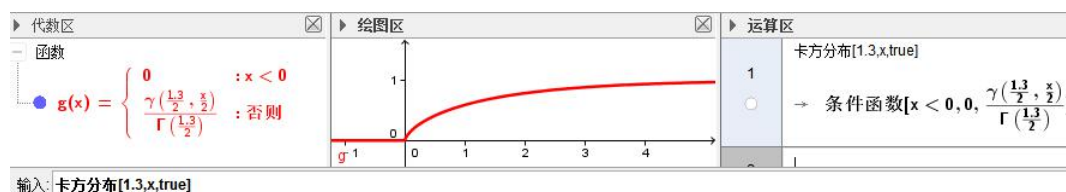
计算卡方分布的变量值 v 处的累积分布函数, 自变量 x 为适当自由度的卡方分布概率 P(X ≤ v)。

注：返回给定 x 坐标的概率(或给定 x 坐标左侧的卡方分布曲线下面积)。



ChiSquared(<Degrees of Freedom>, x, <Boolean Cumulative>); 卡方分布(<自由度>, x, <是否累积?true|false>)。

如果逻辑值是“true”，创建卡方分布累积分布函数，否则创建卡方分布函数。



CAS Syntax (运算区语法)

ChiSquared(<Degrees of Freedom>, <Variable Value>); 卡方分布(<自由度>, <变量值>)。

计算卡方分布的变量值 v 处的累积分布函数, 自变量 x 为适当自由度的卡方分布概率 P(X ≤ v)。

案例：“卡方分布(4, 3)”得出“ $p(2, \frac{3}{2})$ ”，相当于 0.44。

释义：若 n 个相互独立的随机变量  $\xi_1$ 、 $\xi_2$ 、……、 $\xi_n$ ，均服从标准正态分布（也称独立同分布于标准正态分布），则这 n 个服从标准正态分布的随机变量的平方和构成一新的随机变量，其分布规律称为卡方分布（chi-squaredistribution），其中参数 n 称为自由度，正如正态分布中均值或方差不同就是另一个正态分布一样，自由度不同就是另一个卡方分布。卡方分布是由正态分布构造而成的一个新的分布，当自由度 n 很大时，分布近似为正态分布。

## 2.6.14 Cauchy. 柯西分布

Cauchy(<Median m>, <Scale s>, <Variable Value>); 柯西分布(<中位数>, <尺度参数 λ>, <变量值>)。

计算柯西分布的变量值 v 处的累积分布函数。给定中位数和尺度参数自变量为 x 的柯西分布概



率  $P(X \leq v)$ 。

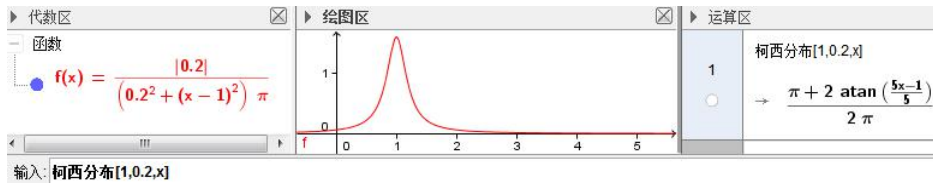
注：返回给定  $x$  坐标的概率(或给定  $x$  坐标左侧的柯西分布曲线下面积)。

Cauchy(<Median  $m$ >, <Scale  $s$ >, <Variable Value>, <Boolean Cumulative>); 柯西分布(<中位数>, <尺度参数  $\lambda$ >, <变量值>, <是否累积?true|false>)。

如果累积是“true”，创建变量值对应的累积柯西函数分布。否则，创建柯西分布的密度概率函数。

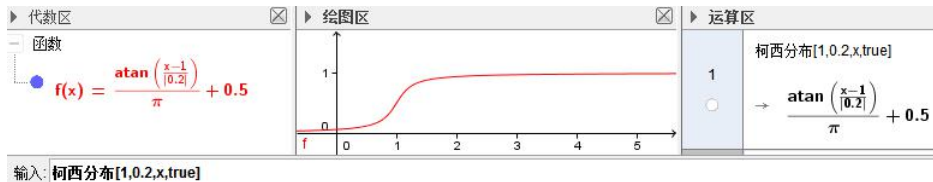
Cauchy(<Median  $m$ >, <Scale  $s$ >,  $x$ ); 柯西分布(<中位数>, <尺度参数  $\lambda$ >,  $x$ )。

创建柯西分布的密度概率函数 (PDF)。



Cauchy(<Median  $m$ >, <Scale  $s$ >,  $x$ , <Boolean Cumulative>); 柯西分布(<中位数>, <尺度参数  $\lambda$ >,  $x$ , <是否累积?true|false>)。

如果累积是“true”，创建累积柯西函数分布。否则，创建柯西分布的密度概率函数。

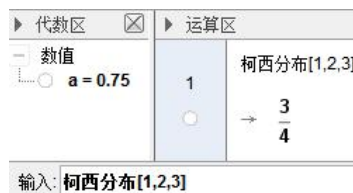


CAS Syntax (运算区语法)

Cauchy(<Median  $m$ >, <Scale  $s$ >, <Variable Value>); 柯西分布(<中位数>, <尺度参数  $\lambda$ >, <变量值>)。

计算柯西分布的变量值  $v$  处的累积分布函数，自变量  $x$  为给定中位数和尺度参数的柯西分布概率  $P(X \leq v)$ 。

案例：“柯西分布(1, 2, 3)” 得出  $\frac{3}{4}$ 。

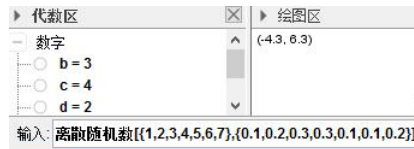


释义：柯西分布是一个数学期望不存在的连续型分布函数，它同样具有自己的分布密度，满足分布函数  $F(X) = 1/2 + 1/\pi * \arctan x$ ,  $-\infty < x < +\infty$ , 密度函数  $\phi(x) = 1/(\pi(1+x^2))$ ,  $-\infty < x < +\infty$  的称为标准柯西分布。重要特性之一就是期望和方差均不存在。

## 2.6.15 RandomDiscrete. 离散随机数

RandomDiscrete(<List of Numbers>, <List of (Relative) Probabilities>); 离散随机数(<数字列表>, <(相对)概率列表>)。

根据第二个概率分布表从第一个列表中产生随机数。第二个列表中的概率值和不必为 1，概率归一化处理。



## 2.6.16 Logistic. 逻辑分布

Logistic(<Mean>, <Scale>, <Variable Value>); 逻辑斯特分布 (<平均数  $\mu$ >, <尺度参数  $\lambda$ >, <变量值> )。

计算逻辑斯特分布的变量值  $v$  处的累积分布函数值, 如当自变量  $x$  为平均数  $\mu$  和尺度参数  $\lambda$  的逻辑斯特分布的概率  $P(X \leq v)$ 。

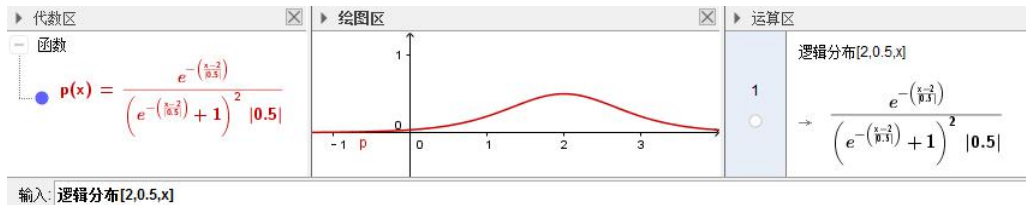


注: 返回给定  $x$  坐标的概率值 (或给定  $x$  坐标左侧的逻辑斯特分布曲线下面积)。

Logistic(<Mean>, <Scale>, <Variable Value>, <Boolean Cumulative>); 逻辑斯特分布 (<平均数  $\mu$ >, <尺度参数  $\lambda$ >, <变量值>, <是否累积? true|false>)。

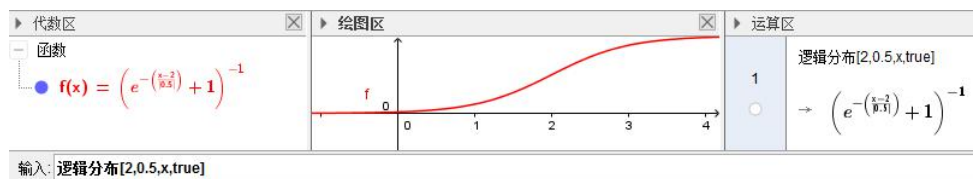
Logistic(<Mean>, <Scale>,  $x$  ); 逻辑分布 (<平均数  $\mu$ >, <尺度参数  $\lambda$ >,  $x$ )。

创建给定参数平均数  $\mu$  和尺度参数  $\lambda$  的逻辑斯特密度概率函数。



Logistic(<Mean>, <Scale>,  $x$ , <Boolean Cumulative>); 逻辑斯特分布 (<平均数  $\mu$ >, <尺度参数  $\lambda$ >,  $x$ , <是否累积? true|false>)。

如果累积是 “true”, 创建逻辑斯特累积概率函数, 否则创建逻辑斯特分布概率函数。



CAS Syntax (运算区语法)

Logistic(<Mean>, <Scale>,  $x$  ); 逻辑斯特分布 (<平均数  $\mu$ >, <尺度参数  $\lambda$ >,  $x$ )。

创建给定参数平均数  $\mu$  和尺度参数  $\lambda$  的逻辑斯特密度概率函数。

Logistic(<Mean>, <Scale>,  $x$ , <Boolean Cumulative>); 逻辑斯特分布 (<平均数  $\mu$ >, <尺度参数  $\lambda$ >,  $x$ , <是否累积? true|false>)。

如果累积是 true, 创建逻辑斯特累积概率函数, 否则创建逻辑斯特分布概率函数。

Logistic(<Mean>, <Scale>, <Variable Value>); 逻辑斯特分布 (<平均数  $\mu$ >, <尺度参数  $\lambda$ >, <变量值>)。

计算逻辑斯特分布的变量值  $v$  处的累积分布函数值, 如当自变量  $x$  为平均数  $\mu$  和尺度参数  $\lambda$  的逻辑斯特分布的概率  $P(X \leq v)$ 。

注: 返回给定  $x$  坐标的概率值 (或给定  $x$  坐标左侧的逻辑斯特分布曲线下面积)。

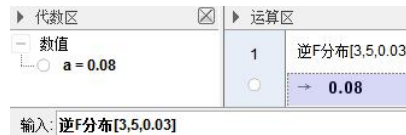
## 2.6.17 InverseFDistribution. 逆 F 分布

InverseFDistribution(<Numerator Degrees of Freedom>, <Denominator Degrees of Freedom>, <Probability>); 逆 F 分布(<分子自由度>, <分母自由度>, <概率>)。

当 F 分布  $p$  由自由度给定时, 计算概率  $p$  的 F 分布积累分布函数的逆值。

换种说法, 当  $X$  是 F 分布随机变量时, 找到能使  $P(X \leq t) = p$  的  $t$ 。

概率  $p$  必须在  $(0, 1)$  内。



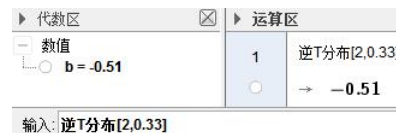
## 2.6.18 InverseTDistribution. 逆 T 分布

InverseTDistribution(<Degrees of Freedom>, <Probability>); 逆 T 分布(<自由度>, <概率>)。

当 T 分布  $p$  由自由度  $d$  给定时, 计算概率  $p$  的 T 分布积累分布函数的逆值。

换种说法, 当  $X$  是 T 分布随机变量时, 找到能使  $P(X \leq r) = p$  的  $r$ 。

概率  $p$  必须在  $(0, 1)$  内。



## 2.6.19 InversePoisson. 逆泊松分布

InversePoisson(<Mean>, <Probability>); 逆泊松分布(<平均数>, <概率>)。

当  $p$  是概率且  $X$  是给定平均数  $\lambda$  的泊松分布随机变量, 返回  $P(X \leq n) \geq p$  的最小整数  $n$ 。

注: 参见“泊松分布”指令。



## 2.6.20 InverseHyperGeometric. 逆超几何分布

InverseHyperGeometric(<Population Size>, <Number of Successes>, <Sample Size>, <Probability>); 逆超几何分布(<总体容量>, <成功次数>, <样本容量>, <概率>)。

当  $p$  是概率且  $X$  是给定的总体容量、成功次数和样本容量的超几何分布随机变量, 返回  $P(X \leq n) \geq p$  的最小整数  $n$ 。

注: 参见“超几何分布”指令。



## 2.6.21 InverseLogNormal. 逆对数正态分布

$\text{InverseLogNormal}(\langle \text{Mean} \rangle, \langle \text{Standard Deviation} \rangle, \langle \text{Probability} \rangle)$ ; 逆对数正态分布(⟨平均数⟩, ⟨标准差⟩, ⟨概率⟩)。

当给定均值  $\mu$  和标准偏差  $\sigma$  时, 计算概率  $p$  的对数正态分布函数的逆值。

也就是说, 当  $X$  是对数正态分布分布随机变量时, 找到能使  $P(X \leq r) = p$  的  $t$ 。

概率  $p$  必须在  $(0, 1)$  内。



## 2.6.22 InverseBinomial. 逆二项分布

$\text{InverseBinomial}(\langle \text{Number of Trials} \rangle, \langle \text{Probability of Success} \rangle, \langle \text{Probability} \rangle)$ ; 逆二项分布(⟨试验次数⟩, ⟨成功概率⟩, ⟨概率⟩)。

当  $p$  是概率且  $X$  是给定的试验次数和成功概率的逆二项分布随机变量, 返回  $P(X \leq n) \geq p$  的最小整数  $n$ 。

注: 参见“二项式分布”指令。



## 2.6.23 InverseChiSquared. 逆卡方分布

$\text{InverseChiSquared}(\langle \text{Degrees of Freedom} \rangle, \langle \text{Probability} \rangle)$ ; 逆卡方分布(⟨自由度⟩, ⟨概率⟩)。

当卡方分布  $p$  由自由度  $d$  给定时, 计算概率  $p$  的卡方分布积累分布函数的逆值。

换种说法, 当  $X$  是卡方随机变量时, 找到能使  $P(X \leq t) = p$  的  $t$ 。

概率  $p$  必须在  $(0, 1)$  内。



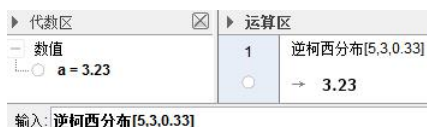
## 2.6.24 InverseCauchy. 逆柯西分布

$\text{InverseCauchy}(\langle \text{Median} \rangle, \langle \text{Scale} \rangle, \langle \text{Probability} \rangle)$ ; 逆柯西分布(⟨中位数⟩, ⟨尺度参数  $\lambda$ ⟩, ⟨概率⟩)。

当柯西分布  $p$  由中位数  $m$  和尺度参数  $\lambda$  给定时, 计算概率  $p$  的柯西分布积累分布函数的逆值。

换种说法, 当  $X$  是柯西随机变量时, 找到能使  $P(X \leq t) = p$  的  $t$ 。

概率  $p$  必须在  $(0, 1)$  内。



## 2.6.25 InverseLogistic. 逆逻辑分布

$\text{InverseLogistic}(\langle \text{Mean} \rangle, \langle \text{Scale} \rangle, \langle \text{Probability} \rangle)$ ; 逆逻辑分布( $\langle \text{平均数} \rangle, \langle \text{尺度参数} \lambda \rangle, \langle \text{概率} \rangle$ )。

当逻辑分布  $p$  由平均数  $\mu$  和尺度参数  $\lambda$  给定时, 计算概率  $p$  的逻辑分布积累分布函数的逆值。

换种说法, 当  $X$  是逻辑随机变量时, 找到能使  $P(X \leq t) = p$  的  $t$ 。

概率  $p$  必须在  $(0, 1)$  内。



## 2.6.26 InversePascal. 逆帕斯卡分布

$\text{InversePascal}(\langle \text{Number of Successes} \rangle, \langle \text{Probability of Success} \rangle, \langle \text{Probability} \rangle)$ ; 逆帕斯卡分布( $\langle \text{成功次数} \rangle, \langle \text{成功概率} \rangle, \langle \text{概率} \rangle$ )。

当  $p$  是概率且  $X$  是给定成功次数和成功概率的帕斯卡分布随机变量, 返回  $P(X \leq n) \geq p$  的最小整数  $n$ 。

注: 参见“帕斯卡”指令。



## 2.6.27 InverseZipf. 逆齐普夫分布

$\text{InverseZipf}(\langle \text{Number of Elements} \rangle, \langle \text{Exponent} \rangle, \langle \text{Probability} \rangle)$ ; 逆齐普夫分布( $\langle \text{元素数量} \rangle, \langle \text{指数} \rangle, \langle \text{概率} \rangle$ )。

当  $p$  是概率且  $X$  是给定元素数量和指数的齐普夫分布随机变量, 返回  $P(X \leq n) \geq p$  的最小整数  $n$ 。

注: 参见“齐普夫分布”指令。



## 2.6.28 InverseWeibull. 逆威布尔分布

$\text{InverseWeibull}(\langle \text{Shape} \rangle, \langle \text{Scale} \rangle, \langle \text{Probability} \rangle)$ ; 逆威布尔分布( $\langle \text{形状参数} k \rangle, \langle \text{尺度参数} \lambda \rangle, \langle \text{概率} \rangle$ )。

当威布尔分布  $p$  由形状参数  $k$  和尺度参数  $\lambda$  给定时, 计算概率  $p$  的威布尔分布积累分布函数的逆值。

换种说法, 当  $X$  是威布尔分布随机变量时, 找到能使  $P(X \leq t) = p$  的  $t$ 。

概率  $p$  必须在  $(0, 1)$  内。



## 2. 6. 29 InverseNormal. 逆正态分布

InverseNormal(<Mean>, <Standard Deviation>, <Probability>); 逆正态分布(<平均数>, <标准差>, <概率>).

当  $\Phi^{-1}$  是累积分布函数  $\Phi$  在  $N(0, 1)$  内的逆分布时, 计算给定概率  $p$ 、平均数  $\mu$  和标准差  $\sigma$  的函数  $\Phi^{-1}(p) \cdot \sigma + \mu$ 。

注: 返回正态分布曲线下方给定概率左侧的  $x$  坐标部分。



## 2. 6. 30 InverseExponential. 逆指数分布

当指数分布  $p$  由率参数  $\lambda$  给定时, 计算概率  $p$  的指数分布积累分布函数的逆值。

换种说法, 当  $X$  是指数随机变量时, 找到能使  $P(X \leq t) = p$  的  $t$ 。

概率  $p$  必须在  $(0, 1)$  内。



## 2. 6. 31 InverseGamma. 逆伽玛分布

InverseGamma(<Alpha>, <Beta>, <Probability>); 逆伽玛分布(<形状参数  $\alpha$ >, <尺度参数  $\beta$ >, <概率>).

当伽玛分布  $p$  由参数  $\alpha$  和参数  $\beta$  给定时, 计算概率  $p$  的伽玛分布积累分布函数的逆值。

换种说法, 当  $X$  是伽玛随机变量时, 找到能使  $P(X \leq t) = p$  的  $t$ 。

概率  $p$  必须在  $(0, 1)$  内。



## 2. 6. 32 Pascal. 帕斯卡分布

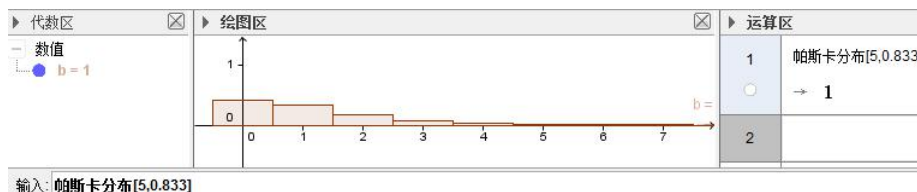
Pascal(<Number of Successes>, <Probability of Success>); 帕斯卡分布(<成功次数>, <成功概率>).

返回帕斯卡分布的一个条形图。

参数:

成功次数：单独伯努利试验成功的次数。

成功概率：一次试验成功的概率。

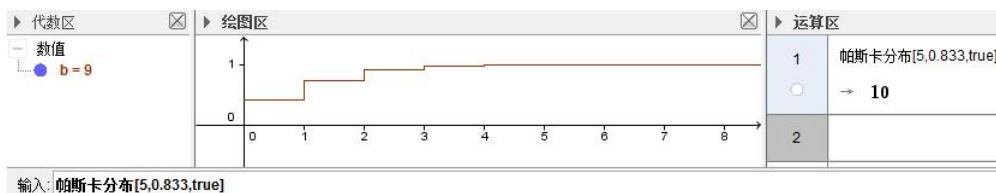


Pascal(<Number of Successes>,<Probability of Success>,<Boolean Cumulative>); 帕斯卡分布(<成功次数>,<成功概率>,<是否累积?true|false>)。

当累积是“false”，返回一个帕斯卡分布条形图。

当累积是“true”，返回一个累积帕斯卡分布条形图。

前两个参数同上。



Pascal(<Number of Successes>,<Probability of Success>,<Variable Value>,<Boolean Cumulative>); 帕斯卡分布(<成功次数>,<成功概率>,<变量值>,<是否累积? true|false>)。

设  $X$  是帕斯卡随机变量且  $v$  是变量值。

当累积是“false”，返回  $P(X=v)$ 。

当累积是“true”，返回  $P(X \leq v)$ 。

前两个参数同上。



CAS Syntax (运算区语法)

Pascal(<Number of Successes>,<Probability of Success>,<Variable Value>,<Boolean Cumulative>); 帕斯卡分布(<成功次数>,<成功概率>,<变量值>,<是否累积? true|false>)。

设  $X$  是帕斯卡随机变量且  $v$  是变量值。

当累积是“false”，返回  $P(X=v)$ 。

当累积是“true”，返回  $P(X \leq v)$ 。

案例：单独伯努利试验成功的次数是  $n=1$ ，一次试验成功概率是  $p=\frac{1}{6}$ 。变量值  $v=2$  且“累积”为“false”。

“帕斯卡分布( $n, p, v, \text{false}$ )”得出  $\frac{25}{216}$ 。

## 2.6.33 Zipf. 齐普夫分布

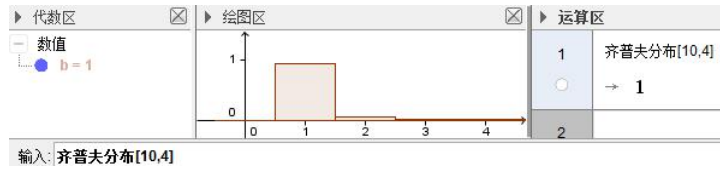
Zipf(<Number of Elements>,<Exponent>); 齐普夫分布(<元素数量>,<指数>)。

返回齐普夫条形图。

参数：

元素数量：研究元素的数量。

指数：分布的指数特性。

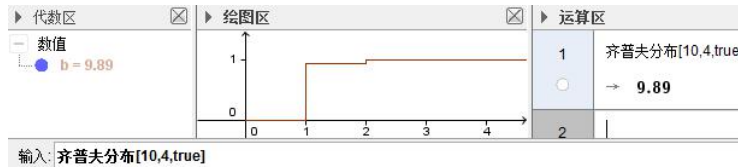


Zipf(<Number of Elements>, <Exponent>, <Boolean Cumulative>); 齐普夫分布(<元素数量>, <指数>, <是否累积?true|false>).

如果累积是“false”，创建齐普夫分布密度概率条形图。

如果累积是“true”，创建累积齐普夫分布密度概率条形图。

前两个参数同上。



Zipf(<Number of Elements>, <Exponent>, <Variable Value v>, <Boolean Cumulative>); 齐普夫分布(<元素数量>, <指数>, <变量值>, <是否累积?true|false>).

设 X 是齐普夫随机变量。

当累积是“false”时，返回  $P(X=v)$ 。

当累积是“true”时，返回  $P(X \leq v)$ 。

前两个参数同上。



CAS Syntax (运算区语法)

Zipf(<Number of Elements>, <Exponent>, <Variable Value v>, <Boolean Cumulative>); 齐普夫分布(<元素数量>, <指数>, <变量值>, <是否累积?true|false>).

设 X 是齐普夫随机变量。

当累积“false”时，返回  $P(X=v)$ 。

当累积“true”时，返回  $P(X \leq v)$ 。

案例：“齐普夫分布(10, 1, 5, false)”得出  $\frac{504}{7381}$ 。

## 2.6.34 RandomBetween. 区间随机数

RandomBetween(<Minimum Integer>, <Maximum Integer>); 区间随机数(<最小整数>, <最大整数>).

生成一个在最小值与最大值之间的整数(含端点)。

案例：“区间随机数(0, 10)”得出在 0 与 10 之间的一个整数(含端点)。



RandomBetween(<Minimum Integer>, <Maximum Integer>, <Boolean Fixed>); 区间随机数(<最小整数>, <最大整数>, <是否固定?true|false>).



如果布尔值<锁定>="true", 随机产生一个在最小值与最大值之间的整数(含端点)只会更新一次(在文件被载入之后的更新算起, 即使进行了撤销/重做操作也同样视为已更新处理)。

案例: “区间随机数(0, 10, true)” 得出一个在 0 和 10 之间的数(含端点)。



注: 按“F9”键观察两个语法间的不同。从截图看, 两个结果没有区别, 都是得到了规定区间内的随机数。但按“F9”键刷新后, 会发现第一个语法指令的随机数会随着刷新而更新随机数。

### CAS Syntax (运算区语法)

RandomBetween(<Minimum Integer>, <Maximum Integer>); 区间随机数(<最小整数>, <最大整数>)。

生成一个在最小值与最大值之间的整数(含端点)。

案例: “区间随机数(0, 10)” 得出一个在 0 和 10 之间的数(含端点)。

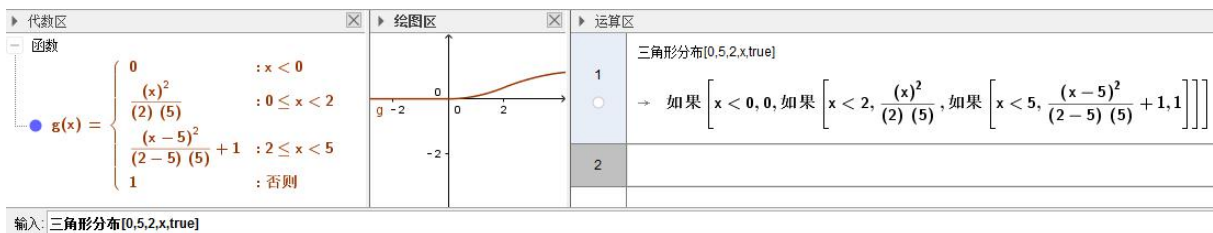
注: 参见“设置种子”指令、“随机元素”指令、“随机二项分布数”指令、“正态分布随机数”指令、“泊松分布随机数”指令、“均匀分布随机数”指令。

## 2.6.35 Triangular. 三角形分布

Triangular(<Lower Bound>, <Upper Bound>, <Mode>, <Variable Value>); 三角形分布(<下界>, <上界>, <模式>, <变量值>)。

计算三角形分布的变量值  $v$  处的累积分布函数值, 如给定参数最小、最大和模式的自变量为  $x$  的三角形分布的概率  $P(X \leq v)$ 。

注: 返回给定  $x$  坐标的概率值(或给定  $x$  坐标左侧的三角形分布曲线下面积)。

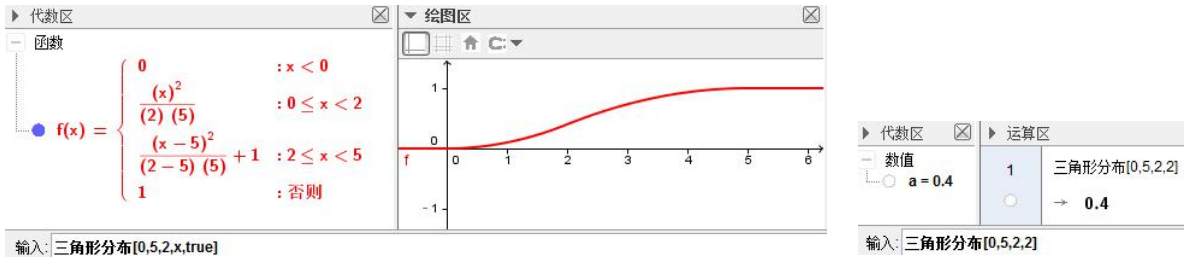


Triangular(<Lower Bound>, <Upper Bound>, <Mode>, <Variable Value>, <Boolean Cumulative e>); 三角形分布(<下界>, <上界>, <模式>, <变量值>, <是否累积?true|false>)。



Triangular(<Lower Bound>, <Upper Bound>, <Mode>,  $x$ , <Boolean Cumulative>); 三角形分布(<下界>, <上界>, <模式>,  $x$ , <是否累积?true|false>)。

如果累积是“true”, 创建累积三角形分布函数, 否则创建三角形分布密度概率函数。



**CAS Syntax (运算区语法)**

**Triangular(<Lower Bound>, <Upper Bound>, <Mode>, <Variable Value>);** 三角形分布(<下界>, <上界>, <模式>, <变量值>).

计算三角形分布的变量值  $v$  处的累积分布函数值, 如给定参数最小、最大和模式的自变量为  $x$  的三角形分布的概率  $P(X \leq v)$ 。

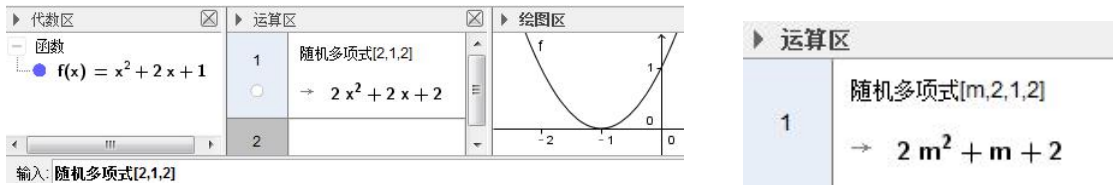
**案例:** “三角形分布(0, 5, 2, 2)” 得出 0.4。

**2.6.36 RandomPolynomial. 随机多项式**

**随机多项式(<次数>, <最小系数>, <最大系数>)**

按照给定的相关数值随机产生一个多项式。

**案例:** “随机多项式(0, 1, 2)” 得出 “1” 或 “2”; “随机多项式(2, 1, 2)” 得出 “ $2x^2+2x+2$ ” 或 “ $x^2+2x+2$ ” 或 “...”。



**CAS Syntax (运算区语法)**

**随机多项式(<次数>, <最小系数>, <最大系数>)**

按照给定的相关数值随机产生一个多项式。

**随机多项式(<变量>, <次数>, <最小系数>, <最大系数>)**

按照给定相关数值随机产生一个指定变量的多项式。

**案例:** “随机多项式(a, 0, 1, 2)” 得出 “1” 或 “2”; “随机多项式(a, 2, 1, 2)” 得出系数是 1 或者 2 的多项式, 如 “ $2a^2+2a+2$ ” 或 “ $a^2+2a+2$ ” 或 “...”。

**注:** 按 “F9” 键后, 会更新多项式。如果最小值和最大值不是整数, 会自动四舍五入取整。

**2.6.37 RandomBinomial. 随机二项分布数**

**RandomBinomial(<Number of Trials>, <Probability>);** 随机二项分布数(<试验次数>, <概率>).

由一个试验次数为  $n$  和概率为  $p$  的二项分布生成一个随机数。



**案例：**“随机二项分布数(3, 0.1)”给出“ $j \in \{0, 1, 2, 3\}$ ”（随机为列表中任意一个数），一个事件在三次试验中出现  $j$  次的概率为 0.1，其得出随机数  $j$  的概率。

**CAS Syntax (运算区语法)**

**RandomBinomial(<Number of Trials>, <Probability>); 随机二项分布数(<试验次数>, <概率>)**。

由一个试验次数为  $n$  和概率为  $p$  的二项分布生成一个随机数。

**案例：**“随机二项分布数(3, 0.1)”给出  $j \in \{0, 1, 2, 3\}$ ，一个事件在三次试验中出现  $j$  次的概率为 0.1 其得出随机数  $j$  的概率。

**注：**参见“设置种子”指令、“随机元素”指令、“区间随机数”指令、“正态分布随机数”指令、“泊松分布随机数”指令、“均匀分布随机数”指令。

按“F9”键后，会更新结果随机数。

## 2.6.38 RandomPointIn. 随机内点

**RandomPointIn(<Region>); 随机内点(<多边形|完全二次曲线>)**。

在给定的多边形或圆锥曲线内创建随机点。

**RandomPointIn(<List of Points>); 随机内点(<点 1>, <点 2>, <点 3>, <...>)**。

返回给定点确定的多边形内部的随机点。

**案例：**“随机内点(多边形(A, B, C))”和“随机内点(A, B, C)”都是返回三角形 ABC 内的随机点。想要得到点列中的随机点，需要使用“随机元素”指令。

**RandomPointIn(<xMin>, <xMax>, <yMin>, <yMax>); 随机内点(<x 最小值>, <x 最大值>, <y 最小值>, <y 最大值>)**。

创建横坐标在区间( $x$  最小,  $x$  最大), 纵坐标在区间( $y$  最小, 有最大)的随机点。

按“F9”键后，会更新结果随机点。

## 2.6.39 Weibull. 威布尔分布

**Weibull(<Shape>, <Scale>, <Variable Value>); 威布尔分布(<形状参数  $k$ >, <尺度参数  $\lambda$ >, <变量值>)**。

计算威布尔分布的变量值  $v$  处的累积分布函数值，如给定的形状参数  $k$  和尺度参数  $\lambda$  的自变量为  $x$  的威布尔分布的概率  $P(X \leq v)$ 。

**注：**返回给定  $x$  坐标的概率值(或给定  $x$  坐标左侧的威布尔分布曲线下面积)。

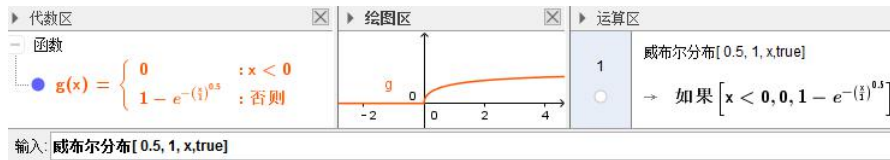


**Weibull(<Shape>, <Scale>, <Variable Value>, <Boolean Cumulative>); 威布尔分布(<形状参数  $k$ >, <尺度参数  $\lambda$ >, <变量值>, <是否累积?true|false>)**。

如果累积是“true”，创建累积威布尔分布函数，否则创建威布尔分布密度概率函数。

**Weibull(<Shape>, <Scale>,  $x$ , <Boolean Cumulative>); 威布尔分布(<形状参数  $k$ >, <尺度参数  $\lambda$ >,  $x$ , <是否累积?true|false>)**。

如果累积是“true”，创建  $x$  主变量累积威布尔分布函数，否则创建威布尔分布密度概率函数。



### CAS Syntax (运算区语法)

**Weibull**(**<Shape>**, **<Scale>**, **<Variable Value>**); 威布尔分布(**<形状参数 k>**, **<尺度参数 λ>**, **<变量值>**)。

计算威布尔分布的变量值  $v$  处的累积分布函数值，如给定的形状参数  $k$  和尺度参数  $\lambda$  的自变量为  $x$  的威布尔分布的概率  $P(X \leq v)$ 。

**案例：**“威布尔分布(0.5, 1, 0)” 得出 0；“威布尔分布(0.5, 1, 1)” 得出  $1 - \frac{1}{e}$ 。

## 2.6.40 Normal. 正态分布

**Normal**(**<Mean μ>**, **<Standard Deviation σ>**, **<Variable Value v>**); 正态分布(**<平均数>**, **<标准差>**, **<变量值>**)。

当  $\Phi$  是给定参数平均数  $\mu$  和标准差  $\sigma$  的积累分布函数，计算给定变量  $v$  的  $\Phi\left(\frac{x-\mu}{\sigma}\right)$ 。

**注：**返回给定  $x$  坐标的概率值(或给定  $x$  坐标左侧的正态分布曲线下面积)。

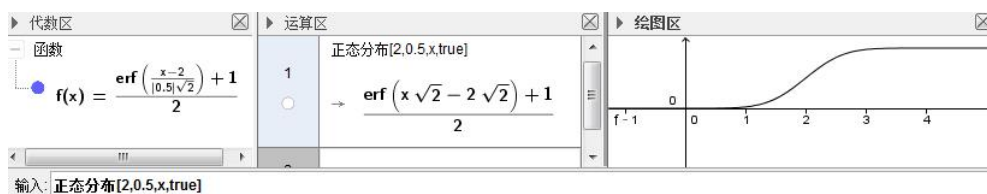


**Normal**(**<Mean μ>**, **<Standard Deviation σ>**, **<Variable Value v>**, **<Boolean Cumulative e>**); 正态分布(**<平均数>**, **<标准差>**, **<变量值>**, **<是否累积?true|false>**)。

如果累积是“true”，创建给定参数平均数  $\mu$  和标准差  $\sigma$  的正态分布累积概率函数，否则创建正态分布概率函数。

**Normal**(**<Mean>**, **<Standard Deviation>**, **x**, **<Boolean Cumulative>**); 正态分布(**<平均数>**, **<标准差>**, **x**, **<是否累积?true|false>**)。

如果累积是“true”，创建主变量  $x$  给定参数平均数  $\mu$  和标准差  $\sigma$  的正态分布累积概率函数，否则创建正态分布概率函数。



### CAS Syntax (运算区语法)

**Normal**(**<Mean>**, **<Standard Deviation>**, **<Variable Value>**); 正态分布(**<平均数>**, **<标准差>**, **<变量值>**)。

当  $\Phi$  是给定参数平均数  $\mu$  和标准差  $\sigma$  的积累分布函数，计算给定变量  $v$  的  $\Phi\left(\frac{x-\mu}{\sigma}\right)$ 。

**案例：**“正态分布(2, 0.5, 1)” 得出  $\frac{\text{erf}(-\sqrt{2})+1}{2}$ 。

## 2.6.41 RandomNormal. 正态分布随机数

**RandomNormal(<Mean>, <Standard Deviation>); 正态分布随机数(<平均数>, <标准差>)**。

由一个指定平均数（均值）与标准差的正态分布生成一个随机数。

**案例：**“正态分布随机数(3, 0.1)”由一个平均数（均值）为3和标准差为0.1的正态分布得出一随机数。



**CAS Syntax (运算区语法)**

**RandomNormal(<Mean>, <Standard Deviation>); 正态分布随机数(<平均数>, <标准差>)**。

由一个指定平均数（均值）与标准差的正态分布生成一个随机数。

**案例：**“正态分布随机数(3, 0.1)”由一个平均数（均值）为3和标准差为0.1的正态分布得出一随机数。

**注：**参见“设置种子”指令、“随机元素”指令、“区间随机数”指令、“随机二项分布数”指令、“泊松分布随机数”指令、“均匀分布随机数”指令。

按“F9”键后，会更新结果随机数。

## 2.6.42 Exponential. 指数分布

**Exponential(<Lambda>, <Variable Value>); 指数分布(<率参数 λ>, <变量值>)**。

如果逻辑值是“true”，创建指数分布累积分布函数，否则创建指数分布函数。

**注：**返回给定 x 坐标的概率(或给定 x 坐标左侧的爱尔朗分布曲线下面积)。

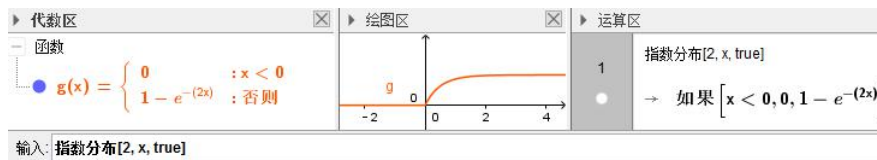


**Exponential(<Lambda>, <Variable Value>, <Boolean Cumulative>); 指数分布(<率参数 λ>, <变量值>, <是否累积?true|false>)**。

计算指数分布的变量值 v 处的累积分布函数，适当自由度的指数分布概率  $P(X \leq v)$ 。

**Exponential(<Lambda>, x, <Boolean Cumulative>); 指数分布(<率参数 λ>, x, <是否累积?true|false>)**。

计算指数分布的变量值 v 处的累积分布函数，自变量 x 为适当自由度的指数分布概率  $P(X \leq v)$ 。



**CAS Syntax (运算区语法)**

**Exponential(<Lambda>, <Variable Value>); 指数分布(<率参数 λ>, <变量值>)**。

计算指数分布的变量值 v 处的累积分布函数，自变量 x 为适当率参数 λ 的指数分布概率  $P(X \leq v)$ 。

**案例：**“指数分布(2, 1)”得出  $1 - \frac{1}{e^2}$ 。相当于 0.86。

## 2.6.43 nPr. 组合数

$nPr(\langle \text{Number } n \rangle, \langle \text{Number } r \rangle)$ ; 组合数( $\langle \text{数值 } n \rangle, \langle \text{数值 } r \rangle$ )。

返回从  $n$  个元素中取出  $r$  个元素的不重复组合数。

案例：“组合数(10, 2)” 得出 90。

CAS Syntax (运算区语法)

$nPr(\langle \text{Number } n \rangle, \langle \text{Number } r \rangle)$ ; 组合数( $\langle \text{数值 } n \rangle, \langle \text{数值 } r \rangle$ )。

返回从  $n$  个元素中取出  $r$  个元素的不重复组合数。

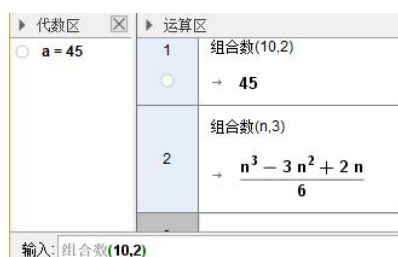
案例：“组合数(10, 2)” 得出 90。

案例：“组合数( $n, 3$ )” 得出  $\frac{n^3 - 3n^2 + 2n}{6}$ 。

注：参见“二项式系数”指令。

$nPr(\langle \text{Number} \rangle, \langle \text{Number} \rangle)$ ; 组合数( $\langle \text{数值 } n \rangle, \langle \text{数值 } r \rangle$ )

案例：“组合数(10, 2)” 得出 90。



## 2. 6. 44 Gamma. 伽玛分布

$\text{Gamma}(\langle \text{Number } \alpha \rangle, \langle \text{Number } \beta \rangle, \langle \text{Variable Value } v \rangle)$ ; 伽玛分布( $\langle \text{形状参数} \rangle, \langle \text{尺度参数} \rangle, \langle \text{变量值} \rangle$ )。

计算伽玛分布的变量值  $v$  处的累积分布函数, 自变量  $x$  为参数  $\alpha$  和  $\beta$  的伽玛分布概率  $P(X \leq v)$ 。

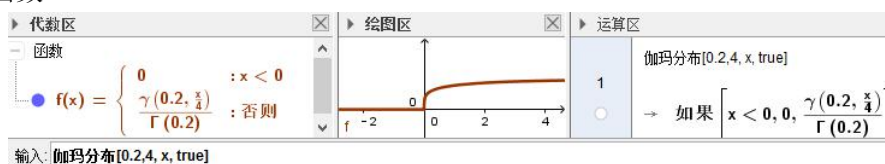


$\text{Gamma}(\langle \text{Number } \alpha \rangle, \langle \text{Number } \beta \rangle, \langle \text{Variable Value } v \rangle, \langle \text{Boolean Cumulative} \rangle)$ ; 伽玛分布( $\langle \text{形状参数} \rangle, \langle \text{尺度参数} \rangle, \langle \text{变量值} \rangle, \langle \text{是否累积? true|false} \rangle$ )。

如果累积是“true”，创建伽玛分布的累积概率函数, 否则创建伽玛分布概率分布函数。

$\text{Gamma}(\langle \text{Number } \alpha \rangle, \langle \text{Number } \beta \rangle, x, \langle \text{Boolean Cumulative} \rangle)$ ; 伽玛分布( $\langle \text{形状参数} \rangle, \langle \text{尺度参数} \rangle, x, \langle \text{是否累积? true|false} \rangle$ )

如果累积是“true”，创建主变量  $x$  的伽玛分布的累积概率函数, 否则创建主变量  $x$  的伽玛分布概率分布函数。



CAS Syntax (运算区语法)

$\text{Gamma}(\langle \text{Number } \alpha \rangle, \langle \text{Number } \beta \rangle, \langle \text{Variable Value } v \rangle)$ ; 伽玛分布( $\langle \text{形状参数} \rangle, \langle \text{尺度参数} \rangle, \langle \text{变量值} \rangle$ )。

计算伽玛分布的变量值  $v$  处的累积分布函数, 自变量  $x$  为参数  $\alpha$  和  $\beta$  的伽玛分布概率  $P(X \leq v)$ 。

注：返回给定  $x$  坐标的概率(或给定  $x$  坐标左侧的伽玛分布曲线下面积)。

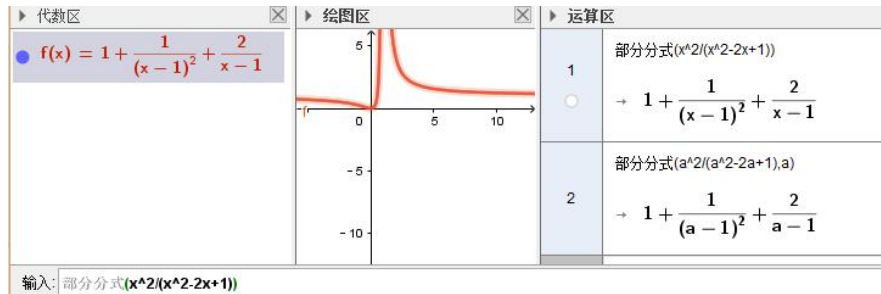
## 2.7 Function. 函数与微积分

### 2.7.1 PartialFractions. 部分分式

PartialFractions(<Function>); 部分分式(<函数>)。

得出指定函数的主函数变量尽量多部分分式。在绘图区给出函数的图象。

案例：“部分分式( $x^2/(x^2-2x+1)$ )”得出  $1 + \frac{1}{(1+x)^2} + \frac{2}{x-1}$ ”。



CAS Syntax (运算区语法)

PartialFractions(<Function>); 部分分式(<函数>)。

得出指定函数的主函数变量尽量多部分分式。

案例：“部分分式( $x^2/(x^2-2x+1)$ )”得出  $1 + \frac{1}{(1+x)^2} + \frac{2}{x-1}$ ”。

PartialFractions(<Function>, <Variable>); 部分分式(<函数>, <变量>)。

得出指定函数的指定函数变量尽量多部分分式。

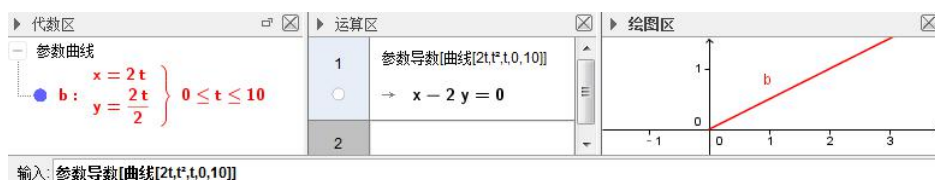
案例：“部分分式( $a^2/(a^2-2a+1), a$ )”得出  $1 + \frac{1}{(1+a)^2} + \frac{2}{a-1}$ ”。

### 2.7.2 ParametricDerivative. 参数导数

Derivative(<Curve>); 参数导数(<曲线>)。

返回一条新的参数曲线，指定为： $(x(t), \frac{y'(t)}{x'(t)})$ 。

案例：“参数导数(曲线( $2t, t^2, t, 0, 10$ ))”返回参数曲线“( $x(t)=2t, y(t)=t$ )”。指令中作为参数的曲线是：函数  $f(x)=\frac{x^2}{4}$  求导结果：函数  $f'(x)=\frac{x}{2}$ 。

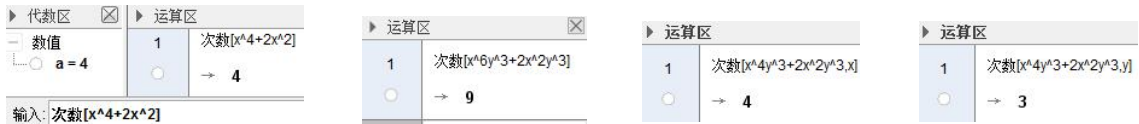


### 2.7.3 Degree. 次数

Degree(<Polynomial>); 次数(<多项式>)。

给出多项式的次方数(主变量)。

案例：“次数( $x^4+2x^2$ )”得出4。



CAS Syntax (运算区语法)

**Degree**(<Polynomial>); 次数(<多项式>)。

给出多项式的次方数(主变量或单项式)。

案例：“次数( $x^4+2x^2$ )”得出4；“次数( $x^6y^3+2x^2y^3$ )”得出9。

**Degree**(<Polynomial>, <Variable>); 次数(<多项式>, <变量>)。

给出多项式的给定变量的次方数。

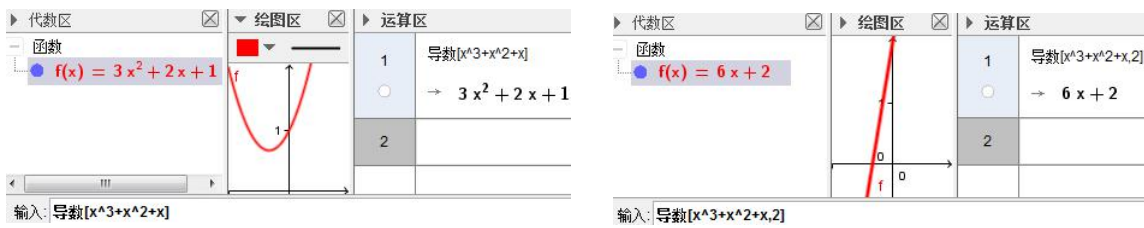
案例：“次数( $x^4y^3+2x^2y^3, x$ )”得出4；“次数( $x^4y^3+2x^2y^3, y$ )”得出3。

## 2.7.4 Derivative. 导数

**Derivative**(<Function>); 导数(<函数>)。

返回函数关于主变量的导数。

案例：“导数( $x^3+x^2+x$ )”得出“ $3x^2+2x+1$ ”。



**Derivative**(<Curve>); 导数(<曲线>)。

返回曲线的导数。

案例：“导数(曲线( $\cos(t), t \sin(t), t, 0, \pi$ ))”得出曲线“ $x=-\sin(t), y=\sin(t)+t\cos(t)$ ”。

注：它只适于参数曲线。



**Derivative**(<Function>, <Number>); 导数(<函数>, <阶数>)。

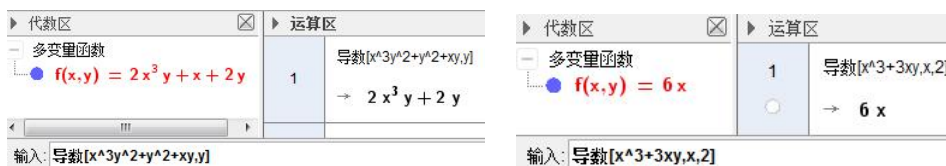
返回函数关于主变量的 n 阶导数。

案例：“导数( $x^3+x^2+x, 2$ )”得出“ $6x+2$ ”。

**Derivative**(<Function>, <Variable>); 导数(<函数>, <变量>)。

返回函数关于指定变量的偏导数。

案例：“导数( $x^3y^2+y^2+xy, y$ )”得出“ $2x^3y+x+2y$ ”。

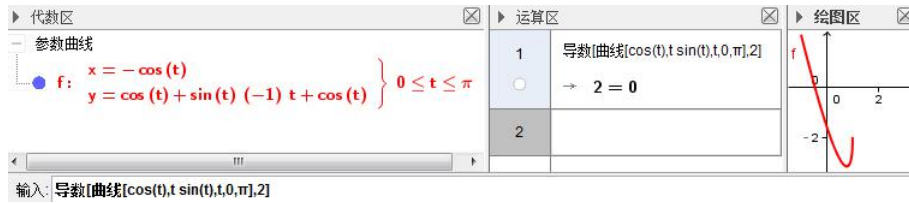


**Derivative**(<Curve>, <Number>); 导数(<曲线>, <阶数>)。

返回曲线的 n 阶导数。

案例：“导数(曲线( $\cos(t), t \sin(t), t, 0, \pi$ ), 2)”得出曲线“ $x=-\cos(t), y=2\cos(t)-t\sin(t)$ ”。





注：它只适于参数曲线。

注：可以使用“f'(x)”替代“导数(f)”，或“f''(x)”替代“导数(f,2)”，以此类推。

Derivative(<Function>, <Variable>, <Number>); 导数(<函数>, <变量>, <阶数>).

返回函数关于指定变量的 n 阶偏导数。

案例：“导数(x<sup>3</sup>+3xy, x, 2)”得出“6x”。

CAS Syntax (运算区语法)

Derivative(<Expression>); 导数(<表达式>).

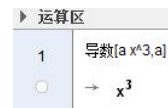
返回表达式关于主变量的导数。

案例：“导数(x<sup>2</sup>)”得出 2x。

Derivative(<Expression>, <Variable>); 导数(<表达式>, <变量>).

返回表达式关于给定主变量的导数。

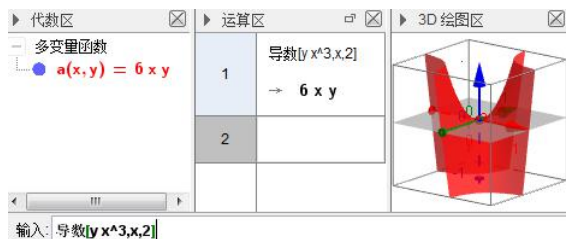
案例：“导数(a x<sup>3</sup>, a)”得出 x<sup>3</sup>。



Derivative(<Expression>, <Variable>, <Number>); 导数(<表达式>, <变量>, <阶数>).

返回表达式关于给定变量的 n 阶导数。

案例：“导数(y x<sup>3</sup>, x, 2)”得出 6xy；“导数(x<sup>3</sup>+3xy, x, 2)”得出 6x。



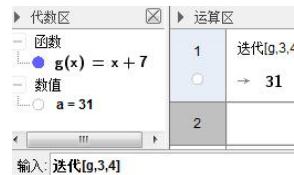
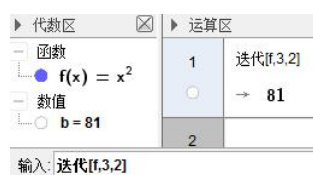
## 2.7.5 Iteration. 迭代

Iteration(<Function>, <Start Value>, <Number of Iterations>); 迭代(<函数>, <起始值>, <迭代次数>).

用指定的起始值将函数迭代 n 次。

案例：定义函数“f(x)=x<sup>2</sup>”后指令“迭代(f, 3, 2)”给出结果(3<sup>2</sup>)<sup>2</sup>=81。

案例：重复叠加。想要得到 3 多次叠加 7：设“g(x)=x+7”，然后“迭代(g, 3, 4)”得出(((3+7)+7)+7)+7=31。



Iteration(<Expression>, <Variable Name>, ..., <Start Values>, <Number of Iterations>); 迭代(<表达式>, <变量>, <起始值>, <次数>).

用指定的起始值将表达式迭代  $n$  次。结果显示同一个参数变量迭代的最后一个元素（终像）。

迭代是将原像（种子）按照一定的规则，变为初像，初像再按照同样的规则再执行下去的组合变换。本指令返回的是迭代的终像，不是迭代的过程。其中“次数”就是迭代深度，“起始值”就是原像（种子），“变量”是迭代规则代表“起始值”的参数，“表达式”就是迭代规则。

**案例：**设  $a=1, b=4$ ，“迭代( $a1+b1, a1, b1, \{a, b\}, 5$ )”得到“23”。具体为：当迭代次数为 1 时， $a1=a=1, b1=b=4$ ，当迭代次数为 2 时，表达式  $a1+b1=5$ ；当迭代次数为 3 时， $a1=4, b1=5$ ，表达式  $a1+b1=9$ ；当迭代次数为 4 时， $a1=5, b1=9$ ，表达式  $a1+b1=14$ ；当迭代次数为 5 时， $a1=9, b1=14$ ，表达式  $a1+b1=23$ ；

**注：**“起始值”必须使用花括弧圈住；想得到迭代过程，请使用“迭代列表”指令。

**注：**如果迭代的初始值和表达式针对几何对象，迭代的结果就是几何对象。

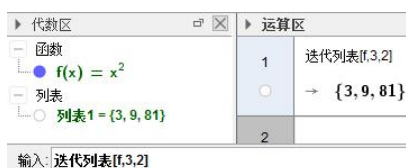
## 2.7.6 IterationList. 迭代列表

**IterationList**(**<Function>**,**<Start Value>**,**<Number of Iterations n>**)；**迭代列表**(**<表达式>**,**<起始值>**,**<迭代次数>**)。

给出长度为  $n+1$  的列表，其中第一个元素是起始值函数值，其后为函数每次迭代的结果值。

**案例：**定义函数“ $f(x)=x^2$ ”后指令“迭代列表( $f, 3, 2$ )”给出列表  $\{3, 9, 81\}$ 。其中第一个元素是初始值。

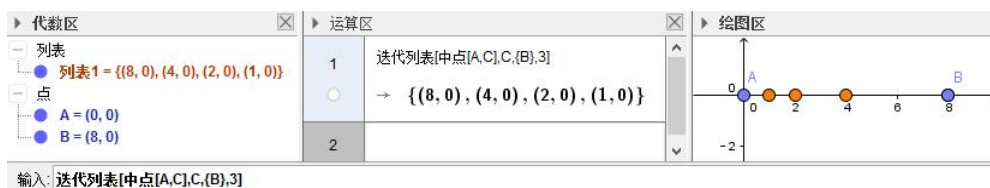
**注：**参见“迭代”指令。



**IterationList**(**<Expression>**,**<Variable Name>**,**...**,**<Start Values>**,**<Number of Iterations ns>**)；**迭代列表**(**<表达式>**,**<变量>**,**<起始值>**,**<次数>**)。

给出长度为  $n+1$  的列表，其中第一个元素是起始值函数值，其后为函数每次迭代的结果值。每次迭代后，列表中最后的元素会替代表式中的变量，这就要求有几个起始值就有几个变量。

**案例：**设有点 A 和 B，指令“迭代列表(中点(A, C), C, {B}, 3)”迭代值为： $C_0=B, C_1$ =中点(A,  $C_0$ ),  $C_2$ =中点(A,  $C_1$ ),  $C_3$ =中点(A,  $C_2$ )且得出列表  $\{C_1, C_2, C_3, C_4\}$ ，对于 A(0, 0)和 B(8, 0)的结果就是列表“ $\{(8, 0), (4, 0), (2, 0), (1, 0)\}$ ”。



**案例：**设有数值“ $f_0$ ”和“ $f_1$ ”，“迭代列表( $a+b, a, b, \{f_0, f_1\}, 5$ )”得出的列表前两个元素是  $f_0$  和  $f_1$ ，后边的就是迭代计算值， $f_2=f_0+f_1, f_3=f_1+f_2, f_4=f_2+f_3, f_5=f_3+f_4$ ，对于“ $f_0=f_1=1$ ”的结果就是列表“ $\{1, 1, 2, 3, 5, 8, \}$ ”



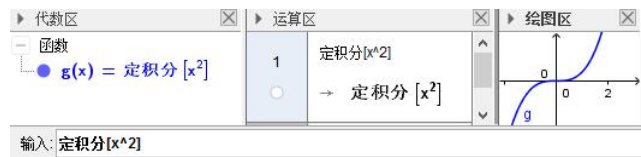
**注：**参见“迭代”指令。

**注：**如果迭代的初始值和表达式针对几何对象，迭代列表的结果就是几何对象。

## 2.7.7 NIntegral. 定积分

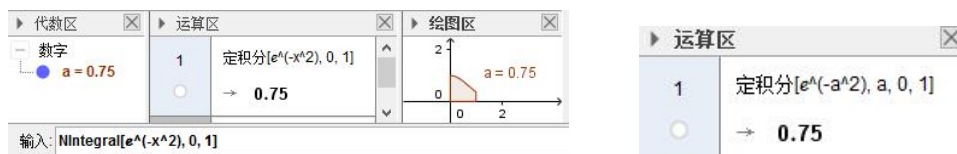
**NIntegral(<Function>); 定积分(<函数>)**。

得出给定函数的定积分。



**NIntegral(<Function>, <Start x-Value>, <End x-Value>); 定积分(<函数>, <x-起始值>, <x-终止值>)**。

计算给定函数在指定区间的定积分。



**CAS Syntax (运算区语法)**

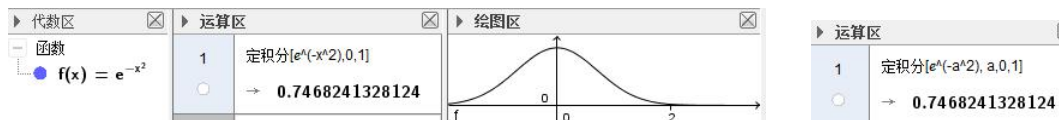
**NIntegral(<Function>, <Variable>, <Start Value>, <End Value>); 定积分(<函数>, <变量>, <积分下限>, <积分上限>)**

计算给定函数在指定区间指定变量的定积分。

**NIntegral(<Function f>, <Start x-Value a>, <End x-Value b>); 定积分(<函数>, <x-积分下限>, <x-积分上限>)**。

设 a 是 x 初始值, b 是 x 终止值, f 是函数。求函数自 a 到 b 的定积分值。

**案例:** “定积分( $e^{-x^2}$ , 0, 1)” 得出 0.746824132812427。(精度需要提前设定)



**NIntegral(<Function f>, <Variable t>, <Start variable-Value a>, <End variable-Value b>); 定积分(<函数>, <变量>, <积分下限>, <积分上限>)**。

设 a 是初始值, b 是终止值, f 是函数, t 是变量。求函数自 a 到 b 的定积分值。

**案例:** “定积分( $e^{-a^2}$ , a, 0, 1)” 得出 0.746824132812427。

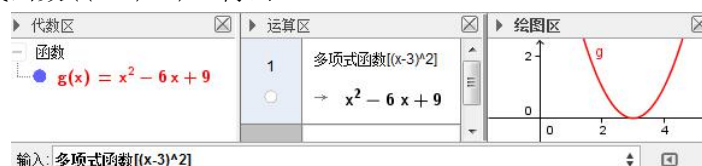
**释义:** 定积分简单讲就是求函数  $f(x)$  在区间  $(a, b)$  中图线下包围的面积。即由  $y=0$ 、 $x=a$ 、 $x=b$  和  $y=f(x)$  所围成图形的面积。

## 2.7.8 Polynomial. 多项式函数

**Polynomial(<Function>); 多项式函数(<函数>)**。

得出多项式函数的展开式。

**案例:** “多项式函数  $((x-3)^2)$ ” 得出  $x^2-6x+9$ 。

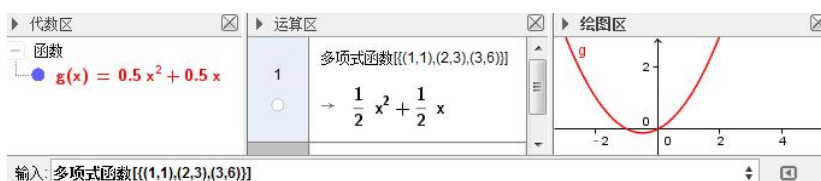


**Polynomial(<List of Points>); 多项式函数(<点列>)**。

由穿过给定的 n 个点而创建次方程为 n-1 次方插值多项式。

**案例:** “多项式函数  $((1, 1), (2, 3), (3, 6))$ ” 得出 “ $0.5x^2+0.5x$ ”。

注：如果是三点，就是创建过三点的抛物线。多个点，则是  $n-1$  次函数。有时，不能定义。



### CAS Syntax (运算区语法)

**Polynomial(<Function>); 多项式函数(<函数>)**。

展开函数写为多项式形式 (合并同类项)。

案例：“多项式函数  $((x-3)^2+(a+x)^2)$ ” 得出 “ $2x^2+(2a-6)x+a^2+9$ ”。



**Polynomial(<Function>, <Variable>); 多项式函数(<函数>, <变量>)**。

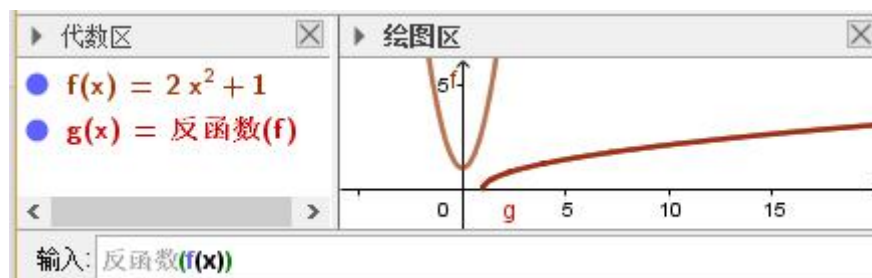
展开函数写为指定变量多项式形式 (合并同类项)。

案例：“多项式函数  $((x-3)^2+(a+x)^2, a)$ ” 得出 “ $a^2+2xa+2x^2-6x+9$ ”。

## 2.7.9 NInvert. 反函数

**NInvert(<Function>); 反函数(<函数>)**

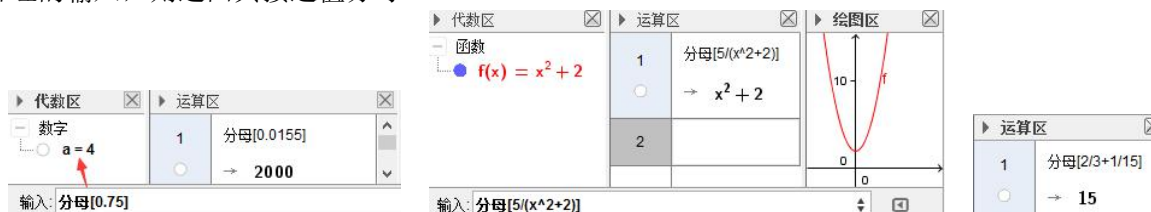
新加入指令，官网没有帮助。得到给定函数的反函数，并绘制图象。



## 2.7.10 Denominator. 分母

**Denominator(<Number>); 分母(<数字>)**。

对于一个有理数返回它的分母。因采用了数值计算方法，限制了输出很小的数字分母。对于不合理的输入，则返回其接近值分母。



**Denominator(<Function>); 分母(<函数>)**。

返回函数的分母。

案例：“分母  $(5/(x^2+2))$ ” 得出 “ $f(x)=(x^2+2)$ ”。

注：作为一种快速数值方法 (只要数值不是太大)，假设  $a$  是一个分式，可以用于指令：“公分母  $(a, a)$ ” 得到分母和 “round  $(a * 公分母(a, a))$ ” 得到分子。

参见“分子”指令。

### CAS Syntax (运算区语法)

Denominator(<Expression>); 分母(<表达式>)。

返回一个有理数或者表达式的分母。

案例：“分母(2/3+1/15)”得出 15。

## 2.7.11 Numerator. 分子

Numerator(<Number>); 分子(<数字>)。

对于一个有理数返回它的分子。因采用了数值计算方法，限制了输出很小的数字分子。对于不合理的输入，则返回其接近值分子。

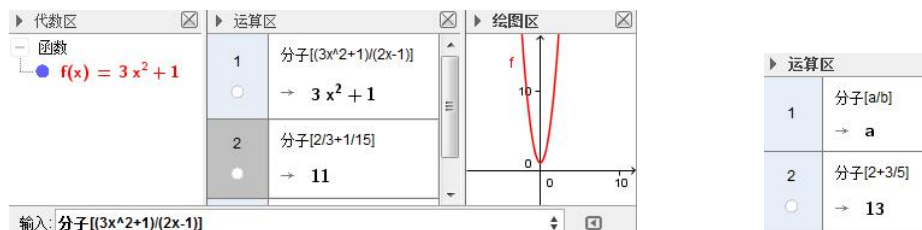


Numerator(<Function>); 分子(<函数>)。

返回函数的分子。

案例：“分子((3x^2+1)/(2x-1))”得出  $f(x)=3x^2+1$ 。

注：作为一种快速数字方法（数值不是太大）可以在指令栏如此：“公分母(a, a)”求出公分母而“round(a\*公分母(a, a))”得到分子。参见“分母”指令。



### CAS Syntax (运算区语法)

Numerator(<Expression>); 分子(<表达式>)。

返回表达式的分子。

案例：“分子(2/3+1/15)”得出 11。

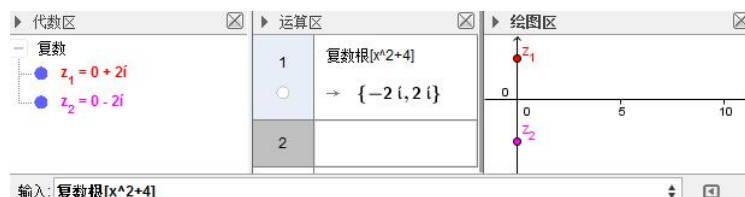
如果在 GeoGebra 中没有提前定义变量 a、b 和 c，然后“分子(a/b)”得出“a”而“分子(a+b/c)”得出“ac+b”。

## 2.7.12 ComplexRoot. 复数根

ComplexRoot(<Polynomial>); 复数根(<多项式>)。

计算多项式关于 x 的复数根，绘图区创建对应的点。

案例：“复数根(x^2+4)”得出 (0+2i) 和 (0-2i)。



### CAS Syntax (运算区语法)

ComplexRoot(<Polynomial>); 复数根(<多项式>)。

找到多项式关于 x 的复数根。

案例：“复数根( $x^2+4$ )”得出 $(0+2i)$ 和 $(0-2i)$ 。

注：可使用“复数解”指令替代。

### 2.7.13 InflectionPoint (TurningPoint) . 拐点

InflectionPoint(<Polynomial>); 拐点(<多项式>)。

得出多项式的全部拐点，其为函数图象上的点对象。

案例：“拐点( $x^3$ )”得出“(0, 0)”。



### 2.7.14 Normalize. 归一化

Normalize(<List of Numbers>); 归一化(<数字列表>)。

返回给定点归一化格式的数字列表。

案例：指令“归一化({1, 2, 3, 4, 5})”返回列表“{0, 0.25, 0.5, 0.75, 1}”。



Normalize(<List of Points>); 归一化(<点列>)。

返回给定点归一化格式的点列。

案例：指令“归一化({(1, 5), (2, 4), (3, 3), (4, 2), (5, 1)})”返回点列“{(0, 1), (0.25, 0.75), (0.5, 0.5), (0.75, 0.25), (1, 0)}”。



注：这个指令不支持3D。

归一化操作使用线性函数“ $\frac{x - \text{最小值}[\text{列表}]}{\text{最大值}[\text{列表}] - \text{最小值}[\text{列表}]}$ ”，将值  $x$  映射到区间(0,1)。

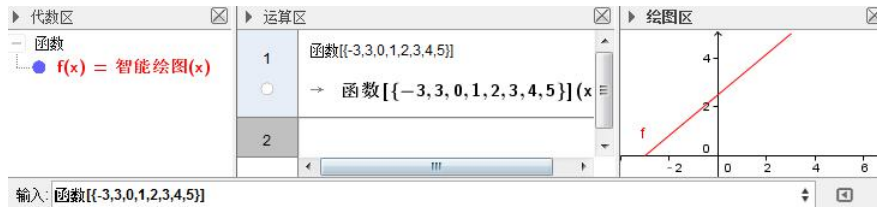
### 2.7.15 Function. 函数

Function(<List of Numbers>); 函数(<{x-起始值, x-终止值, 区间内等距的若干纵坐标值}>)。

得出下述的函数：前两个数值确定了初始的  $x$ -值与终止的  $x$ -值。剩余的数值是函数  $y$ -值所确定的等距取值区间。

案例：“函数({2, 4, 0, 1, 0, 1, 0})”得出一个在  $x=2$  与  $x=4$  之间的三角波浪；“函数({-3, 3, 0, 1, 2, 3, 4, 5})”得出在  $x=-3$  与  $x=3$  之间、斜率=1 的线型方程。





Function(<Function>, <Start x-Value>, <End x-Value>); 函数(<函数>, <x-起始值>, <x-终止值>)

生成函数 f 在数值区间上的函数图象。

**案例：**“f(x)=函数(x<sup>2</sup>, -1, 1)”得到函数“x<sup>2</sup>”在区间(-1, 1)上的函数图象。如果输入“g(x)=2 f(x)”，将会得到“g(x)=2 x<sup>2</sup>”，但这个函数的定义域并不限制在(-1, 1)区间上。

**注：**只是在限定区域内显示图形，不是设定了函数的变量范围。要限定函数变量范围，必须使用“如果”指令，用法如下：“f(x)=如果(-5<x&&x<5, x<sup>2</sup>+x+1)”，这时如果输入：f(8)，那么就会出现“b 未定义”。

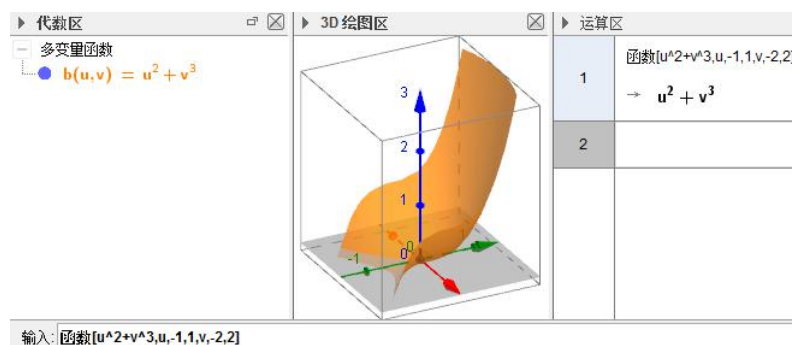
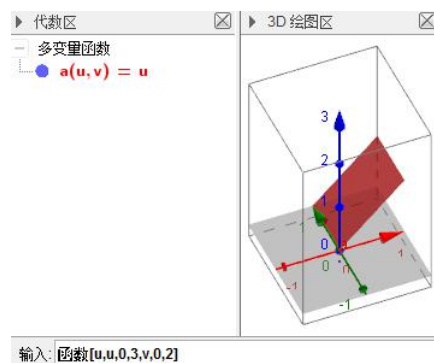
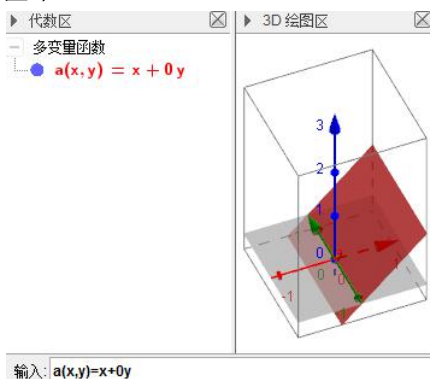


Function(<Expression>, <Parameter Variable 1>, <Start Value>, <End Value>, <Parameter Variable 2>, <Start Value>, <End Value>); 函数(<表达式>, <参变量 1>, <起始值>, <终止值>, <参变量 2>, <起始值>, <终止值>)

在 3D 空间限定双变量函数特定表面的外观。

**案例：**表达式“a(x, y)=x+0y”创建双变量函数，其图象在 3D 中是平面 z 为 a(x, y)=x。“函数(u, u, 0, 3, v, 0, 2)”创建双变量函数 b(u, v)=u，它在 3D 空间中的图象为 a(x, y)=x 的 z 平面内“多边形((0, 0, 0), (3, 0, 3), (3, 2, 3), (0, 2, 0))”。

“函数(u<sup>2</sup>+v<sup>3</sup>, u, -1, 1, v, -2, 2)”在指令栏和运算区都可以执行，形成的结果在代数区、运算区和 3D 绘图区中。



运算区语法:

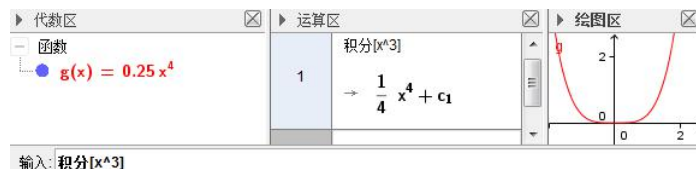
Function(<Function>,<Start x-Value>,<End x-Value>); 函数(<函数>,<x-起始值>,<x-终止值>).

## 2.7.16 Integral. 积分

Integral(<Function>); 积分(<函数>).

给出函数关于主变量的不定积分。

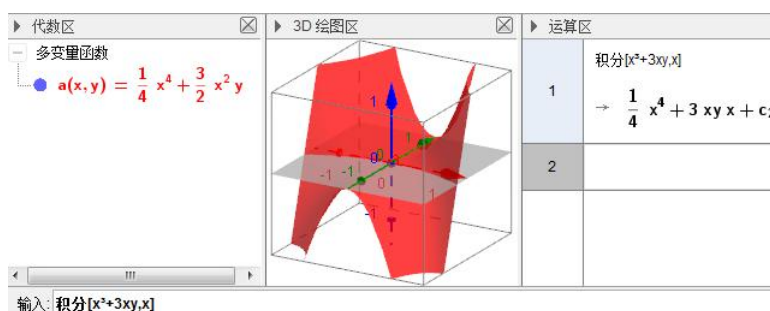
案例: “积分( $x^3$ )” 得出 “ $0.25x^4$ ”。



Integral(<Function>,<Variable>); 积分(<函数>,<变量>).

给出函数关于指定变量的偏积分。

案例: “积分( $x^3+3xy, x$ )” 给出 “ $a(x, y) = \frac{1}{4}x^4 + \frac{3}{2}x^2y$ ”



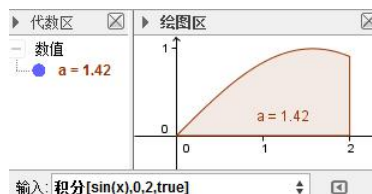
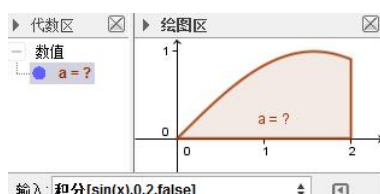
Integral(<Function>,<Start x-Value>,<End x-Value>); 积分(<函数>,<x-积分下限>,<x-积分上限>).

给出函数在区间(x-起始值, x-终止值)上关于主变量的不定积分。

注: 指令结果会用阴影线标识函数的图象与 x-轴之间的区域。

Integral(<Function>,<Start x-Value>,<End x-Value>,<Boolean Evaluate>); 积分(<函数>,<x-积分下限>,<x-积分上限>,<是否给出积分值?true|false>).

给出函数在区间(x-起始值, x-终止值)上关于主变量 “x” 的定积分。如果<是否给出积分值?>为 “true”, 则用阴影线标识相关区域并给出积分值; 如果<是否给出积分值?>为 “false”, 则用阴影线标识相关区域但不给出积分值。



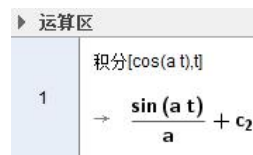
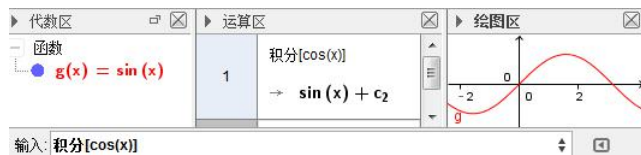
CAS Syntax (运算区语法)

Integral(<Function>); 积分(<函数>).

给出函数关于主变量的不定积分。

案例: “积分( $\cos(x)$ )” 给出  $\sin(x)+c1$ 。





**Integral(<Function>, <Variable>)**

给出函数关于指定变量的不定积分。

**案例：**“积分(cos(a t), t)”给出“ $\frac{\sin(at)}{a} + c1$ ”。

**Integral(<Function>, <Start x-Value>, <End x-Value>); 积分(<函数>, <x-积分下限>, <x-积分上限>)**。

给出函数在区间(x-起始值, x-终止值)上关于主变量的不定积分。

**案例：**“积分(cos(x), a, b)”得出“sin(b)-sin(a)”。

**Integral(<Function>, <Variable>, <Start x-Value>, <End x-Value>); 积分(<函数>, <变量>, <x-积分下限>, <x-积分上限>)**。

返回函数关于给定变量在区间(x-积分下限, x-积分上限)的不定积分。

**案例：**“积分(cos(t), t, a, b)”得出“sin(u)-sin(x)”。



## 2.7.17 IntegralBetween. 积分介于

**IntegralBetween(<Function>, <Function>, <Number>, <Number>); 积分介于(<函数 1>, <函数 2>, <x-积分下限>, <x-积分上限>)**。

给出两个函数 f 和 g 的差值 f(x) - g(x) 在区间(a,b)上关于主变量的不定积分，其中 a 为第一个数值参数而 b 为第二个。

**注：**这个指令同样在函数 f 图象和 g 图象间构造阴影。

**IntegralBetween(<Function>, <Function>, <Number>, <Number>, <Boolean Evaluate>); 积分介于(<函数 1>, <函数 2>, <x-积分下限>, <x-积分上限>, <是否给出积分值? true|false>)**。

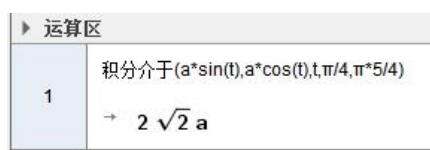
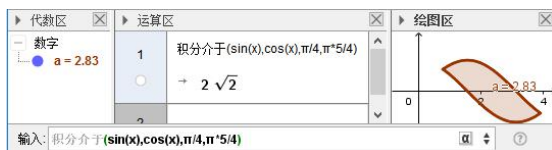
给出两个函数 f 和 g 的差值 f(x) - g(x) 在区间(a,b)上关于主变量的不定积分，其中 a 为第一个数值参数而 b 为第二个。如果<是否给出积分值?>为 true，则用阴影线标识相关区域并给出积分值；如果<是否给出积分值?>为 false，则用阴影线标识相关区域但不给出积分值。

**CAS Syntax (运算区语法)**

**IntegralBetween(<Function>, <Function>, <Number>, <Number>); 积分介于(<函数 1>, <函数 2>, <x-积分下限>, <x-积分上限>)**。

给出两个函数 f 和 g 的差值 f(x) - g(x) 在区间(a,b)上关于主变量的不定积分，其中 a 为第一个数值参数而 b 为第二个。

**案例：**“积分介于(sin(x),cos(x), π/4, π\*5/4)”得出  $2\sqrt{2}$ 。



**IntegralBetween(<Function>, <Function>, <Variable>, <Number>, <Number>); 积分介于(<函数 1>, <函数 2>, <变量>, <积分下限>, <积分上限>)**。

给出两个函数 f 和 g 的差值 f(x) - g(x) 在区间(a,b)上关于给定变量的不定积分，其中 a 为第一个

数值参数而 b 为第二个。

案例：“积分介于 $(a*\sin(t), a*\cos(t), t, \pi/4, \pi*5/4)$ ”得出“ $2\sqrt{2}a$ ”。

### 2.7.18 Limit. 极限

**Limit(<Function>, <Value>); 极限(<函数>, <数值>)**。

计算指定主函数变量趋向某值时函数极限。

案例：“极限 $((x^2+x)/x^2, +\infty)$ ”得出 1。



注：GeoGebra 不是所有的极限都能计算出来，这种情况下将返回未定义（错误的结果是未定义）。

**CAS Syntax (运算区语法)**

**Limit(<Expression>, <Value>); 极限(<表达式>, <数值>)**。

计算指定主函数变量趋向某值时表达式的极限。

案例：“极限 $(a \sin(x)/x, 0)$ ”得出 a。

**Limit(<Expression>, <Variable>, <Value>); 极限(<表达式>, <变量>, <数值>)**。

计算指定主函数变量趋向指定值时表达式的极限。

案例：“极限 $(a \sin(v)/v, v, 0)$ ”得出 a。

注：GeoGebra 不是所有的极限都能计算出来，这种情况下将返回未定义（错误的结果是未定义）。

如果要求分段定义函数的极限，需要使用右极限或左极限，例如“右极限(如果 $(x>1, x^2, -2x), 1)$ ”。

参见“渐近线”指令，“左极限”指令和“右极限”指令。

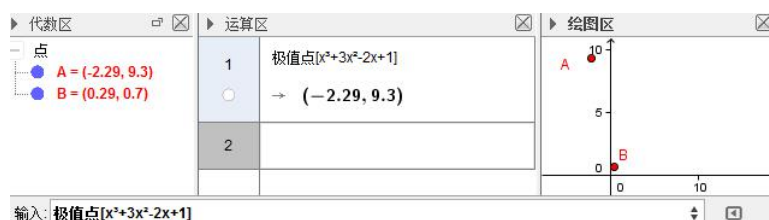
### 2.7.19 Extremum. 极值点

这个指令在英语变型中有不同的拼写：Extremum(US)、TurningPoint(UK+Aus)。

**Extremum(<Polynomial>); 极值点(<多项式>)**。

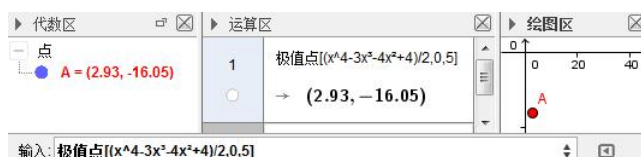
得出多项式函数的所有局部极值并在函数图象上创建对应的点。

案例：“极值点 $(x^3+3x^2-2x+1)$ ”创建极值点 $(0.29, 9.30)$ 和 $(-2.29, 9.30)$ 且显示在绘图区。



**Extremum(<Function>, <Start x-Value>, <End x-Value>); 极值点(<连续函数>, <x-起始值>, <x-终止值>)**。

计算（数字表示）函数在开区间 $(\langle x-起始值 \rangle, \langle x-终止值 \rangle)$ 上的极值。



案例：“极值点 $((x^4-3x^3-4x^2+4)/2, 0, 5)$ ”创建区间内的极值点 $(2.93, -16.05)$ 并显示在绘图

区。

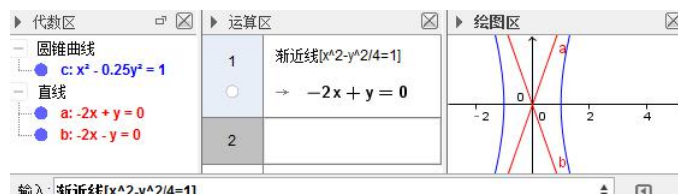
注：函数应在区间( $x$ -起始值), ( $x$ -终止值)上连续，否则在间断点附近可能得出错误极值。

## 2.7.20 Asymptote. 渐近线

Asymptote( $\langle$ Conic $\rangle$ ); 渐近线( $\langle$ 双曲线 $\rangle$ )。

得出圆锥曲线的全部渐近线。

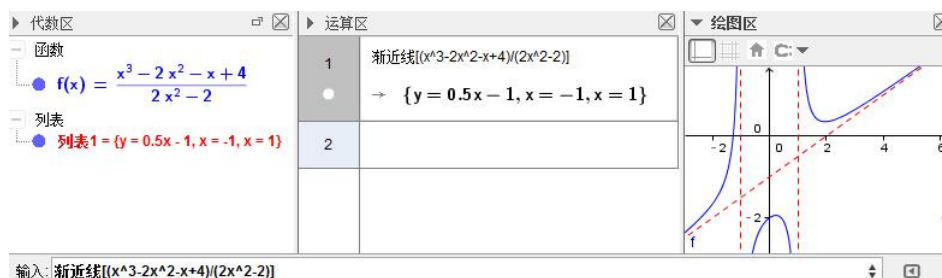
案例：“渐近线( $x^2 - y^2/4 = 1$ )”返回直线“ $-2x + y = 0$ ”和“ $-2x - y = 0$ ”。



Asymptote( $\langle$ Function $\rangle$ ); 渐近线( $\langle$ 函数 $\rangle$ )。

GeoGebra 会试图找到函数的渐近线并将它们返回在一个列表中。有可能不全部找出来，例如类似于  $\ln(x)$  这样的“非-有理函数”的垂直渐近线。

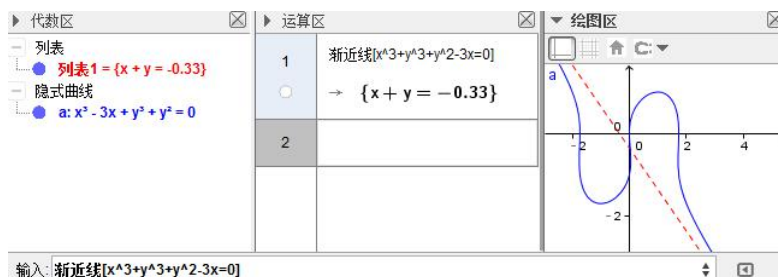
案例：“渐近线( $(x^3 - 2x^2 - x + 4)/(2x^2 - 2)$ )”返回列表“ $\{y = 0.5x - 1, x = 1, x = -1\}$ ”。



Asymptote( $\langle$ Implicit Curve $\rangle$ ); 渐近线( $\langle$ 隐式曲线 $\rangle$ )。

得出一个包含隐式曲线（高次曲线）全部渐近线的列表。

案例：“渐近线( $x^3 + y^3 + y^2 - 3x = 0$ )”返回列表“ $\{x + y = -0.33\}$ ”。



## 2.7.21 SolveODE. 解常微分方程

SolveODE( $\langle$ f'(x, y) $\rangle$ ); 解常微分方程( $\langle$ f'(x, y) $\rangle$ )。

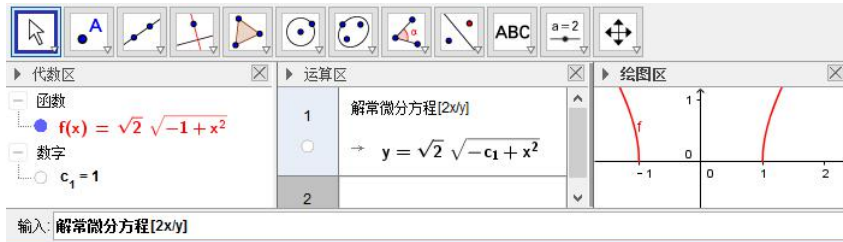
尝试寻找一阶常微分方程 (ODE)  $\frac{dy}{dx}(x) = f(x, y(x))$  的精确解。

案例：“解常微分方程 ( $2x/y$ )”得出“ $(y = \sqrt{2\sqrt{-c_1 + x^2}})$ ”。 $c_1$  为常数。

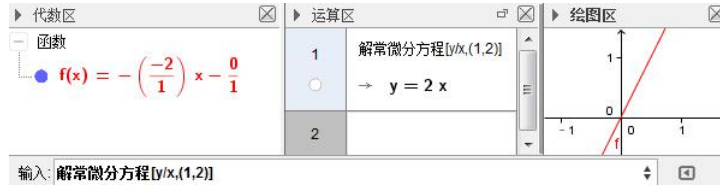
注： $c_1$  会自动创建为滑动条数值，且以辅助对象存在于代数区。

SolveODE( $\langle$ f'(x, y) $\rangle$ ,  $\langle$ Point on f $\rangle$ ); 解常微分方程( $\langle$ f'(x, y) $\rangle$ ,  $\langle$ f 上的点 $\rangle$ )。

尝试寻找一阶常微分方程  $\frac{dy}{dx}(x) = f(x, y(x))$  精确的解，并使其解通过指定的点。



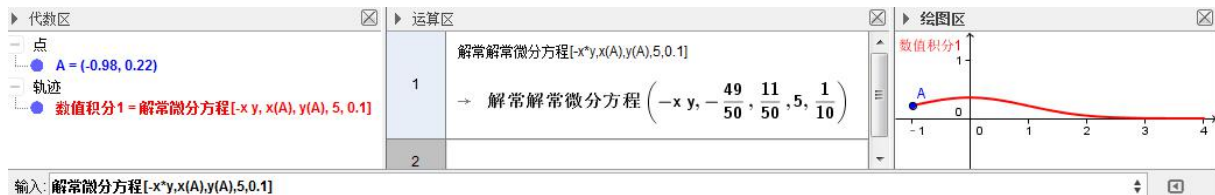
案例：“解常微分方程(y/x, (1, 2))”得出“y=2x”。



SolveODE(<f' (x, y)>, <Start x>, <Start y>, <End x>, <Step>); 解常微分方程(<f' (x, y)>, <起始 x>, <起始 y>, <终止 x>, <步长>).

