

GeoGebra 经典 5

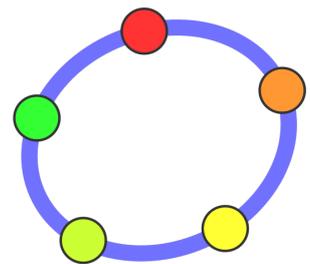
中文指令帮助 2024 先行版

作者：赵林

组织：GeoGebra 学园

时间：November 15, 2024

GeoGebra 版本：5.2.849



献给热爱 GeoGebra 的你

前言

在 2018 年我开始学习 GeoGebra 时，常常参考学习唐家军老师的《指令汇编》手册。从这本手册中，我获取了大量宝贵的作图知识和经验。这本手册对学习 GeoGebra 的教师和学习者来说都是必备的入门指南，它涵盖的内容丰富全面。每次阅读，我都能从中汲取新的灵感和知识，无论从什么角度阅读，它总是令人耳目一新。掌握和理解这本手册的主要内容的人，可以说已经达到 GeoGebra 入门或者精通作图的水平。

然而，该书的最新一次更新是在 2019 年，至今已有五年。期间，许多指令的中文名称已发生变化，GeoGebra 也新增了一些指令和功能。为了与 GeoGebra 的新版本保持同步，并帮助新手更好地学习，我尝试对手册进行了一些修改和更新。

笔者采用了 LaTeX 语言重新编排了指令汇编，使手册更加美观。内容主要参考了官方网站的手册和唐老师的汇编，增加了新增的指令，并纠正了原文中的一些细小的可能的错误和过时的用法。同时，笔者也分享了一些个人的心得、使用体验和示例。本手册中的大部分示例都来自与官网给出的例子，少部分是笔者追加的，部分翻译笔者认为翻译的不好，在本手册中笔者对其进行了修改，修改的地方在手册中都有说明。由于官网引用的部分内容来自与被墙的英文维基百科，故笔者把这些链接改成了 mathworld.wolfram.com 上的相关链接。

由于笔者自身能力有限，修改后的手册不可避免地存在一些错误，还请读者多多包涵。此外，由于笔者我的写作水平一般，遣词造句上可能不够完善，数学规范和标点符号也可能存在瑕疵，也恳请读者谅解。通过编写本手册，笔者对许多指令有了新的理解，也提升了自己的水平。

如果读者对本手册的内容有任何疑问，欢迎通过邮件与笔者联系：136547444@qq.com。本书不用于商业目的，仅供 GeoGebra 爱好者学习使用，切勿上传到百度文库等流氓网站上牟利。

再次感谢唐老师和其他 GeoGebra 先行者们的贡献。

赵林

2024 年 8 月

目录

第 1 章 函数与运算	1
1.1 常用数学符号和基本运算	1
1.1.1 常用符号	1
1.1.2 基本运算	1
1.2 布尔运算	1
1.3 数学函数	2
第 2 章 指令	5
2.1 3d 三维指令	5
2.1.1 Side. 侧面	5
2.1.2 OrthogonalPlane. 垂直平面	6
2.1.3 Ends. 底面	6
2.1.4 Height. 高度	6
2.1.5 Prism. 棱柱	6
2.1.6 Pyramid. 棱锥	7
2.1.7 Plane. 平面	7
2.1.8 Sphere. 球面	8
2.1.9 Surface. 曲面	8
2.1.10 Top. 上底	10
2.1.11 Volume. 体积	10
2.1.12 CylinderInfinite. 无限长圆柱	10
2.1.13 ConeInfinite. 无限长圆锥	10
2.1.14 Bottom. 下底	10
2.1.15 IntersectConic. 相交曲线	11
2.1.16 Cylinder. 圆柱	11
2.1.17 Cone. 圆锥	11
2.1.18 Net. 展开图	12
2.1.19 Octahedron. 正八面体	12
2.1.20 Icosahedron. 正二十面体	12
2.1.21 Cube. 正六面体	12
2.1.22 Dodecahedron. 正十二面体	12
2.1.23 Tetrahedron. 正四面体	13
2.1.24 PlaneBisector. 中垂面	13
2.2 GeoGebra 指令	14
2.2.1 CASLoaded	14
2.2.2 AxisStepx.x 轴步长	14
2.2.3 AxisStepY.y 轴步长	14
2.2.4 DynamicCoordinates. 动态坐标	14
2.2.5 Object. 对象	15
2.2.6 ToolImage. 工具图标	15
2.2.7 SlowPlot. 缓慢绘制	16

2.2.8	Corner. 角点	17
2.2.9	Name. 名称	18
2.2.10	SetConstructionStep. 设置作图步骤	19
2.2.11	ToPoint. 转换为点	19
2.2.12	ToComplex. 转换为复数	19
2.2.13	ToPolar. 转换为极坐标形式	19
2.2.14	ConstructionStep. 作图步序	19
2.3	Financial. 财务	19
2.3.1	Rate. 利率	20
2.3.2	Payment. 每期付款额	20
2.3.3	Periods. 期数	20
2.3.4	FutureValue. 未来值	20
2.3.5	PresentValue. 现值	20
2.4	代数指令	20
2.4.1	Cross. 叉积	20
2.4.2	Division. 除法	21
2.4.3	Dot. 点积	21
2.4.4	CompleteSquare. 顶点式	21
2.4.5	CommonDenominator. 公分母	22
2.4.6	NextPrime. 后一质数	22
2.4.7	Simplify. 化简	22
2.4.8	Solutions. 解集	22
2.4.9	NSolve. 近似解	23
2.4.10	Nsolutions. 近似解集	23
2.4.11	Solve. 精确解	23
2.4.12	PreviousPrime. 前一质数	23
2.4.13	Div. 商式	23
2.4.14	IFactor. 实数域因式分解	24
2.4.15	IsPrime. 是否为质数	24
2.4.16	IsFactored. 是否已分解	24
2.4.17	Factor. 因式分解	25
2.4.18	Divisors. 因数个数	25
2.4.19	DivisorsSum. 因数和	25
2.4.20	DivisorsList. 因数列表	25
2.4.21	RightSide. 右边	25
2.4.22	Mod. 余式	26
2.4.23	Expand. 展开	27
2.4.24	PrimeFactors. 质因数	27
2.4.25	ToBase. 转换进制	27
2.4.26	FromBase. 转换为十进制	27
2.4.27	GCD. 最大公约数	27
2.4.28	Max. 最大值	28
2.4.29	LCM. 最小公倍数	29
2.4.30	Min. 最小值	29
2.4.31	LeftSide. 左边	29

2.5 Probability. 概率	29
2.5.1 FDistribution.F 分布	29
2.5.2 nCr	30
2.5.3 TDistribution.t 分布	30
2.5.4 Erlang. 爱尔朗分布	30
2.5.5 BetaDist. 贝塔分布	31
2.5.6 Bernoulli. 伯努利分布	31
2.5.7 Poisson. 泊松分布	31
2.5.8 RandomPoisson. 泊松分布随机数	32
2.5.9 HyperGeometric. 超几何分布	32
2.5.10 LogNormal. 对数正态分布	32
2.5.11 BinomialDist. 二项分布	33
2.5.12 Binomial. 二项式系数	33
2.5.13 Uniform. 均匀分布	33
2.5.14 RandomUniform. 均匀分布随机数	33
2.5.15 ChiSquared. 卡方分布	34
2.5.16 Cauchy. 柯西分布	34
2.5.17 RandomDiscrete. 离散随机数	34
2.5.18 Logistic. 逻辑分布	34
2.5.19 InverseFDistribution. 逆 F 分布	35
2.5.20 InverseTDistribution. 逆 t 分布	35
2.5.21 InverseBeta. 逆贝塔分布	35
2.5.22 InversePoisson. 逆泊松分布	35
2.5.23 InverseHyperGeometric. 逆超几何分布	35
2.5.24 InverseLogNormal. 逆对数正态分布	35
2.5.25 InverseBinomial. 逆二项分布	36
2.5.26 InverseBinomialMinimumTrials. 逆二项最小实验	36
2.5.27 InverseChiSquared. 逆卡方分布	36
2.5.28 InverseCauchy. 逆柯西分布	36
2.5.29 InverseLogistic. 逆逻辑分布	36
2.5.30 InversePascal. 逆帕斯卡分布	36
2.5.31 InverseZipf. 逆齐普夫分布	37
2.5.32 InverseWeibull. 逆威布尔分布	37
2.5.33 InverseNormal. 逆正态分布	37
2.5.34 InverseExponential. 逆指数分布	37
2.5.35 InverseGamma. 逆伽玛分布	37
2.5.36 Pascal. 帕斯卡分布	37
2.5.37 Zipf. 齐普夫分布	38
2.5.38 RandomBetween. 区间随机数	38
2.5.39 Triangular. 三角形分布	38
2.5.40 RandomPolynomial. 随机多项式	38
2.5.41 RandomBinomial. 随机二项分布数	39
2.5.42 RandomPointIn. 随机内点	39
2.5.43 Weibull. 威布尔分布	39
2.5.44 Normal. 正态分布	39

2.5.45 RandomNormal. 正态分布随机数	40
2.5.46 Exponential. 指数分布	40
2.5.47 Gamma. 伽玛分布	40
2.6 Function 函数与微积分	40
2.6.1 PartialFractions. 部分分式	40
2.6.2 ParametricDerivative. 参数导数	40
2.6.3 Derivative. 导数	41
2.6.4 Iteration. 迭代	41
2.6.5 IterationList. 迭代列表	42
2.6.6 NIntegral. 定积分	42
2.6.7 Degree. 多项式次数	43
2.6.8 Polynomial. 多项式函数	43
2.6.9 NInvert. 反函数	43
2.6.10 Denominator. 分母	44
2.6.11 Numerator. 分子	44
2.6.12 ComplexRoot. 复数根	45
2.6.13 TurningPoint. 拐点	45
2.6.14 Normalize. 归一化	45
2.6.15 Function. 函数	45
2.6.16 Integral. 积分	46
2.6.17 IntegralBetween. 积分介于	47
2.6.18 Limit. 极限	47
2.6.19 Extremum. 极值点	47
2.6.20 Asymptote. 渐近线	48
2.6.21 SolveODE. 解常微分方程	48
2.6.22 NSolveODE. 解常微分方程组	50
2.6.23 RectangleSum. 矩形法则	50
2.6.24 RemovableDiscontinuity. 可去间断点	51
2.6.25 Root. 零点	51
2.6.26 Roots. 零值点	52
2.6.27 RootList. 零值点列	52
2.6.28 PathParameter. 路径值	52
2.6.29 OsculatingCircle. 密切圆	53
2.6.30 SVD. 奇异值分解	53
2.6.31 PlotSolve. 求解绘图	54
2.6.32 Curvature. 曲率	54
2.6.33 CurvatureVector. 曲率向量	54
2.6.34 CurveCartesian. 曲线	54
2.6.35 TrigCombine. 三角合并	55
2.6.36 TrigSimplify. 三角化简	55
2.6.37 TrigExpand. 三角展开	55
2.6.38 UpperSum. 上和	55
2.6.39 IsVertexForm. 是否为顶点式	56
2.6.40 NDerivative. 数值导数	56
2.6.41 TaylorSeries. 泰勒公式	56

2.6.42 TrapezoidalSum. 梯形法则	56
2.6.43 Coefficients. 系数列表	56
2.6.44 LowerSum. 下和	57
2.6.45 SlopeField. 斜率场	57
2.6.46 Spline. 样条曲线	57
2.6.47 Factors. 因式	58
2.6.48 ImplicitCurve. 隐式曲线	58
2.6.49 ImplicitDerivative. 隐式微分	59
2.6.50 LimitAbove. 右极限	59
2.6.51 LeftSum. 左和	59
2.6.52 LimitBelow. 左极限	59
2.7 Geometry. 几何	60
2.7.1 PenStroke	60
2.7.2 Radius. 半径	60
2.7.3 Envelope. 包络	60
2.7.4 Difference. 差集	60
2.7.5 Length. 长度	60
2.7.6 OrthogonalLine. 垂线	61
2.7.7 Vertex. 顶点	61
2.7.8 Polygon. 多边形	62
2.7.9 Direction. 方向向量	62
2.7.10 AffineRatio. 仿射比 λ	63
2.7.11 RigidPolygon. 刚体多边形	63
2.7.12 Locus. 轨迹	63
2.7.13 LocusEquation. 轨迹方程	64
2.7.14 Arc. 弧线	65
2.7.15 CrossRatio. 交比	65
2.7.16 Intersect. 交点	66
2.7.17 Angle. 角度	66
2.7.18 AngularBisector. 角平分线	68
2.7.19 Distance. 距离	68
2.7.20 Type. 类型	69
2.7.21 Area. 面积	69
2.7.22 Point. 描点	69
2.7.23 PointIn. 内点	70
2.7.24 InteriorAngles. 内角	70
2.7.25 Tangent. 切线	70
2.7.26 ClosestPointRegion. 区域内最近点	72
2.7.27 Cubic. 三次曲线	72
2.7.28 TriangleCurve. 三角曲线	72
2.7.29 TriangleCenter. 三角形中心	73
2.7.30 Trilinear. 三线坐标点	73
2.7.31 Sector. 扇形	74
2.7.32 Ray. 射线	74
2.7.33 ArePerpendicular. 是否垂直	74

2.7.34	AreConcurrent. 是否共点	75
2.7.35	AreCollinear. 是否共线	75
2.7.36	AreConcyclic. 是否共圆	75
2.7.37	AreParallel. 是否平行	75
2.7.38	AreCongruent. 是否全等	75
2.7.39	AreEqual. 是否相等	75
2.7.40	IsTangent. 是否相切	76
2.7.41	CircumcircleArc. 外接圆弧	76
2.7.42	CircumcircleSector. 外接圆扇形	76
2.7.43	Segment. 线段	76
2.7.44	IntersectPath. 相交路径	76
2.7.45	Slope. 斜率	77
2.7.46	Centroid. 形心	77
2.7.47	CircleArc. 圆弧	77
2.7.48	CircleSector. 圆扇形	78
2.7.49	PolyLine. 折线	78
2.7.50	Prove. 证明	78
2.7.51	ProveDetails. 证明过程	78
2.7.52	Line. 直线	79
2.7.53	LineBisector. 中垂线	79
2.7.54	Midpoint. 中点	79
2.7.55	Barycenter. 重心	80
2.7.56	Perimeter. 周长	80
2.7.57	Circumference. 周界长	80
2.7.58	ClosestPoint. 最近点	80
2.8	Transformation. 几何变换	81
2.8.1	Mirror. 对称	81
2.8.2	Translate. 平移	81
2.8.3	Shear. 切变	81
2.8.4	Stretch. 伸缩	82
2.8.5	Dilate. 位似	82
2.8.6	Rotate. 旋转	83
2.9	script. 脚本	83
2.9.1	Button. 按钮	83
2.9.2	PlaySound. 播放声音	84
2.9.3	ExportImage. 导出图片	85
2.9.4	ZoomIn. 放大	86
2.9.5	SetValue. 赋值	87
2.9.6	Checkbox. 复选框	88
2.9.7	CopyFreeObject. 复制自由对象	89
2.9.8	AttachCopyToView. 附加副本	89
2.9.9	UpdateConstruction. 更新作图	89
2.9.10	Turtle. 海龟	89
2.9.11	TurtleBack. 后退	89
2.9.12	Slider. 滑动条	90