用指定的(x, y)起始点、x的终止点和步长的数值来解一阶常微分方程 $\frac{dy}{dx}(x)=f(x,y)$ 。

案例：“解常微分方程(-x\*y, x(A), y(A), 5, 0.1)”使用之前已定义的点A作为起始点解常微分方程： $\frac{dy}{dx} = -xy$ 。



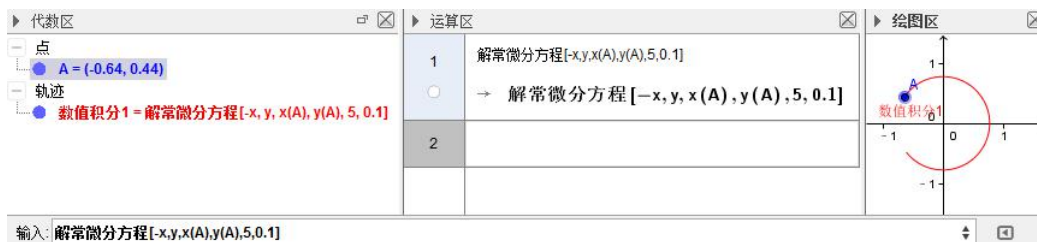
注：“长度(<轨迹>)”可以让查找出有多少个点在轨迹计算列中；“最前元素(<轨迹>, <数值>)”可以让提取一些点成为一个列表。

若要寻找“反向”结果，只需要将终止x输入为一个负值，例如“解常微分方程(-x\*y, x(A), y(A), -5, 0.1)”。



SolveODE(<y'>, <x'>, <Start x>, <Start y>, <End t>, <Step>); 解常微分方程(<y'>, <x'>, <起始 x>, <起始 y>, <终止 t>, <步长>).

用指定的起始点、一个按照步长变化的t作为终止值，解一阶常微分方程 $\frac{dy}{dx} = \frac{f(x,y)}{g(x,y)}$ 。指令的这种形式不论第一步求解是否失败仍会运作。



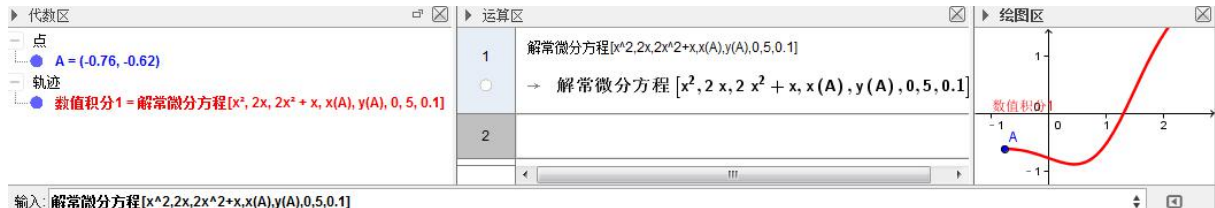
**案例：**“解常微分方程(-x, y, x(A), y(A), 5, 0.1)”使用先前已定义的点 A 作为起始点解常微分方程  $\frac{dy}{dx} = \frac{-x}{y}$ 。

**注：**若要寻找“反向”结果，只需要将终止 t 输入为一个负值，例如解常微分方程(-x, y, x(A), y(A), -5, 0.1)。

**SolveODE(<b(x)>, <c(x)>, <f(x)>, <Start x>, <Start y>, <Start y'>, <End x>, <Step>); 解常微分方程(<b(x)>, <c(x)>, <f(x)>, <起始 x>, <起始 y>, <S 起始 y'>, <终止 x>, <步长>。**

解二阶常微分方程“ $y''+b(x)y'+c(x)y=f(x)$ ”。

**案例：**“解常微分方程( $x^2, 2x, 2x^2+x, x(A), y(A), 0, 5, 0.1$ )”使用之前已定义的点 A 作为起始点解二阶常微分方程。



**注：**返回的结果总是为轨迹。当前的算法是基于龙格-库塔的数值方法（库塔法）。

**注：**参见“斜率场”指令。

**CAS Syntax (运算区语法)**

**SolveODE(<Equation>); 解常微分方程(<方程>。**

尝试寻找一或二阶常微分方程 (ODE) 的精确解。一和二阶可以使用 y' 和 y'' 分别指定。

**案例：**“解常微分方程( $y'=y/x$ )”得出“ $y=c_2x$ ”。



**SolveODE(<Equation>, <Point(s) on f>); 解常微分方程(<方程>, <f 上的点>。**

尝试寻找一或二阶过给定点的常微分方程 (ODE) 的精确解。

**案例：**“解常微分方程( $y'=y/x, (1, 2)$ )”得出“ $y=2x$ ”。

**SolveODE(<Equation>, <Point(s) on f>, <Point(s) on f'>); 解常微分方程(<方程>, <f 上的点>, <f'上的点>。**

尝试寻找一或二阶过给定点 f 和 f' 的常微分方程 (ODE) 的精确解。

**案例：**“解常微分方程( $y''-3y'+2=x, (2,3), (1,2)$ )”，得出  $y = \frac{-9x^2 e^3 + 30x e^3 - 32(e^3)^2 + 138e^3 + 32e^{3x}}{54e^3}$ 。

**SolveODE(<Equation>, <Dependent Variable>, <Independent Variable>, <Point(s) on f>); 解常微分方程(<方程>, <因变量>, <自变量>, <f 上的点>。**

尝试寻找一或二阶过给定点的常微分方程 (ODE) 的精确解。

**案例：**“解常微分方程( $v'=v/w, v, w, (1, 2)$ )”得出“ $v=2w$ ”。

**SolveODE(<Equation>, <Dependent Variable>, <Independent Variable>, <Point(s) on f>, <Point(s) on f'>); 解常微分方程(<方程>, <因变量>, <自变量>, <f 上的点>, <f'上的点>。**

尝试寻找一或二阶过给定点 f 和 f' 的指定变量的常微分方程 (ODE) 的精确解。

**案例：**“解常微分方程( $v'=v/w, v, w, (1, 2), (0, 2)$ )”得出“ $v=2w$ ”。

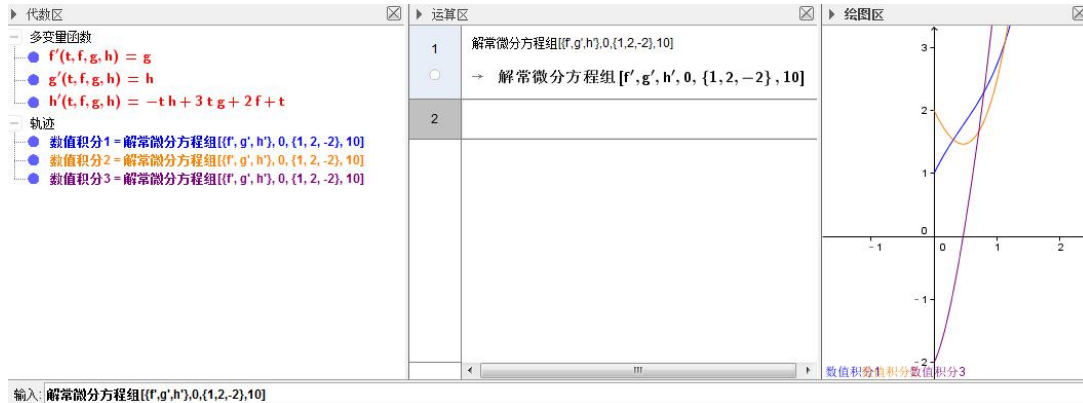
**注：**为了兼容指令栏，如果第一个参数不是携带 y' 或 y'' 的表达式，支持左边是 y' 的常微分方程。

## 2.7.22 NSolveODE. 解常微分方程组

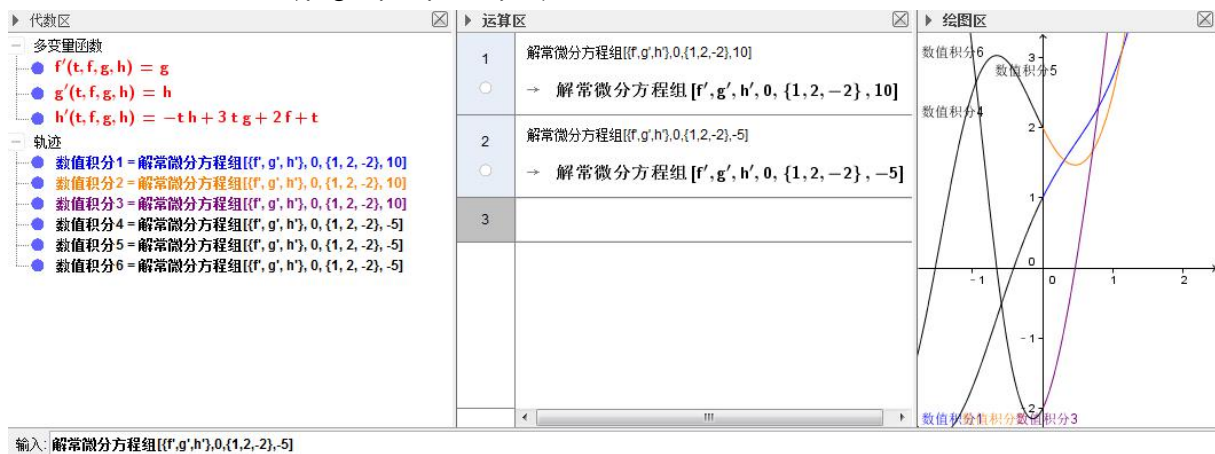
NSolveODE(<List of Derivatives>, <Initial x-coordinate>, <List of Initial y-coordinates>, <Final x-coordinate>); 解常微分方程组(<导数列表>, <x 坐标初值>, <y 坐标初值列表>, <x 坐标终值>)

解(用数值求解方法)常微分方程组。

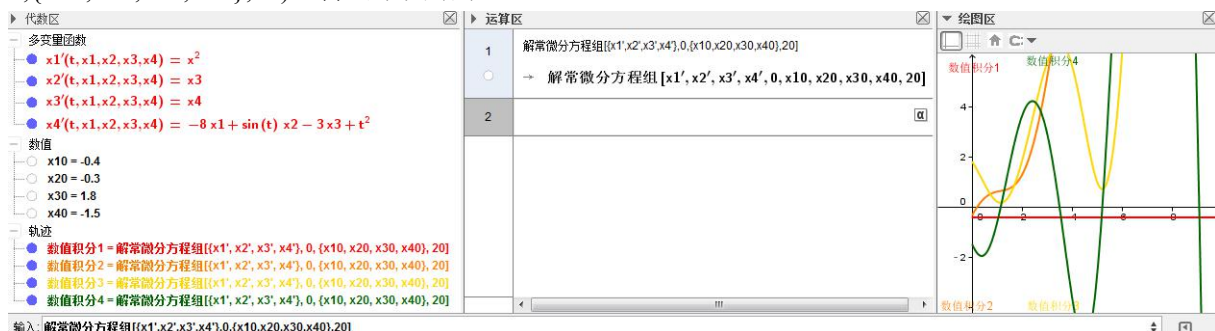
案例: 设  $f(t, f, g, h) = g$ ;  $g'(t, f, g, h) = h$ ;  $h'(t, f, g, h) = -th + 3tg + 2f + t$ ; “解常微分方程组( $\{f, g', h'\}$ , 0,  $\{1, 2, -2\}$ , 10)” , 得到 3 个数值积分。



“解常微分方程组( $\{f, g', h'\}$ , 0,  $\{1, 2, -2\}$ , -5)” 又得到 3 个数值积分(这次是反向)。



案例: 设:  $x1'(t, x1, x2, x3, x4) = x^2$ ;  $x2'(t, x1, x2, x3, x4) = x3$ ;  $x3'(t, x1, x2, x3, x4) = x4$ ;  $x4'(t, x1, x2, x3, x4) = -8x1 + \sin(t)x2 - 3x3 + t^2$ ;  $x10 = -0.4$ ;  $x20 = -0.3$ ;  $x30 = 1.8$ ;  $x40 = -1.5$ 。“解常微分方程组( $\{x1', x2', x3', x4'\}$ , 0,  $\{x10, x20, x30, x40\}$ , 20)” 得出下图结果。



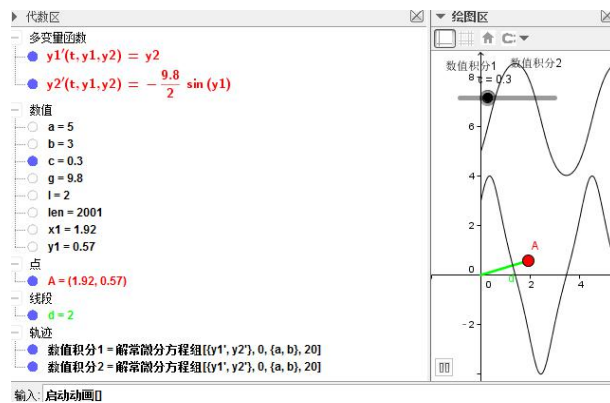
案例: 顺序在指令栏贴入以下指令并执行, 会制作一个动态钟摆。

```

g=9.8
l=2
a=5
b=3
y1'(t,y1,y2)=y2
y2'(t,y1,y2)=(-g/l) sin(y1)
解常微分方程组[[y1',y2'],0,{a,b},20]
len=长度(数值积分1)
c=参数[0,1,1/len,1,100,false,true,true,false]
x1=l sin(y(描点(数值积分1,c)))
y1=-l cos(y(描点(数值积分1,c)))
A=(x1,y1)
线段((0,0),A)
启动动画()

```

$g=9.8$ ;  $l=2$ ;  $a=5$  (起始位置);  $b=3$  (拉力);  $y_1'(t,y_1,y_2)=y_2$ ;  $y_2'(t,y_1,y_2)=(-g/l) \sin(y_1)$ ; “解常微分方程组( $\{y_1',y_2'\},0,\{a,b\},20$ )”;  $len=长度(数值积分1)$ ;  $c=参数(0,1,1/len,1,100,false,true,true,false)$ ;  $x_1=l \sin(y(描点(数值积分1,c)))$ ;  $y_1=-l \cos(y(描点(数值积分1,c)))$ ;  $A=(x_1,y_1)$ ; 线段( $(0,0),A$ ); 启动动画()。



### 2.7.23 RectangleSum. 矩形法则

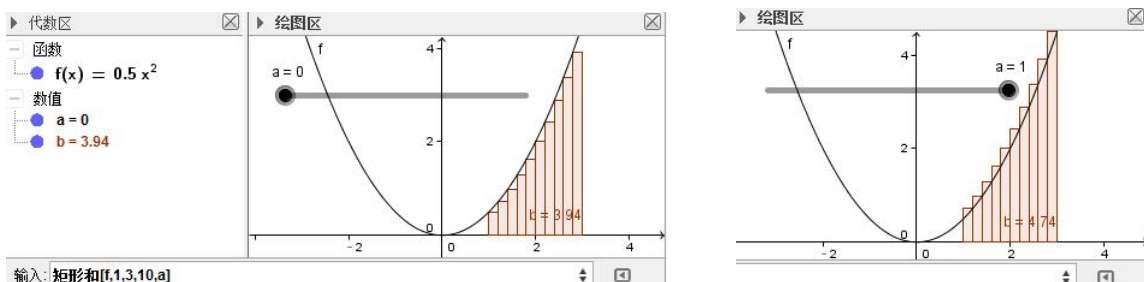
RectangleSum(<Function>, <Start x-Value>, <End x-Value>, <Number of Rectangles>, <Position for rectangle start>); 矩形法则(<函数>, <x-起始值>, <x-终止值>, <矩形数量>, <矩形起始位置>).

用  $n$  个矩形计算函数在区间内的矩形(面积)和, “矩形起始位置”由各段左长(每个矩形区段左上角的顶点到函数距离)来决定, 由一个分数(小数)  $d$  ( $0 \leq d \leq 1$ ) 控制。当  $d=0$  时等于“左和”指令, 而当  $d=1$  时则计算指定函数的“右和”。这个指令同样会画出左和的矩形组块。

**注:** 这个指令绘制出与左和相同的矩形。参见指令: “上和”、“下和”、“左和”、“梯形法则”。

**案例:** 做函数  $f(x)=0.5x^2$  的一个矩形法则。

顺序在指令栏分别键入并执行  $f(x)=0.5x^2$ ; “ $a=0$ ”; “矩形法则( $f, 1, 3, 10, a$ )”。改变  $a$  的值可见“矩形起始位置”的变化。

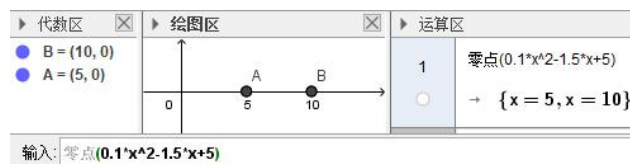


### 2.7.24 Root. 零点

Root(<Polynomial>); 零点(<多项式>).

得出多项式全部的根, 也就是函数图象与 x 轴的交点。

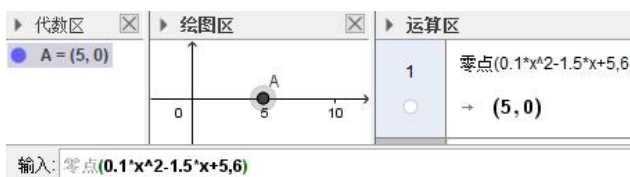
案例: “零点( $0.1x^2-1.5x+5$ )” 得出 “A=(5, 0)” 和 “B=(10, 0)”。



Root(<Function>, <Initial x-Value>); 零点(<函数>, <x-初值>).

以 x-初值为起始值用数值迭代法得出函数的一个根。

案例: “零点( $0.1x^2-1.5x+5, 6$ )” 得出 “A=(5, 0)”。



Root(<Function>, <Start x-Value>, <End x-Value>); 零点(<函数>, <x-起始值>, <x-终止值>).

设初始值 a 和终止值 b, 使用指令可得出用数值迭代法解得的区间在 (a, b) 上函数一个根。

案例: “零点( $0.1x^2-1.5x+5, 8, 13$ )” 得出 “A=(10, 0)”。

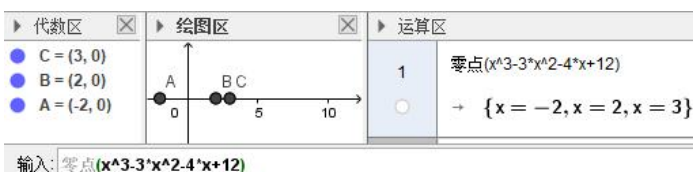
CAS Syntax (运算区语法)

Root(<Polynomial>); 零点(<多项式>).

得出多项式全部的根, 也就是函数图象与 x-轴的交点。

案例: “零点( $x^3-3x^2-4x+12$ )” 得出 “{x=3, x=2, x=-2}”。

注: 在运算区, 这个指令是 “精确解” 指令的特例。

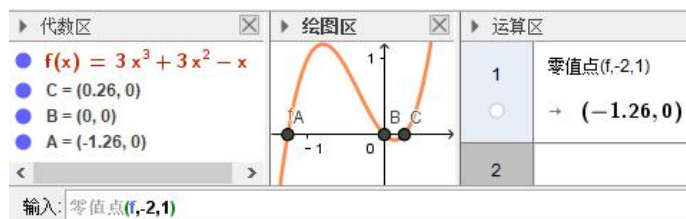


## 2.7.25 Roots. 零值点

Roots(<Function>, <Start x-Value>, <End x-Value>); 零值点(<函数>, <x-起始值>, <x-终止值>).

计算函数在指定区间上的零值点 (函数的根)。

案例: “零值点(f, -2, 1)” 用于 “ $f(x)=3x^3+3x^2-x$ ” 得出 “A=(-1.264, 0), B=(0, 0), C=(0.264, 0)”。



注: 计算函数在指定区间上的, 必须是区间上的连续函数。由于这是一种数值算法得出的结果, 因此有时不一定能得出所有的零值点。运算区得到的不是所有的零值点。



参见“零点”指令。

### 2.7.26 RootList. 零值点列

**RootList(<List>); 零值点列(<数字列表>)**

将数字列表  $\{a_1, a_2, \dots, a_n\}$  转换为点列  $\{(a_1, 0), (a_2, 0), \dots, (a_n, 0)\}$ , 同时在绘图区绘制这些点。

**案例:** “零值点列( $\{3, 4, 5, 2, 1, 3\}$ )” 返回点列 “ $\{(3, 0), (4, 0), (5, 0), (2, 0), (1, 0), (3, 0)\}$ ”。

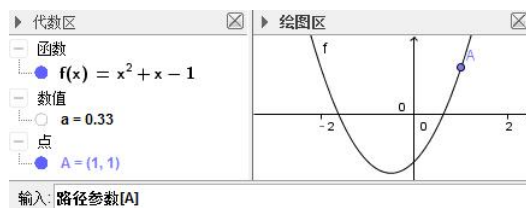


### 2.7.27 PathParameter. 路径值

**PathParameter(<Point On Path>); 路径值(<路径上的点>)**

返回属于一条路径上的点的参数 (也就是一个范围从 0 到 1) 的数值。

**案例:** 假设函数 “ $f(x) = x^2 + x - 1$ ” 且点 “ $A = (1, 1)$ ” 是函数上一个点 (在函数图象上构造的点)。“路径值(A)” 得出 “ $a = 0.47$ ”。(有时不是此值)



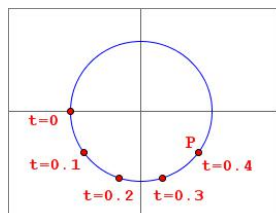
在下表中用到两个函数:  $f(x) = \frac{x}{1+|x|}$ ,  $\phi(X, A, B) = \frac{\vec{AX} \cdot \vec{AB}}{|\vec{AB}|^2}$ 。函数 1 是将全部实数映射到区间(-

1, 1)之内所用的函数; 函数 2 是从线 AB 到实数域的线性映射, 其将线的左端点 A 映射为 0, 右端点 B 为 1。这两个函数就能把路径上的点的路径值确定在-1 和 1 之间。下表就是不同路径上的点的路径值计算方法。

直线 AB	$\frac{(f(\phi(X, A, B)) + 1)}{2}$
射线 AB	$f(\phi(X, A, B))$
线段 AB	$\phi(X, A, B)$
圆(圆心为 C 半径为 r)	点 $X = C + (r \cdot \cos \alpha, r \cdot \sin \alpha)$ , 其中 $\alpha \in (-\pi, \pi)$ , 其路径值为 $\frac{\alpha + \pi}{2\pi}$ 。

椭圆(中心为C 半轴长为a <sup>→</sup> ,b <sup>→</sup> )	点 $X=C+a^{\vec{}}\cdot\cos\alpha+b^{\vec{}}\cdot(\sin\alpha)$ , 其中 $\alpha\in(-\pi,\pi)$ , 其路径值为 $\frac{\alpha+\pi}{2\pi}$ 。
抛物线(顶点为V, 准线为v <sup>→</sup> )	点 $V+p\cdot t^2\cdot v^{\vec{}}+p\cdot t\cdot v^{\vec{}}$ 的参数为 $\frac{f(t)+1}{2}$
折线 $A_1\dots A_n$	若 X 在 $A_kA_{k+1}$ 上, 则其参数为 $\frac{k-1+\phi(X, A, B)}{n}$ 。
多边形 $A_1\dots A_n$	若 X 在 $A_kA_{k+1}$ 上(假设 $A_{n+1}=A_1$ ), 则 X 路径值为 $\frac{k-1+\phi(X, A, B)}{n+1}$ 。
路径集合 $L=\{p_1,\dots,p_n\}$	若 X 在 $p_k$ 上, 且 X 相对于 $p_k$ 的参数为 t 的话, 则相对于 L 的参数为 $\frac{k-1+t}{n}$
点集 $L=\{A_1,\dots,A_n\}$	$A_k$ 的路径值为 $\frac{k-1}{n}$ 。指令“描点(L, t)”会得到“ $A_{tn+1}$ ”。
双曲线、轨迹、曲线方程式, 目前没有公式	

下图显示出圆上的点与其参数之间的关系(圆为  $c:x^2+y^2=1$ )。



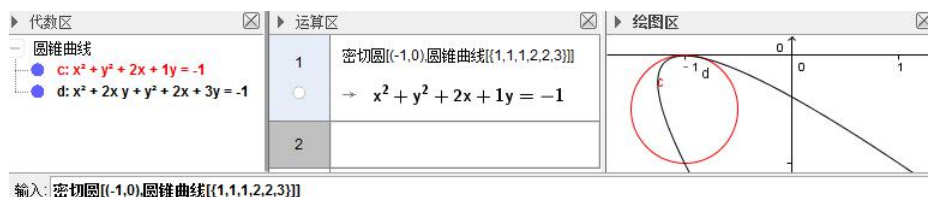
**编者注:**除了线段的路径值与“几何画板”的“点的值”相同,其余的都有特殊定义。因为定义过于宽泛,没有几何画板中“点的值”更实用。

## 2.7.28 OsculatingCircle. 密切圆

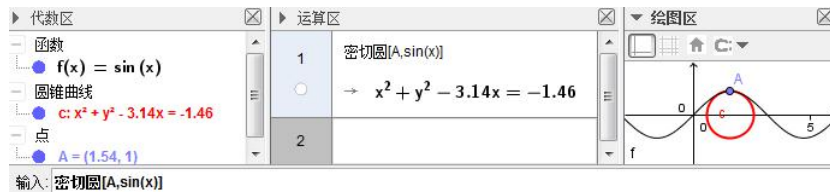
**OsculatingCircle(<Point>, <Object>); 密切圆(<点>, <对象>)**。

得出对象(函数、曲线、圆锥曲线)在指定点上的密切圆。

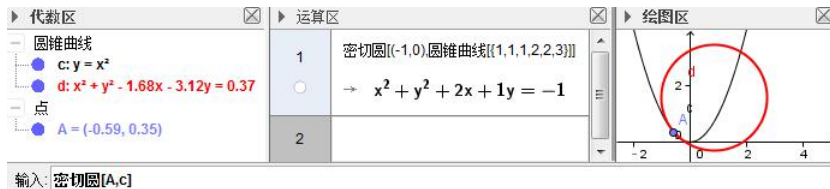
**案例:**“密切圆((0,0),  $x^2$ )”得出“ $x^2+y^2-y=0$ ”; “密切圆((1,0), 曲线( $\cos(t), \sin(2t), t, 0, 2\pi$ ))”得出“ $x^2+y^2+6x=7$ ”; “密切圆((-1,0), 圆锥曲线({1, 1, 1, 2, 2, 3}))”得出“ $x^2+y^2+2x+1y=-1$ ”。



案例：已知“A=(1.54, 1)”为“y=sin(x)”曲线上一点，“密切圆(A, sin(x))”得出“c:x<sup>2</sup>+y<sup>2</sup>-3.142x=-1.467”。



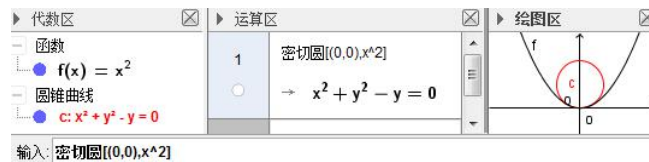
案例：圆锥曲线“c:y=x<sup>2</sup>”，点A为其上一点，“密切圆(A, c)”得到圆锥曲线的密切圆。



OsculatingCircle(<Point>, <Function>); 密切圆(<点>, <函数>)。

得出函数在指定点上的密切圆。

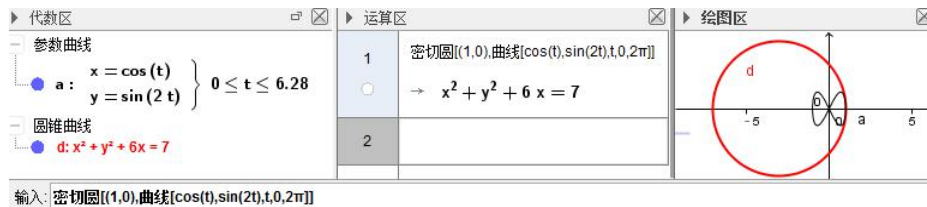
案例：“密切圆((0, 0), x<sup>2</sup>)”得出“x<sup>2</sup>+y<sup>2</sup>-y=0”。



OsculatingCircle(<Point>, <Curve>); 密切圆(<点>, <曲线>)。

得出曲线在指定点上的密切圆。

案例：“密切圆((1, 0), 曲线(cos(t), sin(2t), t, 0, 2π))”得出“x<sup>2</sup>+y<sup>2</sup>+6x=7”。



注：自 GeoGebra5，这个指令也适于圆锥曲线。

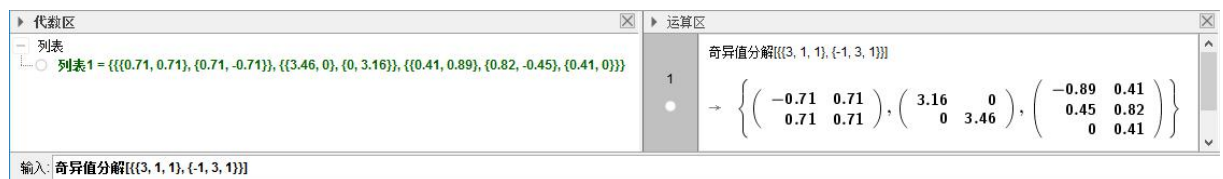
释义：给定一个曲线及其上一点 P，与曲线在该点相外切的圆中半径最小，或与曲线在该点相内切的圆中半径最大的圆，称为曲线的密切圆(Osculatingcircle)或曲率圆。曲线在 P 点的曲率，便是密切圆半径的倒数。

注：“密切圆(<点>, <对象>)”指令可以取代另外两个指令。

## 2.7.29 SVD. 奇异值分解

SVD(<Matrix>); 奇异值分解(<矩阵>)。

返回矩阵的分解值（形如 3 阶矩阵）。



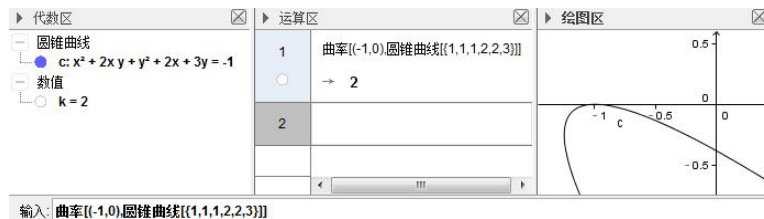
案例：“奇异值分解( $\{\{3, 1, 1\}, \{-1, 3, 1\}\}$ )”得出“ $\left\{ \begin{pmatrix} -0.71 & 0.71 \\ 0.71 & 0.71 \end{pmatrix}, \begin{pmatrix} 3.16 & 0 \\ 0 & 3.46 \end{pmatrix}, \begin{pmatrix} -0.89 & 0.41 \\ 0.45 & 0.82 \\ 0 & 0.41 \end{pmatrix} \right\}$ ”

### 2.7.30 Curvature. 曲率

Curvature( $\langle \text{Point} \rangle, \langle \text{Object} \rangle$ ); 曲率( $\langle \text{点} \rangle, \langle \text{对象} \rangle$ )。

得出对象(函数, 曲线, 圆锥曲线)在给定点上的曲率。

案例：“曲率( $(0, 0), x^2$ )”得出2; “曲率( $(0, 0)$ , 曲线 $(\cos(t), \sin(2t), t, 0, \pi)$ )”得出0; “曲率( $(-1, 0)$ , 圆锥曲线 $(\{1, 1, 1, 2, 2, 3\})$ )”得出2。



Curvature( $\langle \text{Point} \rangle, \langle \text{Function} \rangle$ ); 曲率( $\langle \text{点} \rangle, \langle \text{函数} \rangle$ )。

计算函数在给定点处的曲率。

案例：“曲率( $(0, 0), x^2$ )”得出2。



Curvature( $\langle \text{Point} \rangle, \langle \text{Curve} \rangle$ ); 曲率( $\langle \text{点} \rangle, \langle \text{曲线} \rangle$ )。

计算曲线在给定点处的曲率。

案例：“曲率( $(0, 0)$ , 曲线 $(\cos(t), \sin(2t), t, 0, \pi)$ )”得出0。

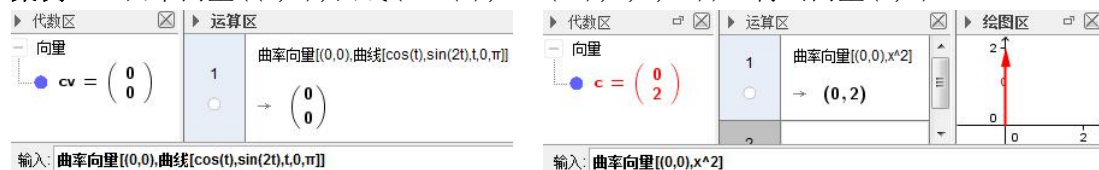
注：“Curvature( $\langle \text{Point} \rangle, \langle \text{Object} \rangle$ ); 曲率( $\langle \text{点} \rangle, \langle \text{对象} \rangle$ )”指令取代了另外两个指令。

### 2.7.31 CurvatureVector. 曲率向量

CurvatureVector( $\langle \text{Point} \rangle, \langle \text{Curve} \rangle$ ); 曲率向量( $\langle \text{点} \rangle, \langle \text{曲线} \rangle$ )。

得出曲线在给定点处的曲率向量。

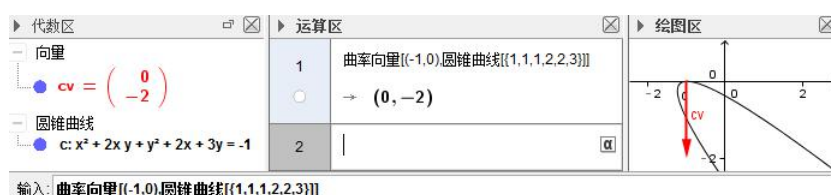
案例：“曲率向量( $(0, 0)$ , 曲线 $(\cos(t), \sin(2t), t, 0, \pi)$ )”得出向量 $(0, 0)$ 。



CurvatureVector( $\langle \text{Point} \rangle, \langle \text{Object} \rangle$ ); 曲率向量( $\langle \text{点} \rangle, \langle \text{曲线对象} \rangle$ )。

得出对象(函数、曲线、圆锥曲线)在给定点处的曲率向量。

案例：“曲率向量( $(0, 0), x^2$ )”得出向量 $(0, 2)$ ; “曲率向量( $(0, 0)$ , 曲线 $(\cos(t), \sin(2t), t, 0, \pi)$ )”得出向量 $(0, 0)$ ; “曲率向量( $(-1, 0)$ , 圆锥曲线 $(\{1, 1, 1, 2, 2, 3\})$ )”得出向量 $(0, -2)$ 。

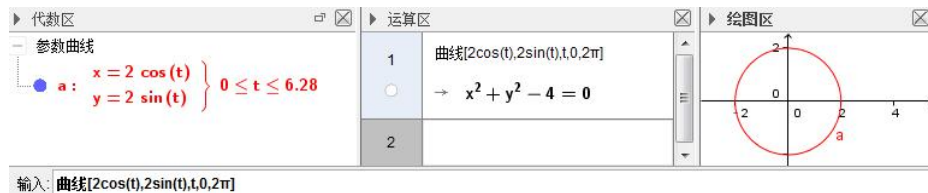


## 2.7.32 Curve (CurveCartesian). 曲线

Curve(<Expression>, <Expression>, <Parameter Variable>, <Start Value>, <End Value>);  
曲线(<x(t)>, <y(t)>, <参数值 t>, <t-起始值>, <t-终止值>).

得出给定使用参数的 x 表达式 (第一个表达式) 和 y 表达式 (第二个表达式) 在给定区间 (起始值, 终止值) 内的笛卡尔参数曲线。

案例: “曲线(2cos(t), 2sin(t), t, 0, 2π)” 创建一个以坐标系原点为圆心、半径为 2 的圆。



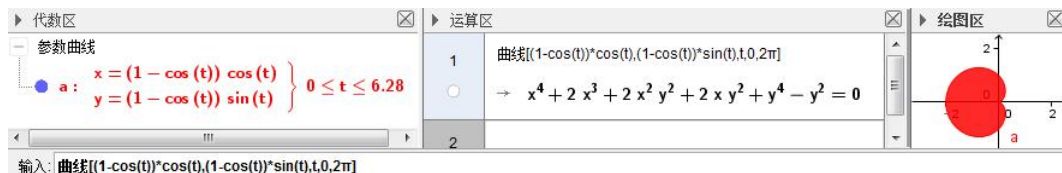
Curve(<Expression>, <Expression>, <Expression>, <Parameter Variable>, <Start Value>, <End Value>); 曲线(<x(t)>, <y(t)>, <z(t)>, <参数值 t>, <t-起始值>, <t-终止值>).

得出给定使用参数的 x 表达式 (第一个表达式), y 表达式 (第二个表达式) 和 z 表达式 (第三个表达式) 在给定区间 (起始值, 终止值) 内的笛卡尔参数曲线。

案例: “曲线(cos(t), sin(t), t, 0, 10π)” 创建一个 3D 螺旋。

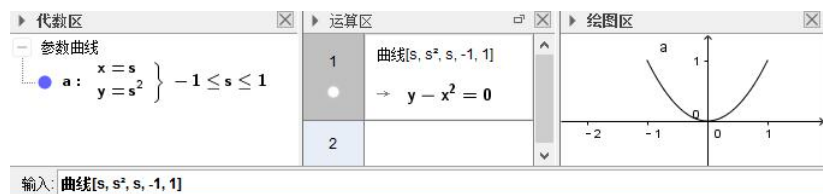


案例: “曲线((1-cos(t))\*cos(t), (1-cos(t))\*sin(t), t, 0, 2π)” 生成心形曲线。



注: 参数名称除了 t 以外, 可以使用其他符号。

案例: “曲线(s, s^2, s, -1, 1)” 得出区间内函数, 如下图结果。



注: <t-终止值>必须大于或等于<t-起始值>而且都必须都是有限值。

注: x、y 和 z 不允许作为参数变量。

注: 更多细节参见“曲线”集合对象。参见“导数”指令和“参数导数”指令。

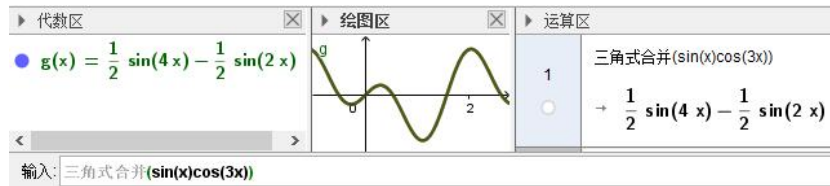
## 2.7.33 TrigCombine. 三角式合并

TrigCombine(<Expression>); 三角式合并(<表达式>).

将一个包括三角函数乘积的三角函数表达式, 转化为一个只包含变量之和, 没有三角函数乘积

的三角函数表达式。

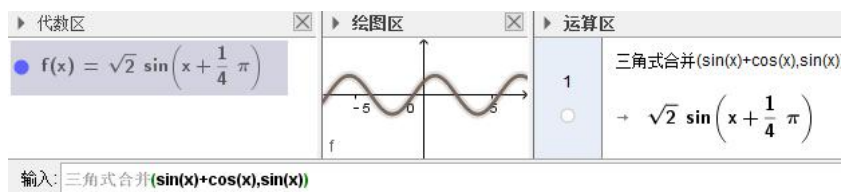
案例：“三角式合并( $\sin(x)\cos(3x)$ )”给出“ $f(x) = \frac{1}{2}\sin(x) - \frac{1}{2}\sin(2x)$ ”。



**TrigCombine**(<Expression>, <Target Function>); 三角式合并(<表达式>, <目标函数>).

将一个包括三角函数乘积的三角函数表达式，转化为一个只包含变量之和的没有三角函数乘积的三角函数表达式，其偏向于指定的目标函数。

案例：“三角式合并( $\sin(x)+\cos(x), \sin(x)$ )”给出“ $f(x) = \sqrt{2}\sin(x + \frac{1}{4}\pi)$ ”。



**CAS Syntax (运算区语法)**

**TrigCombine**(<Expression>); 三角式合并(<表达式>).

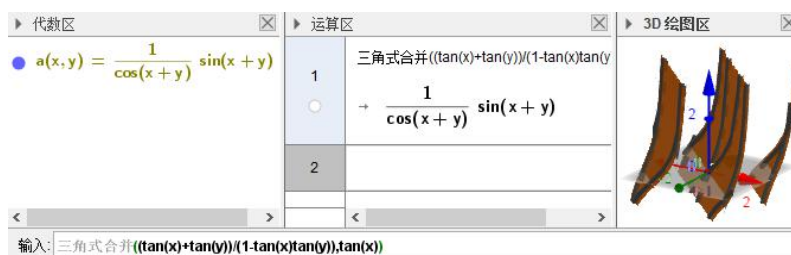
将一个包括三角函数乘积的三角函数表达式，转化为一个只包含变量之和没有三角函数乘积的三角函数表达式。

案例：“三角式合并( $\sin(x)\cos(3x)$ )”给出“ $f(x) = \frac{1}{2}\sin(x) - \frac{1}{2}\sin(2x)$ ”。

**TrigCombine**(<Expression>, <Target Function>); 三角式合并(<表达式>, <目标函数>).

将一个包括三角函数乘积的三角函数表达式，转化为一个只包含变量之和的没有三角函数乘积的三角函数表达式，其偏向于指定的目标函数。

案例：“三角式合并( $\sin(x)+\cos(x), \sin(x)$ )”给出“ $f(x) = \sqrt{2}\sin(x + \frac{1}{4}\pi)$ ”。



案例：“三角式合并( $(\tan(x)+\tan(y))/(1-\tan(x)\tan(y)), \tan(x)$ )”给出“ $\tan(x+y)$ ”（或者等量形式）。

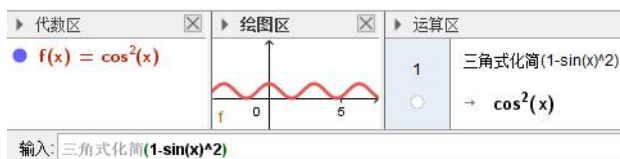
注：参见“三角式展开”指令和“三角式化简”指令。“三角式合并”的分类不是十分苛刻，诸如“两角和的正弦公式”等也被列入其中。

## 2.7.34 TrigSimplify. 三角式化简

**TrigSimplify**(<Expression>); 三角式化简(<表达式>).

化简指定的三角函数表达式。

案例：“三角式化简(1-sin(x)^2)”给出“f(x)=cos^2(x)”。



“三角式化简(sin(x)^2-cos(x)^2+1)”给出“f(x)=2sin^2(x)”。

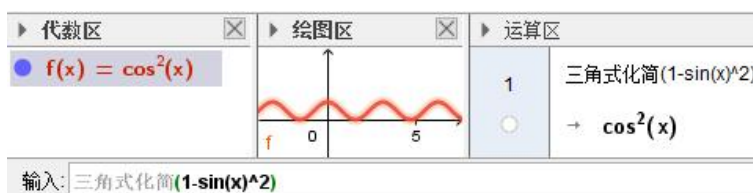


CAS Syntax (运算区语法)

TrigSimplify(<Expression>); 三角式化简(<表达式>).

化简指定的三角函数表达式。

案例：“三角式化简(1-sin(x)^2)”给出“cos^2(x)”。



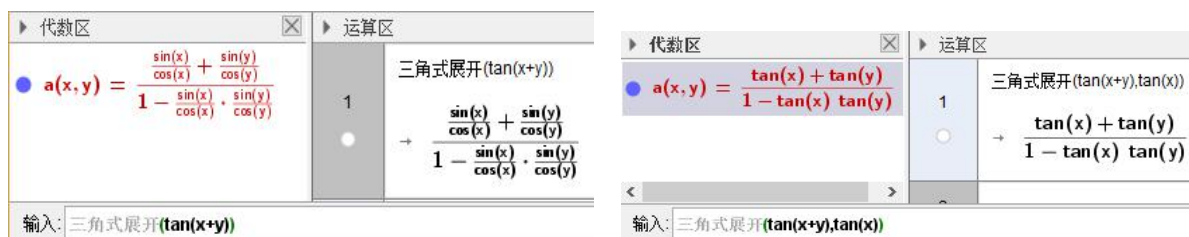
注：在运算区，这个可对任意变量 x 工作。参见“三角式展开”指令和“三角式合并”指令。

## 2.7.35 TrigExpand. 三角式展开

TrigExpand(<Expression>); 三角式展开(<表达式>).

将一个三角函数表达式转化为一个三角函数内只有单变量作为参数的表达式。

案例：“三角式展开(tan(x+y))”给出“a(x,y) =  $\frac{\sin(x) + \sin(y)}{\cos(x) \cos(y)}$ ”。



TrigExpand(<Expression>, <Target Function>); 三角式展开(<表达式>, <目标函数>).

将一个三角函数表达式转化为一个三角函数内只有单变量作为参数的表达式，其偏向于指定的目标函数。

案例：“三角式展开(tan(x+y), tan(x))”给出“a(x,y) =  $\frac{\tan(x) + \tan(y)}{1 - \tan(x) \tan(y)}$ ”。

CAS Syntax (运算区语法)

TrigExpand(<Expression>); 三角式展开(<表达式>);

将一个三角函数表达式转化为一个三角函数内只有单变量作为参数的表达式。

案例：“三角式展开( $\tan(x+y)$ )”给出“ $a(x, y) = \frac{\frac{\sin(x)}{\cos(x)} + \frac{\sin(y)}{\cos(y)}}{1 - \frac{\sin(x)}{\cos(x)} \cdot \frac{\sin(y)}{\cos(y)}}$ ”。

**TrigExpand(<Expression>, <Target Function>); 三角式展开(<表达式>, <目标函数>)**。

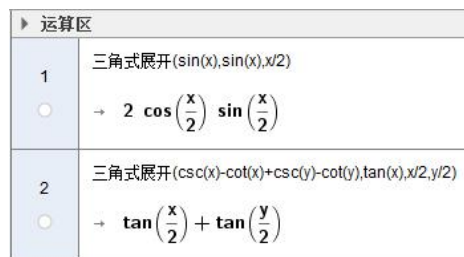
将一个三角函数表达式转化为一个三角函数内只有单变量作为参数的表达式，其偏向于指定的目标函数。

案例：“三角式展开( $\tan(x+y), \tan(x)$ )”给出“ $a(x, y) = \frac{\tan(x) + \tan(y)}{1 - \tan(x)\tan(y)}$ ”。

**TrigExpand(<Expression>, <Target Function>, <Target Variable>)三角式展开(<表达式>, <目标函数>, <目标变量>)**。

将一个三角函数表达式转化为一个三角函数内只有单变量作为参数的表达式，其偏向于指定的目标函数与目标变量。

案例：“三角式展开( $\sin(x), \sin(x), x/2$ )”给出  $2\cos(\frac{x}{2})\sin(\frac{x}{2})$ 。



**TrigExpand(<Expression>, <Target Function>, <Target Variable>, <Target Variable>); 三角式展开(<表达式>, <目标函数>, <目标变量 1>, <目标变量 2>)**。

将一个三角函数表达式转化为一个三角函数内只有单变量作为参数的表达式，其偏向于指定的目标函数与目标变量。

案例：“三角式展开( $\csc(x)-\cot(x)+\csc(y)-\cot(y), \tan(x), x/2, y/2$ )”给出“ $\tan(\frac{x}{2}) + \tan(\frac{y}{2})$ ”。

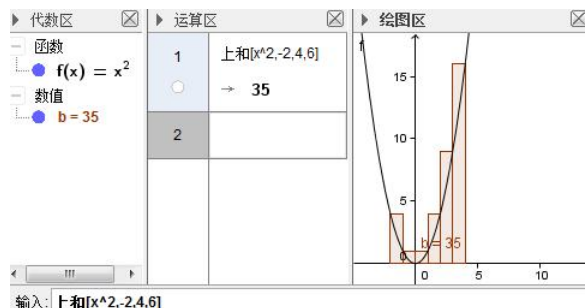
注：参见“三角式化简”指令和“三角式合并”指令。

### 2.7.36 UpperSum. 上和

**UpperSum(<Function>, <Start x-Value>, <End x-Value>, <Number of Rectangles>); 上和(<函数>, <x-起始值>, <x-终止值>, <矩形数量>)**。

用 n 个矩形计算指定函数在区间(x-起始值, x-终止值)内的上和。

案例：上和( $x^2, -2, 4, 6$ )得出 35。



注：这个指令绘制“上和”矩形并给出和的值。参见“下和”、“左和”、“矩形法则”和“梯



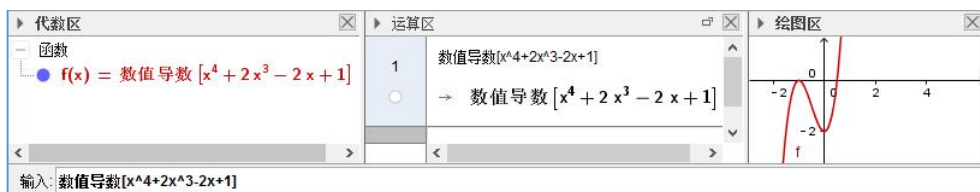
形法则”指令。

## 2.7.37 NDerivative. 数值导数

**NDerivative**(<Function>); 数值导数(<函数>)。

用数值计算得出给定函数的 1 阶倒数。

**案例:** “数值导数( $x^4+2x^3-2x+1$ )”在绘图区绘制出函数“ $f(x)=4x^3+6x^2-2$ ”的函数图象,但函数  $f(x)$  的方程并不在代数区列出。

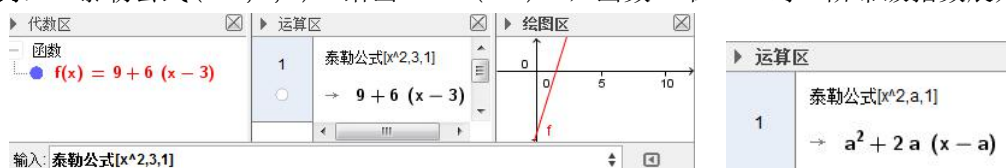


## 2.7.38 TaylorPolynomial. 泰勒公式

**TaylorPolynomial**(<Function>, <x-Value>, <Order Number>); 泰勒公式(<函数>, <x 值>, <阶数>)。

创建按指定阶数在“x-值”时给定函数的幂级数展开式。

**案例:** “泰勒公式( $x^2, 3, 1$ )”给出“ $9+6(x-3)$ ”, 函数  $x^2$  在  $x=3$  时一阶幂级数展开。



**CAS Syntax (运算区语法)**

**TaylorPolynomial**(<Expression>, <x-Value>, <Order Number>); 泰勒公式(<表达式>, <x 值>, <阶数>)。

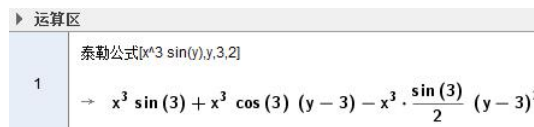
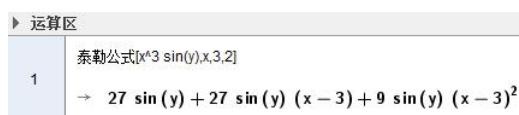
创建按指定阶数在 x-值时给定表达式的幂级数展开式。

**案例:** “泰勒公式( $x^2, a, 1$ )”给出“ $a^2+2a(x-a)$ ”, 表达式  $x^2$  在  $x=3$  时一阶幂级数展开。

**TaylorPolynomial**(<Expression>, <Variable>, <Variable Value>, <Order Number>); 泰勒公式(<表达式>, <变量>, <变量值>, <阶数>)。

创建关于指定变量按指定阶数在 x-值时给定表达式的幂级数展开式。

**案例:** “泰勒公式( $x^3 \sin(y), x, 3, 2$ )”给出“ $27\sin(y)+27\sin(y)(x-3)+9\sin(y)(x-3)^2$ ”, 表达式  $x^3 \sin(y)$  关于  $x$  在  $x=3$  时的 2 阶幂级数展开式。



“泰勒公式( $x^3 \sin(y), y, 3, 2$ )”给出“ $\sin(3)x^3+\cos(3)x^3(y-3)-x^3 \cdot \frac{\sin(3)}{2}(y-3)^2$ ”, 表达式  $x^3 \sin(y)$  关于  $y$  在  $y=3$  时的 2 阶幂级数展开式。

**注:** 阶数必须为一个大于或等于零的整数。

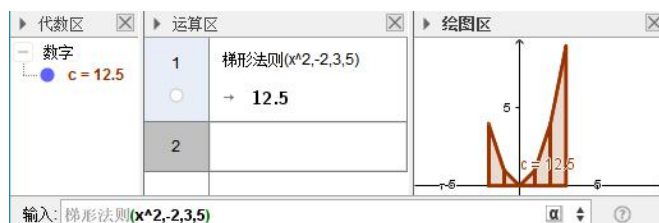
## 2.7.39 TrapezoidalSum. 梯形法则

这个指令在不同英语变型中有不同的拼写：TrapezoidalSum(US)、TrapeziumSum(UK+Aus)。

TrapezoidalSum(<Function>, <Start x-Value>, <End x-Value>, <Number of Trapezoids>);  
 梯形法则(<函数>, <x-起始值>, <x-终止值>, <梯形数量>)。

用  $n$  个梯形计算函数在区间 ( $x$ -起始值,  $x$ -终止值) 内的梯形 (面积) 和。

案例：“梯形法则( $x^2$ , -2, 3, 5)” 得出 12.5。



注：这个指令绘制“梯形法则”的梯形并给出和的值。参见“下和”、“左和”、“矩形法则”、“上和”指令。

## 2.7.40 Coefficients. 系数列表

Coefficients(<Polynomial>); 系数列表(<多项式>)。

得出多项式  $(a_k x^k + a_{k-1} x^{k-1} + \dots + a_1 x + a_0)$  的所有系数组成的列表。

案例：“系数列表( $x^3 - 3x^2 + 3x$ )” 得出 “{1, -3, 3, 0}”，多项式  $x^3 - 3x^2 + 3x + 0$  的系数列表。



Coefficients(<Conic>); 系数列表(<圆锥曲线>)。

返回基本格式的圆锥曲线 “ $ax^2 + by^2 + c + dxy + ex + fy = 0$ ” 系数列表  $\{a, b, c, d, e, f\}$ 。

注：给定 (隐式直线) “ $a: 3x + 2y - 2 = 0$ ”： $x(a)$  返回 3,  $y(a)$  返回 2, 以及  $z(a)$  返回 -2。对于由 “ $1: ax + by + c = 0$ ” 所代表一条隐式直线, 可以使用语句  $x(a)$ 、 $y(a)$ 、 $z(a)$  来获得它的系数。

案例：给定直线 “ $1: 3x + 2y - 2 = 0$ ”：“ $x(a)$ ” 返回 3, “ $y(a)$ ” 返回 2, “ $z(a)$ ” 返回 -2。

CAS Syntax (运算区语法)

Coefficients(<Polynomial>); 系数列表(<多项式>)。

得出多项式主变量的所有系数列表。

案例：“系数列表( $x^3 - 3x^2 + 3x$ )” 得出  $\{1, -3, 3, 0\}$ , 多项式 “ $x^3 - 3x^2 + 3x + 0$ ” 的系数列表。

Coefficients(<Polynomial>, <Variable>); 系数列表(<多项式>, <变量>)。

得出多项式给出变量的所有系数列表。

案例：“系数列表( $a^3 - 3a^2 + 3a$ ,  $a$ )” 得出 “ $\{1, -3, 3, 0\}$ ”, 多项式 “ $a^3 - 3a^2 + 3a$ ” 的系数列表;  
 “系数列表( $a^3 - 3a^2 + 3a$ ,  $x$ )” 得出 “ $a^3 - 3a^2 + 3a$ ”。

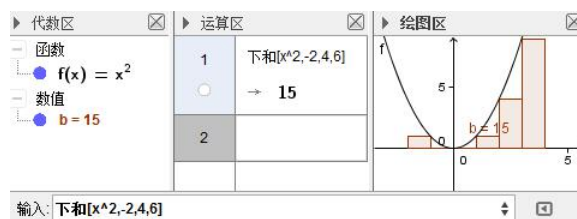


## 2.7.41 LowerSum. 下和

LowerSum(<Function>, <Start x-Value>, <End x-Value>, <Number of Rectangles>); 下和(<函数>, <x-起始值>, <x-终止值>, <矩形数量>)。

用  $n$  个矩形计算指定函数在区间 ( $x$ -起始值,  $x$ -终止值) 内的下和。

案例：“下和( $x^2$ , -2, 4, 6)” 得出 15。



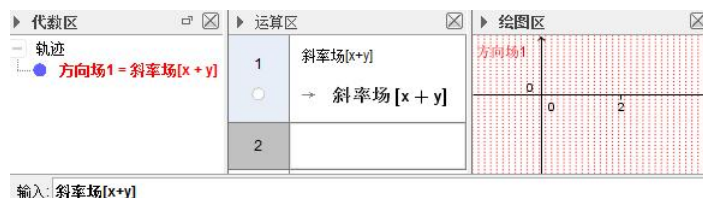
注：这个指令同样会画出“下和”的矩形组块并给出和的值。参见“上和”、“左和”、“矩形法则”与“梯形法则”指令。

## 2.7.42 SlopeField. 斜率场

SlopeField( $\langle f(x, y) \rangle$ ); 斜率场( $\langle f(x, y) \rangle$ )。

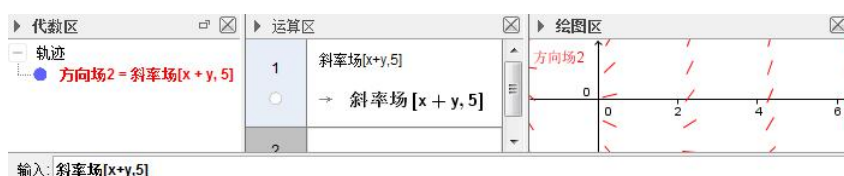
绘制微分方程  $\frac{dy}{dx} = f(x, y)$  的斜率场。

案例：“斜率场( $x+y$ )” 绘制斜率场。



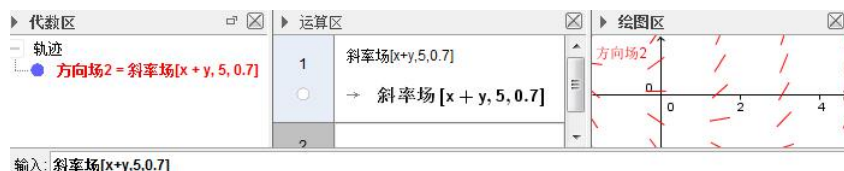
SlopeField( $\langle f(x, y) \rangle$ ,  $\langle \text{Number } n \rangle$ ); 斜率场( $\langle f(x, y) \rangle$ ,  $\langle \text{数值 } n \rangle$ )。

绘制微分方程  $\frac{dy}{dx} = f(x, y)$  的斜率场，用单边  $n$  个 ( $n \times n$ ) 网格（当绘图区视图为正方形）绘制，否则就用更小的网格。默认单边为 40 ( $40 \times 40$  个网格)。



SlopeField( $\langle f(x, y) \rangle$ ,  $\langle \text{Number } n \rangle$ ,  $\langle \text{Length Multiplier } a \rangle$ ); 斜率场( $\langle f(x, y) \rangle$ ,  $\langle \text{数值 } n \rangle$ ,  $\langle \text{长度倍增器 } a \rangle$ )。

绘制微分方程  $\frac{dy}{dx} = f(x, y)$  的斜率场。长度倍增  $0 < a \leq 1$  决定了每格中的线段长。



SlopeField( $\langle f(x, y) \rangle$ ,  $\langle \text{Number } n \rangle$ ,  $\langle \text{Length Multiplier } a \rangle$ ,  $\langle \text{Min } x \rangle$ ,  $\langle \text{Min } y \rangle$ ,  $\langle \text{Max } x \rangle$ ,  $\langle \text{Max } y \rangle$ ); 斜率场( $\langle f(x, y) \rangle$ ,  $\langle \text{数值 } n \rangle$ ,  $\langle \text{长度倍增器 } a \rangle$ ,  $\langle x \text{ 最小} \rangle$ ,  $\langle y \text{ 最小} \rangle$ ,  $\langle x \text{ 最大} \rangle$ ,  $\langle y \text{ 最大} \rangle$ )。

在指定的矩形范围内绘制微分方程  $\frac{dy}{dx} = f(x, y)$  的斜率场（而不是填充绘图区视图）。



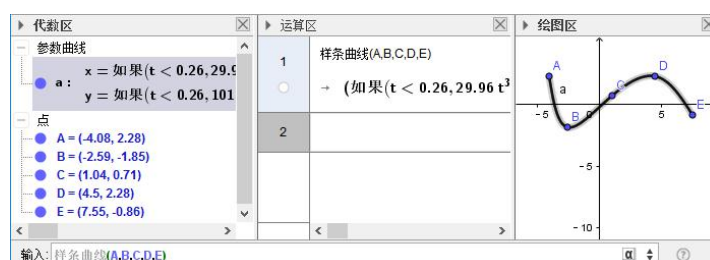
注：使用“移动视窗”、“放大”和“缩小”工具，观察变化。

注：参见“解常微分方程”、“轨迹”和“积分”指令。

### 2.7.43 Spline. 样条曲线

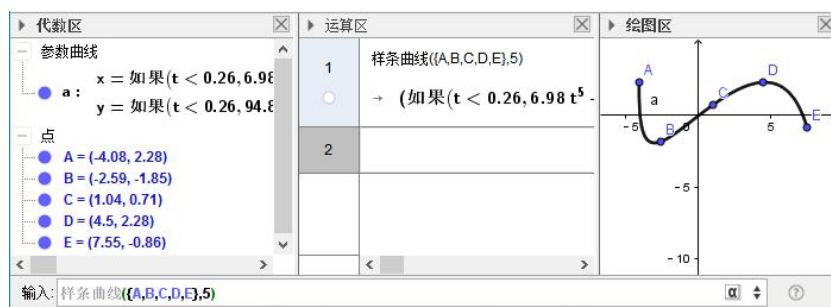
**Spline(<List of Points>); 样条曲线(<点列>)。**

创建穿过所有点的三次样条。



**Spline(<List of Points>, <Order  $\geq 3$ >); 样条曲线(<点列>, <次数  $\geq 3$ >)。**

创建按给定次数穿过所有点的样条曲线。



注：次数决定了曲线的光滑程度。

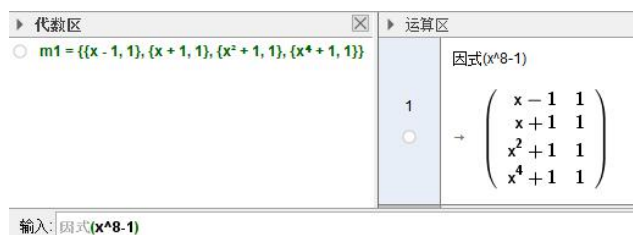
释义：样条的结果是一条曲线。样条算法分别利用 x 和 y 的坐标值：首先确定对应点的 t 值（通过点间的几何距离），然后使用 t 趋向 x 和 t 趋向 y 逼近找到三次样条。

### 2.7.44 Factors. 因式

**Factors(<Polynomial>); 因式(<多项式>)。**

给出形式为{因式,指数}的列表列表，这些因式按相应指数升幂后的乘积等于所给的多项式。因式按次方程度升序排列。

案例：“因式( $x^8-1$ )”得出  $\{\{x^4+1, 1\}, \{x^2+1, 1\}, \{x+1, 1\}, \{x-1, 1\}\}$ 。

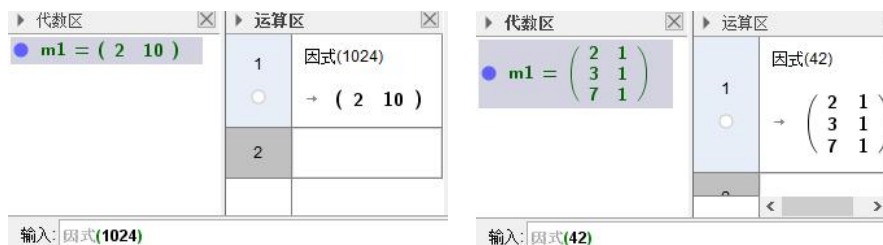


注：并非所有的因式在实数上不可通约。

### Factors(<Number>); 因式(<数值>).

给出形式为{质数,指数}的列表列表, 这些质数按相应指数升幂后的乘积等于所给的数值。质数按升序排列。

案例: “因式(1024)” 得出 “{{2,10}}”, 因  $1024=2^{10}$ ; “因式(42)” 得出 “{{2,1},{3,1},{7,1}}”, 因  $42=2^1 \cdot 3^1 \cdot 7^1$ 。



注: 参见“质因数”指令和“因式分解”指令。

### CAS Syntax (运算区语法)

### Factors(<Polynomial>); 因式(<多项式>).

给出形式为{因式,指数}的列表列表, 这些因式按相应指数升幂后的乘积等于所给的多项式。因式按次方程度降序排列。

案例: “因式( $x^8-1$ )” 得出  $\{x^4+1, 1\}, \{x^2+1, 1\}, \{x+1, 1\}, \{x-1, 1\}$ , 列放如图。

注: 并非所有的因式在实数上不可通约。

### Factors(<Number>); 因式(<数值>).

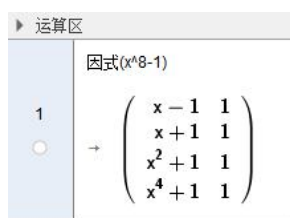
给出形式为{质数,指数}的列表列表, 这些质数按相应指数升幂后的乘积等于所给的数值。质数按升序排列。

案例: “因式(1024)” 得出 “{{2,10}}”, 以(2 10)形式列放。因  $1024=2^{10}$ ; “因式(42)” 得出 “{{2,1},{3,1},{7,1}}”, 如图列放。因  $42=2^1 \cdot 3^1 \cdot 7^1$ 。

注: 参见“质因数”指令和“因式分解”指令。

注: 在运算区, 可以输入未定义的变量, 也能输出正确的矩阵。

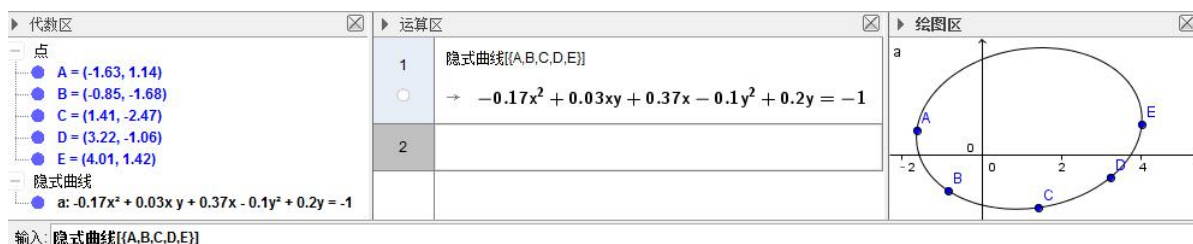
案例: “因式( $a^8-1$ )” 得到如图结果。



## 2.7.45 ImplicitCurve. 隐式曲线

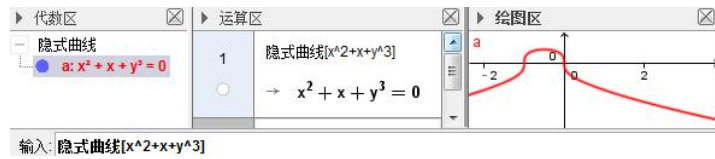
### ImplicitCurve(<List of Points>); 隐式曲线(<点列>).

创建通过由点集所指定点的隐式曲线。列表的长度必须是  $n(n+3)/2$ , 以得出次方阶为  $n$  的隐式曲线。



### ImplicitCurve(<f(x,y)>); 隐式曲线(<f(x,y)>)

创建隐式曲线  $f(x,y)=0$ 。当前的  $f(x,y)$  必须是一个  $x$  和  $y$  的多项式。

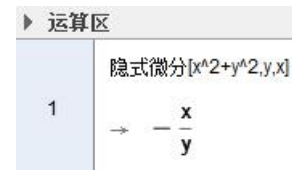


## 2.7.46 ImplicitDerivative. 隐式微分

**ImplicitDerivative(<f(x,y)>); 隐式微分(<f(x,y)>)**。

给出指定表达式的隐式微分。

案例：“隐式微分(x+2y)”得出-0.5。



**CAS Syntax (运算区语法)**

**ImplicitDerivative(<f(x,y)>); 隐式微分(<f(x,y)>)**。

给出指定表达式的隐式微分。

案例：“隐式微分(x+2y)”得出  $-\frac{1}{2}$ 。

**ImplicitDerivative(<Expression>,<Dependent Variable>,<Independent Variable>); 隐式微分(<表达式>,<因变量>,<自变量>)**

给出指定表达式的隐式微分。

案例：“隐式微分(x^2+y^2,y,x)”得出  $-\frac{x}{y}$ 。

注：参见“微分”指令

## 2.7.47 LimitAbove. 右极限

**LimitAbove(<Function>,<Value>); 右极限(<函数>,<数值>)**。

计算函数从右侧无限接近指定的主函数变量值时的单侧极限。

案例：“右极限(1/x,0)”得出  $\infty$ 。

注：GeoGebra 不是所有的极限都能计算出来，这种情况下将返回未定义(错误的结果是未定义)。



**CAS Syntax (运算区语法)**

**LimitAbove(<Expression>,<Value>); 右极限(<表达式>,<数值>)**。

计算表达式从右侧无限接近指定的主函数变量值时的单侧极限。

案例：“右极限(1/x,0)”得出  $\infty$ 。

**LimitAbove(<Expression>,<Variable>,<Value>); 右极限(<表达式>,<变量>,<数值>)**。

计算表达式从右侧无限接近多元函数指定的主函数变量值时的单侧极限。

案例：“右极限(1/a,a,0)”得出  $\infty$ 。

注: GeoGebra 不是所有的极限都能计算出来, 这种情况下将返回未定义(错误的结果是未定义)。  
 注: 参见“极限”指令和“左极限”指令。

## 2.7.48 LeftSum. 左和

**LeftSum(<Function>,<Start x-Value>,<End x-Value>,<Number of Rectangles>); 左和(<函数>,<x-起始值>,<x-终止值>,<矩形数量>)**。

用  $n$  个矩形计算函数在区间内的左和(函数与每个矩形区段上边的交点在最左端)。

案例: “左和( $x^2+1,0,2,4$ )” 得出  $a=3.75$ 。

注: 这个指令同样会画出“左和”的矩形组块且给出和的值; 参见“右和”、“梯形法则”、“下和”与“上和”指令。



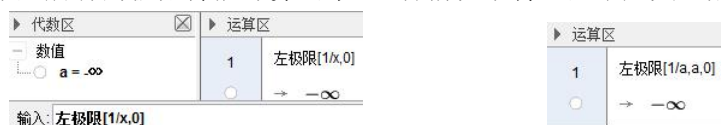
## 2.7.49 LimitBelow. 左极限

**LimitBelow(<Function>,<Value>); 左极限(<函数>,<数值>)**。

计算函数从左侧无限接近指定的主函数变量值时的单侧极限。

案例: “左极限( $1/x,0$ )” 得出 “ $-\infty$ ”。

注: GeoGebra 不是所有的极限都能计算出来, 这种情况下将返回未定义(错误的结果是未定义)。



**CAS Syntax (运算区语法)**

**LimitBelow(<Expression>,<Value>); 左极限(<表达式>,<数值>)**。

计算表达式从左侧无限接近指定的主函数变量值时的单侧极限。

案例: “左极限( $1/x,0$ )” 得出  $\infty$ 。

**LimitBelow(<Expression>,<Variable>,<Value>); 左极限(<表达式>,<变量>,<数值>)**。

计算表达式从左侧无限接近多元函数指定的主函数变量值时的单侧极限。

案例: “左极限( $1/a,a,0$ )” 得出 “ $-\infty$ ”。

注: GeoGebra 不是所有的极限都能计算出来, 这种情况下将返回未定义(错误的结果是未定义)。

注: 参见“极限”指令和“右极限”指令。

## 2.8 Geometry. 几何

### 2.8.1 Radius. 半径

**Radius(<Conic>); 半径(<圆>)**。

返回圆的半径。

案例: 返回圆 “ $c:(x-1)^2+(y-1)^2=9$ ” 的半径, 使用“半径(c)” 得出 “ $a=3$ ”; 返回圆公式的半径, “半径( $(x-2)^2+(y-2)^2=16$ )” 得出 “ $a=4$ ”。



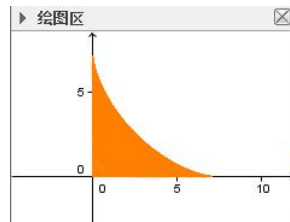
## 2.8.2 envelope. 包络

**Envelope(<Path>,<Point>); 包络(<路径>,<点>).**

当动点限定在其他对象上时，创建输出对象集的包络线方程。

包络线是这样的曲线，它与输出对象集中每个对象都在曲线上某点相切。

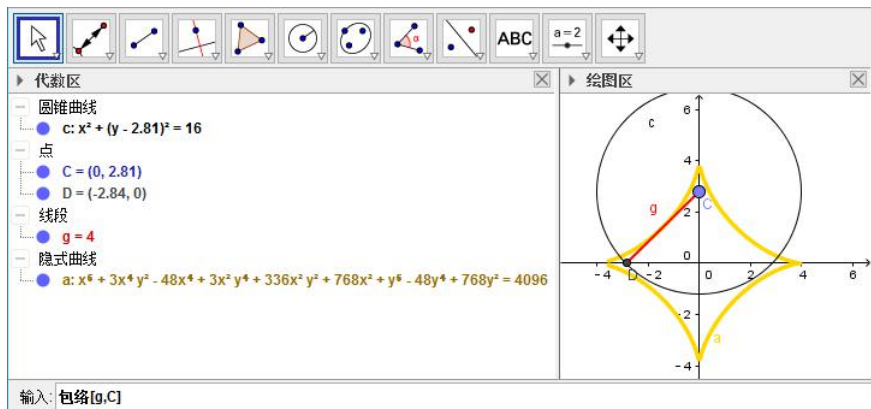
案例：一个梯子靠在墙上滑下来。



踪迹的轮廓线（凹曲线）就是梯子的包络线。严格地说，GeoGebra 把梯子看做线段且看做线段簇计算其包络线。只有可以被代数表达式描述的构件才可以创建包络线。

注：参见“轨迹”指令。

案例：如下图，点 C 为 y 轴上的动点，CD 定长 4，指令“包络(g,C)”得到包络线 a。不管红色线段在任何位置，红线都与曲线 a 相切与某点。



编者注：在最新的几个版本中，这个案例没有成功。

## 2.8.3 Difference. 差异

**Difference(<Polygon>,<Polygon>); 差异(<多边形 1>,<多边形 2>).**

找出两个多边形的不同。只有多边形不是自相交的多边形，此命令才有效。

## 2.8.4 Length. 长度

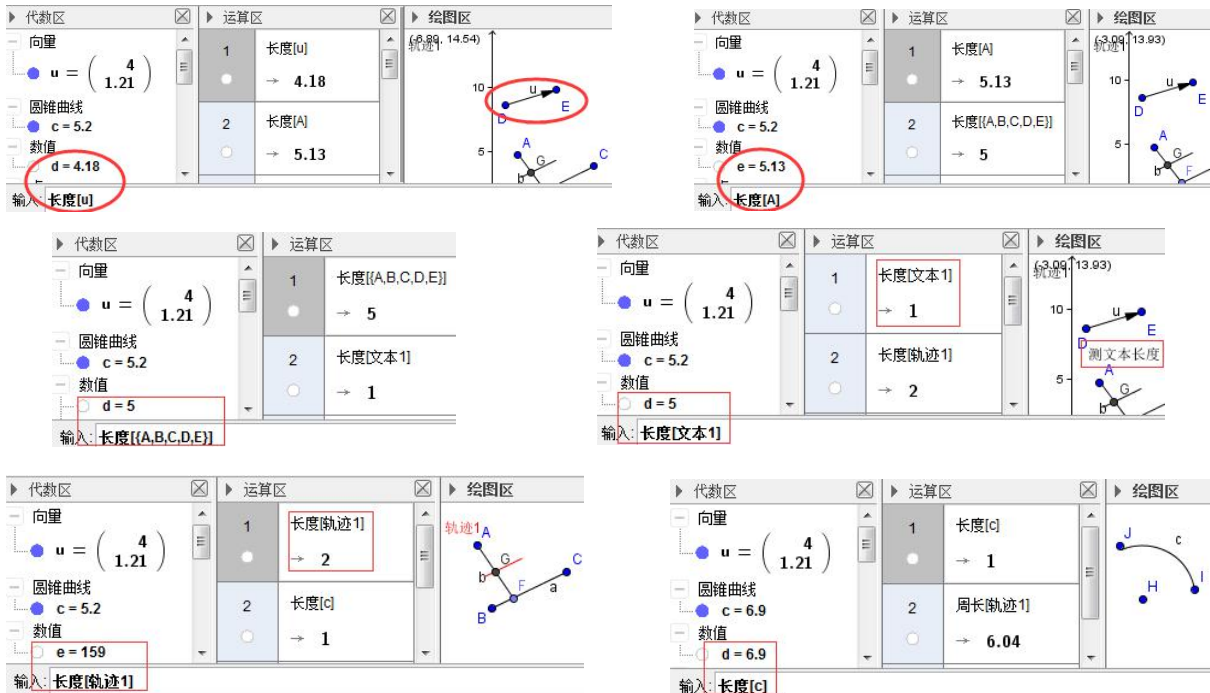
**Length(<Object>); 长度(<对象>).**

得出对象的长度。

案例：



- 1、“长度(<向量>)”得出向量的长度;
  - 2、“长度(<点>)”得出给出点位置向量长度;
  - 3、“长度(<列表>)”得出列表长度,即列表中元素的数量;
  - 4、“长度(<文本>)”得出文本字符数;
  - 5、“长度(<轨迹>)”返回构造给定轨迹使用的采样点数。使用“周长(轨迹)”得到轨迹的长度。
- 详见相关章节。
- 6、“长度(<弧>)”返回圆弧或扇形弧的长度(只有弯曲弧的部分)。



**Length(<Function>,<Start x-Value>,<End x-Value>); 长度(<函数>,<x-起始值>,<x-终止值>).**  
 得出函数图象在指定区间上的长度。

案例：“长度(2x,0,1)”返回 2.23606797749979,  $\sqrt{5}$  的近似值。



**Length(<Function>,<Start Point>,<End Point>); 长度(<函数>,<起始点>,<起始点>).**  
 得出函数图象在两点之间的长度。

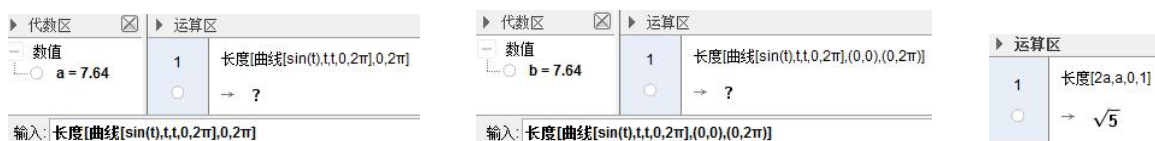
注：如果指定的点不在函数图象上,则用它们的 x 轴坐标值确定计算区间。

**Length(<Curve>,<Start t-Value>,<End t-Value>); 长度(<曲线>,<t-起始值>,<t-终止值>).**  
 得出两个参数值之间的曲线长度。

案例：“长度(曲线(sin(t),t,t,0,2π),0,2π)”=7.64。

**Length(<Curve>,<Start Point>,<End Point>); 长度(<曲线>,<起始点>,<终止点>).**  
 得出曲线上两点之间的长度。

案例：“长度(曲线(sin(t),t,t,0,2π),(0,0),(0,2π))”=7.64。



### CAS Syntax (运算区语法)

**Length(<Function>,<x-start>,<x-end>);** 长度(<函数>,<x-初始值>,<x-终止值>).

得出函数图象在两点之间的长度。

案例：“长度(2x,0,1)”返回 2.23606797749979,  $\sqrt{5}$  的近似值。

**Length(<Function>,<Variable>,<Start Point>,<End Point>);** 长度(<函数>,<变量>,<起始点>,<终止点>).

计算函数图象从起点到终点的长度。

案例：“长度(2a,a,0,1)”= $\sqrt{5}$ 。

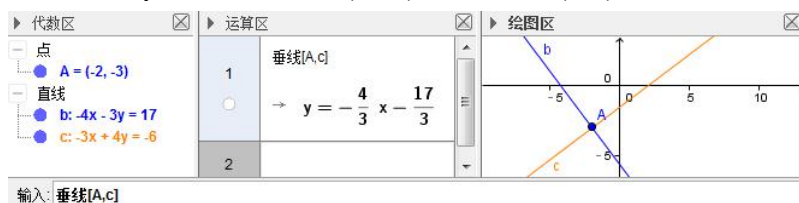
注：参见  “距离/长度”工具。

## 2.8.5 PerpendicularLine (OrthogonalLine) . 垂线

**PerpendicularLine(<Point>,<Line>);** 垂线(<点>,<直线>).

创建一条过指定点并垂直于指定直线的垂线。

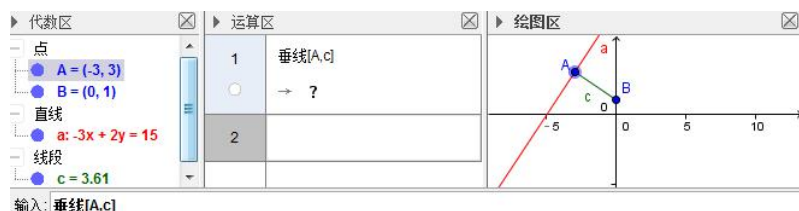
案例：设直线 “c:-3x+4y=-6” 和点 “A=(-2,-3)”，“垂线(A,c)” 得出直线 “d:-4x-3y=17。”



**PerpendicularLine(<Point>,<Segment>);** 垂线(<点>,<线段>).

创建一条过指定点并垂直于指定线段的垂线。

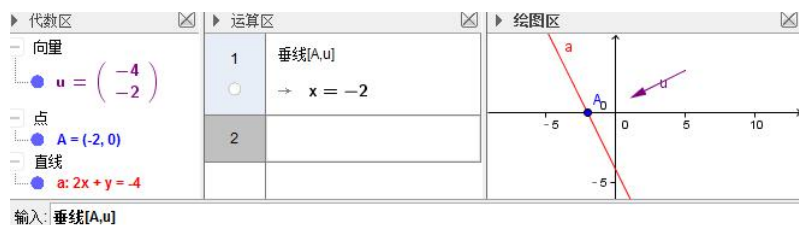
案例：设 c 是两个点 “A=(-3,3)” 与 “B=(0,1)” 之间的线段，“垂线(A,c)” 得出直线 “d:-3x+2y=15”。




**PerpendicularLine(<Point>,<Vector>);** 垂线(<点>,<向量>).

创建一条过指定点并垂直于指定向量的垂线。

案例：设 u 是一个两点间向量：“u=向量((5,3),(1,1))” 且设点 “A=(-2,0)”，“垂线(A,u)” 得出 “直线 c:2x+y=-4”。



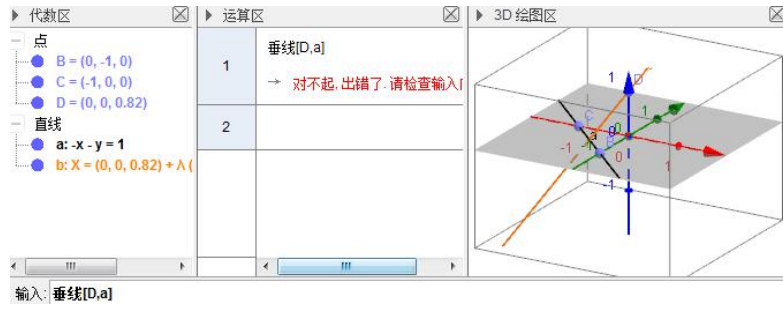
注：参见  “垂线”工具。

注：自 GeoGebra 5，这个命令适于 3D 对象。

**PerpendicularLine(<Point>,<Line>);** 垂线(<点>,<直线>).

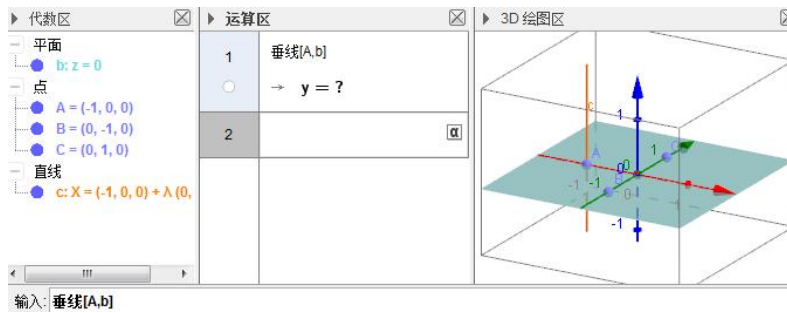
创建一条过指定点并垂直于指定线段的垂线。

注：如果在 3D 中，点在直线上，则这个指令会得出未定义。



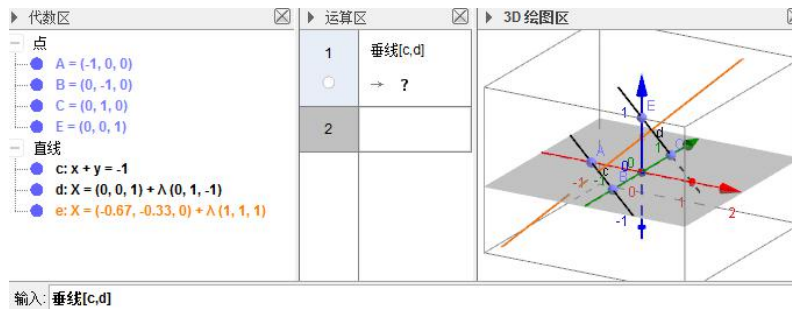
**PerpendicularLine(<Point>,<Plane>); 垂线(<点>,<平面>).**

创建一条过指定点并垂直于指定平面的垂线。



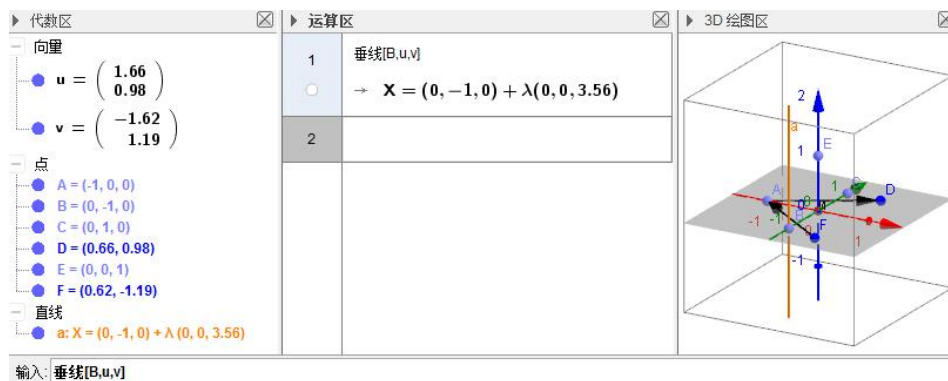
**PerpendicularLine(<Line>,<Line>); 垂线(<直线 1>,<直线 2>).**