2.9.13 ParseToFunction. 解析为函数	90
2.9.14 ParseToNumber. 解析为数	91
2.9.15 StartRecord. 开始记录	91
2.9.16 TurtleDown. 落笔	91
2.9.17 StartAnimation. 启动动画	91
2.9.18 TurtleForward. 前进	92
2.9.19 Delete. 删除	92
2.9.20 SetBackgroundColor. 设置背景颜色	92
2.9.21 SetDecoration. 设置标记	93
2.9.22 SetLabelMode. 设置标签模式	93
2.9.23 SetCaption. 设置标题	94
2.9.24 SetPointSize. 设置点径	94
2.9.25 SetPointStyle. 设置点型	95
2.9.26 SetDynamicColor. 设置动态颜色	95
2.9.27 SetPerspective. 设置格局	95
2.9.28 SetTrace. 设置跟踪	96
2.9.29 SetTooltipMode. 设置工具提示模式	96
2.9.30 SetFixed. 设置固定	97
2.9.31 SetActiveView. 设置活动视图	97
2.9.32 SetVisibleInView. 设置可见性	97
2.9.33 SetViewDirection. 设置视图方向	97
2.9.34 SetFilling. 设置填充	98
2.9.35 SetLayer. 设置图层	98
2.9.36 SetImage. 设置图片	98
2.9.37 SetLevelOfDetail. 设置细节级别	98
2.9.38 SetConditionToShowObject. 设置显示条件	99
2.9.39 SetLineThickness. 设置线径	99
2.9.40 SetLineStyle. 设置线型	99
2.9.41 SetColor. 设置颜色	99
2.9.42 SetSeed. 设置种子	100
2.9.43 SetSpinSpeed. 设置转速	100
2.9.44 SetCoords. 设置坐标	100
2.9.45 SetAxesRatio. 设置坐标轴比例	100
2.9.46 Textfield. 输入框	101
2.9.47 DataFunction. 数据函数	101
2.9.48 ZoomOut. 缩小	101
2.9.49 TurtleUp. 抬笔	101
2.9.50 GetTime. 系统时间	101
2.9.51 ShowLabel. 显示标签	102
2.9.52 ShowLayer. 显示图层	102
2.9.53 ShowGrid. 显示网格	102
2.9.54 ShowAxes. 显示坐标轴	102
2.9.55 SelectObjects. 选择	103
2.9.56 Pan. 移动视图	103
2.9.57 HideLayer. 隐藏图层	103

2.9.58	TurtleRight. 右转	103
2.9.59	ReadText. 阅读文本	103
2.9.60	RunClickScript. 运行单击脚本	104
2.9.61	RunUpdateScript. 运行更新脚本	104
2.9.62	Execute. 执行	104
2.9.63	CenterView. 中心定位	105
2.9.64	Repeat. 重复	105
2.9.65	Rename. 重命名	105
2.9.66	TurtleLeft. 左转	105
2.10	离散数学	105
2.10.1	DelauneyTriangulation.Delaunay 三角网	105
2.10.2	Voronoi.Voronoi 图	106
2.10.3	TravelingSalesman. 旅行商问题	106
2.10.4	ConvexHull. 凸包	107
2.10.5	ShortestDistance. 最短距离	107
2.10.6	MinimumSpanningTree. 最小生成树	107
2.11	列表	107
2.11.1	Flatten. 扁平列表	107
2.11.2	Union. 并集	107
2.11.3	Insert. 插入	108
2.11.4	Product. 乘积	108
2.11.5	PointList. 点列	109
2.11.6	Join. 合并	109
2.11.7	Unique. 互异	110
2.11.8	Intersection. 交集	110
2.11.9	Reverse. 逆序排列	110
2.11.10	Frequency. 频数列表	110
2.11.11	TiedRank. 平秩列表	112
2.11.12	Remove. 去除	112
2.11.13	RemoveUndefined. 去除未定义对象	112
2.11.14	Sort. 升序排列	112
2.11.15	RandomElement. 随机元素	113
2.11.16	IndexOf. 索引	113
2.11.17	Take. 提取	114
2.11.18	Sequence. 序列	114
2.11.19	OrdinalRank. 序数列表	115
2.11.20	SelectedElement. 选定的元素	115
2.11.21	SelectedIndex. 选定索引	116
2.11.22	Zip. 映射	116
2.11.23	Element. 元素	116
2.11.24	Append. 追加	117
2.11.25	Classes. 组限	117
2.11.26	Last. 最后元素	117
2.11.27	First. 最前元素	118
2.12	逻辑	118

2.12.1 Relation. 关系	118
2.12.2 If. 如果	119
2.12.3 KeepIf. 条件子列	120
2.13 数据区	120
2.13.1 Cell. 单元格	120
2.13.2 CellRange. 单元格区域数值列表	120
2.13.3 ColumnName. 列名称	121
2.13.4 Column. 列序	121
2.13.5 FillCells. 填充单元格	121
2.13.6 FillColumn. 填充列	122
2.13.7 FillRow. 填充行	122
2.13.8 Row. 行序	122
2.14 统计	122
2.14.1 Mad.mad	122
2.14.2 Mean	122
2.14.3 SigmaXX	123
2.14.4 SigmaXY	123
2.14.5 SigmaYY	124
2.14.6 Spearman.Spearman 秩相关系数	124
2.14.7 stdev	124
2.14.8 stdevp	125
2.14.9 Sxx	125
2.14.10 Sxy	125
2.14.11 Syy	126
2.14.12 TTest.t 检验	126
2.14.13 Percentile. 百分位数	126
2.14.14 ZProportionEstimate. 单比例 z 估计	127
2.14.15 ZProportionTest. 单比例 z 检验	127
2.14.16 TMeanEstimate. 单均值 t 估计	127
2.14.17 ZMeanEstimate. 单均值 z 估计	128
2.14.18 ZMeanTest. 单均值 z 检验	128
2.14.19 Q3. 第三四分位数	128
2.14.20 Q1. 第一四分位数	129
2.14.21 HarmonicMean. 调和平均数	129
2.14.22 FitLog. 对数拟合	130
2.14.23 FitPoly. 多项式拟合	130
2.14.24 Variance. 方差	130
2.14.25 ANOVA. 方差分析	131
2.14.26 SDX. 横坐标标准差	131
2.14.27 MeanX. 横坐标平均数	131
2.14.28 GeometricMean. 几何平均数	131
2.14.29 RootMeanSquare. 均方根	131
2.14.30 ChiSquaredTest. 卡方检验	132
2.14.31 RSquare. 可决系数 R 方	132
2.14.32 FitLogistic. 逻辑斯蒂曲线拟合	132

2.14.33	FitPow. 幂函数拟合	133
2.14.34	Fit. 拟合曲线	133
2.14.35	FitLineX. 拟合直线 X	133
2.14.36	FitLineY. 拟合直线 Y	133
2.14.37	TTestPaired. 配对样本 t 检验	134
2.14.38	FitGrowth. 生长曲线拟合	134
2.14.39	ZProportion2Estimate. 双样本比例 z 估计	134
2.14.40	ZProportion2Test. 双样本比例 z 检验	134
2.14.41	TMean2Estimate. 双样本均值 t 估计	135
2.14.42	ZMean2Estimate. 双样本均值 z 估计	135
2.14.43	ZMean2Test. 双样本均值 z 检验	135
2.14.44	TTest2. 双总体 t 检验	136
2.14.45	Shuffle. 随机排列	136
2.14.46	SumSquaredErrors. 误差平方和	136
2.14.47	CorrelationCoefficient. 相关系数	136
2.14.48	Covariance. 协方差	137
2.14.49	Sample. 样本	137
2.14.50	SampleSDX. 样本点横坐标标准差	138
2.14.51	SampleSDY. 样本点纵坐标标准差	138
2.14.52	SampleVariance. 样本方差	138
2.14.53	FitImplicit. 隐函数拟合	138
2.14.54	FitSin. 正弦拟合	138
2.14.55	FitExp. 指数拟合	139
2.14.56	Median. 中位数	139
2.14.57	Mode. 众数	139
2.14.58	Sum. 总和	139
2.14.59	SDY. 纵坐标标准差	141
2.14.60	MeanY. 纵坐标平均数	141
2.15	图表	141
2.15.1	StickGraph. 棒图	141
2.15.2	ResidualPlot. 残差图	142
2.15.3	DotPlot. 点阵图	143
2.15.4	StepGraph. 阶梯图	143
2.15.5	StemPlot. 茎叶图	144
2.15.6	ContingencyTable. 列联表	144
2.15.7	FrequencyTable. 频数表	145
2.15.8	FrequencyPolygon. 频数多边形	145
2.15.9	PieChart. 扇形图	146
2.15.10	BarChart. 条形图	146
2.15.11	BoxPlot. 箱形图	147
2.15.12	LineGraph. 折线图	148
2.15.13	NormalQuantilePlot. 正态分位数图	148
2.15.14	Histogram. 直方图	149
2.15.15	HistogramRight. 直方图右和	149
2.16	文本	150