创建两条给定直线的公垂线。



**PerpendicularLine(<Point>,<Direction>,<Direction>); 垂线(<点>,<向量 1>,<向量 2>).**

创建过给定点垂直给定方向（可以是直线或向量）的直线。

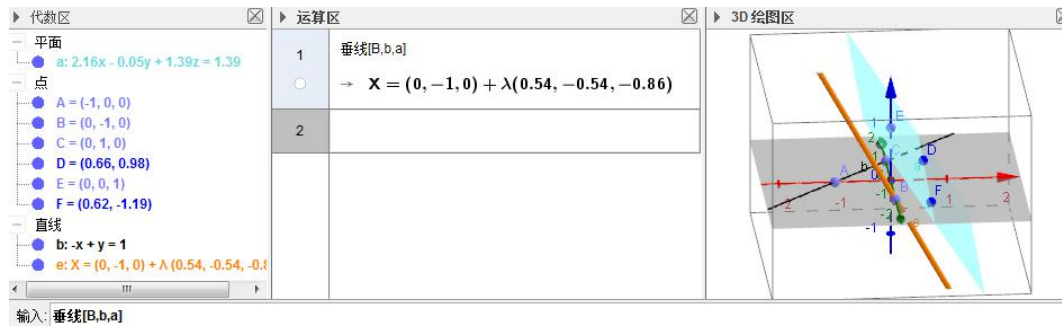


**PerpendicularLine(<Point>,<Line>,<Context>); 垂线(<点>,<直线>,<平面 xOy | 3D space>).**

创建过给定点垂直于给定直线且平行于给定平面的直线。

案例：“垂线(<点>,<直线>,<平面>)”创建平行于平面过给定点垂直于给定直线的垂线。

“垂线(<点>,<直线>,<3D 平面>)”创建平行于平面过给定点垂直于给定直线的垂线。



编者注：自 524 版，增加了过已知点做一次函数垂线功能。“垂线(<点>,<一次函数>)”即可。

## 2.8.6 Vertex. 顶点

**Vertex(<Conic>); 顶点(<圆锥曲线>)。**

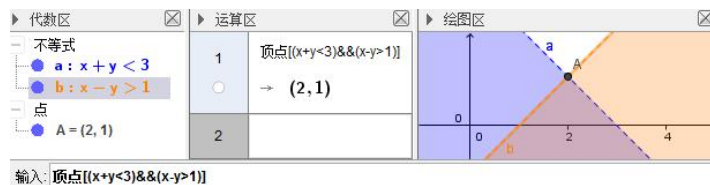
返回圆锥曲线的（全部）顶点。



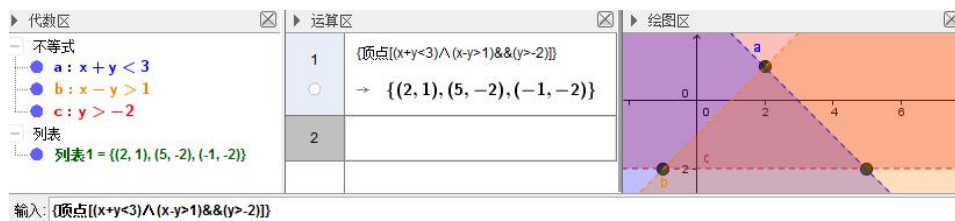
**Vertex(<Inequality>); 顶点(<不等式>)。**

返回边界的交叉点。

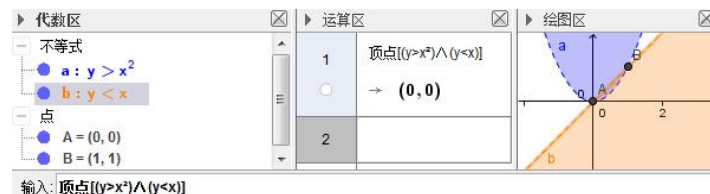
案例：“顶点((x+y<3)&&(x-y>1))”返回点“A=(2,1)”。



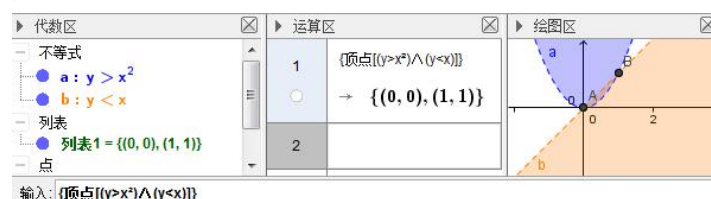
“{顶点((x+y<3)^(x-y>1)&&(y>-2))}”返回“列表 1={ (2,1), (5,-2), (-1,-2) }”。



“顶点((y>x^2)^(y<x))”返回两个点“A=(0,0)”和“B=(1,1)”。

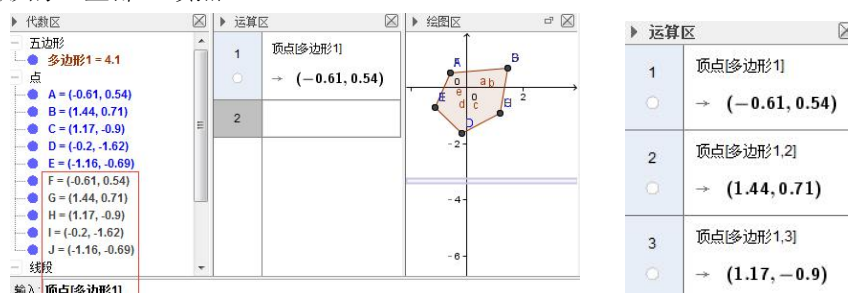


“{顶点((y>x^2)^(y<x))}”返回“列表 1={ (0,0), (1,1) }”。



### Vertex(<Polygon>); 顶点(<多边形>).

返回多边形的（全部）顶点。



### Vertex(<Polygon>,<Index>); 顶点(<多边形>,<索引>).

返回多边形的第 n 个顶点。

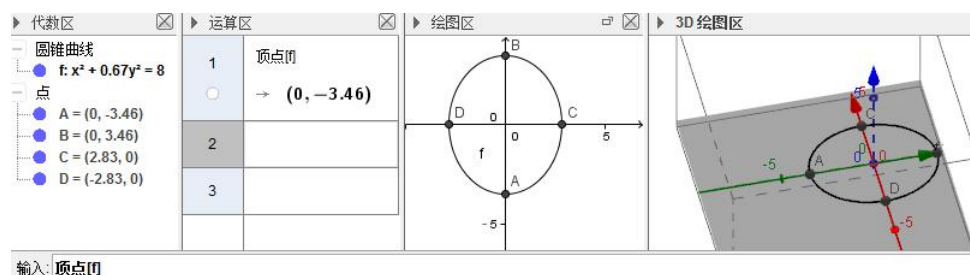
注：想得到多边形/圆锥曲线/不等式对象的顶点列，使用“{顶点(对象)}”。

### Vertex(<Segment>,<Index>); 顶点(<线段>,<索引>).

返回线段的起点（索引=0）和终点（索引=1）。

注：自 GeoGebra 5，这个指令同样适用于 3D 对象。

### Vertex(<Conic>); 顶点(<圆锥曲线>).



返回圆锥曲线的（全部）顶点。

### Vertex(<Polygon>); 顶点(<多边形>).

返回多边形的（全部）顶点。

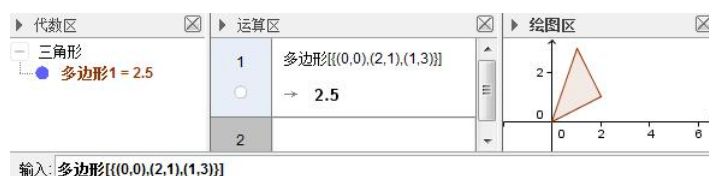


## 2.8.7 Polygon. 多边形

### Polygon(<List of Points>); 多边形(<点列>).

返回由点列中点定义的多边形。

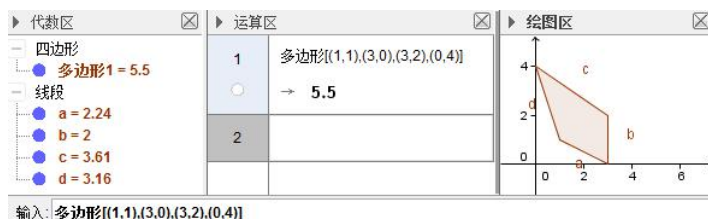
案例：“多边形{(0,0),(2,1),(1,3)}”得出三角形。



**Polygon(<Point>,...,<Point>); 多边形(<点 1>,...,<点 n>)。**

返回一个由给定点所确定的多边形。

案例：“多边形((1,1),(3,0),(3,2),(0,4))”得出一个四边形。



**Polygon(<Point>,<Point>,<Number of Vertices>); 多边形(<点 1>,<点 2>,<顶点数>)。**

创建一个有 n 个顶点的正 n 边形。

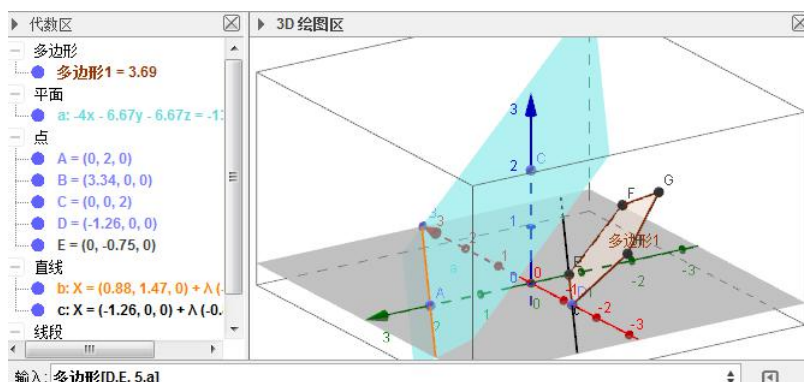
案例：“多边形((1,1),(4,1),6)”得出一个正六边形。



注：参见 “多边形” 和 “正多边形” 工具。

**Polygon(<Point>,<Point>,<Number of Vertices>,<Direction>); 多边形(<点 1>,<点 2>,<顶点数>,<方向向量>)。**

创建一个正 n 边形，其方向性由方向物决定（如有可能平行于一个平面）。

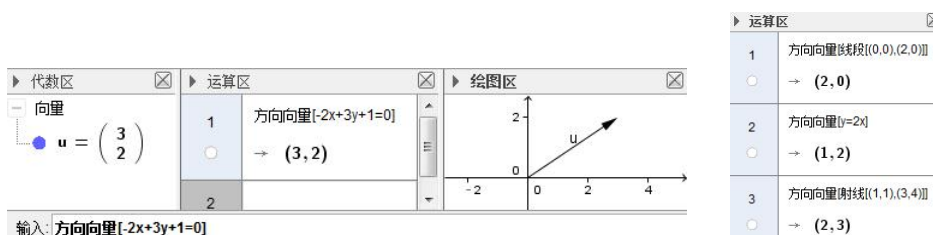


## 2.8.8 Direction. 方向向量

**Direction(<Line>); 方向向量(<直线|射线|线段>)。**

得出直线|射线|线段的方向向量。

案例：“方向向量(-2x+3y+1=0)”得出向量  $u = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ 。

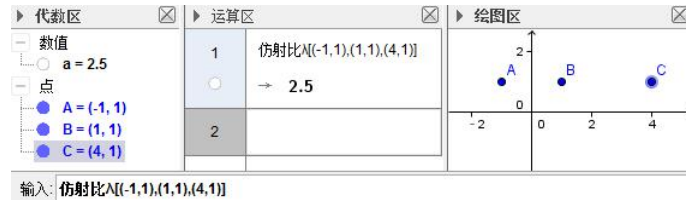


注：方程式为  $ax+by=c$  的直线其方向向量为  $(b,-a)$ 。

### 2.8.9 AffineRatio. 仿射比 $\lambda$

**AffineRatio(<Point A>,<Point B>,<Point C>); 仿射比  $\lambda$  (<点 1>,<点 2>,<点 3>)**。  
返回共线三点 A、B 和 C 的仿射比  $\lambda$ ， $C=A+\lambda*AB$ （即  $\lambda=AC$  距离  $\div$   $AB$  距离）。

案例：“仿射比  $\lambda((-1,1),(1,1),(4,1))$ ” 得出 2.5。



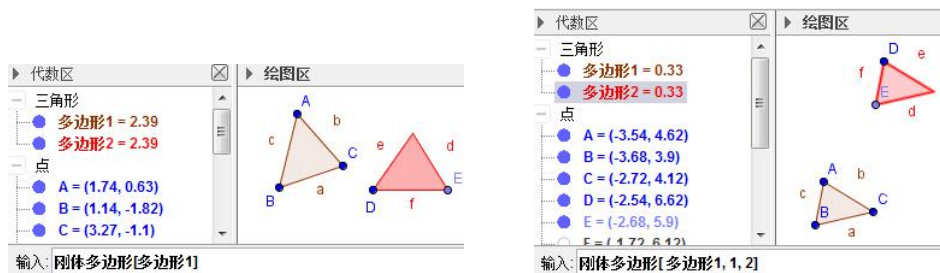
### 2.8.10 RigidPolygon. 刚体多边形

**RigidPolygon(<Polygon>); 刚体多边形(<多边形>)**。

创建任意多边形的一个副本，其只能通过拖动第一个顶点而进行移动，以及拖动第二个顶点而进行旋转。

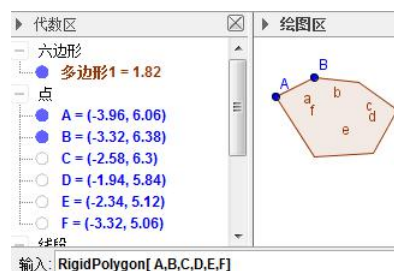
**RigidPolygon(<Polygon>,<Offset x>,<Offset y>); 刚体多边形(<多边形>,<x 偏移量>,<y 偏移量>)**。

按给定的偏移量创建任意多边形的一个副本，其只能通过拖动第一个顶点而进行移动，以及拖动第二个顶点而进行旋转。



**RigidPolygon(<Free Point>,...,<Free Point>); 刚体多边形(<自由点 1>,...,<自由点 n>)**。

创建其形状不能被更改的多边形。这个多边形可以拖动其第一个顶点而进行移动，以及拖动其第二个顶点而进行旋转。



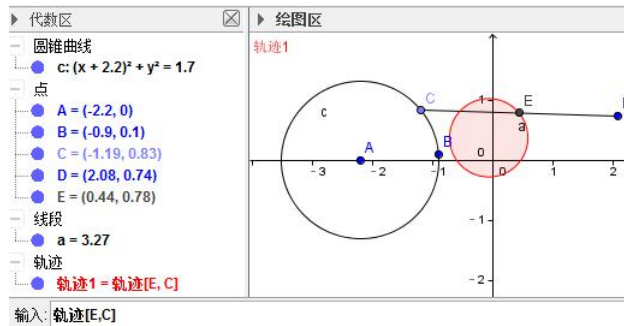
注：副本链接了源多边形的每一个改变。如果要改变副本的形状，只能更改其来源多边形。

### 2.8.11 Locus. 轨迹

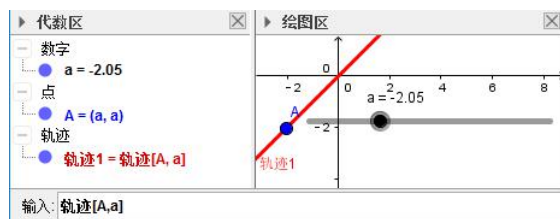
**Locus(<Point Creating Locus Line Q>,<Point P>); 轨迹(<构造轨迹的点>,<控制点>)**。

返回从属于点 P 的轨迹点 Q 的轨迹。

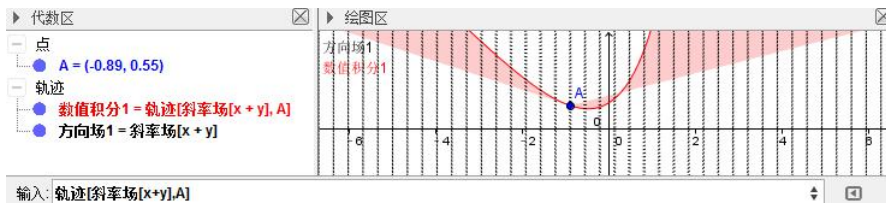
注：点 P 需要是对象上（如直线、线段、圆）的点。返回第一个参数点的轨迹曲线，这个点是依赖于第二个参数点的派生点。



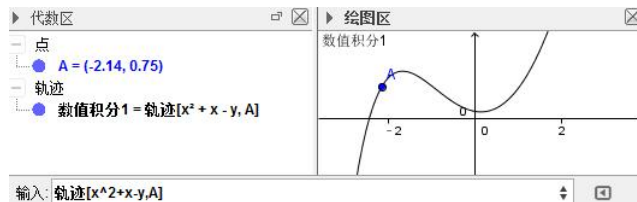
**Locus(<Point Creating Locus Line Q>,<Slider t>); 轨迹(<构造轨迹的点>,<滑动条>).**  
 返回从属于采用滑动条值的点 Q 的轨迹曲线。



**Locus(<Slopefield>,<Point>); 轨迹(<斜率场>,<点>).**  
 返回相当于过给定点的斜率场的轨迹曲线。



**Locus(<f(x,y)>,<Point>); 轨迹(<f(x,y)>,<点>).**  
 返回相当于解微分方程  $dy/dx=f(x,y)$  的轨迹曲线。结果是由数值计算方法得出。



轨迹是作为辅助对象出现的一种特殊对象类型，除了“轨迹”指令，一些离散数学指令和解常微分方程指令的结果也是这类特殊对象。轨迹是路径类型的对象并可用于和路径相关的指令，例如“描点”指令。它们的特性（描点的路径值）取决于它们是如何被构造的，比如说，“周长”指令和“最前元素”指令。

**注：**参见 “轨迹”工具。

**警告：**当从属点是使用两个参数的“描点”指令或是“路径值”指令这两种情况所建的时候，轨迹不能被定义。

## 2.8.12 LocusEquation. 轨迹方程

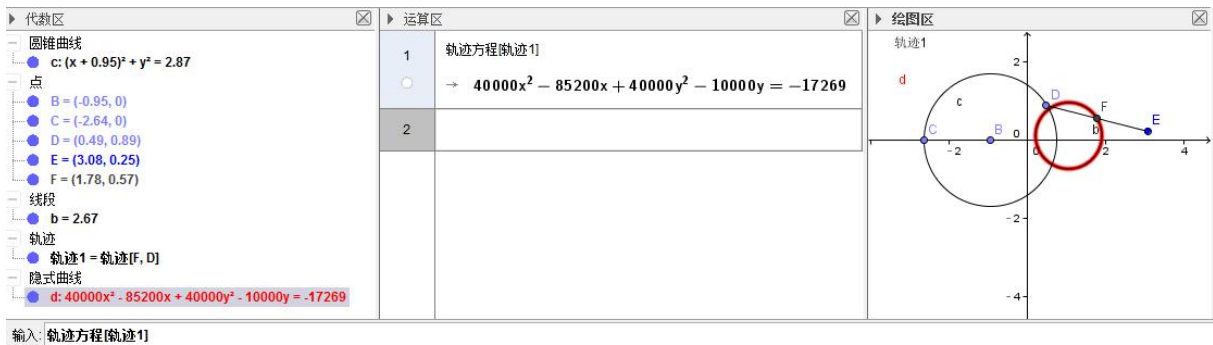
**LocusEquation(<Locus>); 轨迹方程(<轨迹>).**

计算轨迹的方程并以隐式曲线表示。

**LocusEquation(<Point Creating Locus Line Q>,<Point P>); 轨迹方程(<轨迹点>,<动点>).**

使用轨迹点和一个动点计算轨迹方程并以隐式曲线表示。





**案例：**先构造一个抛物线轨迹：创建自由点 A 和 B，过两点的直线 d（这将是抛物线的准线）。创建自由的 F 点为焦点。在直线 d 上创建动点 P，然后过点 P 创建直线 d 的垂线 p。创建线段 FP 的垂直平分线 b，直线 p 和 b 交点 Q 就是构造轨迹的点。

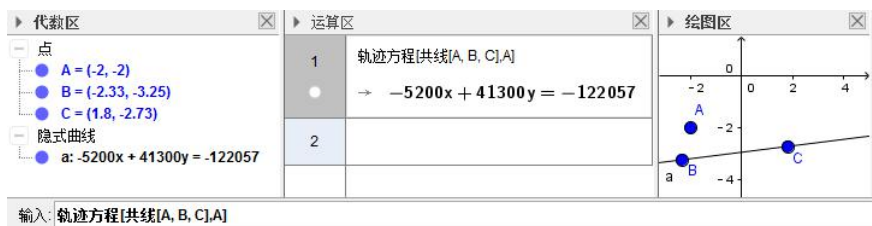
现在，“轨迹方程(Q,P)”会找出并绘制轨迹的确切方程。



**LocusEquation(<Boolean Expression>, <Free Point>); 轨迹方程(<布尔表达式>, <自由点>).**

假设布尔值条件被满足，计算自由点的轨迹。

**案例：**对于自由点 A、B、C，“轨迹方程(是否共线(A,B,C),A)”计算点 A 处于与 B 和 C 共线位置时的轨迹，即过 BC 的直线。



**注：**轨迹必须是点驱动的(不能是滑条参数)；指令只对有限的几何图形如点、线、圆和圆锥曲线有效(射线和线段视为“无限”线)；如果轨迹太复杂，会返回“未定义”；如果没有轨迹，隐式曲线会返回空集  $0=1$ ；如果轨迹是整个平面则隐式曲线是方程  $0=0$ ；因为是基于代数的计算，有时轨迹会出现偏差；在 GeoGebra 通道有更多的信息和案例；参见“轨迹”指令。

### 2.8.13 Arc. 弧线

**Arc(<Circle>, <Point M>, <Point N>); 弧线(<圆>, <点 1>, <点 2>).**

返回给定圆的定向弧线(逆时针方向)，其端点为 M 和 N。

**Arc(<Ellipse>, <Point M>, <Point N>); 弧线(<椭圆>, <点 1>, <点 2>).**

返回给定椭圆的定向弧线(逆时针方向)，其端点为 M 和 N。

**Arc(<Circle>, <Parameter Value>, <Parameter Value>); 弧线(<圆>, <参数值 1>, <参数值 2>).**

返回给定圆的弧线，其端点由指定的参数值所确定。

注：下列参数形式可用于指令之中：“圆： $(r \cos(t), r \sin(t))$ ”，其中  $r$  是圆的半径。  
**Arc**(**<Ellipse>**, **<Parameter Value>**, **<Parameter Value>**)；**弧线**(**<椭圆>**, **<参数值 1>**, **<参数值 2>**)。

返回给定椭圆的弧线，其端点由指定的参数值所确定。

注：下列参数形式可用于指令之中：“椭圆： $(a \cos(t), b \sin(t))$ ”，其中  $a$  和  $b$  是主轴和副轴的长。

注：参见“三点圆弧”指令。

### 2.8.14 CrossRatio. 交比

**CrossRatio**(**<Point A>**, **<Point B>**, **<Point C>**, **<Point D>**)；**交比**(**<点 1>**, **<点 2>**, **<点 3>**, **<点 4>**)。

计算四个共线点 A、B、C 和 D 的交比  $\lambda$ ，其中： $\lambda = \text{仿射比 } \lambda(B, C, D) / \text{仿射比 } \lambda(A, C, D)$ 。

案例：“交比((-1,1),(1,1),(3,1),(4,1))”得出 1.2。

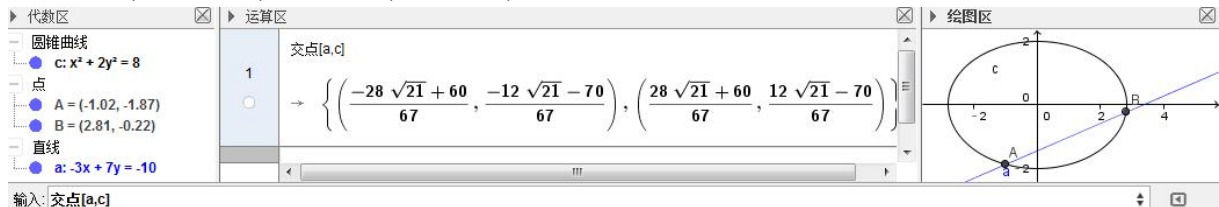


### 2.8.15 Intersect. 交点

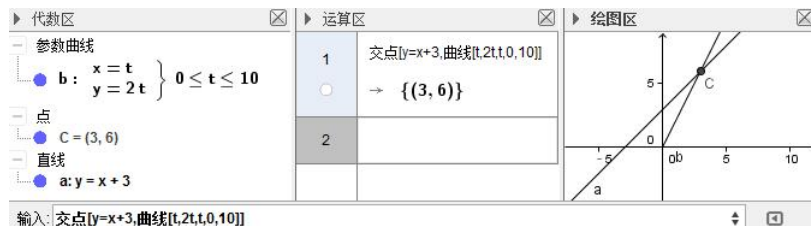
**Intersect**(**<Object>**, **<Object>**)；**交点**(**<对象 1>**, **<对象 2>**)。

得出两个对象的交点。

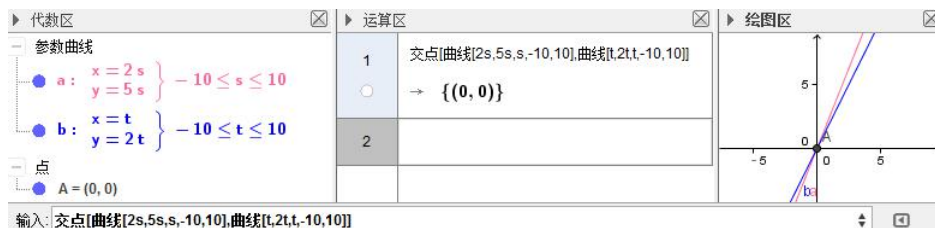
案例：设直线“ $a: -3x + 7y = -10$ ”和椭圆“ $c: x^2 + 2y^2 = 8$ ”，“交点(a,c)”得出直线与椭圆的两个交点“ $E = (-1.02, -1.87)$ ”和“ $F = (2.81, -0.22)$ ”。



“交点(y=x+3, 曲线(t,2t,t,0,10))”得出“ $C = (3, 6)$ ”。



“交点(曲线(2s,5s,s,-10,10), 曲线(t,2t,t,-10,10))”得出“ $A = (0, 0)$ ”。



注：得出两个对象（在可视范围内）的全部交点。这里的对象可以是直线、线段、射线、折线、多边形、圆、函数等，不过有些对象之间不能找到交点，有些复杂的函数会出现交点不全现象。

**Intersect(<Object>,<Object>,<Index of Intersection Point>);** 交点(<对象 1>,<对象 2>,<交点索引>)。

得出两个对象的第 n 个交点。

**案例：**设函数 “ $a(x)=x^3+x^2-x$ ” 和直线 “ $b:-3x+5y=4$ ”，“交点(a,b,2)” 得出函数与直线的交点 “ $A=(-0.43,0.54)$ ”。



**Intersect(<Object>,<Object>,<Initial Point>);** 交点(<对象 1>,<对象 2>,<起点>)。

从起点开始使用（数值）迭代法得出两个对象的一个交点。

**案例：**设函数 “ $a(x)=x^3+x^2-x$ ”，直线 “ $b:-3x+5y=4$ ” 和起点 “ $C=(0,0.8)$ ”，“交点(a,b,C)”，使用（数值）迭代法得出函数与直线的交点 “ $B=(-0.43,0.54)$ ”。



**Intersect(<Function>,<Function>,<Start x-Value>,<End x-Value>);** 交点(<函数 1>,<函数 2>,<x-起始值>,<x-终止值>)。

得出两个函数在指定数值区间上的全部交点。

**案例：**设两个函数 “ $f(x)=x^3+x^2-x$ ” 和 “ $g(x)=4/5+3/5x$ ”，“交点(f,g,-1,2)” 得出两个函数在闭区间(-1,2)上的交点 “ $A=(-0.43,0.54)$ ” 和 “ $B=(1.1,1.46)$ ”。



**Intersect(<Curve 1>,<Curve 2>,<Parameter 1>,<Parameter 2>);** 交点(<曲线 1>,<曲线 2>,<参数 1>,<参数 2>)。

从指定参数开始以迭代法寻找两条曲线的一个交点。

**案例：**设 “ $a=$ 曲线 $(\cos(t),\sin(t),t,0,\pi)$ ” 和 “ $b=$ 曲线 $(\cos(t)+1,\sin(t),t,0,\pi)$ ”，“交点(a,b,0,2)” 得出交点 “ $A=(0.5,0.87)$ ”。

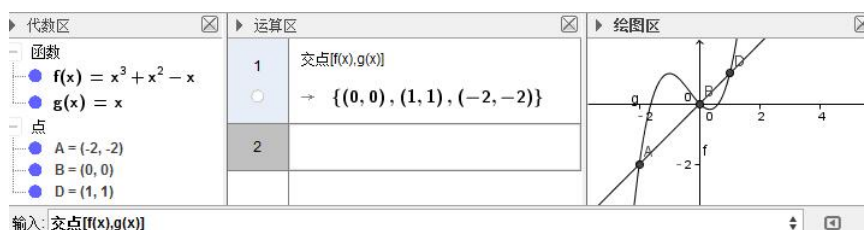


**CAS Syntax（运算区语法）**

**Intersect(<Function>,<Function>); 交点(<函数 1>,<函数 2>)。**

得出一个包含两个函数对象全部交点的列表。

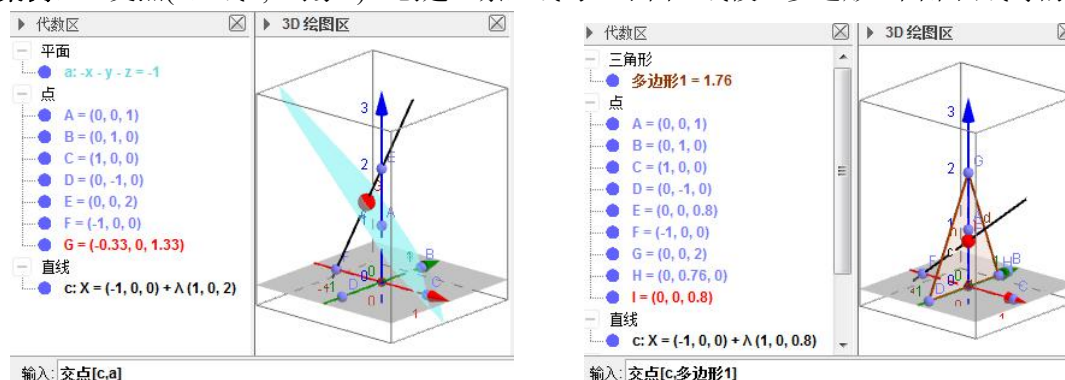
**案例：**函数“ $f(x)=x^3+x^2-x$ ”和“ $g(x)=x$ ”。 “交点( $f(x),g(x)$ )” 得出这两个函数的交点列： $\{(1,1),(0,0),(-2,-2)\}$ 。



注：自 GeoGebra 5，这个命令适于 3D 对象。

**Intersect(<Object>,<Object>); 交点(<对象 1>,<对象 2>)。**

**案例：**“交点(<直线>,<对象>)” 创建一条直线与一平面、线段、多边形、圆锥曲线等的交点。



“交点(<平面>,<对象>)” 创建一个平面与一条线段、多边形、圆锥曲线的交点。

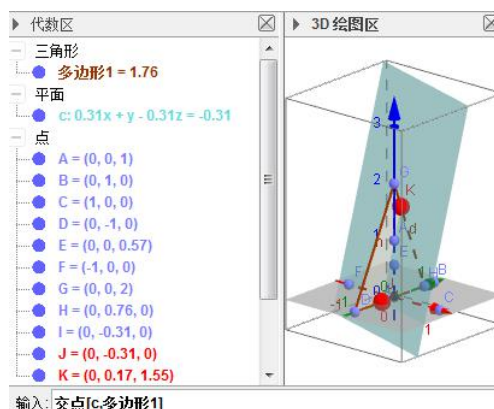
“交点(<圆锥曲线 1>,<圆锥曲线 1>)” 创建两个圆锥曲线的交点。


“交点(<平面 1>,<平面 2>)” 创建两个平面的交线。

“交点(<平面>,<多面体>)” 创建平面和多面体的相贯线。

“交点(<球 1>,<球 2>)” 创建两个球的圆形相交线。

“交点(<平面>,<二次曲面>)” 创建平面和二次曲线（球、圆锥、圆柱等）的相贯线。



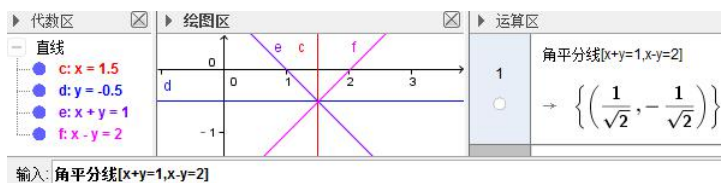
注：想得到交点的列表，可以使用“{交点(a,b)}”；参见“相交曲线”和“相交路径”指令；参见  “交点”工具。

## 2.8.16 AngleBisector. 角平分线

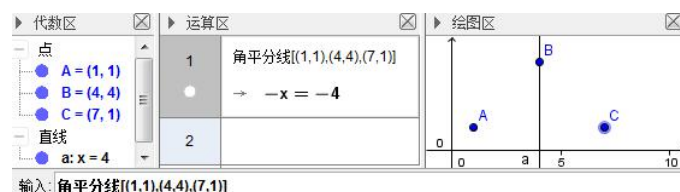
**AngleBisector(<Line>,<Line>); 角平分线(<直线 1>,<直线 2>)。**

返回由两条直线确定的全部（两条）角平分线。

案例：“角平分线( $x+y=1, x-y=2$ )”得出“ $a:x=1.5$ ”和“ $b:y=-0.5$ ”。



**AngleBisector**(<Point>, <Point>, <Point>); 角平分线(<点 1>, <顶点 2>, <点 3>)  
返回由三个点确定的角的角平分线。



案例：“角平分线((1, 1), (4, 4), (7, 1))”得出“ $a:x=4$ ”。

注：第二个点（点 2）是这个角的顶点；参见“角平分线”工具。

## 2.8.17 Distance. 距离

**Distance**(<Point>, <Object>); 距离(<点>, <对象>)

得出一个点与一个对象之间的最短距离。

案例：“距离((2, 1),  $x^2+(y-1)^2=1$ )”得到 1。



注：这个指令可用于点、线段、直线、圆锥（二次）曲线、函数和隐式曲线。对于函数使用更适于多项式的数值算法。

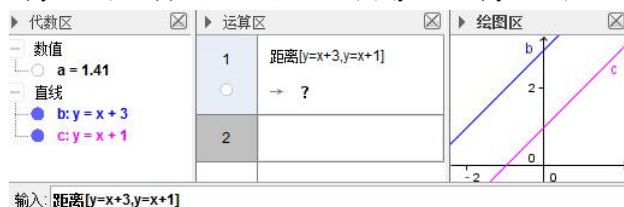
案例：设有函数  $f$  和点  $A$ ，“距离( $A, f$ )”得出点  $A$  和点“ $(x(A), f(x(A)))$ ”间的距离。

编者注：这个案例官网解释不对，应该是点与函数上最近点的距离。

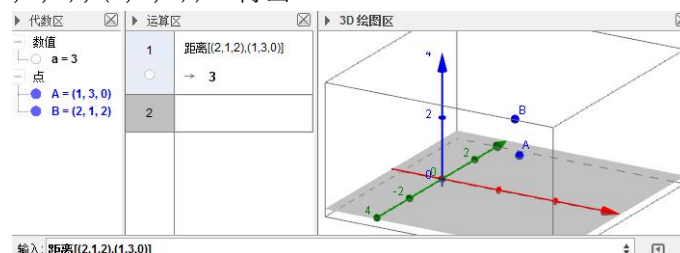
**Distance**(<Line>, <Line>); 距离(<直线 1>, <直线 2>)

得出两条直线之间的距离。相交直线之间的距离为 0。因此，这个指令只对平行线有实际意义。

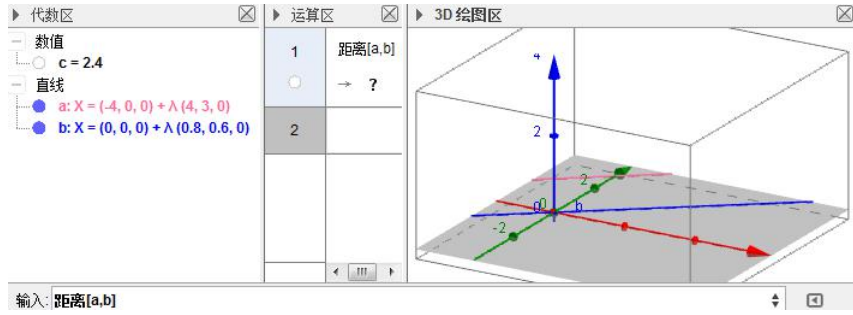
案例：“距离( $y=x+3, y=x+1$ )”得出 1.41；“距离( $y=3x+1, y=x+1$ )”=0。



案例：“距离((2, 1, 2), (1, 3, 0))”得出 3。



案例：设 “ $a: X = (-4, 0, 0) + \lambda * (4, 3, 0)$ ”， “ $b: X = (0, 0, 0) + \lambda * (0.8, 0.6, 0)$ ”。 “距离(a, b)” 得出 2.4。



Distance(<Plane>, <Plane>); 距离(<平面 1>, <平面 2>)。

得出两个平行平面的距离。

注：参见 “距离/长度” 工具。

### 2.8.18 Angle. 角度

Angle(<Object>); 角度(<对象 圆锥曲线|向量|点|数字|多边形>)。

圆锥曲线：返回圆锥曲线主轴的偏转角度（基于 x 轴偏转，见 “轴线” 指令）

案例：“角度( $x^2/4+y^2/9=1$ )” 得出  $90^\circ$  或者 1.57（默认角度为弧度）。



向量：返回指定向量与 x-轴之间的角度。

案例：“角度(向量((1,1)))” 得出  $45^\circ$  或者弧度制时的值。

点：返回指定点的位置向量与 x 轴和原点为顶点的角度。

案例：“角度((1,1))” 得出  $45^\circ$  或者弧度制时的值。

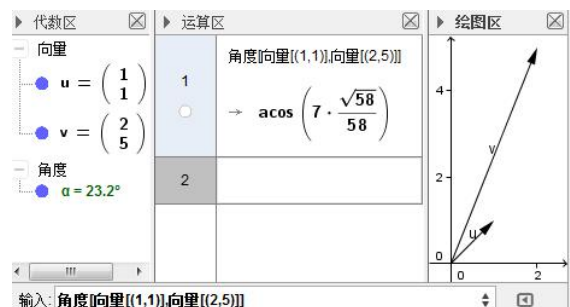
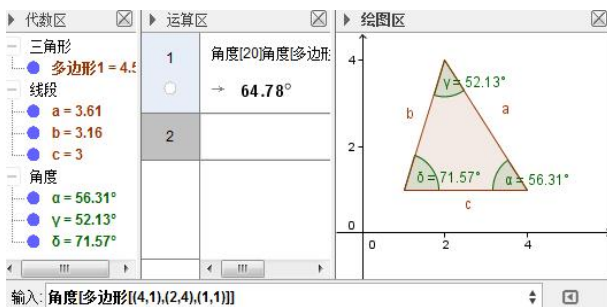


数值：弧度值转换为角度值，（结果在 0 到  $2\pi$  之间）。

案例：“角度(20)” 得出  $65.92^\circ$ （当角度单位为度时）。

多边形：创建多边形正向所有内角角度（多边形逆时针构造为正）。

案例：“角度(多边形((4,1),(2,4),(1,1)))” 得出  $56.31^\circ$ ,  $52.13^\circ$  和  $71.57^\circ$  或者弧度制时的值。



注：如果多边形是以逆时针方向选取，可以得到内角。如果是顺时针方向选取，可以得到外角。

**Angle(<Vector>,<Vector>); 角度(<向量 1>,<向量 2>)**。

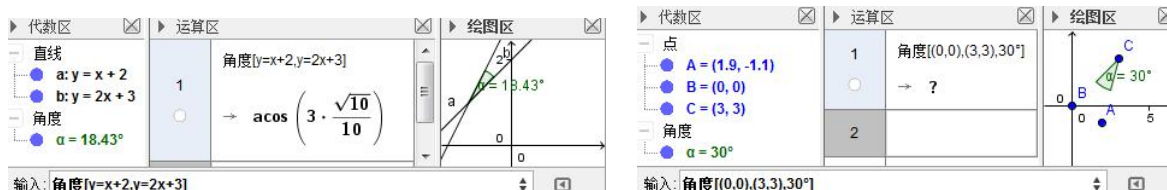
返回两个向量之间的角度 ( $0^\circ$  到  $360^\circ$ ) 或  $(0, 2\pi)$ ，取决于默认角度单位。

案例：“角度(向量((1,1),向量((2,5)))” 得出  $23.2^\circ$  或者弧度制时的值。

**Angle(<Line>,<Line>); 角度(<直线 1>,<直线 2>)**。

返回两条直线方向矢量之间的角度 ( $0^\circ$  到  $360^\circ$ ) 或  $(0, 2\pi)$ ，取决于默认角度单位。

案例：“角度( $y=x+2,y=2x+3$ )” 得出  $18.43^\circ$  或者弧度制时的值。



案例：“角度(直线((-2,0,0),(0,0,2)),直线((2,0,0),(0,0,2)))” 得出  $90^\circ$  或者 1.57 (默认角度为弧度)。

**Angle(<Point>,<Apex>,<Point>); 角度(<点>,<顶点>,<点>)**。

返回由三个点确定的角的角度值 ( $0^\circ$  到  $360^\circ$ )。

案例：“角度((1,1),(1,4),(4,2))” 得出  $56.31^\circ$  或者弧度制时的值。

**Angle(<Point>,<Apex>,<Angle>); 角度(<点>,<顶点>,<角度[弧度]>)**。

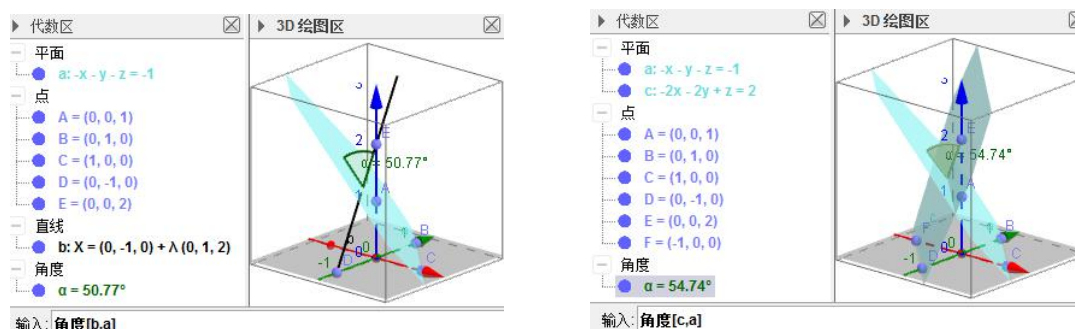
返回指定点、按顶点旋转  $\alpha$  角度后的位置点。

案例：“角度((0,0),(3,3), $30^\circ$ )” 得出  $30^\circ$  和点(1.9,-1.1)。

注：使用“旋转(<点>,<角度>,<顶点>)” 同样能创建这个点。

**Angle(<Line>,<Plane>); 角度(<直线>,<平面>)**。

返回直线和平面间的夹角。



案例：“角度(直线((1,2,3),(-2,-2,0)), $z=0$ )” 得出  $30.96^\circ$  或者弧度制的值。

**Angle(<Plane>,<Plane>); 角度(<平面 1>,<平面 2>)**。

返回给定两个平面间的夹角。

案例：“角度( $2x-y+z=0,z=0$ )” 得出  $114.09^\circ$  或者弧度制的值。



**Angle(<Point>,<Point>,<Point>,<Direction>); 角度(<点 1>,<点 2>,<点 3>,<方向>)**。

返回由直线或者平面给定方向，由三个点确定的角度，结果根据默认角度单位不同，介于区间  $(0, 360^\circ)$  或  $(0, 2\pi)$  内。

注：使用方向能够在 3D 中设定角度显示在  $(0, 180^\circ)$  或  $(180^\circ, 360^\circ)$ ，对于给定点 A、B、C，指令“角度(A, B, C)”和“角度(A, B, C)”就返回在限定区间的实际值。

案例：“角度((1, -1, 0), (0, 0, 0), (-1, -1, 0), zAxis)” 得出  $270^\circ$ ，而“角度((-1, -1, 0), (0, 0, 0), (1, -1, 0), zAxis)” 得出  $90^\circ$  或者弧度的值。



注：角度的单位（度数或弧度）可在选项菜单（高级）的“全局参数”设置中选择。所有由角度指令所建立或赋值的对象角，结果介于  $0$  与  $2\pi$  或  $0^\circ$  与  $360^\circ$  之间。

如果想将一个角度对象的最大尺寸（角度值）限制在  $180^\circ$  之内，可在属性对话框的常规选项卡中设定。

注：参见 “角度” 和 “定值角” 工具。

### 2.8.19 Area. 面积

Area(<Conic>); 面积(<圆或椭圆>)。

计算一个平面圆锥曲线（圆或椭圆）的面积。

案例：“面积( $x^2+y^2=2$ )” 得出 6.28。



Area(<Polygon>); 面积(<多边形>)。

计算多边形的面积。

Area(<Point>, ..., <Point>); 面积(<点 1>, ..., <点 n>)。

计算由给定点所定义的多边形面积。

案例：“面积((0, 0), (3, 0), (3, 2), (0, 2))” 得出 6。



注：计算多边形的代数区域面积，如果是自相交多边形，会有异常结果；若要计算两个函数图象之间的面积，需要使用指令“积分介于”；参见 “面积” 工具。

### 2.8.20 Point. 描点

Point(<Object>); 描点(<几何对象>)。

返回几何对象上的点。结果点能在路径上移动。



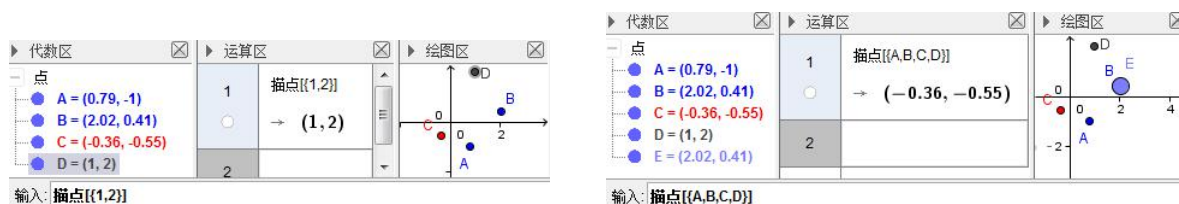


运算区只生成第一个符合条件的点，绘图区的点与代数区的点表达式联动。

**Point(<List>); 描点(<有序数组列表>)。**

将一个只包含两个数值的集合转换成一个点。

**案例：**“描点({1, 2})”得出(1, 2)。



①将一个只包含两个数值的列表转换成一个点。如：新点1({1, 2})=(1, 2)。②如果列表是点集，输出一个随机动点，可以与点集中任何点坐标相同，并可拖动和动画，形成跳点效果。“描点({A, B, C, D})”得出“E(E=A/B/C/D/E, 可拖动、可动画, 点E就在点A、B、C、D间跳跃)”。

**Point(<Object>, <Parameter>); 描点(<几何对象>, <路径值>)。**

按指定的路径值返回一个在几何对象上的定点。



按指定的路径值返回一个在几何对象上的定点，它的位置由参数决定。对象可以是线段、向量、圆、椭圆、多边形，路径值为0-1之间任意数，如果输入<0的数按0处理，输入>1的数按1处理。对于直线、射线、抛物线、双曲线、函数等无限长的线，这个指令虽然可以用，但是结果不可预知。

用此命令可较快输出对象的等分点，同时引入变量作参数，可以控制点的运动范围。

**案例：**下图显示出圆上的点与其参数之间的关系（圆为  $c: x^2 + y^2 = 1$ ）。如果输入“绘制点(c, 0.4)”，则会得到图中的P点。反过来输入“路径值(P)”，则会传回参数0.4。



**Point(<Point>, <Vector>); 描点(<点>, <向量>)。**

创建一个按向量将指定点平移之后的点。

**注：**参见 “点”工具和 “对象上的点”工具。



## 2.8.21 PointIn. 内点

**PointIn(<Region>); 内点(<区域>)。**

返回一个限定在指定区域内的点。

**案例：**“内点(c)”得到点A (c为一圆, A为圆内一点且可拖动)。

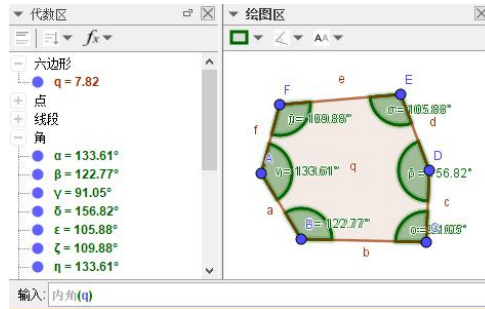


注：参见  “附着/脱离点”工具；区域可以是圆、椭圆、多边形、不等式等；参见  “对象上的点”工具。

## 2.8.22 InteriorAngles. 内角

InteriorAngles(<Polygon>); 内角(<多边形>).

返回给定多边形的所有内角值。

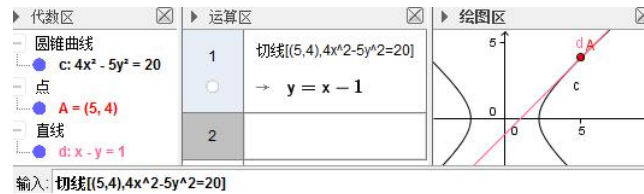


## 2.8.23 Tangent. 切线

Tangent(<Point>, <Conic>); 切线(<点>, <圆锥曲线>).

创建圆锥曲线过指定点的（全部）切线。

案例：“切线((5, 4),  $4x^2 - 5y^2 = 20$ )”得出“ $x - y = 1$ ”。

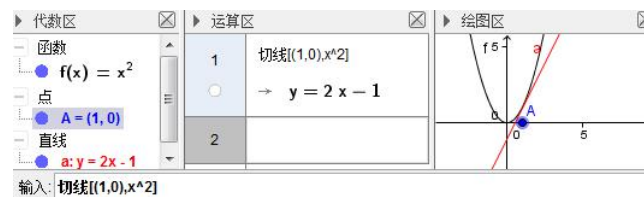


Tangent(<Point>, <Function>); 切线(<点>, <函数>).

创建函数在函数点  $x = x(A)$  上的切线。

注：“ $x(A)$ ”是指定点 A 的 x 轴坐标。

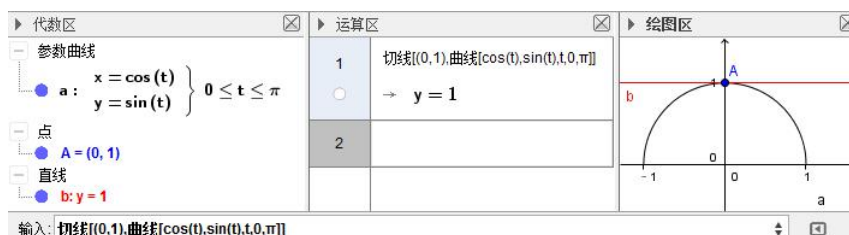
案例：“切线((1, 0),  $x^2$ )”得出“ $y = 2x - 1$ ”。



Tangent(<Point on Curve>, <Curve>); 切线(<曲线上的点>, <曲线>).

创建过曲线上的指定点并与曲线相切的切线。

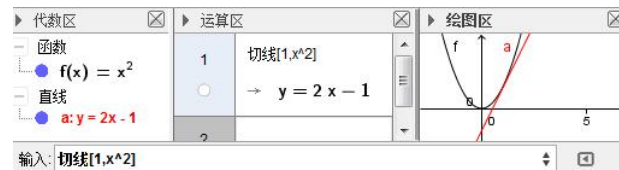
案例：“切线((0, 1), 曲线( $\cos(t)$ ,  $\sin(t)$ ,  $t, 0, \pi$ ))”得出“ $y = 1$ ”。



Tangent(<x-Value>, <Function>); 切线(<横坐标 x 值>, <函数>).

创建函数在  $x = \text{“x 值”}$  时的切线。

案例：“切线(1,  $x^2$ )” 得出 “ $y = 2x - 1$ ”。



**Tangent(<Line>, <Conic>); 切线(<直线>, <圆锥曲线>).**

创建圆锥曲线与指定直线平行的（全部）切线。

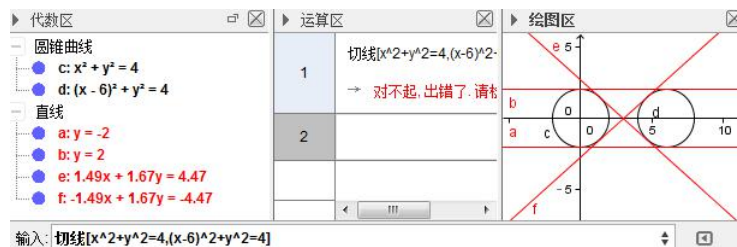
案例：“切线( $y = 4, x^2 + y^2 = 4$ )” 得出 “ $y = 2$ ” 和 “ $y = -2$ ”。



**Tangent(<Circle>, <Circle>); 切线(<圆锥曲线 1>, <圆锥曲线 2>).**

创建两个圆的公切线（最多可得 4 条切线）。

案例：“切线( $x^2 + y^2 = 4, (x - 6)^2 + y^2 = 4$ )” 得出  $y = 2, y = -2, 1.49x + 1.67y = 4.47$  和  $-1.49x + 1.67y = -4.47$ 。

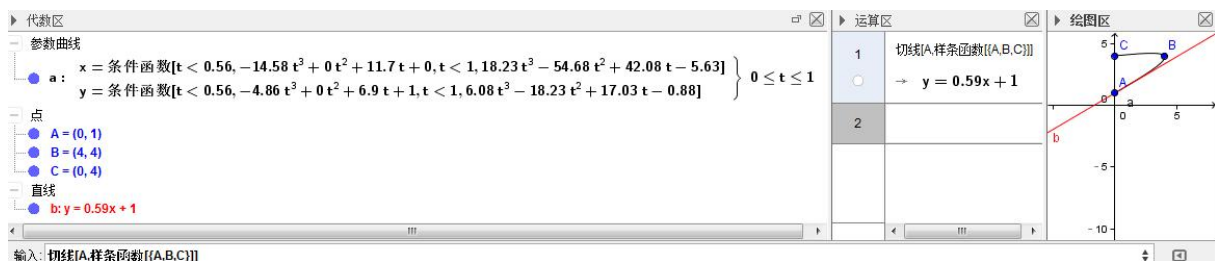


注：参见  “切线” 工具

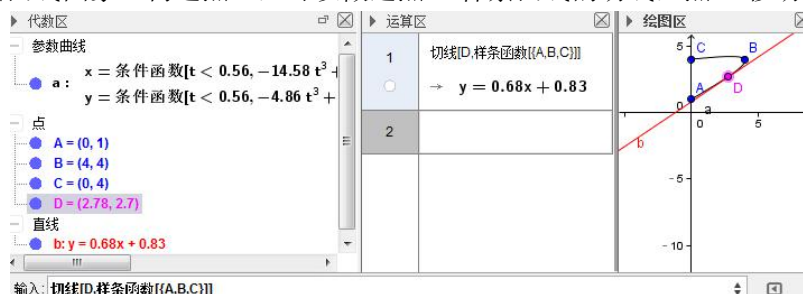
**Tangent(<Point>, <Spline>); 切线(<点>, <样条曲线>).**

创建给定点的样条曲线的切线。

案例：设  $A = (0, 1)$ 、 $B = (4, 4)$ 、 $C = (0, 4)$ 。“切线( $A$ , 样条函数( $\{A, B, C\}$ ))” 得出直线 “ $a: y = 0.1x + 4$ ”。



如果在样条曲线图象上构造点  $D$ ，可以做过点  $D$  样条曲线的切线，点  $D$  移动，切线随动。

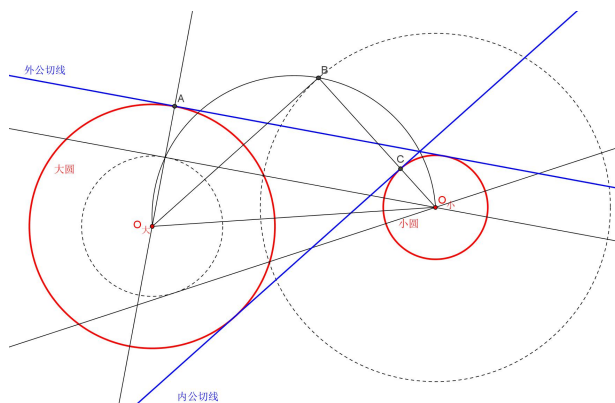


编者注：此条新版中没有描述，但功能确实存在。

#### 附：尺规方法作两圆的内外公切线

作外公切线：①以大圆圆心、半径之差作圆；②以小圆圆心向新圆作切线；③以大圆圆心向切线作垂线交大圆于 A；④再以 A 点作切线的平行线。

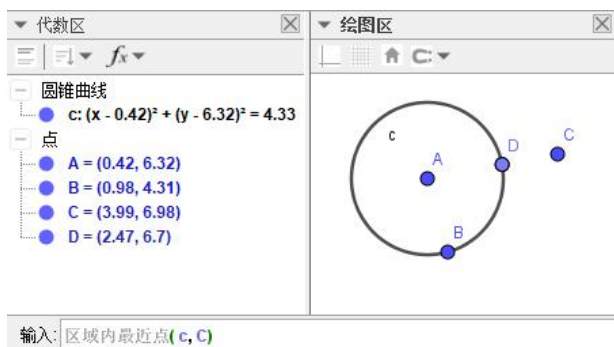
作内公切线：①任意圆心及两圆半径和画圆；②再以圆心距为直径画圆；③两新作圆交点与另一圆圆心相连，即得平行于公切线的直线。



### 2.8.24 ClosestPointRegion. 区域内最近点

ClosestPointRegion(<Region>, <Point>); 区域内最近点(<区域>, <点>)

在区域内构建一个距离给定点最近的点。



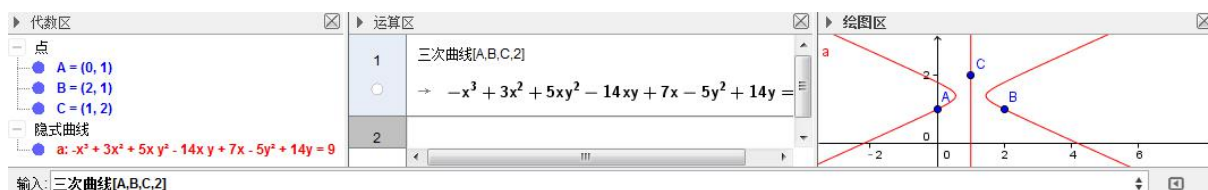
点 D 为对象上的点，改变点 C 的位置，D 在圆内（圆周和内部）可以跟随移动。

### 2.8.25 Cubic. 三次曲线

Cubic(<Point>, <Point>, <Point>, <Number>); 三次曲线(<点 1>, <点 2>, <点 3>, <类型数值 1\_ 纽伯格立方|2\_汤姆森立方|3\_麦克立方|4\_达布立方|5\_拿破仑/费尔巴哈立方|7\_卢卡斯立方|17\_第一布罗卡立方|18\_第二布罗卡立方>).

根据给定的序数值和三角形，输出三次隐式曲线。

案例：设 A=(0, 1), B=(2, 1), C=(1, 2)。“三次曲线(A, B, C, 2)”得出隐式曲线“ $x^3 - 3x^2 - 5xy^2 + 14xy - 7x + 5y^2 - 14y = -9$ ”。



### 常见的三次曲线


序数	三次曲线
1	纽伯格三次曲线
2	汤姆森三次曲线
3	麦克三次曲线
4	达布三次曲线
5	拿破仑/费尔巴哈三次曲线
7	卢卡斯三次曲线
17	第一布罗卡三次曲线
18	第二布罗卡三次曲线

**注：**本命令还在开发，支持的序数  $n$  在变化中。如果上例中建立滑动条  $n$ ，指令“三次曲线(A, B, C, n)”，改变  $n$  的值，会得到目前支持的曲线表达式和图象。

#### 2.8.26 CircumcircularSector. 三点扇形

**CircumcircularSector**(**<Point>**, **<Point>**, **<Point>**); 三点扇形(**<点 1>**, **<点 2>**, **<点 3>**)。


创建一个穿过三点形，其中第一个点为起点，第三个点为圆环弧线的终点。

**注：**参见  “三点扇形”工具。

#### 2.8.27 CircumcircularArc. 三点圆弧

**CircumcircularArc**(**<Point>**, **<Point>**, **<Point>**); 三点圆弧(**<点 1>**, **<点 2>**, **<点 3>**)。

创建一条由穿过三点所确定的圆弧线，其中第一个点为起点，第三个点为圆环弧线的终点。

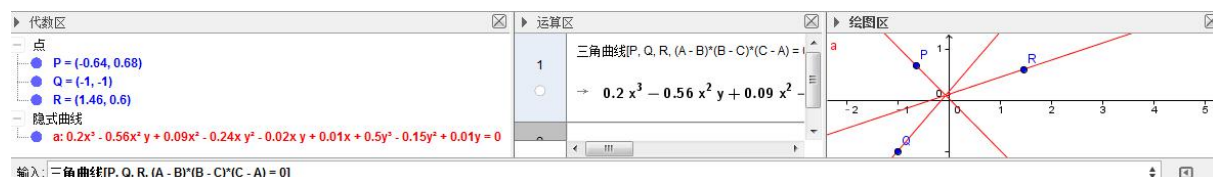
**注：**参见  “三点圆弧”工具。

#### 2.8.28 TriangleCurve. 三角曲线

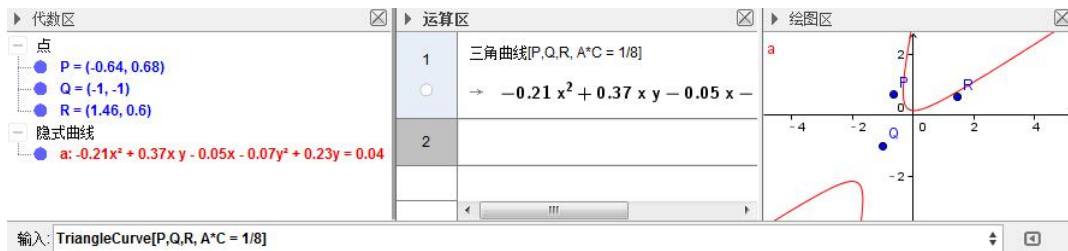
**TriangleCurve**(**<Point P>**, **<Point Q>**, **<Point R>**, **<Equation in A, B, C>**); 三角曲线(**<点 1>**, **<点 2>**, **<点 3>**, **<方程>**)。

创建隐式多项式，点 P、Q、R 确定的重心坐标方程由第四个参数确定：重心坐标做 ABC 方程提供的计算（P、Q、R 替代 A、B、C 做方程规定计算）。

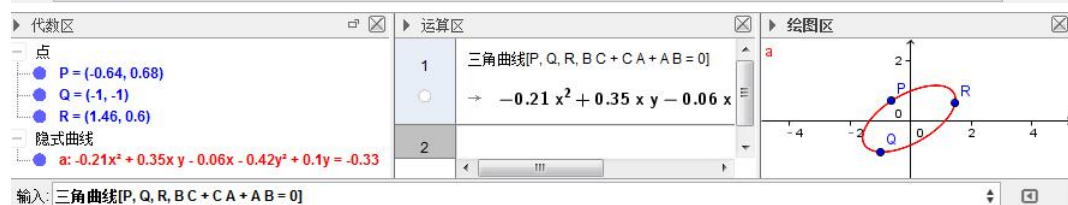
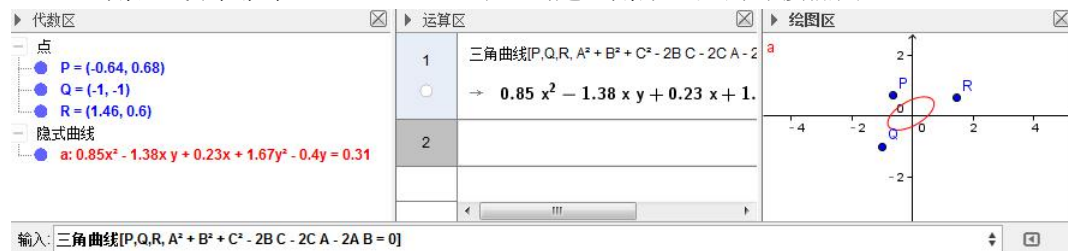
**案例：**有点 P、Q 和 R，“三角曲线(P, Q, R, (A-B)\*(B-C)\*(C-A)=0)”给出由三角形 PQR 中线确定的三次曲线。



**案例：**“三角曲线(P, Q, R, A\*C=1/8)”创建一个 PQ 和 QR 是其切线，且把三角形 PQR 分为面积相等两部分的双曲线。



**案例：**“三角曲线(P, Q, R,  $A^2+B^2+C^2-2B C-2C A-2A B=0$ )”创建三角形PQR的斯泰纳椭圆（内切椭圆）。“三角曲线(P, Q, R,  $B C + C A + A B=0$ )”创建三角形PQR的外接椭圆。



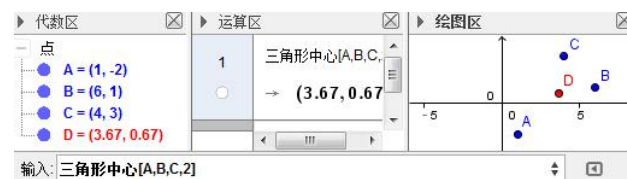
**注：**输入点可以被命名为A, B或C, 但在方程中不能使用如：“X(A)”的计算, 因为其被定义为重心坐标。

### 2.8.29 TriangleCenter. 三角形中心

TriangleCenter(<Point>, <Point>, <Point>, <Number>); 三角形中心(<点 1>, <点 2>, <点 3>, <数字 1\_内心|2\_重心|3\_外心|4\_垂心|5\_九点中心|6\_类似重心|7\_热尔岗点|8\_奈格尔点|13\_费马点>).

给出三角形ABC的第n个中心。N<3054。

**案例：**设“A=(1, -2)”, “B=(6, 1)”, “C=(4, 3)”。 “三角形中心(A, B, C, 2)”得出三角形ABC的重心“D=(3.67, 0.67)”。



常见的三角形中心

序数 n	中心
1	内心
2	重心
3	外心
4	垂心
5	九点中心

6	类似重心
7	热耳尔岗点
8	奈格尔点
13	费尔马点

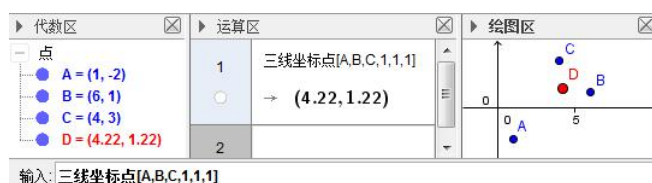
### 2.8.30 Trilinear. 三线坐标点

**Trilinear**(**<Point>**, **<Point>**, **<Point>**, **<Number>**, **<Number>**, **<Number>**); 三线坐标点(**<点 A>**, **<点 B>**, **<点 C>**, **<数值 a>**, **<数值 b>**, **<数值 c>**)。

创建由给定数值确定的给定点三角形的三线坐标点。

点	数字 1	数字 2	数字 3
A	1	0	0
B	0	1	0
C	0	0	1
外心	$\cos(A)$	$\cos(B)$	$\cos(C)$
内心	1	1	1
A-旁心	-1	1	1
B-旁心	1	-1	1
C-旁心	1	1	-1
重心	$\frac{1}{a}$	$\frac{1}{b}$	$\frac{1}{c}$
垂心	$\cos(B) \cos(C)$	$\cos(A) \cos(C)$	$\cos(A) \cos(B)$

**案例:** A、B、C 为已知点, “三线坐标点(A, B, C, 1, 1, 1)” 得出 D(D 为其内心)。



**注:** 平面几何中, 关于给定三角形的三线坐标描述了一个点到三角形三条边的相对距离。三线坐标是齐次坐标的一个例子, 经常简称为三线。设 A、B 和 C 不仅表示三角形的顶点, 也表示在相应顶点的角。一些熟知点的三线如上表, 也可以如下描述:

垂心= $\sec A : \sec B : \sec C$ ;

九点圆圆心= $\cos(B-C) : \cos(C-A) : \cos(A-B)$ ;

重心= $bc : ca : ab = 1/a : 1/b : 1/c = \csc A : \csc B : \csc C$ ;

类似重心= $a : b : c = \sin A : \sin B : \sin C$ 。

### 2.8.31 Sector. 扇形

**Sector**(**<Conic>**, **<Point>**, **<Point>**); 扇形(**<圆或椭圆>**, **<点 1>**, **<点 2>**)。

得出一个圆锥曲线上两点之间的扇形。

案例：设“ $c:x^2+2y^2=8$ ”为一个椭圆，“ $D=(-2.83, 0)$ ”和“ $E=(0, -2)$ ”是椭圆上的两个点。“扇形(c, D, E)”得出“ $d=4.44$ ”。



设“ $c:x^2+y^2=9$ ”为一个圆，“ $A=(3, 0)$ ”和“ $B=(0, 3)$ ”是圆上的两个点。“扇形(c, A, B)”得出“ $d=7.07$ ”。



注：本项只能用于圆或椭圆。

**Sector**(<Conic>, <Parameter Value>, <Parameter Value>); 扇形(<圆或椭圆>, <参数值 1\_角度|弧度>, <参数值 2\_角度|弧度> )。

得出一个圆锥曲线上、两个 0 到  $2\pi$  参数值之间的圆锥扇形。

案例：设“ $c:x^2+y^2=9$ ”为一个圆。“扇形(c, 0,  $3/4\pi$ )”得出“ $d=10.6$ ”。



注：内定下列参数形式可用：“圆： $(r \cos(t), r \sin(t))$ ”，其中  $r$  是圆的半径。“椭圆： $(a \cos(t), b \sin(t))$ ”，其中  $a$  和  $b$  是半主轴和半副轴的长度。

## 2.8.32 Ray. 射线


**Ray**(<Start Point>, <Point>); 射线(<起点>, <点>)。

创建一条起于一个点并穿过另一个点的射线。

**Ray**(<Start Point>, <Direction Vector>); 射线(<起点>, <方向向量>)。

创建一条起于给定起点并具有指定方向向量的射线。

注：当计算射线与其他对象的交点时，只考虑位于射线上的交点。若要更改这项设置，可以使用“显示延长线上的交点”选项。

参见  “射线”工具。

## 2.8.33 ArePerpendicular. 是否垂直

**ArePerpendicular**(<Line>, <Line>); 是否垂直(<直线 1>, <直线 2>)。

判断直线是否垂直。

通常情况下这个指令计算数值上的结果。这个判断可以被“证明”指令改变。

案例：“是否垂直(直线((-1,0),(0,-1)),直线((0,0),(2,2)))”得出“true”因为给定的直线互相垂直。

注：参见“是否共线”、“是否全等”、“是否共点”、“是否共圆”、“是否平行”和“是否相等”等指令。





### 2.8.34 AreConcurrent. 是否共点

**AreConcurrent(<Line>,<Line>,<Line>); 是否共点(<直线 1>,<直线 2>,<直线 3>).**

判断直线是否共点。如果线是平行，被认为在无穷远处有共点，如此这个指令返回 true。

通常情况下这个指令计算数值上的结果。这个判断可以被“证明”指令改变。

**案例：**“是否共点(直线((1,2),(3,4)),直线((1,2),(3,5)),直线((1,2),(3,6)))”得出“true”因为三线共点“(1,2)”。



**注：**参见参见“是否共线”、“是否垂直”、“是否共圆”、“是否平行”和“是否相等”等指令。

### 2.8.35 AreCollinear. 是否共线

**AreCollinear(<Point>,<Point>,<Point>), 是否共线(<点 1>,<点 2>,<点 3>).**

判断点是否共线。

通常情况下这个指令计算数值上的结果。这个判断可能被“证明”指令改变。

**案例：**“是否共线((1,2),(3,4),(5,6))”得出“true”，当这些点在同一条直线上。



**注：**参见参见“是否共圆”、“是否全等”、“是否垂直”、“是否共点”、“是否平行”和“是否相等”等指令。

### 2.8.36 AreConcyclic. 是否共圆

**AreConcyclic(<Point>,<Point>,<Point>,<Point>); 是否共圆(<点 1>,<点 2>,<点 3>,<点 4>).**

判断点是否共圆。

通常情况下这个指令计算数值上的结果。这个判断可能被“证明”指令改变。



**案例：**“是否共圆((1,2),(3,4),(1,4),(3,2))”得出“true”因为这些点在同一个圆上。

**注：**参见“是否共线”、“是否全等”、“是否垂直”、“是否共点”、“是否平行”和“是否相等”等指令。

### 2.8.37 AreParallel. 是否平行

**AreParallel(<Line>, <Line>); 是否平行(<直线 1>, <直线 2>。**

判断直线是否平行。

通常情况下这个指令计算数值上的结果。这个结果可被“证明”指令修改。

**案例：**“是否平行(直线((1,2),(3,4)), 直线((5,6),(7,8)))”得出“true”因为给定的直线是平行的。



**注：**参见“是否共线”、“是否共点”、“是否共圆”、“是否全等”、“是否垂直”和“是否相等”指令。

### 2.8.38 AreCongruent. 是否全等

**AreCongruent(<Object>, <Object>); 是否全等(<几何对象 1>, <几何对象 2>。**

判断对象是否全等。

通常情况下这个指令使用代数计算判断，结果可能被“证明”指令修改。

**案例：**指令“是否全等(圆周((0,0),1),  $x^2+y^2=1$ )”得出“true”，因为两个圆有相同的半径。



**注：**参见“是否共线”、“是否共点”、“是否共圆”、“是否平行”、“是否垂直”和“是否相等”等指令。

### 2.8.39 AreEqual. 是否相等

**AreEqual(<Object>, <Object>); 是否相等(<几何对象 1>, <几何对象 2>。**

判断对象是否相等。

一般使用代数计算判断，结果可被“证明”指令修改。



**案例：**指令“是否相等(圆周((0,0),1),  $x^2+y^2=1$ )”得出“true”，因为两个圆有相同的圆心

和半径。

指令“是否相等(线段((1, 2), (3, 4)), 线段((3, 4), (1, 6)))”得出“false”，而“线段((1, 2), (3, 4))  $\perp$  线段((3, 4), (1, 6))”得出“true”因为后者只比较线段长度。

注：参见“是否共线”、“是否共点”、“是否共圆”、“是否平行”、“是否垂直”和“是否全等”等指令。

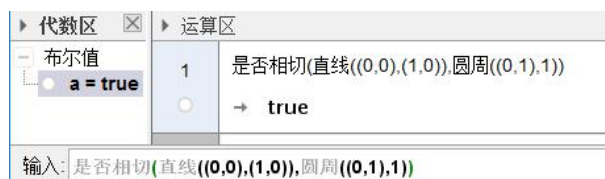
## 2.8.40 IsTangent. 是否相切

**IsTangent(<Line>, <Conic>); 是否相切(<直线>, <圆锥曲线>)**

判断给定直线是否与圆锥曲线相切。

通常情况下这个指令计算数值上的结果。这个结果可被“证明”指令修改。

案例：指“是否相切(直线((0,0), (1,0)), 圆周((0,1), 1))”得出“true”。



注：参见“是否共线”、“是否共点”、“是否共圆”、“是否相等”、“是否平行”、“是否垂直”和“是否全等”等指令。

## 2.8.41 Segment. 线段

这个指令在不同的英语变型中有不同拼写：Interval (Aus)、Segment (UK+US)。

**Segment(<Point>, <Point>); 线段(<点 1>, <点 2>)**。

创建一条两点之间的线段。

**Segment(<Point>, <Length>); 线段(<点>, <长度>)**。

创建一条以指定点为起点，长为指定长度的线段。

注：求线段与其他对象的交点时，只考虑位于线段上的交点。如要改变计算范围，可以勾选属性对话框中的“显示延长线上的交点”选项。

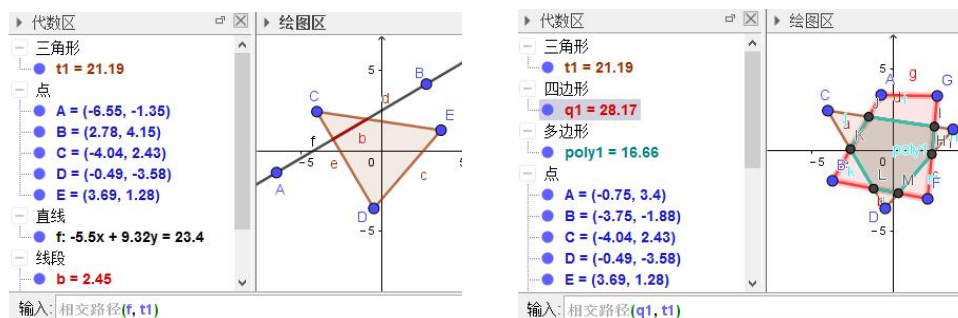
参见  “线段”和  “定长线段”工具。

## 2.8.42 IntersectPath. 相交路径

**IntersectPath(<Line>, <Polygon>); 相交路径(<直线>, <多边形>)**。

创建线和多边形之间的相交路径。

案例：“相交路径(a, 三角形)”创建直线 a 和三角形两个交点间的线段。



**IntersectPath(<Polygon>, <Polygon>); 相交路径(<多边形 1>, <多边形 2>)**。

创建给定的两个多边形相交形成的多边形。

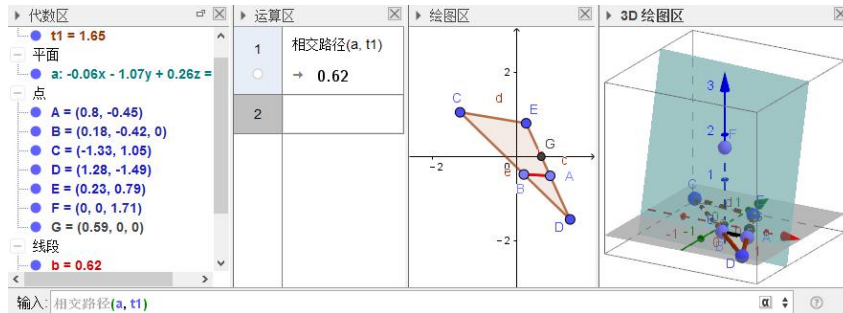
**案例：**“相交路径(四边形,三角形)”创建给定两个多边形的交叉多边形。

**注：**新的多边形可以是四边形，五边形或六边形。这取决于给定的多边形的顶点的位置。

**IntersectPath(<Plane>,<Polygon>); 相交路径(<平面>,<多边形>)。**

创建给定的平面和多边形相交形成的相交路径。

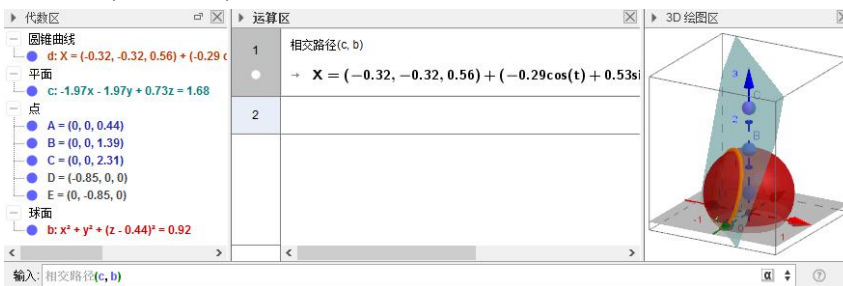
**案例：**相交路径(平面 a,三角形)创建平面 a 和三角形所在平面的多边形的两个交点间的线段。



**IntersectPath(<Plane>,<Quadric>); 相交路径(<平面>,<二次曲面>)。**

创建的平面和二次曲面的相交路径。

**案例：**“相交路径(平面 a,球)”创建作为平面和球交叉的圆。



**注：**参见“交点”和“相交曲线”指令。

### 2.8.43 Slope. 斜率

**Slope(<Line>); 斜率(<直线|射线|线段>)。**

返回给定线的斜率。

**注：**这个指令还会标画出斜率角，其大小可以在属性对话框的样式表单中进行更改。

**注：**参见“斜率”工具。

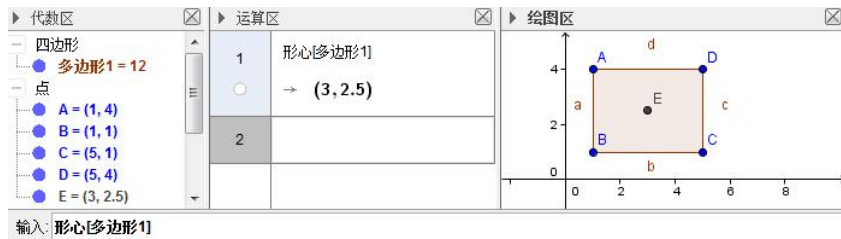
**斜率角的位置：**对于一条使用直线工具或直线指令由点 A 和 B（按此顺序）而定义的直线，斜率角被放置在点 A。对于由输入定义的直线 l（由方程式而得，例如  $1:x+2y=3$ ），斜率角被放置在 y-截距（直线 l 上 x-轴坐标值为零的点）。如果想把斜率角放在其他的位置，可以按这些构建进行操作。①选择直线工具，点击直线 l 上两点从而创建点 C 和 D；将会同时创建一条新的直线。②在创建的直线上使用斜率工具。③隐藏点 D。④移动点 C 以将斜率角调整到适当的位置。

### 2.8.44 Centroid. 形心

**Centroid(<Polygon>); 形心(<多边形>)。**

返回多边形的形心。

**案例：**设多边形的顶点为 A=(1, 4)、B=(1, 1)、C=(5, 1) 和 D=(5, 4)，“多边形(A, B, C, D)”得出“多边形 1=12”，“形心(多边形 1)”得到“O=(3, 2.5)”。



### 2.8.45 CircularArc. 圆弧

**CircularArc(<Midpoint>, <Point A>, <Point B>); 圆弧(<圆心>, <点 1>, <点 2>).**

以圆心为中心点在另外两点之间逆时针创建一条圆弧线。

注：点 B 不一定非要位于弧线之上。英文“CircleArc”也可以。

注：参见“圆弧”工具。

### 2.8.46 CircularSector. 圆扇形

**CircularSector(<Midpoint>, <Point A>, <Point B>); 圆扇形(<圆心>, <点 1>, <点 2>).**

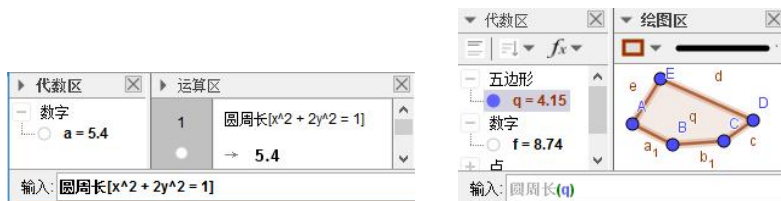
在两个点之间构造圆扇形。

注：点 B 不必在扇形的弧上：参见“扇形”工具。

### 2.8.47 Circumference. 圆周长

**Circumference(Conic); 圆周长(<圆锥曲线>).**

如果给定的圆锥（二次）曲线是一个圆或椭圆，这个指令返回它的圆周长。否则结果为未定义。



**Circumference(Polygon); 圆周长(<多边形>).**

返回一个多边形的周长。

编者注：这个操作在新版本中没有，但功能存在。

### 2.8.48 PolyLine. 折线

**Polyline(<List of Points>); 折线(<点列>).**

创建一个开链多边形（如一串首尾依次连接的线段），其初始顶点是点列中的第一个点，末尾顶点是点列中的最后一个点。

注：折线的长度会在代数区显示。

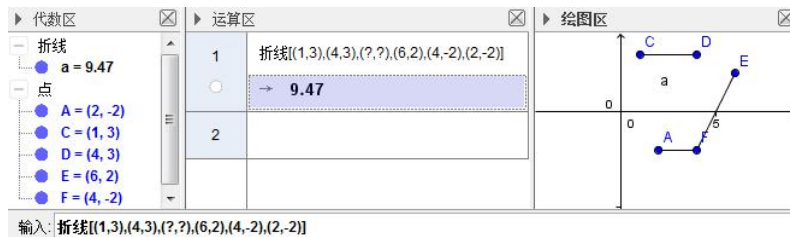
**Polyline(<Point>, ..., <Point>); 折线(<点 1>, ..., <点 n>).**

创建一个开链多边形（如一串首尾依次连接的线段），其初始顶点是首先输入的点，末尾顶点是最后输入的点。

注：折线的长度会在代数区显示。

注：在 4.0 以上的版本，可以创建不连续折线。

案例：“折线((1, 3), (4, 3), (? , ?), (6, 2), (4, -2), (2, -2))” 得出 f=9.47。



注：参见“多边形”指令。

技巧：有时需要一个封闭的折线，如此制作：设有 ABCD 四个点，新增一个坐标与 A 点一致的附点，如折线(A, B, C, D, (x(A), y(A)))，即可构成封闭折线。

## 2.8.49 Prove. 证明

**Prove(<Boolean Expression>); 证明(<布尔表达式>)**。

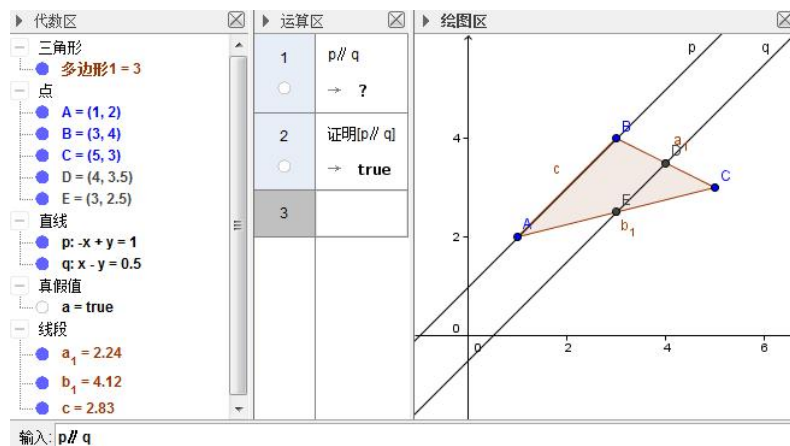
返回给定布尔表达式是否始终是 true 或 false。

通常情况下，GeoGebra 通过数值计算判断一个布尔表达式是真还是假，而“证明”指令使用符号的方法来确定一个语句是否一直是真或假。如果 GeoGebra 不能确定真假，结果是“未定义”。

案例：定义三个自由点，“A=(1, 2)”、“B=(3, 4)”和“C=(5, 6)”指令“是否共线(A, B, C)”得出“true”，因为通过对点的当前坐标进行了数字计算。使用“证明(是否共线(A, B, C))”会得到“false”结果，因为当改变点时，这三个点不总是共线。



案例：使用顶点 A、B 和 C 定义一个三角形，定义“D=中点(B, C)”，“E=中点(A, C)”，“p=直线(A, B)”，“q=直线(D, E)”。现在，“p//q”和“证明(p//q)”都得出“true”，因为三角形的中位线总是平行于对边。



注：参见“证明过程”指令、“布尔值”和“函数与运算”章节。

## 2.8.50 ProveDetails. 证明过程

**ProveDetails(<Boolean Expression>); 证明过程(<布尔表达式>)**。

返回自动证明结果的一些细节。

通常情况下，GeoGebra 通过数值计算判断一个布尔表达式是真还是假，而“证明过程”指令使用符号的方法来确定一个命题是否一直是真或假。这个指令类似“证明”指令，但同时给出结果的细节列表：

如果 GeoGebra 不能确定答案，返回一个空“列表 {}”。

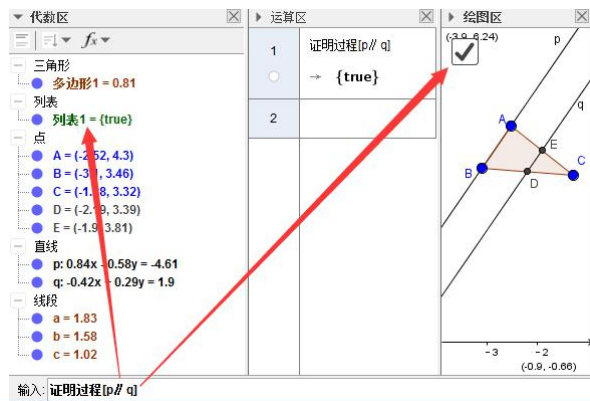
如果不总是“true”，返回一个元素的列表：{false}。

如果命题总是“true”，返回一个元素的列表：{true}。

如果命题在一定条件下“true”，返回一个包括多个元素，包括布尔值 true 和所谓附加条件的列表，诸如：“{true, {“是否共线(A, B, C), 是否全等(C, D)”}}”。这意味着特定条件没有被满足，则命题返回“true”。

如果命题在一定条件下是“true”，但这些条件因故不能转换为可读懂格式。返回“列表{true, {...}}”。

**案例：**使用顶点 A、B 和 C 定义一个三角形，定义“D=中点(B, C)”，“E=中点(A, C)”，“p=直线(A, B)”，“q=直线(D, E)”。现在，如果“证明过程(p//q)”返回“{true, {“是否全等(A, B)”}}”，意味着如果 A 和 B 不重合，三角形的中位线 DE 就平行于边 AB。