2.16.1	TableText. 表格文本	150
2.16.2	Split. 拆分	151
2.16.3	FractionText. 分数文本	151
2.16.4	SurdText. 根式文本	152
2.16.5	LaTeX. 公式文本	152
2.16.6	ScientificText. 科学记数法	153
2.16.7	ContinuedFraction. 连分式	153
2.16.8	VerticalText. 竖排文本	154
2.16.9	ReplaceAll. 替换所有	154
2.16.10	UnicodeToText. 统一码转换为文本	155
2.16.11	UnicodeToLetter. 统一码转换为字母	155
2.16.12	Text. 文本	155
2.16.13	TextToUnicode. 文本转换为统一码	155
2.16.14	Ordinal. 序数	156
2.16.15	RotateText. 旋转文本	156
2.16.16	LetterToUnicode. 字母转换为统一码	156
2.17	向量与矩阵	157
2.17.1	UnitOrthogonalVector. 单位法向量	157
2.17.2	Identity. 单位矩阵	157
2.17.3	UnitVector. 单位向量	158
2.17.4	OrthogonalVector. 法向量	158
2.17.5	ReducedRowEchelonForm. 简化行梯阵式	158
2.17.6	MatrixRank. 矩阵的秩	158
2.17.7	Invert. 逆反	158
2.17.8	Dimension. 维度	158
2.17.9	Vector. 向量	159
2.17.10	Determinant. 行列式	159
2.17.11	ApplyMatrix. 应用矩阵	159
2.17.12	Transpose. 转置	160
2.18	优化指令	161
2.18.1	Maximize. 最大值点	161
2.18.2	Minimize. 最小值点	161
2.19	圆锥曲线	161
2.19.1	Excentricity. 半焦距	161
2.19.2	Semicircle. 半圆	162
2.19.3	SecondAxisLength. 副半轴长	162
2.19.4	SecondAxis. 副轴	162
2.19.5	Diameter. 共轭直径	162
2.19.6	Polar. 极线	162
2.19.7	Parameter. 焦参数	163
2.19.8	Focus. 焦点	163
2.19.9	Eccentricity. 离心率	163
2.19.10	Incircle. 内切圆	163
2.19.11	Parabola. 抛物线	163
2.19.12	Hyperbola. 双曲线	163

2.19.13 Ellipse. 椭圆	164
2.19.14 Circle. 圆周	164
2.19.15 Conic. 圆锥曲线	165
2.19.16 Center. 中心	165
2.19.17 Axes. 轴线	165
2.19.18 FirstAxisLength. 主半轴长	165
2.19.19 FirstAxis. 主轴	165
2.19.20 Directrix. 准线	166
第 3 章 附录	167
3.1 中英文查询	167
3.2 部分通用的英文名	167
3.3 修改指令名和指令提示	167

第 1 章 函数与运算

1.1 常用数学符号和基本运算

1.1.1 常用符号

1. e : 欧拉数/自然常数. 键盘输入 (同时按): $\text{Alt}+e$.
2. i : 虚数单位. 键盘输入: $\text{Alt}+i$.
3. π : 圆周率. 键盘输入: $\text{Alt}+p$ 或输入 “pi” .
4. $^\circ$: 度数单位符号. 键盘输入: $\text{Alt}+o$ 或者使用中文输入法输入 “°” .

1.1.2 基本运算

运算	符号	说明
加法	+	用于文本的运算时起联结作用, 例如 $1+^{\circ}a^{\circ}$, 返回的结果为“1a”
减法	-	
乘法/数量积	* 或空格	用作数量积时参考内积指令
叉积	\otimes	详见叉积指令
乘方/幂	^ 或者 **	键盘上按 $\text{Alt}+$ 数字可以得到整数次幂, 例如按 $\text{Alt}+3$ 可以得到 3 次幂
阶乘	!	
除法	/	
求横坐标	x(< 点 >)	x(< 直线 >) 返回直线方程中 x 的系数 A
求纵坐标	y(< 点 >)	y(< 直线 >) 返回直线方程中 y 的系数 B
求竖坐标	z(< 点 >)	z(< 直线 >) 返回直线方程中常数项 C

示例:

```
x(2x+y+1=0) >>>2
```

1.2 布尔运算

布尔运算的结果称为布尔值, 布尔值只有 2 个值即 true 和 false , 在布尔值参与其他运算时, true 和 false 分别被作为 1 和 0 使用. 比如指令: $1+\text{true}$ 返回的结果为 2. 下表为常用的布尔运算.

运算	菜单选择	键盘输入
是否等于	$\stackrel{?}{=}$	$==$
不等于	\neq	$!=$
小于	$<$	$<$
大于	$>$	$>$
小于等于	\leq	$<=$
大于等于	\geq	$>=$
且	\wedge	$\& \&$
或	\vee	$\ $ (即两根竖线)
非	\neg	! 或者 $<>$
属于	\in	
平行于	//	
垂直于	\perp	

注意: 想快速输入这些符号, 除了点开指令栏右边的 α 符号用鼠标点击输入 (见图 1.1) 和使用键盘输入以外, 也可以将这些符号输入到输入法的自定义短语或者词库中.



图 1.1

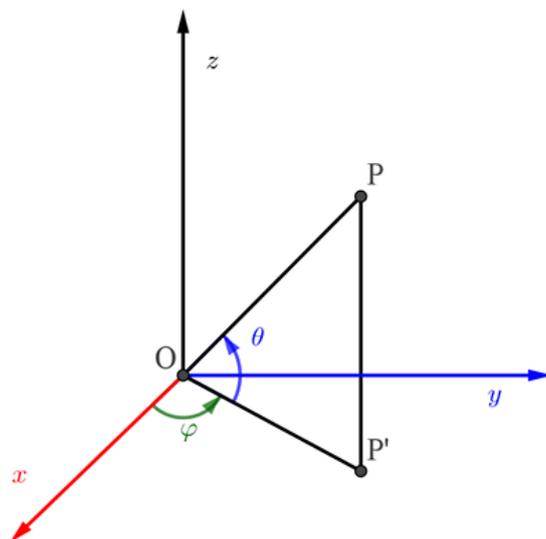


图 1.2

注意: 不等式可看成特殊的一类二元函数, 例如: $a:x+y>1$ 这个不等式, 输入 $a(1,2)$, 相当于判断点 $(1,2)$ 是否满足 a 这个不等式, 返回的结果为 `true`, 输入 $a(x+1,y)$, 相当于把不等式向左平移 1 个单位.

1.3 数学函数

在 GeoGebra 的指令帮助中, 总共内置有 66 个函数, 双击指令帮助中的函数, 函数名自动粘贴到指令栏, 在小括号中输入相应的变量或者参数. 区别于指令, 内置数学函数都是使用英文, 不翻译成中文使用. 以下是这些函数的函数名及含义 (大小写务必保持正确):

1. $\text{sqrt}(x)$: 根号, 算数平方根.
2. $\text{cbrt}(x)$: 立方根, 三次根号.
3. $\text{abs}(x)$: 绝对值函数, 可用于求向量和复数的模, 也可以求点的极径.
4. $\text{sgn}(x)$: 符号函数, 当 x 为正数时, 返回 1, $x = 0$ 时, 返回 0, x 为负数时返回 -1. $\text{sgn}(x)$ 也可写为 $\text{sign}(x)$. 用解析式来写即:

$$f(x) = \begin{cases} 1, & x > 0; \\ 0, & x = 0; \\ -1, & x < 0. \end{cases}$$

5. $\text{alt}((x,y,z))$: 求三维点 (x,y,z) 的高度角, 如图 1.1 中的 θ , 范围为 $[-90^\circ, 90^\circ]$.
6. $\text{arg}(x)$: 复数的辅角, 即复数的三角形式 $z = r(\cos \theta + i \sin \theta)$ 中的辅角 θ , 可用于求极坐标 $(r; \theta)$ 的极角 θ 和

球坐标 $(a; \varphi; \theta)$ 的方位角, 如图 1.2 中的 φ , 取值范围为: $(-180^\circ, 180^\circ]$.

7. conjugate(x): 共轭复数.
8. floor(x): 向下取整, 也称地板函数, 即求不大于 x 的最大整数, 即 $\lfloor x \rfloor$.
9. ceil(x): 向上取整, 也称天花板函数, 即求不小于 x 的最小整数, 即 $\lceil x \rceil$.
10. round(x): 四舍五入.
11. log(b,x): 以 b 为底 x 的对数.
12. exp(x): e 的 x 次幂.
13. ln(x): 自然对数, 以 e 为底的对数.
14. lg(x): 常用对数, 以 10 为底的对数.
15. ld(x): 以 2 为底的对数.
16. sin(x): 正弦函数.
17. asin(x): 反正弦函数, 也写作 arcsin(x), 等同于 asind(x).
18. cos(x): 余弦函数.
19. acos(x): 反余弦函数, 也写作 arccos(x), 等同于 acosd(x).
20. tan(x): 正切函数.
21. atan(x): 反正切函数, 也写作 arctan(x), 等同于 atand(x).
22. sinh(x): 双曲正弦函数, 定义为: $\frac{e^x + e^{-x}}{2}$, 可用作双曲线的参数方程.
23. asinh(x): 反双曲正弦函数
24. cosh(x): 双曲余弦函数, 定义为: $\frac{e^x - e^{-x}}{2}$, 可用作双曲线的参数方程.
25. acosh(x): 反双曲余弦函数.
26. tanh(x): 双曲正切函数, 定义为: $\frac{\sinh x}{\cosh x}$.
27. atanh(x): 反双曲正切函数.
28. sec(x): 正割函数, 即 $\frac{1}{\cos x}$.
29. sech(x): 双曲正割函数, 即 $\frac{1}{\cosh x}$.
30. cosec(x): 余割函数, 即 $\frac{1}{\sin x}$.
31. cosech(x): 双曲余割函数, 即 $\frac{1}{\sinh x}$.
32. cot(x): 余切函数.
33. coth(x): 双曲余切函数, 定义为: $\frac{\cosh x}{\sinh x}$.
34. asind(x): 等同于 asin(x).
35. acosd(x): 等同于 acos(x).
36. atand(x): 等同于 atan(x).
37. atan2(y, x): 反正切函数, 返回点 (x,y) 与 x 轴非负半轴的夹角, 范围是 $(-180^\circ, 180^\circ]$.
38. erf(x): 高斯误差函数 (Gaussian Error Function), 定义: $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\eta^2} d\eta$
39. gamma(x): **Gamma 函数** $\Gamma(x)$
40. beta(a, b): **Beta 函数**
41. gamma(a, x): **不完全 Gamma 函数**
42. beta(a, b, x): **不完全 Beta 函数**
43. gammaRegularized(a, x): **Regularized Gamma Function**.
44. betaRegularized(a, b, x): **Regularized Beta Function**
45. psi(x): Psi 函数, 它是 Gamma 函数对数的导数, 即 $\psi(x) = (\ln(\Gamma(x)))'$.
46. nroot(x, n): $x^{\frac{1}{n}}$, 其中 x 为非负实数, n 为非 0 实数.
47. fractionalPart(x): 小数部分. `fractionalPart(-1.2)` 返回 -0.2.
48. real(x): 实部.
49. imaginary(x): 虚部.
50. sinIntegral(x): 正弦积分.

51. `cosIntegral(x)`: 余弦积分.
52. `expIntegral(x)`: 指数积分.
53. `random()`: 产生 0-1 的随机数, `random(a,b)` 返回 a, b 之间的整数随机数.
54. `zeta(x)`: 黎曼 ζ 函数.
55. `Dirac(x)`: 狄拉克 δ 函数.
56. `Heaviside(x)`: 赫维赛德函数, 当 $x > 0$ 时, 返回 1, 当 $x \leq 0$ 时返回 0, 用分段函数的形式可以写成:

$$f(x) = \begin{cases} 1, & x > 0; \\ 0, & x \leq 0 \end{cases}$$

注意: 此外有一些类似于函数的用法:

- 如果列表 $l1$ 为数或者点的列表, 则 $l1(n)$ 表示 $l1$ 的第 n 个元素, $l1(-n)$ 表示 $l1$ 倒数第 n 个元素.
- 如果定义了曲线 $a(t)$, 则 $a(b)$ 表示 t 取 b 时, 所对应的点, 曲面 $b(s, t)$ 也是类似用法.

第2章 指令

GeoGebra 的指令主要在指令栏中输入,并返回相应的值和对象,在空白公式框、脚本、运算区也可以使用,语法和指令栏中的语法基本一致.有些指令只能在运算区使用,称为 CAS 专有指令,有些指令在绘图区和 3D 区用法不同.一般来说,GGB 的工具都与与之对应的指令,但个别的工具和指令的名称是不对应的,比如反演工具对应的指令叫做对称,也有个别工具没有指令可以实现,大部分的属性设置也都可以通过指令区实现.指令的构成一般是指令名,括号,和参数,比如:函数(<函数>,<x-起始值>,<x-终止值>),其中括号可以是小括号()也可以是中括号[],把鼠标挪到括号的右侧,可以看到该括号以及与之成对的括号变成绿色.所有的标点符号必须为英文半角符号.比如输入指令:函数(x^2 ,1,2)返回 1 个定义域在 1 到 2 之间的函数 x^2 ,注意输入指令时,指令提示中的<>是不需要输入的.

在输入中文指令时,一般输入指令的前 2 个汉字,指令栏中就会出现指令的提示,可以用鼠标或键盘的上下键来选择相应的指令,如图 2.1,选好后,第一个参数变成蓝色,如图 2.2,输完 1 个参数后,可以用键盘向右键快速跳转到下一个参数的输入,输完所有参数后按下回车键,则一个指令输入完成.也在指令帮助菜单下,找到要输入的指令,点击粘贴,即可把指令粘贴到指令栏.

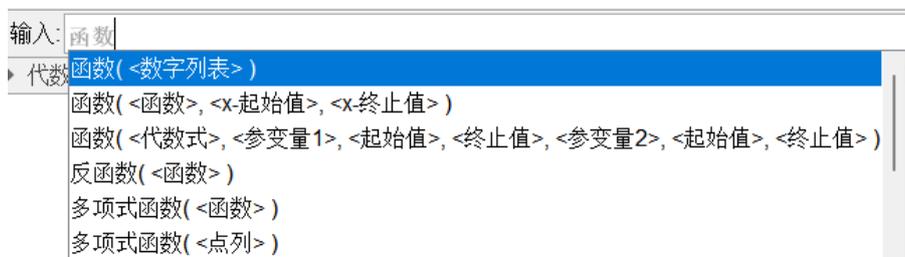


图 2.1



图 2.2

当语言选择中文时,系统可以识别输入的中文指令和英文指令,但仅对中文指令有指令提示,而英文指令没有提示.当语言选择英文时,则只能识别英文指令,中文指令无法使用.在脚本中,如果要使用英文指令,尽量保证英文指令的大小写和系统的指令保持一致,在指令栏中输入英文指令则不必考虑大小写的问题.在使用 Execute 指令(执行指令)时,其要执行的指令必须采用英文指令,且大小写必须和内置指令完全一致.例如:

```
执行({"if(x>0,1,x^2)"})
```

由于系统的正确英文指令为 If 而此处为 if,导致指令无法被执行.