注：参见“证明”指令、“布尔值”和“算法技术细节”。

### 2.8.51 Line. 直线

**Line(<Point>, <Point>); 直线(<点 1>, <点 2>).**

创建一条过点 A 和 B 的直线。

**Line(<Point>, <Parallel Line>); 直线(<点>, <平行线>).**

创建过给定点平行于给定直线的直线。

**Line(<Point>, <Direction Vector>); 直线(<点>, <方向向量>).**

创建一条给定点与向量平行的直线。

注：参见“直线”和“平行线”工具。

### 2.8.52 PerpendicularBisector (LineBisector). 中垂线

**PerpendicularBisector(<Segment>); 中垂线(<线段>).**


得出一条线段的中垂线。

**PerpendicularBisector(<Point>, <Point>); 中垂线(<点 1>, <点 2>).**

得出两点间的线段的中垂线。

**PerpendicularBisector(<Point>, <Point>, <Direction>); 中垂线(<点 1>, <点 2>, <方向向量>).**

得出两点间的线段平行于方向的中垂线。“方向”可以是向量、轴、直线或是线段。

注：参见  “中垂线”工具。

## 2.8.53 Midpoint. 中点

**Midpoint(<Segment>); 中点(<线段>)**。

返回线段的中点。

案例：设“s=线段((1, 1), (1, 5))”，“中点(s)”得到“(1, 3)”。



**Midpoint(<Conic>); 中点(<圆锥曲线>)**。

返回圆锥曲线的中心点。

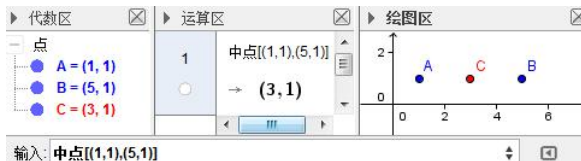
案例：“中点( $x^2+y^2=4$ )”得到“(0, 0)”。

**Midpoint(<Interval>); 中点(<区间>)**。

返回区间的中点（数值对象）。

**Midpoint(<Point>, <Point>); 中点(<点 1>, <点 2>)**

返回两个点的中心点。




案例：“中点((1, 1), (5, 1))”得到“(3, 1)”。

**Midpoint(<Quadric>); 中点(<二次曲面>)**。

返回给定二次曲面（如球面、椎体等）的中心点。

编者注：这个操作在新版本中没有，但功能存在。

案例：指令“中点( $x^2+y^2+z^2=1$ )”得出点“(0, 0, 0)”

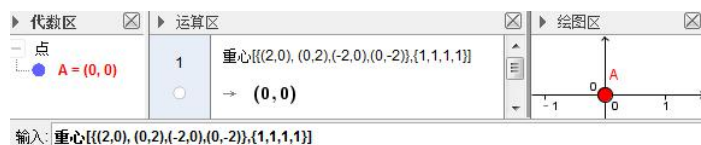
注：参见  “中点/中心”工具。

## 2.8.54 Barycenter. 重心

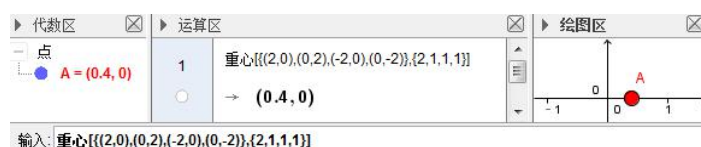
**Barycenter(<List of Points>, <List of Weights>); 重心(<点列>, <权重列表>)**

使用一定的格式对位置进行加权平均计算，设置系统点列的中心点。

案例：“重心({(2, 0), (0, 2), (-2, 0), (0, -2)}, {1, 1, 1, 1})”得出点“A(0, 0)”。



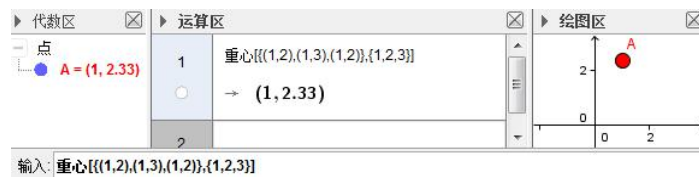
“重心({(2, 0), (0, 2), (-2, 0), (0, -2)}, {2, 1, 1, 1})”得出点“A(0.4, 0)”。





点的 x 坐标计算为： $\frac{1}{2+1+1+1} * (2*2+1*0+1*(-2)+1*0) = \frac{1}{5} * 2 = 0.4$ 。

案例：“重心({(1, 2), (1, 3), (1, 2)}, {1, 2, 3})” 得出点 A(1, 2.33)。



## 2.8.55 Perimeter. 周长

**Perimeter(<Polygon>); 周长(<多边形>)**。

返回多边形的周长。

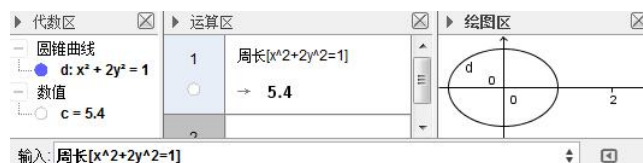
案例：“周长(多边形((1, 2), (3, 2), (4, 3)))” 得出 6.58。



**Perimeter(<Conic>); 周长(<圆锥曲线>)**。

如果指定的圆锥曲线是一个圆形或椭圆形，这个指令返回对象的周长。否则是未定义。

案例：“周长( $x^2+2y^2=1$ )” 得出 5.4。



**Perimeter(<Locus>); 周长(<轨迹>)**。



如果给定的轨迹是有限的（由有限的点构成），这个指令返回它的等效（近似）周长。否则结果是未定义。

## 2.8.56 ClosestPoint. 最近点

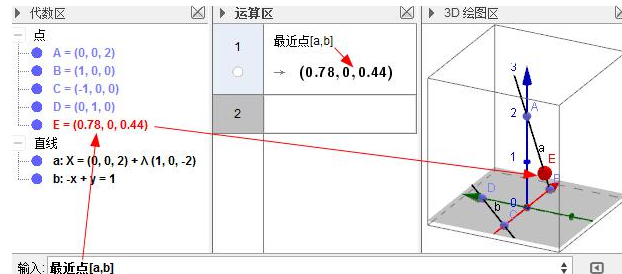
**ClosestPoint(<Path>, <Point>); 最近点(<路径>, <点>)**。

返回一条路径上最接近所选点的一个新点。

**注：**对于函数类型的对象，这个指令现在使用的是最接近的点（而不是垂直点）。最适用于多项式，对其他的函数使用时，这个指令的数值算法较不稳定。



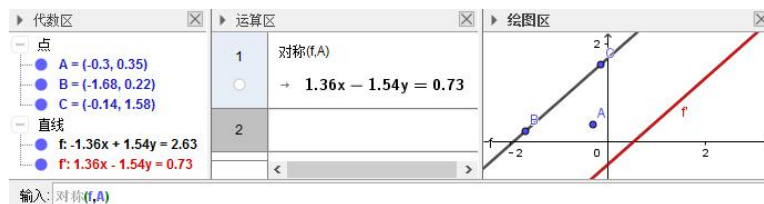
**ClosestPoint(<Line>, <Line>); 最近点(<直线 1>, <直线 2>)**。  
返回第一条直线上最接近于第二条直线的点。



## 2.9 Transformation. 几何变换

### 2.9.1 Reflect (Mirror) . 对称

**Reflect(<Object>, <Point>); 对称(<几何对象>, <对称中心点>)**。  
将几何对象以指定点为对称点进行镜像反射。



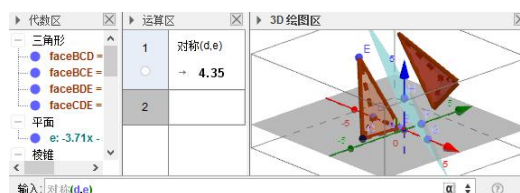
注：多边形关于一个对称点进行反射时，会同时创建反射后的顶点和线段。

**Reflect(<Object>, <Line>); 对称(<几何对象>, <对称轴 直线|射线|线段>)**。  
以指定直线为对称轴，将对象（例如图片）进行镜像反射。



注：多边形关于一条对称轴进行反射时，会同时创建反射后的顶点和线段。

**Reflect(<Object>, <Plane>); 对称(<几何对象>, <对称平面>)**。  
关于平面反射几何对象。  
注：就是平面镜像对称。



注：参见“中心对称”、“轴对称”、“平面对称”和“反演”工具。

**Reflect(<Object>, <Circle>); 对称(<几何对象>, <反演基圆>)**。

几何对象关于指定圆的反演图像。



注：参见 “中心对称”、 “轴对称”和 “反演”工具。

## 2.9.2 Translate. 平移

**Translate(<Object>,<Vector>); 平移(<几何对象>,<向量>)**。

按向量移动几何对象。

注：平移一个多边形时，会同时创建平移后的顶点和线段。

**Translate(<Vector>,<Start Point>); 平移(<向量>,<起点>)**。

将向量移动到新的起点。

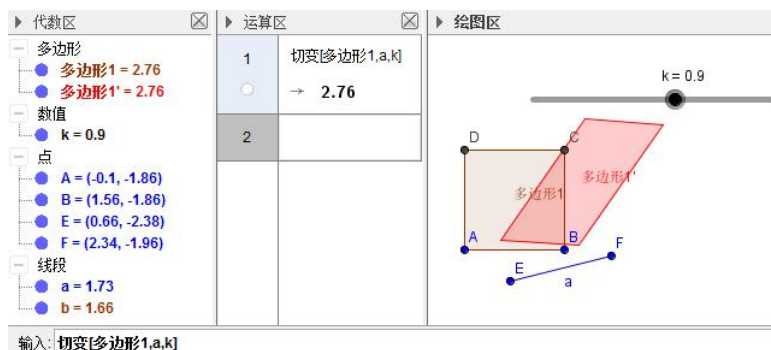
注：参见 “平移”工具。

## 2.9.3 Shear. 切变

**Shear(<Object>,<Line>,<Ratio>); 切变(<几何对象>,<直线|射线|线段>,<比率>)**。

切变一个对象将会：对象上的点与直线的距离保持不变；到直线距离为  $d$  的点沿直线方向平移“ $d \times$ 比率”距离（转换的方向是关于“直线|射线|线段”的不同半平面）。

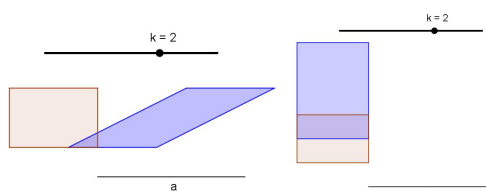
一个切变平面会保持原有面积。



本例中，“多边形 1”是正方形，是要切变的对象，线段  $a$  决定了切变的方向， $k$  是变换比。

注：由矩阵  $M = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}$  或  $\begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}$  确定的变换称为切变变换，对应的矩阵称为切变变换矩阵。

以  $\begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}$  为例，它把平面上的点  $(x,y)$  沿  $x$  轴方向平移  $|ky|$  个单位，当  $ky > 0$  时沿  $x$  轴正方向移动，当  $ky < 0$  时沿  $x$  轴负方向移动，当  $ky = 0$  时原地不动。切变变换有如下性质：①  $x$  轴（或  $y$  轴）上的点是不动点；② 保持图形面积大小不变，点间的距离和夹角大小可以改变且点的运动是沿坐标轴方向进行的。切变变换的实质是横（纵坐标）成比例地运动。

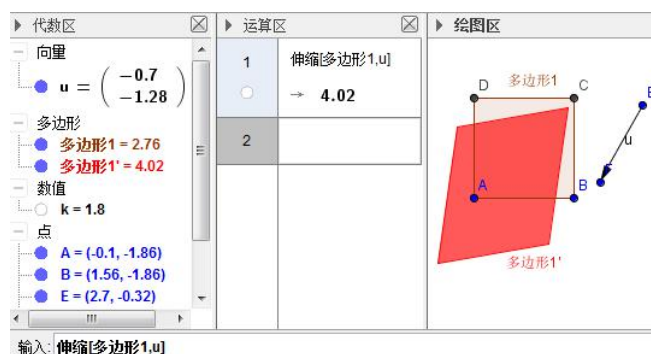


左图一为切变(多边形  $1,a,k$ )，  
左图二为伸缩(多边形  $1,a,k$ )。  
蓝色为变换后的图形。

## 2.9.4 Stretch. 伸缩

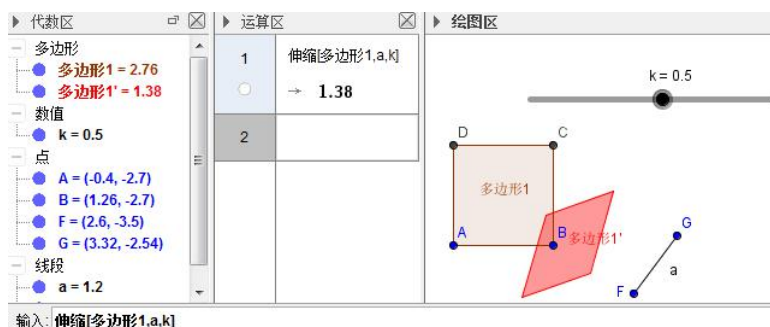
**Stretch(<Object>,<Vector>); 伸缩(<几何对象>,<向量>)**。

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，则伸缩后位置不变；其他的构成点与起点垂线之间的距离，伸缩后转换为原距离乘以给定比率）。



**Stretch(<Object>,<Line>,<Ratio>); 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（对象的全部构成点中，如果其位置在指定直线方向上，则伸缩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，伸缩后转换为原距离乘以指定比率）。



## 2.9.5 Dilate (Enlarge). 位似


这个指令在不同的英语变型中拼写不同：Dilate (US)、Enlarge (UK+Aus)。

**Dilate(<Object>,<Dilation Factor>); 位似(<几何对象>,<位似比例>)**。

以原点为位似中心将对象按指定的位似比例进行位似。

**Dilate(<Object>,<Dilation Factor>,<Dilation Center Point>); 位似(<几何对象>,<位似比例>,<位似中心点>)**。

以指定的位似中心将对象按给定的位似比例进行缩放。

注：位似一个多边形时，GeoGebra 会同时创建位似后多边形的全部顶点和线段；参见  “位似”工具。

## 2.9.6 Rotate. 旋转

**Rotate(<Object>,<Angle>); 旋转(<几何对象>,<角度|弧度>)**。


将几何对象围绕坐标原点旋转指定角度。

**Rotate(<Object>,<Angle>,<Point>); 旋转(<几何对象>,<角度|弧度>,<旋转中心>)**。

将几何对象围绕给定点旋转指定角度。

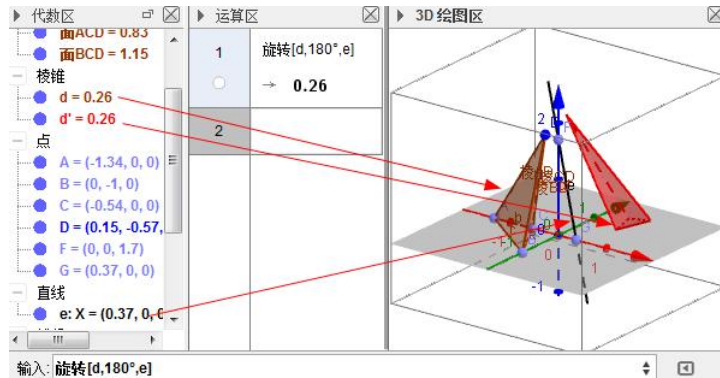
**注：**向量不能围绕坐标原点旋转，但可以绕起点旋转；当旋转一个多边形、线段或圆弧，同样创建顶点/端点和边（本例的多边形）镜像。这个指令也能旋转图片。

旋转文本使用“旋转文本”指令。

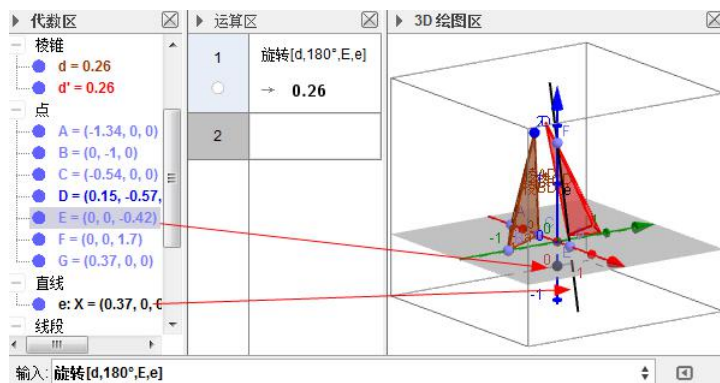
参见  “旋转”指令。

**Rotate(<Object>,<Angle>,<Axis of Rotation>);** 旋转(<几何对象>,<角度|弧度>,<旋转轴>)。

将几何对象绕给定的旋转轴旋转指定的角度。



**Rotate(<Object>,<Angle>,<Point on Axis>,<Axis Direction or Plane>);** 旋转(<几何对象>,<角度|弧度>,<轴上的点>,<轴方向或平面>)。



## 2.10 ScriptingCommands. 脚本指令

顾名思义，脚本指令是为编程准备的，属于“后台”执行的指令，但许多脚本指令在“指令栏”和运算区也能执行，故许多脚本指令案例的截图，就使用指令栏的执行结果。另外，许多指令栏指令也能用于脚本中，如“ $f(x)=x^2$ ”或“ $c=$ 圆周 $((0,0),5)$ ”等。

### 2.10.1 Button. 按钮

**Button();** 按钮( )。

创建一个新按钮。


**Button(<Caption>);** 按钮(“<标题>”)

创建给定标题的按钮。



**案例：**“按钮(“开始”)”创建一个位于绘图区左上角，标题为“开始”的按钮。（至于此按钮

执行的动作，需要在此按钮的“属性”中的“脚本”中设置。）

注：参见  “按钮”工具。

## 2.10.2 PlaySound. 播放声音

**PlaySound(<URL>); 播放声音(<网址>)。**

播放一个 MP3 文件 (\*.mp3)。

**案例：**“播放声音(“http://archive.geogebra.org/static/welcome\_to\_geogebra.mp3”)”；

“播放声音(“#1264825”)”，播放一个传到 GeoGebra 的 MP3；

“播放声音(“https://drive.google.com/uc?id=0B7xCmZaU3oU2eXFNUzd6Z1ZJSOU&authuser=0&export=download”)”，播放谷歌的一个 MP3；

“播放声音(“https://www.dropbox.com/s/27skpv82odjp7ej/material-1264825.mp3?dl=1”)”，播放 DropBox 的 MP3。

**PlaySound(<Boolean Play>); 播放声音(<是否播放? true|false>)。**

暂停或重新播放。

播放声音(true)=播放,播放(false)=暂停。

**PlaySound(<Note Sequence>, <Instrument>); 播放声音(<音符顺序>, <乐器>)。**

播放由特定乐器演奏的一串音符。

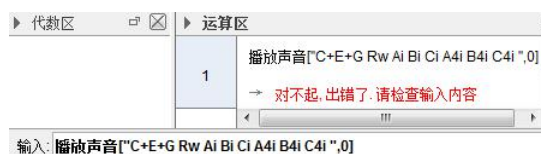
音符顺序是使用 JFugue 符指令组成的文本串。

乐器是播放音符串时使用的默认乐器。

基本指令见下表。全部指令设置在 JFugue 大全(英语)中有介绍。

基本 JFugue 指令符	指令
A-G	通过这些字符播放音乐;用 b 和#实现升调和降调; 增加一个数字改变八度, e. g. A4; 默认音符长度为 4。
(number)	通过数字 (0-127) 播放 MIDI 音符。
+	同时播放多个音符 e. g. C+E+G。
R	停止。
w, h, q, i, s	追加在最后或暂停时; 设置音符长度: w=全音符 h=半音符 q=四分音符 i=八分音符 s=十六分音符。
/n	追加到一个音符或休止; 集长度为 n (十进制数): /1=全音符/.5=半音符/.25=四分音符/2=twomeasures。
I(number)	设置乐器。
V	V 后跟一个数字从 0 到 15 台 MIDI 通道。V9 设置 MIDI 乐器通道。
Space	播放前面的音符、音符组合或暂停

**案例：**“播放声音(“C+E+G Rw Ai Bi Ci A4i B4i C4i”,0)”演奏四分之一节 CEG 合奏; 休止一节; 演奏八分之一节 A, B, C; 低八度再演奏一次。



使用钢琴乐器。

注：尽管在运算区显示出错信息，但也有正常声音输出。

案例：“播放声音(“I(56) C5q D5q I(71) G5q F5q”,0)”使用不同乐器播放音符。喇叭声=56和单簧管=71。

案例：“播放声音(“V0 A3q B3q C3q B3q V1 A2h C2h”,0)”不同声音合奏。

PlaySound(<Note>,<Duration>,<Instrument>); 播放声音(<音符>,<持续时间>,<乐器>)。

播放一个音符。

音符是在下表中列出代表音乐符号的 0 到 127 间的一个整数。当音符=60 时，演奏中音 C。

持续时间是演奏音符的时间。

乐器是代表演奏音符合成乐器的一个整数。见可能乐器的技术说明。

多数乐器都支持，但在不同的电脑平台有差异。

MIDI 符 八度音阶	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	51	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

PlaySound(<Function>,<Min Value>,<Max Value>); 播放声音(<函数>,<最小值>,<最大值>)。

通过范围在(-1, 1)的时间值函数播放一个声音。时间单位是秒且声音从最小时间到最大。声音采用 8 位采样变换率为每秒 8000 样本。

案例：“播放声音(tan(1/(sin(10 x^2))),0,10)”播放 10 秒貌似摩托车启动的声音。

“播放声音(sin(440 2Pi x),0,1)”播放 1 秒钟 440 赫兹（音符 A）纯正弦波音。

PlaySound(<Function>,<Min Value>,<Max Value>,<Sample Rate>,<Sample Depth>); 播放声音(<函数>,<最小值>,<最大值>,<采样率>,<样本深度>)。

通过范围在(-1, 1)的时间值函数播放一个声音。时间单位是秒且声音从最小时间到最大。采样方法由“样本深度”和“采样率”决定。

“采样率”是采样函数值每秒变化数值。允许数值为 8000、11025、16000、22050 或 44100。

“样本深度”每节样本数据大小。允许值为 8 和 16。

以下为国际通用 GM 音色表：

#### 钢琴类 (Piano)

音色号	英文	英文简写	中文
-----	----	------	----

1	AcousticGrandPiano	GrandPno	原声大钢琴
2	BrightGrandPiano	Britepno	亮音钢琴
3	ElectricGrandPiano	E. GrandPno	电子大钢琴
4	Honky-tonkPiano	HonkyTonk	酒吧钢琴
5	ElectricPiano1	E. Piano1	电钢琴 1
6	ElectricPiano2	E. Piano2	电钢琴 2
7	Harpsichord	Harpsi	拨弦古钢琴
8	Clavinet	Clavi	电子击弦古钢琴

半音性打击乐器 (ChromaticPercussion)

9	Celesta	---	钢片琴
10	Glockenspiel	Glocken	钟琴
11	MusicBox	---	八音盒
12	Vibraphone	Vibes	颤音琴
13	Marimba	---	马林巴
14	Xylophone	Xylophon	木琴
15	TubularBells	TubulBel	管钟
16	Dulcimer	---	扬琴

风琴类 (Organ)

17	HammondOrgan	HarhondOrgn	哈蒙德风琴
18	PercussiveOrgan	PercOrgn	击音管风琴
19	RockOrgan	RockOrgn	摇滚风琴
20	ChurchOrgan	ChurchOrgn	教堂管风琴
21	ReedOrgan	ReedOrgn	簧片风琴
22	Accordion	---	手风琴
23	Harmonica	---	口琴
24	TangoAccordion	TangoAcid	探戈手风琴

吉它类 (Guitar)

25	AcousticGuitar (nylon)	NylonGtr	尼龙弦吉它
26	AcousticGuitar (steel)	SteelGtr	钢弦吉它
27	ElectricGuitar (jazz)	JazzGtr	爵士吉它
28	ElectricGuitar (clean)	CleanGtr	纯音吉它
29	ElectricGuitar (muted)	Mute. Gtr	闷音吉它
30	OverdrivenGuitar	Overdriver	激励音吉它
31	DistortionGuitar	Dist. Gtr	失真吉它
32	GuitarHarmonics	GtrHarmo	吉它泛音

贝司类 (Bass)

33	AcousticBass	Aco. Bass	原声贝司
34	ElectricBass (finger)	FngeBass	指弹电贝司
35	ElectricBass (pick)	PickBass	拨片电贝司
36	FretlessBass	Fretless	无品贝司
37	StapBass1	---	打弦贝司 1
38	StapBass2	---	打弦贝司 2



39	SynthBass1	SynBass1	合成贝司 1
40	SynthBass2	SynBass2	合成贝司 2

弦乐器 (Strings)

41	Violin	---	小提琴
42	Viola	---	中提琴
43	Cello	---	大提琴
44	Contrabass	Contrabs	低音提琴
45	TremoloStrings	Trem. Str	弦乐震音
46	PizzicatoStrings	Pizz. Str	弦乐拨音
47	OrchestralHarp	Harp	竖琴
48	Timpani	---	定音鼓

合奏 (唱) 组 (Ensemble)

49	StringEnsemble1	Strings1	弦乐组 1
50	StringEnsemble2	Strings2	弦乐组 2
51	SynthStrings1	Syn. Str1	合成弦乐组 1
52	SynthStrings2	Syn. Str2	合成弦乐组 2
53	ChoirAahs	Choir. Aah	唱诗班啊声
54	VoiceOohs	VoiceOoh	哦声合唱
55	SynthVoice	Synvoice	合成人声
56	OrchestraHit	Orch. Hit	管弦乐齐奏

铜管乐器 (Brass)

57	Trumpet	---	小号
58	Trombone	---	长号
59	Tuba	---	大号
60	MutedTrumpet	Mute. Trp	小号加弱音器
61	FrenchHorn	Fr. Horn	法国号
62	BrassSection	BrasSect	铜管组
63	SynthBrass1	SynBras1	合成铜管 1
64	SynthBrass2	SynBras2	合成铜管 2

簧片乐器 (Reed)

65	SopranoSax	SprnoSax	高音萨克斯
66	AltoSax	---	中音萨克斯
67	TenorSax	---	次中音萨克斯
68	BaritoneSax	Bari. Sax	低音萨克斯
69	Oboe	---	双簧管
70	EnglishHorn	Eng. Horn	英国管
71	Bassoon	---	大管
72	Clarinet	---	单簧管

管鸣乐器 (Pipe)

73	Piccolo	---	短笛
----	---------	-----	----

74	Flute	---	长笛
75	Recorder	---	竖笛
76	PanFlute	---	牧笛
77	BottleBlow	Bottle	瓶笛
78	Shakuhachi	Shakhchi	尺巴
79	Whistle	---	口哨
80	Ocarina	---	陶笛

合成领奏 (SynthLead)

89	Pad1(newage)	NewagePd	背景 1 (新时代)
90	Pad2(warm)	WarmPd	背景 2 (温暖的)
91	Pad3(polysynth)	PolySyPd	背景 3 (复合合成)
92	Pad4(choir)	ChoirPad	背景 4 (唱诗班)
93	Pad5(bowed)	BowedPad	背景 5 (弓弦音色)
94	Pad6(metallic)	MetalPad	背景 6 (金属般)
95	Pad7(halo)	HaloPad	背景 7 (问候)
96	Pad8(sweep)	SweepPad	背景 8 (宽阔的)

合成效果 (SynthEffects)

97	FX1(rain)	Rain	效果 1 (下雨)
98	FX2(soundtrack)	SoundTrk	效果 2 (音轨)
99	FX3(crystal)	Crystal	效果 3 (晶体)
100	FX4(atmosphere)	Atmosphr	效果 4 (气氛)
101	FX5(brightness)	Bright	效果 5 (明亮)
102	FX6(goblins)	Goblins	效果 6
103	FX7(echoes)	Echo	效果 7 (回声)
104	FX8(sci-fi)	Sci-Fi	效果 8

民间乐器 (Ethnic)

105	Sitar	---	西塔尔
106	Banjo	---	班卓
107	Samisen	---	三味线
108	Koto	---	日本筝
109	Kalimba	---	卡林巴
110	Bagpipe	---	风笛
111	Fiddle	---	小提琴
112	Shanai	---	山奈

打击乐 (Percussive)

113	TinkleBell	TnklBell	铃铛
114	Agogo	---	阿果果
115	SteelDrums	SteelDrm	钢鼓
116	Woodbiock	Woodblok	帮子
117	TaikoDrom	TaikoDrm	太叩鼓

118	MelodicTom	MelodTom	旋律性通通鼓
119	SynthDrum	Syn. Drm	合成鼓
120	ReverseCymbal	RevCymbal	反钹

#### 音响效果 (SoundEffects)

121	GuitarFretNoise	FretNoiz	吉它滑品噪音
122	BreathNoise	Brthnoiz	呼吸声
123	Seashore	---	海浪声
124	BirdTweet	Tweet	鸟叫
125	TelephoneRing	Telephone	电话铃声
126	Helicopter	Helicptr	直升飞机声
127	Applause	---	掌声
128	Gunshot	---	枪声

### 2.10.3 ExportImage. 导出图片

**ExportImage**(**<Property>**,**<Value>**,**<Property>**,**<Value>**,...); 导出图片(**<属性 1>**,**<属性值 1>**,**<属性 2>**,**<属性值 2>**,...).

导出当前活动绘图区域的图片 (或 “view” 参数指定的绘图区域)。

**注:** 以下列示的命令语法中, 必须使用半角英文。与区域相关的图片参数约定为: 绘图区为 1、绘图区 2 为 2、3d 绘图区为 -1。

#### 案例:

“导出图片 (“scale”, 5)” : 当前活动绘图区的图片比例为 5。

显示当前视图的弹出窗口, 以使用户可以右键单击 -> 将图像另存为...

“导出图片 (“filename”, “image.png”): 当前绘图区图片以文件 “image.png” 存放。

“导出图片 (“filename”, “image.png”, “view”, 2)” : 绘图区 2 的图片以文件名 “image.png” 存放。

“导出图片 (“filename”, “image.png”, “view”, -1)” : 3d 绘图区的图片以文件名 “image.png” 存放。

“导出图片 (“filename”, “image.png”, “dpi”, 300)” : 当前绘图区图片以文件名 “image.png” 存放, 300 点格式。

“导出图片 (“filename”, “image.png”, “scale”, 2)” : 以比例 2 (即标称屏幕分辨率的两倍) 保存当前视图的名为 “image.png” 的文件。

“导出图片 (“filename”, “image.png”, “scalecm”, 2, “dpi”, 600)” : 以比例 2 (即标称屏幕分辨率的两倍) 保存当前视图的名为 “image.png” 的文件, 600 点, 比例为 1 个单位 2 厘米。

“导出图片 (“filename”, “image.png”, “width”, 1000)” : 保存当前视图的名为 “image.png” 的文件, 宽度 1000 像素。

“导出图片 (“filename”, “image.png”, “height”, 1000)” : 保存当前视图的名为 “image.png” 的文件, 高度 1000 像素。

“导出图片 (“filename”, “image.png”, “transparent”, true)” : 保存当前视图的名为 “image.png” 的透明图片文档。

“导出图片 (“filename”, “image.png”, “type”, “svg”): 保存当前视图的名为 “image.png” 的 SVG 格式文件。

“导出图片 (“filename”, “image.gif”, “type”, “gif”, “slider”, a, “loop”, true, “time”, 200, ”

width",400)”:保存由滑动条参数“a”控制的当前视图的循环动画 GIF,帧间距为 200ms。

**滑动条参数宽度和数值需要保持较小增量。**

“导出图片(filename,"image.gif","type","gif","view",-1,"rotate",360,"slider",a,"loop",true,"time",200)”:保存由滑块“a”控制的 3d 绘图区的循环动画 GIF,帧间距为 200ms,并在动画期间将视图旋转 360°。

**视区大小和滑动条参数保持较小的增量。**

“导出图片(type,"pdf",filename,"test.pdf)”:创建当前视图的 PDF(如果图形视图 2 打开,则为 2 页 PDF)。

“导出图片(type,"pdf",filename,"test.pdf","slider",n)”:创建当前视图的多页 PDF,其中每个页面对应滑动条“n”的一个步骤。

“pic1=导出图片(view,2,"corner",A,"corner2",B)”:创建绘图区 2 自角点 A 到角点 B 区域的图片。

注:在 GeoGebra 经典 5 中,如果没有文件名参数,图片将被复制到剪贴板。GeoGebra5 不支持某些语法。对于 3D 绘图区,仅支持 PNG 格式。

## 2.10.4 ZoomIn. 放大

**ZoomIn(<Scale Factor>); 放大(<缩放因子>)。**

基于当前比例将绘图区视图以指定缩放因子进行放大,屏幕中心为缩放中心点。

**案例:**“放大(1)”不做改变;“放大(2)”放大视图;“放大(0.5)”等同于“缩小(2)”,缩小视图。

**ZoomIn(<Scale Factor>,<Center Point>); 放大(<缩放因子>,<中心点>)。**

基于当前比例将绘图区视图以指定缩放因子进行放大,第二个参数指定了缩放中心点。

**案例:**“放大(2,(0,0))”就是以原点为中心,放大视图。

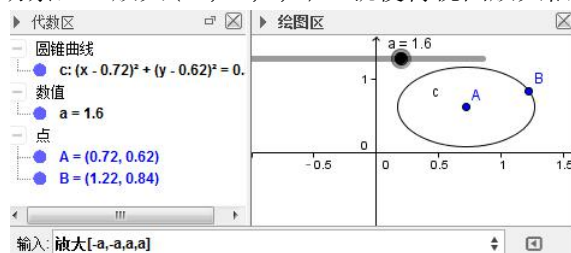
**ZoomIn(<Min x>,<Min y>,<Max x>,<Max y>); 放大(<x 最小值>,<y 最小值>,<x 最大值>,<y 最大值>)。**

将绘图区视图缩放为由顶点(最小 x,最小 y)、(最大 x,最大 y)所确定的矩形区域。

**案例:**“放大(0,1,5,6)”将视图区域放大到角点自(0,1)到(5,6)的矩形内。

**注:**如果这里的参数是从属对象或具有标签设定(不是未定义数值),视图的边界会变为动态。若要避免这种情况,可使用复制自由对象指令。

**案例:**如 a 是一个滑动条,“放大(-a,-a,a,a)”就使得视图放大依赖于滑动条 a。

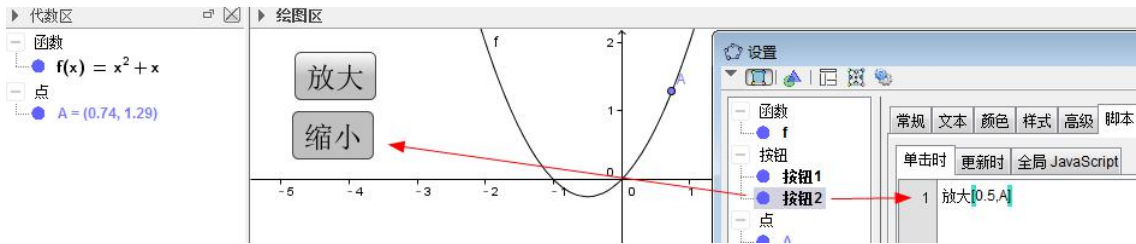


**注:**如果存在多个绘图区视图,则本指令用于激活使用的那个视图。

参见“缩小”、“活动视图”、“平移视图”指令。

**案例:**放大的使用技巧-由切线来线性近似(逼近)函数。

创建一个点 A 在其之上的函数 f,以及一个编程代码(脚本)为“放大(2,A)”的按钮,可以观察到,当放大到一个相当大的“放大倍率”时,函数看上去相当于这个点上的切线。在不同点 A 的位置上使用另一个具有缩小指令的按钮,能让证实这种情况。



## 2.10.5 SetValue. 赋值

**SetValue(<Boolean>, <0|1>); 赋值(<布尔对象>, <0|1>)。**

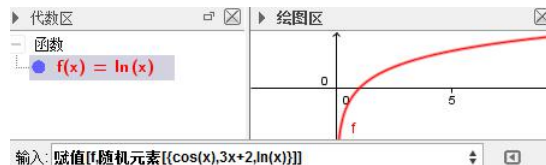
给一个布林型对象/复选框赋值: 1=true, 0=false。

**案例:** 如果 b 是布林型对象, “赋值(b, 1)” 设定布林型对象 b 为 “true”。

**SetValue(<Object>, <Object>); 赋值(<对象 1>, <对象 2>)。**

假设 A 是第一个对象, B 是第二个对象。如果 A 是自由对象或是一个附着在路径或区域上的点, 则取对象 2 的当前值 (也就是, 此后对象 1 的值并不会再因为对象 2 的值发生改变而改变)

**案例:** 如果 f 是一个函数, 使用 “赋值(f, 随机元素({cos(x), 3x+2, ln(x)}))” 定义随机量, 函数 f 会是列表中提供的函数之一。

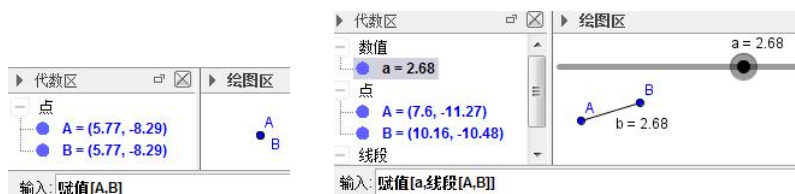


**注:** 因为是 “当前值”, 在指令栏重复执行这个指令, 函数 f 才会随机变为列表中的一个函数。

“赋值(A,B)” 是使用点 B 的值作为当前 A 的值, 但点 B 移动后, 点 A 的值不随着变动;

“赋值(A, (0, 0))”, 给点 A 的当前坐标设定为 (0, 0);

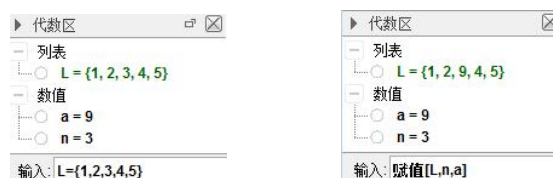
“赋值(a, 线段(A, B))” 将线段 AB 的当前值赋予变量 a, 变为 a 的当前值, 但 a 可以改变。当 A 或 B 改变后, a 并不随着变。



**SetValue(<List>, <Number>, <Object>); 赋值(<列表>, <列表索引>, <对象>)。**

设 n 是 “数值” (列表索引)。赋值指令设置自由列表中第 n 个元素的值为对象的当前值。 “数值” 最大为列表 L 长度+1。

**案例:** 若 L 是一个列表型对象, 值为 {1, 2, 3, 4, 5}; n 是一个数值型对象, 值为 3; a 是一个值为 9 的对象。 “赋值(L, n, a)” 将列表 L 的值设置为 {1, 2, 9, 4, 5}。



**注:** 如果想要在脚本中更改一个对象的值, 尽量使用 “赋值” 指令而不是 “=”。

## 2.10.6 Checkbox. 复选框

**Checkbox()**; 复选框()。

创建一个复选框。

**Checkbox(<Caption>)**; 复选框("<标题>")。

创建给定标题的复选框。

**Checkbox(<List>)**; 复选框(<列表>)。

创建一个复选框，其不被勾选时，隐藏列表中的对象。

**案例：**设点 A 和点 B。“复选框({A,B})”创建复选框 c。当 c 被点击，A 和 B 显示，否则，隐藏。

**Checkbox(<Caption>, <List>)**; 复选框("<标题>", <列表>)。

创建一个指定标题的复选框，其不被勾选时，隐藏列表中的对象。



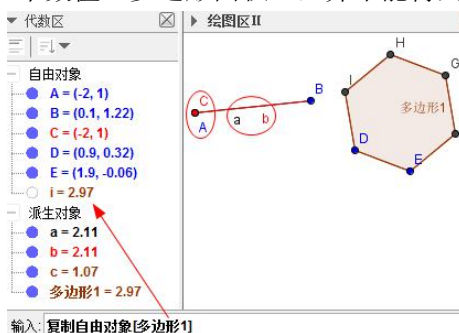
**注：**参见“复选框”工具；如果两次复选框包括了相同的对象，那么这个对象只受后面创建的复选框控制。

## 2.10.7 CopyFreeObject. 复制自由对象

**CopyFreeObject(<Object>)**; 复制自由对象(<对象>)。

创建一个自由对象的副本，携带对象的所有基本属性且辅助对象继续保持为辅助对象。

**案例：**有自由点 A、B，派生对象线段 AB (a) “复制自由对象(A)”和“复制自由对象(a)”都能得到源对象相同位置的一个新对象分别是点 C 和线 b，这两个新对象是自由对象。但如果“复制自由对象(多边形 1)”会得到一个数值 (多边形面积)，并不能得到一个新的自由的多边形。



## 2.10.8 AttachCopyToView. 附加副本

**AttachCopyToView(<Object>, <View 0|1|2>)**; 附加副本(<对象>, <视图 0|1|2>)。

如果视图=0，会创建给定对象一个副本。当“视图=1”或“视图=2”时，这个指令在指定绘图区中创建给定对象大小不变的副本。

**案例：**设“poly=多边形((0,0), (1,0), (1,1), (0,1))”。如果绘图区被激活，“附加副本(poly, 1)”在同样的位置创建一个与原多边形大小相同的正方形。



### 参数意义:

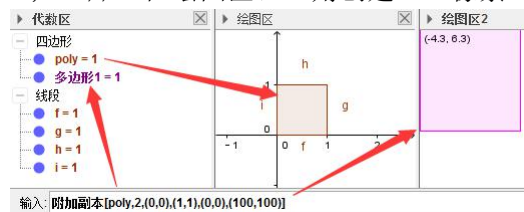
- 0: 副本出现在绘图区, 跟随画面缩放变化, 跟随对象移动;
- 1: 副本出现在绘图区, 不跟随画面缩放变化, 只跟随对象移动;
- 2: 副本出现在绘图区 II, 不跟随画面缩放变化, 只跟随对象移动。

注: 一旦副本被创建, 另外三个(位置)参数被追加到指令的定义之中。

**AttachCopyToView(<Object>, <View 0|1|2>, <Point 1>, <Point 2>, <Screen Point 1>, <Screen Point 2>); 附加副本(<对象>, <视图 0|1|2>, <点 1>, <点 2>, <屏幕点 1>, <屏幕点 2>)**。

如果视图=0, 会创建给定对象一个副本。当“视图=1”或“视图=2”时, 这个指令创建在指定绘图区中大小变化为点 1 绑定到屏幕点 1, 点 2 绑定到屏幕点 2 的给定对象副本。

案例: 设“poly=多边形((0, 0), (1, 0), (1, 1), (0, 1))”。如果绘图区被激活, “附加副本(poly, 2, (0, 0), (1, 1), (0, 0), (100, 100))”在绘图区左上角创建 100 像素 x100 像素的正方形。



注: 即使对象的某些部分在角点 1 和 3 的矩形框选范围以外, 其所有部分都会被映射。

视图参数的意义:

- 0: 在原对象位置复制出对象, 复制出的对象的大小和位置跟随原对象, 后面的对应关系无作用。
- 1: 在正确的位置复制出对象, 复制出的对象的大小不会缩放, 位置跟随原对象。
- 2: 在偏移了一个正确的位置的地方复制出对象, 复制出的对象的大小会缩放, 位置跟随原对象。

后面四个点的意义:

<点 1>, <点 2>, <屏幕点 1>, <屏幕点 2>表示的是一种对应关系。其中点 1 与屏幕点 1 对应, 点 2 与屏幕点 2 对应。屏幕点的坐标系是以视图的左上角为坐标原点, 向右是 x 轴正方向, 向上是 y 轴正方向, 单位是像素。

恰当的对对应关系例如 (0, 0), (10, -10), (0, 0), (100, 100)

对象向上复制对象也向上, 对象向右复制对象跟随向右。

如果对应关系直接写为 (0, 0), (10, 10), (0, 0), (100, 100)

对象向上复制对象也向下, 对象向右复制对象跟随向右。

对应关系中尽量不使用点对象(例如 A), 那样对应关系会变得非常复杂。

## 2.10.9 UpdateConstruction. 更新作图

**UpdateConstruction(); 更新作图()**。

重算全部的对象(随机数重新生成)同于“F9”键或“Ctrl+R”。

注: 若要刷新视图(例如去除绘图区视图中的跟踪轨迹)可以用“放大(1)”来代替, 这等同于“(Ctrl)+(F)”(刷新)。如果运用了两个绘图区视图, 还需要首先使用激活视图(1)或激活视图(2)。

**UpdateConstruction(<Number of times>); 更新作图(<更新次数>)**。

执行指令更新作图几次。

案例: “更新作图(2)”刷新作图两次(如将多次投掷骰子的结果记录到表格)。若要多次执行这个指令, 可以运用“序列(更新作图(), k, 1, 10)”。

## 2.10.10 Turtle. 海龟

**Turtle(); 海龟()**。

构造一只海龟图。初始的海龟在原点位置，且在绘图区的左下角出现一个操作海龟的按钮。

**注：**在 GeoGebra 中，会使用小卡通图做为绘图的主题对象，这个指令就是构造一个海龟示意图。而一些 GeoGebra 指令会指示海龟的移动动作。



**注：**参见“前进”、“后退”、“左转”、“右转”、“抬笔”和“落笔”指令。

### 2.10.11 TurtleBack. 后退

**TurtleBack**( $\langle$ Turtle $\rangle$ ,  $\langle$ Distance $\rangle$ ); **后退**( $\langle$ 海龟 $\rangle$ ,  $\langle$ 路程 $\rangle$ )。

海龟按照路程后退。

上右图中，输入指令“后退(海龟 1, 3)”后，点击左下角按钮，海龟就后退（x 轴负方向）3 个单位，缓慢移动到了 (-3, 0) 的位置。

**注：**参见“前进”、“海龟”、“左转”和“右转”指令。

### 2.10.12 Slider. 滑动条

**Slider**( $\langle$ Min $\rangle$ ,  $\langle$ Max $\rangle$ ,  $\langle$ Increment $\rangle$ ,  $\langle$ Speed $\rangle$ ,  $\langle$ Width $\rangle$ ,  $\langle$ Is Angle $\rangle$ ,  $\langle$ Horizontal $\rangle$ ,  $\langle$ Animating $\rangle$ ,  $\langle$ Boolean Random $\rangle$ ); **参数**( $\langle$ 最小值 $\rangle$ ,  $\langle$ 最大值 $\rangle$ ,  $\langle$ 增量 $\rangle$ ,  $\langle$ 速度 $\rangle$ ,  $\langle$ 宽度 (px) $\rangle$ ,  $\langle$ 角度?true|false $\rangle$ ,  $\langle$ 水平?true|false $\rangle$ ,  $\langle$ 启动动画? true|false $\rangle$ ,  $\langle$ 随机?true|false $\rangle$ )

创建一个滑动条（变量）。指令中的参数可进行下列设置：

**最小值，最大值：**设置滑动条的取值范围—这些参数是强制性的。

**增量：**设置滑动条值的增量—默认为：0.1。

**速度：**设置滑动条在动画时的速度—默认为：1。

**宽度：**设置滑动条的显示像素宽度—默认为：100。

**角度：**设置滑动条是否为一个角度值。这项参数可以是 true 或 false—默认为：false。

**水平：**设置滑动条在显示时是否为一条横向摆放（true）、还是一条纵向摆放（false）的线段—默认为：true。

**动画：**设置滑动条自动播放动画—默认为：false。

**随机：**设置滑动条在（最小值，最大值）区间上是否设为连续的数值（false），还是在此区间上为随机数（true）—默认为：false。

**注：**参见<sup>a=2</sup>“滑动条”工具。

### 2.10.13 ParseToFunction. 解析为函数

**ParseToFunction**( $\langle$ Function $\rangle$ ,  $\langle$ String $\rangle$ ); **解析为函数**( $\langle$ 函数 $\rangle$ ,  $\langle$ 字符串 $\rangle$ )。

在使用本指令前必须定义函数 f 为自由对象，将字符串解析为函数 f。

**案例：**定义“ $f(x)=3x^2+2$ ”和“ $text1=f(x)=3x+1$ ”。“解析为函数(f, text1)”返回“ $f(x)=3x+1$ ”且绘制出函数图象。相当于将文本型的函数表达式，转换为可进行函数运算的表达式。

**注：**用到的文字必须是全函数文字（x 的多项式或  $f(x)=x$  的多项式），否则结果为“未定义”；参见“解析到数”指令。



## 2.10.14 ParseToNumber. 解析为数

**ParseToNumber(<Number>, <String>); 解析为数(<数值>, <字符串>)**。

在使用本指令前定义数值 a 为自由对象，解析字符串且保存为 a。

**案例：**定义 a=3 和 text1="6"。“解析为数(a, text1)”返回 a=6。相当于将文本型数字转为数值型数字。

**注：**用到的文字必须是全数值文字，否则结果为“未定义”；参见“解析为函数”指令。

## 2.10.15 StartRecord. 开始记录

**StartRecord( ); 开始记录( )**。

如果当前是暂停，重启所有到电子表格的记录动作，且为每个对象保存一个值。

**StartRecord(<Boolean>); 开始记录(<true|false>)**。

当布尔值是“false”，暂停记录到表格的动作，否则重启之（且为每个对象保存为一个值）

## 2.10.16 TurtleDown. 落笔

**TurtleDown(<Turtle>); 落笔(<海龟>)**。

命令给定的海龟可留下踪迹。

**注：**本命令使得海龟移动时能留下痕迹；参见“抬笔”指令。



## 2.10.17 Pan. 平移视图

**Pan(<x>, <y>); 平移视图(<x>, <y>)**。

平移视图，x 为向左平移的像素量，y 为向上平移的像素量。

**Pan(<x>, <y>, <z>); 平移视图(<x>, <y>, <z>)**。

如果是 3D 视图区在 (x, y, z) 方向移动当前视图，否则在 2D 视图区在 (x, y) 方向移动视图。

**注：**如果同时呈现多个绘图区，会用于激活的区域；参见“放大”、“缩小”和“激活绘图区”指令。

## 2.10.18 StartAnimation. 启动动画

**StartAnimation( ); 启动动画( )**。

重启被暂停的动画。

**StartAnimation(<Boolean>); 启动动画(<true|false>)**。

如果布尔值是 false，暂停所有动画，否则重启动画。

**StartAnimation(<Point or Slider>, <Point or Slider>, ... ); 启动动画(<滑动条|点>, <滑动条|点>, ...)**。

启动给定的点和滑动条动画，点必须是在路径上。

**StartAnimation(<Point or Slider>, <Point or Slider>, ..., <Boolean>); 启动动画(<滑动条|点>, <滑动条|点>, ..., <true|false>)**。

启动(当布尔值=true)或长停(当布尔值=false)给定点和滑动条的动画，点必须是在路径上。

注：参见另作使用手册“动画”相关章节。

## 2.10.19 StartLogging. 启动日志

**StartLogging**("<Sensor>", <Variable>, <Limit (optional)>, "<Sensor>", <Variable>, <Limit (optional)>, ...); 启动日志("<感应器 1>", <变量 1>, "<感应器 2>", <变量 1>, ...)

记录诸如来自电话等感应器的数据。由数字、列表或是数据函数形式的变量确定每种感应器的记录值。限定参数确定记录数据的长度。当前支持的传感器变量的值有：

"Ax", "Ay", "Az": 加速计 (m/s<sup>2</sup>);

"Ox", "Oy", "Oz": 定向(度数);

"Mx", "My", "Mz": 磁场 (μT);

"proximity": 邻近度(cm);

"light": 照度(lx);

"loudness": 响度;

"time": 时间戳;

"datacount": 计数;

"appID": 不是感应器，后跟应包含由移动应用程序生成的代码的文本。

注：参见“停止日志”指令。“感应器”功能目前只支持在线与平板版。

## 2.10.20 TurtleForward. 前进

**TurtleForward**(<Turtle>, <Distance>); 前进(<海龟>, <路程>)。

海龟按照路程前进。

输入指令“前进(海龟 1, 3)”后，点击左下角播放三角按钮，海龟就前进（x 轴正方向）3 个单位，缓慢移动到新的位置。

注：路程可以是负值。参见“海龟”、“后退”、“左转”、“右转”指令。

## 2.10.21 Delete. 删除

**Delete**(<Object>); 删除(<对象>)。

删除对象以及所有它的从属对象。


案例：“删除(a)”清除掉 a。

**CAS Syntax (运算区语法)**

**Delete**(<Object>); 删除(<对象>)。

在 GeoGebra 中删除对象以及从属对象，并去除运算区中所有指派给这个对象的数值。

案例：“删除(a)”清除掉 a。

注：参见  “删除”工具。

## 2.10.22 SetBackgroundColor. 设置背景颜色

这个指令在不同的英语变型中拼写不同：SetBackgroundColor (US)、SetBackgroundColour (UK+Aus)。

**SetBackgroundColor**(<Object>, <Red>, <Green>, <Blue>); 设置背景颜色(<对象>, <红色 0-1>, <绿色 0-1>, <蓝色 0-1>)。

更改指定对象的背景颜色，用于文本对象和表格区的对象。红、绿和蓝能表示大量的相关颜色

组成，其值最小 0，最大 1。超过这个区间的数值 t，使用函数“ $2|t/2 - \text{round}(t/2)|$ ”映射。

案例：“设置背景颜色(A2, 255, 0, 0)”，工作表中“A1”单元格更改背景色。



### 常用的基本颜色

名称	红色值 Red	绿色值 Green	蓝色值 Blue
黑色	0	0	0
蓝色	0	0	255
绿色	0	255	0
青色	0	255	255
红色	255	0	0
洋红色	255	0	255
黄色	255	255	0
白色	255	255	255

**SetBackgroundColor(<Object>, <"Color">); 设置背景颜色(<对象>, "<颜色">")。**

改变给定对象的背景颜色。这个用于文本和表格区对象。“颜色”键入文本。这个指令接受超过 100 个英文颜色名称（见参考：颜色）。部分颜色名称见下图，名称显示的颜色就是字体或者文字背景的颜色。

Black	Indigo	Light Gray
Dark Gray	Purple	Pink
Gray	Brown	Violet
Dark Blue	Orange	Yellow
Blue	Gold	Light Yellow
Dark Green	Lime	Light Orange
Green	Cyan	Light Violet
Maroon	Turquoise	Light Purple
Crimson	Light Blue	Light Green
Red	Aqua	White
Magenta	Silver	

注：如果在脚本中使用颜色指令，必须使用英文名称。

### 2.10.23 SetDecoration. 设置标记

**SetDecoration(<Object>, <Number>); 设置标记(<线段|角>, <标记数字代码 0—6|7> )**

更改给定对象的标记（请参阅对象属性中的“样式”选项卡）。对象必须是角或线段，第二个

参数是标注样式的数字代码，对于可能的值，请参见下表。

值	线段标记	角标记
1	一划	二弧
2	二划	三弧
3	三划	一划
4	一箭头	二划
5	二箭头	三划
6	三箭头	顺时针箭头
7	(无)	逆时针箭头

## 2.10.24 SetLabelMode. 设置标签模式

**SetLabelMode(<Object>, <Number>);** 设置标签模式(<对象>, <0\_名称|1\_名称+数值|2\_数值|3\_标题>)。

从下面表单中选给定对象的标签模式。与下列表中不同整数按照 0 对待。

数值	模式
0	名称
1	名称与数值
2	数值
3	标题

**注：**此命令也可以用窗口菜单实现：点选对象→“属性”→“常规”→“显示标签”右的下拉菜单中选择想要的模式。

## 2.10.25 SetCaption. 设置标题

**SetCaption(<Object>, <Text>);** 设置标题(<对象>, “<标题文本>”)。

更改指定对象的标题，文本必须用一对英文的双引号“”括住。

**案例：**“设置标题(A, “圆心”)”。



**注：**这个指令是为对象的标题赋予内容，标题在“标签模式”选择“标题”时才会显示。

## 2.10.26 SetPointSize. 设置点径

**SetPointSize(<Point>, <Number>);** 设置点径(<点|多边形|多面体|展开图>, <数值 0—9>)。

改变点的大小。

**SetPointSize(<Polygon>, <Number>);** 设置点径(<点|多边形|多面体|展开图>, <数值 0—9>)。

改变多边形顶点的大小。

**SetPointSize(<Polyhedron>, <Number>);** 设置点径(<点|多边形|多面体|展开图>, <数值 0—9>)。

改变多面体顶点的大小。

**SetPointSize(<Net>, <Number>);** 设置点径(<点|多边形|多面体|展开图>, <数值 0—9>)。

改变展开图顶点的大小。

**注：**自 GeoGebra 5 这个指令可以用于设置多边形、多面体和展开图的顶点大小；“0”隐藏对象。可以将点径设为 0 或是大于 9 的数值，但在属性对话框中只能在范围 (0, 9) 之内进行更改。

**注：**也可以针对点（未名点）列表操作。如“点列={{(1, 2), (3, 4)}}”，指令“设置点径(点列, 5)”改变点列中所有点的大小。

## 2. 10. 27 SetPointSize. 设置点型

**SetPointSize**(**<Point>**, **<Number>**); 设置点型(**<点>**, **<数值 0\_圆点|1\_叉形|2\_空心点|3\_十字|4\_菱形|5\_空心菱形|6\_上三角|7\_下三角|8\_左三角|9\_右三角>**)。

依据下表更改指定点的点型（在范围 (0, 9) 之外的数值按 0 处理）。

数值	型号名称	符号
0	圆点	●
1	叉形	×
2	空心点	○
3	十字	+
4	菱形	◆
5	空心菱形	◇
6	上三角	▲
7	下三角	▼
8	左三角	◀
9	右三角	▶

## 2. 10. 28 SetDynamicColor. 设置动态颜色

这个指令在不同的英语变型中拼写不同：**SetDynamicColor** (US)、**SetDynamicColour**。

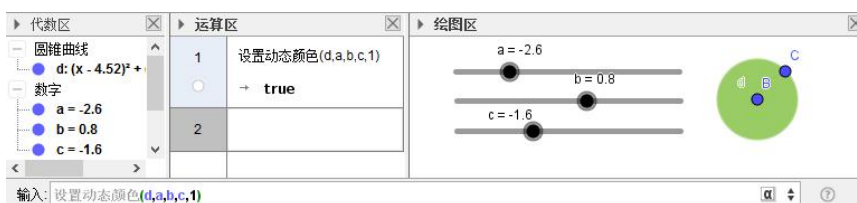
**SetDynamicColor**(**<Object>**, **<Red>**, **<Green>**, **<Blue>**); 设置动态颜色(**<对象>**, **<红>**, **<绿>**, **<蓝>**)。

设置对象的动态颜色。

**SetDynamicColor**(**<Object>**, **<Red>**, **<Green>**, **<Blue>**, **<Opacity>**); 设置动态颜色(**<对象>**, **<红>**, **<绿>**, **<蓝>**, **<虚实 0—100>**)。

设置对象的动态颜色和透明度。

**案例：**“设置动态颜色(d, a, b, c, 1)”返回由参数 a、b、c 控制颜色，不透明的圆。



**注：**所有数值有一个数值范围 0（含/透明的）到 1（不含/不透明的）。

## 2.10.29 SetFixed. 设置对象锁定

`SetFixed(<Object>, <true|false>)`; 设置设置对象锁定(<对象>, <true|false>)

使对象设置设置对象锁定(选 true)或不设置设置对象锁定(选 false)于绘图区特定位置。

注: 固定的对象可以右键拖移。如果想删除固定的对象, 需要先右键在菜单中解除其固定状态。

## 2.10.30 SetPerspective. 设置格局

`SetPerspective(<Text>)`; 设置格局(“<文本 A\_代数区|B\_概率统计|C\_运算区|D\_绘图区 II |G\_绘图区|L\_作图过程|P\_属性|R\_数据分析|S\_表格区|T\_3D 绘图区>”)。

更改视图的布局和视觉效果, 文本参数描述了所需的布局。文本应是一个有效的表达式-视图是由变量(字母)表示: 水平排列的字母代表区域水平并列放置, 用分号“/”划分的表示它们垂直排列。

字母	视区
A	代数区
B	概率统计
C	运算区
D	绘图区 2
G	绘图区
L	作图过程
P	属性
R	数据分析 (离线版)
S	表格区
T	3D 绘图区

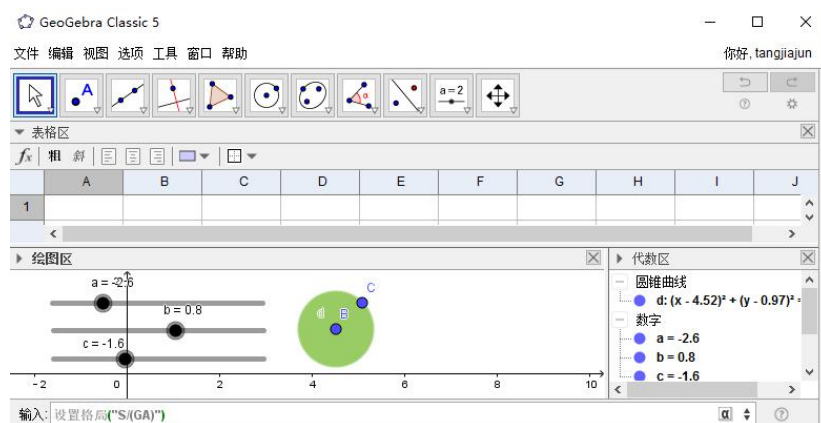
案例:

“设置格局(“G”)”: 设置只显示绘图区。

“设置格局(“AGS”)”: 设置显示代数区、绘图区和表格区, 且水平排列。

“设置格局(“S/G”)”: 设置显示表格区和绘图区, 表格区在上方, 绘图区在下方排列。

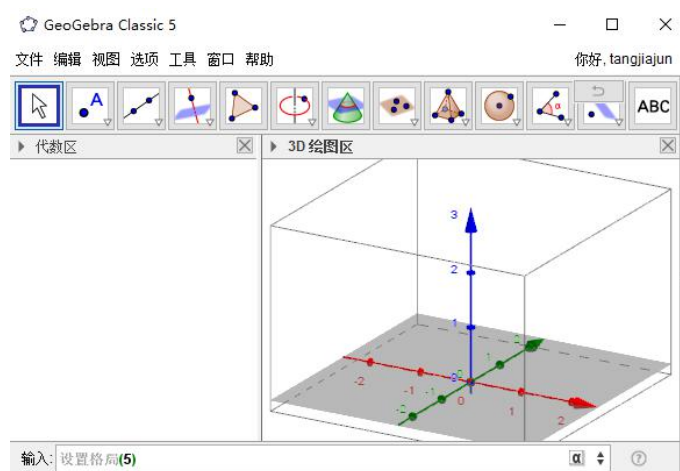
“设置格局(“S/(GA)”)”: 如上条, 底部左边是绘图区, 右边是代数区。



作为替代，文本中可以使用如下预先定义好视区的数字：

文本	视区
"1"	代数区和绘图区
"2"	几何区(绘图区)
"3"	表格区
"4"	运算区
"5"	3D 绘图区
"6"	概率统计

这些大致分别对应于“AG”，“G”，“SG”，“CG”，“AT”和“B”，但也可能会影响到“指令栏”和“工具栏”的显示。



注：参见“活动视图”指令。

### 2.10.31 SetTrace. 设置跟踪

`SetTrace(<Object>, <true|false>)`；设置跟踪(<对象>, <true|false>)。

切换特定对象的跟踪“开/关”。

案例：创建点 A，然后键入“设置跟踪(A, true)”。选择“移动”工具且拖动点，显示它的踪迹。



注：使用“放大(1)”可以清除所有踪迹，但不是唯一方法。

### 2.10.32 SetTooltipMode. 设置工具提示模式

`SetTooltipMode(<Object>, <Number>)`；设置工具提示模式(<对象>, <数值 0\_自动|1\_开启|2\_关闭|3\_标题|4\_下一单元格>)。

从下面表单中选给定对象的工具提示模式（在范围(0, 4)以外的值按照 0 对待）：

数值	模式
----	----

0	自动
1	开
2	关
3	标题
4	下一单元格

当鼠标悬停在对象上方时，出现的提示信息可以如上表设定。

**注：**在范围(0, 4)之外的数值被处理为0；当对象是单元格创建的对象时，“4”表示显示的是其右边单元格内的信息。

**释义：**对于已有对象，当鼠标接近时，会出现提示信息，以方便进一步操作。对象“标题”中的内容将会显示。“下一单元格”是显示指定位置右侧单元格内部的内容。

### 2.10.33 SetActiveView. 设置活动视图

**SetActiveView(<View>); 设置活动视图 (<视图编号 1/G\_绘图区|2/D\_绘图区II|-1/T\_3D绘图区|A\_代数区|S\_表格区|C\_运算区>)**

激活给定绘图区。

**注：**参见“放大”、“缩小”、“平移视图”和“格局”指令。“1/G\_绘图区”表示“1”或者“G”都可以激活绘图区。

**案例：**“设置活动视图(“T”)”和“设置活动视图(-1)”都是激活3D绘图区。

**SetActiveView(<Plane>); 设置活动视图 (<平面>)**。

激活指定平面的视图。

**注：**在3D视图中，右键任意一个平面a，都可以在右键菜单中勾选“创建a的平面图”。此后，就可以在格局中出现这个平面a的附加视图，本命令就是用于激活这个附加视图。

### 2.10.34 SetVisibleInView. 设置可见性

**SetVisibleInView(<Object>, <View Number 1|2>, <Boolean>); 设置可见性 (<对象>, <视图编号 1\_绘图区|2\_绘图区II>, <true|false>)**。

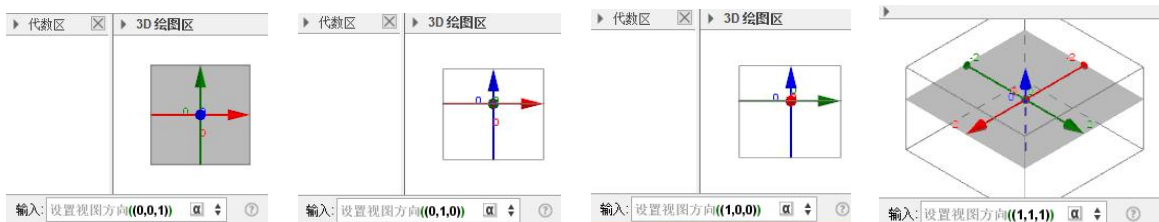
在指定的绘图区(1或2)显示(true)或隐藏(false)对象。

### 2.10.35 SetViewDirection. 设置视图方向

**SetViewDirection(<Direction>); 设置视图方向 (<方向, eg. (0, 0, 1)>)**。

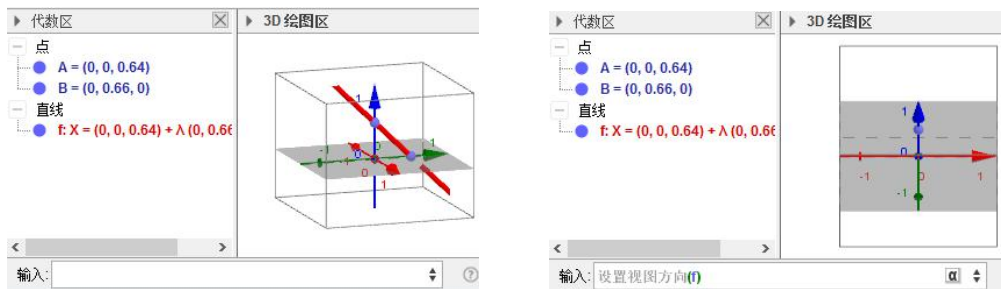
设置3D视图为指定方向

**案例：**“设置视图方向((0, 0, 1))”：视角为从z轴正方向向负方向正视。



**注：**视图方向也可以设置指向一条直线、一个线段和一个平面，等等；参见“视图方向”工具。





如上图左有直线  $f$ ；上图右就是直线  $f$  指定的视向结果（视线沿着  $f$  构造方向）。

### 2.10.36 SetFilling. 设置填充

**SetFilling(<Object>, <Number>); 设置填充(<对象>, <数值 0—1>)。**

更改给定对象的模糊度（不透明程度）。数值必须是区间  $(0, 1)$ ，“0”意味着透明，而“1”意味着 100%不透明，忽略其他数值。

### 2.10.37 SetLayer. 设置图层

**SetLayer(<Object>, <Layer>); 设置图层(<对象>, <层次>)。**

设置指定对象的图层，图层的数值必须是在数集  $\{0, 1, \dots, 9\}$  中的整数。

**注：**0 表示最下面的图层（远离使用者），9 表示最上面的图层。

### 2.10.38 SetLevelOfDetail. 设置细节级别

**SetLevelOfDetail(<Surface>, <Level of Detail>); 设置细节级别(<曲面>, <细节级别 0|1>)**  
设置表面光滑程度。

**编者注：**官网没有及时的帮助，且功能没有验证。

### 2.10.39 SetConditionToShowObject. 设置显示条件

**SetConditionToShowObject(<Object>, <Condition>); 设置显示条件(<对象>, <条件>)。**

设置对象在指定的条件下显示。

**案例：**设有一个 A 点与一个数值变量  $t$ ，如果作以下的设定：“设置显示条件  $(A, t > 1)$ ”  
这时只有在“ $t > 1$ ”的情况下，A 才会显示。

**注：**对象名与条件式都不需要用“双引号”。

显示条件可以是真假值、数值对象的大小关系（ $\neq$ 、 $\leq$ 、 $\geq$ 、 $>$ 、 $<$ ）、对象的属于关系（ $\in$ ）等。

如：“设置显示条件  $(A, a)$ ”：当  $a$  有意义时显示 A；“设置显示条件  $(A, x(B) \neq x(C))$ ”：当 B 和 C 的横坐标相同时显示 A；“设置显示条件  $(A, 2 < s < 9)$ ”：当  $s$  在区间内时显示 A；“设置显示条件  $(A, s \in \{0, 1, 2, 3, 4, 8, 9\})$ ”：当  $s$  是列表中任意一个元素时显示 A。

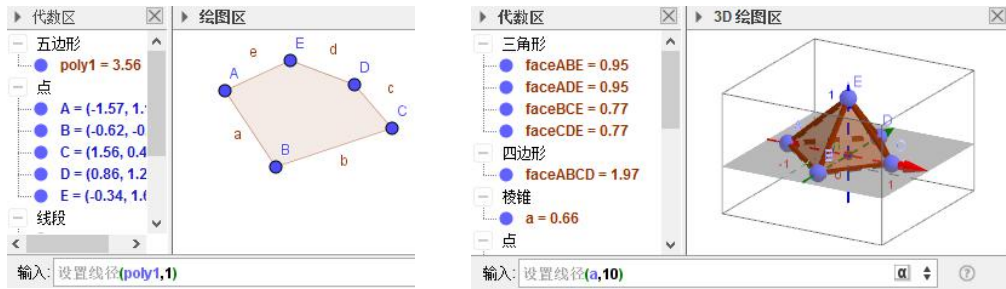
### 2.10.40 SetLineThickness. 设置线径

**SetLineThickness(<Object>, <Number>); 设置线径(<直线|射线|线段>, <数值 1-13>)**

设  $N$  为数值。线径指令将指定线的线径设置为  $N/2$  像素。

**注：**自 GeoGebra 5 这个指令可用于设置多边形和多面体的线径（边线和棱）。

只支持范围 0 到 13 内的数值。



## 2.10.41 SetLineStyle. 设置线型

`SetLineStyle(<Line>, <Number>)`; 设置线型 (<直线|射线|线段>, <数值 0\_实线|1\_长划线|2\_短划线|3\_点|4\_划线-点>)

依据下表更改指定对象的线型（在范围 (0, 4) 之外的数值看做 0）。

数值	型号
0	实线
1	长虚线
2	短虚线
3	点虚线
4	点划线

## 2.10.42 SetColor. 设置颜色

这个指令在不同的英语变型中拼写不同：`SetColor` (US)、`SetColour` (UK+Aus)。

`SetColor(<Object>, <Red>, <Green>, <Blue>)`; 设置颜色 (<对象>, <红>, <绿>, <蓝>)。

更改指定对象的颜色。红、绿和蓝能表示大量的相关颜色组成，其值最小 0，最大 1。超过这个区间的数值  $t$ ，使用函数  $2|t/2 - \text{round}(t/2)|$  映射到 0 到 1 的区间。

`SetColor(<Object>, <"Color">)`; 设置颜色 (<对象>, <"颜色">)。

改变给定对象的颜色。“颜色”键入文本。这个指令接受超过 100 个英语颜色名称（见参考：颜色）。其中一些可以使用本地语言，列示如下。

Black	Indigo	Light Gray
Dark Gray	Purple	Pink
Gray	Brown	Violet
Dark Blue	Orange	Yellow
Blue	Gold	Light Yellow
Dark Green	Lime	Light Orange
Green	Cyan	Light Violet
Maroon	Turquoise	Light Purple
Crimson	Light Blue	Light Green
Red	Aqua	White

Magenta	Silver	
---------	--------	--

注：在 GeoGebra 脚本指令中使用，必须使用颜色的英文名称。

### 2.10.43 SetSeed. 设置种子

**SetSeed(<Integer>); 设置种子(<整数>)。**

设置种子使得后续随机数由种子决定。

**案例：**“设置种子(33)”：无可见现象。

**注：**参见“区间随机数”指令、“随机元素”指令、“均匀分布”指令、“随机二项分布数”指令、“正态分布随机数”指令、“泊松分布随机数”指令。

设置种子指令必须和随机数指令同时使用才有效，单独在输入框输入无效。可写在对象脚本，或嵌套在“执行”指令中。

1. 脚本：

```
SetSeed(2)
```

```
a=RandomBetween(0,100)
```

2. 执行指令

“执行(“SetSeed(5)”, “a=RandomBetween(0,100)”)” (请查阅“执行”指令)。

**注：**本指令独立运行时，可以输入中文，比如“设置种子(2)”，但嵌套在“执行”指令中的语法，必须是英文的，这个指令也不例外。

### 2.10.44 SetSpinSpeed. 设置转速

**SetSpinSpeed(<Number>); 设置转速(<速度值>)。**

设置 3D 视图绕当前垂直屏幕的显示轴的转速。键入值的符号和值定义以下旋转：

如果数字为正数，则 3D 视图逆时针旋转。

如果数字为负数，那么 3D 视图顺时针旋转。

如果数字为 0，那么 3D 视图不旋转。

### 2.10.45 SetCoords. 设置坐标

**SetCoords(<Object>,<x>,<y>); 设置坐标(<对象>,<横坐标>,<纵坐标>)。**

改变笛卡尔坐标系中已有对象坐标。这个指令使用了坐标的值，不是对象的定义，所以对象仍然是自由对象。

**注：**这个指令可作用于滑动条参数、布尔值、按钮、复选框、输入框、图像等，不包括几何对象和函数。如果选择了“屏幕上的绝对位置”，横坐标 x、纵坐标 y 则为屏幕像素单位。

### 2.10.46 SetAxesRatio. 设置坐标轴比例

**SetAxesRatio(<Number>,<Number>); 设置坐标轴比例(<数值 1>,<数值 2>)。**

更改激活绘图区中的坐标轴比例，使得 x-轴上 X 的单位对应为“数值 1”的像素值，y-轴上 Y 的单位对应“数值 2”，点(0,0)保持其坐标。(x-轴不动，y-轴按比例缩放)

**SetAxesRatio(<Number>,<Number>,<Number>); 设置坐标轴比例(<数值 1>,<数值 2>,<数值 3>)。**

这个操作与上一条相似，用于 3D 绘图区。

## 2.10.47 InputBox (Textfield) . 输入框

**InputBox()**; 输入框( )。

创建一个新的输入框。

**InputBox(<Linked Object>)**; 输入框(<链接对象>)。

创建一个输入框并将其与一个链接对象相关联。



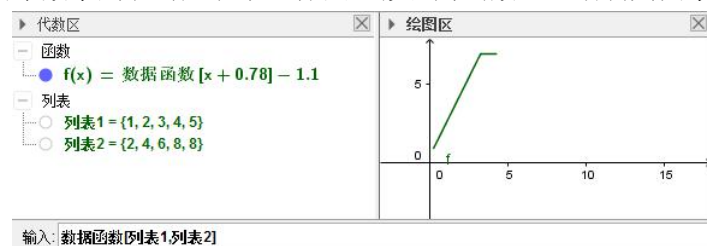
输入框内的内容与点的坐标链接，移动点，框内数据随变；修改框内数据，点会移动到新的坐标位置。

注：参见<sup>a=</sup>“输入框”工具。

## 2.10.48 DataFunction. 数据函数

**DataFunction(<List of Numbers>, <List of Numbers>)**; 数据函数(<数字列表 1>, <数字列表 2>)。

将同等元素的两个数字列表对应表示坐标点，拟合为函数，且绘制其图象。



案例：设有两个数字列表“列表 1={1,2,3,4,5}”和“列表 2={2,4,6,8,8}”，指令“数据函数(列表 1,列表 2)”得出函数解析式并绘制出函数图象。

## 2.10.49 ZoomOut. 缩小

**ZoomOut(<Scale Factor>)**; 缩小(<缩放因子>)。

基于当前比例将绘图区视图以指定缩放因子进行缩小，屏幕中心为缩放中心点。

案例：“缩小(2)”缩小视图。

**ZoomOut(<Scale Factor>, <Center Point>)**; 缩小(<缩放因子>, <中心点>)。

基于当前比例将绘图区视图以指定缩放因子进行缩小，第二个参数指定了缩放中心点。

案例：“缩小(2, (0, 0))”：以原点为中心，缩小视图区。

注：“缩小(t)”和“缩小(t, A)”分别等效于“放大(1/t)”和“放大(1/t, A)”；如果存在多个绘图区视图，则用于激活使用的那个；参见“放大”、“活动视图”、“平移视图”指令。

## 2.10.50 TurtleUp. 抬笔

**TurtleUp(<Turtle>)**; 抬笔(<海龟>)

指示给定的海龟停止追踪踪迹。

注：参见“落笔”指令。

## 2.10.51 StopLogging. 停止日志

**StopLogging()**；停止日志()。

停止记录诸如电话等感应器类数据。

注：参见“启动日志”指令。

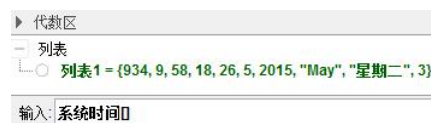
## 2.10.52 GetTime. 系统时间

**GetTime()**；系统时间()。

创建一个以下序的当前系统时间的列表：

毫秒、秒、分、时 (0-23)、日、月 (1-12)、年、月份 (文本格式)、星期 (文本格式)、星期日数 (1=周日、2=周一, 类推)

案例：“系统时间()”返回一个如 {647, 59, 39, 23, 28, 2, 2011, "February", "Monday", 2} 的列表。

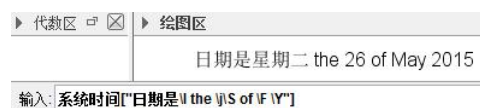


**GetTime("<Format>")**；系统时间("<格式>")。

当格式为文本，使用反斜杠 (\) 前缀下列字符：

d, D, j, l, N, S, w, z, W, F, m, M, n, t, L, Y, y, a, A, g, G, h, H, i, s, U-这些字符的解释在 <http://php.net/manual/en/function.date.php>

案例：“系统时间("日期是 \l the \j\S of \F \Y")”会给出“日期是 Thursday the 5th of July 2012”。



年：Y (2014) , y (14)

月：F 或 M (Jul) , m 或 n (7)

日：d 或 J (11) ; z (191:第几天)

星期：l (小写 L) (星期五) , N 或 w (5)

时：G 或 H (21) g 或 h (9)

分：i (20)

秒：s (57)

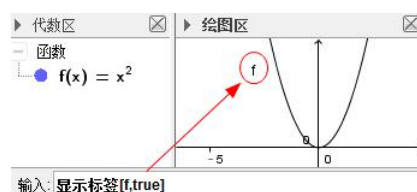
上下午：a (pm) , A (PM)

## 2.10.53 ShowLabel. 显示标签

**ShowLabel(<Object>, <Boolean>)**；显示标签(<对象>, <true|false>)。

在绘图区显示或隐藏给定对象的标签。

案例：假设“ $f(x)=x^2$ ”。 “显示标签(f, true)”显示函数的标签。

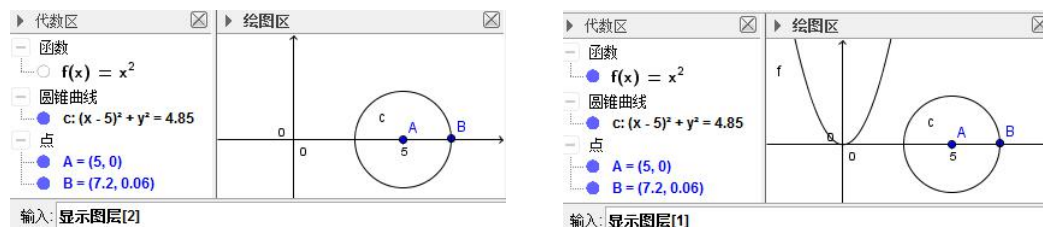


## 2.10.54 ShowLayer. 显示图层

**ShowLayer(<Number>); 显示图层(<数值>)。**

使给定图层的所有对象处于显示状态。不覆盖“设置显示条件”的设置。

**案例：**“显示图层(2)”使所有处于第二图层的对象显示。



函数  $f$  在第一图层，圆在第二图层，在代数区都设置为隐藏。分别执行两条指令后显示如图。

## 2.10.55 ShowGrid. 显示网格

**ShowGrid( ); 显示网格( )。**

在激活绘图区显示网格。

**ShowGrid(<Boolean>); 显示网格(<true|false>)。**

在激活绘图区显示/隐藏网格。

**案例：**“显示网格(true)”在激活绘图区显示网格；“显示网格(false)”在激活绘图区隐藏网格。

**ShowGrid(<View>, <Boolean>); 显示网格(<视图 1|2>, <true|false>)。**

在由数字 1 或 2 (在 GeoGebra5 的 3D 绘图区中, 或 3) 指定的绘图区中显示/隐藏网格。

**案例：**“显示网格(1, true)”在绘图区显示网格；“显示网格(2, false)”在绘图区 II 中隐藏网格。

**注：**参见“显示坐标轴”指令。

## 2.10.56 ShowAxes. 显示坐标轴

**ShowAxes( ); 显示坐标轴( )**

在激活绘图区显示坐标轴。

**ShowAxes(<Boolean>); 显示坐标轴(<true|false>)。**

在激活绘图区显示/隐藏坐标轴。

**案例：**“显示坐标轴(true)”在激活的绘图区显示坐标轴；“显示坐标轴(false)”在激活的绘图区隐藏坐标轴。

**ShowAxes(<View>, <Boolean>); 显示坐标轴(<视图>, <true|false>)。**

在由数字 1 或 2 (在 GeoGebra5 的 3D 绘图区中, 或 3) 指定的绘图区中显示/隐藏坐标轴。

**案例：**“显示坐标轴(1, true)”在绘图区中显示坐标轴；“显示坐标轴(2, false)”在绘图区 II 中隐藏坐标轴。

**注：**参见“显示网格”指令。

## 2.10.57 SelectObjects. 选择

**SelectObjects( ); 选择( )。**

取消所有已选定对象。

SelectObjects(<Object>, <Object>, ... ); 选择(<对象 1>, <对象 2>, ... )。

取消当前所有选定对象并选定参数列出的对象。所有参数对象必须是有标签的对象。

**案例：**设有点 A、B 和 C。“选择(A, B, C)”选定点 A、B 和 C；指令“选择(中点(A, B))”无效（因为中点没有提前被定义，也就是没有标签）。

**注：**这个指令会马上取消对象在进程中的任何拖拽操作（在脚本中十分有用）。

## 2.10.58 HideLayer. 隐藏图层

HideLayer(<Number>); 隐藏设置图层(<图层编号(0-9)>)。

使所有在指定图层上的对象不可视。不会覆写“设置显示条件”的设置。

**注：**参见“显示图层”指令。

## 2.10.59 TurtleRight. 右转

TurtleRight(<Turtle>, <Angle>); 右转(<海龟>, <角度>)。

海龟按照角度右转。

**案例：**当指令“右转(海龟, 1)”执行到出现暂停按钮时，海龟向右转 1 弧度。否则按启动按钮执行这个旋转。

**注：**如果输入“右转(海龟, 1)”，海龟会向右转向  $1^\circ$ 。

**案例：**结合“重复”指令，绘制卡通八边形，体会海龟转向的应用。

会用到的“重复”指令为：Repeat(<Number>, <Scripting Command>, <Scripting Command>, ...); 重复(<数值>, <脚本指令 1>, <脚本指令 2>, ...)

当 n 是给定的数值，重复脚本指令 n 次。

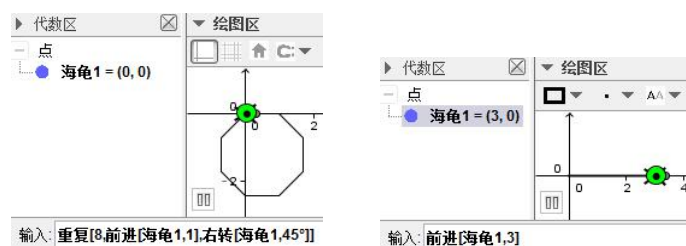
**步骤：**

1、在指令栏输入“海龟()”；

2、点击显示在左下角的▶“播放”按钮；

3、在指令栏输入“重复(8, 前进(海龟 1, 1), 右转(海龟 1,  $45^\circ$ ))”。

可以看到海龟缓慢移动绘制出正八边形，在海龟运动过程中，会发现海龟的坐标在改变。如果在执行步骤 3 前加入“抬笔(海龟 1)”，则海龟走出正八边形，但因为停止踪迹，而不绘制正八边形。



参见：“海龟”、“前进”、“后退”和“左转”指令。

## 2.10.60 ReadText. 阅读文本

ReadText(<Text>); 阅读文本("<文本>")

指示屏幕阅读器立即阅读给定文本

**注：**此命令允许为视障用户提供信息，使他们的更易于访问软件。要听到阅读，需要安装屏幕阅读器，如 NVDA 或 VoiceOver。目前，它仅支持 GeoGebra 的在线版本。

## 2. 10. 61 RunClickScript. 运行单击脚本

**RunClickScript(<Object>); 运行单击脚本(<对象>)**。

运行与对象相关联的“单击时”脚本(如果有的话)。

**案例:** 设 A 和 B 是两个点。点 B 的“单击时”脚本是“赋值(B, (1, 1))”。设置点 A 的“单击时”脚本为“运行单击脚本(B)”，当点 A 被点击时，点 B 会移动到(1, 1)位置。

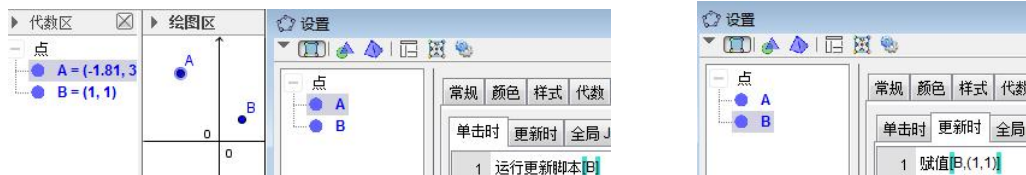


注: 参见“运行更新脚本”指令。

## 2. 10. 62 RunUpdateScript. 运行更新脚本

**RunUpdateScript(<Object>); 运行更新脚本(<对象>)**。

运行对象的“更新时”脚本(如果有的话)。



注: 参见“运行单击脚本”指令。

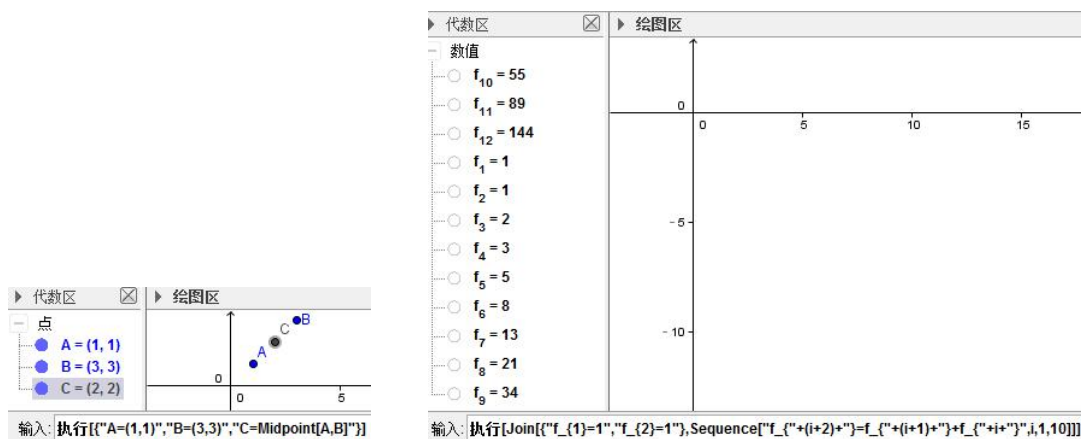
## 2. 10. 63 Execute. 执行

**Execute(<List of Texts>); 执行(<文本列表>)**。

执行以文本形式输入在列表中的指令。

**注:** 请注意使用这个指令时，文本中包含的指令名称必须使用英文，不管在 GeoGebra 语言选项中勾选了何种语言。

**案例:** “执行({“A=(1, 1)”, “B=(3, 3)”, “C=Midpoint(A, B)”})” 创建点 A、B 和它们的中点 C。  
“执行(join({“f\_{1}=1”, “f\_{2}=1”}, Sequence(“f\_{i+(i+2)+}=f\_{i+(i+1)+}+f\_{i+i}”, i, 1, 10)))” 创建斐波那契数列的前 10 个元素。



**Execute(<List of Texts>, <Parameter>, ... , <Parameter>); 执行(<文本列表>, <参数 1>, <**



参数 2), ... )。

用“参数 1”代换“%1”，“参数 2”代换“%2”，并对列表中的文本以此类推。最多可以指定 9 个参数。代换之后，所得脚本会被执行。

案例：“执行({“Midpoint(%1,%2)”},A,B)”创建线段 AB 的中点。



注：这个指令中的指令名称必须是英语(us)才能工作。

## 2.10.64 CenterView. 中心定位

**CenterView(<Center Point>); 中心定位(<视图中心设置坐标(x,y)|视图中心点>)**。

把绘图区的中心移动到指定点。

案例：“中心定位((0,0))”将原点移动到屏幕（视图区）的中心。

注：如果存在多个绘图区，则应用于被激活的那一个。

## 2.10.65 Repeat. 重复

**Repeat(<Number>,<Scripting Command>,<Scripting Command>,...); 重复(<数值>,<脚本指令 1>,<脚本指令 2>,...)**。

当 n 是给定的数值，重复脚本指令 n 次。

案例：依次执行以下指令，会动态绘制出正八边形。

“海龟()”；点击视区左下角的“▶”（播放）按钮；“重复(8,前进(海龟 1,1),右转(海龟 1,45°))”。



## 2.10.66 Rename. 重命名

**Rename(<Object>,<Name>); 重命名(<对象>,<名称>)**。

将指定对象的标签设置为指定名称。

案例：设“ $c:x^2+2y^2=2$ ”。“重命名(c,ell)”设置圆锥曲线的标签为“ell”。

## 2.10.67 TurtleLeft. 左转

**TurtleLeft(<Turtle>,<Angle>); 左转(<海龟>,<角度>)**。

海龟按照给定角度左转。

案例：当指令“左转(海龟,1)”执行到出现暂停按钮时，海龟向左转向 1 弧度。否则按启动按钮执行这个旋转。

注：如果输入“左转(海龟,1)”，海龟会向左转向  $1^\circ$ 。

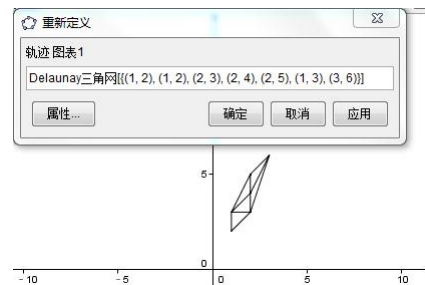
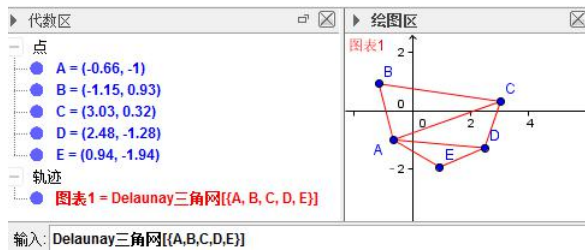
## 2.11 DiscreteMath. 离散数学

### 2.11.1 DelaunayTriangulation. Delaunay 三角网

**DelaunayTriangulation(<List of Points>); Delaunay 三角网(<点列>)**。

创建给定点列的三角网。返回的对象是轨迹，所以是辅助对象。

注：Delaunay 是人名。

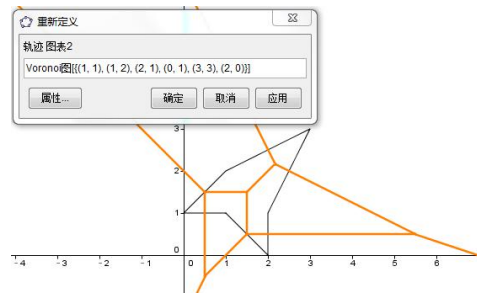
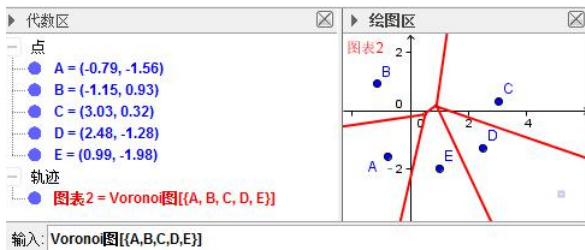


### 2.11.2 Voronoi. Voronoi 图

**Voronoi(<List of Points>); Voronoi 图(<点列>)**。

绘制一个给定点的 Voronoi 图。返回对象是轨迹，所以它是辅助对象。

注：Voronoi 是人名。由一组连接两邻点直线的垂直平分线组成的连续多边形组成。N 个在平面上有区别的点，按照最邻近原则划分平面；每个点与它的最近邻区域相关联。



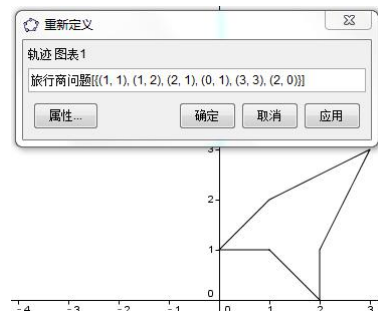
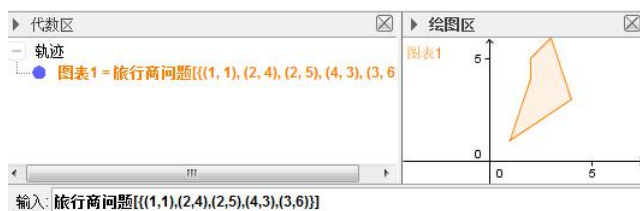
### 2.11.3 TravelingSalesman. 旅行商问题

这个指令在不同的英语变型中拼写不同：TravelingSalesman(US)、TravellingSalesman(UK+Aus)。

**TravelingSalesman(<List of Points>); 旅行商问题(<点列>)**。

返回通过所有给定点的最短的封闭路径，且每个点只经过一次。返回的对象是一个轨迹。

案例：“旅行商问题({(1,1),(2,4),(2,5),(4,3),(3,6)})” 如下图。

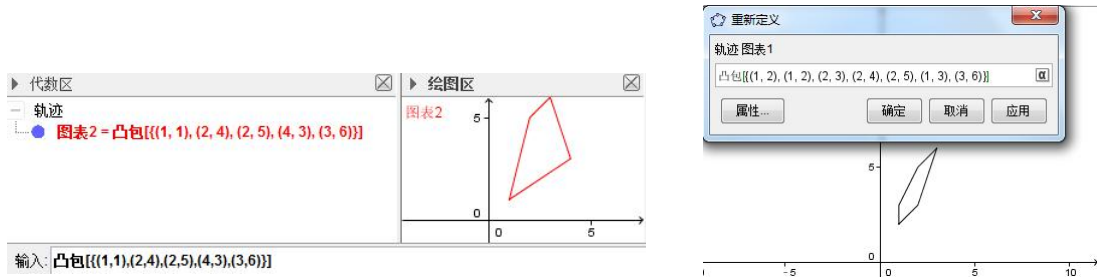


## 2.11.4 ConvexHull. 凸包

**ConvexHull(<List of Points>); 凸包(<点列>)。**

创建给定点集的凸包。返回的对象是轨迹，所以是辅助对象。

案例：“凸包({(1,1),(2,4),(2,5),(4,3),(3,6)})”如下图



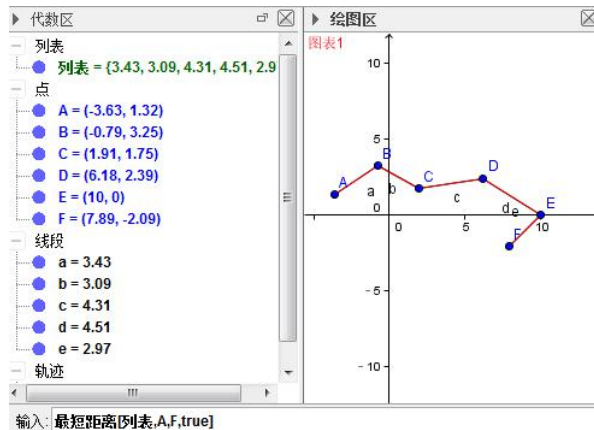
**凸包(<点集>)**

创建给定点集的凸包，返回的对象是一个轨迹。

## 2.11.5 ShortestDistance. 最短距离

**ShortestDistance(<List of Segments>, <Start Point>, <End Point>, <Boolean Weighted>); 最短距离(<线段列表>, <起始点>, <结束点>, <是否加权, true|false>)。**

找到给定起点和终点之间最短路径的系列线段。如果加权是 false，每条边的权重认为是 1（寻找边数最少的路径），否则是给定线段的长度（寻找几何最短路径）。



## 2.11.6 MinimumSpanningTree. 最小生成树

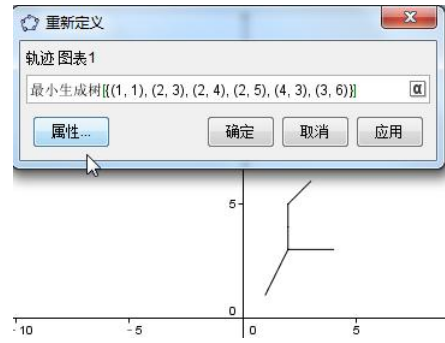
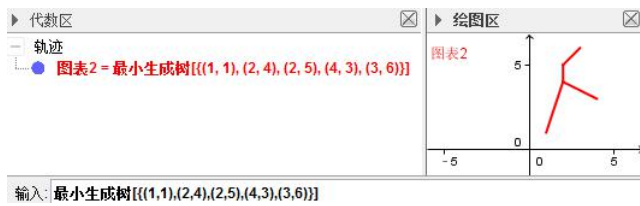
**MinimumSpanningTree(<List of Points>); 最小生成树(<点列>)。**

返回给定权重是边 (u, v) 的 u 和 v 间为欧氏几何距离顶点原图的最小生成树。结果对象是轨迹。

注：一个有 n 个结点的连通图的生成树是原图的极小连通子图，且包含原图中的所有 n 个结点，并且有保持图连通的最少的边。比较上两图，可见最小生成树是凹包内德洛内三角形顶点的另外一种表现顶点的形式。

**最小生成树(<点集>)**

一个有 n 个结点的连通图的生成树是原图的极小连通子图，且包含原图中的所有 n 个结点，并且有保持图连通的最少的边。最小生成树可以用 kruskal（克鲁斯卡尔）算法或 Prim（普里姆）算法求出。



### 释义:

(一) Delaunay (德洛内、狄洛尼) 三角网:

1、定义: 一系列相连但不重叠的三角形的列表, 而且这些三角形的外接圆不包含这个面域的其他人任何点。

2、性质: Delaunay 三角网是唯一的; 每个 Delaunay 三角形的外接圆不包含面内其他任何点, 即 Delaunay 三角网的空外接圆性质。这是创建 Delaunay 三角网的一项判别标准; 三角网的外边界构成了点集 P 的凸多边形“外壳”; 在由点集 V 中所能形成的三角网中, Delaunay 三角网中三角形的最小角度是最大的。

(二) Voronoi 图 (泰森多边形或 Dirichlet 图)

由一组由连接两邻点直线的垂直平分线组成的连续多边形组成。N 个在平面上有区别的点, 按照最近邻原则划分平面; 每个点与它的最近邻区域相关联。Delaunay 三角形是由与相邻 Voronoi 多边形共享一条边的相关点连接而成的三角形。Delaunay 三角形的外接圆圆心是与三角形相关的 Voronoi 多边形的一个顶点。Voronoi 三角形是 Delaunay 图的偶图。

(三) 最小生成树

一个有 n 个结点的连通图的生成树是原图的极小连通子图, 且包含原图中的所有 n 个结点, 并且有保持图连通的最少的边。

1、定义: 一系列相连但不重叠的三角形的列表, 而且这些三角形的外接圆不包含这个面域的其他人任何点。

2、性质: Delaunay 三角网是唯一的; 每个 Delaunay 三角形的外接圆不包含面内其他任何点, 即 Delaunay 三角网的空外接圆性质。这是创建 Delaunay 三角网的一项判别标准; 三角网的外边界构成了点集 P 的凸多边形“外壳”; 在由点集 V 中所能形成的三角网中, Delaunay 三角网中三角形的最小角度是最大的。

## 2.12 List. 列表

### 2.12.1 Flatten. 扁平列表

**Flatten(<List>); 扁平列表(<列表>)**。

将多个列表变为一个。

**案例:** “扁平列表({2, 3, {5, 1}}, {{2, 1, {3}}})” 得出 “list1={2, 3, 5, 1, 2, 1, 3}”。



**注:** 不管子列表有几层, 都是将原来的元素放置在一层列表中。

## 2.12.2 Union. 并集

**Union(<List>, <List>); 并集(<列表 1>, <列表 2>)。**

合并两个列表且去除重复元素。

**案例：**“并集({1, 2, 3, 4, 5}, {3, 2, 1, 7})”得出“{1, 2, 3, 4, 5, 7}”；“并集({4, 2, 3, 1}, {3, 4, 5, 6})”得出“{1, 2, 3, 4, 5, 6}”。



**Union(<Polygon>, <Polygon>); 并集(<多边形 1>, <多边形 2>)。**

新多边形是<多边形 1>和<多边形 2>有重合时所能覆盖的全部联合区域。两个多边形不能自我合并（同一个多边形的自我并集），而且只能用于联合单纯的多边形（非开链多边形，也不适用于圆锥曲线）。

**注：**Union（并集）与Join（合并）不同，后者包含重复元素；创建的重叠多边形不会派生边线；“并集”两个没有重合区域的多边形结果为“未定义”。

## 2.12.3 Insert. 插入

**Insert(<List>, <List>, <Position>); 插入(<列表 1>, <列表 2>, <列表 2 中序号位置>)。**

将列表 1 中的全部元素插入到列表 2 中的指定位置。

**案例：**“插入({11, 12}, {1, 2, 3, 4, 5}, 3)”放置第一列表中的元素插入到第二个列表中的第三个位置（及其后）而得的新列表“{1, 2, 11, 12, 3, 4, 5}”。

**注：**如果位置是一个负整数，则插入的位置从右边数起。



**案例：**“插入({11, 12}, {1, 2, 3, 4, 5}, -2)”将第一个列表中的元素插入到第二个列表中的最后一个元素之前而得的新列表“{1, 2, 3, 4, 11, 12, 5}”。

**Insert(<Object>, <List>, <Position>); 插入(<对象>, <列表>, <列表中序号位置>)。**

将对象插入到列表中的指定位置。

**案例：**“插入( $x^2$ , {1, 2, 3, 4, 5}, 3)”放置“ $x^2$ ”到第三个位置且创建列表“{1, 2,  $x^2$ , 3, 4, 5}”。



**注：**如果位置是一个负整数，则插入的位置从右边数起。

**案例：**“插入( $x^2$ , {1, 2, 3, 4, 5}, -1)”放置  $x^2$  到列表尾部且创建列表“{1, 2, 3, 4, 5,  $x^2$ }”。

## 2.12.4 Product. 乘积

**Product(<List of Raw Data>); 乘积(<原始数据列表>)。**

计算列表中所有数值的乘积。

**案例：**“乘积({2, 5, 8})”输出“a=80”。

**Product(<List of Numbers>, <Number of Elements>); 乘积(<数据列表>, <前若干元素数量>)**。

计算列表中前 n 个数值的乘积。



**案例：**“乘积({1, 2, 3, 4}, 3)”输出“a=6”。

**Product(<List of Numbers>, <List of Frequencies>); 乘积(<数据列表>, <频数列表>)**。

将数集中的每个元素按频数集中对应元素位置的值进行升幂，然后计算乘积。

**案例：**“乘积({20, 40, 50, 60}, {4, 3, 2, 1})”输出“a=1536000000000000”；“乘积({sqrt(2), cbrt(3), sqrt(5), cbrt(-7)}, {4, 3, 2, 3})”输出“a=-420”。



**注：**两个集合的长度必须相等；“乘积({6, 5, 4}, {1, 2, 3})” = “ $6^1 * 5^2 * 4^3 = 9600$ ”。

**Product(<Expression>, <Variable>, <Start Index>, <End Index>); 乘积(<表达式>, <变量>, <起始值>, <终止值>)**。

将变量取从初始值到终止值的每个整数值并分别代入表达式求得表达式的计算结果，计算所有结果的乘积。

**案例：**“乘积(x+1, x, 2, 3)”输出“12”。即  $(2+1) \times (3+1) = 12$ 。



### CAS Syntax (运算区语法)

**Product(<List of Expressions>); 乘积(<表达式列表>)**。

计算列表中所有元素的乘积。

**案例：**“乘积({1, 2, x})”输出“2x”。

**Product(<Expression>, <Variable>, <Start Index>, <End Index>); 乘积(<表达式>, <变量>, <起始值>, <终止值>)**。

将变量取从初始值到终止值的每个整数值并分别代入表达式求得表达式的计算结果，计算所有结果的乘积。

**案例：**“乘积(x+1, x, 2, 3)”输出“12”。即  $(2+1) \times (3+1) = 12$ 。

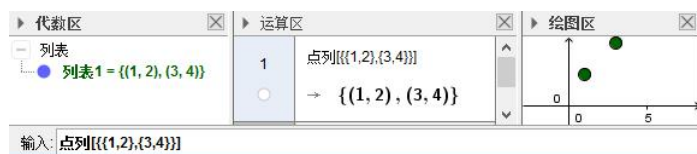


## 2.12.5 PointList. 点列

**PointList(<List>); 点列(<有序数对列表的列表>)**。

依据有序数对列表创建点列。

**案例：**“点列({{1, 2}, {3, 4}})”返回“{(1, 2), (3, 4)}”。



注：指定列表须由双元素的子列表构成。

## 2.12.6 Join. 合并

**Join(<List of Lists>); 合并(<列表的列表>)。**

将多个子列表合并为一个较长列表。

注：新列表包含原有列表中的所有元素，即使是重复的元素，新列表中的元素不会重新排序。

案例：“合并({{1, 2}})”创建列表“{1, 2}”；“合并({{1, 2, 3}, {3, 4}, {8, 7}})”创建列表“{1, 2, 3, 3, 4, 8, 7}”。



注：只是将第一层的列表元素合并到一起，并不打开更深层的列表元素。

**Join(<List>, <List>, ... ); 合并(<列表 1>, <列表 2>, ... )。**

合并两个（或更多）列表。

注：新列表包含原有列表中的所有元素，即使是重复的元素，新列表中的元素不会重新排序。

案例：“合并({5, 4, 3}, {1, 2, 3})”创建列表“{5, 4, 3, 1, 2, 3}”。

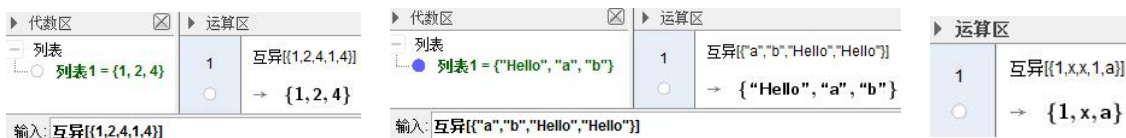


## 2.12.7 Unique. 互异

**Unique(<List>); 互异(<列表>)。**

返回将指定列表中的元素升序排列后的新列表，重复元素只记一次。可用于数字列表和文本列表。

案例：“互异({1, 2, 4, 1, 4})”得出“{1, 2, 4}”；“互异({"a", "b", "Hello", "Hello"})”得出“{"Hello", "a", "b"}”



**CAS Syntax (运算区语法)**

**Unique(<List>); 互异(<列表>)。**

返回将指定列表中的元素升序排列后的新列表，重复元素只记一次。

案例：“互异({1, x, x, 1, a})”得出“{1, x, a}”。

注：参见“频数”指令。

## 2.12.8 Intersection. 交集

**Intersection(<List>, <List>); 交集(<列表 1>, <列表 2>)**。

给出包含两个列表中所有共有元素的列表。

**案例：**设 “list={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}” 和 “list1={2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30}” ， “交集(list, list1)” 得出一个新列表 “list2={2, 4, 6, 8, 10, 12, 14}” ； “交集({1, 2, 3, 4}, {3, 4, 5, 6})” 得出 “{3, 4}” 。



**注：**不要与 Intersect（交点）指令混淆。也不同于并集，交集指令只能用于列表类型的对象，而并集指令还能用于多边形。

## 2.12.9 Reverse. 逆序排列

**Reverse(<List>); 逆序排列(<列表>)**。

将列表中的元素逆序（比照原来的顺序）排列。

**案例：**“逆序排列(list1)” 倒转 “list1={(1, 2), (3, 4), (5, 6)}” 创建 “list2={(5, 6), (3, 4), (1, 2)}” 。



CAS Syntax（运算区语法）

**Reverse(<List>); 逆序排列(<列表>)**。

将列表中的元素逆序排列。

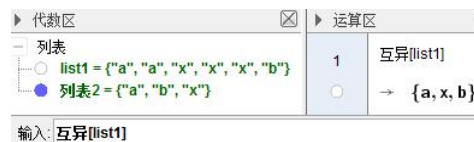
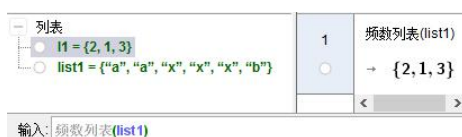
**案例：**“逆序排列({1, 2, 3, 4})” 倒转列表创建 “{4, 3, 2, 1}” 。

## 2.12.10 Frequency. 频数列表

**Frequency(<List of Raw Data>); 频数列表(<原始数据列表>)**。

输出一个指定的原始数据列表中各个互异值出现次数的列表。原始数据列表中的元素可以是数值或文本；所得频数集中的各个频数按互异值的升序排列。若要得出对应的互异值集合可以使用“互异”指令。

**案例：**输入 “list1={“a”, “a”, “x”, “x”, “x”, “b”}” ， “频数列表(list1)” 返回列表 “{2, 1, 3}” 。“互异(list1)” 返回列表 “{“a”, “b”, “x”}” 。在 list 中，a 是 2 个、b 是 1 个、x 是 3 个。



**Frequency(<Boolean Cumulative>, <List of Raw Data>); 频数列表(<是否累积? true|false>, <原始数据列表>)**。

输出一个列表，若是否累积?=false，返回的频数列表等同于“频数列表(<原始数据列表>)”；若是否累积?=true，则将“频数列表(<原始数据列表>)”中的频数进行累计。

**案例：**输入 “list1={0, 0, 0, 1, 1, 2}” 。“频数列表(true, list1)” 返回列表 “{3, 5, 6}” 。“频数列表(false, list1)” 返回列表 “{3, 2, 1}” 。“互异(list1)” 返回列表 “{0, 1, 2}” 。





在列表 list1 中，0 有 3 个、1 有 2 个、2 有 1 个。如果不累加得到列表 “{3,2,1}”。如果累加得到列表 “{3,3+2,3+2+1}” = “{3,5,6}”。

Frequency(<List of Class Boundaries>, <List of Raw Data>); 频数列表(<组界列表>, <原始数据列表>)。

输出一个列表，各个频数的值是以区间 (a, b) 为间隔范围中的原始数据个数，由组界列表所确定的所有组的组界 a 和 b 是一对连续延伸的数值。最大分组的区段是区间 (a, b)。

案例：“频数列表({1, 2, 3}, {1, 1, 1, 2, 3})” 返回列表 “{3, 2}”。列表 “{1, 2, 3}” 给出了两个组限为半闭半开区间 (1, 2)、(2, 3)。列表 “{1, 1, 1, 2, 3}” 中，属于第一个区间的有 3 个，第二个区间的有 2 个。



Frequency(<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>); 频数列表(<是否累积? true|false>, <组界列表>, <原始数据列表>)。

输出一个列表，若是否累积?=false，返回的频数列表等同于频数列表(<组界列表>, <原始数据列表>); 若是否累积?=true，则将频数列表(<组界列表>, <原始数据列表>)中的频数进行累计。

Frequency(<List of Class Boundaries>, <List of Raw Data>, <Use Density>, <Density Scale Factor>(optional) ); 频数列表(<组界列表>, <原始数据列表>, <应用密度>, <密度缩放因子(可选)>)

返回一个类似于直方图指令的频数列表（或密度频数列表）。若应用密度=false，返回的频数列表等同于频数列表(组界列表, <原始数据列表>)。若应用密度=true，返回每组密度频数的频数列表。

案例：设原始数据列表 “data={1, 2, 2, 2, 3, 3, 4, 4, 4, 4}” 和组界列表 “classes={0, 2, 5}”；因此 “频数列表(classes, data, false)” 和 “频数列表(classes, data)” 都会得出频数列表 “{1, 9}”，而 “频数列表(classes, data, true)” 返回密度频数列表 “{0.5, 3}”。



Frequency(<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>, <Use Density>, <Density Scale Factor>(optional) ); 频数列表(<是否累积? true|false>, <组界列表>, <原始数据列表>, <应用密度>, <密度缩放因子(可选)>)

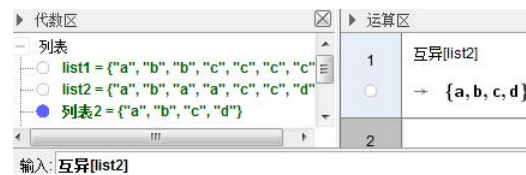
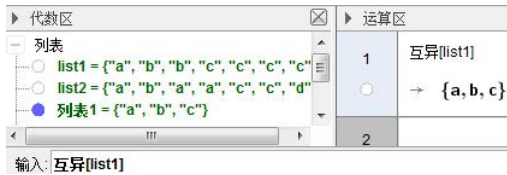
返回一个类似于直方图指令的频数列表（或密度频数列表），可选择频数是否累积。

Frequency(<List of Text>, <List of Text>); 频数列表(<文本列表 1>, <文本列表 2>)

返回一个矩阵，其为两个文本列表中所有可能的字符配对次数（两个文本对应位置的字符进行配对）。矩阵的行对应文本 1 中的互异值（字符），列对应文本 2 中的互异值（字符）。分别使用互异指令可得到各自的互异值（即文本 1 和文本 2 分别含有哪些字符，不计重复）。

案例：

设 “list1={“a”,“b”,“b”,“c”,“c”,“c”,“c”}” 和 “list2={“a”,“b”,“a”,“a”,“c”,“c”,“d”}” 。然后, “频数列表(list1,list2)” 返回矩阵。在矩阵中每一行的数字和分别是 1、2、4, 是 list1 中的元素频率。每一列的数字和是 3、1、2、1, 是 list2 中的元素频率。



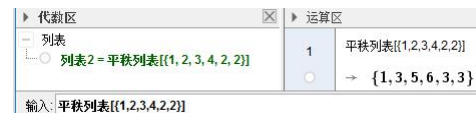
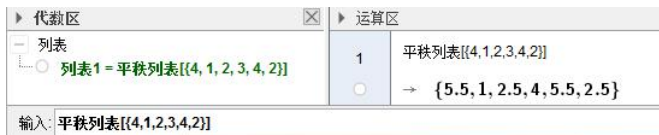
注: 参见“列联表”指令。

### 2. 12. 11 TiedRank. 平秩列表

**TiedRank(<List>); 平秩列表(<列表>)**。

返回一个列表, 其中的元素是给定列表 L 中元素的排序号(排序号使用“排序(L)”获得)。如果在 L 中有多个相同元素, 自动顺延序号, 然后相同元素的排位号取它们平均值。

**案例:** “平秩列表({4, 1, 2, 3, 4, 2})” 返回 “{5.5, 1, 2.5, 4, 5.5, 2.5}”; “平秩列表({1, 2, 3, 4, 2, 2})” 返回 “{1, 3, 5, 6, 3, 3}”。



注: 参见“序数列表”指令。

**案例:** “平秩列表({1, 2, 3, 4, 2, 2})” 得出 “{1, 3, 5, 6, 3, 3}”。先进行排序, 排序({1, 2, 3, 4, 2, 2})={1, 2, 2, 2, 3, 4} 输出的序数赋给对应值, 多个相同值求平均, 从而输出一个对应关系 1=1, 2=3, 3=5, 4=6 然后按照源元素的位置排列序号。

### 2. 12. 12 Remove. 去除

**Remove(<List>, <List>); 去除(<列表 1>, <列表 2>)**

从列表 1 中将列表 2 中的元素去除一次。

**案例:** “去除({1, 3, 4, 4, 9}, {1, 4, 5})” 得到列表 “{3, 4, 9}”。



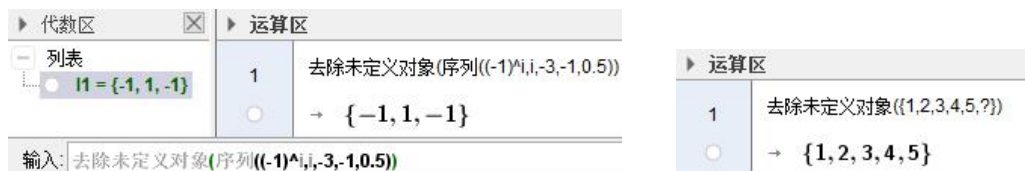
注: 参见“去除未定义对象”指令。如果想得到集合论的不同列表, 键入 “{1, 3, 4, 4, 9}\{1, 4, 5}” (从第一个列表中, 扣除后边列表中指定的所有元素)。

### 2. 12. 13 RemoveUndefined. 去除未定义对象

### RemoveUndefined(<List>); 去除未定义对象(<列表>)。

去除一个列表中未定义的对象。

**案例：**“去除未定义对象(序列 $((-1)^i, i, -3, -1, 0.5)$ )”得到“ $\{-1, 1, -1\}$ ”，去除了序列指令所得列表中的第 2、4 个元素，因为表达式“ $(-1)^{1.5}$ ”和“ $(-1)^{2.5}$ ”无定义；“去除未定义对象( $\{1, 2, 3, 4, 5, ?\}$ )”得到“ $\{1, 2, 3, 4, 5\}$ ”；“去除未定义对象( $\{1, 2, 3, 4, 5, 6\}$ )”得出“ $\{1, 2, 3, 4, 5, 6\}$ ”。



**注：**参见“去除”指令。

## 2.12.14 Sort. 升序排列

### Sort(<List>); 升序排列(<列表>)。

对一个元素类型为数值、文本或点的列表进行排序（升序）而得的新列表。

**注：**点列按 x 轴坐标进行排序。

**案例：**“升序排列( $\{3, 2, 1\}$ )”给出列表“ $\{1, 2, 3\}$ ”；“升序排列( $\{\text{"pears"}, \text{"apples"}, \text{"figs"}\}$ )”给出字母序元素列表；“升序排列( $\{\text{"个人"}, \text{"集体"}, \text{"国家"}\}$ )”得出“ $\{\text{"个人"}, \text{"国家"}, \text{"集体"}\}$ ”；“升序排列( $\{(3, 2), (2, 5), (4, 1)\}$ )”给出“ $\{(2, 5), (3, 2), (4, 1)\}$ ”。



**注：**用文本、字母排序，必须注意标点的规范，原则上引号、逗号都用英文。以上几个图代数区显示的是“定义”不是“数值”。

### Sort(<Values>, <Keys>); 升序排列(<数据列表>, <索引列表>)

将前面列表中的<数值>，按后面的索引列表的对应次序进行排序。

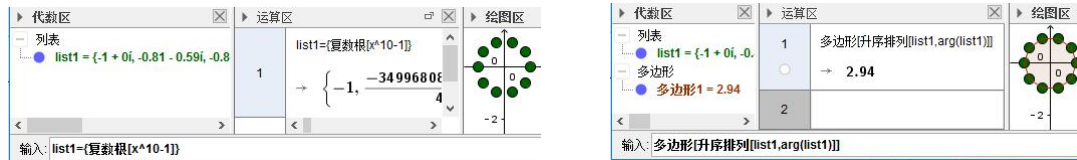
**案例：**“升序排列( $\{1, 5, 2, 3\}, \{1, 2, 3, 1\}$ )”得到“ $\{1, 3, 5, 2\}$ ”。第一个列表中的数据按照第二个列表中分别对应的位次出现在新的列表中。1 和 3 都是在新表的第一位，但 1 小于 3 故顺序为“1, 3”。



**案例：**要将一个多项式的列表“ $list1 = \{x^3, x^2, x^6\}$ ”按次方程度进行排序，首先创建次方数的从属列表“ $list2 = \text{映射}(\text{次数}(a), a, list1)$ ”，然后，使用指令“升序排列( $list1, list2$ )”可得出符合要求的列表“ $list3 = \{x^2, x^3, x^6\}$ ”。



若要绘制具有按参数排序的、像  $x^{10}-1$  的复数根这类形式顶点的多边形，按照参数排序创建“ $list1=\{\text{复数根}(x^{10}-1)\}$ ”，然后使用指令“多边形(升序排列(list1,arg(list1)))” (arg()是复数的幅角函数)，得出“多边形 1=2.94”。



注：可用对象排序的变通方法在列表排序进阶教程中有介绍。

## 2.12.15 RandomElement. 随机元素

**RandomElement(<List>); 随机元素(<列表>)**。

返回从列表中随机选择的元素（相同概率）。列表中元素的类型必须相同。

**案例：**“随机元素({3, 2, -4, 7})”得出列表“{-4, 2, 3, 7}”中的一个元素“随机元素({(0, 1), (1, 2), (2, 3)})”得到“(0, 1)、(1, 2)、(2, 3)”中的一个。

注：参见“元素”指令。

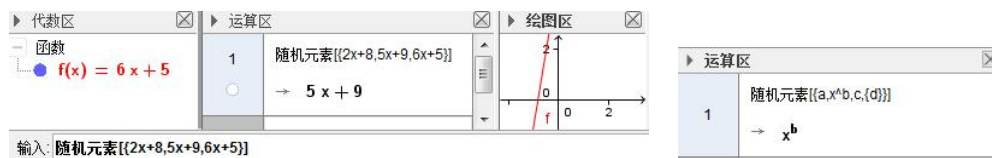


**CAS Syntax (运算区语法)**

**RandomElement(<List>); 随机元素(<列表>)**。

返回从列表中随机选择的元素（相同概率）。列表中元素的类型不必相同。

**案例：**“随机元素({2x+8, 5x+9, 6x+5})”得出列表中任意一个函数。



“随机元素({a,  $x^b$ , c, {d}})”得出列表中任意一个元素。

注：参见“设置种子”、“区间随机数”、“随机二项分布数”、“正态分布随机数”、“泊松分布随机数”和“均匀分布随机数”指令。

## 2.12.16 IndexOf. 索引

**IndexOf(<Object>, <List>); 索引(<对象>, <列表>)**。

返回对象在列表中第一次出现的位置。

**案例：**“索引(5, {1, 3, 5, 2, 5, 4})”返回 3。



注：如果对象不在列表中，结果是未定义。

**IndexOf(<Text>, <Text>); 索引(" <文本 1>", " <文本 2>")。**

指明较短的字符组（文本 1）在整个文本（文本 2）中第一次出现的位置。

案例：“索引("Ge", "GeoGebra")”返回 1。

**IndexOf(<Object>, <List>, <Start Index>); 索引(<对象>, <列表>, <起始索引>)。**

返回对象在列表中第一次出现的位置，但搜索从指定的起始索引开始。

案例：“索引(5, {1, 3, 5, 2, 5, 4}, 3)”返回 3；“索引(5, {1, 3, 5, 2, 5, 4}, 4)”返回 5；“索引(5, {1, 3, 5, 2, 5, 4}, 6)”返回未定义。返回的位置号是在列表中的位置，与索引无关。



**IndexOf(<Text>, <Text>, <Start Index>); 索引(" <文本 1>", " <文本 2>", <起始索引>)。**

返回较短文本在全部文本中第一次出现的位置，但搜索从指定的起始索引开始。

案例：“索引("Ge", "GeoGebra", 2)”返回 4。



## 2. 12. 17 Take. 提取

**Take(<List>, <Start Position>); 提取(<列表>, <起始位置>)。**

返回一个列表，包含原列表从起始位置到结尾的元素。

案例：“提取({2, 4, 3, 7, 4}, 3)”得出“{3, 7, 4}”。



**Take(<Text>, <Start Position>); 提取(" <文本>", <起始位置>)。**

返回一个文本，其包含原有文本从起始位置到结尾的字符。

案例：“提取("GeoGebra", 3)”得出文本“oGebra”。

**Take(<List>, <Start Position>, <End Position>); 提取(<列表>, <起始位置>, <终止位置>)。**

返回一个列表，其包含原有列表从起始位置到终止位置的元素。

案例：“提取({2, 4, 3, 7, 4}, 3, 4)”得出“{3, 7}”。



**Take(<Text>, <Start Position>, <End Position>); 提取(" <文本>", <起始位置>, <终止位置>)。**

返回一个文本，其包含原有文本从起始位置到终止位置的字符。

案例：“提取(“GeoGebra”, 3, 6)”得出“oGeb”。

CAS Syntax (运算区语法)

Take(<List>, <Start Position>); 提取(<列表>, <起始位置>)。

返回一个列表，包含原列表从起始位置到结尾的元素。

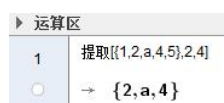
案例：“提取({2, 4, 3, 7, 4}, 3)”得出“{3, 7, 4}”。



Take(<List>, <Start Position>, <End Position>); 提取(<列表>, <起始位置>, <终止位置>)。

返回一个列表，其包含原有列表从起始位置到终止位置的元素。

案例：“提取({1, 2, a, 4, 5}, 2, 4)”得出“{2, a, 4}”。



## 2. 12. 18 Sequence. 序列

Sequence(<Expression>, <Variable i>, <Start Value a>, <End Value b>); 序列(<表达式>, <变量>, <起始值>, <终止值>)。

得出一个列表，其元素为：将变量从起始值开始到终止值结束依次增加1，将变量的每个取值代入表达式而得的结果（数值或对象）。

案例：“序列((2, i), i, 1, 5)”创建一个 y 坐标值从 1 到 5 的点列：“{(2, 1), (2, 2), (2, 3), (2, 4), (2, 5)}”。



Sequence(<Expression>, <Variable i>, <Start Value a>, <End Value b>, <Increment>); 序列(<表达式>, <变量>, <起始值>, <终止值>, <增量>)。

得出一个列表，其元素为：将变量从起始值开始到终止值结束依次增加增量，将变量的每个取值代入表达式而得的结果（数值或对象）。

案例：“序列((2, i), i, 1, 3, 0.5)”创建一个点的 y 坐标从 1 增加为 3，增量为 0.5 的点列：“{(2, 1), (2, 1.5), (2, 2), (2, 2.5), (2, 3)}”。



注：由于起始值和终止值可以是动态值，可以使用滑动条作为参数。

Sequence(<End Value b>); 序列(<终止值>)。

创建从 1 开始（依次递增 1）到终止值结束为止的数集。

案例：“序列(4)”得到“{1, 2, 3, 4}”；“2^序列(4)”得到“{2, 4, 8, 16}”。



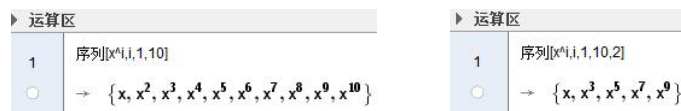
注：更多关于列表操作参见“列表”。

### CAS Syntax (运算区语法)

Sequence(<Expression>, <Variable i>, <Start Value a>, <End Value b>); 序列(<表达式>, <变量>, <起始值>, <终止值>).

得出一个列表，其元素为：将变量从起始值开始到终止值结束依次增加1，将变量的每个取值代入表达式而得的结果（数值或对象）。

案例：“序列( $x^i$ ,  $i$ , 1, 10)”生成序列列表“ $\{x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}\}$ ”。



Sequence(<Expression>, <Variable i>, <Start Value a>, <End Value b>, <Increment>); 序列(<表达式>, <变量>, <起始值>, <终止值>, <增量>).

得出一个列表，其元素为：将变量从起始值开始到终止值结束依次增加增量，将变量的每个取值代入表达式而得的结果（数值或对象）。

案例：“序列( $x^i$ ,  $i$ , 1, 10, 2)”生成序列“ $\{x, x^3, x^5, x^7, x^9\}$ ”。

注：由于起始值和终止值可以是动态值，可以使用滑动条作为参数。

Sequence(<End Value b>); 序列(<终止值>).

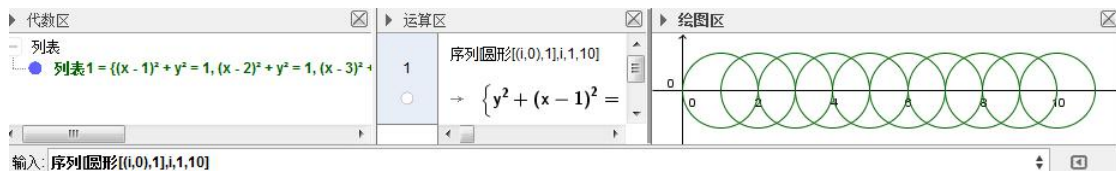
创建从1开始（依次递增1）到终止值结束为止的数集。

案例：序列(5)生成序列{1, 2, 3, 4, 5}。

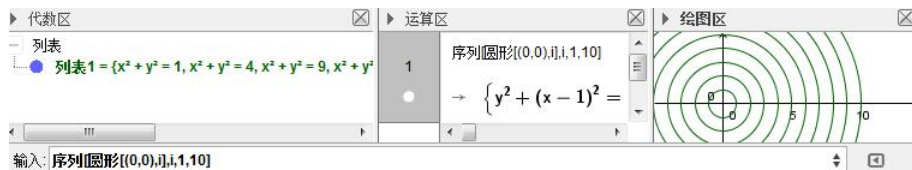
注：更多关于列表操作参见“列表”。

附：通过序列命令可做出一系列物件（自己可以在软件上尝试）：

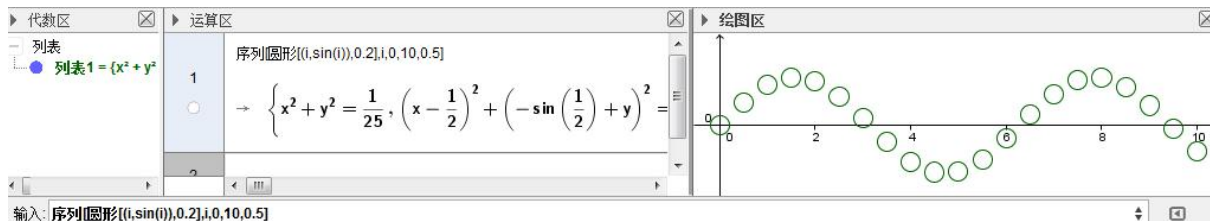
“序列(圆周(( $i$ , 0), 1),  $i$ , 1, 10)”：



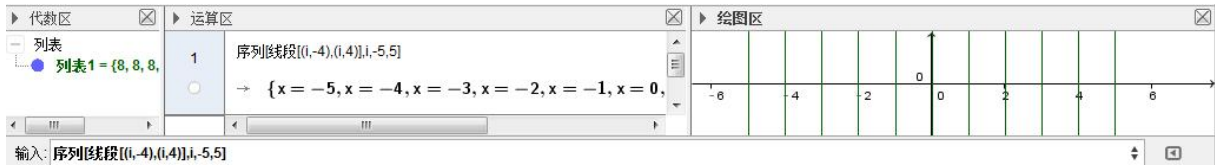
“序列(圆周((0, 0),  $i$ ),  $i$ , 1, 10)”：



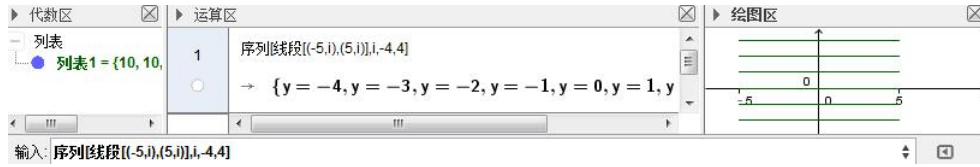
“序列(圆周(( $i$ ,  $\sin(i)$ ), 0.2),  $i$ , 0, 10, 0.5)”：



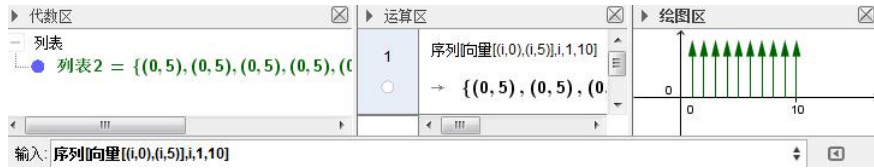
“序列(线段(( $i$ , -4), ( $i$ , 4)),  $i$ , -5, 5)”：



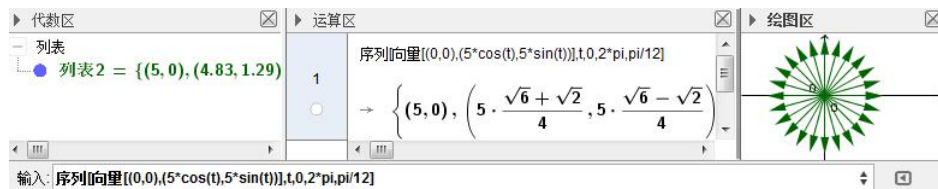
“序列(线段((-5, i), (5, i)), i, -4, 4)” :



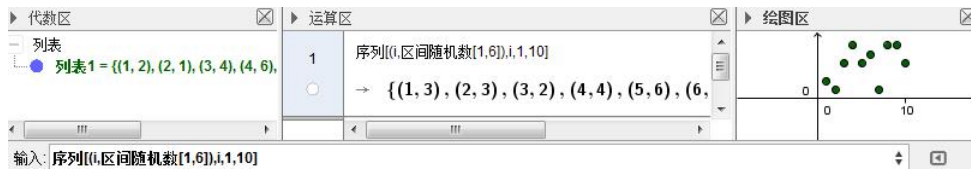
“序列(向量((i, 0), (i, 5)), i, 1, 10)” :



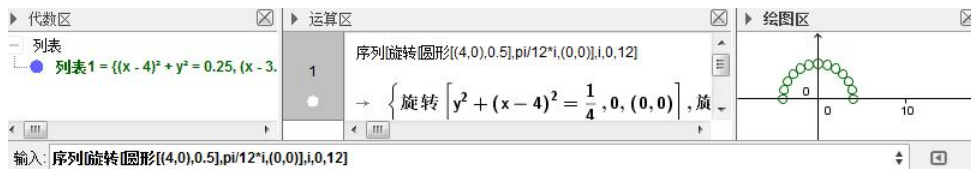
“序列(向量((0, 0), (5\*cos(t), 5\*sin(t))), t, 0, 2\*pi, pi/12)” :



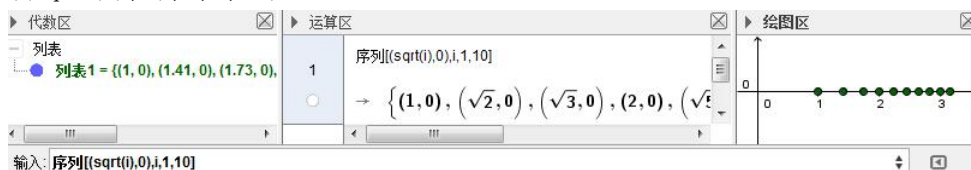
“序列((i, 区间随机数(1, 6)), i, 1, 10)” :



“序列(Rotate(Circle((4, 0), 0.5), pi/12\*i, (0, 0)), i, 0, 12)” :



“序列((sqrt(i), 0), i, 1, 10)” :



“序列(圆周((5\*cos(t), 3\*sin(t)), 0.4), t, 0, 2pi, pi/50)” :



案例: Sequence(<Start value i>, <End value n>); 序列(<起始值>, <终止值>).  
创建从起始值到终止值的整数列表(升序或者降序)



**案例：**“序列(7, 13)”创建列表“{7, 8, 9, 10, 11, 12, 13}”；“序列(18, 14)”创建列表“{18, 17, 16, 15, 14}”；“序列(-5, 5)”创建列表“{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5}”。

**注：**可以使用简单的语法取代“序列(7, 13)”格式的指令，“7..13”可以得到同样的结果。



## 2.12.19 OrdinalRank. 序数列表

**OrdinalRank(<List>); 序数列表(<列表>).**

返回一个列表，其元素是指定列表L中元素的排序值（元素排序使用“排序(L)”）。如果列表L中有多个相同元素，会在排序中占据下一个位置，此序位会关联这些元素。

**案例：**“序数列表({4, 1, 2, 3, 4, 2})”返回“{5, 1, 2, 4, 6, 3}”；“序数列表({3, 2, 2, 1})”返回“{4, 2, 3, 1}”。



**注：**参见“平秩列表”指令。

## 2.12.20 SelectedIndex. 选定索引

**SelectedIndex(<List>); 选定索引(<列表>).**

返回一个下拉列表中当前所选择元素的对应索引。



**注：**参见“选定元素”指令；默认返回1。

## 2.12.21 SelectedElement. 选定元素

**SelectedElement(<List>); 选定元素(<列表>).**

返回一个在列表对应的下拉列表中当前所选择元素。

**案例：**“选定元素({(2, 2), 2, 1, 4, 5, 6})”得出“(2, 2)”。



注：参见“选定索引”指令；默认输出列表中的第一个元素。

## 2.12.22 Zip. 映射

Zip(<Expression>, <Var1>, <List1>, <Var2>, <List2>, ...); 映射(<表达式>, <变量 1>, <列表 1>, <变量 2>, <列表 2>, ...).

生成一个新列表，其元素是：将指定列表中相应位置的元素依次作为变量代入表达式而取得的结果。新列表的长度是输入的列表中最短的长度。

案例：设 P, Q, R, S 是一些点，“映射(中点(A, B), A, {P, Q}, B, {R, S})”返回一个取线段 PR 和 QS 中点的列表 {PR 的中点坐标, QS 的中点坐标}。



本例中，表达式为“中点(A, B)”；列表 {P, Q} 中的两个元素依次作为 A，列表 {R, S} 中的两个元素依次作为 B，参与表达式“中点(A, B)”的运算；结果就是求 PR 中点和 QS 中点。

案例：设多项式的列表 “list1={x^2, x^3, x^6}”，“映射(次数(a), a, list1)”返回列表 “{2, 3, 6}”。



案例：“映射(n+m, n, {1, 2, 3}, m, {1, 2})”得到 “{2, 4}”。

第二个列表只有两个元素，故结果列表长度就是 2。

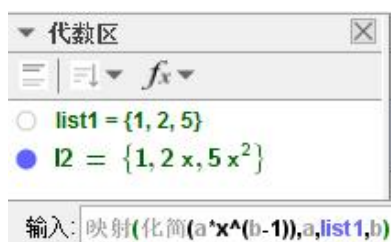
注：每个列表中的元素必须是同一类型；映射指令类似于其它编程语言中被称为“map”的概念。

事实上，映射指令完全可以只使用单一列表作为参数。这使它在需要对一个列表中的元素进行逐个调用的时候，相比序列指令更为简短便捷从而替代使用。例如：“映射(a^2, a, 数集)”就比“序列(元素(数集, a)^2, a, 1, 长度(数集))”简捷得多。



案例：“映射(f(2), f, {x+1, x+3})”得到 “{3, 5}”。

案例：假设 list1={1, 2, 5}，“映射(化简(a\*x^(b-1)), a, list1, b)”得到 “{1, 2x, 5x^2}”。



## 2.12.23 Element. 元素

**Element(<List>, <Position of Element n>); 元素(<列表>, <元素位置>)**。  
调用列表中的第 n 个元素。



**案例：**“元素({1, 3, 2}, 2)”得出 3，列表“{1, 3, 2}”中的第二个元素。

**Element(<Matrix>, <Row>, <Column>); 元素(<矩阵>, <行序>, <列序>)**。

调用矩阵中指定行列（行序、列序）位置上的元素。

**案例：**“元素({{1, 3, 2}, {0, 3, -2}}, 2, 3)”得出 -2，矩阵第二行的第三个元素。

**Element(<List>, <Index1>, <Index2>, ...); 元素(<列表>, <索引 1>, <索引 2>, ...)**。

给定的 n 维集合或矩阵，可以设置 n 个索引，以调用指定坐标位置上的元素（或指定范围内的元素集合或矩阵）。

**案例：**设“ $L = \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$ ”。然后“元素(L, 1, 2, 1)”得出 3（第一行，第二组，第一个），“元素(L, 2, 2)”得出“{7, 8}”（第二行，第二组）。



**注：**这个指令只能针对列表或矩阵本身包含一种类型的对象（只是数值或点等）。

### CAS Syntax (运算区语法)

**Element(<List>, <Position of Element n>); 元素(<列表>, <元素位置>)**。

得出列表中的第 n 个元素。

**案例：**“元素({a, b, c}, 2)”得出 b，列表{a, b, c}中第二个元素。

**Element(<Matrix>, <Row>, <Column>); 元素(<矩阵>, <行序>, <列序>)**。

得出矩阵中指定行列（行序、列序）位置上的元素。

**案例：**“元素({{a, b, c}, {d, e, f}}, 2, 3)”得出 f，第二行第三个元素。



**注：**参见“最前元素”指令、“最后元素”指令和“随机元素”指令。

## 2. 12. 24 Append. 追加

**Append(<List>, <Object>); 追加(<列表>, <对象>)**。

向列表中追加对象。

**案例：**“追加({1, 2, 3}, 4)”创建列表“{1, 2, 3, 4}”。

**Append(<Object>, <List>); 追加(<对象>, <列表>)**。

向对象追加一个列表。

**案例：**“追加(4, {1, 2, 3})”创建列表“{4, 1, 2, 3}”。



**案例：**在原有的列表中增加一个新的元素。假设“ $A=\{1, 2, 3\}$ ”，“赋值(A,追加(A,4))”可将A的值改为 $\{1, 2, 3, 4\}$ 。等同于“赋值(A,4,4)”，参见“赋值”指令。不能使用“A=追加(A,4)”，否则会出现“循环式定义”错误。

## 2.12.25 Classes. 组限

**Classes**(**<List of Data>**,**<Start>**,**<Width of Classes>**)；**组限**(**<数据列表>**,**<起点>**,**<组的宽度>**)。

给出数据列表的组限。第一个界值（最小值）等于起点，最后一个界值（最大值）至少是数集中的最大值；分组指令将最小值与最大值之间的距离分隔成等距（组的宽度）的区段。

**案例：**“组限( $\{0.1, 0.2, 0.4, 1.1\}, 0, 1$ )”给出“ $\{0, 1, 2\}$ ”。



**Classes**(**<List of Data>**,**<Number of Classes>**)；**组限**(**<数据列表>**,**<组的数量>**)。

给出数据列表的组限。第一个界值（最小值）等于数据列表中的最小值，最后一个界值（最大值）是数据列表中的最大值；分组指令将最小值与最大值之间的距离分隔成等距的n个区段。

**案例：**“组限( $\{1, 3, 5, 7, 8, 9, 10\}, 3$ )”给出“ $\{1, 4, 7, 10\}$ ”。

**注：**按照约定最后一组使用  $a \leq x \leq b$  外，每组使用  $a \leq x < b$  规则确定组界。

## 2.12.26 Last. 最后元素

**Last**(**<List>**)；**最后元素**(**<列表>**)。

给出只包含指定列表中最后一个元素的新列表。

**案例：**“最后元素( $\{1, 4, 3\}$ )”得出列表“ $\{3\}$ ”。



**注：**想得到最后一个元素使用“元素( $\{1, 4, 3\}, 3$ )”。

**Last**(**<List>**,**<Number of elements>**)；**最后元素**(**<列表>**,**<后若干元素数量>**)。

给出包含指定列表中最后n个元素的新列表。

**案例：**“最后元素( $\{1, 4, 3\}, 2$ )”得出“ $\{4, 3\}$ ”。

**Last**(**<Text>**)；**最后元素**(**"<文本>"**)。

给出文本的最后一个字符。



**案例：**“最后元素("Hello")”得出“o”。

**Last**(**<Text>**,**<Number of elements>**)；**最后元素**(**"<文本>"**,**<后若干字符数量>**)。

给出文本的后n个字符。

**案例：**“最后元素("Hello",2)”得出“lo”。

**CAS Syntax** (运算区语法)

**Last(<List>); 最后元素(<列表>)。**

给出只包含指定列表中最后一个元素的新列表。

**案例：**“最后元素({1, 4, 3})”得出“{3}”。

**注：**想得到最后一个元素使用元素({1, 4, 3}, 3)。

**Last(<List>, <Number of elements>); 最后元素(<列表>, <后若干元素数量>)。**

给出包含指定列表中最后 n 个元素的新列表。

**案例：**“最后元素({1, 4, 3}, 2)”得出“{4, 3}”。

**注：**参见“最前元素”指令。

## 2.12.27 First. 最前元素

**First(<List>); 最前元素(<列表>)。**

给出只包含指定集合中第一个元素的新集合。

**案例：**“最前元素({1, 4, 3})”得出“{1}”。

**注：**调用集合内的第一个元素可以使用指令“元素({1, 4, 3}, 1)”。



**First(<List>, <Number n of elements>); 最前元素(<列表>, <前若干元素数量>)。**

给出包含指定集合中前 n 个元素的新集合。

**案例：**“最前元素({1, 4, 3}, 2)”得出{1, 4}。

**First(<Text>); 最前元素("<文本>")。**

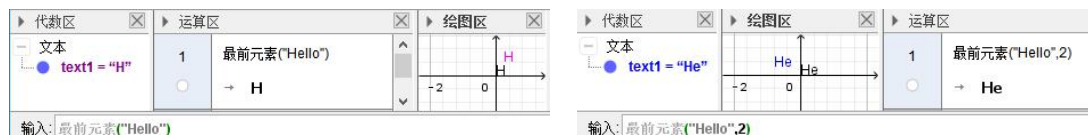
给出文本的第一个字符。

**案例：**“最前元素("Hello")”给出“H”。

**First(<Text>, <Number n of elements>); 最前元素("<文本>", <前若干元素数量>)。**

给出文本的前 n 个字符。

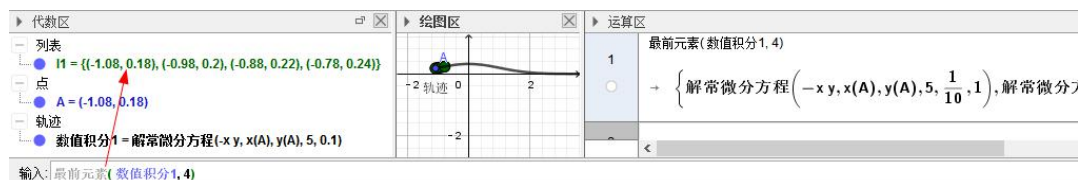
**案例：**“最前元素("Hello", 2)”得出“He”。



**First(<Locus>, <Number n of elements>); 最前元素(<轨迹>, <前若干元素数量>)。**

这个指令适用于：

由解常微分方程指令生成的轨迹，本指令返回常微分方程求解算法的前 n 个步骤解出的数值所对应的一系列点。



**注：**使用“最短距离”、“旅行商问题”、“Voronoi 图”、“最小生成树”和“凸包”指令这些指令生成的轨迹，这个指令返回图形的顶点。

**CAS Syntax (运算区语法)**

**First(<List>); 最前元素(<列表>)。**

给出只包含指定集合中第一个元素的新集合。

**案例：**最前元素( $\{1, 4, 3\}$ )得出 $\{1\}$ 。

**注：**想得到第一个元素使用元素( $\{1, 4, 3\}, 1$ )。

**First(<List>, <Number n of elements>); 最前元素(<列表>, <前若干元素数量>)。**

给出包含指定集合中前 n 个元素的新集合。

**案例：**最前元素( $\{1, 4, 3\}, 2$ )得出 $\{1, 4\}$ 。

**注：**参见“最后元素”指令。

## 2.13 Logical. 逻辑

### 2.13.1 Relation. 关系

**Relation(<List>); 关系(<列表>)**

显示一个消息框，提供有关两个或更多（最多 4 个）对象之间关系的信息。

**Relation(<Object>, <Object>); 关系(<对象 1>, <对象 2>)。**

在弹出的信息框中显示两个对象之间关系的信息。

该指令能够比较出的关系：

两点、两线段、两多边形、两函数：是否相等。

两线(除两线段之外的情况)：是相交、平行、还是垂直关系。

两圆：相等、相交、不相交。

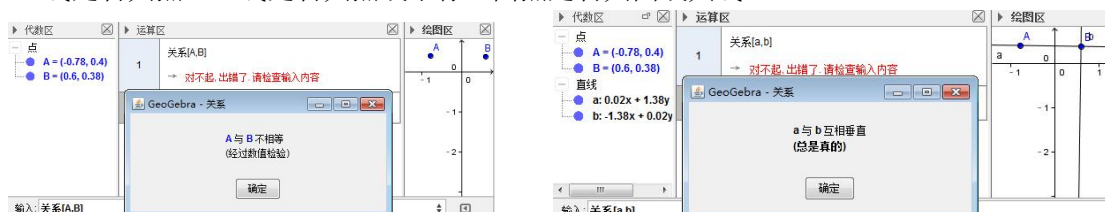
一点与一线(函数)：点是否在线(函数)上。

点与多边形(圆)：点是否在多边形的边界上(不能确定是否在多边形内)。

线段与多边形：是否相等。

线与圆：是否相交、相切。

三线是否共点、三线是否共点或平行、四点是否共圆或共线。



在运算区运行指令，尽管提示出错，但也有关系结果显示出来。



**注：**参见  $a^?b$  “关系判断”工具。

### 2.13.2 If. 如果

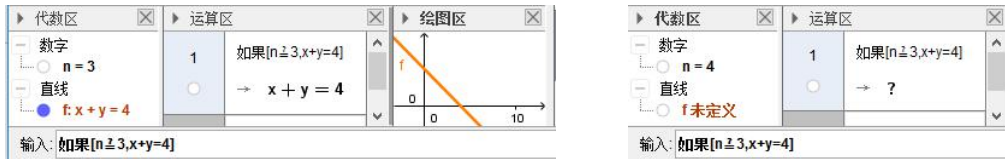
**If(<Condition>, <Then>); 如果(<条件>, <结果>)。**

如果条件满足，创建一个对象；如果条件不满足，创建一个未定义对象。

**案例：**

设  $n=3$ ，“如果( $n \neq 3, x+y=4$ )”得出直线“ $x+y=4$ ”。设  $n=4$ ，“如果( $n \neq 3, x+y=4$ )”创建一个

未定义对象。



**释义：**指令的“条件”部分是难点，这里需要进行一些关系判定，例如大小是否相等、是否是包含关系等。常用到的“条件”有：（判断是否相等用“==”（也可用辅助输入“≐”））

真假值判断：真： $a \doteq \text{true}$ ；假： $a \doteq \text{false}$ 。

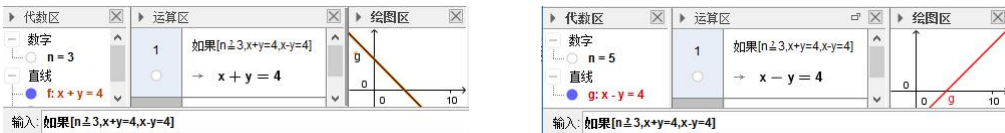
数值判断： $a \doteq 5$ 、 $a > 5$ 、 $a < 5$ 、 $a \neq 5$ 、 $a \leq 5$ 、 $a \geq 5$ 、 $A = B$ 、 $A \neq (0, 0)$ 、 $x(A) > 5$ 、 $y(A) < 10$ 。

包含关系： $a \in \{1, 3, 4, 5\}$ 、 $\{A\} \subseteq \{(0, 0)\}$ 、 $\{x(A)\} \subset \{1, 2, 3, 4, 7\}$ 。

If(<Condition>, <Then>, <Else>); 如果(<条件>, <结果>, <否则>).

如果条件满足，创建一个对象；如果条件不满足，创建另一个对象。所有对象必须是相同的类型。

**案例：**设  $n$  为一个数值型对象，“如果( $n \doteq 3, x+y=4, x-y=4$ )”，当  $n=3$  时得出直线“ $x+y=4$ ”，当  $n$  不等于 3 时得出直线“ $x-y=4$ ”。



“如果指令”可以用于创建条件函数。条件函数可以在许多指令中当做参数使用，比如导数，积分和交点。

**释义：**与前一个指令类似，将这个指令直接输入，起到的效果是创建对象，直接将对象的定义写为“条件”，不过这个指令创建出的对象等于是两个对象（前一个指令是有无一个对象的效果），只要条件改变，这个对象会跟着发生改变。与上面指令一样这个指令也可以与设定值指令嵌套使用，从而更好的完成课件的交互效果。

① “如果”指令中除了嵌套设定值之外，也可以和设定对象属性（绘图区属性）的指令配合，起到动态更改对象的状态（绘图区属性）的作用。

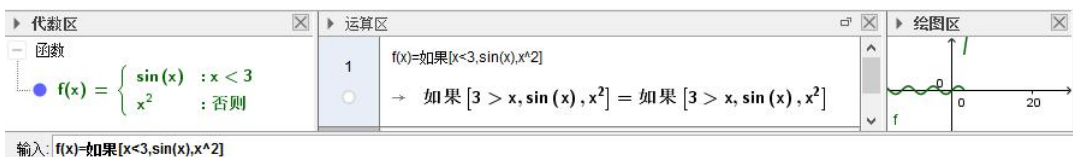
可以配合的指令有：

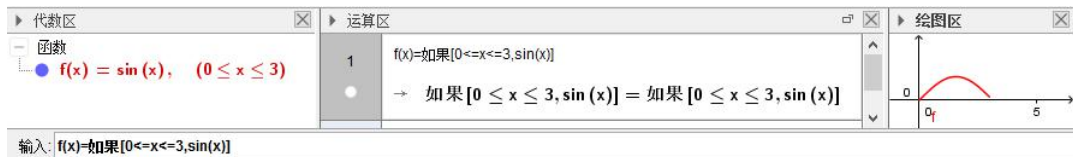
背景色、标签模式、标题、点径、点型、动态颜色、放大、更新作图、工具提示模式、开启动画、平移视图、删除、设置坐标、缩小、显示标签、显示对象、显示图层、显示网格、显示坐标轴、线宽、线型、颜色、隐藏图层、运行单击脚本、运行更新脚本、中心定位、跟踪、坐标轴比例等。

② “如果”指令在与其他指令嵌套时往往会有两种情况：一般使用其他指令嵌套“如果”指令，有时也用“如果”指令嵌套其他指令。一般来说，“如果”指令没有“否则”的时，是“如果”指令嵌套其他(其他指令作为“如果”指令的执行对象)。“如果”指令有“否则”时，是其他嵌套“如果”指令（“如果”指令得到的结果作为其他指令的一部分）。

语法示例：“如果( $a > 0$ , 中心定位( $0, 0$ ))”；“如果( $a > 0$ , 放大( $1$ ))”；“如果( $x(A) > 0$ , 设置标签模式( $A, 1$ ))”；“开启动画(如果( $a > 0$ , true, false))”；“设置显示条件( $A$ , 如果( $x(A) > 0$ , true, false))”；“显示网格(如果( $a > 0$ , true, false))”；“设置标签模式( $A$ , 如果( $x(A) > 0$ , 1, 2))”；“设置跟踪( $A$ , 如果( $x(A) > 0$ , true, false))”

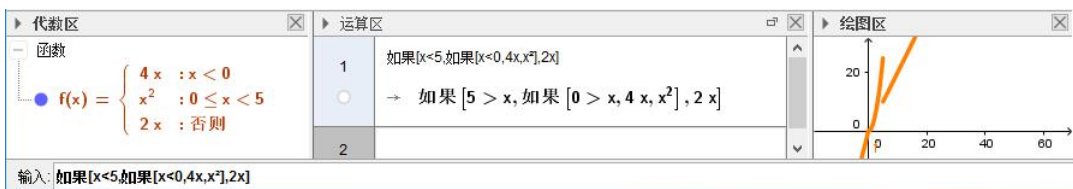
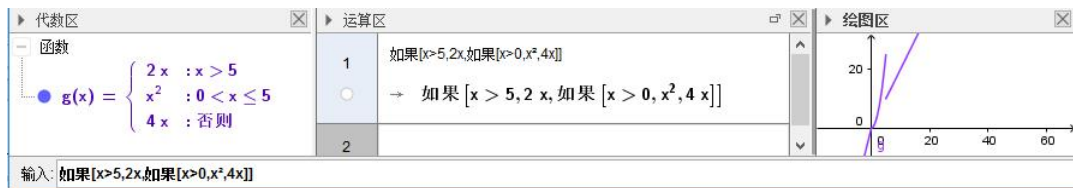
**案例：**“ $f(x) = \text{如果}(x < 3, \sin(x), x^2)$ ”得出分段函数：“如果  $x < 3, \sin(x)$ ”和“如果  $x \geq 3, x^2$ ”；“ $f(x) = \text{如果}(0 < x \leq 3, \sin(x))$ ”得出函数：“ $x$  在 0 和 3 之间  $\sin(x)$ ”（其它情况为未定义）。



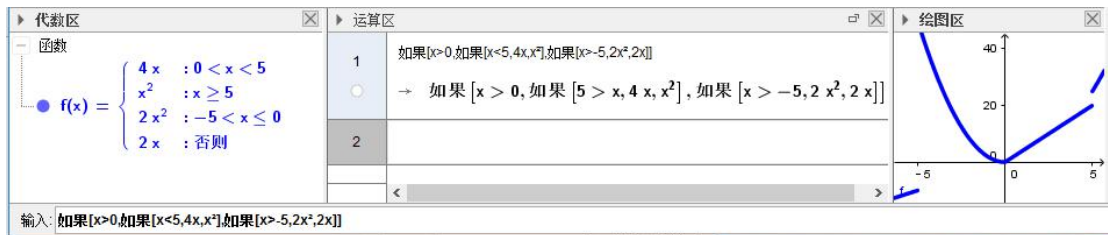


用“如果”指令可以非常方便的创建出分段函数，被视为一般的引用参数，可接受自变量函数作为其中一个参数的指令，例如微分、积分，还有相交等指令。通常一个“如果”指令可以将函数分为两段（一般输入的是  $x > a$ ，或  $x < a$ ，即便输入其他的表达式  $0 < x < 10$ ，虽然分为了三段，但是另外两端的函数关系是一样的）。如果想创建一个多段的函数，可用多个“如果”指令嵌套。

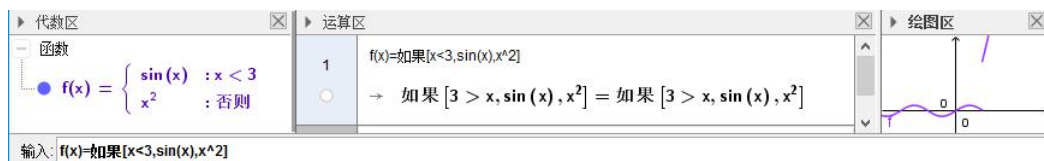
三段常用的方式是：“如果  $(x > 5, 2x, \text{如果}(x > 0, x^2, 4x))$ ”或“如果  $(x < 5, \text{如果}(x < 0, 4x, x^2), 2x)$ ”。



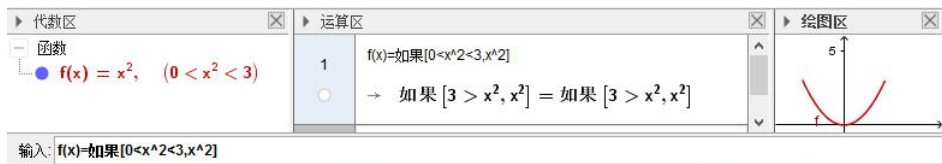
四段常用的方式是：“如果  $(x > 0, \text{如果}(x < 5, 4x, x^2), \text{如果}(x > -5, 2x^2, 2x))$ ”。



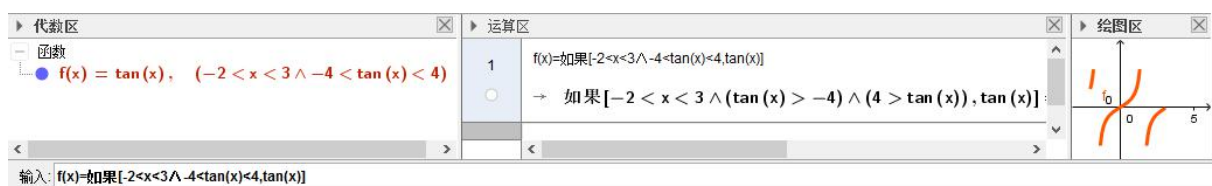
“ $f(x) = \text{如果}(x < 3, \sin(x), x^2)$ ”定义分段函数  $f(x)$ ，相当于：“ $f(x) = \text{如果}(0 \leq x \leq 3, \sin(x))$ ”，同时限定函数的定义域和值……。



“ $f(x) = \text{如果}(0 < x^2 < 3, x^2)$ ”定义函数“ $f(x) = x^2$ ”，其值域是开区间“(0, 3)”。



“ $f(x) = \text{如果}(-2 < x < 3 \wedge -4 < \tan(x) < 4, \tan(x))$ ”定义函数“ $f(x) = \tan(x)$ ”，其定义域是开区间“(-2, 3)”，值域是开区间“(-4, 4)”。





注：“如果(<条件>,  $f(x)$ ,  $g(x)$ )”求导的结果是“如果(<条件>,  $f'(x)$ ,  $g'(x)$ )”。它不会考虑<条件>的限制而做出任何的调整。参见布尔值在条件语句中的应用。

### 脚本中的“如果”指令

“如果”指令可用于脚本，其通过判断限定条件从而执行不同的操作。

案例：设  $n$  是一个数值型对象， $A$  是一个点。指令“如果(取余( $n$ , 7)  $\neq$  0, 设置坐标( $A$ ,  $n$ , 0), 设置坐标( $A$ ,  $n$ , 1))”按指定条件调整点  $A$  的坐标。此例中使用指令“设置坐标( $A$ ,  $n$ , 如果(取余( $n$ , 7)  $\neq$  0, 0, 1))”将会更便捷。

注：在许多编程语言里，“如果”语句是一个命令，意味着“如果满足一个条件，就执行这样的操作，否则执行另一个操作”。但在 GeoGebra 里，“如果”语句的执行参数(<那么>或<否则>)不能是指令(命令)，只能是值，其中一个预设的执行参数(<那么>或<否则>)的值成为结果。因此，如果想要在  $a > 2$  的条件下将  $b$  的值设定为 2，正确的方法是“赋值( $b$ , 如果( $a > 2$ , 2,  $b$ ))”。“如果”指令的引用参数必须是一个对象或脚本指令，不能是定义(命令)。语法“ $b = \text{如果}(a > 1, 2, 3)$ ”是正确的，但“ $b = 2$ ”或“ $b = 3$ ”不能作为“条件指令”的参数。

### 2.13.3 IsInteger. 是否为整数

IsInteger(<Number>); 是否为整数(<数值>)。

根据数值是否为一个整数返回 true 或 false。

案例：“是否为整数(972/9)”返回“true”。



### 2.13.4 IsDefined (Defined) . 是否已定义

IsDefined(<Object>); 是否已定义(<对象>)。

根据对象是否已被定义返回 true 或 false。

案例：“是否已定义(圆周((1,1),-2))”返回“false”。

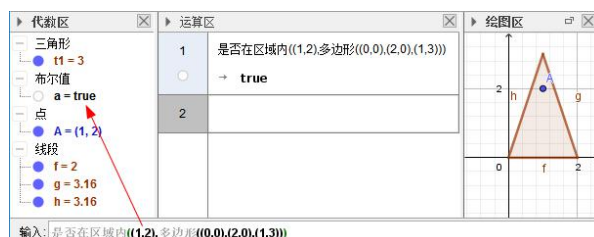


注：不是根据已有对象判断，是判断对象能否被定义。本例中，圆半径为负数，不能被定义。

### 2.13.5 IsInRegion. 是否在区域内

IsInRegion(<Object>); 是否在区域内(<点>, <区域>)。

若点在指定区域内则返回 true，否则返回 false。



案例：“是否在区域内((1, 2), 多边形((0, 0), (2, 0), (1, 3)))”返回“true”。

### 2.13.6 CountIf. 条件计数

**CountIf(<Condition>, <List>); 条件计数(<条件>, <列表>)。**

计算列表中满足条件的元素个数。

案例：“条件计数(x<3, {1, 2, 3, 4, 5})”给出数值 2；“条件计数(x<3, A1:A10)”，A1:A10 为表格中的单元格范围，计算所有值小于 3 的单元格数量。

注：除上述情况之外的列表只能使用的条件是“x==a”或“x!=a”即只能判断 x 是否等于某值。

**CountIf(<Condition>, <Variable>, <List>); 条件计数(<条件>, <变量>, <列表>)。**

这允许更灵活的语法，例如：对于点 P, Q, R，“条件计数(x(A)<3, A, {P, Q, R})”将计算只有 x-轴坐标值小于 3 的那些点，变量 A 重复运用于 P 然后 Q 然后 R 进行逐个检查。

案例：“条件计数(x(A)<3, A, {(0, 1), (4, 2), (2, 2)})”给出数值 2。



#### (1)条件计数(<条件>, <列表>)

计算列表中满足条件的元素个数。对于数集可使用任意形式的判断条件；对于其他对象类型的列表只能使用“x==条件”或“x!=条件”这样的判断条件。

释义：在条件计数中的列表可以是数集、其他列表、单元格区域等；如果是数集（单元格区域中的单元格都是数）条件可以是关于 x 的任意不等式关系；除上述情况之外的列表只能使用的条件时 x==a 或 x!=a 即只能判断 x 是否等于某值。

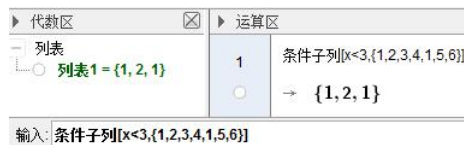
如：“条件计数(x<3, {1, 2, 3, 4, 5})”得到 2；“条件计数(x<3, A1:A10)”：A1:A10 为工作表中的单元格范围，计算所有值小于 3 的单元格数量；“条件计数(x==5, {1, 2, 3, 4, 7, 8, 6, (5, 5)})”得到 0；“条件计数(x==“人”, {1, 2, 3, 4, 7, 8, 6, “人”})”得到 1；“条件计数(x!=5, {1, 2, 3, 4, 7, 8, 6, (5, 5)})”得到 8。

### 2.13.7 KeepIf. 条件子列

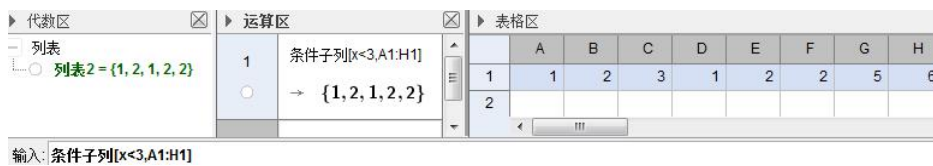
**KeepIf(<Condition>, <List>); 条件子列(<条件>, <列表>)。**

创建一个列表，包含列表中满足条件的元素。

案例：“条件子列(x<3, {1, 2, 3, 4, 1, 5, 6})”返回新列表。



案例：“条件子列(x<3, A1:H1)={1, 2, 1, 2, 2}”。

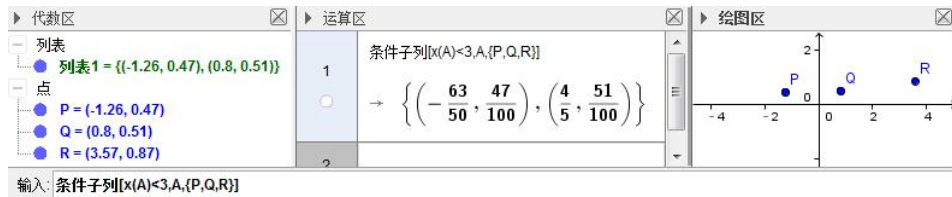


注：用于数集时可使用以 x 为变量的任意判断条件。用于其它对象类型的列表时，只能使用 x = 常量 (<对象>) 或 x ≠ 常量 (<对象>) 这两种条件判断的形式之一。

**KeepIf(<Condition>, <Variable>, <List>); 条件子列(<条件>, <变量>, <列表>)。**

此语法允许更灵活的条件判断形式。

**案例：**对于点 P、Q、R，“条件子列(x(A)<3,A,{P,Q,R})”将筛选列表中那些 x 轴坐标小于 3 的点。变量 A 依次检验 P、然后 Q、然后 R 是否满足判断条件。



“条件子列(x=="人",{1,2,3,4,7,8,6,"人"})”得出“{“人”}”；



“条件子列(x!=5,{1,2,3,4,7,8,6,(5,5)})”得出“{1,2,3,4,7,8,6,(5,5)}”。



## 2.14 Statistics. 统计

**编者注：**有一些新的指令在不断完善中，只有使用格式，没有汉化，本汇编也列示了。

### 2.14.1 Mad. Mad

**编者注：**就是“平均对偏差”指令，参见本章节相关介绍。

### 2.14.2 Mean. Mean

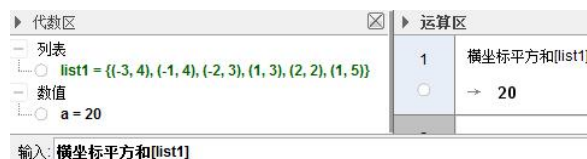
**编者注：**就是“平均数”指令，参见本章节相关介绍。

### 2.14.3 SigmaXX. 横坐标平方和

**SigmaXX(<List of Points>); 横坐标平方和(<点列>)。**

计算给定点的 x 坐标的平方和。

**案例：**设“list={(-3,4),(-1,4),(-2,3),(1,3),(2,2),(1,5)}”是点列。“横坐标平方和(List)”得出“a=20”。



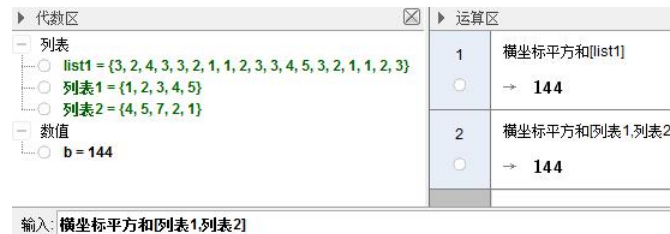
**SigmaXX(<List of Raw Data>); 横坐标平方和(<原始数据列表>)。**

计算给定数值的平方和。

**案例：**要算出列表的方差可用“横坐标平方和(list) / “长度(list)” - “平均数(list)”<sup>2</sup>。SigmaXX(<List of Numbers>, <List of Frequencies>); 横坐标平方和(<数据列表>, <频数列表>)

计算给定数值加权平方和。

**案例：**设“list1={3, 2, 4, 3, 3, 2, 1, 1, 2, 3, 3, 4, 5, 3, 2, 1, 1, 2, 3}”是一个数据列表。“互异(list1)”得出“列表1={1, 2, 3, 4, 5}”且“频数列表(list1)”得出“列表2={4, 5, 7, 2, 1}”。指令“横坐标平方和(列表1, 列表2)”得出“a=144”。与“横坐标平方和(list1)”相同结果。



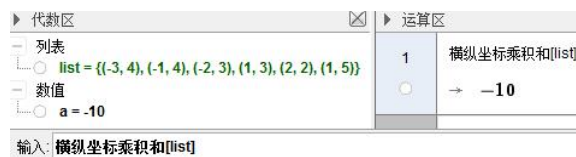
**编者注：**在早期版本中有中文指令，新版本暂时没有中文名。

## 2.14.4 SigmaXY. 横纵坐标乘积和

SigmaXY(<List of Points>); 横纵坐标乘积和(<点列>)。

计算 x 和 y 坐标的乘积和。

**案例：**“横纵坐标乘积和({(1, 2), (2, 3)})”得到  $1 \times 2 + 2 \times 3 = 8$ 。; 输入“list={(-3, 4), (-1, 4), (-2, 3), (1, 3), (2, 2), (1, 5)}”得到一个点列，“横纵坐标乘积和(list)”得出 a=-10。



**案例：**可以使用“横纵坐标乘积和(list) / “长度(list)” - “横坐标平均数(list)” \* “纵坐标平均数(list)”算出点列的方差。

SigmaXY(<List of x-coordinates>, <List of y-coordinates>); 横纵坐标乘积和(<x 坐标列表>, <y 坐标列表>)。

计算 x 和 y 坐标的乘积和。

**案例：**设有点“A=(-3, 4)”、“B=(-1, 4)”、“C=(-2, 3)”和“D=(1, 3)”。“{x(A), x(B), x(C), x(D)}”得出点的 x 坐标列表“列表1={-3, -1, -2, 1}”而“{y(A), y(B), y(C), y(D)}”得出点的 y 坐标列表“列表2={4, 4, 3, 3}”。指令“横纵坐标乘积和(列表1, 列表2)”得出“a=-19” ( $-3 \times 4 + -1 \times 4 + -2 \times 3 + 1 \times 3 = -19$ )。

**编者注：**在早期版本中有中文指令，但目前暂时没有中文名。

## 2.14.5 SigmaYY. 纵坐标平方和

SigmaYY(<List of Points>); 纵坐标平方和(<点列>)。

计算给定点 y 坐标的平方和。



**案例：**设“list={(-3, 4), (-1, 4), (-2, 3), (1, 3), (2, 2), (1, 5)}”是点列。“纵坐标平方和(list)”得出“a=79”。

**编者注：**在早期版本中有中文指令，但目前暂时没有中文名。

## 2. 14. 6 Spearman. 秩相关系数

**Spearman(<List of Points>); 秩相关系数(<点列>)。**

返回列表中点的 x 和 y 坐标秩相关系数。

**案例：**设“list={(-3, 4), (-1, 4), (-2, 3), (1, 3), (2, 2), (1, 5)}”是一个点列。“秩相关系数(list)”得出“a=-0.37”。



**Spearman(<List of Numbers>, <List of Numbers>); 秩相关系数(<数据列表 1>, <数据列表 2>)。**

返回两个列表的秩相关系数。

**案例：**设“list1={3, 2, 4, 5, 1, 6, 8, 9}”和“list2={5, 6, 8, 2, 1, 3, 4, 7}”是两个列表。“秩相关系数(list1, list2)”得出 a=0.24。

**释义：**秩相关系数又称等级相关系数，或顺序相关系数，是将两要素的样本值按数据的大小顺序排列位次，以各要素样本值的位次代替实际数据而求得的一种统计量。

## 2. 14. 7 Stdev. Sstdev

**编者注：**就是“样本标准差”指令，参见本章节相关介绍。

## 2. 14. 8 Stdevp. Stdevp

**编者注：**就是“标准差”指令，参见本章节相关介绍。

## 2. 14. 9 Sxx. Sxx

**Sxx(<List of Numbers>); Sxx(<数字列表>)。**

计算统计  $(\sum x^2 - (\sum x)^2 / n)$ 。

**案例：**“Sxx({1, 2, 3, 4, 5})” =  $1+4+9+16+25 - (1+2+3+4+5)^2 / 5 = 10$

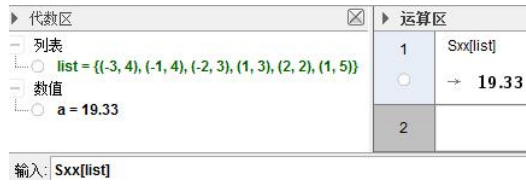


**Sxx(<List of Points>); Sxx(<点列>)。**

使用给定点的 x 坐标计算统计  $(\sum x^2 - (\sum x)^2 / n)$ 。

**案例：**“list={(-3, 4), (-1, 4), (-2, 3), (1, 3), (2, 2), (1, 5)}”，“Sxx(list)” = 19.33;

“ $S_{xx}(\{(1, 2), (2, 5), (3, 6), (4, 1), (5, 6)\})$ ”  $=1+4+9+16+25-(1+2+3+4+5)^2/5=10$



## 2. 14. 10 Sxy. Sxy

$S_{xy}(\langle \text{List of Points} \rangle)$ ;  $S_{xy}(\langle \text{点列} \rangle)$ 。

计算统计值  $(\sum (xy) - (\sum x)(\sum y)/n)$  ( $x$ 、 $y$  为每个点的纵横坐标值)。

案例：“ $S_{xy}(\{(1, 2), (2, 5), (3, 6), (4, 1), (5, 6)\})$ ”  $=2+10+18+4+30-(1+2+3+4+5)(2+5+6+1+6)/5=4$



$S_{xy}(\langle \text{List of Numbers} \rangle, \langle \text{List of Numbers} \rangle)$ ;  $S_{xy}(\langle \text{数据列表 1} \rangle, \langle \text{数据列表 2} \rangle)$ 。

计算  $x$  在第一个列表中  $y$  在第二个中的统计值  $\sum (xy) - (\sum x)(\sum y)/n$ 。

案例：“ $S_{xy}(\{0, 1, 2, 4, 5\}, \{1, 2, 3, 4, 5\})$ ”  $=0+2+6+16+25-(0+1+2+4+5)(1+2+3+4+5)/5=13$

## 2. 14. 11 Syy. Syy

$S_{yy}(\langle \text{List of Points} \rangle)$ ;  $S_{yy}(\langle \text{点列} \rangle)$ 。

使用列表中点的  $y$  坐标计算统计值  $(\sum y^2 - (\sum y)^2/n)$ 。

案例：“ $S_{yy}(\{(1, 2), (2, 5), (3, 6), (4, 1), (5, 6)\})$ ”  $=4+25+36+1+36-(2+5+6+1+6)^2/5=22$ 。



## 2. 14. 12 TTest. T 检验

$TTest(\langle \text{List of Sample Data} \rangle, \langle \text{Hypothesized Mean} \rangle, \langle \text{Tail} \rangle)$ ; T 检验 ( $\langle \text{样本数据列表} \rangle, \langle \text{假设平均数} \rangle, \langle \text{尾} \rangle$  “<”\_总体均值小于假设均值|“>”\_总体均值大于假设均值|“≠”\_总体均值不等于假设均值)。

执行使用给定样本数据列表的总体平均数单样本  $t$  检验。

假设平均数是总体平均数假定为无效假设。

尾部有如“<”, “>”, “≠”的可能值。这个指定如下供选假设。

“<”=总体均值小于假设均值

“>”=总体均值大于假设均值

“≠”=总体均值不等于假设均值

结果返回在是形如 {可能性,  $t$  检验统计数} 的列表中。

案例：“T 检验 ({1, 2, 3, 4, 5}, 3, “<”)” 得出 “{0.5, 0}”。



TTest(<Sample Mean>, <Sample Standard Deviation>, <Sample Size>, <Hypothesized Mean>, <Tail>); T 检验(<样本平均数>, <样本标准差>, <样本容量>, <假设平均数>, <尾“<”\_总体均值小于假设均值|“>”\_总体均值大于假设均值|“≠”\_总体均值不等于假设均值> )。