2.1 3d 三维指令

2.1.1 Side. 侧面

指令

侧面(<二次曲面>)

说明: 创建二次曲面的侧面(圆柱圆锥)

示例: 已知圆柱 a, 输入指令 `侧面 (a)`, 返回圆柱 a 的侧面 b, 同时显示侧面的值 (面积), 在 3D 区画出该侧面的图形.

2.1.2 OrthogonalPlane. 垂直平面

指令

垂直平面 (< 点 >, < 直线 >)

说明: 创建过给定点垂直于给定直线的面.

指令

垂直平面 (< 点 >, < 向量 >)

说明: 创建过给定点垂直于给定向量的面.

2.1.3 Ends. 底面

指令

底面 (< 二次曲面 >)

说明: 创建有限二次曲面的上底和下底.

2.1.4 Height. 高度

指令

高度 (< 立体图形 >)

说明: 算给定立体图形的“定向”高度, 立体图形包括棱柱, 棱锥, 圆柱, 圆锥, 正多面体等.

2.1.5 Prism. 棱柱

指令

棱柱 (< 点 1 >, < 点 2 >, ...)

说明: 返回由给定点定义的棱柱, 前 n 个点为底面多边形的顶点, 最后 1 个点为另 1 个底面上的点.

指令

棱柱 (< 多边形 >, < 最高点 >)

说明: 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.

指令

棱柱 (< 多边形 >, < 高度 >)

说明: 创建一个多边形为底且给定高度 (高度可以为负数, 规定与法向量同向的为正) 的直棱柱.

2.1.6 Pyramid. 棱锥

指令

棱锥 (< 点 1>,< 点 2>,< 点 3>,< 点 4>,...)

说明: 返回由给定点定义的棱锥.

指令

棱锥 (< 多边形 >,< 顶点 >)

说明: 创建以多边形为底, 给定点为顶点的棱锥.

指令

棱锥 (< 多边形 >,< 高度 >)

说明:

返回由多边形为底和给定高度定义的正棱锥, 顶点在底面的投影为底面多边形的重心.

示例: 已知 $t1 = \triangle ABC$, 则 `棱锥(t1,3)` 返回棱锥 ABCD, 其中 D 到 $\triangle ABC$ 的重心的距离为 3.

2.1.7 Plane. 平面

指令

平面 (< 多边形 >)

说明: 创建过多边形的平面.

指令

平面 (< 圆锥曲线 >)

说明: 创建过圆锥曲线的平面.

指令

平面 (< 点 >,< 平面 >)

说明: 创建过给定点且与已知平面平行的面.

指令

平面 (< 点 >,< 直线 >)

说明: 创建过给定点和直线的平面.

指令

平面 (< 直线 >,< 直线 >)

说明: 创建过给定直线的平面 (当两条直线在同一平面内时)

指令

平面 (< 点 >,< 点 >,< 点 >)

说明: 创建过三点的平面.

指令

平面 (< 点 >, < 向量 >, < 向量 >)

说明: 创建由一点和 2 个向量作为基底的平面.

Tips: 平面 a 的方程为 $Ax + By + Cz + D = 0$, 其中 (A, B, C) 即平面 a 的法向量, 输入指令 `系数列表(a)` 可得到列表 A, B, C, D , 若在平面 a 上任取 1 点 P , P 与法向量的数量积等于 $-D$.

2.1.8 Sphere. 球面

指令

球面 (< 球心 >, < 半径 >)

说明: 由球心和半径创建球面

示例:

球面((0,0,0),1) $\gg x^2 + y^2 + z^2 = 1$

指令

球面 (< 球心 >, < 球面上一点 >)

说明: 以球心和球面上 1 点画球面.

2.1.9 Surface. 曲面

指令

曲面 (< 函数 >, < 角度 | 弧度 >)

说明: 创建以 1 个旋转曲面, 将给定的函数从 0 旋转到 x 轴的给定角度..

指令

曲面 (< 曲线 >, < 角度 | 弧度 >, < 轴线 >)

说明: 创建一个旋转曲面, 将曲线绕着给定的旋转轴从 0 旋转到给定角度.



注意: 在当前的版本里, 除了曲线以外, 线段, 折线, 多边形, 圆, 弧线等对象都可以以这种方式产生旋转曲面, 而不再局限于曲线, 旋转的角度必须为正角, 可以使用弧度制也可以采用角度制, 如果角度为负值, 则返回未定义对象.

示例: 如图 2.1, 已知折线 f , 指令: `曲面(f, 2pi, z 轴)` 产生圆台曲面 a .

指令

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

说明: 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) .

示例: `曲面((R + r cos(u)) cos(v), (R + r cos(u)) sin(v), r sin(u), u, 0, 2 pi, v, 0, 2 pi)` 返回: 如图 2.2 所示的圆环面.



图 2.3

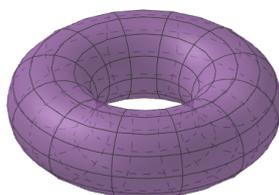


图 2.4

注意: 这里的表达式可以是直角坐标也可以是参数方程、极坐标以及球面坐标, 而且也可以混合使用, 三个表达式即可以分开写, 也可以合在一个表达式中, 如果只有 x, y 两个表达式时, 必须将表达式用小括号括起来. 以下的表达式都是合乎语法的:

曲面($s, t, 0, s, 0, 1, t, 0, 1$) >>>返回1个平面

曲面($s \ u+t \ v, s, 0, 1, t, 0, 1$) #其中 u, v 为2个三维向量

曲面($A+(1;s;t), s, 0, 2\pi, t, -\pi/2, \pi/2$) #以A为圆心, 1为半径的球, 这是球坐标与直角坐标的混合使用.

曲面($(1;s)+t \ (1;s;s/2), s, 0, 2\pi, t, -1, 1$) #极坐标与球坐标的混用.

曲面($(s, t), s, 0, 10, t, 0, 10$) #只有 x, y 表达式, 需要用括号括在一起, 此时3D区没有这个曲面的图像, 在绘图区可以显示这个曲面(实质上是网格, 注意和3D曲面的区别, 见图2.3)

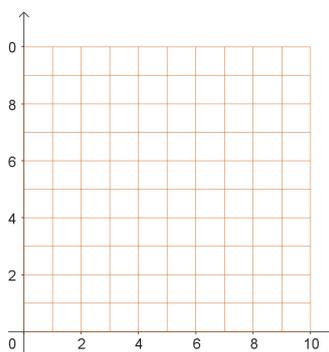


图 2.5

2.1.10 Top. 上底

指令

上底 (< 二次曲面 >)

说明: 创建有限二次曲面的上底

2.1.11 Volume. 体积

指令

体积 (< 立体图形 >)

说明: 计算给定立体图形的体积.

Tips: 圆柱、圆锥、棱柱、棱锥等立体图形的值即体积.

2.1.12 CylinderInfinite. 无限长圆柱

指令

无限长圆柱 (< 轴线 >, < 半径 >)

说明: 用给定半径和轴线创建无限圆柱.

指令

无限长圆柱 (< 轴线上点 >, < 向量 >, < 半径 >)

说明: 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.

指令

无限长圆柱 (< 轴线上点 1 >, < 轴线上点 2 >, < 半径 >).

说明: 创建给定半径和过给定两点线为对称轴的无限圆柱.