执行使用给定统计数据的总体平均数单样本 t 检验。

假设平均数和尾部定义如上。

结果返回在是形如 {可能性, t 检验统计数} 的列表中。

案例：“T 检验 (4, 1, 12, 4, “≠”) ” 得出 “{1, 0}”。



释义：t 检验分为单总体检验和双总体检验，双总体 t 检验是检验两个样本平均数与其各自所代表的总体的差异是否显著。双总体 t 检验又分为两种情况：独立样本 t 检验和配对样本 t 检验。

### 2. 14. 13 Percentile. 百分位数

Percentile(<List of Numbers>, <Percent>); 百分位数(<数据列表>, <百分数>)。

假设给定的百分数为 P。当列表为升序排列时，返回从数据列表中切除前 p 的数值。百分数必须在区间 “0<P≤1” 内。

案例：“百分位数 ({1, 2, 3, 4}, 0.25)” 得出 1.25。



注：四分位数和百分位数使用不同的计算规则且不总能返回一致的结果。

案例：“第一四分位数 ({1, 2, 3, 4})” 得出 1.5。“百分位数 ({1, 2, 3, 4}, 0.25)” 得出 1.25。



释义：百分位数：一组 n 个观测值按数值大小排列，处于 p%位置的值称第 p 百分位数。

### 2. 14. 14 SD. 标准差

SD(<List of Numbers>); 标准差(<原始数据列表>)。

计算列表中数值的标准差。

案例：“标准差 ({1, 2, 3, 4, 5})” 得出 1.41。“标准差 ({1, 2, 3})” 得出 0.82



SD(<List of Numbers>, <List of Frequencies>); 标准差(<数据列表>, <频数列表>)。

计算给定数值的加权标准差。

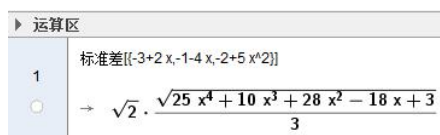
**案例：**“标准差({20, 40, 41, 42, 40, 54}, {20, 6, 4, 5, 2})” 得出 5.96

**CAS Syntax (运算区语法)**

**SD(<List of Numbers>); 标准差(<原始数据列表>)**。

计算列表中数值的标准差。

**案例：**“标准差({1, 2, 3, 4, 5})” 得出  $\sqrt{2}$ ；“标准差({-3+2 x, -1-4 x, -2+5 x^2})” 如同求值  $\sqrt{2} \cdot \frac{\sqrt{25x^4+10x^3+28x^2-18x+3}}{3}$ ”。



**注：**参见“平均数”指令。

**释义：**标准差 (StandardDeviation)，又常称均方差，但不同于均方误差，常使用作为统计分布程度上的测量。是指总体各单位原始值与其平均数离差平方的算术平均数的平方根。它反映组内个体间的离散程度，用  $\sigma$  表示。

## 2.14.15 ZProportionEstimate. 单比例 Z 估计

**ZProportionEstimate(<Sample Proportion>, <Sample Size>, <Level>); 单比例 Z 估计(<样本比例>, <样本容量>, <置信水平>)**。

使用给定的样本统计量和置信水平计算群体置信区间估计比例。结果列表的形式返回为{置信下限, 置信上限}。

## 2.14.16 ZProportionTest. 单比例 Z 检验

**ZProportionTest(<Sample Proportion>, <Sample Size>, <Hypothesized Proportion>, <Tail>); 单比例 Z 检验(<样本比例>, <样本容量>, <假设比例>, <尾 “<”\_总体比例小于假设比例|“>”\_总体比例大于假设比例|“≠”\_总体比例不等于假设比例>)**

使用给定的样本统计量做单比例 Z 检验。“假设比例”是总体估算比例。“尾”值可以是“<”、“>”和“≠”。这些指定的备择假设含义如下：

“<”：总体比例小于假设比例。

“>”：总体比例大于假设比例。

“≠”：总体比例不等于假设比例。

返回形式为{概率值, Z 检验统计量}的列表结果。

## 2.14.17 TMeanEstimate. 单均值 T 估计

**TMeanEstimate(<List of Sample Data>, <Level>); 单均值 T 估计(<样本数据列表>, <置信水平>)**。

根据给定的数据表和置信水平，计算总体均值 T 置信区间估计。结果返回在形如{下置信度限, 上置信度限}的列表中。

**TMeanEstimate(<Sample Mean>, <Sample Size>, <Level>); 单均值 T 估计(<样本平均数>, <样本容量>, <置信水平>)**

根据给定的数据数据和置信水平，计算总体均值 T 置信区间估计。结果返回在形如{下置信度限,



上置信度限)的列表中。

### 2.14.18 ZMeanEstimate. 单均值 Z 估计

**ZMeanEstimate**(**<List of Sample Data>**,**< $\sigma$ >**,**<Level>**); 单均值 Z 估计(**<样本数据列表>**,**<标准差>**,**<置信水平>**)。

根据给定的数据表、标准差和置信水平,计算总体均值 Z 置信区间估计。结果返回在形如{下置信度限, 上置信度限}的列表中。

**ZMeanEstimate**(**<Sample Mean>**,**< $\sigma$ >**,**<Sample Size>**,**<Level>**);单均值 Z 估计(**<样本平均数>**,**<标准差>**,**<样本容量>**,**<置信水平>**)

根据给定的数据数据、标准差和置信水平,计算总体均值 Z 置信区间估计。结果返回在形如{下置信度限, 上置信度限}的列表中。

### 2.14.19 ZMeanTest. 单均值 Z 检验

**ZMeanTest**(**<List of Sample Data>**,**< $\sigma$ >**,**<Hypothesized Mean>**,**<Tail>**); 单均值 Z 检验(**<样本数据列表>**,**<标准差>**,**<假设均值>**,**<尾" < " \_总体均值小于假设均值 | " > " \_总体均值大于假设均值 | " ≠ " \_总体均值不等于假设均值>**)。

使用给定的样本数据表做均值 Z 检验。“假设均值”是总体估算均值。“尾”值可能是“<”、“>”和“≠”。这些指定的备择假设含义如下:

“<”: 总体均值小于假设均值。

“>”: 总体均值大于假设均值。

“≠”: 总体均值不等于假设均值。

返回形式为{概率值, Z 检验统计量}的列表结果。

**ZMeanTest**(**<Sample Mean>**,**< $\sigma$ >**,**<Sample Size>**,**<Hypothesized Mean>**,**<Tail>**); 单均值 Z 检验(**<样本平均数>**,**<标准差>**,**<样本容量>**,**<假设均值>**,**<尾" < " \_总体均值小于假设均值 | " > " \_总体均值大于假设均值 | " ≠ " \_总体均值不等于假设均值>**)。

使用给定的样本数据统计量做均值 Z 检验。“假设均值”是总体估算均值。“尾”值可能是“<”、“>”和“≠”。这些指定的备择假设含义如下:

“<”: 总体均值小于假设均值。

“>”: 总体均值大于假设均值。

“≠”: 总体均值不等于假设均值。

返回形式为{概率值, Z 检验统计量}的列表结果。

### 2.14.20 Q3. 第三四分位数

**Q3**(**<List of Raw Data>**); 第三四分位数(**<原始数据列表>**)。

确定列表元素的第三个四分位数。

案例: “第三四分位数({1, 2, 3, 4})” 得出 3.5。



**Q3**(**<List of Numbers>**,**<List of Frequencies>**); 第三四分位数(**<数据列表>**,**<频数列表>**)。

确定关联频数列表元素的第三四分位数。

**案例：**“第三四分位数({1, 2, 3, 4}, {3, 2, 4, 2})” 得出 3。

**注：**GeoGebra 使用 Moore & McCabe (2002)方法计算四分位数, 见 <http://mathworld.wolfram.com/Quartile.html>。英文拼写为“Quartile3”也可以。

**释义：**第三个四分位数：第 75 百分位数又称第三个四分位数 (ThirdQuartile), 用“Q3”或“ThirdQuartile”均可代表本指令。

## 2.14.21 Q1. 第一四分位数

**Q1(<List of Raw Data>); 第一四分位数(<原始数据列表>)。**

确定列表元素的第一个四分位数。

**案例：**“第一四分位数({1, 2, 3, 4})” 得出 1.5。



**Q1(<List of Numbers>, <List of Frequencies>); 第一四分位数(<数据列表>, <频数列表>)。**

确定关联频数列表元素的第一四分位数。

**案例：**“第一四分位数({1, 2, 3, 4}, {3, 2, 4, 2})” 得出 1。

**注：**GeoGebra 使用 Moore & McCabe (2002)方法计算四分位数, 见 <http://mathworld.wolfram.com/Quartile.html>。英文拼写为“Quartile1”也可以。

**释义：**第一个四分位数：第 25 百分位数又称第一个四分位数 (FirstQuartile), 用“Q1”或“FirstQuartile”均可代表本指令。

## 2.14.22 HarmonicMean. 调和平均数

**HarmonicMean(<List of Numbers>); 调和平均数(<数据列表>)。**

返回给定数据列表的调和平均数。

**案例：**“调和平均数({13, 7, 26, 5, 19})” 得出 9.79。



**释义：**数学调和平均数定义为：数值倒数的平均数的倒数。

## 2.14.23 FitLog. 对数拟合

**FitLog(<List of Points>); 对数拟合(<点列>)。**

计算对数拟合曲线。

**案例：**“对数拟合({(e, 1), (e^2, 4)})” 得出 “-2+3 ln(x)”。



CAS Syntax (运算区语法)

FitLog(<List of Points>); 对数拟合(<点列>)。

计算对数拟合曲线。

案例: 对数拟合( $\{(e, 1), (e^2, 4)\}$ )得出“ $3 \ln(x)-2$ ”。

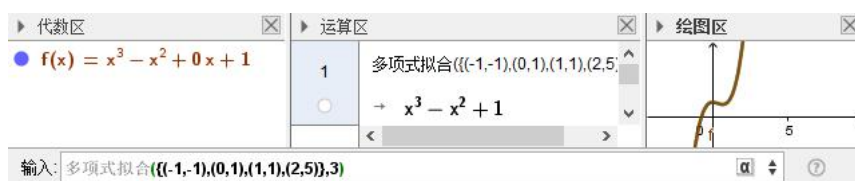
注: 欧拉数 $e$ 可以通过按“ALT+e”获得; 参见“指数拟合”指令、“多项式拟合”指令、“幂函数拟合”指令和“正弦拟合”指令。

## 2.14.24 FitPoly. 多项式拟合

FitPoly(<List of Points>, <Degree of Polynomial>); 多项式拟合(<点列>, <多项式次数>)。

计算 $n$ 次多项式的回归。

案例: “多项式拟合( $\{(-1, -1), (0, 1), (1, 1), (2, 5)\}, 3$ )”得出“ $f(x)=x^3-x^2+1$ ”。



CAS Syntax (运算区语法)

FitPoly(<List of Points>, <Degree of Polynomial>); 多项式拟合(<点列>, <多项式次数>)。

计算 $n$ 次多项式的回归。

案例: “多项式拟合( $\{(-1, -1), (0, 1), (1, 1), (2, 5)\}, 3$ )”得出“ $x^3-x^2+1$ ”。

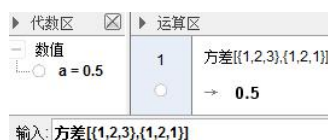
注: 列表中至少有多项式“次数 $n$ ”+1个点; 参见“指数拟合”指令、“对数拟合”指令、“幂函数拟合”指令和“正弦拟合”指令。

## 2.14.25 Variance. 方差

Variance(<List of Raw Data>); 方差(<原始数据列表>)。

计算列表元素的方差。

案例: “方差( $\{1, 2, 3\}$ )”得出 0.67。



Variance(<List of Numbers>, <List of Frequencies>); 方差(<数据列表>, <频数列表>)。

计算列表数据的加权方差。

案例: “方差( $\{1, 2, 3\}, \{1, 2, 1\}$ )”得出 0.5。

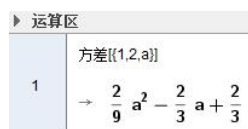
CAS Syntax (运算区语法)

Variance(<List of Numbers>); 方差(<原始数据列表>)。

计算列表元素的方差。如果列表中有未定义变量, 方差中会给出数学公式。

案例: “方差( $\{1, 2, a\}$ )”得出“ $\frac{2}{9}a^2 - \frac{2}{3}a + \frac{2}{3}$ ”; “方差( $\{1, 4, 5, 6, 7\}, \{1, 2, 4, 5, 6\}$ )”得

出 2.24。



**释义：**方差是各个数据与平均数之差的平方的和的平均数，用字母 D 表示。在概率论和数理统计中，用来度量随机变量和其数学期望（即均值）之间的偏离程度。

### 2. 14. 26 ANOVA. 方差分析

ANOVA(<List>, <List>, ...); 方差分析(<数据列表 1>, <数据列表 2>, ... )。

对指定的数列表执行单向方差分析。

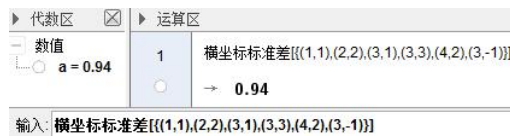
结果返回一个形如 {P 值, F 检验统计量} 的列表。

### 2. 14. 27 SDX. 横坐标标准差

SDX(<List of Points>); 横坐标标准差(<点列>)。

计算点列所有点的 x 坐标的标准差。

**案例：**“横坐标标准差({(1, 1), (2, 2), (3, 1), (3, 3), (4, 2), (3, -1)})” 得出 a=0.97。

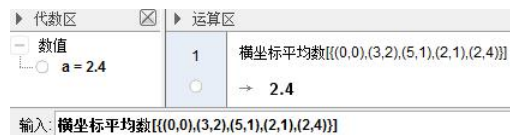


### 2. 14. 28 MeanX. 横坐标平均数

MeanX(<List of Points>); 横坐标平均数(<点列>)。

计算列表中点的 x 坐标的平均数。

**案例：**“横坐标平均数({(0, 0), (3, 2), (5, 1), (2, 1), (2, 4)})” 得出 2.4。



### 2. 14. 29 GeometricMean. 几何平均数

GeometricMean(List of Numbers); 几何平均数(<数据列表>)。

返回给定数据列表的几何平均数。

**案例：**“几何平均数(13, 7, 26, 5, 19)” 得出 11.76



### 2. 14. 30 RootMeanSquare. 均方根

RootMeanSquare(<List of Numbers>); 均方根(<数据列表>)。

返回给定数据列表的均方根。

**案例：**“均方根({3, 4, 5, 3, 2, 3, 4})” 得出 3.5456。



**释义：**均方根：将  $n$  个项的平方和除以  $n$  后开平方的结果，即均方根，也称方均根，英语简称 RMS。

## 2. 14. 31 ChiSquaredTest. 卡方检验

**ChiSquaredTest(<Matrix>); 卡方检验(<矩阵>)。**

执行一个观察计数矩阵对独立假设确定的预计数矩阵的卡方检验。预计数矩阵是内部计算。每个预计数使用以下规则从观察计数矩阵的行合计和列合计中获得。

预计数=行合计×列合计÷观察计数合计。

结果返回在形式为“{概率值, 卡方检验统计量}”的列表中。

**案例：**“卡方检验({{1, 2, 1}, {3, 2, 3}})”得出“{0.69, 0.75}”。



**ChiSquaredTest(<List>, <List>); 卡方检验(<列表 1>, <列表 2>)。**

执行给定观察计数列表对给定预计数列表的最佳回归检验。

结果返回在形式为“{概率值, 卡方检验统计量}”的列表中。

**案例：**“卡方检验({1, 2, 3, 4}, {3, 2, 4, 2})”得出“{0.31, 3.58}”。

**ChiSquaredTest(<Matrix>, <Matrix>); 卡方检验(<矩阵 1>, <矩阵 2>)。**

执行关于给定观察计数矩阵对给定预计数矩阵卡方检验。

结果返回在形式为“{概率值, 卡方检验统计量}”的列表中。

**案例：**“卡方检验({{1, 2, 1}, {3, 2, 3}}, {{2, 3, 2}, {4, 2, 3}})”得出“{0.45, 1.58}”。

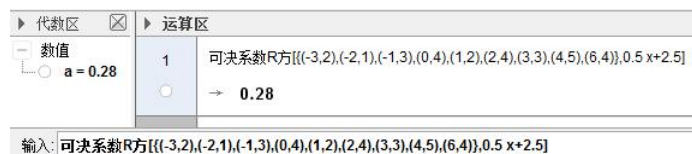


## 2. 14. 32 RSquare. 可决系数 R 方

**RSquare(<List of Points>, <Function>); 可决系数 R 方(<点列>, <函数>)。**

计算在列表中点的  $y$  值和列表中  $x$  对应的函数值间可决系数  $R^2 = 1 - SSE/S_{yy}$ 。

**案例：**“可决系数 R 方({(-3, 2), (-2, 1), (-1, 3), (0, 4), (1, 2), (2, 4), (3, 3), (4, 5), (6, 4)}, 0.5 x+2.5)”得出 0.28。



**释义：**可决系数：回归平方和（ESS）在总变差（TSS）中所占的比重称为可决系数，可决系数可以作为综合度量回归模型对样本观测值拟合优度的度量指标。可决系数越大，说明在总变差中由模型作出了解释的部分占的比重越大，模型拟合优度越好。反之可决系数小，说明模型对样本观测

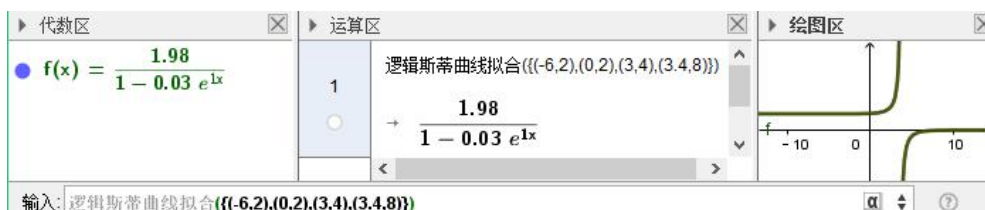
值的拟合程度越差。

### 2.14.33 FitLogistic. 逻辑斯蒂曲线拟合

FitLogistic(<List of Points>); 逻辑斯蒂曲线拟合(<点列>)。

计算形如  $a/(1+b e^{-kx})$  的拟合曲线。

案例：“逻辑斯蒂曲线拟合({(-6, 2), (0, 2), (3, 4), (3.4, 8)})” 得出  $\frac{1.98}{1-0.03e^{1x}}$ 。



CAS Syntax (运算区语法)

FitLogistic(<List of Points>); 逻辑斯蒂曲线拟合(<点列>)。

计算形如  $a/(1+b e^{-kx})$  的拟合曲线。

案例：逻辑斯蒂曲线拟合({(-6, 2), (0, 2), (3, 4), (3.4, 8)}) 得出  $\frac{1.98}{1-0.03e^{1x}}$ 。

注：首尾数据点相当接近曲线；点列至少要有 3 个点，最好更多；参见“拟合曲线”、“指数拟合”、“拟合直线 Y”、“拟合直线 X”、“对数拟合”、“多项式拟合”、“幂函数拟合”和“正弦拟合”指令。

释义：逻辑斯蒂曲线拟合(LogisticRegression)是回归分析的一种，是一个被 logistic 方程归一化后的线性回归。样本只有两个类别的问题经常被称为二项逻辑斯蒂曲线拟合，有多个类别(大于等于 3 个类别)的通常称为多项逻辑斯蒂曲线拟合或多分类逻辑斯蒂曲线拟合。宗旨是以线性分割面分割各类别，同时，对任意给定样本，保持该样本对各类别的事后概率之和为 1。

### 2.14.34 FitPow. 幂函数拟合

FitPow(<List of Points>); 幂函数拟合(<点列>)。

计算形如  $a x^b$  的拟合曲线。

案例：“幂函数拟合({(1, 1), (3, 2), (7, 4)})” 创建拟合曲线 “ $f(x)=0.97x^{0.71}$ ”。



CAS Syntax (运算区语法)

FitPow(<List of Points>); 幂函数拟合(<点列>)。

计算形如  $a x^b$  的拟合曲线。

案例：“幂函数拟合({(1, 1), (3, 2), (7, 4)})” 得出 “ $0.97x^{0.71}$ ”。

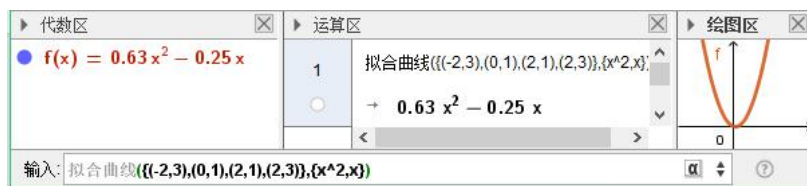
注：所有点需是坐标系第一象限点；参见“指数拟合”指令、“对数拟合”指令、“多项式拟合”指令和“正弦拟合”指令。

### 2.14.35 Fit. 拟合曲线

Fit(<List of Points>, <List of Functions>); 拟合曲线(<点列>, <函数列表>).

计算对点列的函数拟合曲线。

案例：“拟合曲线({(-2, 3), (0, 1), (2, 1), (2, 3)}, {x^2, x})” 得出 “f(x)=0.625 x^2-0.25x”。

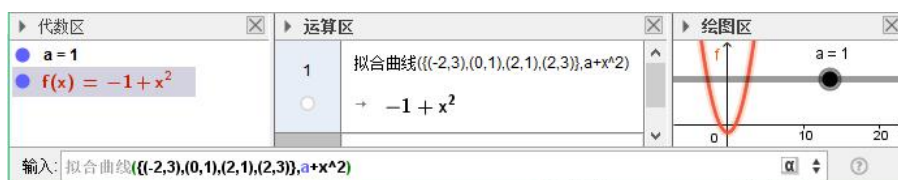


注：设  $L=\{A, B, C, \dots\}$ ,  $f(x)=1$ ,  $g(x)=x$ ,  $h(x)=e^x$ ,  $F=\{f, g, h\}$ 。拟合曲线(L, F) 计算格式如  $a+bx+c e^x$  的函数作为点列的拟合曲线。

Fit(<List of points>, <Function>); 拟合曲线(<点列>, <函数>).

计算点列的最小二次误差函数。函数必须依赖一个或多个滑动条参数，可以调整函数初始值使得函数更完善。非线性迭代可能不能汇聚，调整滑动条找到合适的初始值会有助于此。

案例：设 a 是区间在 -5 到 5 且增量为 1 滑动条参数。“拟合曲线({(-2, 3), (0, 1), (2, 1), (2, 3)}, a+x^2)” 得出 “-1+x^2”。



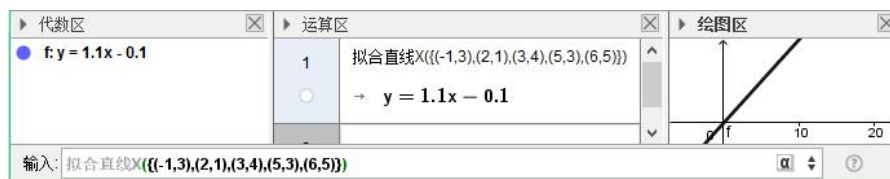
注：其他点回归指令是“指数拟合”、“生长曲线拟合”、“拟合直线 Y”、“拟合直线 X”、“对数拟合”、“逻辑斯蒂曲线拟合”、“多项式拟合”、“幂函数拟合”和“正弦拟合”。

### 2.14.36 FitLineX. 拟合直线 X

FitLineX(<List of Points>); 拟合直线 X(<点列>).

计算点 x 坐标关于 y 坐标的回归。

案例：“拟合直线 X({(-1, 3), (2, 1), (3, 4), (5, 3), (6, 5)})” 得出 “1.1x-0.1”。

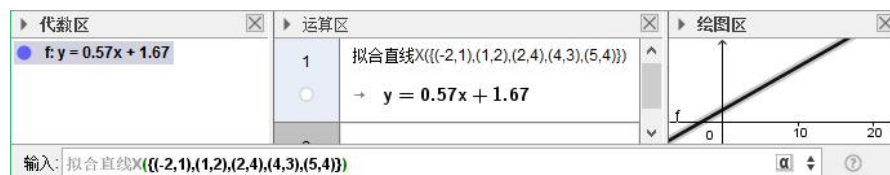


CAS Syntax (运算区语法)

FitLineX(<List of Points>); 拟合直线 X(<点列>).

计算点 x 坐标关于 y 坐标的回归。

案例：“拟合直线 X({(-2, 1), (1, 2), (2, 4), (4, 3), (5, 4)})” 得出 “0.57x+1.67”。



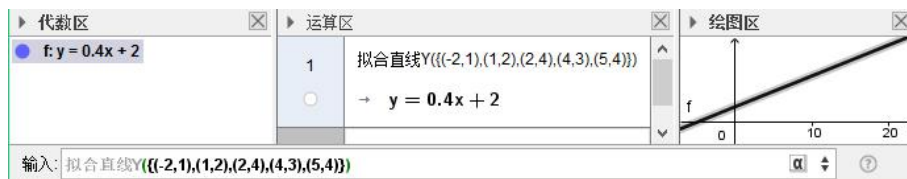
注：参见“最佳拟合直线”工具和“拟合直线 Y”指令。

### 2.14.37 FitLine (FitLineY) . 拟合直线 Y

FitLine(<List of Points>); 拟合直线 Y(<点列>)。

计算点  $y$  坐标关于  $x$  坐标的回归。

案例：“拟合直线 Y({(-2, 1), (1, 2), (2, 4), (4, 3), (5, 4)})” 得出 “ $0.4x+2$ ”。



CAS Syntax

FitLine(<List of Points>); 拟合直线 Y(<点列>)。

计算点  $y$  坐标关于  $x$  坐标的回归。

案例：“拟合直线 Y({(-2, 1), (1, 2), (2, 4), (4, 3), (5, 4)})” 得出 “ $0.4x+2$ ”。

注：参见“最佳拟合直线”工具和“拟合直线 X”指令。

## 2. 14. 38 TTestPaired. 配对 T 检验

TTestPaired(<List of Sample Data 1>, <List of Sample Data 2>, <Tail>); 配对 T 检验(<样本数据 1 列表>, <样本数据 2 列表>, <尾“<” $\mu$  小于 0 | “>” $\mu$  大于 0 | “ $\neq$ ” $\mu$  不等于 0 ( $\mu$  为总体的平均配对差异)>).

使用给定配对样本数据列表执行一个配对 T 检验。Tail 有“<”、“>”、“ $\neq$ ”可能值提供如下假设选项：

“<”： $\mu$  小于 0。

“>”： $\mu$  大于 0。

“ $\neq$ ”： $\mu$  不等于 0。

( $\mu$  为总体的平均配对差异)

结果返回在是形如 “{可能性, t 检验统计数}” 的列表中。

案例：“配对 T 检验 ({1, 2, 3, 4, 5}, {1, 1, 3, 5, 5}, “<”)” 得出 “{0.5, 0}”。



## 2. 14. 39 Mad. 平均绝对偏差

MAD(<List of Numbers>); 平均绝对偏差(<原始数据列表>)。

计算列表中数字与列表所有数值的平均值绝对偏差的平均值。

案例：“平均绝对偏差 ({1, 2, 3, 4, 5})” 得到 1.2。

编者注：就是测量列表中数值的离散度。

MAD( <List of Numbers>, <List of Frequencies>); 平均绝对偏差(<数字列表>, <频数列表>)。

计算列表中数字与列表所有数值的平均值绝对偏差的权重平均值。

案例：“平均绝对偏差 ({20, 40, 41, 42, 40, 54}, {20, 6, 4, 5, 2})” 得到 5.79。

注：参见“标准差”指令。

## 2. 14. 40 Mean. 平均数



**Mean(<List of Raw Data>); 平均数(<原始数据列表>)**。

计算给定列表元素的算术平均数。

**案例：**“平均数({1, 2, 3, 2, 4, 1, 3, 2})”得出“a=2.25”，而“平均数({1, 3, 5, 9, 13})”得出“a=6.2”。



**Mean(<List of Numbers>, <List of Frequencies>); 平均数(<数据列表>, <频数列表>)**。

计算给定列表元素的加权平均数。

**案例：**“平均数({1, 2, 3, 4}, {6, 1, 3, 6})”得出“a=2.56”而“平均数({1, 2, 3, 4}, {1, 1, 3, 6})”得出“a=3.27”。



**CAS Syntax (运算区语法)**

**Mean(<List of Numbers>); 平均数(<原始数据列表>)**。

计算给定列表元素的算术平均数。

**案例：**“平均数({1, 2, 3, 5, 44})”得出 a=11。

**注：**参见“横坐标平均数”、“纵坐标平均数”和“标准差”指令。

## 2.14.41 FitGrowth. 生长曲线拟合

**FitGrowth(<List of Points>); 生长曲线拟合(<点列>)**。

计算形如  $ab^x$  的点列函数。(如果不知道指数增长的含义, 见“指数拟合(<点列>)”。

**案例：**“生长曲线拟合({(0, 1), (2, 3), (4, 3), (6, 4)})”得出“ $1.31 \cdot 1.23^x$ ”。



**CAS Syntax (运算区语法)**

**FitGrowth(<List of Points>); 生长曲线拟合(<点列>)**。

计算形如  $ab^x$  的点列函数。

**案例：**“生长曲线拟合({(0, 1), (2, 3), (4, 3), (6, 4)})”得出“ $1.31 \cdot 1.23^x$ ”。

**注：**参见“拟合曲线”、“指数拟合”、“拟合直线 Y”、“拟合直线 X”、“对数拟合”、“逻辑斯蒂曲线拟合”、“多项式拟合”、“幂函数拟合”和“正弦拟合”指令。

## 2.14.42 ZProportion2Estimate. 双样本比例 Z 估计

**ZProportion2Estimate(<Sample Proportion 1>, <Sample Size 1>, <Sample Proportion 2>, <Sample Size 2>, <Level>); 双样本比例 Z 估计(<样本比例 1>, <样本容量 1>, <样本比例 2>, <样本容量 2>, <置信水平>)**。

计算给定样本数据集和置信水平的双样本差异比例 Z 置信区间估计。

结果返回在形如“{下置信度限, 上置信度限}”的列表中。

## 2.14.43 ZProportion2Test. 双样本比例 Z 检验

ZProportion2Test(<Sample Proportion 1>, <Sample Size 1>, <Sample Proportion 2>, <Sample Size 2>, <Tail>); 双样本比例 Z 检验 (<样本比例 1>, <样本容量 1>, <样本比例 2>, <样本容量 2>, <尾“<”\_总体比例之差小于 0|”>”\_总体比例之差大于 0|”≠”\_总体比例之差不等于 0)>。

执行使用给定样本数据列表的两个总体平均数差异 Z 检验。

尾部有如“<”, “>”, “≠”的可能值, 决定以下假设选项:

“<”: 总体比例之差小于 0。

“>”: 总体比例之差大于 0。

“≠”: 总体比例之差不等于 0。

结果返回在是形如 “{可能性, Z 检验统计数}” 的列表中。

## 2.14.44 TMean2Estimate. 双样本均值 T 估计

TMean2Estimate(<List of Sample Data 1>, <List of Sample Data 2>, <Level>, <Boolean Pooled>); 双样本均值 T 估计 (<样本数据 1 列表>, <样本数据 2 列表>, <置信水平>, <是否合并? true | false>)。

计算给定样本数据集和置信水平的双样本差异均值 t 置信区间估计。

如果是否合并=true, 计算中总体方差假定相等且样本标准差合并。

如果是否合并=false, 总体方差不假定相等且样本标准差不合并。

结果返回在形如 “{下置信度限, 上置信度限}” 的列表中。

TMean2Estimate(<Sample Mean 1>, <Sample Standard Deviation 1>, <Sample Size 1>, <Sample Mean 2>, <Sample Standard Deviation 2>, <Sample Size 2>, <Level>, <Boolean Pooled>); 双样本均值 T 估计 (<样本 1 平均数>, <样本 1 标准差>, <样本 1 容量>, <样本 2 平均数>, <样本 2 标准差>, <样本 2 容量>, <置信水平>, <是否合并? true | false>)。

计算给定样本数据集和置信水平的双样本差异均值 t 置信区间估计。

是否合并同上。结果返回在形如 “{下置信度限, 上置信度限}” 的列表中。

## 2.14.45 ZMean2Estimate. 双样本均值 Z 估计

ZMean2Estimate(<List of Sample Data 1>, <List of Sample Data 2>, < $\sigma_1$ >, < $\sigma_2$ >, <Level 1>); 双样本均值 Z 估计 (<样本数据 1 列表>, <样本数据 2 列表>, <标准差 1>, <标准差 2>, <置信水平>)。

计算给定样本数据集和置信水平的双样本差异均值 Z 置信区间估计。

结果返回在形如 “{下置信度限, 上置信度限}” 的列表中。



**案例:** 给定两个样本数据列表 “列表 1={1, 4, 5, 4, 1, 3, 4, 2}”, “列表 2={2, 1, 3, 1, 2, 5, 2, 4}”。列表 1 的标准差是 “ $\sigma_1=\sqrt{2}$ ”, 列表 2 的标准差为 “ $\sigma_2=\sqrt{1.75}$ ”, 且置信水平是 0.75。

“双样本均值 Z 估计 (列表 1, 列表 2,  $\sigma_1$ ,  $\sigma_2$ , 0.75)” 得出 “列表 3={-0.29, 1.29}”。

ZMean2Estimate(<Sample Mean 1>, < $\sigma_1$ >, <Sample Size 1>, <Sample Mean 2>, < $\sigma_2$ >, <Sampl

e Size 2>, <Level>); 双样本均值 Z 估计 (<样本 1 平均数>, <标准差 1>, <样本 1 容量>, <样本 2 平均数>, <标准差 2>, <样本 2 容量>, <置信水平>)。

计算给定样本数据集和置信水平的双样本差异均值 Z 置信区间估计。

结果返回在形如 “{下置信度限, 上置信度限}” 的列表中。

#### 2.14.46 ZMean2Test. 双样本均值 Z 检验

ZMean2Test(<List of Sample Data 1>, < $\sigma$  1>, <List of Sample Data 2>, < $\sigma$  2>, <Tail>); 双样本均值 Z 检验 (<样本数据 1 列表>, <标准差 1>, <样本数据 2 列表>, <标准差 2>, <尾 “<”\_总体均值之差小于 0|“>”\_总体均值之差大于 0|“≠”\_总体均值之差不等于 0> )。

执行使用给定样本数据列表的两个总体均值差异 Z 检验。

尾部有如“<”, “>”, “≠”的可能值, 决定以下假设选项:

“<”: 总体均值之差小于 0。

“>”: 总体均值之差大于 0。

“≠”: 总体均值之差不等于 0。

结果返回在是形如 “{概率, Z 检验统计数}” 的列表中。

ZMean2Test(<Sample Mean 1>, < $\sigma$  1>, <Sample Size 1>, <Sample Mean 2>, < $\sigma$  2>, <Sample Size 2>, <Tail>); 双样本均值 Z 检验 (<样本 1 平均数>, <标准差 1>, <样本 1 容量>, <样本 2 平均数>, <标准差 2>, <样本 2 容量>, <尾 “<”\_总体均值之差小于 0|“>”\_总体均值之差大于 0|“≠”\_总体均值之差不等于 0> )。

执行使用给定样本统计数据两个总体均值差异 Z 检验。“尾”定义同上。

#### 2.14.47 TTest2. 双总体 T 检验

TTest2(<List of Sample Data 1>, <List of Sample Data 2>, <Tail>, <Boolean Pooled>); 双总体 T 检验 (<样本数据 1 列表>, <样本数据 2 列表>, <尾 “<”\_总体均值之差小于 0|“>”\_总体均值之差大于 0|“≠”\_总体均值之差不等于 0>, <是否合并? true|false> )。

执行使用给定样本数据列表的两个总体平均数差异 t 检验。

尾部有如“<”, “>”, “≠”的可能值, 决定以下假设选项:

“<”: 总体均值之差小于 0。

“>”: 总体均值之差大于 0。

“≠”: 总体均值之差不等于 0。

如果是否合并=true, 计算中总体方差假定相等且样本标准差合并。

如果是否合并=false, 计算中总体方差不假定相等且样本标准差不合并。

结果返回在是形如 “{可能性, t 检验统计数}” 的列表中。

TTest2(<Sample Mean 1>, <Sample Standard Deviation 1>, <Sample Size 1>, <Sample Mean 2>, <Sample Standard Deviation 2>, <Sample Size 2>, <Tail>, <Boolean Pooled>); 双总体 T 检验 (<样本 1 平均数>, <样本 1 标准差>, <样本 1 容量>, <样本 2 平均数>, <样本 2 标准差>, <样本 2 容量>, <尾 “<”\_总体均值之差小于 0|“>”\_总体均值之差大于 0|“≠”\_总体均值之差不等于 0>, <是否合并? true|false> )。

执行使用给定统计数据两个总体平均数差异 t 检验。尾部和是否合并定义如上。结果返回在是形如 “{可能性, t 检验统计数}” 的列表中。

#### 2.14.48 Shuffle. 随机排列

**Shuffle(<List>); 随机排列(<列表>)。**

返回相同元素但顺序重新随机排列的新列表。

**案例：**“随机排列({3, 5, 1, 7, 3})”得到新列表“{5, 1, 3, 3, 7}”；“随机排列(序列(20))”给出随机顺序的前 20 个数。



**注：**可以在视图菜单中刷新或按 F9 键重新计算列表。参见“随机元素”和“区间随机数”指令。

## 2.14.49 SumSquaredErrors. 误差平方和

**SumSquaredErrors(<List of Points>, <Function>); 误差平方和(<点列>, <函数>)。**

计算在列表中的 y 坐标和列表中 x 变量的函数值之间的误差平方和“SSE”。

**案例：**如有一个点列“L={A, B, C, D, E}”和计算样表“f(x)=多项式拟合(L, 1)”和“g(x)=多项式拟合(L, 2)”，然后通过比较“sse\_f=误差平方和(L, f)”和“sse\_g=误差平方和(L, g)”，揣摩最小误差平方和，决定两个函数中哪个能提供最佳适应(回归)。g(x)更合适。



**释义：**误差平方和：在相同的条件下各次测定值  $x_i$  对测定平均值  $\bar{x}$  的偏差平方后再加和。

## 2.14.50 CorrelationCoefficient (PMCC) . 相关系数

**CorrelationCoefficient(<List of x-coordinates>, <List of y-coordinates>); 相关系数(<x 坐标列表>, <y 坐标列表>)。**

用给定的 x 坐标列表、y 坐标列表，计算出相关系数。

**案例：**“相关系数({1, 3, 2, 1, 5, 2}, {1, 6, 4, 3, 3, 2})”得出“a=0.36”。



**CorrelationCoefficient(<List of Points>); 相关系数(<点列>)。**

用给定点的坐标，计算给定点列的相关系数。

**案例：**“相关系数({(1, 1), (3, 6), (2, 4), (1, 3), (5, 3), (2, 2)})”得出“a=0.36”。

## 2.14.51 Covariance. 协方差

**Covariance(<List of Numbers>, <List of Numbers>); 协方差(<数据列表 1>, <数据列表 2>)**。  
计算指定列表中元素间的协方差。

**案例:** “协方差({1, 2, 3}, {1, 3, 7})” 得出 2。列表 {1, 2, 3} 和 {1, 3, 7} 间的协方差。

**Covariance(<List of Points>); 协方差(<点列>)**。

计算指定点 x 和 y 坐标间的协方差。



**案例:** “协方差({(1, 1), (2, 3), (3, 7)})” 得出 2。{1, 2, 3} 和 {1, 3, 7} 间的协方差。

**CAS Syntax (运算区语法)**

**Covariance(<List of Numbers>, <List of Numbers>); 协方差(<数据列表 1>, <数据列表 2>)**。  
计算指定列表中元素间的协方差。

**案例:** “协方差({1, 2, 3}, {1, 3, 7})” 得出 2。列表 {1, 2, 3} 和 {1, 3, 7} 间的协方差。

**Covariance(<List of Points>); 协方差(<点列>)**。

计算指定点 x 和 y 坐标间的协方差。

**案例:** “协方差({(1, 1), (2, 3), (3, 7)})” 得出 2。{1, 2, 3} 和 {1, 3, 7} 间的协方差。

**释义:** 在概率论和统计学中, 协方差用于衡量两个变量的总体误差。而方差是协方差的一种特殊情况, 即当两个变量是相同的情况。期望值分别为  $E(X)$  与  $E(Y)$  的两个实数随机变量,  $X$  与  $Y$  之间的协方差定义为:  $Cov(X, Y) = E((X - E(X))(Y - E(Y))) = E(XY) - E(X)E(Y)$ 。直观上来看, 协方差表示的是两个变量总体误差的期望。

## 2. 14. 52 Sample. 样本

**Sample(<List>, <Size>); 样本(<列表>, <容量>)**。

根据给出的样本构建一个包含 n 个随机元素的新列表, 源列表中的元素可以多次选取。

**案例:** “样本({1, 2, 3, 4, 5}, 5)” 得出样表 “列表={1, 2, 1, 5, 4}”。



**Sample(<List>, <Size>, <With Replacement>); 样本(<列表>, <容量>, <是否重复选择 true|false>)**。

随机选取列表中元素返回含有 n 个元素的列表。当且仅当最后一个参数是 true 时, 元素可以被多次选取。

**案例:** “样本({1, 2, 3, 4, 5}, 5, true)” 得出样表 “列表 1={2, 3, 3, 4, 5}”; “样本({1, 4, 5, 6, 7}, 10, false)” 得出 “未定义”。

**CAS Syntax (运算区语法)**

**Sample(<List>, <Size>); 样本(<列表>, <容量>)**。

返回从列表中随机选取的 n 个元素的列表, 元素可以多次选用。

**案例:** “样本({-5, 2, a, 7, c}, 3)” 得出样表 “{a, 7, -5}”。

**Sample(<List>, <Size>, <With Replacement>); 样本(<列表>, <容量>, <是否重复选择 true|false>)**。

随机选取列表中元素返回含有 n 个元素的列表。当且仅当最后一个参数是 true 时, 元素可以被多次选取。

**案例：**列表可以包含另一个列表，如：设“List1={1, 2, 3}”。“样本({List1, 4, 5, 6, 7, 8}, 3, false)”得出样表“{6, {1, 2, 3}, 4}”。



**释义：**研究中实际观测或调查的一部分个体称为样本(sample)，研究对象的全部称为总体。

## 2. 14. 53 SampleSD. 样本标准差

**SampleSD(<List of Numbers>);** 样本标准差(<原始数据列表>)。

返回给定数据列表的样本标准差。

**案例：**“样本标准差({1, 2, 3})”得出 1。

**SampleSD(<List of Numbers>, <List of Frequencies>);** 样本标准差(<数据列表>, <频数列表>)。

返回给定频数数据列表的样本标准差。

**案例：**“样本标准差({1, 2, 3, 4, 4}, {1, 1, 1, 2})”得出 1.08。



**CAS Syntax (运算区语法)**

**SampleSD(<List of Numbers>);**

返回给定数据列表的样本标准差。如果列表中含有未定义变量，在样本标准差中会给出一个公式。

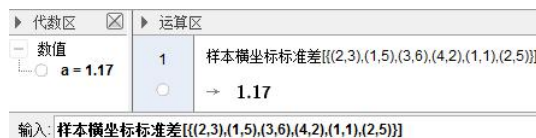
**案例：**“样本标准差({1, 2, a})”得出  $\frac{\sqrt{a^2 - 3a + 3}}{\sqrt{3}}$ 。

## 2. 14. 54 SampleSDX. 样本点横坐标标准差

**SampleSDX(<List of Points>);** 样本点横坐标标准差(<点列>)。

返回依据给定列表点的 x 坐标样本标准差。

**案例：**“样本点横坐标标准差({(2, 3), (1, 5), (3, 6), (4, 2), (1, 1), (2, 5)})”得出 “a=1.17”。



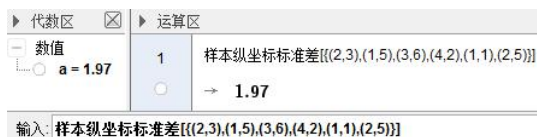
**注：**参见“样本点纵坐标标准差”指令。

## 2. 14. 55 SampleSDY. 样本点纵坐标标准差

**SampleSDY(<List of Points>);** 样本点纵坐标标准差(<点列>)。

返回依据给定列表点的 y 坐标样本标准差。

**案例：**“样本点纵坐标标准差({(2, 3), (1, 5), (3, 6), (4, 2), (1, 1), (2, 5)})”得出 “a=1.97”。



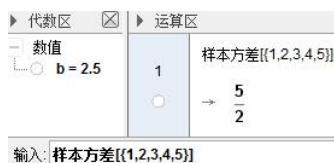
注：参见“样本点横坐标标准差”指令。

## 2.14.56 SampleVariance. 样本方差

SampleVariance(<List of Raw Data>); 样本方差(<原始数据列表>)。

返回给定数据列表的样本方差。

案例：“样本方差({1, 2, 3, 4, 5})”得出 a=2.5。



SampleVariance(<List of Numbers>, <List of Frequencies>); 样本方差(<数据列表>, <频数列表>)。

返回给定频数数据列表的样本方差。

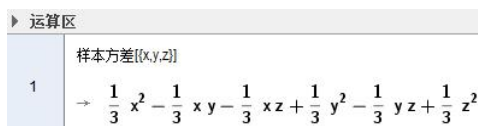
案例：“样本方差({1, 2, 3, 4, 5}, {3, 2, 4, 4, 1})”得出 a=1.29。

CAS Syntax (运算区语法)

SampleVariance(<List of Numbers>); 样本方差(<原始数据列表>)。

返回给定数据列表的样本方差。如果列表中含有未定义变量，在样本方差中会给出一个公式。

案例：“样本方差({x, y, z})”得出  $\frac{1}{3}x^2 - \frac{1}{3}xy - \frac{1}{3}xz - \frac{1}{3}y^2 - \frac{1}{3}yz - \frac{1}{3}z^2$ 。”

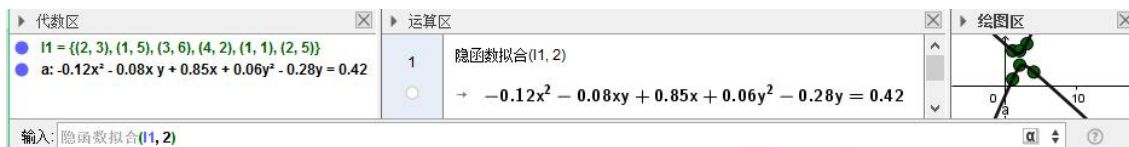


释义：样本方差：样本中各数据与样本平均数的差的平方和的平均数。

## 2.14.57 FitImplicit. 隐函数拟合

FitImplicit(<List of Points>, <Order>); 隐函数拟合(<点列>, <次数>)。

试图找到过这些点的 2 次以上最佳回归隐式曲线。至少须有  $\frac{n(n+3)}{2}$  个点。



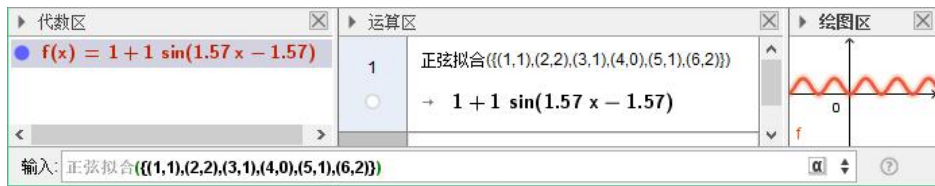
注：参见“拟合曲线”、“指数拟合”、“生长曲线拟合”、“拟合直线 X”、“对数拟合”、“逻辑斯蒂曲线拟合”、“多项式拟合”、“幂函数拟合”和“正弦拟合”指令。

## 2.14.58 FitSin. 正弦拟合

FitSin(<List of Points>); 正弦拟合(<点列>)。

计算形如  $a+b \sin(c x+d)$  的拟合曲线。

**案例：**“正弦拟合( $\{(1, 1), (2, 2), (3, 1), (4, 0), (5, 1), (6, 2)\}$ )”得出“ $f(x)=1+1 \sin(1.5708 x-1.5708)$ ”。



#### CAS Syntax (运算区语法)

**FitSin**(**<List of Points>**); **正弦拟合**(**<点列>**)。

计算形如  $a+b \sin(c x+d)$  的拟合曲线。

**案例：**正弦拟合( $\{(1, 1), (2, 2), (3, 1), (4, 0), (5, 1), (6, 2)\}$ )得出  $1+1 \sin(1.5708 x-1.5708)$ 。

**注：**列表最少需要 4 个点，最好更多；列表要包括至少 2 个极值点，开头的 2 个局部极值点不应与曲线上绝对极值点差别太大；参见“指数拟合”指令、“对数拟合”指令、“多项式拟合”指令和“幂函数拟合”指令。

### 2. 14. 59 FitExp. 指数拟合

**FitExp**(**<List of Points>**); **指数拟合**(**<点列>**)。

计算形如  $e^{bx}$  指数拟合曲线。

**案例：**“指数拟合( $\{(0, 1), (2, 4)\}$ )”得出“ $e^{0.69x}$ ”。



#### CAS Syntax (运算区语法)

**FitExp**(**<List of Points>**); **指数拟合**(**<点列>**)。

计算指数拟合曲线。

**案例：**指数拟合( $\{(0, 1), (2, 4)\}$ )得出  $e^{0.69x}$ 。

**注：**如果想要得到格式为  $a b^x$  的结果请用生长曲线拟合指令；欧拉数  $e$  可以通过按“ALT+e”获得；参见“拟合曲线”、“生长曲线拟合”、“拟合直线 Y”、“拟合直线 X”、“对数拟合”、“逻辑斯蒂曲线拟合”、“多项式拟合”、“幂函数拟合”和“正弦拟合”指令。

### 2. 14. 60 Median. 中位数

**Median**(**<List of Raw Data>**); **中位数**(**<原始数据列表>**)。

确定列表中元素的中位数。

**案例：**“中位数( $\{1, 2, 3\}$ )”得出 2；“中位数( $\{1, 1, 8, 8\}$ )”得出 4.5。



**Median**(**<List of Numbers>**,**<List of Frequencies>**); **中位数**(**<数据列表>**,**<频数列表>**)。

计算列表中元素的加权中位数。

**案例：**“中位数( $\{1, 2, 3\}, \{4, 1, 3\}$ )”得出 1.5；“中位数( $\{1, 2, 3\}, \{4, 1, 6\}$ )”得出 3。





### CAS Syntax (运算区语法)

**Median(<List of Numbers>); 中位数(<原始数据列表>)**。

确定列表中元素的中位数。

**案例:** “中位数({1, 2, 3})” 得出 2; “中位数({1, 1, 8, 8})” 得出 4.5。

**注:** 如果给出的列表长度是偶数, 会返回两个中心元素的算术平均值; 参见“平均数”指令。

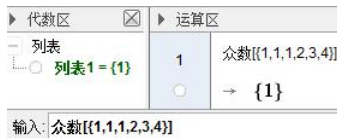
**释义:** 中位数 (Median): 将数据排序后, 位置在最中间的数值。即将数据分成两部分, 一部分大于该数值, 一部分小于该数值。中位数的位置: 当样本数为奇数时, 中位数=(N+1)/2; 当样本数为偶数时, 中位数为“N/2”与“(1+N)/2”的均值。

## 2. 14. 61 Mode. 众数

**Mode(<List of Numbers>); 众数(<数据列表>)**。

确定列表中元素的众数。

**案例:** “众数({1, 2, 3, 4})” 返回一个空列表 {}; “众数({1, 1, 1, 2, 3, 4})” 返回列表 {1}; “众数({1, 1, 2, 2, 3, 3, 4})” 返回列表 “{1, 2, 3}”。



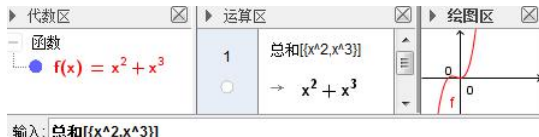
**释义:** 众数是一组数据中出现次数最多 (占比例最多) 的那个数值, 有时, 众数在一组数中有好几个。

## 2. 14. 62 Sum. 总和

**Sum(<List>); 总和(<列表>)**。

计算列表中所有元素的和。

**案例:** “总和({1, 2, 3})” 得出数值 “a=6”。“总和({x^2, x^3})” 得出 “f(x)=x^2+x^3”; “总和(序列(i, i, 1, 100))” 得出数值 “a=5050”; “总和({(1, 2), (2, 3)})” 得出点 “A=(3, 5)”; “总和({(1, 2), 3})” 得出点 “B=(4, 5)”; “总和({“a”, “b”, “c”})” 得出文本 “abc”。

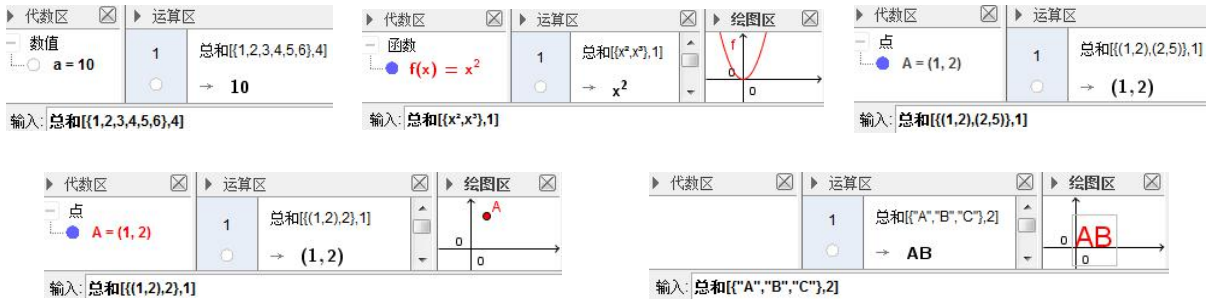


**Sum(<List>, <Number of Elements>); 总和(<列表>, <元素数量>)**。

计算列表中前 n 个元素的和。

**案例:** “总和({1, 2, 3, 4, 5, 6}, 4)” 得出数值 “a=10”; “总和({x^2, x^3}, 1)” 得出 “x^2” (输出一个函数); “总和({(1, 2), (2, 5)}, 1)” 得出 “(1, 2)” (输出一个点); “总和({(1, 2), 2}, 1)” 得

出“(1, 2)”(输出一个点); “总和({“A”, “B”, “C”}, 2)”得出“AB”(输出一个文本)。



**总和**( $\langle \text{List} \rangle, \langle \text{List of Frequencies} \rangle$ ); **总和**( $\langle \text{列表} \rangle, \langle \text{频数列表} \rangle$ )。

返回关联频数的列表数值的总和。

**案例:** “总和( $\{1, 2, 3, 4, 5\}, \{3, 2, 4, 4, 1\}$ )”得出“a=40”: 数集中 1, 2, 3, 4, 5 计算的次数分别为 3、2、4、4、1。



**注:** 这个指令适于数值、点、向量、文本和函数。

**CAS Syntax (运算区语法)**

**Sum**( $\langle \text{List} \rangle$ ); **总和**( $\langle \text{列表} \rangle$ )。

计算列表中所有元素的和。

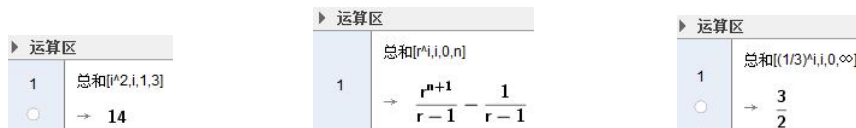
**案例:** “总和( $\{1, 2, 3\}$ )”得出 6; “总和( $\{a, b, c\}$ )”得出“a+b+c”。

**Sum**( $\langle \text{Expression} \rangle, \langle \text{Variable} \rangle, \langle \text{Start Value} \rangle, \langle \text{End Value} \rangle$ ); **总和**( $\langle \text{表达式} \rangle, \langle \text{变量} \rangle, \langle \text{初始值} \rangle, \langle \text{终止值} \rangle$ )。

计算给定表达式的变量从起始值到终止值的函数值的累加和。终止值可以是无穷大。

**案例:** “总和( $i^2, i, 1, 3$ )”得出 14, 即  $1^2 + 2^2 + 3^2 = 14$ ; “总和( $r^i, i, 0, n$ )”得出  $\frac{r^{n+1}}{r-1} - \frac{1}{r-1}$ ;

“总和( $(1/3)^i, i, 0, \infty$ )”得出  $\frac{3}{2}$ 。

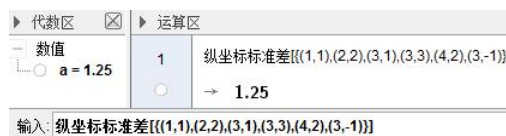


## 2. 14. 63 SDY. 纵坐标标准差

**SDY**( $\langle \text{List of Points} \rangle$ ); **纵坐标标准差**( $\langle \text{点列} \rangle$ )。

计算点列所有点的 y 坐标的标准差。

**案例:** “纵坐标标准差( $\{(1, 1), (2, 2), (3, 1), (3, 3), (4, 2), (3, -1)\}$ )”得出“a=1.25”。



## 2. 14. 64 MeanY. 纵坐标平均数

**MeanY(<List of Points>); 纵坐标平均数(<点列>)。**

计算列表中点的 y 坐标的平均数。

**案例：**“纵坐标平均数( $\{(0, 0), (3, 2), (5, 1), (2, 1), (2, 4)\}$ )” 得出 “ $a=1.6$ ”。



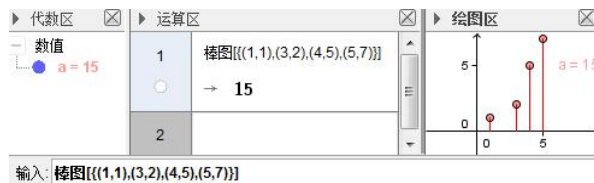
## 2.15 Chart. 图表

### 2.15.1 StickGraph. 棒图

**StickGraph(<List of Points>); 棒图(<点列>)。**

绘制给定点列的棒图。从 x 轴上点的 x 坐标值点做到点的垂线段。

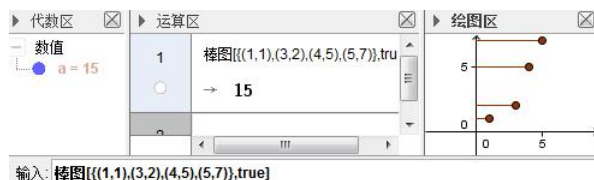
**案例：**“棒图( $\{(1, 1), (3, 2), (4, 5), (5, 7)\}$ )”。



**StickGraph(<List of Points>, <Boolean Horizontal>); 棒图(<点列>, <是否水平? true|false>)。**

绘制给定点的棒图。如果“是否水平”是 true，横线是自 y 轴上每个点纵坐标值点开始到点的线段。如果“是否水平”是 false，从 x 轴上点的 x 坐标值点做到点的垂线段。

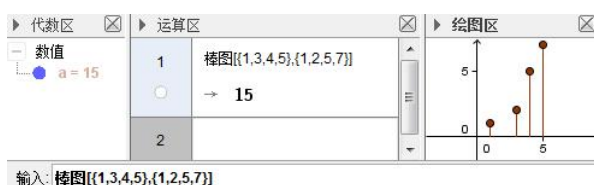
**案例：**“棒图( $\{(1, 1), (3, 2), (4, 5), (5, 7)\}, true)$ ”



**StickGraph(<List of x-coordinates>, <List of y-coordinates>); 棒图(<x 坐标列表>, <y 坐标列表>)。**

绘制两个列表为坐标的点的棒图。做每个自 x 轴坐标值点到给定点的垂线段。

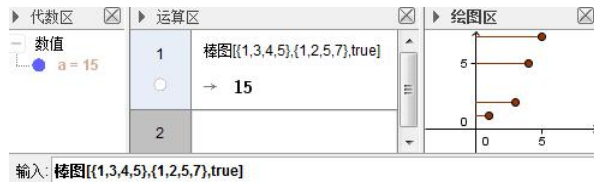
**案例：**“棒图( $\{1, 3, 4, 5\}, \{1, 2, 5, 7\}$ )”。



**StickGraph(<List of x-coordinates>, <List of y-coordinates>, <Boolean Horizontal>); 棒图(<x 坐标列表>, <y 坐标列表>, <是否水平? true|false>)。**

绘制两个列表为坐标的点的棒图。如果“是否水平”是 true，横线是自 y 轴上每个点纵坐标值点开始到点的线段。如果“是否水平”是 false，从 x 轴上点的 x 坐标值点做到点的垂线段。

**案例：**“棒图( $\{1, 3, 4, 5\}, \{1, 2, 5, 7\}, true)$ ”。



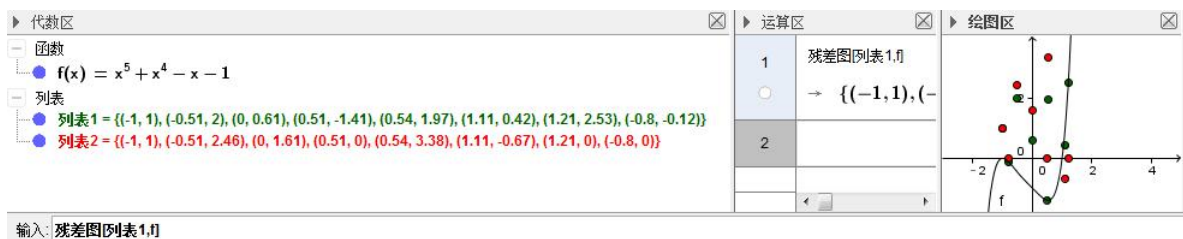
## 2.15.2 ResidualPlot. 残差图

**ResidualPlot(<List of Points>, <Function>); 残差图(<点列>, <函数>)**。

返回 x 坐标等于列表中元素的 x 坐标, y 坐标关于函数 f 残差的点列。

如果 L 中第 n 个元素是点 (a, b), 那么第 n 个元素的结果就是 (a, b-fa)。

**案例:** 设有点列 “列表 1 = {(-1, 1), (-0.51, 2), (0, 0.61), (0.51, -1.41), (0.54, 1.97), (1.11, 0.42), (1.21, 2.53), (-0.8, -0.12)}” 和函数 “ $f(x) = x^5 + x^4 - x - 1$ ”。指令 “残差图(列表 1, f)” 输出列表 “列表 2 = {(-1, 1), (-0.51, 2.46), (0, 1.61), (0.51, 0), (0.54, 3.38), (1.11, -0.66), (1.21, 0), (-0.8, 0)}” 且在绘图区绘出相应的点。

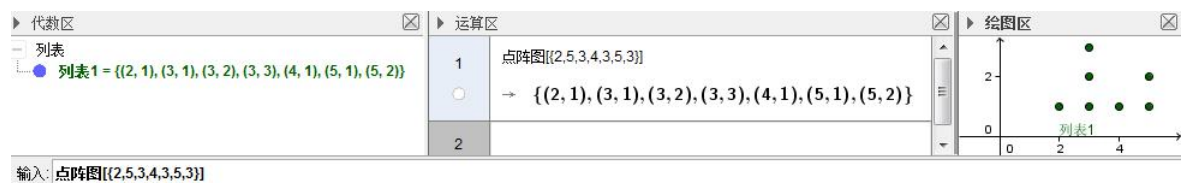


## 2.15.3 DotPlot. 点阵图

**DotPlot(<List of Raw Data>); 点阵图(<原始数据列表>)**。

根据给定的原始数据列表返回点阵图, 数据 n 为返回点的横坐标, 纵坐标为 n 在原始数据集中出现的频次 k, 即返回点列包括点 (n, 1), (n, 2), …, (n, k)。

**案例:** “点阵图({2, 5, 3, 4, 3, 5, 3})” 输出 “{(2, 1), (3, 1), (3, 2), (3, 3), (4, 1), (5, 1), (5, 2)}”。



**DotPlot(<List of Raw Data>, <Stack Adjacent Dots(optional)>, <Scale Factor(optional)>); 点阵图(<原始数据列表>, <堆栈相邻点(可选)>, <缩放因子(可选)>)**。

根据给定的原始数据列表返回点阵图, 数据 n 为返回点的横坐标, 纵坐标为 n 在原始数据集中出现的频次 k, 返回列表包括点 (n, 1), (n, 2), …, (n, k)。

如果选了一个缩放因子 s, 返回点列包括点 (n, 1s), (n, 2s), …, (n, ks)。

堆栈相邻点意味着一个逻辑值 (真或假): 如果选真, 点就堆栈 (点纵向互相靠近)。如果选假, 结果等同于 <堆栈相邻点(可选)>空。

指令点阵图同样适用于文本列表。

**案例:** “点阵图({“red”, “red”, “red”, “blue”, “blue”})” 输出 “{(1, 1), (1, 2), (2, 1), (2, 2), (2, 3)}”。



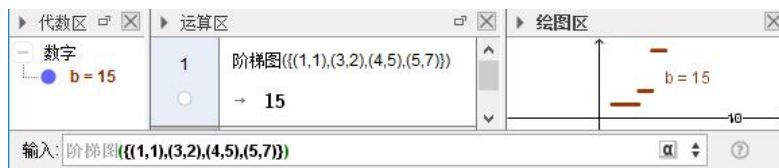
**注：**如果使用了文本列表点阵图指令会按照字母顺序排列(如 blue 出现 2 次，red 出现 3 次，B 的字母顺序在 R 前边，故得到 Blue(1, 1), (1, 2) 和 red(2, 1), (2, 2), (2, 3))。

## 2.15.4 StepGraph. 阶梯图

**StepGraph(<List of Points>); 阶梯图(<点列>)。**

绘制一个给定的点列的阶梯图。每个点用水平线段连接到列表中的下一个点。

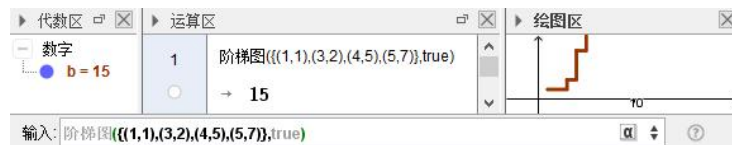
**案例：**“阶梯图({(1, 1), (3, 2), (4, 5), (5, 7)})”。



**StepGraph(<List of Points>, <Boolean Join>); 阶梯图(<点列>, <是否连接? true|false>)。**

绘制一个给定的点列的阶梯图。如果“连接”为 FALSE，水平线段绘制到下一点的 x 坐标，但垂直线段不绘制。如果“连接”为 true，每个点水平和垂直线段连接列表中的下一个点。

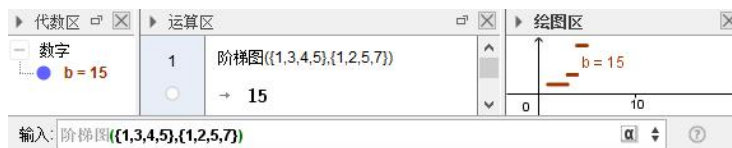
**案例：**“阶梯图({(1, 1), (3, 2), (4, 5), (5, 7)}, true)”。



**StepGraph(<List of x-coordinates>, <List of y-coordinates>); 阶梯图(<x 坐标列表>, <y 坐标列表>)。**

按照给定 x、y 坐标点列绘制阶梯图，每个点水平连接到下一个点。

**案例：**“阶梯图({1, 3, 4, 5}, {1, 2, 5, 7})”。



**StepGraph(<List of x-coordinates>, <List of y-coordinates>, <Boolean Join>); 阶梯图(<x 坐标列表>, <y 坐标列表>, <是否连接? true|false>)。**

按照给定 x、y 坐标点列绘制阶梯图。如果“连接”为 FALSE，水平线段绘制到下一点的 x 坐标，但垂直线段不绘制。如果“连接”为 true，每个点水平和垂直线段连接列表中的下一个点。

**案例：**“阶梯图({1, 3, 4, 5}, {1, 2, 5, 7}, true)”。



**StepGraph(<List of x-coordinates>, <List of y-coordinates>, <Boolean Join>, <Point Style>); 阶梯图(<点列>, <是否连接? true|false>, <点型 0\_不画点| 1\_实心点在右侧| 2\_实心点在右侧, 空心点在左侧| -1\_实心点在左侧| -2\_实心点在左侧, 空心点在右侧>)。**

如上描述绘制一个阶梯图。

点型可选“-2、-1、0、1、2”如下决定绘图中点的类型。

0: 不绘制点;

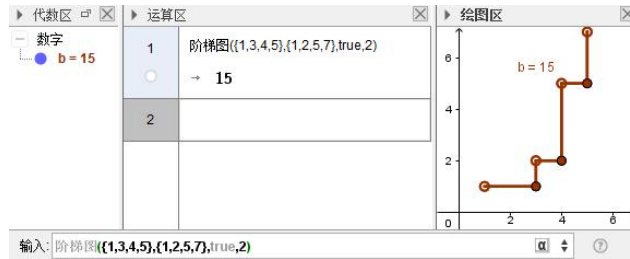
1: 右边实心点;

2: 右边实心点, 左边空心点;

-1: 左边实心点;

-2: 左边实心点, 右边空心点。

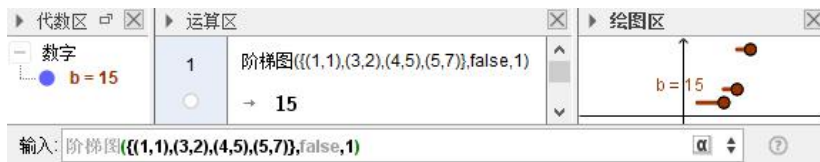
案例: “阶梯图({1, 3, 4, 5}, {1, 2, 5, 7}, true, 2)”。



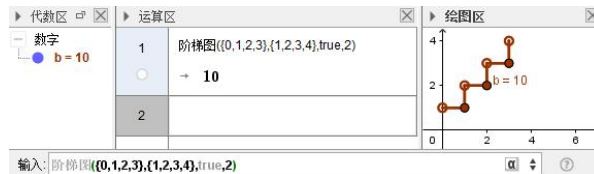
StepGraph(<List of Points>, <Boolean Join>, <Point Style>); 阶梯图(<x 坐标列表>, <y 坐标列表>, <是否连接? true|false>, <点型 0\_不画点 | 1\_实心点在右侧 | 2\_实心点在右侧, 空心点在左侧 | -1\_实心点在左侧 | -2\_实心点在左侧, 空心点在右侧>)

同上要求绘制阶梯图。

案例: “阶梯图({(1, 1), (3, 2), (4, 5), (5, 7)}, false, 1)”。



案例: “阶梯图({0, 1, 2, 3}, {1, 2, 3, 4}, true, 2)”, 见下图。



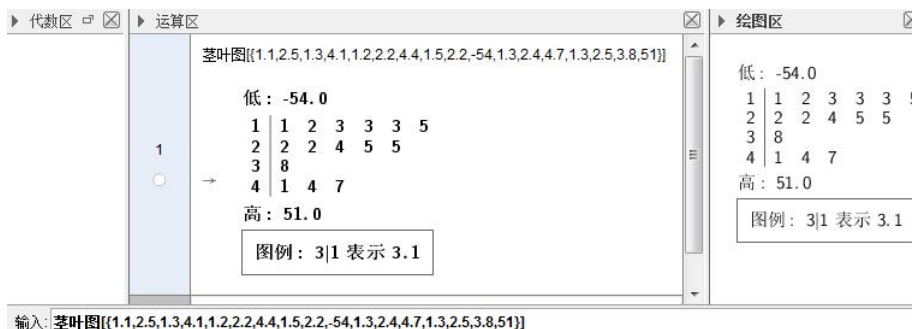
## 2. 15. 5 StemPlot. 茎叶图

这个指令在不同的英语变型中拼写不同: StemPlot (US)、StemAndLeaf (UK+Aus)。

StemPlot(<List>); 茎叶图(<列表>)。

返回给定数据列表的茎叶图。离群值从图中移开并分散列放。

离群值按照区间( $Q1-1.5(Q3-Q1)$ ,  $Q3+1.5(Q3-Q1)$ )外部值定义。



案例：“茎叶图({1.1, 2.5, 1.3, 4.1, 1.2, 2.2, 4.4, 1.5, 2.2, -54, 1.3, 2.4, 4.7, 1.3, 2.5, 3.8, 5.1})”。

StemPlot(<List>, <Adjustment-1|0|1>); 茎叶图(<列表>, <调节 -1\_默认茎单位除以 10|0\_没变化|1\_默认茎单位乘以 10> )。

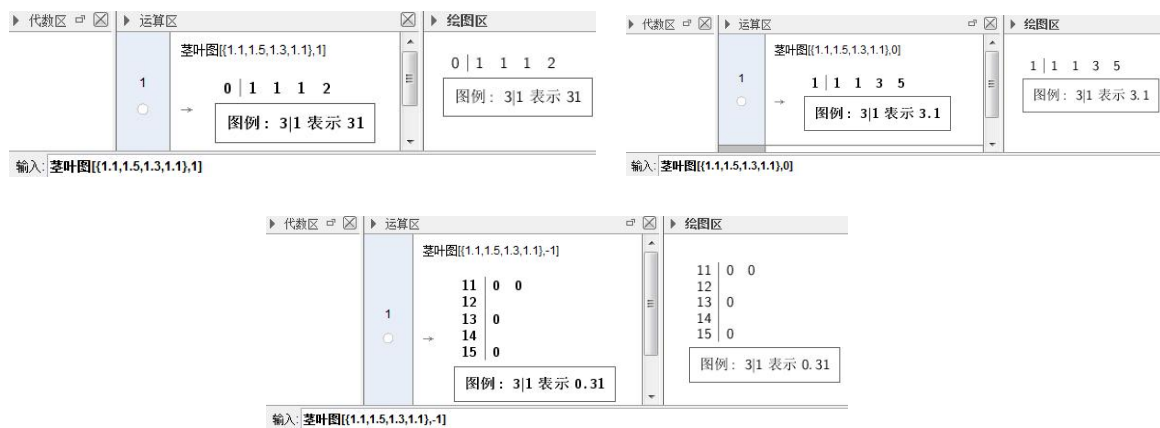
返回给定数据列表的茎叶图。

如果调节参数=-1 默认茎叶单位除以 10

如果调节参数=0 啥也不变

如果调节参数=1 默认茎叶单位乘以 10。

案例：“茎叶图({1.1, 1.5, 1.3, 1.1}, 1)”；“茎叶图({1.1, 1.5, 1.3, 1.1}, 0)”；“茎叶图({1.1, 1.5, 1.3, 1.1}, -1)”。



## 2.15.6 ContingencyTable. 列联表

ContingencyTable(<List of Text>, <List of Text>); 列联表(<文本列表 1>, <文本列表 2>).

用给定的列表绘制列联表。使用文本列表 1 中的唯一值作为表的行值，文本列表 2 中的唯一值作为列表中的列值。

案例：“列联表({"a", "b", "c"}, {"1", "2", "3"})”；



“列联表({"a", "b", "c"}, {"1", "2", "1"})”。



ContingencyTable(<List of Text>, <List of Text>, <Options>); 列联表(<文本列表 1>, <文本列表 2>, <选项 "|\_显示列百分比|"\_显示行百分比|"+\_显示总百分比|"\_显示预期计数|"\_k\_显示卡方贡献|"\_显示卡方检验结果>).

用给定的列表按照上述要求绘制列联表，“选项”控制了表中显示的计算结果。

注：“选项”的可能值有：“|”、“\_”、“+”、“e”、“k”和“=”。

- “|”：显示列百分比；
- “\_”：显示行百分比；
- “+”：显示总百分比；
- “e”：显示预期计数；
- “k”：显示卡方贡献；
- “=”：显示卡方检验结果。

多个选项可同时输入，如：“|k”，“|\_+”，“ek=”等。

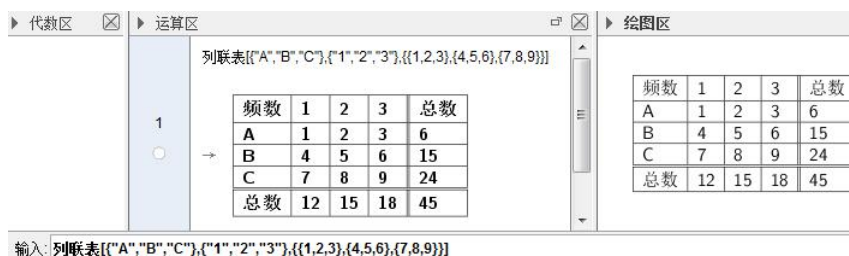
案例：“列联表({“a”,“b”,“c”},{“1”,“2”,“3”},“|”)”。



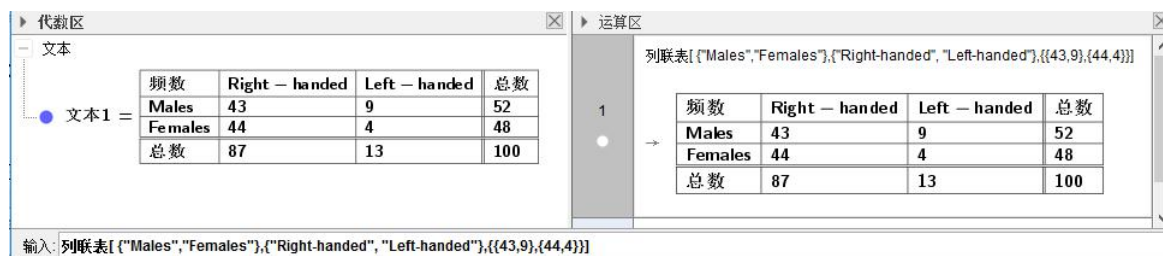
ContingencyTable(<List of Row Values>, <List of Column Values>, <Frequency Table>);  
列联表(<行数据列表>, <列数据列表>, <频数表>).

用给定的行、列和频数列表绘制列联表。

案例：“列联表({“A”,“B”,“C”},{“1”,“2”,“3”},{{1,2,3},{4,5,6},{7,8,9}})”。



案例：指令“列联表({“Males”,“Females”},{“Right-handed”,“Left-handed”},{{43,9},{44,4}})”得出下图列联表。



ContingencyTable(<List of Row Values>, <List of Column Values>, <Frequency Table>, <Options>); 列联表(<行数据列表>, <列数据列表>, <频数表>, <选项 “|”\_显示列百分比|“\_”\_显示行百分比|“+”\_显示总百分比|“e”\_显示预期计数|“k”\_显示卡方贡献|“=”\_显示卡方检验结果>).

用给定的行、列和频数列表绘制列联表。“选项”控制了表中显示的计算结果同上所述。

案例：指令“列联表({“Males”,“Females”},{“Right-handed”,“Left-handed”},{{43,9},{44,4}},”\_”)”



得出结果如下：

输入: 列联表[{"Males", "Females"}, {"Right-handed", "Left-handed"}, {(43,9), (44,4)}, "\_"]

案例：“列联表({“A”, “B”, “C”}, {"1”, “2”, “3”}, {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}, “|ke”)”。

输入: 列联表[{"A", "B", "C"}, {"1", "2", "3"}, {{1,2,3}, {4,5,6}, {7,8,9}}, "|ke"]

## 2. 15.7 FrequencyTable. 频数表

FrequencyTable(<List of Raw Data>); 频数表(<原始数据列表>, <缩放因子(可选)>).

返回一个首列包括原始数据列表 L 唯一元素排序且第二列包括首列值在列表 L 中出现次数的表单(文本格式)。列表 L 可以是数值的或文本的。

案例：“频数表({2, 1, 3, 3, 2, 1, 1, 4, 5, 5})”。

输入: 频数表[{2,1,3,3,2,1,1,4,5,5}]

输入: 频数表[true, {2,1,3,3,2,1,1,4,5,5}]

FrequencyTable(<Boolean Cumulative>, <List of Raw Data>); 频数表(<是否累积? true|false>, <原始数据列表>).

如果累积是“false”，返回与频数表(<原始数据列表>)相同的结果。

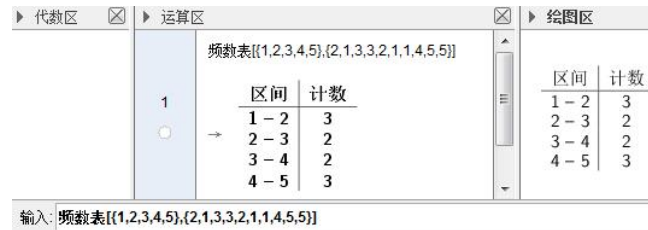
如果累积是“true”，返回的表单首列与与频数表(<原始数据列表>)相同而第二列包含首列值在列表 L 中累积出现次数。

案例：“频数表(true, {2, 1, 3, 3, 2, 1, 1, 4, 5, 5})”。

FrequencyTable(<List of Class Boundaries>, <List of Raw Data>); 频数表(<组界列表>, <原始数据列表>).

返回一个首列包含组界而第二列包含原始数据列表中的归类于首列边界的数值出现次数的表单（文本格式）除最高分类外区间都是(a, b)格式，最高分类区间是(a, b)格式。

案例：“频数表({1, 2, 3, 4, 5}, {2, 1, 3, 3, 2, 1, 1, 4, 5, 5})”。

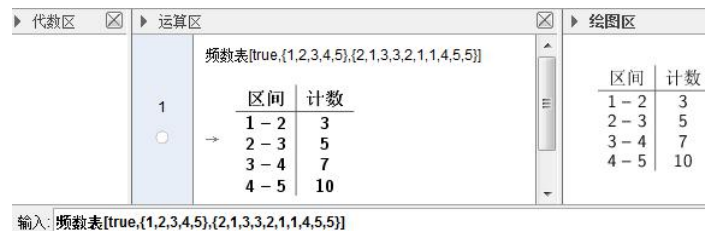


释义：原始数据被组界列表非为 n-1 个小列表，其中 n 等于组界列表的元素数，上面例子中组界列表就能够分出四个区间，(1, 2)，(2, 3)，(3, 4)，(4, 5)。注意四个列表除最后一个之外，其他列表都是包含最小值，不包含最大值。所以上例中四个区间各有 3, 2, 2, 3 个元素。

FrequencyTable(<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>); 频数表(<是否累积? true|false>, <组界列表>, <原始数据列表>).

如果累积是“false”，返回与频数表(<组界列表>, <原始数据列表>)相同的表单。如果累积是“true”，返回的表单首列与与频数表(<原始数据列表>)相同而第二列包含首列值在列表 L 中累积出现次数。

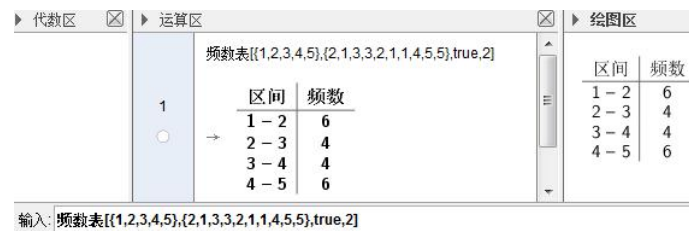
案例：“频数表(true, {1, 2, 3, 4, 5}, {2, 1, 3, 3, 2, 1, 1, 4, 5, 5})”。



FrequencyTable(<List of Class Boundaries>, <List of Raw Data>, <Use Density>, <Density Scale Factor (optional)>); 频数表(<组界列表>, <原始数据列表>, <应用密度>, <密度缩放因子(可选)>).

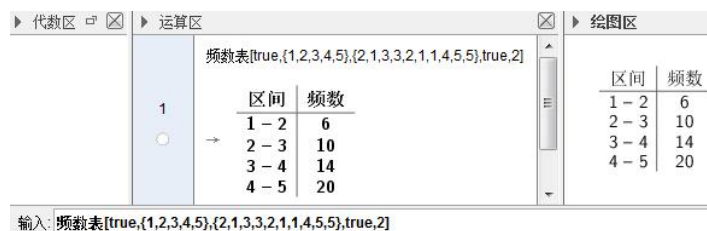
返回一个首列包含组界而第二列包含相应直方图指令频数的表单（文本格式）。

案例：“频数表({1, 2, 3, 4, 5}, {2, 1, 3, 3, 2, 1, 1, 4, 5, 5}, true, 2)”。



FrequencyTable(<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>, <Use Density>, <Density Scale Factor (optional)>); 频数表(<是否累积? true|false>, <组界列表>, <原始数据列表>, <应用密度>, <密度缩放因子(可选)>).

返回一个首列包含组界而第二列包含相应直方图指令频数的表单（文本格式）。



案例：“频数表(true, {1, 2, 3, 4, 5}, {2, 1, 3, 3, 2, 1, 1, 4, 5, 5}, true, 2)”。

FrequencyTable(<List of Raw Data>, <Scale Factor (optional)>); 频数表(<原始数据列表>, <缩放因子(可选)>)

返回一个首列包括<原始数据列表>唯一元素排序且第二列频数包括首列值出现次数乘以<缩放因子>的表单(文本格式)。列表可以是数值的或文本的。

案例：“频数表({1, 1, 1, 2, 2, 3, 3, 4, 5}, 2)” 返回一个首列值是 1、2、3、4、5 而第二列频数 6、4、4、2、2。



注：在列表中 1 出现了三次，所以 1 事件数 (=3) 乘以缩放因子 2 得到了第二列中的 6。

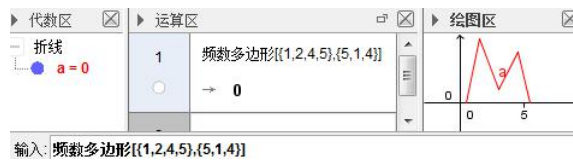
注：这个指令与“频数列表”和“直方图”指令相似。参见这些指令章节的相关案例。

## 2. 15.8 FrequencyPolygon. 频数多边形

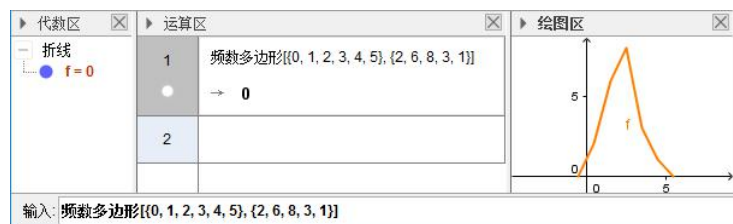
FrequencyPolygon(<List of Class Boundaries>, <List of Heights>); 频数多边形(<组界列表>, <高度列表>)

创建顶点为给定高度的频数多边形。组界列表确定每个顶点的 x 坐标。

案例：“频数多边形({1, 2, 4, 5}, {5, 1, 4})”。



案例：指令“频数多边形({0,1,2,3,4,5},{2,6,8,3,1})” 绘制出相应的线图。

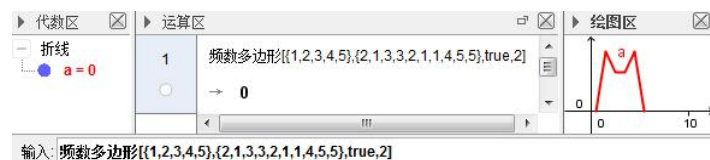


FrequencyPolygon(<List of Class Boundaries>, <List of Raw Data>, <Boolean Use Density>, <Density Scale Factor(optional)>); 频数多边形(<组界列表>, <原始数据列表>, <是否应用密度? True|false>, <密度缩放因子(可选)>)

使用原始数据创建一个频数多边形。组界确定顶点 x 坐标值和每个类有多少个数据元素。顶点的 y 坐标由以下决定。

如果使用密度为 true, 高度=(密度缩放因子)\*(类频)/(类宽); 如果使用密度为 false, 高度=类频; 默认情况下, 使用密度为 true 而密度缩放因子为 1。

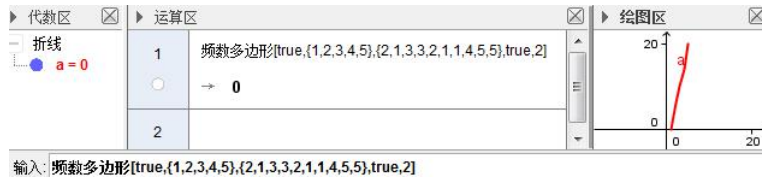
案例：“频数多边形({1, 2, 3, 4, 5}, {2, 1, 3, 3, 2, 1, 1, 4, 5, 5}, true, 2)”。



FrequencyPolygon(<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data a>, <Boolean Use Density>, <Density Scale Factor(optional)>); 频数多边形(<是否累积? True|false>, <组界列表>, <原始数据列表>, <是否应用密度? True|false>, <密度缩放因子(可选)>).

如果累积是 true 将建立每个顶点 y 坐标等于已有频数和累加的一个频数多边形。

案例: “频数多边形(true, {1, 2, 3, 4, 5}, {2, 1, 3, 3, 2, 1, 1, 4, 5, 5}, true, 2)”。



释义: 频数多边形(frequencypolygon)是用组织图来造成的同组织图一样是用来表达这些数据的整体规律或趋势, 从而分析这些数据的分布情况。在直方图的基础上, 把直方图顶部的中点(即组中值)用直线连接起来, 再把原来的直方图抹掉就是频数多边形图。

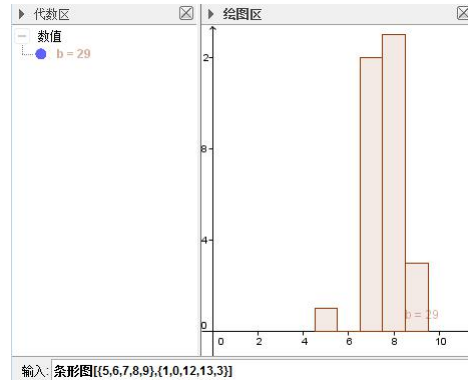
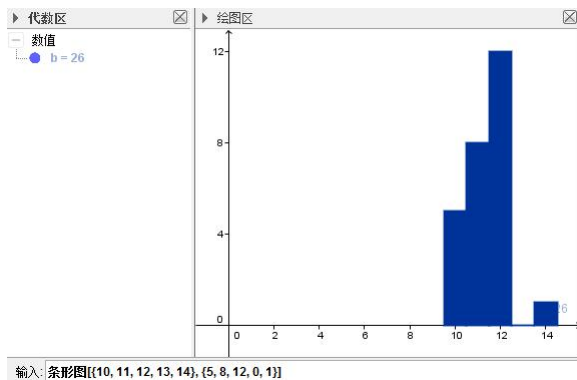
## 2. 15.9 BarChart. 条形图

BarChart(<List of Data>, <List of Frequencies>); 条形图(<数据列表>, <频数列表>).

使用列表中的数据对应的频数创建一个条形图。

注: 原始数据列表中的数值需要递增排列。

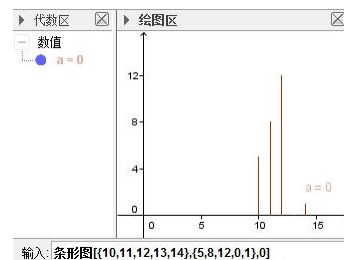
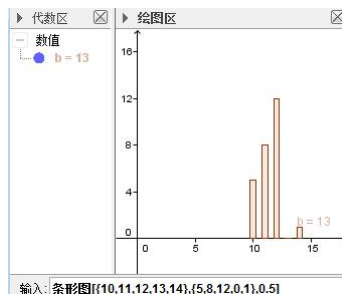
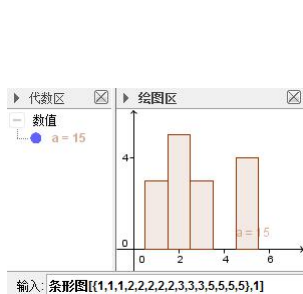
案例: “条形图({10, 11, 12, 13, 14}, {5, 8, 12, 0, 1})”; “条形图({5, 6, 7, 8, 9}, {1, 0, 12, 13, 3})”; “条形图({0.3, 0.4, 0.5, 0.6}, {12, 33, 13, 4})”。



BarChart(<List of Raw Data>, <Width of Bars>); 条形图(<原始数据列表>, <条形宽度>)

创建给定条宽度的原始数据条形图。

案例: “条形图({1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 5, 5, 5}, 1)”。



BarChart(<List of Data>, <List of Frequencies>, <Width of Bars>); 条形图(<数据列表>, <频数列表>, <条形宽度>).

使用数据列表和对应的频数创建一个条有给定宽度的条形图。

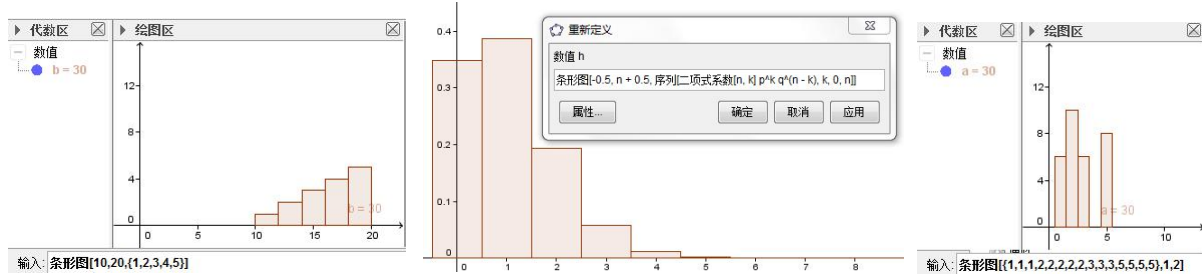
案例: “条形图({10, 11, 12, 13, 14}, {5, 8, 12, 0, 1}, 0.5)” 条间留有缝隙; “条形图({10, 11, 1

2, 13, 14}, {5, 8, 12, 0, 1}, 0)” 输出线图。

**BarChart(<Start Value>, <End Value>, <List of Heights>); 条形图(<起始值>, <终止值>, <高度列表>)**。

创建一个符合给定区间的条形图：数值条的数量取决于列表的长度，列表元素是条的高度。

**案例：**“条形图(10, 20, {1, 2, 3, 4, 5})” 给出一个区间(10, 20)内指定高度的 5 个条的条形图。



**BarChart(<Start Value>, <End Value>, <Expression>, <Variable>, <From Number>, <To Number>); 条形图(<起始值>, <终止值>, <表达式>, <变量>, <从数值 1>, <到数值 2>)**。

创建一个符合给定区间(起始值, 终止值)的条形图，使用表达式变量从数值 1(c) 到数值 2(d) 间的函数值(k) 作为条形的高度。

**案例：**假设  $p=0.1, q=0.9, n=10$ ，然后“条形图(-0.5, n+0.5, 二项式系数(n, k)\* $p^k q^{n-k}, k, 0, n$ )”，输出一个区间(-0.5, n+0.5)内的条形图。条形的高度从属于给定表达式的计算可能性。

**BarChart(<Start Value>, <End Value>, <Expression>, <Variable>, <From Number>, <To Number>, <Step Width>); 条形图(<起始值>, <终止值>, <表达式>, <变量>, <从数值 1>, <到数值 2>, <步长>)**。

创建一个符合给定区间(起始值, 终止值)的条形图，使用表达式变量从数值 1(c) 到数值 2(d) 间步长为 s 的函数值(k) 作为条形的高度。

**BarChart(<List of Raw Data>, <Width of Bars>, <Vertical Scale Factor(optional)>); 条形图(<原始数据列表>, <条形宽度>, <竖直缩放因子(可选)>)**。

创建给定原始数据的一个条使用给定宽度且高度取决于竖直缩放因子的条形图。

**案例：**“条形图({1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 5, 5, 5, 5}, 1, 2)”。

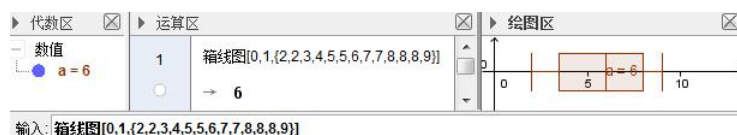
**注：**在对象属性中可以指定每个条不同的颜色/填充。

### 2. 15. 10 BoxPlot. 箱线图

**BoxPlot(yOffset, yScale, List of Raw Data); 箱线图(y 轴方向偏移量, y 轴方向范围, 原始数据列表)。**

使用给定原始数据创建一个在坐标系中垂直方位取决于 y 轴方向偏移量，高度受 y 轴方向范围影响的箱线图。

**案例：**“箱线图(0, 1, {2, 2, 3, 4, 5, 5, 6, 7, 7, 8, 8, 8, 9})”

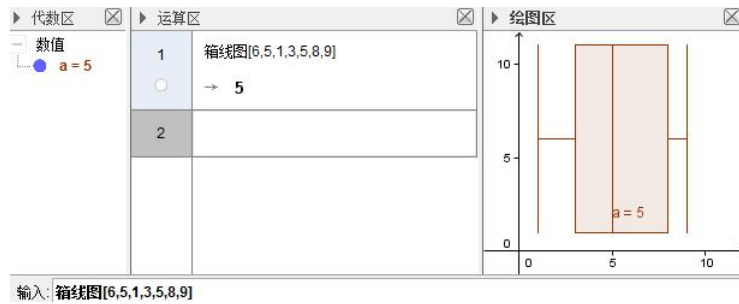


**注：**<y 轴方向偏移量>：箱线图的纵向中心位置；<y 轴方向范围>：箱线图的宽度的一半。

**BoxPlot(yOffset, yScale, Start Value, Q1, Median, Q3, End Value); 箱线图(y 轴方向偏移量, y 轴方向范围, 初始值, 第一四分位数, 中位数, 第三四分位数, 终止值)。**

创建给定统计数据在区间(初始值, 终止值)的箱线图。

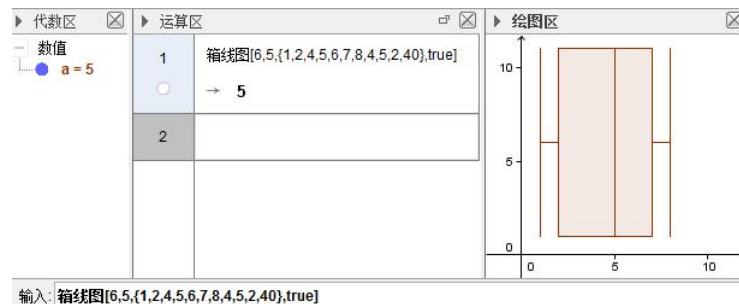
案例：“箱线图(6, 5, 1, 3, 5, 8, 9)”。



**BoxPlot**(<yOffset>, <yScale>, <List of Raw Data>, <Boolean Outliers>); 箱线图(<y 轴方向偏移量>, <y 轴方向范围>, <原始数据列表>, <是否离群值?true|false>)

画出一个箱线图, <是否离群值?true|false>: true 确认离群值, 然后再做箱线图; false 不确认离群值, 直接画箱线图。

案例：“箱线图(6, 5, {1, 2, 4, 5, 6, 7, 8, 4, 5, 2, 40}, true)”。

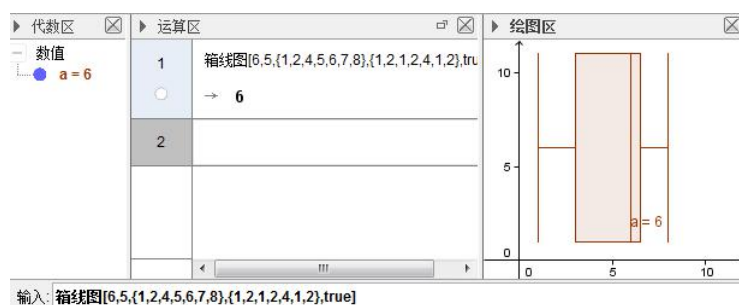


离群值画“X”的箱线图。这个指令中, 离群值是值是指低于“第一四分位数值-1.5\*IQR”或高于“第三四分位+1.5\*IQR”。

**BoxPlot**(<yOffset>, <yScale>, <List of Data>, <List of Frequencies>, <Boolean Outliers>); 箱线图(<y 轴方向偏移量>, <y 轴方向范围>, <数据列表>, <频数列表>, <是否离群值?true|false>)

依据频数画数据的箱线图。

案例：“箱线图(6, 5, {1, 2, 4, 5, 6, 7, 8}, {1, 2, 1, 2, 4, 1, 2}, true)”。

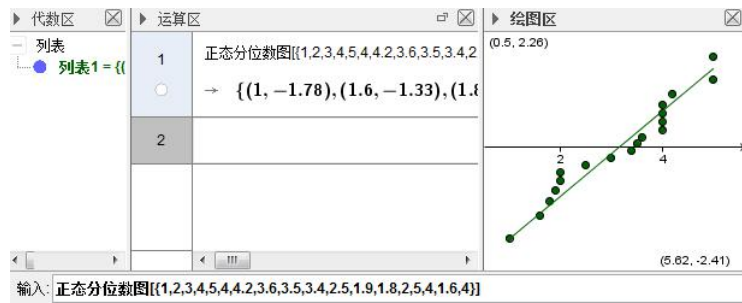


## 2.15.11 NormalQuantilePlot. 正态分位数图

**NormalQuantilePlot**(<List of Raw Data>); 正态分位数图(<原始数据列表>).

根据给定的原始数据列表创建正态分位数图, 绘制一条穿过点的线显示出正态数据的理想图示。点的形成是在 x 轴上使用数据值 y 轴上使用预期正态分数 (Z 分数)。

案例：“正态分位数图({1, 2, 3, 4, 5, 4, 4, 2, 3, 6, 3, 5, 3, 4, 2, 5, 1, 9, 1, 8, 2, 5, 4, 1, 6, 4})”。

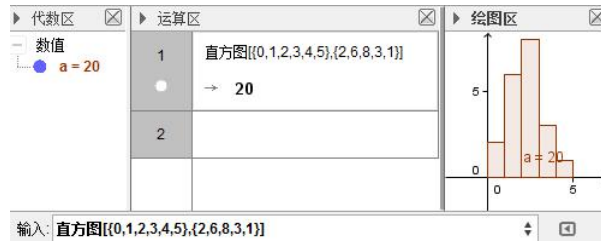


### 2.15.12 Histogram. 直方图

Histogram(<List of Class Boundaries>, <List of Heights>); 直方图(<组界列表>, <高度列表>).

创建条为给定高度值的直方图。组界决定了直方图每个条形的宽度和位置。

案例：“直方图({0, 1, 2, 3, 4, 5}, {2, 6, 8, 3, 1})” 创建有 5 个给定高度条的直方图。第一个条的位置在区间 (0, 1)，第二个条的位置在区间 (1, 2)，以此类推。



Histogram(<List of Class Boundaries>, <List of Raw Data>, <Boolean Use Density>, <Density Scale Factor>(optional)); 直方图(<组界列表>, <原始数据列表>, <是否应用密度? True|false>, <密度缩放因子(可选)>).

使用原始数据创建一个直方图。组界列表决定直方图每个条的宽度和位置且决定每个类有多少个数据元素。条的高度如下决定。

如果使用密度为 true, 高度=(密度缩放因子)\*(类频)/(类宽)

如果使用密度为 false, 高度=类频

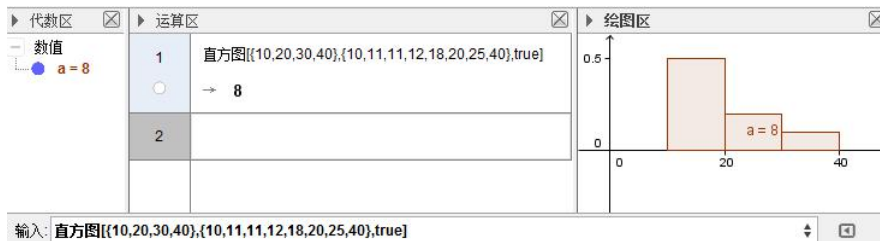
默认情况下, 使用密度为 true 而密度缩放因子为 1. 创建一个总面积为 n (数据值) 的直方图。

注: 原始数据的所有元素都需要在组界区间内, 否则返回“未定义”。

注: 约定每个类使用  $a \leq x < b$  规则最后一个类使用  $a \leq x \leq b$  除外。

案例: (默认直方图)

“直方图({10, 20, 30, 40}, {10, 11, 11, 12, 18, 20, 25, 40}, true)” 创建一个 3 条直方图, 第一个条高度 0.5, 第二个条高度 0.2, 第三个条高度 0.1。

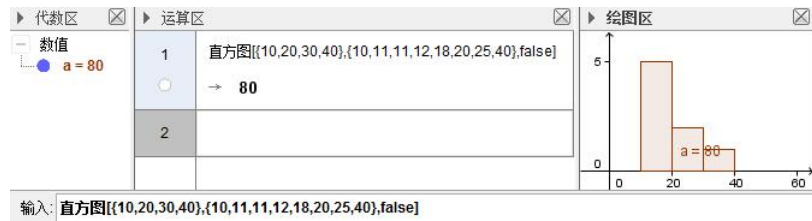


直方图总面积=0.5\*10+0.2\*10+0.1\*10=8。

案例: (计数直方图)

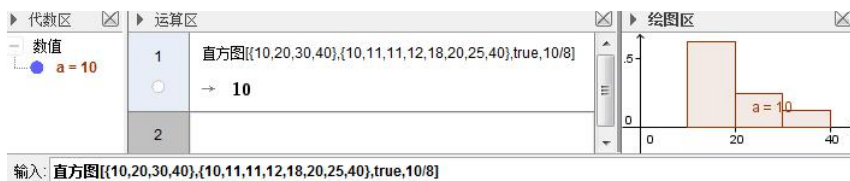
“直方图({10, 20, 30, 40}, {10, 11, 11, 12, 18, 20, 25, 40}, false)” 创建一个 3 条直方图。第一个

条高度 5，第二个条高度 2，第三个条高度 1。这个直方图没有使用密度缩放且条的高度等于类中值得个数。



### 案例：(相对频数直方图)

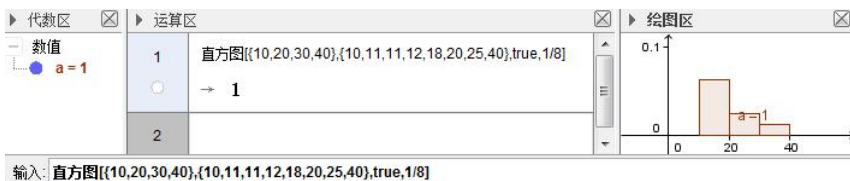
“直方图({10, 20, 30, 40}, {10, 11, 11, 12, 18, 20, 25, 40}, true, 10/8)” 创建一个 3 条直方图。第一个条高度 0.625，第二个条高度 0.25，第三个条高度 0.125。这个直方图使用了密度缩放因子给出了条的高度（等于类中值的比例）。



如果数据值的数目是  $n$ ，而类有恒定的宽度  $w$ ，那么密度缩放因子= $w/n$  创建一个相对频数直方图。

### 案例：(标准直方图)

“直方图({10, 20, 30, 40}, {10, 11, 11, 12, 18, 20, 25, 40}, true, 1/8)” 创建一个 3 条直方图。第一个条高度 0.625，第二个条高度 0.25，第三个条高度 0.125。



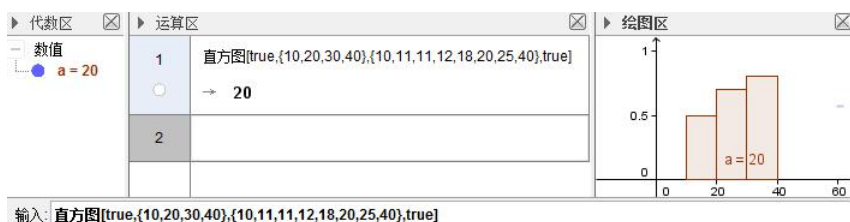
这个直方图总面积= $0.0625*10+0.025*10+0.0125*10=1$ 。

如果数据值的数量是  $n$ ，那么密度缩放因子= $1/n$  创建一个总面积为 1 的标准直方图。对于直方图回归密度曲线非常有用。

Histogram(<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>, <Boolean Use Density>, <Density Scale Factor>(optional)); 直方图(<是否累积? True|false>, <组界列表>, <原始数据列表>, <是否应用密度? True|false>, <密度缩放因子(可选)>)

如果累积是 true 将建立每个条的高等于已有频数和累加的一个直方图。

案例：“直方图(true, {10, 20, 30, 40}, {10, 11, 11, 12, 18, 20, 25, 40}, true)” 创建一个 3 条直方图，第一条高 0.5，第二条高 0.7，第三条高 0.8。



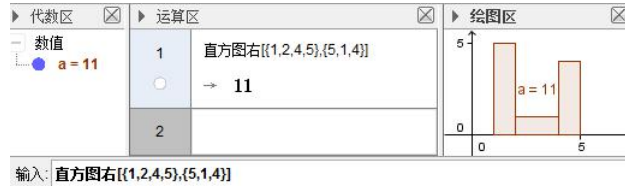
## 2. 15. 13 HistogramRight. 直方图右



HistogramRight(<List of Class Boundaries>, <List of Heights>); 直方图右(<组界列表>, <高度列表>)

同于“直方图(<组界列表>, <高度列表>)”。

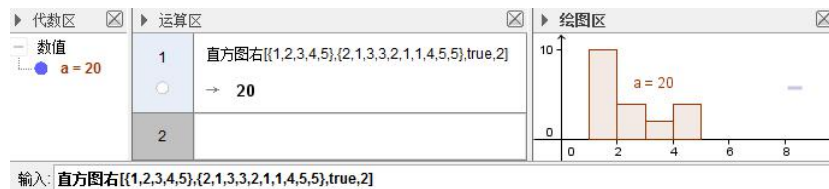
案例：“直方图右({1, 2, 4, 5}, {5, 1, 4})”。与“直方图(<组界列表>, <高度列表>)”得到的图形相同。



HistogramRight(<List of Class Boundaries>, <List of Raw Data>, <Boolean Use Density>, <Density Scale Factor>(optional)); 直方图右(<组界列表>, <原始数据列表>, <是否应用密度? True|false>, <密度缩放因子(可选)>)

同于“直方图(<组界列表>, <原始数据列表>, <是否应用密度? True|false>, <密度缩放因子(可选)>)”，所不同的是，如果一个数据等于一类的右边界，它被包括在这一类，而不是在下一个。

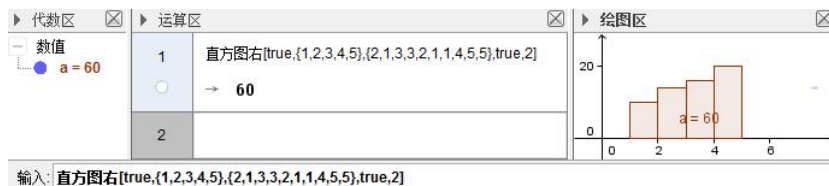
案例：“直方图右({1, 2, 3, 4, 5}, {2, 1, 3, 3, 2, 1, 1, 4, 5, 5}, true, 2)”。画出一个直方图，该指令主要是用组界列表画出区间的为：(1, 2)，(2, 3)，(3, 4)，(4, 5)包含最大值，不包含最小值



HistogramRight(<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>, <Boolean Use Density>, <Density Scale Factor>(optional)); 直方图右(<是否累积? True|false>, <组界列表>, <原始数据列表>, <是否应用密度? True|false>, <密度缩放因子(可选)>)

同于“直方图(<是否累积? True|false>, <组界列表>, <原始数据列表>, <是否应用密度? True|false>, <密度缩放因子(可选)>)”。所不同的是，如果一个数据等于一类的右边界，它被包括在这一类，而不是在下一个。

案例：“直方图右(true, {1, 2, 3, 4, 5}, {2, 1, 3, 3, 2, 1, 1, 4, 5, 5}, true, 2)”。



注：约定每个类使用  $a < x \leq b$  规则，第一个类使用  $a \leq x \leq b$  除外。

## 2.16 Text. 文本

### 2.16.1 TableText. 表格文本

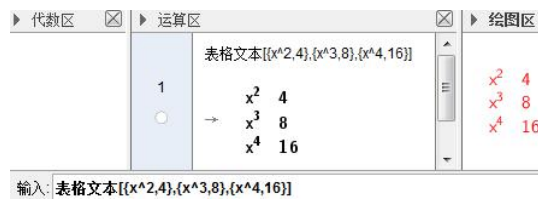
TableText(<List>, <List>, ...); 表格文本(<列表 1>, <列表 2>, ...)

创建一个由列表对象组成表格的文本对象。

注：默认情况下，每个列表显示在自己的行。

案例：“表格文本({x^2, 4}, {x^3, 8}, {x^4, 16})”创建有三行和两列的文本对象的表格。表格

中所有条目左对齐。



“表格文本(序列( $i^2$ ,  $i$ , 1, 10))”创建含有一行的文本对象表格。表格中所有条目左对齐。



`TableText(<List>, <List>, ..., <Alignment of Text>)`; 表格文本(<列表 1>, <列表 2>, ..., <对齐方式“v”\_垂直|“h”\_水平|“l”\_靠左|“r”\_靠右|“c”\_居中|...>)

创建含有对象列表内容表格文本。“文本对齐”选项控制文本表格对齐方向。

注：可用值是“vl”、“vc”、“vr”、“v”、“h”、“hl”、“hc”、“hr”。默认“hl”。

“v”=vertical（垂直的），列表元素为一列。

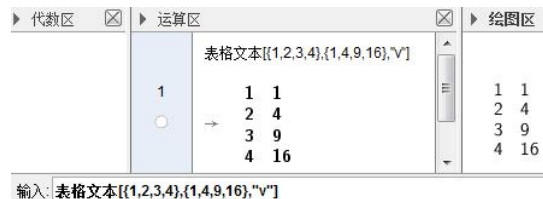
“h”=horizontal（水平的），列表元素为一行。

“l”=左对齐

“r”=右对齐

“c”=居中

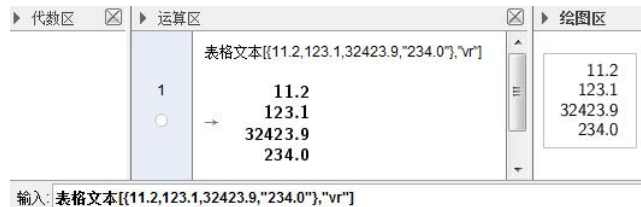
案例：“表格文本({1, 2, 3, 4}, {1, 4, 9, 16}, “v”)”创建有两列和四行元素左对齐的表格文本。



“表格文本({1, 2, 3, 4}, {1, 4, 9, 16}, “h”)”创建有两行和四列元素左对齐的表格文本。



“表格文本({11.2, 123.1, 32423.9, “234.0”}, “vr”)”创建元素右对齐只有一列的表格文本。



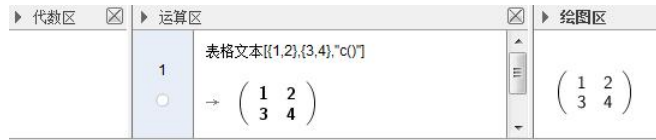
还可插入不同类型框架：使用以下符号“|||”、“||”、“{}”、“()”或“()”。

行分隔符，使用符号“\_”。

列分隔符，使用符号“|”。

以及不同色彩。

案例：“表格文本({1, 2}, {3, 4}, “c()”)”。



输入: 表格文本[[1,2],[3,4],"c0"]

“表格文本({1, 2}, {3, 4}, "c|\_")”。



输入: 表格文本[[1,2],[3,4],"c|\_"]

“表格文本({1, 2}, {3, 4}, "|\_|")”。



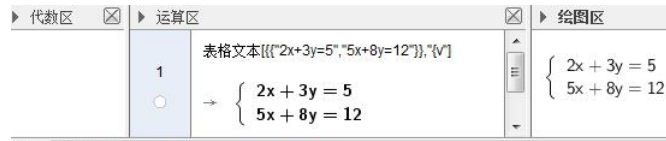
输入: 表格文本[[1,2],[3,4],"|\_|"]

“表格文本({1, 2}, {3, 4}, "|||\_|")”。



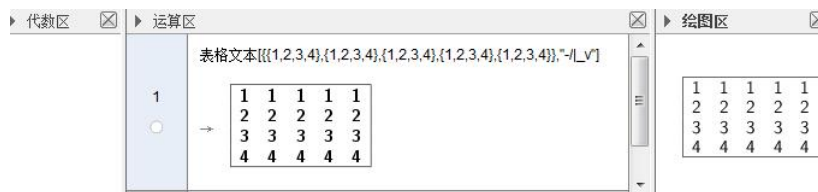
输入: 表格文本[[1,2],[3,4],"|||\_|"]

“表格文本({{"2x+3y=5", "5x+8y=12"}}, {"v"})”。



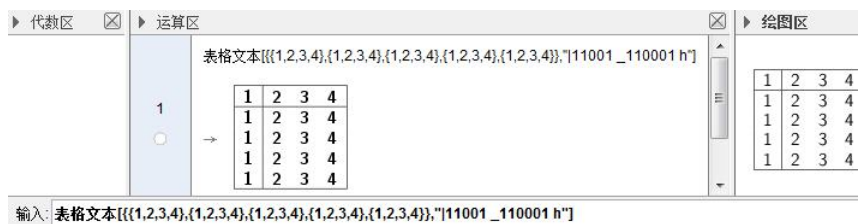
输入: 表格文本[{{"2x+3y=5","5x+8y=12"}}, {"v"}]

“表格文本({{1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}}, {"-/\_v"})”创建只有边框无分隔线的表格。



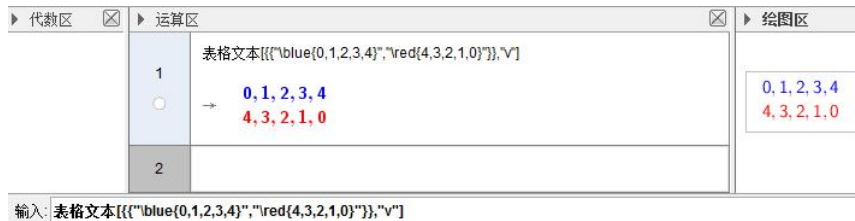
输入: 表格文本[{{1,2,3,4},{1,2,3,4},{1,2,3,4},{1,2,3,4},{1,2,3,4}}, {"-/\_v"}]

“表格文本({{1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4}}, {"|11001\_110001 h"})”创建向右数第一列和上边第一行的下方有分隔线的表格文本。语句中数值 1 表示这个数字行的下方有分隔线数值 0 表示没有分隔线或边框。



输入: 表格文本[{{1,2,3,4},{1,2,3,4},{1,2,3,4},{1,2,3,4},{1,2,3,4}}, {"|11001\_110001 h"}]

“表格文本({{"\blue{0, 1, 2, 3, 4}", "\red{4, 3, 2, 1, 0"}}, {"v"})”创建首行文本为蓝色，第二行文本为红色的表格文本。



**注：** 点击表格文本对象显示其样式栏，允许用户自定义对象外观、背景和文本颜色、文本和线型。

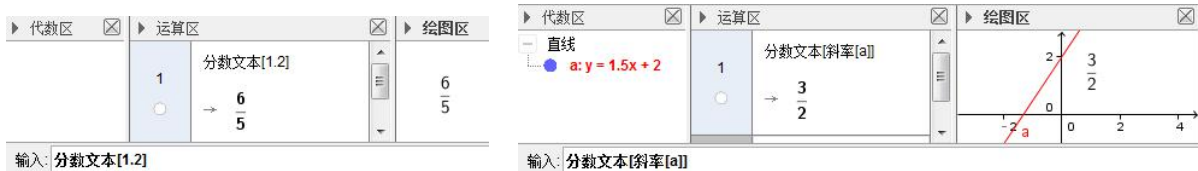
**注：** 表格文字并不要求每个列表中的元素个数相同，但是输入的文本除数字之外都必须用双引号才能够识别，否则识别为对象，将显示对象数值或其他。

## 2.16.2 FractionText. 分数文本

**FractionText(<Number>); 分数文本(<数字>)。**

转换数值为分数，它以文本（LaTeX）形式出现在绘图区。

**案例：** “分数文本(1.2)” 给出文本对象 “6/5”。

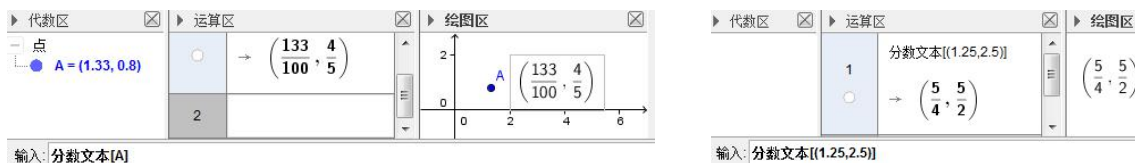


**案例：** 如果 “a:y=1.5 x+2” 是直线，然后 “分数文本(斜率(a))” 给出文本格式分数  $\frac{3}{2}$ 。

**FractionText(<Point>); 分数文本(<点>)。**

将点的坐标以分数形式显示在绘图区。

**案例：** 如果 “A=(1.33, 0.8)” 是一个点，然后 “分数文本(A)” 给出文本格式的坐标。“分数文本((1.25, 2.5))” 如下右图。



**注：** 参见 “根式文本” 指令。

## 2.16.3 SurdText. 根式文本

**SurdText(<Number>); 根式文本(<数值>)。**

创建将数值对象表现为  $\frac{a+b\sqrt{c}}{d}$  形式的文本。

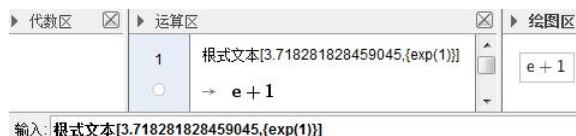
**案例：** “根式文本(2.414213562373095)” 创建 “ $1+\sqrt{2}$ ”；“根式文本(2.439230484541326)” 创建 “ $\frac{7+3\sqrt{3}}{5}$ ”。



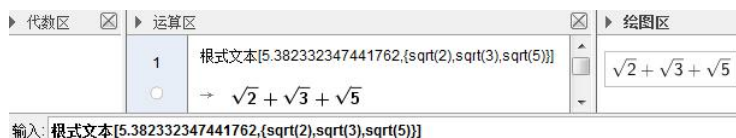
**SurdText (<Number>, <List>); 根式文本 (<数值>, <列表>)**。

创建使用列表中常量多重组合表示的文本，如果列表为空会使用常见常量。

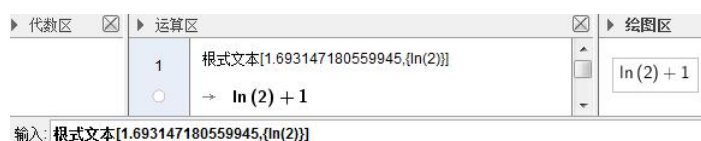
**案例：**“根式文本(3.718281828459045, {exp(1)})”创建“e+1”。



“根式文本(5.382332347441762, {sqrt(2), sqrt(3), sqrt(5)})”创建“ $\sqrt{2} + \sqrt{3} + \sqrt{5}$ ”。



“根式文本(1.693147180559945, {ln(2)})”创建“ln(2)+1”。



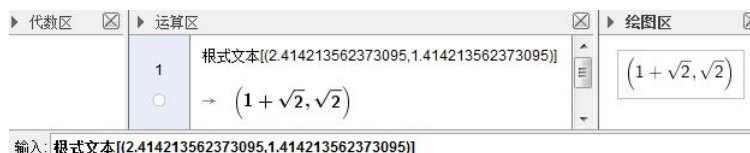
这里的无理数列表可以包括根式、常数 e、常数  $\pi$ 、对数等。

常用到的函数：sqrt(x)、cbrt(x)、exp(x)、ln(x)、lg(x)、ld(x)等。

**SurdText (<Point>); 根式文本 (<数值>, <列表>)**。

创建将点坐标表现为  $\frac{a+b\sqrt{c}}{d}$  形式的文本。

**案例：**“根式文本((2.414213562373095, 1.414213562373095))”创建“(1+ $\sqrt{2}$ ,  $\sqrt{2}$ )”。



**注：**想在一个文本对象中使用这个指令，需要在这个文本对象的属性对话框-文本选项卡中启用(勾选) LaTeX 数学式复选框。

由于这个指令将输入的值作为十进制的循环小数进行处理，因此结果有时会出乎意料。

如果不能找到一个合适的答案，则返回数值结果。

**案例：**“根式文本(1.23456789012345)”返回“1.23456789012345”。

**注：**参见“分数文本”和“科学计数法”指令。

## 2.16.4 FormulaText (LaTeX) . 公式文本

**FormulaText (<Object>); 公式文本 (<对象>)**。

将对象以 LaTeX 文本方式返回公式。

**案例：**设“a=2”且“f(x)=a x^2”。“公式文本(f)”返回“ $2x^2$ ” (以 LaTeX 文本格式)。



**注：**默认情况下，变量值都替代变量。

**FormulaText(<Object>, <Boolean for Substitution of Variables>); 公式文本(<对象>, <是否替换变量? true|false>)**。

将对象以 LaTeX 文本方式返回公式。布尔值变量决定值替代变量 (true) 还是在文本中显示变量名称 (false)。

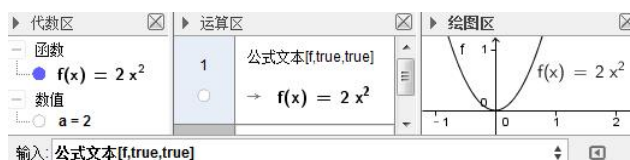
**案例:** 设 “a=2” 和 “f(x)=a x<sup>2</sup>”。“公式文本(f, true)” 返回 “2 x<sup>2</sup>” (以 LaTeX 文本格式)。“公式文本(f, false)” 返回 “a x<sup>2</sup>” (以 LaTeX 文本格式)。



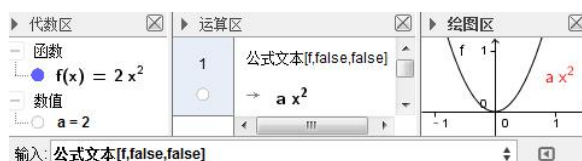
**FormulaText(<Object>, <Boolean for Substitution of Variables>, <Boolean Show Name>); 公式文本(<对象>, <是否替换变量? true|false>, <是否显示名称? true|false>)**。

将对象以 LaTeX 文本方式返回公式。第一个布尔值变量决定值替代变量 (true) 还是在文本中显示变量名称 (false)，第二个布尔值决定是在文本中显示对象名称 (true) 或不显示 (false)。

**案例:** 设 “a=2” 和 “f(x)=a x<sup>2</sup>”。“公式文本(f, true, true)” 返回 “f(x)=2 x<sup>2</sup>” (以 LaTeX 文本格式)。



“公式文本(f, false, false)” 返回 “a x<sup>2</sup>” (以 LaTeX 文本格式)。



**注:** 指令栏输入 “<math>”, 可替代本指令。

### 2.16.5 ScientificText. 科学计数法

这个指令在不同的英语变型中拼写不同: StandardForm (UK)、ScientificText (US+Aus)。

**ScientificText(<Number>); 科学计数法(<数字>)**。

用科学记数法显示数值。

**案例:** “科学计数法(0.002)” 给出 “2×10<sup>-3</sup>”。



**ScientificText(<Number>, <Precision>); 科学计数法(<数字>, <精确度>)**。

用科学记数法显示数值, 舍入到 (保留) 的有效位数由精确度指定。

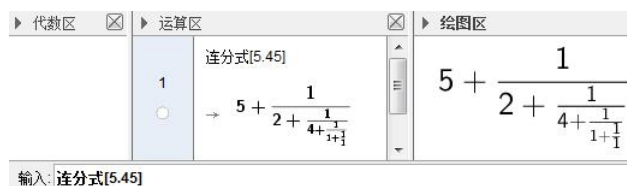
**案例:** “科学计数法(e, 5)” 给出 “2.7183×10<sup>0</sup>”。

### 2.16.6 ContinuedFraction. 连分式

**ContinuedFraction(<Number>); 连分式(<数字>)**。

创建给定数值的连分式。结果是 LaTeX 文本对象。公式计算代数精度是  $10^{-8}$ 。

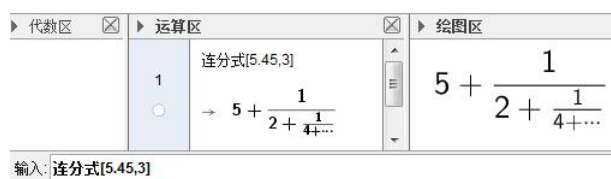
案例：“连分式(5.45)”。



**ContinuedFraction(<Number>, <Level>);** 连分式(<数字>, <层级>)。

创建给定数值的连分式。商的数值不超过层级，但上的数值总是达不到上文提及的精度。

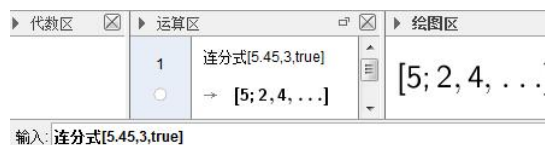
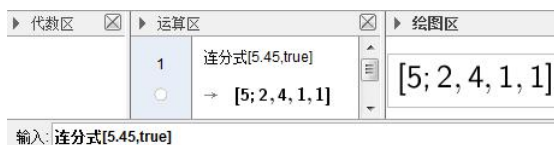
案例：“连分式(5.45, 3)”。



**ContinuedFraction(<Number>, <Level>, <Shorthand true|false>);** 连分式(<数字>, <层级>, <速记 true|false>)。

前两个参数同上，层级参数是可选项。当速记是 true，结果使用速记语法：LaTeX 文本包括连分式的整数部分列表。

案例：“连分式(5.45, true)”给出“(5; 2, 4, 1, 1)”；“连分式(5.45, 3, true)”给出“(5; 2, 4, ...)”。



## 2.16.7 VerticalText. 竖排文本

**VerticalText(<Text>);** 竖排文本(“<文本>”)。

返回将文本对象逆时针旋转 90° 后的新文本。呈现的结果使用 LaTeX 格式。

案例：“竖排文本(“ABC”)”。



**VerticalText(<Text>, <Point>);** 竖排文本(“<文本>”, <点>)。

将文本按照竖排在固定点显示。

注：移动点时，文本跟随。

## 2.16.8 UnicodeToText. 统一码转换为文本

**UnicodeToText(<List of Integers>);** 统一码转换为文本(<统一字符编码整数列表>)。

转换整数统一码数值到文本。

案例：“统一码转换为文本({104, 101, 108, 108, 111})”得出文本“hello”。“统一码转换为

文本({65, 66, 67}) = “ABC”



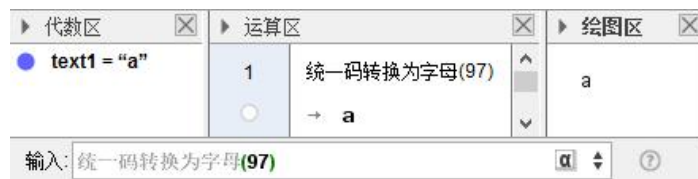
注：参见“文本转换为统一码”指令和“统一码转换为字母”指令。

## 2.16.9 UnicodeToLetter. 统一码转换为字母

UnicodeToLetter(<Integer>); 统一码转换为字母(<整数>)。

把在绘图区中以文本格式显示的整数统一码转换为字母。

案例：“统一码转换为字母(97)”得出文本“a”。



注：参见“字母转换为统一码”指令和“统一码转换为文本”指令。

注：统一码数值与字母的对应：

48-57: 0-9; 65-90: A-Z; 97-122: a-z。

参见“字母转换为统一码”和“统一码转换为文本”指令。

## 2.16.10 Text. 文本

Text(<Object>); 文本(<对象>)。

将对象的公式返回为文本对象。

注：默认情况下，变量值会替代变量。

案例：如“a=2”且“c=a<sup>2</sup>”，然后“文本(c)”返回文本“4”。



Text(<Object>, <Boolean for Substitution of Variables>); 文本(<对象>, <是否替换变量? true|false>)。

将对象的公式返回为文本对象。布尔值决定变量值替代变量(true)或在文本中显示变量名称(false)。

案例：当“a=2”且“c=a<sup>2</sup>”，然后，“文本(c, true)”返回文本“4”；“文本(c, false)”返回文本“a<sup>2</sup>”。

Text(<Object>, <Point>); 文本(<对象>, <点>)。

将对象的公式返回位于给定点的文本对象。

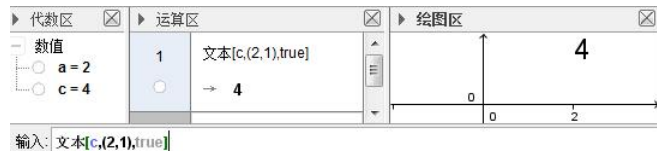
案例：“文本(“hello”, (2, 3))”返回位于点“(2, 3)”上的的文本“hello”。

Text(<Object>, <Point>, <Boolean for Substitution of Variables>); 文本(<对象>, <点>, <是否替换变量? true|false>)。

将对象的公式返回位于给定点的文本对象。布尔值决定变量值替代变量(true)或在文本中显示变量名称(false)。



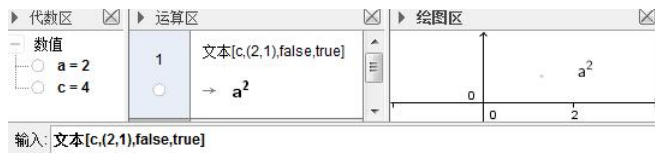
案例：如“a=2”和“c=a<sup>2</sup>”，然后“文本(c, (2, 1), true)”返回位于点“(2, 1)”的文本“4”。



`Text(<Object>, <Point>, <Boolean for Substitution of Variables>, <Boolean for LaTeX formula>)`; 文本(<对象>, <点>, <是否替换变量? true|false>, <是否应用 LaTeX 公式? true|false>e)。

将对象的公式返回位于给定点的文本对象。第一个布尔值决定变量值替代变量 (true) 或在文本中显示变量名称(false)。如果第二个布尔值是 true，结果使用 LaTeX。

案例：如“a=2”和“c=a<sup>2</sup>”，然后“文本(c, (2, 1), false, true)”返回使用 LaTeX 位于点(2, 1)的文本“a<sup>2</sup>”。



注：参见<sup>ABC</sup>“文本”工具。

### 2.16.11 TextToUnicode. 文本转换为统一码

`TextToUnicode("<Text>")`; 文本转换为统一码("<文本>")。

把文本逐个转换为一个统一码列表。

案例：“文本转换为统一码("Some text")”给出文本的统一码列表“{83, 111, 109, 101, 32, 116, 101, 120, 116}”。



案例：“text1="hello"”然后“文本转换为统一码(text1)”给出统一码数据列表“{104, 101, 108, 108, 111}”。



注：参见“统一码转换为文本”指令和“字母转换为统一码”指令。

### 2.16.12 Ordinal. 序数

`Ordinal(<Integer>)`; 序数(<自然数>)。

将一个自然数转换成序数（作为一个文本对象）。

案例：“序数(5)”返回“5th”。



编者注：验证没有“th”，这个转换就只是数字由数值型变成文本型了。

### 2.16.13 RotateText. 旋转文本

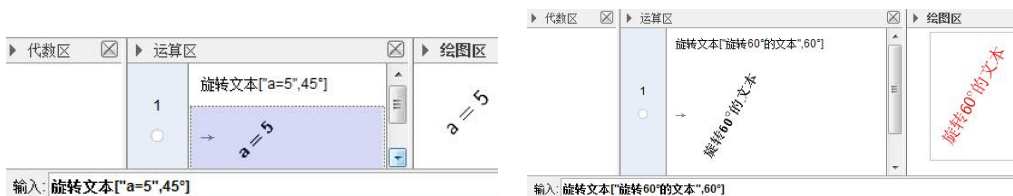
**RotateText(<Text>, <Angle>); 旋转文本("<文本>", <角度|弧度>)**。

创建一个按指定角度旋转了的文本。LaTeX 格式作为结果的源。

**案例：**“旋转文本("a=5", 45°)”

**注：**文本需要在双引号内；文本围绕文本方框的左上角（也可认为是第四个角点）旋转；旋转角度默认为弧度，也可明确使用角度符号°。

**注：**文本需要在双引号内；文本围绕文本方框的左上角（也可认为是第四个角点）旋转；旋转角度默认为弧度，也可明确使用角度符号°。



### 2.16.14 LetterToUnicode. 字母转换为统一码

**LetterToUnicode("<Letter>"); 字母转换为统一码("<字母>")**。

转换单个字母为它的统一码数字。

**案例：**“字母转换为统一码("a")”返回数字 97。



**注：**字母需要在一对儿引号内；参见“统一码转换为字母”指令和“文本转换为统一码”指令。

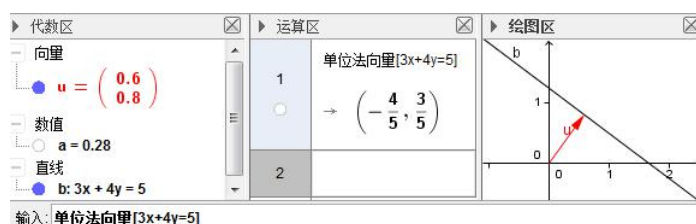
## 2.17 Vector&Matrix. 向量与矩阵

### 2.17.1 UnitPerpendicularVector (UnitOrthogonalVector) . 单位法向量

**UnitPerpendicularVector(<Line>); 单位法向量(<直线|射线>)**。

返回给定线的长度为 1 的法向量。

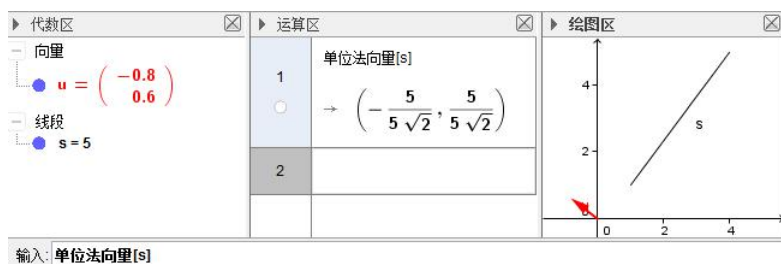
**案例：**“单位法向量(3x+4y=5)”得到向量 u。



**UnitPerpendicularVector(<Segment>); 单位法向量(<线段>)。**

返回给定线段的长度为 1 的法向量。

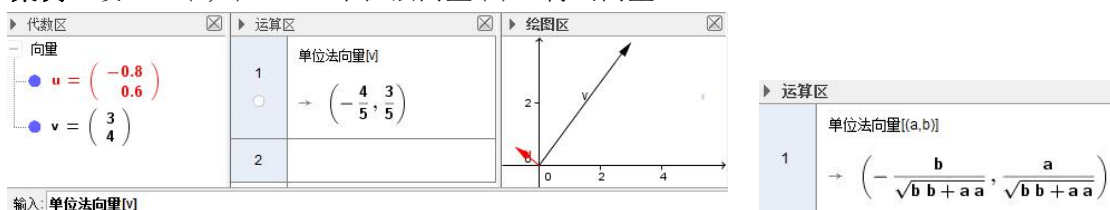
**案例：** 设 “s=线段((1, 1), (4, 5))”。“单位法向量(s)” 得出向量 u。



**UnitPerpendicularVector(<Vector>); 单位法向量(<向量>)。**

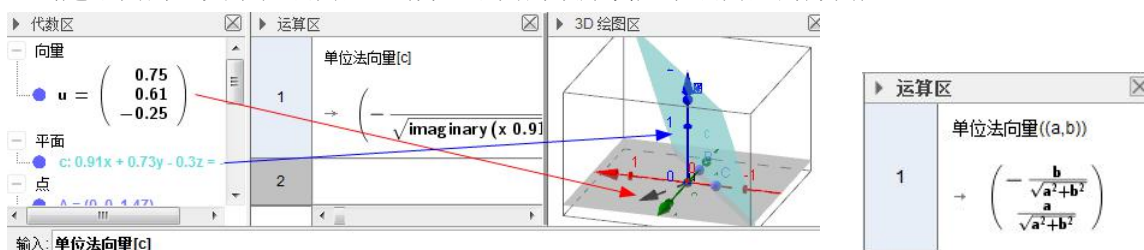
返回给定向量的长度为 1 的法向量。向量需要提前定义。

**案例：** 设 “v=(3, 4)”。“单位法向量(v)” 得出向量 u。



**UnitPerpendicularVector(<Plane>); 单位法向量(<平面>)。**

创建平面的正交单位法向量。(构造平面的顺序决定了法向量的方向)



**CAS Syntax (运算区语法)**

**UnitPerpendicularVector(<Vector>); 单位法向量(<向量>)。**

返回给定向量的长度为 1 的法向量。

**案例：** “单位法向量((a, b))” 得出  $(-\frac{b}{\sqrt{bb+aa}}, \frac{a}{\sqrt{bb+aa}})$ 。格式有时会不同。

**注：** 参见“法向量”指令。

## 2.17.2 Identity. 单位矩阵

**Identity(<Number>); 单位矩阵(<数值>)。**

给出给定阶数的单位矩阵。

**案例：** “单位矩阵(3)” 得出 3 阶单位矩阵。



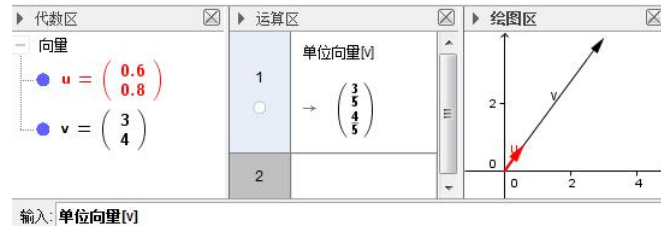
**注：** 如果 A 是一个 n 阶方阵， $A^0$  得出的结果与“单位矩阵(n)” 相同。

### 2.17.3 UnitVector. 单位向量

UnitVector(<Vector>); 单位向量(<向量>)。

得出与给定向量方向相同长度为 1 的向量。向量需要定义过。

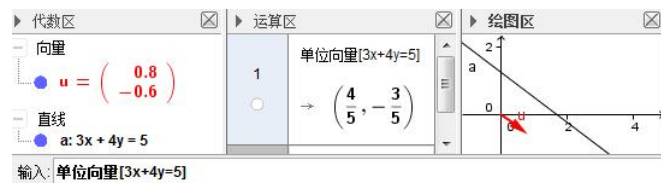
案例：设 “ $v=(3, 4)$ ”。“单位向量( $v$ )” 得出向量  $u$ 。



UnitVector(<Line>); 单位向量(<几何对象>)。

返回与给定直线同向的长度 1 的向量。

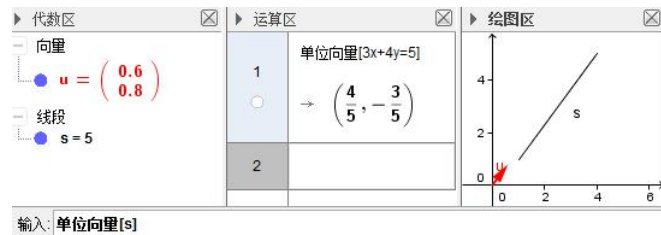
案例：“单位向量( $3x+4y=5$ )” 得出向量  $u$ 。



UnitVector(<Object>); 单位向量(<几何对象>)

返回与给定几何对象决定的向量同向的长度 1 的向量。

案例：设 “ $s$ =线段( $(1, 1), (4, 5)$ )” ，“单位向量( $s$ )” 得出向量  $u$ 。



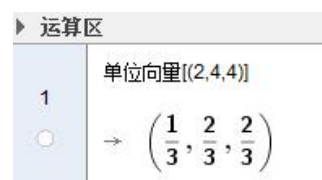
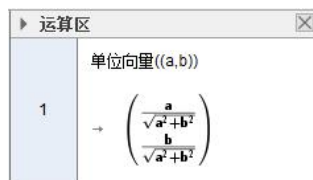
案例：设有多边形 1 存在，指令“单位向量(多边形 1)” 给出多边形 1 的法向量，这个法向量从点(0,0,0)出发，在 3D 空间内。

CAS Syntax (运算区语法)

UnitVector(<Vector>); 单位向量(<向量>)。

得出与给定向量方向相同长度为 1 的向量。

案例：“单位向量( $(a, b)$ )”。



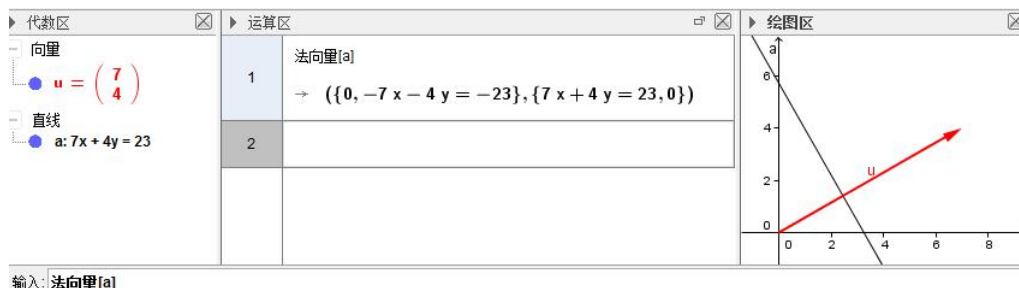
案例：“单位向量( $(2, 4, 4)$ )”。

### 2.17.4 PerpendicularVector (OrthogonalVector). 法向量

PerpendicularVector(<Line>); 法向量(<直线>)。

返回直线的法向量。

案例：设“直线((1, 4), (5, -3))”是直线 a。“法向量(a)”得出直线 a 的法向量“ $u=(7, 4)$ ”。

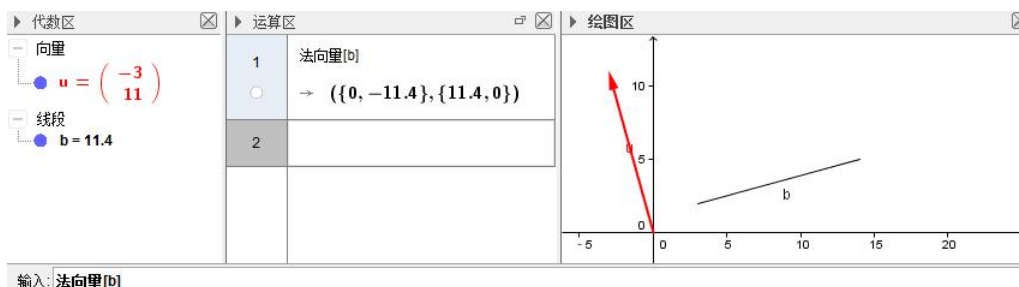


注：具有方程式“ $ax+by=c$ ”形式的一条直线，其法向量为“(a, b)”。

**PerpendicularVector(<Segment>); 法向量(<线段>)**。

返回与线段相同长度的线段法向量。

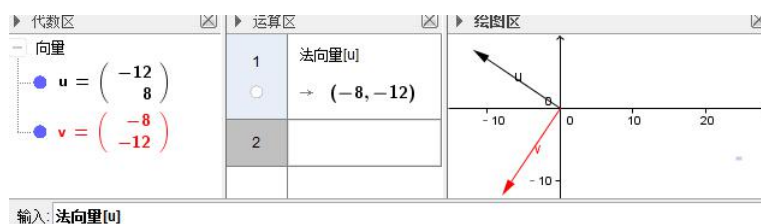
案例：设“线段((3, 2), (14, 5))”是线段 k。“法向量(k)”得出线段 k 的法向量“ $u=(-3, 11)$ ”。



**PerpendicularVector(<Vector>); 法向量(<向量>)**。

返回给定向量的法向量。

案例：设“向量((-12, 8))”是向量 u。“法向量(u)”得出向量 u 的法向量“ $v=(-8, -12)$ ”。



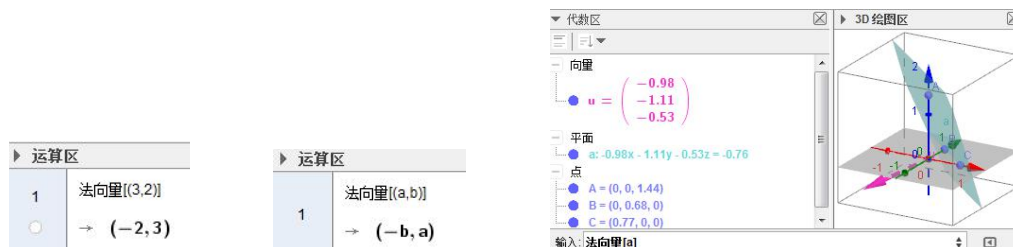
注：携带设置坐标(a, b)的向量其法向量为(-b, a)。

**CAS Syntax (运算区语法)**

**PerpendicularVector(<Vector>); 法向量(<向量>)**。

返回给定向量的法向量。

案例：“法向量((3, 2))”得出“{-2, 3}”；“法向量((a, b))”得出向量“{-b, a}”。



**PerpendicularVector(<Plane>); 法向量(<平面>)**。

创建平面正交向量。

案例：“法向量(a)”得出“平面 a”的法向量。“法向量(xoy 平面)”，得到 xoy 平面的法向量“ $u=(0, 0, 1)$ ”。

注：参见“单位法向量”指令。

## 2.17.5 ReducedRowEchelonForm. 简化行梯阵式

**ReducedRowEchelonForm(<Matrix>); 简化行梯阵式(<矩阵>)**。

返回矩阵的简约行梯阵形式。

**案例：**“简化行梯阵式({{1, 6, 4}, {2, 8, 9}, {4, 5, 6}})”得到矩阵。



**CAS Syntax (运算区语法)**

**ReducedRowEchelonForm(<Matrix>); 简化行梯阵式(<矩阵>)**。

返回矩阵的简约行梯阵形式。

**案例：**“简化行梯阵式({{1, 6, 4}, {2, 8, 9}, {4, 5, 6}})”得到矩阵见上图。

**注：**参见“逆反”和“转置”指令。

## 2.17.6 MatrixRank. 矩阵的秩

**MatrixRank(<Matrix>); 矩阵的秩(<矩阵>)**。

返回给定矩阵的秩。

**案例：**“矩阵的秩({{2, 2}, {1, 1}})”得出 1; “矩阵的秩({{1, 2}, {3, 4}})”得出 2; 设“A={{1, 2, 3}, {1, 1, 1}, {2, 2, 2}}”为一个 3×3-矩阵, “矩阵的秩(A)”得出 2。

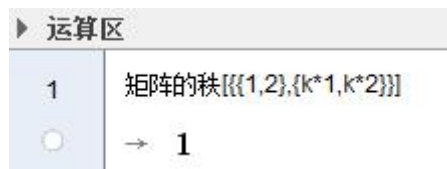
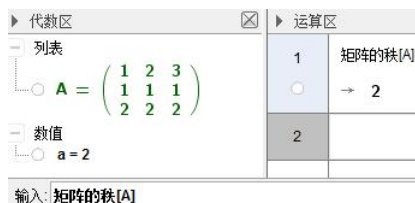


**CAS Syntax (运算区语法)**

**MatrixRank(<Matrix>); 矩阵的秩(<矩阵>)**。

返回给定矩阵的秩。

**案例：**“矩阵的秩({{2, 2}, {1, 1}})”得出 1; “矩阵的秩({{1, 2}, {3, 4}})”得出 2; “矩阵的秩({{1, 2}, {k\*1, k\*2}})”得出 1。



**释义：**将矩阵做初等行变换后, 非零行的个数叫行秩, 将其进行初等列变换后, 非零列的个数叫列秩。矩阵的秩是方阵经过初等行变换或者列变换后的行秩或列秩。矩阵的列秩和行秩总是相等的, 因此它们可以简单地称作矩阵 A 的秩, 通常表示为  $r(A)$ ,  $rk(A)$  或  $rankA$ 。

## 2.17.7 Invert. 逆反

**Invert(<Matrix>); 逆反(<矩阵>)**。

给出指定矩阵的逆矩阵。

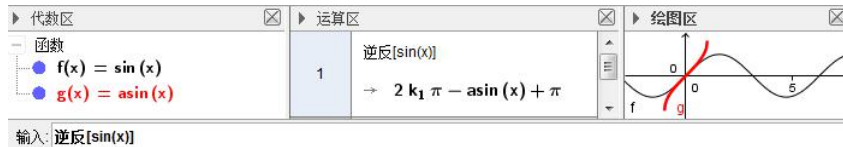
案例：“逆反({{1,2},{3,4}})”得出矩阵的逆反矩阵。



Invert(<Function>); 逆反(<函数>)。

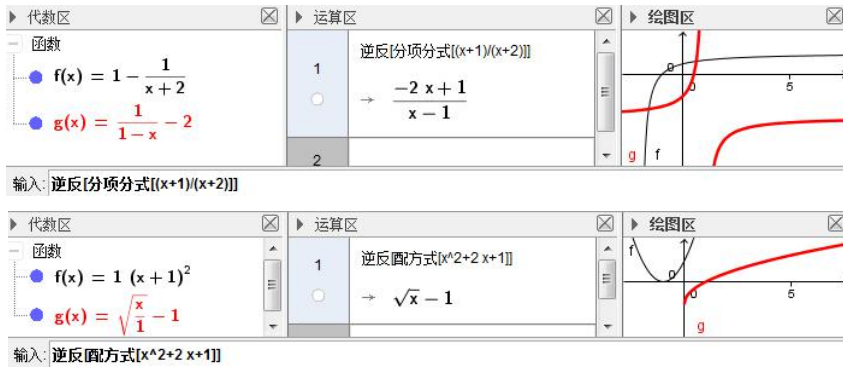
给出函数的反函数。

案例：“逆反(sin(x))”得出“asin(x)”。



注：函数的方程式中必须只包含一个 x，并且不考虑其定义域与值域，例如  $f(x)=x^2$  或  $f(x)=\sin(x)$ 。若函数的方程式中存在多于一个的 x，可加入另外的指令：

案例：“逆反(分项分式((x+1)/(x+2)))”和“逆反(配方式(x^2+2 x+1))”得出反函数。



CAS Syntax (运算区语法)

Invert(<Matrix>); 逆反(<矩阵>)。

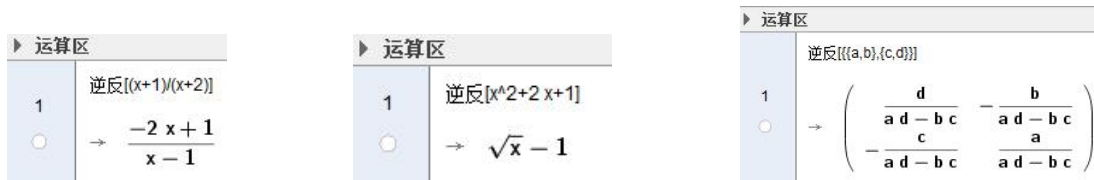
给出指定矩阵的逆矩阵。

案例：“逆反({{a,b},{c,d}})”得出矩阵的逆反矩阵。

Invert(<Function>); 逆反(<函数>)。

给出函数的反函数。

案例：“逆反((x+1)/(x+2))”得出  $\frac{-2x+1}{x-1}$ ”；“逆反(x^2+2 x+1)”得出  $\sqrt{x}-1$ ”。



注：在运算区，这个指令能针对函数包括多个 x 执行；参见“转置”和“简化行梯阵式”指令。

## 2.17.8 Dimension. 维度

Dimension(<Object>); 维度(<点|向量|矩阵>)

给出一个向量或是一个矩阵的维度。

案例：“维度({1,2,0,-4,3})”得出 5。

案例：“维度({{1, 2}, {3, 4}, {5, 6}})” 得出 “{3, 2}”。



CAS Syntax (运算区语法)

Dimension(<Object>); 维度(<点|向量|矩阵>)

给出一个向量或是一个矩阵的维度。

案例：“维度({1, 2, 0, -4, 3})” 得出 5。

案例：“维度({{a, b}, {c, d}, {e, f}})” 得出 “{3, 2}”。



## 2. 17.9 Vector. 向量

Vector(<Point>); 向量(<终点(原点为起点)>).

返回给定点的位置向量。

案例：“向量((3, 2))” 得出起点是原点的向量 u。

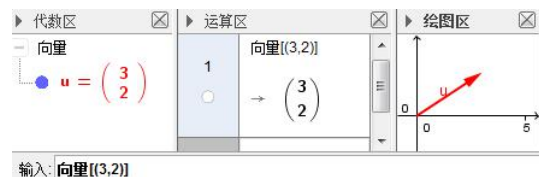
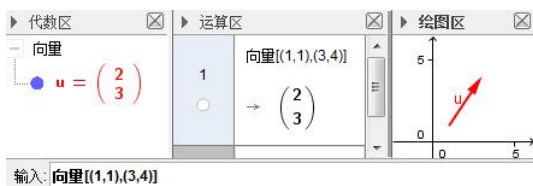
自 GeoGebra 5.0, 这个指令适用于 3D 空间点。




Vector(<Start Point>, <End Point>); 向量(<起点>, <终点>).

创建一个从起点到终点的向量。

案例：“向量((1, 1), (3, 4))” 得出一个向量 u。



注：参见  “向量” 工具。

## 2. 17.10 Determinant. 行列式

Determinant(<Matrix>); 行列式(<矩阵>).

给出矩阵的行列式的值。

案例：“行列式({{1, 2}, {3, 4}})” 得出 “a=-2”。

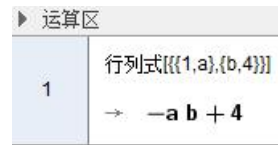
CAS Syntax (运算区语法)

Determinant(<Matrix>); 行列式(<矩阵>).

给出矩阵的行列式的值。如果矩阵含有没有定义的变量，给出的行列式值中含有公式。

案例：“行列式({{1, a}, {b, 4}})” 得出 “-a b+4”。





### 2.17.11 ApplyMatrix. 应用矩阵

**ApplyMatrix(<Matrix>, <Object>); 应用矩阵(<矩阵>, <对象>)**。

如果 P 是二维点，M 是 2 x 2 矩阵，按照 O 到 P 的映射投影形成点 P。

**案例：**设 “M={{cos(π/2), -sin(π/2)}, {sin(π/2), cos(π/2)}}” 是转轴矩阵，而 “u=(2, 1)” 是给定的向量（对象）。“应用矩阵(M, u)” 得出向量 “u’=(-1, 2)”，此结果是将向量 u 进行数学上的正旋转（逆时针）90° 而得。



如果 P 是二维点，M 是点 3x3 矩阵，project 是投影，映射点(x, y, z)到(x/z, y/z)。即：project: (M\*(x(P), y(P), 1))，project 是一个将点(x, y, z)映射到(x/z, y/z)的投影点（基于一个 3x3 矩阵）。

**案例：**设 “M={{1, 1, 0}, {0, 1, 1}, {1, 0, 1}}” 是一个矩阵而 “u=(2, 1)” 是一个给定向量。“应用矩阵(M, u)” 得出向量 “u’=(1, 0.67)” 。实际上  $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 3 \end{pmatrix}$ ，而(3/3=1, 2/3≈0.67)（保留两位小数）。



如果 P 是三维空间点且 M 是 3x3 矩阵，点 M\*P。

如果点 P 是三维空间点且 M 是 2x2 矩阵，矩阵 N 是矩阵 M 转型或 3 阶：给定  $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ ，然后，

$$N = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{pmatrix}。点 N*P。$$

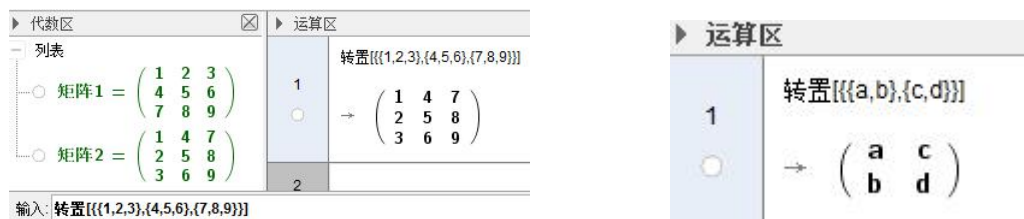
**注：**这里的对象可以是点，向量，也可以是图片（未使用绝对位置）。图片的计算方法是先计算其位置点，然后复制个副本到对应位置。

### 2.17.12 Transpose. 转置

**Transpose(<Matrix>); 转置(<矩阵>)**。

转置矩阵。

案例：“转置({{1, 2, 3}, {4, 5, 6}, {7, 8, 9}})” 得出矩阵的转置矩阵。



CAS Syntax (运算区语法)

Transpose(<Matrix>); 转置(<矩阵>).

转置矩阵。

案例：“转置({{a, b}, {c, d}})” 得出矩阵的转置矩阵。

注：参见“逆反”以及“简化行梯阵式”指令。

## 2.18 Optimization. 优化指令

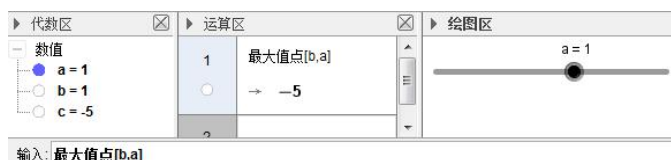
### 2.18.1 Maximize. 最大值点

这个指令在不同的英语变型中拼写不同：Maximize (US)、Maximise (UK+Aus)。

Maximize(<Dependent number>, <Free number>); 最大值点(<因变数>, <滑动条>).

计算自变量（自由的数值对象）给出因变量（派生的数值对象）最大值时的值。自变量（自由的数值对象）必须是一个滑动条，这个滑动条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或避免过多时间计算而放弃。

案例：a 是滑动条参数 (-5, 4)，数值  $b = a^2 / 3$ ，“最大值点(b, a)” 返回 b 最大值时的 a 值。



Maximize(<Dependent Number>, <Point on Path>); 最大值点(<因变数>, <界点>).

路径上的点驱动因变数变化，返回当因变数最大时，路径上的点。

案例：设 C 是线段 AB 上的动点，数值  $b = (x(C))^{2/3}$ ，“最大值点(b, C)” 返回 b 最大值时线段 AB 上的点（下图点 D）。



注：参见“最小值点”指令。

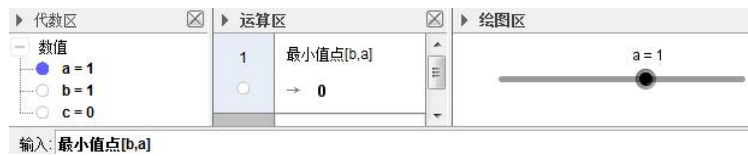
### 2.18.2 Minimize. 最小值点

这个指令在不同的英语变型中拼写不同：Minimize (US)、Minimise (UK+Aus)。

Minimize(<Dependent number>, <Free number>); 最小值点(<因变数>, <参数>).

计算自变量（自由的数值对象）给出因变量（派生的数值对象）最小值时的值。自变量（自由的数值对象）必须是一个滑动条，这个滑动条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或避免过多时间计算而放弃。

**案例：** a 是滑动条参数，数值  $b=a^{2/3}$ ，“最大值点(b, a)” 返回 b 最小值时的 a 值。



**Minimize**(<Dependent Number>, <Point on Path>); **最小值点**(<因变数>, <界点>).

路径上的点驱动因变数变化，返回当因变数最小时，路径上的点。

**案例：** 设 C 是线段 AB 上的动点，数值  $b=(x(C))^{2/3}$ ，“最大值点(b, C)” 返回 b 最小值时线段 AB 上的点（下图点 D）。



注：参见“最大值点”指令。

**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。

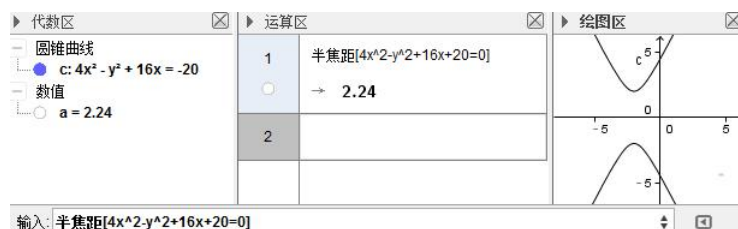
## 2.19 Conic. 圆锥曲线

### 2.19.1 LinearEccentricity. 半焦距

**LinearEccentricity**(<Conic>); **半焦距**(<圆锥曲线>).

计算圆锥曲线的半焦距，比如圆锥曲线中心点和它的一个焦点的距离。

**案例：** “半焦距  $(4x^2 - y^2 + 16x + 20 = 0)$ ” 返回 2.24。



### 2.19.2 Semicircle. 半圆

**Semicircle**(<Point>, <Point>); **半圆**(<点 1>, <点 2>).

创建线段侧方两点间的半圆。



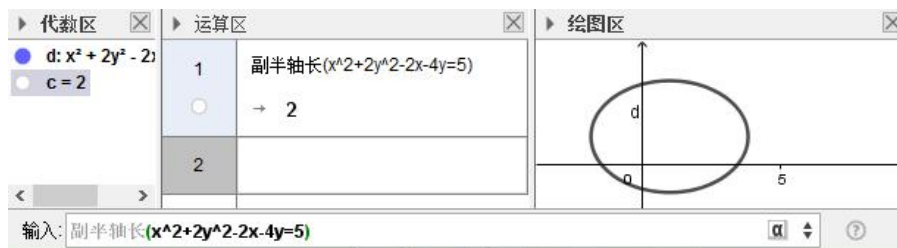
注：参见“半圆”工具。

### 2.19.3 SemiMinorAxisLength (SecondAxisLength) . 副半轴长

**SemiMinorAxisLength(<Conic>); 副半轴长(<圆锥曲线>).**

返回圆锥曲线的副半轴长(副轴一半)。

案例：“副半轴长( $x^2+2y^2-2x-4y=5$ )”得出 2。



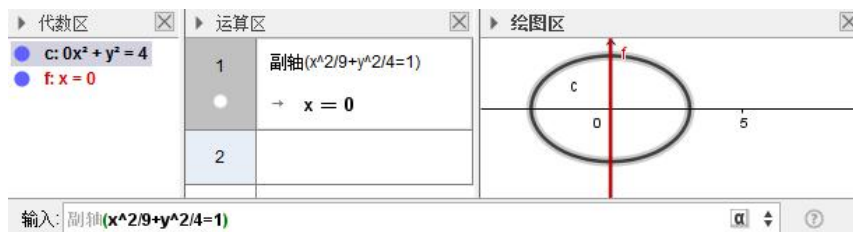
注：参见“主半轴长”指令。

### 2.19.4 MinorAxis (SecondAxis) . 副轴

**MinorAxis(<Conic>); 副轴(<圆锥曲线>).**

返回圆锥曲线的主轴直线方程。

案例：“副轴( $x^2/9+y^2/4=1$ )”返回  $x=0$ 。



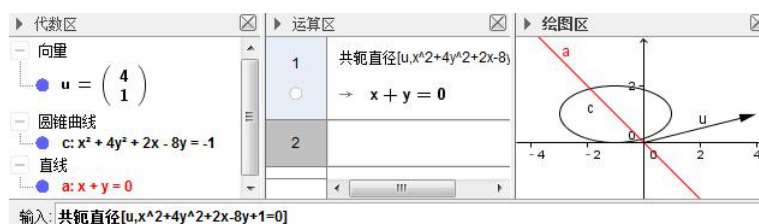
注：参见“主轴”指令。

### 2.19.5 ConjugateDiameter (Diameter) . 共轭直径

**ConjugateDiameter(<Vector>, <Conic>); 共轭直径(<向量>, <圆锥曲线>).**

返回圆锥曲线平行于向量的直径的共轭直径。

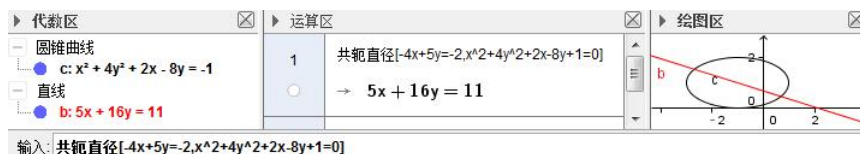
案例：设“ $u=(4, 1)$ ”是一个向量。然后共轭直径( $u, x^2+4y^2+2x-8y+1=0$ )得出直线“ $x+y=0$ ”。



**ConjugateDiameter(<Line>, <Conic>); 共轭直径(<直线>, <圆锥曲线>).**

返回圆锥曲线平行于直线的直径的共轭直径。

案例：“共轭直径 $(-4x+5y=-2, x^2+4y^2+2x-8y+1=0)$ ”得出直线“ $5x+16y=11$ ”。



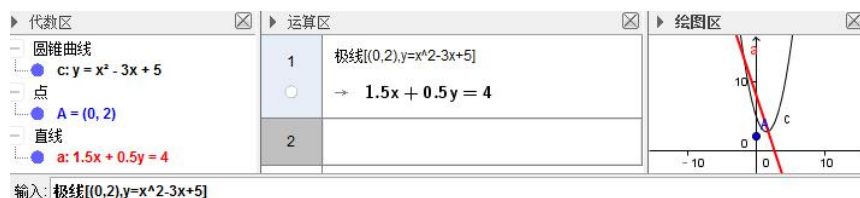
释义：连结椭圆上任意两点的线段叫弦。过椭圆中心的弦叫直径，平行于直径 DE 的弦的中点的轨迹 AB 和直径 DE 互为共轭直径。类似地可定义双曲线的共轭直径。由于上述 DE 直径是任意取的，因此椭圆的共轭直径有无数对。当一对共轭直径互相垂直时，即为椭圆的主轴和副轴。

## 2.19.6 Polar. 极线

Polar(<Point>, <Conic>); 极线(<点>, <圆锥曲线>)。

创建给定点关于圆锥曲线的极线。

案例：“极线 $((0, 2), y=x^2-3x+5)$ ”创建直线“ $1.5x+0.5y=4$ ”。



注：参见  “极线/径线”工具。

Polar(<Line>, <Conic>); 极线(<直线>, <圆锥曲线>)。

上一个指令的逆运算（给出线，求出点）

案例：指令“极线 $(1.5x+0.5y=4, y=x^2-3x+5)$ ”得到点 $(0, 2)$ 。

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。

## 2.19.7 Parameter. 焦参数

Parameter(<Parabola>); 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。

案例：“焦参数 $(y=x^2-3x+5)$ ”返回 0.5。

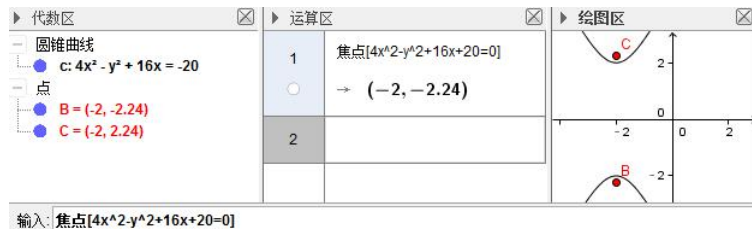


## 2.19.8 Focus. 焦点

Focus(<Conic>); 焦点(<圆锥曲线>)。

得出圆锥曲线的(全部)焦点。

案例：“焦点 $(4x^2-y^2+16x+20=0)$ ”返回给定双曲线的焦点：“ $A=(-2, -2.24)$ ”和“ $B=(-2, 2.24)$ ”。



注：参见“准线”指令。

### 2.19.9 Eccentricity. 离心率

Eccentricity(<Conic>); 离心率(<圆锥曲线>)。

计算一个圆锥曲线的离心率。

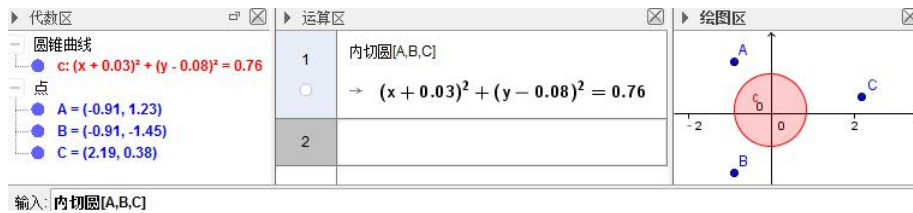
案例：“离心率( $x^2/9+y^2/4=1$ )”返回“a=0.75”。



### 2.19.10 Incircle. 内切圆

Incircle(<Point>, <Point>, <Point>); 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。

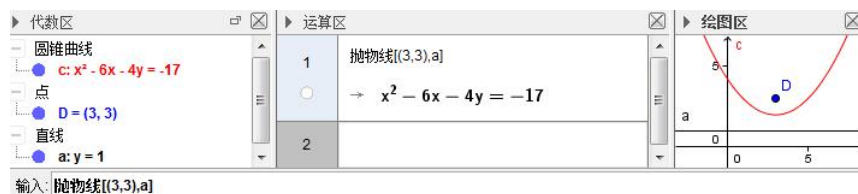


### 2.19.11 Parabola. 抛物线

Parabola(<Point>, <Line>); 抛物线(<焦点>, <准线>)。

返回一个焦点为 F 和准线为 g 的抛物线。

案例：设 a 为“直线((0, 1), (2, 1))”。“抛物线((3, 3), a)”得出“ $x^2 - 6x - 4y = -17$ ”。



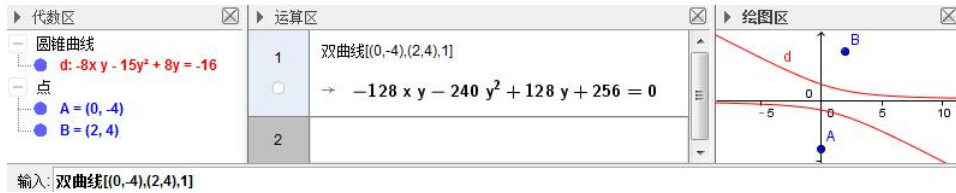
注：参见“抛物线”工具。

### 2.19.12 Hyperbola. 双曲线

Hyperbola(<Focus>, <Focus>, <Semimajor Axis Length>); 双曲线(<焦点 1>, <焦点 1>, <主半轴长>)。

创建给定两个焦点和主半轴长度的双曲线。

案例：“双曲线((0, -4), (2, 4), 1)”得出“-8xy-15y<sup>2</sup>+8y=-16”。

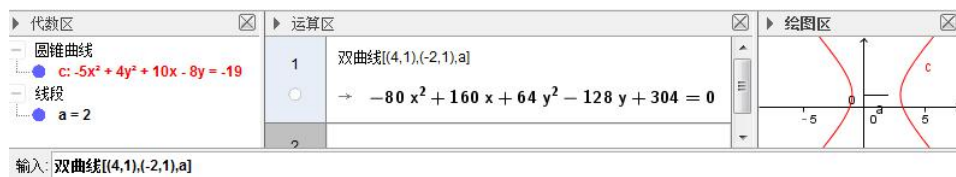


注：如果条件“0<2×主半轴长度<焦点之间的距离”不能满足，将会得到一个椭圆。

Hyperbola(<Focus>, <Focus>, <Segment>); 双曲线(<焦点 1>, <焦点 1>, <半主轴线段>).

按指定的两个焦点创建一个双曲线，其主半轴长度等于给定线段的长度。

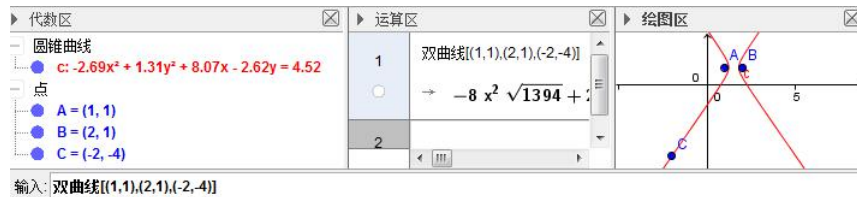
案例：假设 a 为“线段((0, 1), (2, 1))”。“双曲线((4, 1), (-2, 1), a)”得出“-5x<sup>2</sup>+4y<sup>2</sup>+10x-8y=-19”。



Hyperbola(<Focus>, <Focus>, <Point>); 双曲线(<焦点 1>, <焦点 2>, <双曲线上一点>).

创建给定点为焦点且穿过给定点的双曲线。

案例：“双曲线((1, 1), (2, 1), (-2, -4))”得出“-2.69x<sup>2</sup>+1.30y<sup>2</sup>+8.07x-2.62y=4.52”。



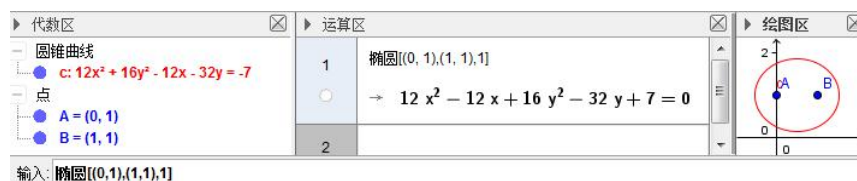
注：参见“双曲线”工具。

### 2.19.13 Ellipse. 椭圆

Ellipse(<Focus>, <Focus>, <Semimajor Axis Length>); 椭圆(<焦点 1>, <焦点 1>, <主半轴长>).

创建给定两个焦点和主半轴长度的椭圆。

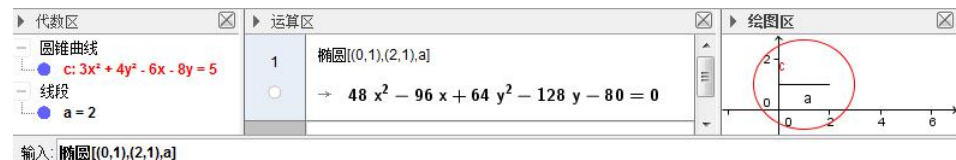
案例：“椭圆((0, 1), (1, 1), 1)”得出“12x<sup>2</sup>+16y<sup>2</sup>-12x-32y=-7”。



注：如果条件“2×主半轴长度>焦点之间的距离”不能满足，将会得到一个双曲线。

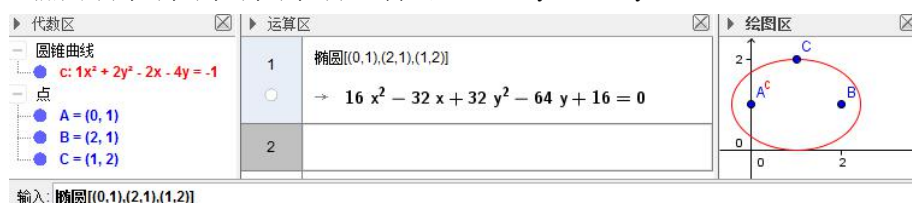
Ellipse(<Focus>, <Focus>, <Segment>); 椭圆(<焦点 1>, <焦点 1>, <半主轴线段>).