2.1.13 ConeInfinite. 无限长圆锥

指令

无限长圆锥 (< 顶点 >, < 轴向量 >, < 半顶角角度 | 弧度 >)

说明: 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, 半顶角为给定角度的无限圆锥. 半顶角的范围为 $(0^\circ, 90^\circ)$

指令

无限长圆锥 (< 顶点 >, < 轴上点 >, < 半顶角角度 | 弧度 >)

无限长圆锥 (< 顶点 >, < 直线 >, < 半顶角角度 >)

说明: 略

2.1.14 Bottom. 下底

参见Top. 上底(略)

2.1.15 IntersectConic. 相交曲线

指令

相交曲线 (< 平面 >, < 二次曲面 >)

说明: 获取二次曲面与平面的交线

指令

相交曲线 (< 二次曲面 1>, < 二次曲面 2>)

说明: 当相交曲线是二次曲线, 返回一个曲线.

注意: 球面和正方体的交线无法由相交曲线求得, 如果要求多边形和二次曲面的交线, 则先要画出多边形所在平面, 再画该平面和二次曲面的交线. 另外可以用相交曲线指令来画正方体和平面的交线.

示例: 如图 2.4,a 为正方体, p 为三角形所在的平面, 则 `相交路径 (a,p)` 返回交线, 即绿色的多边形.

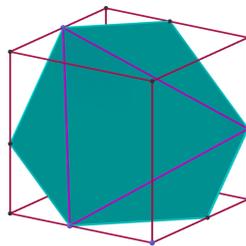


图 2.6

2.1.16 Cylinder. 圆柱

指令

圆柱 (< 圆 >, < 高度 >)

说明: 以给定的底和高构建圆柱.

指令

圆柱 (< 下底圆心 >, < 上底圆心 >, < 半径 >)

说明: 以给定的点为上下底圆心和给定半径构建圆柱.

2.1.17 Cone. 圆锥

指令

圆锥 (< 圆 >, < 高度 >)

说明: 用给定的底和高构建圆锥.

指令

圆锥 (< 底面圆心 >, < 顶点 >, < 底面半径 >)

说明: 以第一个点为圆心, 第二点为顶点用给定的半径构建圆锥.

指令

圆锥 (< 顶点 >, < 向量 >, < 半顶角角度 | 弧度 >).



说明: 以给定点为顶点, 过顶点平行于向量的直线为对称轴, 构建半顶角为给定角度的无限圆锥.

2.1.18 Net. 展开图

指令

展开图 (< 多面体 >, < 展开程度值 0-1 >)



说明: 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.

指令

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



说明: 仅适用于正多面体, 允许通过指定底面和被剪开的棱创建正多面体的不同展开图.

示例: 如图 2.7 `展开图(a, 1, faceABCD, edgeAB, edgeBC)`: 将正方体 a, 以四边形 ABCD 作为底, 剪开棱 AB, BC 完全展开. 原则上正方体完全展开需要剪开 7 条棱, 您可以全部输入 7 条棱的名字, 如果只输入其中几条, 则不一定得到您想要的展开图.

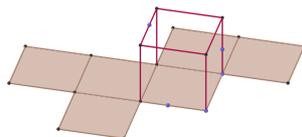


图 2.7

注意: 展开图指令不适用于旋转体, 对于一般的棱柱 (包括长方体) 只有 1 种固定的展开方式, 无法通过指定棱切开.

2.1.19 Octahedron. 正八面体

参见 [Tetrahedron](#). 正四面体及在线英文帮助.

2.1.20 Icosahedron. 正二十面体

参见 [Tetrahedron](#). 正四面体及在线英文帮助

2.1.21 Cube. 正六面体

. 参见 [Tetrahedron](#). 正四面体及在线英文帮助

2.1.22 Dodecahedron. 正十二面体

参见 [Tetrahedron](#). 正四面体及在线英文帮助.

2.1.23 Tetrahedron. 正四面体

指令

正四面体 (< 等边三角形 >)

说明: 以给定的三角形为底构建正四面体.

指令

正四面体 (< 点 >, < 点 >, < 点 >)

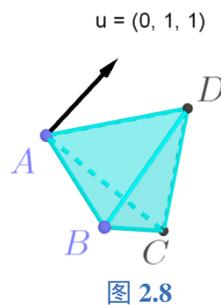
说明: 以底面的三个顶点构建正四面体. 注意: 若只输入 2 个点, 则第三个顶点是以 A,B 中点为圆心, AB 为轴, 半径为 AB 长度的 $\frac{\sqrt{3}}{2}$ 倍的圆上的 1 个点

指令

正四面体 (< 点 >, < 点 >, < 方向 >)

说明: 以给定 2 个点为底面的 2 个顶点, 以给定方向作为底面的法向量方向, 构建正四面体.

示例: 已知点 $A = (0, 0)$, $B = (2, 0)$, $u = (0, 1, 1)$, 则 `正四面体(A,B,u)` 返回正四面体 $ABCD$, 其中平面 ABC 的法向量与 u 同向. 即 $\vec{CB} \otimes \vec{CA}$ 与 u 同向.



2.1.24 PlaneBisector. 中垂面

指令

中垂面 (< 点 1 >, < 点 2 >)

说明: 创建两点间的中垂面.

指令

中垂面 (< 线段 >)

说明: 创建线段的中垂面.

2.2 GeoGebra 指令

2.2.1 CASLoaded

指令

CASLoaded()



说明: 返回一个布尔值, 官方的手册中没有该指令的使用方法.

2.2.2 AxisStepx.x 轴步长

指令

y 轴步长 ()



说明: 返回当前 x 轴的步长幅度, 如果设定了 x 轴的间距, 则步长即间距, 如果没有设定间距, 则步长等于坐标轴上相邻 2 个数字之间的距离.

2.2.3 AxisStepY.y 轴步长

指令

y 轴步长 ()



说明: 返回当前 y 轴的步长幅度, 如果设定了 x 轴的刻度间距, 则步长即刻度间距, 如果没有设定间距, 则步长等于坐标轴上相邻 2 个数字之间的距离. 滚动鼠标滚轮或者按住 Shift 键的同时拖动 x 轴, 都可以改变步长.

参见 [AxisStepx.x 轴步长指令](#)

2.2.4 DynamicCoordinates. 动态坐标

指令

动态设置坐标 (< 点 >, < 横坐标 >, < 纵坐标 >)



说明: 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. 这种点兼具自由点和附属点的特性和优点, 具有像自由点一样的自由, 却又像附属点一样的自律; 没有自由点的不受控制, 也不像其他附属点一样, 无法拉动.

示例:

$$B = \text{动态坐标}[A, \text{round}(x(A)), \text{round}(y(A))]$$

生成 1 个只在格点上运动的点 B .

示例: 已知函数 $y=\sin(x)$, A 是 1 个自由点, 则指令:

$$C = \text{动态坐标}(A, x(A), \min(y(A), \sin(x(A))))$$

生成的点 C , 始终在函数图像上或下方. 由此可见, 可以用动态坐标指令, 限制点在某个不等式区域移动 (类似于内点的作用)

3D 区指令

动态坐标 (< 点 >, < 横坐标 >, < 纵坐标 >, < 竖坐标 z >)



说明: 该指令中的点需要为 3 维点, 否则生成的点只能在 xy 平面上运动, 使用方法和绘图区的使用方法基本一致.

更多的案例见在线帮助.

2.2.5 Object. 对象**指令**

对象 (< 对象名称文本 >)



说明: 返回给定文本格式名称的对象. 结果是从属对象.



注意:

1. “对象”指令与“名称”指令相反.
2. 对象指令不能用于自定义工具.
3. 官网声称已废弃对象指令, 实际上仍然可以使用.

示例: 如图 2.9, 假设一些点 A_1, A_2, \dots, A_{10} 以及一个参数 $n=2$, “对象 (“A”+n)” 创建一个对象 A, 其为点 A_2 的拷贝. 改变 n 的大小在 1 到 10 之间的整数变化, 创建的对象会自动改变为指定的 A_n 点.