按指定的两个焦点创建一个椭圆，其主半轴长度等于给定线段的长度。



**案例：**设 a 为“线段((0, 1), (2, 1))”，“椭圆((0, 1), (2, 1), a)”得出“ $3x^2 + 4y^2 - 6x - 8y = 5$ ”。  
**Ellipse(<Focus>, <Focus>, <Point>); 椭圆(<焦点 1>, <焦点 1>, <椭圆上一点>).**  
 创建一个给定两个焦点并穿过一给定点的椭圆。

**案例：**“椭圆((0, 1), (2, 1), (1, 2))”得出“ $x^2 + 2y^2 - 2x - 4y = -1$ ”。



**注：**参见“椭圆”工具。

## 2.19.14 Circle. 圆周

**Circle(<Point>, <Radius Number>); 圆周(<圆心>, <半径长度>).**

得出一个给定圆心和半径的圆。

**Circle(<Point>, <Segment>); 圆周(<圆心>, <半径>).**

得出一个给定圆心和半径等于给定线段长度的圆。

**Circle(<Point>, <Point>); 圆周(<圆心>, <圆上一点>).**

得出一个给定圆心过一个给定点的圆。

**Circle(<Line>, <Point>); 圆周(<轴线>, <圆上一点>).**

创建直线为轴且过点的圆。

**Circle(<Point>, <Point>, <Point>); 圆周(<点 1>, <点 2>, <点 3>).**

得出过三个给定点的圆（假设三点不共线）。

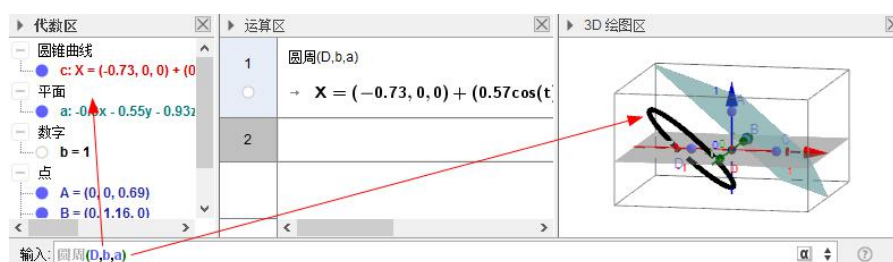
**注：**参见“圆规(半径与圆心)”，“圆(圆心与一点)”，“圆(圆心与半径)”和“圆(过三点)”工具。

**注：**自 GeoGebra5，这个指令同样对 3D 对象起作用。

**Circle(<Point>, <Radius>, <Direction>); 圆周(<圆心>, <半径>, <轴向量>).**

创建一个指定圆心、半径和轴平行于一条直线、向量或平面向量的圆。

**案例：**“圆周(<圆心>, <半径>, <平面>)”得出一个以平面垂直向量为轴平行于平面的圆。



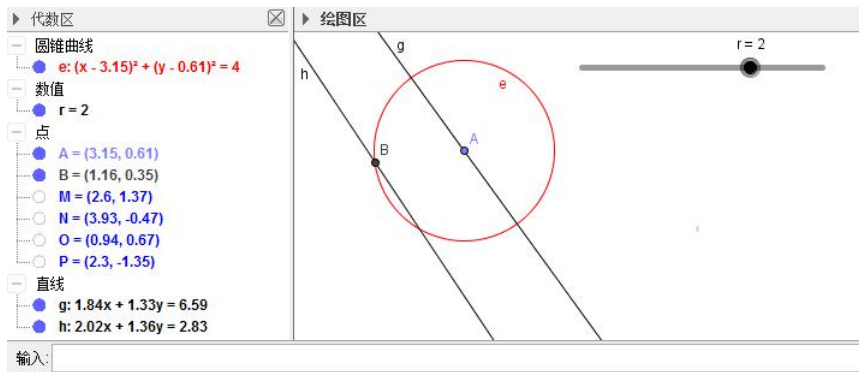
**Circle(<Point>, <Point>, <Direction>); 圆周(<圆心>, <圆上一点>, <轴向量>).**

创建一个指定圆心过点且轴平行于向量的圆。

**辅助案例：**使用圆固定两个对象的距离

圆对于制作两个对象间不变的距离十分有用。要求是有两个点 A 和 B 在两条直线 g（点 A 所在线）和 h（点 B 所在线）上，点 A 可以移动且点 B 与点 A 有一个固定的距离 r。可以定义一个以 A 为圆心半径为 r 的圆，定义圆与直线 h 的交点 B。因为圆与直线有两个交点（当不相切或相离时），隐藏且忽略第二个交点即可。





## 2.19.15 Conic. 圆锥曲线

**Conic**(**<Point>**, **<Point>**, **<Point>**, **<Point>**, **<Point>**); **圆锥曲线**(**<点 1>**, **<点 2>**, **<点 3>**, **<点 4>**, **<点 5>**).

返回一个过给定五点的圆锥曲线。

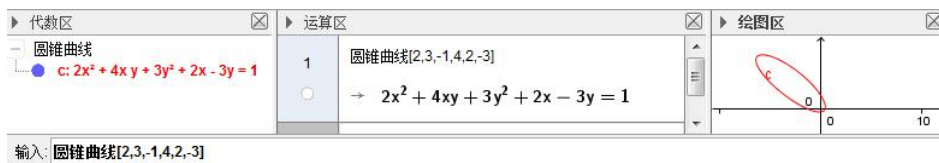
**案例：**圆锥曲线((0, -4), (2, 4), (3, 1), (-2, 3), (-3, -1))得出  $151x^2 - 7xy + 72y^2 + 14x - 42y = 1320$ 。

**注：**如果其中四点在一条直线上，不能定义圆锥曲线。

**Conic**(**<Number a>**, **<Number b>**, **<Number c>**, **<Number d>**, **<Number e>**, **<Number f>**); **圆锥曲线**(**<x 方系数>**, **<y 方系数>**, **<常数项>**, **<xy 系数>**, **<x 系数>**, **<y 系数>**).

返回一个圆锥曲线  $ax^2 + dxy + by^2 + ex + fy = -c$ 。

**案例：**“圆锥曲线(2, 3, -1, 4, 2, -3)”得出  $2x^2 + 4xy + 3y^2 + 2x - 3y = 1$ 。



**注：**参见“圆锥曲线”工具和“系数列表”指令。

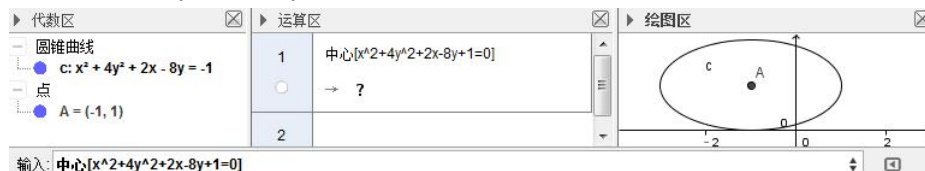
## 2.19.16 Center (Centre). 中心

这个指令在不同的英语变型中拼写不同：Center (US)、Centre (UK+Aus)。

**Center**(**<Conic>**); **中心**(**<圆锥曲线>**)。

返回一个圆、椭圆或双曲线的中心。

**案例：**“中心( $x^2 + 4y^2 + 2x - 8y + 1 = 0$ )”返回点“A=(-1, 1)”。

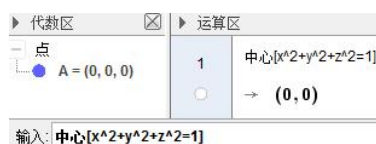


**Center**(**<Conic>**); **中心**(**<圆锥曲线>**)。

返回一个圆、椭圆或双曲线的中心。

**Center**(**<Quadric>**); **中心**(**<二次曲面>**)

创建二次曲面中心(诸如球，圆锥等)。



案例：“中心( $x^2+y^2+z^2=1$ )”得出“(0, 0, 0)”。

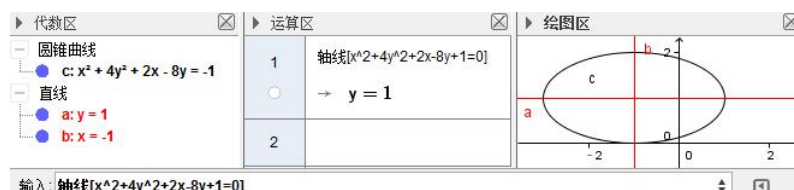
注：参见“中点/中心”工具。

## 2.19.17 Axes. 轴线

**Axes(<Conic>); 轴线(<圆锥曲线>)。**

返回一个圆锥曲线的主轴和副轴的方程。

案例：“轴线( $x^2+4y^2+2x-8y+1=0$ )”返回直线“ $a:y=1$ ”和直线“ $b:x=-1$ ”。

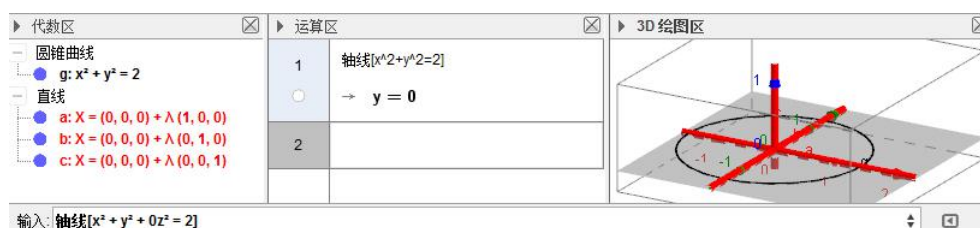


注：参见“主轴”和“副轴”指令。

**Axes(<Quadric>); 轴线(<二次曲面>)。**

创建给定二次曲面的3个轴。

案例：“轴线( $x^2+y^2=3$ )”返回三条直线。“ $a:X=(0, 0, 0)+\lambda(1, 0, 0)$ ”，“ $b:X=(0, 0, 0)+\lambda(0, 1, 0)$ ”和“ $c:X=(0, 0, 0)+\lambda(0, 0, 1)$ ”。



注：如果给定的二次曲面是圆柱，指令得出底面的两个轴和旋转轴。

如果给定的二次曲面是球体，指令得出的三条轴分别平行于坐标轴。

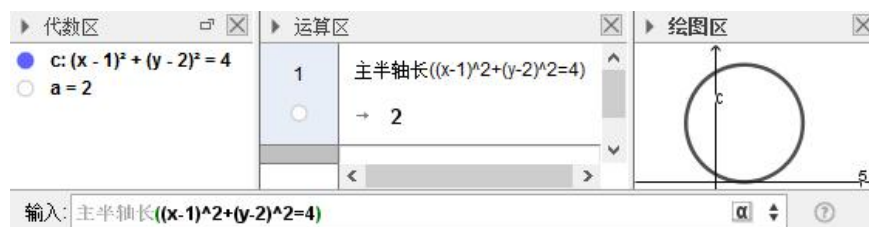
注：需要激活3D绘图区，显示此指令的效果。

## 2.18.18 SemiMajorAxisLength (FirstAxisLength). 主半轴长

**SemiMajorAxisLength(<Conic>); 主半轴长(<圆锥曲线>)。**

返回圆锥曲线的主半轴长(主轴一半)。

案例：“主半轴长( $(x-1)^2+(y-2)^2=4$ )”得出2。



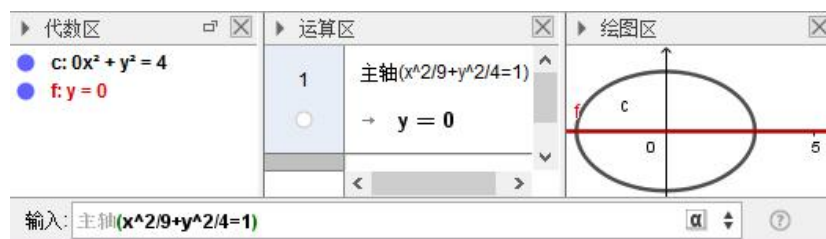
注：参见“副半轴长”指令。

## 2.19.19 MajorAxis (FirstAxis). 主轴

**MajorAxis(<Conic>); 主轴(<主轴>)。**

返回圆锥曲线的主轴直线方程。

案例：“主轴( $x^2/9+y^2/4=1$ )”返回“ $y=0$ ”。



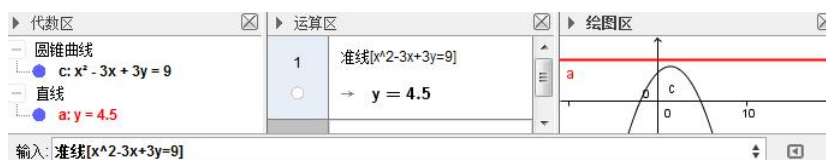
注：参见“副轴”指令。

## 2.19.20 Directrix. 准线

Directrix(<Conic>); 准线(<圆锥曲线>)。

得出圆锥曲线的准线。

案例：“准线( $x^2-3x+3y=9$ )”得出直线“ $y=4.5$ ”。



注：参见“焦点”指令。

## 2.20 CASSpecific. 运算区专属指令

使用默认格局新打开 GeoGebra 时，“运算区指令”不会加载到窗口右侧的“指令帮助”中。当运算区调用过运算区指令后，系统就自动加载“运算区指令”，运算区专属指令帮助才能出现。

多数指令可以同时适用于“指令栏”和“运算区”，在上文的具体指令介绍中，多数指令同时介绍了在两个区域中的使用方法和结果。在“运算区”，可用的接近 200 条指令中，其中至少有以下指令只适于运算区，按照中文拼音序列示如下。

序号	英文名称	中文名称
1	MixedNumber	带分数
2	GroebnerDegRevLex	分次反字典 Groebner 基
3	GroebnerLexDeg	分次字典 Groebner 基
4	CSolve	复数解
5	CSolutions	复数解集
6	CFactor	复数域因式分解
7	CIFactor	复无理数域因式分解
8	SolveCubic	解三次多项式
9	Numeric	近似数
10	Laplace	拉普拉斯变换
11	InverseLaplace	拉普拉斯逆变换
12	Substitute	替换
13	Eliminate	消元
14	Rationalize	有理化
15	ToExponential	转换为指数形式

随着 GeoGebra 的不断更新, 这些指令也会有所变化。

### 2. 20. 1 MixedNumber. 带分数

**MixedNumber**(**<Number>**); **带分数**(**<数值>**)。

将指定的数值转化为一个带分数。参见有理化指令。

**案例:** “带分数(3.5)” 得出  $3+\frac{1}{2}$ ; “带分数(12/3)” 得出=4; “带分数(12/14)” 得出=  $0+\frac{6}{7}$ 。”



### 2. 20. 2 GroebnerDegRevLex. 分次反字典序 Groebner 基

**GroebnerDegRevLex**(**<List of Polynomials>**); **分次反字典序 Groebner 基**(**<多项式列表>**)。

计算由分次反字典序 (也被看做是总额反字典序、degrevlex 序或 grevlex 序) 变量决定的 Groebner 基多项式列表。

**案例:** “分次反字典序 Groebner 基( $\{x^3-y-2, x^2+y+1\}$ )” 得出如图结果。



**GroebnerDegRevLex**(**<List of Polynomials>**, **<List of Variables>**); **分次反字典序 Groebner 基**(**<多项式列表>**, **<变量列表>**)。

计算由指定分次反字典序 (也被看做是总额反字典序、degrevlex 序或 grevlex 序) 变量决定的 Groebner 基多项式列表。

**案例:** “分次反字典序 Groebner 基( $\{x^3-y-2, x^2+y+1\}, \{y, x\}$ )” 得出如图结果。

**注:** 参见 “字典序 Groebner 基” 和 “分次字典序 Groebner 基” 指令。

### 2. 20. 3 GroebnerLexDeg. 分次字典序 Groebner 基

**GroebnerLexDeg**(**<List of Polynomials>**); **分次字典序 Groebner 基**(**<多项式列表>**)。

计算由分次字典序 (也被看做是 grlex、tdeg、lexdeg、总额字典序或不排序) 变量决定的 Groebner 基多项式列表。

**案例:** “分次字典序 Groebner 基( $\{x^3-y-2, x^2+y+1\}$ )” 得出如图结果。



**GroebnerLexDeg**(**<List of Polynomials>**, **<List of Variables>**); **分次字典序 Groebner 基**(**<多项式列表>**, **<变量列表>**)。

计算由指定分次字典序 (也被看做是 grlex、tdeg、lexdeg、总额字典序或不排序) 变量决定的 Groebner 基多项式列表。

**案例:** “分次字典序 Groebner 基( $\{x^3-y-2, x^2+y+1\}, \{y, x\}$ )” 得出如图结果。

注：参见“字典序 Groebner 基”和“分次反字典序 Groebner 基”指令。

## 2. 20. 4 CSolve. 复数解

**CSolve(<Equation>); 复数解(<方程>)**。

对一个所给方程的主变量求解并返回全部解的列表，可以是复数解。

案例：“复数解( $x^2=-1$ )”得出“ $x^2=-1$ ”的复数解为“ $\{x=i, x=-i\}$ ”。

运算区	运算区	运算区
1 复数解[ $x^2=-1$ ] → $\{x=i, x=-i\}$	1 复数解[ $a^2=-1,a$ ] → $\{a=i, a=-i\}$	1 复数解[ $\{y^2=x-1, x=2*y-1\}, \{x,y\}$ ] → $\{\{x=1-2i, y=1-i\}, \{x=1+2i, y=1+i\}\}$

**CSolve(<Equation>, <Variable>); 复数解(<方程>, <变量>)**。

对一个所给方程的未知变量求解并返回全部解的列表，可以是复数解。

案例：“复数解( $a^2=-1, a$ )”得出“ $a^2=-1$ ”的复数解“ $\{a=i, a=-i\}$ ”。

**CSolve(<List of Equations>, <List of Variables>); 复数解(<方程组>, <变量列表>)**。

对一组所给方程组的未知变量求解并返回全部解的列表，可以是复数解。

案例：“复数解( $\{y^2=x-1, x=2*y-1\}, \{x, y\}$ )”得出“ $y^2=x$ ”和“ $x=2*y-1$ ”的复数解集“ $\{\{x=1+2i, y=1+i\}, \{x=1-2i, y=1-i\}\}$ ”。

注：通过按“ALT+i”可以得到复数i；参见“复数解集”指令和“精确解”指令。

## 2. 20. 5 CSolutions. 复数解集

**CSolutions(<Equation>); 复数解集(<方程>)**。

对一个所给方程的主变量求解并返回全部解的列表，可以是复数解。

案例：“复数解集( $x^2=-1$ )”得出“ $x^2=-1$ ”的复数解“ $\{i, -i\}$ ”。

运算区	运算区	运算区
1 复数解集[ $x^2=-1$ ] → $\{i, -i\}$	1 复数解集[ $a^2=-1,a$ ] → $\{i, -i\}$	1 复数解集[ $\{y^2=x-1, x=2*y-1\}, \{x,y\}$ ] → $\begin{pmatrix} 1-2i & 1-i \\ 1+2i & 1+i \end{pmatrix}$

**CSolutions(<Equation>, <Variable>); 复数解集(<方程>, <变量>)**。

对一个给定方程的未知变量求解并返回全部解的列表，可以是复数解。

案例：“复数解集( $a^2=-1, a$ )”得出“ $a^2=-1$ ”的复数解“ $\{i, -i\}$ ”。

**CSolutions(<List of Equations>, <List of Variables>); 复数解集(<方程(组)列表>, <变量列表>)**

对一组所给方程组的未知变量求解并返回全部解的列表，可以是复数解。

案例：“复数解集( $\{y^2=x-1, x=2*y-1\}, \{x, y\}$ )”得出方程组“ $y^2=x-1$ ”和“ $x=2*y-1$ ”的解“ $\{\{1-2i, 1-i\}, \{1+2i, 1+i\}\}$ ”。

注：通过按“ALT+i”可以得到复数i；参见“复数解”指令和“解集”指令。

## 2. 20. 6 CFactor. 复数域因式分解

**CFactor(<Expression>); 复数域因式分解(<表达式>)**。

在复数范围内分解给定的表达式。

案例：“复数域因式分解( $x^2+4$ )”得出“ $(x+2i)(x-2i)$ ”，复数范围内的“ $x^2+4$ ”因式分解。

运算区	运算区	运算区
1 复数域因式分解( $x^2+4$ ) → $(x+2i)(x-2i)$	1 复数域因式分解( $a^2+x^2,a$ ) → $(a+ix)(a-ix)$	1 复数域因式分解( $a^2+x^2,x$ ) → $(x+ia)(x-ia)$

**CFactor(<Expression>, <Variable>); 复数域因式分解(<表达式>, <变量>)**。

在复数范围内，按照给定的变量为主项分解因式。

**案例：**“复数域因式分解( $a^2+x^2, a$ )”得出“ $a^2+x^2$ ”的以  $a$  为主变量的因式“ $(a+xi)(a-xi)$ ”。

**案例：**“复数域因式分解( $a^2+x^2, x$ )”得出“ $a^2+x^2$ ”的以  $x$  为主变量的因式“ $(x+ai)(x-ai)$ ”。

**注：**这个指令超越有理数范畴，参见“因式分解”指令。

## 2. 20.7 CIFactor. 复无理数域因式分解

**CIFactor(<Expression>); 复无理数域因式分解(<表达式>)**。

在复杂无理数范围内的因式分解。

**案例：**“复无理数域因式分解( $x^2+x+1$ )”结果如下左图。



**CIFactor(<Expression>, <Variable>); 复无理数域因式分解(<表达式>, <变量>)**。

做指定变量在复杂无理数内的因式分解。

**案例：**“复无理数域因式分解( $a^2+a+1, a$ )”返回结果如上右图。

参见“实数域因式分解”指令。

## 2. 20.8 SolveCubic. 解三次多项式

**SolveCubic(<Cubic Polynomial>); 解三次多项式(<三次多项式>)**。

解给的三次多项式并返回所有解的列表。

**案例：**“解三次多项式( $x^3-1$ )”得出如下图结果。



**注：**时常需要手工化简结果，如“解三次多项式( $x^3+x^2+x+1$ )”。

## 2. 20.9 Numeric. 近似数

**Numeric(<Expression>); 近似数(<表达式>)**。

尝试对给定的表达式进行数值逼近（取整）。小数位数取决于在选项菜单中设定的精确度。

**案例：**“近似数( $3/2$ )”得出 1.5。



**Numeric(<Expression>, <significant figures>); 近似数(<表达式>, <有效数值个数>)**。

尝试使用输入的有效数值个数对给定的表达式进行数值逼近（取整）。

**案例：**“近似数( $\sin(1), 20$ )”得出 0.84147098480789650665。

**注：**如果没有指定足够的数值个数，那么可能得到一个由于浮点取消的明显错误。如：“近似数( $-500000000/785398163*\sin(785398163/500000000)*1258025227.19^2+500000000/785398163*$

1258025227.19<sup>2</sup>,10)”会给出0。但是指定了其他的有效位数，可能就会因为有效数字位数的精度影响结果，将有效值个数设定为30，就会出现另外不同的结果。比如输入“近似数(-500000000/785398163\*sin(785398163/500000000))\*1258025227.19<sup>2</sup>+500000000/785398163\*1258025227.19<sup>2</sup>,30)”会给出0.318309886344536696694580314215。

运算区	
1	近似数[-500000000/785398163*sin(785398163/500000000)*1258025227.19^2+500000000/785398163*1258025227.19^2,10] → 4096
2	近似数[-500000000/785398163*sin(785398163/500000000)*1258025227.19^2+500000000/785398163*1258025227.19^2,30] → 0.318309886344536696694580314215

运算区	
1	近似数[π,150] → 3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651328230664709384460955058

## 2.20.10 Laplace. 拉普拉斯变换

**Laplace(<Function>); 拉普拉斯变换(<函数>)**。

返回给定函数的拉普拉斯变换。

**案例：**“拉普拉斯变换(sin(t))”返回“ $\frac{1}{s^2+1}$ ”。（实践中，t 会变为 s，其他字母就不变）

运算区		运算区		运算区		运算区	
1	拉普拉斯变换[sin(t)] → $\frac{1}{s^2+1}$	1	拉普拉斯变换[sin(b)] → $\frac{1}{b^2+1}$	1	拉普拉斯变换[sin(a*t),t] → $\frac{a}{a^2+t^2}$	1	拉普拉斯变换[sin(a*t),a] → $\frac{t}{a^2+t^2}$

**Laplace(<Function>,<Variable>); 拉普拉斯变换(<函数>,<变量>)**。

返回给定变量函数的拉普拉斯变换。

**案例：**“拉普拉斯变换(sin(a\*t), t)”返回“ $\frac{a}{a^2+t^2}$ ”；“拉普拉斯变换(sin(a\*t), a)”返回“ $\frac{t}{a^2+t^2}$ ”。

**Laplace(<Function>,<Variable>,<Variable>); 拉普拉斯变换(<函数>,<变量 1>,<变量 2>)**

返回给定两个变量函数的拉普拉斯变换。

**案例：**“拉普拉斯变换(sin(a\*t), a, t)”返回“ $\frac{1}{2t}$ ”；“拉普拉斯变换(sin(a\*t), t, a)”返回“ $\frac{1}{2a}$ ”。

运算区		运算区	
1	拉普拉斯变换[sin(a*t),a,t] → $\frac{1}{2t}$	1	拉普拉斯变换[sin(a*t),t,a] → $\frac{1}{2a}$

**注：**参见“拉普拉斯逆变换”指令。

## 2.20.11 InverseLaplace. 拉普拉斯逆变换

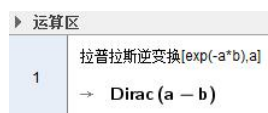
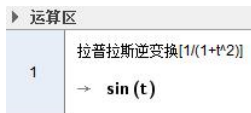
**InverseLaplace(<Function>); 拉普拉斯逆变换(<函数>)**。

返回给定函数的逆斯普拉斯变换。

**案例：**“拉普拉斯逆变换(1/(1+t^2))”返回“sin(t)”。

**InverseLaplace(<Function>,<Variable>); 拉普拉斯逆变换(<函数>,<变量>)**。

返回指定变量函数的逆斯普拉斯变换。



**案例：**“拉普拉斯逆变换( $\exp(-a*b)$ ,  $a$ )”返回“ $\text{Dirac}(a-b)$ ”；“拉普拉斯逆变换( $\exp(-a*b)$ ,  $b$ )”返回“ $\text{Dirac}(b-a)$ ”。

**InverseLaplace**( $\langle \text{Function} \rangle$ ,  $\langle \text{Variable} \rangle$ ,  $\langle \text{Variable} \rangle$ )；拉普拉斯逆变换( $\langle \text{函数} \rangle$ ,  $\langle \text{变量} 1 \rangle$ ,  $\langle \text{变量} 2 \rangle$ )。

返回指定两个变量函数的逆斯普拉斯变换。

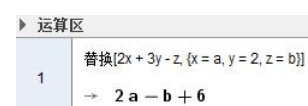
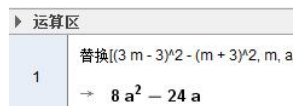
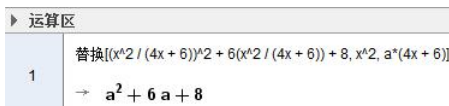
**注：**参见“拉普拉斯”指令。

## 2. 20. 12 Substitute. 替换

**Substitute**( $\langle \text{Expression} \rangle$ ,  $\langle \text{from} \rangle$ ,  $\langle \text{to} \rangle$ )；替换( $\langle \text{表达式} \rangle$ ,  $\langle \text{被替换对象} \rangle$ ,  $\langle \text{替换对象} \rangle$ )。

在表达式中使用替换对象替换被替换对象。

**案例：**“替换( $(x^2/(4x+6))^2+6(x^2/(4x+6))+8, x^2, a*(4x+6)$ )”得出“ $a^2+6a+8$ ”；“替换( $(3m-3)^2-(m+3)^2, m, a$ )”得出“ $8a^2-24a$ ”。替换后，合并了同类项。



**Substitute**( $\langle \text{Expression} \rangle$ ,  $\langle \text{Substitution List} \rangle$ )；替换( $\langle \text{表达式} \rangle$ ,  $\langle \text{赋值列表} \rangle$ )。

按照指定列表中的变量值替换表达式中的变量。

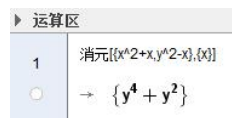
**案例：**“替换( $2x+3y-z, \{x=a, y=2, z=b\}$ )”得出“ $2a-b+6$ ”。

## 2. 20. 13 Eliminate. 消元

**Eliminate**( $\langle \text{List of Polynomials} \rangle$ ,  $\langle \text{List of Variables} \rangle$ )；消元( $\langle \text{多项式集} \rangle$ ,  $\langle \text{变量集} \rangle$ )

分析由多项式定义的代数方程组，并消除给定变量列表中所有变量后，计算等效系统。

**案例：**“消元( $\{x^2+x, y^2-x\}, \{x\}$ )”得出“ $y^4+y^2$ ”。(默认右边是 0 的方程组，将 2 式带入 1 式，结果就是“ $y^4+y^2$ ”)。



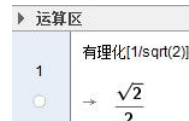
**注：**参见“分次字典序”指令。

## 2. 20. 14 Rationalize. 有理化

**Rationalize**( $\langle \text{Number} \rangle$ )；有理化( $\langle \text{数值} \rangle$ )。

为给定数值转换为分数形式，尽可能将分母有理化。

**案例：**“有理化(3.5)”得出 $\frac{7}{2}$ ；“有理化( $1/\sqrt{2}$ )”得出 $\frac{\sqrt{2}}{2}$ 。



**注：**参见“带分数”指令

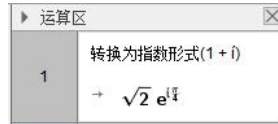


## 2. 20. 15 ToExponential. 转换为指数形式

**ToExponential**(**<Complex Number>**); 转换为指数形式(**<复数>**)。

把一个复数转换成指数形式。

如：“转换为指数形式(1+i)”得出如下结果。

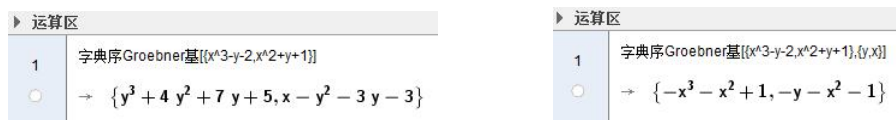


## 2. 20. 16 GroebnerLex. 字典序 Groebner 基

**GroebnerLex**(**<List of Polynomials>**); 字典序 Groebner 基(**<多项式列表>**)。

计算由字典序（也被看做是 lex、plex 或纯词汇序）变量决定的 Groebner 基多项式列表。

案例：“字典序 Groebner 基( $\{x^3-y-2, x^2+y+1\}$ )”得出如图结果。



**GroebnerLex**(**<List of Polynomials>**, **<List of Variables>**); 字典序 Groebner 基(**<多项式列表>**, **<变量列表>**)。

计算由指定字典序（也被看做是 lex、plex 或纯词汇序）变量决定的 Groebner 基多项式列表。

案例：“字典序 Groebner 基( $\{x^3-y-2, x^2+y+1\}, \{y, x\}$ )”得出如图结果。

注：参见“分次反字典序 Groebner 基”和“分次字典序 Groebner 基”指令。

### 3 中文指令名速查 (中文名称; 英文名称; 页码)

- 1、Delaunay 三角网; DelaunayTriangulation; 185
- 2、F 分布; FDistribution; 61
- 3、GeoGebra 指令; GeoGebraCommands; 32
- 4、Mad; Mad; 210
- 5、Mean; Mean; 210
- 6、Sstdev; Stdev; 212
- 7、Stdevp; Stdevp; 212
- 8、Sxx; Sxx; 212
- 9、Sxy; Sxy; 213
- 10、Syy; Syy; 213
- 11、t 分布; TDistribution; 62
- 12、T 检验; TTest; 213
- 13、Voronoi 图; Voronoi; 185
- 14、X 轴步长; AxisStepX; 32
- 15、Y 轴步长; AxisStepY; 32
- 16、爱尔朗分布; Erlang; 63
- 17、按钮; Button; 156
- 18、百分位数; Percentile; 214
- 19、半焦距; LinearEccentricity; 266
- 20、半径; Radius; 118
- 21、半圆; Semicircle; 266
- 22、棒图; StickGraph; 234
- 23、包络; envelope; 119
- 24、扁平列表; Flatten; 187
- 25、标准差; SD; 214
- 26、表格区; Spreadsheet; 41
- 27、表格文本; TableText; 248
- 28、并集; Union; 188
- 29、播放声音; PlaySound; 157
- 30、伯努利分布; Bernoulli; 64
- 31、泊松分布; Poisson; 64
- 32、泊松分布随机数; RandomPoisson; 65
- 33、部分分式; PartialFractions; 86
- 34、财务; Financial; 43
- 35、参数导数; ParametricDerivative; 86
- 36、残差图; ResidualPlot; 235
- 37、侧面; Side; 19
- 38、叉积; Cross; 45
- 39、插入; Insert; 188
- 40、差异; Difference; 119
- 41、超几何分布; HyperGeometric; 65
- 42、乘积; Product; 188
- 43、除法; Division; 45
- 44、垂线; PerpendicularLine (OrthogonalLine); 121
- 45、垂直平面; PerpendicularPlane; 19
- 46、次数; Degree; 86
- 47、代数; Algebra; 45
- 48、带分数; MixedNumber; 275
- 49、单比例 Z 估计; ZProportionEstimate; 215
- 50、单比例 Z 检验; ZProportionTest; 215
- 51、单均值 T 估计; TMeanEstimate; 215
- 52、单均值 Z 估计; ZMeanEstimate; 216
- 53、单均值 Z 检验; ZMeanTest; 216
- 54、单位法向; UnitPerpendicularVector (UnitOrthogonalVector); 257
- 55、单位矩阵; Identity; 258
- 56、单位向量; UnitVector; 259
- 57、单元格; Cell; 41
- 58、单元格区域数字列表; CellRange; 41
- 59、导出图片; ExportImage; 162
- 60、导数; Derivative; 87
- 61、底面; Ends; 20
- 62、第三四分位数; Q3; 216
- 63、第一四分位数; Q1; 217
- 64、点积; Dot; 46
- 65、点列; PointList; 189
- 66、点阵图; DotPlot; 235
- 67、迭代; Iteration; 88
- 68、迭代列表; IterationList; 89
- 69、顶点; Vertex; 123
- 70、定积分; NIntegral; 90
- 71、动态坐标; DynamicCoordinates; 32
- 72、对称; Reflect (Mirror); 153
- 73、对数拟合; FitLog; 217
- 74、对数正态分布; LogNormal; 67
- 75、对象; Object; 34
- 76、多边形; Polygon; 124
- 77、多项式函数; Polynomial; 90
- 78、多项式拟合; FitPoly; 218
- 79、二项分布; BinomialDist; 67

- 80、二项式系数; Binomial (BinomialCoefficient); 69
- 81、法向量; PerpendicularVector (OrthogonalVector); 259
- 82、反函数; NInvert; 91
- 83、方差; Variance; 218
- 84、方差分析; ANOVA; 219
- 85、方向向量; Direction; 125
- 86、仿射比  $\lambda$ ; AffineRatio; 126
- 87、放大; ZoomIn; 163
- 88、分次反字典序 Groebner 基; GroebnerDegRevLex; 275
- 89、分次字典序 Groebner 基; GroebnerLexDeg; 275
- 90、分母; Denominator; 91
- 91、分数文本; FractionText; 251
- 92、分子; Numerator; 92
- 93、附加副本; AttachCopyToView; 165
- 94、复数根; ComplexRoot; 92
- 95、复数解; CSolve; 276
- 96、复数解集; CSolutions; 276
- 97、复数域因式分解; CFactor; 276
- 98、复无理数域因式分解; CIFactor; 277
- 99、复选框; Checkbox; 164
- 100、复制自由对象; CopyFreeObject; 165
- 101、副半轴长; SemiMinorAxisLength (SecondAxisLength); 267
- 102、副轴; MinorAxis (SecondAxis); 267
- 103、赋值; SetValue; 164
- 104、伽玛分布; Gamma; 85
- 105、概率; Probability; 61
- 106、刚体多边形; RigidPolygon; 126
- 107、高度; Height; 20
- 108、根式文本; SurdText; 251
- 109、更新作图; UpdateConstruction; 166
- 110、工具图标; ToolImage; 34
- 111、公分母; CommonDenominator; 46
- 112、公式文本; FormulaText (LaTeX); 252
- 113、共轭直径; ConjugateDiameter (Diameter); 267
- 114、拐点; InflectionPoint (TurningPoint); 93
- 115、关系; Relation; 205
- 116、归一化; Normalize; 93
- 117、轨迹; Locus; 126
- 118、轨迹方程; LocusEquation; 127
- 119、海龟; Turtle; 166
- 120、函数; Function; 93
- 121、函数与微积分; Function; 86
- 122、行列式; Determinant; 263
- 123、行序; Row; 43
- 124、合并; Join; 190
- 125、横纵坐标乘积和; SigmaXY; 211
- 126、横坐标标准差; SDX; 219
- 127、横坐标平方和; SigmaXX; 210
- 128、横坐标平均数; MeanX; 219
- 129、后退; TurtleBack; 167
- 130、后一质数; NextPrime; 47
- 131、弧线; Arc; 128
- 132、互异; Unique; 190
- 133、滑动条; Slider; 167
- 134、化简; Simplify; 47
- 135、缓慢绘制; SlowPlot; 37
- 136、积分; Integral; 95
- 137、积分介于; IntegralBetween; 96
- 138、极限; Limit; 97
- 139、极线; Polar; 268
- 140、极值点; Extremum; 97
- 141、几何; Geometry; 118
- 142、几何变换; Transformation; 153
- 143、几何平均数; GeometricMean; 219
- 144、简化行梯阵式; ReducedRowEchelonForm; 261
- 145、渐近线; Asymptote; 98
- 146、交比; CrossRatio; 129
- 147、交点; Intersect; 129
- 148、交集; Intersection; 190
- 149、焦参数; Parameter; 268
- 150、焦点; Focus; 268
- 151、角点; Corner; 37
- 152、角度; Angle; 133
- 153、角平分线; AngleBisector; 131
- 154、脚本指令; ScriptingCommands; 156
- 155、阶梯图; StepGraph; 236
- 156、解常微分方程; SolveODE; 98
- 157、解常微分方程组; NSolveODE; 100
- 158、解集; Solutions; 47
- 159、解三次多项式; SolveCubic; 277
- 160、解析为函数; ParseToFunction; 167
- 161、解析为数; ParseToNumber; 168
- 162、近似解; NSolve; 48

163、近似解集; NSolutions; 49  
 164、近似数; Numeric; 277  
 165、茎叶图; StemPlot; 237  
 166、精确解; Solve; 50  
 167、矩形法则; RectangleSum; 102  
 168、矩阵的秩; MatrixRank; 261  
 169、距离; Distance; 132  
 170、均方根; RootMeanSquare; 219  
 171、均匀分布; Uniform; 69  
 172、均匀分布随机数; RandomUniform; 70  
 173、卡方分布; ChiSquared; 70  
 174、卡方检验; ChiSquaredTest; 220  
 175、开始记录; StartRecord; 168  
 176、柯西分布; Cauchy; 71  
 177、科学计数法; ScientificText; 253  
 178、可决系数 R 方; RSquare; 220  
 179、拉普拉斯变换; Laplace; 278  
 180、拉普拉斯逆变换; InverseLaplace; 278  
 181、棱柱; Prism; 20  
 182、棱锥; Pyramid; 21  
 183、离散数学; DiscreteMath; 185  
 184、离散随机数; RandomDiscrete; 72  
 185、离心率; Eccentricity; 269  
 186、利率; Rate; 43  
 187、连分式; ContinuedFraction; 253  
 188、列表; List; 187  
 189、列联表; ContingencyTable; 238  
 190、列名称; ColumnName; 41  
 191、列序; Column; 41  
 192、零点; Root; 102  
 193、零值点; Roots; 103  
 194、零值点列; RootList; 104  
 195、路径值; PathParameter; 104  
 196、逻辑; Logical; 205  
 197、逻辑分布; Logistic; 73  
 198、逻辑斯蒂曲线拟合; FitLogistic; 221  
 199、落笔; TurtleDown; 168  
 200、旅行商问题; TravelingSalesman; 185  
 201、每期付款额; Payment; 43  
 202、密切圆; OsculatingCircle; 105  
 203、幂函数拟合; FitPow; 221  
 204、面积; Area; 135  
 205、描点; Point; 135  
 206、名称; Name; 39  
 207、内点; PointIn; 136  
 208、内角; InteriorAngles; 137  
 209、内切圆; Incircle; 269  
 210、拟合曲线; Fit; 222  
 211、拟合直线 X; FitLineX; 222  
 212、拟合直线 Y; FitLine (FitLineY) ; 223  
 213、逆 F 分布; InverseFDistribution; 74  
 214、逆 T 分布; InverseTDistribution; 74  
 215、逆泊松分布; InversePoisson; 74  
 216、逆超几何分布; InverseHyperGeometric; 74  
 217、逆对数正态分布; InverseLogNormal; 75  
 218、逆二项分布; InverseBinomial; 75  
 219、逆反; Invert; 261  
 220、逆伽玛分布; InverseGamma; 77  
 221、逆卡方分布; InverseChiSquared; 75  
 222、逆柯西分布; InverseCauchy; 75  
 223、逆逻辑分布; InverseLogistic; 76  
 224、逆帕斯卡分布; InversePascal; 76  
 225、逆齐普夫分布; InverseZipf; 76  
 226、逆威布尔分布; InverseWeibull; 76  
 227、逆序排列; Reverse; 191  
 228、逆正态分布; InverseNormal; 77  
 229、逆指数分布; InverseExponential; 77  
 230、帕斯卡分布; Pascal; 77  
 231、抛物线; Parabola; 269  
 232、配对 T 检验; TTestPaired; 223  
 233、配方式; CompleteSquare; 50  
 234、频数表; FrequencyTable; 240  
 235、频数多边形; FrequencyPolygon; 242  
 236、频数列表; Frequency; 191  
 237、平均绝对偏差; Mad; 223  
 238、平均数; Mean; 224  
 239、平面; Plane; 22  
 240、平移; Translate; 154  
 241、平移视图; Pan; 168  
 242、平秩列表; TiedRank; 193  
 243、期数; Periods; 44  
 244、齐普夫分布; Zipf; 78  
 245、奇异值分解; SVD; 106  
 246、启动动画; StartAnimation; 168  
 247、启动日志; StartLogging; 169  
 248、前进; TurtleForward; 169  
 249、前一质数; PreviousPrime; 50  
 250、切变; Shear; 154

- 251、切线; Tangent; 137
- 252、球面; Sphere; 23
- 253、区间随机数; RandomBetween; 79
- 254、区域内最近点; ClosestPointRegion; 139
- 255、曲率; Curvature; 107
- 256、曲率向量; CurvatureVector; 107
- 257、曲面; Surface; 23
- 258、曲线; Curve (CurveCartesian); 108
- 259、取余; Mod; 51
- 260、取整; Div; 52
- 261、去除; Remove; 193
- 262、去除未定义对象; RemoveUndefined; 193
- 263、如果; If; 205
- 264、三次曲线; Cubic; 139
- 265、三点扇形; CircumcircularSector; 140
- 266、三点圆弧; CircumcircularArc; 140
- 267、三角曲线; TriangleCurve; 140
- 268、三角式合并; TrigCombine; 108
- 269、三角式化简; TrigSimplify; 109
- 270、三角式展开; TrigExpand; 110
- 271、三角形分布; Triangular; 80
- 272、三角形中心; TriangleCenter; 141
- 273、三维; 3D; 19
- 274、三线坐标点; Trilinear; 142
- 275、删除; Delete; 169
- 276、扇形; Sector; 142
- 277、上底; Top; 24
- 278、上和; UpperSum; 111
- 279、设置背景颜色; SetBackgroundColor; 169
- 280、设置标记; SetDecoration; 170
- 281、设置标签模式; SetLabelMode; 171
- 282、设置标题; SetCaption; 171
- 283、设置点径; SetPointSize; 171
- 284、设置点型; SetPointStyle; 172
- 285、设置动态颜色; SetDynamicColor; 172
- 286、设置对象锁定; SetFixed; 173
- 287、设置格局; SetPerspective; 173
- 288、设置跟踪; SetTrace; 174
- 289、设置工具提示模式; SetTooltipMode; 174
- 290、设置活动视图; SetActiveView; 175
- 291、设置可见性; SetVisibleInView; 175
- 292、设置视图方向; SetViewDirection; 175
- 293、设置填充; SetFilling; 176
- 294、设置图层; SetLayer; 176
- 295、设置细节级别; SetLevelOfDetail; 176
- 296、设置显示条件; SetConditionToShowObject; 176
- 297、设置线径; SetLineThickness; 176
- 298、设置线型; SetLineStyle; 177
- 299、设置颜色; SetColor; 177
- 300、设置种子; SetSeed; 178
- 301、设置转速; SetSpinSpeed; 178
- 302、设置作图步骤; SetConstructionStep; 39
- 303、设置坐标; SetCoords; 178
- 304、设置坐标轴比例; SetAxesRatio; 178
- 305、射线; Ray; 143
- 306、伸缩; Stretch; 155
- 307、升序排列; Sort; 194
- 308、生长曲线拟合; FitGrowth; 224
- 309、实数域因式分解; IFactor; 53
- 310、是否垂直; ArePerpendicular; 143
- 311、是否共点; AreConcurrent; 144
- 312、是否共线; AreCollinear; 144
- 313、是否共圆; AreConcyclic; 144
- 314、是否平行; AreParallel; 145
- 315、是否全等; AreCongruent; 145
- 316、是否为整数; IsInteger; 208
- 317、是否为质数; IsPrime; 53
- 318、是否相等; AreEqual; 145
- 319、是否相切; IsTangent; 146
- 320、是否已定义; IsDefined (Defined); 208
- 321、是否在区域内; IsInRegion; 208
- 322、输入框; InputBox (Textfield); 179
- 323、竖排文本; VerticalText; 254
- 324、数据函数; DataFunction; 179
- 325、数值导数; NDerivative; 112
- 326、双曲线; Hyperbola; 269
- 327、双样本比例 Z 估计; ZProportion2Estimate; 224
- 328、双样本比例 Z 检验; ZProportion2Test; 225
- 329、双样本均值 T 估计; TMean2Estimate; 225
- 330、双样本均值 Z 估计; ZMean2Estimate; 225
- 331、双样本均值 Z 检验; ZMean2Test; 226
- 332、双总体 T 检验; TTest2; 226
- 333、随机多项式; RandomPolynomial; 81
- 334、随机二项分布数; RandomBinomial; 81
- 335、随机内点; RandomPointIn; 82
- 336、随机排列; Shuffle; 227
- 337、随机元素; RandomElement; 195
- 338、缩小; ZoomOut; 179

339、索引; IndexOf; 195  
 340、抬笔; TurtleUp; 179  
 341、泰勒公式; TaylorPolynomial; 112  
 342、梯形法则; TrapezoidalSum; 112  
 343、提取; Take; 196  
 344、体积; Volume; 24  
 345、替换; Substitute; 279  
 346、填充单元格; FillCells; 42  
 347、填充行; FillRow; 42  
 348、填充列; FillColumn; 42  
 349、条件计数; CountIf; 209  
 350、条件子列; KeepIf; 209  
 351、条形图; BarChart; 243  
 352、调和平均数; HarmonicMean; 217  
 353、停止日志; StopLogging; 180  
 354、统计; Statistics; 210  
 355、统一码转换为文本; UnicodeToText; 254  
 356、统一码转换为字母; UnicodeToLetter; 255  
 357、凸包; ConvexHull; 186  
 358、图表; Chart; 234  
 359、椭圆; Ellipse; 270  
 360、威布尔分布; Weibull; 82  
 361、维度; Dimension; 262  
 362、未来值; FutureValue; 44  
 363、位似; Dilate(Enlarge); 155  
 364、文本; Text; 248  
 365、文本; Text; 255  
 366、文本转换为统一码; TextToUnicode; 256  
 367、无限长圆柱; InfiniteCylinder; 25  
 368、无限长圆锥; InfiniteCone; 25  
 369、误差平方和; SumSquaredErrors; 227  
 370、系数列表; Coefficients; 113  
 371、系统时间; GetTime; 180  
 372、下底; Bottom; 25  
 373、下和; LowerSum; 113  
 374、显示标签; ShowLabel; 180  
 375、显示图层; ShowLayer; 181  
 376、显示网格; ShowGrid; 181  
 377、显示坐标轴; ShowAxes; 181  
 378、现值; PresentValue; 44  
 379、线段; Segment; 146  
 380、相关系数; CorrelationCoefficient (PMCC);  
 227  
 381、相交路径; IntersectPath; 146  
 382、相交曲线; IntersectConic; 26  
 383、箱线图; BoxPlot; 244  
 384、向量; Vector; 263  
 385、向量与矩阵; Vector&Matrix; 257  
 386、消元; Eliminate; 279  
 387、协方差; Covariance; 228  
 388、斜率; Slope; 147  
 389、斜率场; SlopeField; 114  
 390、形心; Centroid; 147  
 391、序列; Sequence; 197  
 392、序数; Ordinal; 256  
 393、序数列表; OrdinalRank; 200  
 394、旋转; Rotate; 155  
 395、旋转文本; RotateText; 257  
 396、选定索引; SelectedIndex; 200  
 397、选定元素; SelectedElement; 200  
 398、选择; SelectObjects; 181  
 399、样本; Sample; 228  
 400、样本标准差; SampleSD; 229  
 401、样本点横坐标标准差; SampleSDX; 229  
 402、样本点纵坐标标准差; SampleSDY; 229  
 403、样本方差; SampleVariance; 230  
 404、样条曲线; Spline; 115  
 405、因式; Factors; 115  
 406、因式分解; Factor; 53  
 407、因数个数; Divisors; 54  
 408、因数和; DivisorsSum; 54  
 409、因数列表; DivisorsList; 55  
 410、隐藏图层; HideLayer; 182  
 411、隐函数拟合; FitImplicit; 230  
 412、隐式曲线; ImplicitCurve; 116  
 413、隐式微分; ImplicitDerivative; 117  
 414、应用矩阵; ApplyMatrix; 264  
 415、映射; Zip; 201  
 416、优化指令; Optimization; 265  
 417、有理化; Rationalize; 279  
 418、右边; RightSide; 55  
 419、右极限; LimitAbove; 117  
 420、右转; TurtleRight; 182  
 421、元素; Element; 201  
 422、圆弧; CircularArc; 148  
 423、圆扇形; CircularSector; 148  
 424、圆周; Circle; 271  
 425、圆周长; Circumference; 148

426、圆柱; Cylinder; 26  
 427、圆锥; Cone; 27  
 428、圆锥曲线; Conic; 266  
 429、圆锥曲线; Conic; 272  
 430、阅读文本; ReadText; 182  
 431、运行单击脚本; RunClickScript; 183  
 432、运行更新脚本; RunUpdateScript; 183  
 433、运算区专属指令; CASSpecific; 274  
 434、展开; Expand; 56  
 435、展开图; Net; 27  
 436、长度; Length; 119  
 437、折线; PolyLine; 148  
 438、正八面体; Octahedron; 28  
 439、正二十面体; Icosahedron; 29  
 440、正六面体; Cube; 29  
 441、正十二面体; Dodecahedron; 30  
 442、正四面体; Tetrahedron; 31  
 443、正态分布; Normal; 83  
 444、正态分布随机数; RandomNormal; 83  
 445、正态分位数图; NormalQuantilePlot; 245  
 446、正弦拟合; FitSin; 230  
 447、证明; Prove; 149  
 448、证明过程; ProveDetails; 149  
 449、执行; Execute; 183  
 450、直方图; Histogram; 246  
 451、直方图右; HistogramRight; 247  
 452、直线; Line; 150  
 453、指数分布; Exponential; 84  
 454、指数拟合; FitExp; 231  
 455、质因数; PrimeFactors; 56  
 456、秩相关系数; Spearman; 212  
 457、中垂面; PlaneBisector; 31  
 458、中垂线;  
 PerpendicularBisector(LineBisector); 150  
 459、中点; Midpoint; 151  
 460、中位数; Median; 231  
 461、中心; Center (Centre); 272  
 462、中心定位; CenterView; 184  
 463、众数; Mode; 232  
 464、重复; Repeat; 184  
 465、重命名; Rename; 184  
 466、重心; Barycenter; 151  
 467、周长; Perimeter; 152  
 468、轴线; Axes; 273  
 469、主半轴长; SemiMajorAxisLength  
 (FirstAxisLength); 273  
 470、主轴; MajorAxis (FirstAxis); 273  
 471、转换进制; ToBase; 56  
 472、转换为点; ToPoint; 39  
 473、转换为复数; ToComplex; 39  
 474、转换为极坐标形式; ToPolar; 40  
 475、转换为十进制; FromBase; 57  
 476、转换为指数形式; ToExponential; 280  
 477、转置; Transpose; 264  
 478、追加; Append; 202  
 479、准线; Directrix; 274  
 480、字典序 Groebner 基; GroebnerLex; 280  
 481、字母转换为统一码; LetterToUnicode; 257  
 482、总和; Sum; 232  
 483、纵坐标标准差; SDY; 233  
 484、纵坐标平方和; SigmaYY; 211  
 485、纵坐标平均数; MeanY; 234  
 486、组合数; nPr; 84  
 487、组限; Classes; 203  
 488、最大公约数; GCD; 57  
 489、最大值; Max; 58  
 490、最大值点; Maximize; 265  
 491、最短距离; ShortestDistance; 186  
 492、最后元素; Last; 203  
 493、最近点; ClosestPoint; 152  
 494、最前元素; First; 204  
 495、最小公倍数; LCM; 59  
 496、最小生成树; MinimumSpanningTree; 186  
 497、最小值; Min; 60  
 498、最小值点; Minimize; 265  
 499、左边; LeftSide; 61  
 500、左和; LeftSum; 118  
 501、左极限; LimitBelow; 118  
 502、左转; TurtleLeft; 184  
 503、作图步序; ConstructionStep; 40

## 4 英文指令名速查 (英文名称; 中文名称; 页码)

- 1、3D; 三维; 19
- 2、AffineRatio; 仿射比  $\lambda$ ; 126
- 3、Algebra; 代数; 45
- 4、Angle; 角度; 133
- 5、AngleBisector; 角平分线; 131
- 6、ANOVA; 方差分析; 219
- 7、Append; 追加; 202
- 8、ApplyMatrix; 应用矩阵; 264
- 9、Arc; 弧线; 128
- 10、Area; 面积; 135
- 11、AreCollinear; 是否共线; 144
- 12、AreConcurrent; 是否共点; 144
- 13、AreConcyclic; 是否共圆; 144
- 14、AreCongruent; 是否全等; 145
- 15、AreEqual; 是否相等; 145
- 16、AreParallel; 是否平行; 145
- 17、ArePerpendicular; 是否垂直; 143
- 18、Asymptote; 渐近线; 98
- 19、AttachCopyToView; 附加副本; 165
- 20、Axes; 轴线; 273
- 21、AxisStepX; X轴步长; 32
- 22、AxisStepY; Y轴步长; 32
- 23、BarChart; 条形图; 243
- 24、Barycenter; 重心; 151
- 25、Bernoulli; 伯努利分布; 64
- 26、Binomial (BinomialCoefficient); 二项式系数; 69
- 27、BinomialDist; 二项分布; 67
- 28、Bottom; 下底; 25
- 29、BoxPlot; 箱线图; 244
- 30、Button; 按钮; 156
- 31、CASSpecific; 运算区专属指令; 274
- 32、Cauchy; 柯西分布; 71
- 33、Cell; 单元格; 41
- 34、CellRange; 单元格区域数字列表; 41
- 35、Center (Centre); 中心; 272
- 36、CenterView; 中心定位; 184
- 37、Centroid; 形心; 147
- 38、CFactor; 复数域因式分解; 276
- 39、Chart; 图表; 234
- 40、Checkbox; 复选框; 164
- 41、ChiSquared; 卡方分布; 70
- 42、ChiSquaredTest; 卡方检验; 220
- 43、CIFactor; 复无理数域因式分解; 277
- 44、Circle; 圆周; 271
- 45、CircularArc; 圆弧; 148
- 46、CircularSector; 圆扇形; 148
- 47、CircumcircularArc; 三点圆弧; 140
- 48、CircumcircularSector; 三点扇形; 140
- 49、Circumference; 圆周长; 148
- 50、Classes; 组限; 203
- 51、ClosestPoint; 最近点; 152
- 52、ClosestPointRegion; 区域内最近点; 139
- 53、Coefficients; 系数列表; 113
- 54、Column; 列序; 41
- 55、ColumnName; 列名称; 41
- 56、CommonDenominator; 公分母; 46
- 57、CompleteSquare; 配方式; 50
- 58、ComplexRoot; 复数根; 92
- 59、Cone; 圆锥; 27
- 60、Conic; 圆锥曲线; 266
- 61、Conic; 圆锥曲线; 272
- 62、ConjugateDiameter (Diameter); 共轭直径; 267
- 63、ConstructionStep; 作图步序; 40
- 64、ContingencyTable; 列联表; 238
- 65、ContinuedFraction; 连分式; 253
- 66、ConvexHull; 凸包; 186
- 67、CopyFreeObject; 复制自由对象; 165
- 68、Corner; 角点; 37
- 69、CorrelationCoefficient (PMCC); 相关系数; 227
- 70、CountIf; 条件计数; 209
- 71、Covariance; 协方差; 228
- 72、Cross; 叉积; 45
- 73、CrossRatio; 交比; 129
- 74、CSolutions; 复数解集; 276
- 75、CSolve; 复数解; 276
- 76、Cube; 正六面体; 29
- 77、Cubic; 三次曲线; 139
- 78、Curvature; 曲率; 107
- 79、CurvatureVector; 曲率向量; 107



- 80、Curve (CurveCartesian); 曲线; 108
- 81、Cylinder; 圆柱; 26
- 82、DataFunction; 数据函数; 179
- 83、Degree; 次数; 86
- 84、DelaunayTriangulation; Delaunay 三角网; 185
- 85、Delete; 删除; 169
- 86、Denominator; 分母; 91
- 87、Derivative; 导数; 87
- 88、Determinant; 行列式; 263
- 89、Difference; 差异; 119
- 90、Dilate(Enlarge); 位似; 155
- 91、Dimension; 维度; 262
- 92、Direction; 方向向量; 125
- 93、Directrix; 准线; 274
- 94、DiscreteMath; 离散数学; 185
- 95、Distance; 距离; 132
- 96、Div; 取整; 52
- 97、Division; 除法; 45
- 98、Divisors; 因数个数; 54
- 99、DivisorsList; 因数列表; 55
- 100、DivisorsSum; 因数和; 54
- 101、Dodecahedron; 正十二面体; 30
- 102、Dot; 点积; 46
- 103、DotPlot; 点阵图; 235
- 104、DynamicCoordinates; 动态坐标; 32
- 105、Eccentricity; 离心率; 269
- 106、Element; 元素; 201
- 107、Eliminate; 消元; 279
- 108、Ellipse; 椭圆; 270
- 109、Ends; 底面; 20
- 110、envelope; 包络; 119
- 111、Erlang; 爱尔朗分布; 63
- 112、Execute; 执行; 183
- 113、Expand; 展开; 56
- 114、Exponential; 指数分布; 84
- 115、ExportImage; 导出图片; 162
- 116、Extremum; 极值点; 97
- 117、Factor; 因式分解; 53
- 118、Factors; 因式; 115
- 119、FDistribution; F 分布; 61
- 120、FillCells; 填充单元格; 42
- 121、FillColumn; 填充列; 42
- 122、FillRow; 填充行; 42
- 123、Financial; 财务; 43
- 124、First; 最前元素; 204
- 125、Fit; 拟合曲线; 222
- 126、FitExp; 指数拟合; 231
- 127、FitGrowth; 生长曲线拟合; 224
- 128、FitImplicit; 隐函数拟合; 230
- 129、FitLine (FitLineY); 拟合直线 Y; 223
- 130、FitLineX; 拟合直线 X; 222
- 131、FitLog; 对数拟合; 217
- 132、FitLogistic; 逻辑斯蒂曲线拟合; 221
- 133、FitPoly; 多项式拟合; 218
- 134、FitPow; 幂函数拟合; 221
- 135、FitSin; 正弦拟合; 230
- 136、Flatten; 扁平列表; 187
- 137、Focus; 焦点; 268
- 138、FormulaText (LaTeX); 公式文本; 252
- 139、FractionText; 分数文本; 251
- 140、Frequency; 频数列表; 191
- 141、FrequencyPolygon; 频数多边形; 242
- 142、FrequencyTable; 频数表; 240
- 143、FromBase; 转换为十进制; 57
- 144、Function; 函数与微积分; 86
- 145、Function; 函数; 93
- 146、FutureValue; 未来值; 44
- 147、Gamma; 伽玛分布; 85
- 148、GCD; 最大公约数; 57
- 149、GeoGebraCommands; GeoGebra 指令; 32
- 150、GeometricMean; 几何平均数; 219
- 151、Geometry; 几何; 118
- 152、GetTime; 系统时间; 180
- 153、GroebnerDegRevLex; 分次反字典序 Groebner 基; 275
- 154、GroebnerLex; 字典序 Groebner 基; 280
- 155、GroebnerLexDeg; 分次字典序 Groebner 基; 275
- 156、HarmonicMean; 调和平均数; 217
- 157、Height; 高度; 20
- 158、HideLayer; 隐藏图层; 182
- 159、Histogram; 直方图; 246
- 160、HistogramRight; 直方图右; 247
- 161、Hyperbola; 双曲线; 269
- 162、HyperGeometric; 超几何分布; 65
- 163、Icosahedron; 正二十面体; 29
- 164、Identity; 单位矩阵; 258
- 165、If; 如果; 205
- 166、IFactor; 实数域因式分解; 53
- 167、ImplicitCurve; 隐式曲线; 116
- 168、ImplicitDerivative; 隐式微分; 117
- 169、Incircle; 内切圆; 269

- 170、IndexOf; 索引; 195
- 171、InfiniteCone; 无限长圆锥; 25
- 172、InfiniteCylinder; 无限长圆柱; 25
- 173、InflectionPoint (TurningPoint); 拐点; 93
- 174、InputBox (Textfield); 输入框; 179
- 175、Insert; 插入; 188
- 176、Integral; 积分; 95
- 177、IntegralBetween; 积分介于; 96
- 178、InteriorAngles; 内角; 137
- 179、Intersect; 交点; 129
- 180、IntersectConic; 相交曲线; 26
- 181、Intersection; 交集; 190
- 182、IntersectPath; 相交路径; 146
- 183、InverseBinomial; 逆二项分布; 75
- 184、InverseCauchy; 逆柯西分布; 75
- 185、InverseChiSquared; 逆卡方分布; 75
- 186、InverseExponential; 逆指数分布; 77
- 187、InverseFDistribution; 逆 F 分布; 74
- 188、InverseGamma; 逆伽玛分布; 77
- 189、InverseHyperGeometric; 逆超几何分布; 74
- 190、InverseLaplace; 拉普拉斯逆变换; 278
- 191、InverseLogistic; 逆逻辑分布; 76
- 192、InverseLogNormal; 逆对数正态分布; 75
- 193、InverseNormal; 逆正态分布; 77
- 194、InversePascal; 逆帕斯卡分布; 76
- 195、InversePoisson; 逆泊松分布; 74
- 196、InverseTDistribution; 逆 T 分布; 74
- 197、InverseWeibull; 逆威布尔分布; 76
- 198、InverseZipf; 逆齐普夫分布; 76
- 199、Invert; 逆反; 261
- 200、IsDefined (Defined); 是否已定义; 208
- 201、IsInRegion; 是否在区域内; 208
- 202、IsInteger; 是否为整数; 208
- 203、IsPrime; 是否为质数; 53
- 204、IsTangent; 是否相切; 146
- 205、Iteration; 迭代; 88
- 206、IterationList; 迭代列表; 89
- 207、Join; 合并; 190
- 208、KeepIf; 条件子列; 209
- 209、Laplace; 拉普拉斯变换; 278
- 210、Last; 最后元素; 203
- 211、LCM; 最小公倍数; 59
- 212、LeftSide; 左边; 61
- 213、LeftSum; 左和; 118
- 214、Length; 长度; 119
- 215、LetterToUnicode; 字母转换为统一码; 257
- 216、Limit; 极限; 97
- 217、LimitAbove; 右极限; 117
- 218、LimitBelow; 左极限; 118
- 219、Line; 直线; 150
- 220、LinearEccentricity; 半焦距; 266
- 221、List; 列表; 187
- 222、Locus; 轨迹; 126
- 223、LocusEquation; 轨迹方程; 127
- 224、Logical; 逻辑; 205
- 225、Logistic; 逻辑分布; 73
- 226、LogNormal; 对数正态分布; 67
- 227、LowerSum; 下和; 113
- 228、Mad; Mad; 210
- 229、Mad; 平均绝对偏差; 223
- 230、MajorAxis (FirstAxis); 主轴; 273
- 231、MatrixRank; 矩阵的秩; 261
- 232、Max; 最大值; 58
- 233、Maximize; 最大值点; 265
- 234、Mean; Mean; 210
- 235、Mean; 平均数; 224
- 236、MeanX; 横坐标平均数; 219
- 237、MeanY; 纵坐标平均数; 234
- 238、Median; 中位数; 231
- 239、Midpoint; 中点; 151
- 240、Min; 最小值; 60
- 241、Minimize; 最小值点; 265
- 242、MinimumSpanningTree; 最小生成树; 186
- 243、MinorAxis (SecondAxis); 副轴; 267
- 244、MixedNumber; 带分数; 275
- 245、Mod; 取余; 51
- 246、Mode; 众数; 232
- 247、Name; 名称; 39
- 248、NDerivative; 数值导数; 112
- 249、Net; 展开图; 27
- 250、NextPrime; 后一质数; 47
- 251、NIntegral; 定积分; 90
- 252、NInvert; 反函数; 91
- 253、Normal; 正态分布; 83
- 254、Normalize; 归一化; 93
- 255、NormalQuantilePlot; 正态分位数图; 245
- 256、nPr; 组合数; 84
- 257、NSolutions; 近似解集; 49
- 258、NSolve; 近似解; 48
- 259、NSolveODE; 解常微分方程组; 100
- 260、Numerator; 分子; 92
- 261、Numeric; 近似数; 277

- 262、Object; 对象; 34
- 263、Octahedron; 正八面体; 28
- 264、Optimization; 优化指令; 265
- 265、Ordinal; 序数; 256
- 266、OrdinalRank; 序数列表; 200
- 267、OsculatingCircle; 密切圆; 105
- 268、Pan; 平移视图; 168
- 269、Parabola; 抛物线; 269
- 270、Parameter; 焦参数; 268
- 271、ParametricDerivative; 参数导数; 86
- 272、ParseToFunction; 解析为函数; 167
- 273、ParseToNumber; 解析为数; 168
- 274、PartialFractions; 部分分式; 86
- 275、Pascal; 帕斯卡分布; 77
- 276、PathParameter; 路径值; 104
- 277、Payment; 每期付款额; 43
- 278、Percentile; 百分位数; 214
- 279、Perimeter; 周长; 152
- 280、Periods; 期数; 44
- 281、PerpendicularBisector(LineBisector); 中垂线; 150
- 282、PerpendicularLine (OrthogonalLine); 垂线; 121
- 283、PerpendicularPlane; 垂直平面; 19
- 284、PerpendicularVector (OrthogonalVector); 法向量; 259
- 285、Plane; 平面; 22
- 286、PlaneBisector; 中垂面; 31
- 287、PlaySound; 播放声音; 157
- 288、Point; 描点; 135
- 289、PointIn; 内点; 136
- 290、PointList; 点列; 189
- 291、Poisson; 泊松分布; 64
- 292、Polar; 极线; 268
- 293、Polygon; 多边形; 124
- 294、PolyLine; 折线; 148
- 295、Polynomial; 多项式函数; 90
- 296、PresentValue; 现值; 44
- 297、PreviousPrime; 前一质数; 50
- 298、PrimeFactors; 质因数; 56
- 299、Prism; 棱柱; 20
- 300、Probability; 概率; 61
- 301、Product; 乘积; 188
- 302、Prove; 证明; 149
- 303、ProveDetails; 证明过程; 149
- 304、Pyramid; 棱锥; 21
- 305、Q1; 第一四分位数; 217
- 306、Q3; 第三四分位数; 216
- 307、Radius; 半径; 118
- 308、RandomBetween; 区间随机数; 79
- 309、RandomBinomial; 随机二项分布数; 81
- 310、RandomDiscrete; 离散随机数; 72
- 311、RandomElement; 随机元素; 195
- 312、RandomNormal; 正态分布随机数; 83
- 313、RandomPointIn; 随机内点; 82
- 314、RandomPoisson; 泊松分布随机数; 65
- 315、RandomPolynomial; 随机多项式; 81
- 316、RandomUniform; 均匀分布随机数; 70
- 317、Rate; 利率; 43
- 318、Rationalize; 有理化; 279
- 319、Ray; 射线; 143
- 320、ReadText; 阅读文本; 182
- 321、RectangleSum; 矩形法则; 102
- 322、ReducedRowEchelonForm; 简化行梯阵式; 261
- 323、Reflect (Mirror); 对称; 153
- 324、Relation; 关系; 205
- 325、Remove; 去除; 193
- 326、RemoveUndefined; 去除未定义对象; 193
- 327、Rename; 重命名; 184
- 328、Repeat; 重复; 184
- 329、ResidualPlot; 残差图; 235
- 330、Reverse; 逆序排列; 191
- 331、RightSide; 右边; 55
- 332、RigidPolygon; 刚体多边形; 126
- 333、Root; 零点; 102
- 334、RootList; 零值点列; 104
- 335、RootMeanSquare; 均方根; 219
- 336、Roots; 零值点; 103
- 337、Rotate; 旋转; 155
- 338、RotateText; 旋转文本; 257
- 339、Row; 行序; 43
- 340、RSquare; 可决系数 R 方; 220
- 341、RunClickScript; 运行单击脚本; 183
- 342、RunUpdateScript; 运行更新脚本; 183
- 343、Sample; 样本; 228
- 344、SampleSD; 样本标准差; 229
- 345、SampleSDX; 样本点横坐标标准差; 229
- 346、SampleSDY; 样本点纵坐标标准差; 229
- 347、SampleVariance; 样本方差; 230
- 348、ScientificText; 科学计数法; 253
- 349、ScriptingCommands; 脚本指令; 156
- 350、SD; 标准差; 214

- 351、SDX; 横坐标标准差; 219
- 352、SDY; 纵坐标标准差; 233
- 353、Sector; 扇形; 142
- 354、Segment; 线段; 146
- 355、SelectedElement; 选定元素; 200
- 356、SelectedIndex; 选定索引; 200
- 357、SelectObjects; 选择; 181
- 358、Semicircle; 半圆; 266
- 359、SemiMajorAxisLength (FirstAxisLength); 主半轴长; 273
- 360、SemiMinorAxisLength (SecondAxisLength); 副半轴长; 267
- 361、Sequence; 序列; 197
- 362、SetActiveView; 设置活动视图; 175
- 363、SetAxesRatio; 设置坐标轴比例; 178
- 364、SetBackgroundColor; 设置背景颜色; 169
- 365、SetCaption; 设置标题; 171
- 366、SetColor; 设置颜色; 177
- 367、SetConditionToShowObject; 设置显示条件; 176
- 368、SetConstructionStep; 设置作图步骤; 39
- 369、SetCoords; 设置坐标; 178
- 370、SetDecoration; 设置标记; 170
- 371、SetDynamicColor; 设置动态颜色; 172
- 372、SetFilling; 设置填充; 176
- 373、SetFixed; 设置对象锁定; 173
- 374、SetLabelMode; 设置标签模式; 171
- 375、SetLayer; 设置图层; 176
- 376、SetLevelOfDetail; 设置细节级别; 176
- 377、SetLineStyle; 设置线型; 177
- 378、SetLineThickness; 设置线径; 176
- 379、SetPerspective; 设置格局; 173
- 380、SetPointSize; 设置点径; 171
- 381、SetPointStyle; 设置点型; 172
- 382、SetSeed; 设置种子; 178
- 383、SetSpinSpeed; 设置转速; 178
- 384、SetTooltipMode; 设置工具提示模式; 174
- 385、SetTrace; 设置跟踪; 174
- 386、SetValue; 赋值; 164
- 387、SetViewDirection; 设置视图方向; 175
- 388、SetVisibleInView; 设置可见性; 175
- 389、Shear; 切变; 154
- 390、ShortestDistance; 最短距离; 186
- 391、ShowAxes; 显示坐标轴; 181
- 392、ShowGrid; 显示网格; 181
- 393、ShowLabel; 显示标签; 180
- 394、ShowLayer; 显示图层; 181
- 395、Shuffle; 随机排列; 227
- 396、Side; 侧面; 19
- 397、SigmaXX; 横坐标平方和; 210
- 398、SigmaXY; 横纵坐标乘积和; 211
- 399、SigmaYY; 纵坐标平方和; 211
- 400、Simplify; 化简; 47
- 401、Slider; 滑动条; 167
- 402、Slope; 斜率; 147
- 403、SlopeField; 斜率场; 114
- 404、SlowPlot; 缓慢绘制; 37
- 405、Solutions; 解集; 47
- 406、Solve; 精确解; 50
- 407、SolveCubic; 解三次多项式; 277
- 408、SolveODE; 解常微分方程; 98
- 409、Sort; 升序排列; 194
- 410、Spearman; 秩相关系数; 212
- 411、Sphere; 球面; 23
- 412、Spline; 样条曲线; 115
- 413、Spreadsheet; 表格区; 41
- 414、StartAnimation; 启动动画; 168
- 415、StartLogging; 启动日志; 169
- 416、StartRecord; 开始记录; 168
- 417、Statistics; 统计; 210
- 418、Stdev; Sstdev; 212
- 419、Stdevp; Stdevp; 212
- 420、StemPlot; 茎叶图; 237
- 421、StepGraph; 阶梯图; 236
- 422、StickGraph; 棒图; 234
- 423、StopLogging; 停止日志; 180
- 424、Stretch; 伸缩; 155
- 425、Substitute; 替换; 279
- 426、Sum; 总和; 232
- 427、SumSquaredErrors; 误差平方和; 227
- 428、SurdText; 根式文本; 251
- 429、Surface; 曲面; 23
- 430、SVD; 奇异值分解; 106
- 431、Sxx; Sxx; 212
- 432、Sxy; Sxy; 213
- 433、Syy; Syy; 213
- 434、TableText; 表格文本; 248
- 435、Take; 提取; 196
- 436、Tangent; 切线; 137
- 437、TaylorPolynomial; 泰勒公式; 112
- 438、TDistribution; t分布; 62
- 439、Tetrahedron; 正四面体; 31

- 440、Text; 文本; 248
- 441、Text; 文本; 255
- 442、TextToUnicode; 文本转换为统一码; 256
- 443、TiedRank; 平秩列表; 193
- 444、TMean2Estimate; 双样本均值 T 估计; 225
- 445、TMeanEstimate; 单均值 T 估计; 215
- 446、ToBase; 转换进制; 56
- 447、ToComplex; 转换为复数; 39
- 448、ToExponential; 转换为指数形式; 280
- 449、ToolImage; 工具图标; 34
- 450、Top; 上底; 24
- 451、ToPoint; 转换为点; 39
- 452、ToPolar; 转换为极坐标形式; 40
- 453、Transformation; 几何变换; 153
- 454、Translate; 平移; 154
- 455、Transpose; 转置; 264
- 456、TrapezoidalSum; 梯形法则; 112
- 457、TravelingSalesman; 旅行商问题; 185
- 458、TriangleCenter; 三角形中心; 141
- 459、TriangleCurve; 三角曲线; 140
- 460、Triangular; 三角形分布; 80
- 461、TrigCombine; 三角式合并; 108
- 462、TrigExpand; 三角式展开; 110
- 463、TrigSimplify; 三角式化简; 109
- 464、Trilinear; 三线坐标点; 142
- 465、TTest; T 检验; 213
- 466、TTest2; 双总体 T 检验; 226
- 467、TTestPaired; 配对 T 检验; 223
- 468、Turtle; 海龟; 166
- 469、TurtleBack; 后退; 167
- 470、TurtleDown; 落笔; 168
- 471、TurtleForward; 前进; 169
- 472、TurtleLeft; 左转; 184
- 473、TurtleRight; 右转; 182
- 474、TurtleUp; 抬笔; 179
- 475、UnicodeToLetter; 统一码转换为字母; 255
- 476、UnicodeToText; 统一码转换为文本; 254
- 477、Uniform; 均匀分布; 69
- 478、Union; 并集; 188
- 479、Unique; 互异; 190
- 480、UnitPerpendicularVector  
(UnitOrthogonalVector); 单位法向; 257
- 481、UnitVector; 单位向量; 259
- 482、UpdateConstruction; 更新作图; 166
- 483、UpperSum; 上和; 111
- 484、Variance; 方差; 218
- 485、Vector; 向量; 263
- 486、Vector&Matrix; 向量与矩阵; 257
- 487、Vertex; 顶点; 123
- 488、VerticalText; 竖排文本; 254
- 489、Volume; 体积; 24
- 490、Voronoi; Voronoi 图; 185
- 491、Weibull; 威布尔分布; 82
- 492、Zip; 映射; 201
- 493、Zipf; 齐普夫分布; 78
- 494、ZMean2Estimate; 双样本均值 Z 估计; 225
- 495、ZMean2Test; 双样本均值 Z 检验; 226
- 496、ZMeanEstimate; 单均值 Z 估计; 216
- 497、ZMeanTest; 单均值 Z 检验; 216
- 498、ZoomIn; 放大; 163
- 499、ZoomOut; 缩小; 179
- 500、ZProportion2Estimate; 双样本比例 Z 估计;  
224
- 501、ZProportion2Test; 双样本比例 Z 检验; 225
- 502、ZProportionEstimate; 单比例 Z 估计; 215
- 503、ZProportionTest; 单比例 Z 检验; 215

## 5 软件指令列表截图

打开 GeoGebra 软件后，点击“指令栏”最右侧的指令帮助开关（小问号）会出现软件“指令帮助”。在 GeoGebra 中，“函数”和“命令”统称为“指令”，列放在一起，鼠标双击其中一个指令，指令携带输入规范会自动跳入“指令栏”内。当在运算区被打开或执行过运算区指令时，在最下方会出现“运算区指令”行。

“所有指令”是其下方 19 个指令（不含“运算区指令”）汇总，方便全局查找某个指令。但“运算区指令”中某些专属指令，没有列示在“所有指令”中。本节不对“所有指令”进行截图列示。

在 5.0.527.0 版本中，数学函数有 54 个，指令有 483 个，其中“运算区指令”有 189 个，其中多数在“指令栏”也可用，只是格式和返回的结果略微不同。只有 16 个指令是仅限于运算区的，已经在“2.20”章节列示。“运算区指令”也截图列示。其它指令（包括“数学函数”）顺序列示。

**指令帮助**

- 数学函数
- + 所有指令
- + 3D
- + GeoGebra
- + 表格
- + 财务
- + 代数
- + 概率
- + 函数与微积分
- + 几何
- + 几何变换
- + 脚本
- + 离散数学
- + 列表
- + 逻辑
- + 统计
- + 图表
- + 文本
- + 向量与矩阵
- + 优化指令
- + 圆锥曲线
- + 运算区

**数学函数**

sqrt(x)	cbrt(x)
abs(x)	sgn(x)
arg(x)	conjugate(x)
floor(x)	ceil(x)
round(x)	log(b,x)
exp(x)	ln(x)
lg(x)	ld(x)
sin(x)	asin(x)
cos(x)	acos(x)
tan(x)	atan(x)
sinh(x)	asinh(x)
cosh(x)	acosh(x)
tanh(x)	atanh(x)
sec(x)	sech(x)
cosec(x)	cosech(x)
cot(x)	coth(x)
asind(x)	acosd(x)
atand(x)	atan2(y, x)
erf(x)	gamma(x)
beta(a, b)	gamma(a, x)
beta(a, b, x)	gammaRegularized(a, x)
betaRegularized(a, b, x)	psi(x)
polyGamma(m, x)	nroot(x, n)
fractionalPart(x)	real(x)
imaginary(x)	sinIntegral(x)
cosIntegral(x)	explntegral(x)
random()	zeta(x)

**3D**

- 侧面
- 垂直平面
- 底面
- 高度
- 棱柱
- 棱锥
- 平面
- 球面
- 曲面
- 上底
- 体积
- 无限长圆柱
- 无限长圆锥
- 下底
- 相交曲线
- 圆柱
- 圆锥
- 展开图
- 正八面体
- 正二十面体
- 正六面体
- 正十二面体
- 正四面体
- 中垂面
- GeoGebra**
- x轴步长
- y轴步长
- 动态坐标
- 对象
- 工具图标
- 缓慢绘制
- 角点
- 名称
- 设置作图步骤
- 转换为点
- 转换为复数

- 转换为复数
- 转换为极坐标形式
- 作图步骤
- **表格**
  - 单元格
  - 单元格区域数字列表
  - 列名称
  - 列序
  - 填充单元格
  - 填充列
  - 填充行
  - 行序
- **财务**
  - 利率
  - 每期付款额
  - 期数
  - 未来值
  - 现值
- **代数**
  - ShowSteps
  - 叉积
  - 除法
  - 点积
  - 顶点式
  - 公分母
  - 后一质数
  - 化简
  - 解集
  - 近似解
  - 近似解集
  - 精确解
  - 前一质数
  - 取余
  - 取整
  - 实数域因式分解
  - 是否为质数
  - 因式分解
  - 因数个数
  - 因数和

- 因数和
- 因数列表
- 右边
- 展开
- 质因数
- 转换进制
- 转换为十进制
- 最大公约数
- 最大值
- 最小公倍数
- 最小值
- 左边
- **概率**
  - F分布
  - t分布
  - 爱尔朗分布
  - 伯努利分布
  - 泊松分布
  - 泊松分布随机数
  - 超几何分布
  - 对数正态分布
  - 二项分布
  - 二项式系数
  - 均匀分布
  - 均匀分布随机数
  - 卡方分布
  - 柯西分布
  - 离散随机数
  - 逻辑分布
  - 逆F分布
  - 逆t分布
  - 逆泊松分布
  - 逆超几何分布
  - 逆对数正态分布
  - 逆二项分布
  - 逆卡方分布
  - 逆柯西分布
  - 逆逻辑分布

- 逆逻辑分布
- 逆帕斯卡分布
- 逆齐普夫分布
- 逆威布尔分布
- 逆正态分布
- 逆指数分布
- 逆伽玛分布
- 帕斯卡分布
- 齐普夫分布
- 区间随机数
- 三角形分布
- 随机多项式
- 随机二项分布数
- 随机内点
- 威布尔分布
- 正态分布
- 正态分布随机数
- 指数分布
- 组合数
- 伽玛分布
- **函数与微积分**
  - 部分分式
  - 参数导数
  - 次数
  - 导数
  - 迭代
  - 迭代列表
  - 定积分
  - 多项式函数
  - 反函数
  - 分母
  - 分子
  - 复数根
  - 拐点
  - 归一化
  - 函数
  - 积分
  - 积分介于

- 积分介于
- 极限
- 极值点
- 渐近线
- 解常微分方程
- 解常微分方程组
- 矩形法则
- 零点
- 零值点
- 零值点列
- 路径值
- 密切圆
- 奇异值分解
- 曲率
- 曲率向量
- 曲线
- 三角式合并
- 三角式化简
- 三角式展开
- 上和
- 数值导数
- 泰勒公式
- 梯形法则
- 系数列表
- 下和
- 斜率场
- 样条曲线
- 因式
- 隐式曲线
- 隐式微分
- 右极限
- 左和
- 左极限
- **几何**
  - 半径
  - 包络
  - 差异
  - 长度
  - 垂线

- 垂线
- 顶点
- 多边形
- 方向向量
- 仿射比 $\lambda$
- 刚体多边形
- 轨迹
- 轨迹方程
- 弧线
- 交比
- 交点
- 角度
- 角平分线
- 距离
- 面积
- 描点
- 内点
- 内角
- 切线
- 区域内最近点
- 三次曲线
- 三点扇形
- 三点圆弧
- 三角曲线
- 三角形中心
- 三线坐标点
- 扇形
- 射线
- 是否垂直
- 是否共点
- 是否共线
- 是否共圆
- 是否平行
- 是否全等
- 是否相等
- 是否相切
- 线段
- 相交路径
- 斜率

- 斜率
- 形心
- 圆弧
- 圆扇形
- 圆周长
- 折线
- 证明
- 证明过程
- 直线
- 中垂线
- 中点
- 重心
- 周长
- 最近点
- 几何变换**
- 对称
- 平移
- 切变
- 伸缩
- 位似
- 旋转
- 脚本**
- 按钮
- 播放声音
- 导出图片
- 放大
- 赋值
- 复选框
- 复制自由对象
- 附加副本
- 更新作图
- 海龟
- 后退
- 滑动条
- 解析为函数
- 解析为数
- 开始记录
- 落笔

- 落笔
- 平移视图
- 启动动画
- 启动日志
- 前进
- 删除
- 设置背景颜色
- 设置标记
- 设置标签模式
- 设置标题
- 设置点径
- 设置点型
- 设置动态颜色
- 设置对象锁定
- 设置格局
- 设置跟踪
- 设置工具提示模式
- 设置活动视图
- 设置可见性
- 设置视图方向
- 设置填充
- 设置图层
- 设置细节级别
- 设置显示条件
- 设置线径
- 设置线型
- 设置颜色
- 设置种子
- 设置转速
- 设置坐标
- 设置坐标轴比例
- 输入框
- 数据函数
- 缩小
- 抬笔
- 停止日志
- 系统时间
- 显示标签
- 显示图层

- 显示图层
- 显示网格
- 显示坐标轴
- 选择
- 隐藏图层
- 右转
- 阅读文本
- 运行单击脚本
- 运行更新脚本
- 执行
- 中心定位
- 重复
- 重命名
- 左转
- 离散数学**
- Delaunay三角网
- Voronoi图
- 旅行商问题
- 凸包
- 最短距离
- 最小生成树
- 列表**
- 扁平列表
- 并集
- 插入
- 乘积
- 点列
- 合并
- 互异
- 交集
- 逆序排列
- 频数列表
- 平秩列表
- 去除
- 去除未定义对象
- 升序排列
- 随机元素
- 索引



- 索引
- 提取
- 序列
- 序数列表
- 选定索引
- 选定元素
- 映射
- 元素
- 追加
- 组限
- 最后元素
- 最前元素
- 逻辑**
- 关系
- 如果
- 是否为整数
- 是否已定义
- 是否在区域内
- 条件计数
- 条件子列
- 统计**
- mad
- mean
- SigmaXX
- SigmaXY
- SigmaYY
- Spearman秩相关系数
- stdev
- stdevp
- Sxx
- Sxy
- Syy
- t检验
- 百分位数
- 标准差
- 单比例z估计
- 单比例z检验
- 单均值估计
- 单均值t估计
- 单均值z估计
- 单均值z检验
- 第三四分位数
- 第一四分位数
- 调和平均数
- 对数拟合
- 多项式拟合
- 方差
- 方差分析
- 横坐标标准差
- 横坐标平均数
- 几何平均数
- 均方根
- 卡方检验
- 可决系数R方
- 逻辑斯蒂曲线拟合
- 幂函数拟合
- 拟合曲线
- 拟合直线X
- 拟合直线Y
- 配对样本t检验
- 平均绝对偏差
- 平均数
- 生长曲线拟合
- 双样本比例z估计
- 双样本比例z检验
- 双样本均值t估计
- 双样本均值z估计
- 双样本均值z检验
- 双总体t检验
- 随机排列
- 误差平方和
- 相关系数
- 协方差
- 样本
- 样本标准差
- 样本点横坐标标准差
- 样本点纵坐标标准差
- 样本点纵坐标标准差
- 样本方差
- 隐函数拟合
- 正弦拟合
- 指数拟合
- 中位数
- 众数
- 总和
- 纵坐标标准差
- 纵坐标平均数
- 图表**
- 棒图
- 残差图
- 点阵图
- 阶梯图
- 茎叶图
- 列联表
- 频数表
- 频数多边形
- 条形图
- 箱线图
- 正态分位数图
- 直方图
- 直方图右和
- 文本**
- 表格文本
- 分数文本
- 根式文本
- 公式文本
- 科学记数法
- 连分式
- 竖排文本
- 统一码转换为文本
- 统一码转换为字母
- 文本
- 文本转换为统一码
- 序数
- 旋转文本
- 旋转文本
- 字母转换为统一码
- 向量与矩阵**
- 单位法向量
- 单位矩阵
- 单位向量
- 法向量
- 简化行梯阵式
- 矩阵的秩
- 逆反
- 维度
- 向量
- 行列式
- 应用矩阵
- 转置
- 优化指令**
- 最大值点
- 最小值点
- 圆锥曲线**
- 半焦距
- 半圆
- 副半轴长
- 副轴
- 共轭直径
- 极线
- 焦参数
- 焦点
- 离心率
- 内切圆
- 抛物线
- 双曲线
- 椭圆
- 圆周
- 圆锥曲线
- 中心
- 轴线
- 主半轴长
- 主轴
- 准线

准线	单元格区域数字列表	极值点
<b>运算区</b>	导数	假定
ApproximateSolution	点积	简化行梯阵式
AssumeInteger	点列	交点
BinomialCoefficient	迭代	交集
CorrectSolution	迭代列表	角度
ExpandOnly	顶点式	角平分线
F分布	定积分	解常微分方程
mean	对数拟合	解集
nPr	对象	解三次多项式
Regroup	多项式函数	近似解
SolveODEPoint	多项式拟合	近似解集
stdev	二项分布	近似数
stdevp	二项式系数	精确解
t分布	法向量	矩阵的秩
半径	方差	距离
扁平列表	分次反字典序Groebner基	均匀分布随机数
标准差	分次字典序Groebner基	卡方分布
并集	分母	柯西分布
泊松分布	分子	科学记数法
泊松分布随机数	复数根	拉普拉斯变换
部分分式	复数解	拉普拉斯逆变换
叉积	复数解集	列序
长度	复数域因式分解	零点
超几何分布	复无理数域因式分解	零值点列
乘积	复制自由对象	幂函数拟合
除法	公分母	面积
垂线	拐点	描点
次数	函数	逆二项分布
带分数	合并	逆反
单位法向量	后一质数	逆序排列
单位矩阵	互异	逆正态分布
单位向量	化简	帕斯卡分布
单元格	积分	平均数
单元格区域数字列表	积分介于	平面
	极限	平移

平移	行列式	
奇异值分解	行序	
齐普夫分布	序列	
前一质数	样本	
切线	样本标准差	
区间随机数	样本方差	
取余	因式	
取整	因式分解	
如果	因数个数	
三角式合并	因数和	
三角式化简	因数列表	
三角式展开	隐式微分	
实数域因式分解	映射	
是否为质数	有理化	
双曲线	右边	
随机多项式	右极限	
随机二项分布数	元素	
随机排列	圆周	
随机元素	圆周长	
索引	圆锥曲线	
泰勒公式	约当对角化	
特征向量	展开	
特征值	正态分布	
提取	正态分布随机数	
替换	直线	
条件计数	指数分布	
椭圆	指数拟合	
威布尔分布	质因数	
维度	中垂线	
系数列表	中点	
线段	中位数	
向量	中心	
消元	转换进制	
协方差	转换为点	
行列式	转换为复数	
	转换为极坐标形式	
	转换为十进制	转换为极坐标形式
	转换为指数形式	转换为十进制
	转置	转换为指数形式
	追加	转置
	字典序Groebner基	追加
	总和	字典序Groebner基
	组合数	总和
	最大公约数	组合数
	最大值	最大公约数
	最后元素	最大值
	最前元素	最后元素
	最小公倍数	最前元素
	最小值	最小公倍数
	左边	最小值
	左极限	左边
	伽玛分布	左极限
		伽玛分布

本软件每天都在优化，新版本的更新变化，敬请参见官网以下地址，了解版本更新说明：

[https://wiki.geogebra.org/en/Reference:Changelog\\_6.0](https://wiki.geogebra.org/en/Reference:Changelog_6.0)