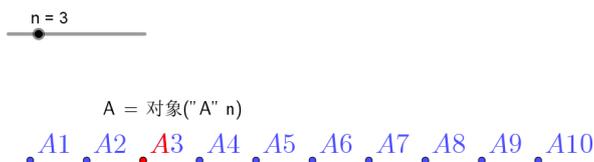


图 2.9

示例: 如果想知道绘图区中是否存在 1 个名为 K 的对象, 如果我们使用指令: `是否已定义(K)`, 那么当 K 存在时, 该指令范围 true, 当 K 不存在时, 系统会让我们选择是否生成滑动条, 而无法输出 false 的结果. 因此, 这里可以使用 `是否已定义(对象("K"))` 指令, 则无论 K 是否存在, 都可以输出正确的结果.

2.2.6 ToolImage. 工具图标**指令**

工具图标 (< 数字 >)

工具图标 (< 数字 >, < 点 >)

工具图标 (< 数字 >, < 点 >, < 点 >)



说明: 这 3 个指令都返回对应序号的工具图标, 大小为 32×32 , 点用于图片的位置, 分别为图片的角点 1 和角点 2, 如果不设定角点, 则图标可以自由移动.

详细的序号与图标的对应关系, 请查看在线帮助及图 2.10 和 2.11.

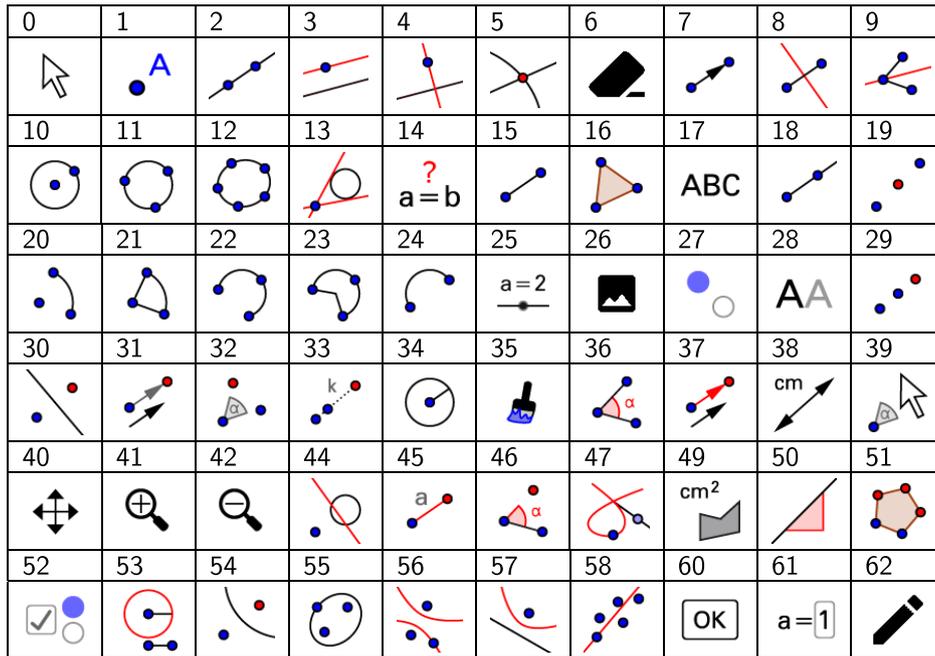


图 2.10

2.2.7 SlowPlot. 缓慢绘制

指令

缓慢绘制 (< 函数 >).

说明: 创建一个函数图象的动画 (默认重复播放): 函数自左向右画出. 动画由一个滑动条控制, 这个滑动条随指令同时构造. 其中函数图像的范围从角点 1 到角点 2.

指令

缓慢绘制 (< 函数 >, < 是否重复?true/false>)

说明: 缓慢绘制的时候, 选择是否重复播放动画, 当第 2 个参数选择 **false** 时, 动画播放 1 次结束

64	65	66	67	68	70	71	72	73	501
1001	1002	1003	1004	1005	1006	1007	1008	1009	1010
2001	2002	2003	2004	2005	2020	2021	2022	2040	2041
2042	2043	2044							

图 2.11

参见Corner. 角点.

2.2.8 Corner. 角点

指令

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 >)

说明: 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.

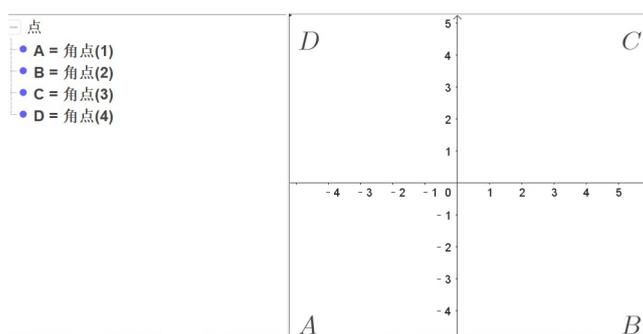


图 2.12

指令

角点 (< 绘图区编号数字 | 图片 | 文字 >, < 角点数字 1-左下角 | 2-右下角 | 3-右上角 | 4-左上角 >)

说明: 返回绘图区或图片或数字的角点, 数字和绘图区的对应关系如下表:

绘图区编号	区
1	绘图区
2	绘图区 2
-1	3D 绘图区

3D 区角点 1-8 表示 8 个卦限的角落, 如图 2.13

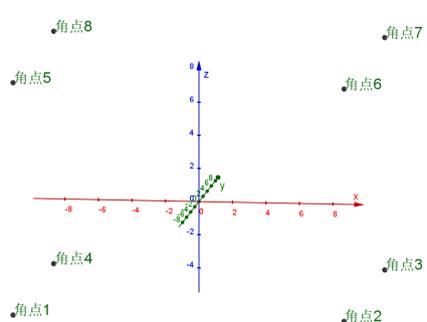


图 2.13

说明: 对于 $n = 9$, 返回点 $(wh0)$, 其中 w 和 h 是图形的宽度和高度以像素为单位查看; 当 $n = 10$ 返回点 $(WH0)$, 其中 W 和 H 是像素; 当 $n = 11$ 时, 返回视图方向 (对于平行投影) 或眼睛位置 (例如, 透视) 投影; 对于 $n = 12$, 返回屏幕从左到右的方向 (眼睛的位置); 对于 $n = 13$, 则返回 xyz 轴的刻度, 默认值为 50,50,50, 可以通过该角点的值计算出 3D 区的视图被放大或者缩小的多少倍.

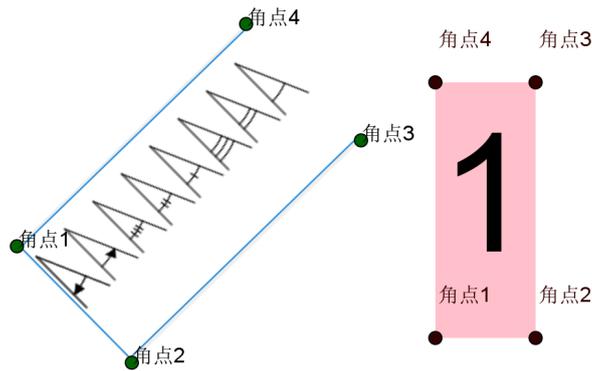


图 2.14

说明: 如图 2.14 图片和文本框的角点都是逆时针排列, 以”左下角”为角点 1, 文本和图片都必须取消勾选屏幕绝对位置选项.

编者注:

官网的在线帮助中称在序列和映射中不能使用角点 (< 文本 >, < 数字 >), 笔者在试验后发现是可以的, 图 2.14 中的角点即为序列所作.

示例:

跟随视角转动的大圆 (图 2.12):

圆周((0, 0, 0), 1, 向量((0, 0, 0), 角点(-1, 11)))

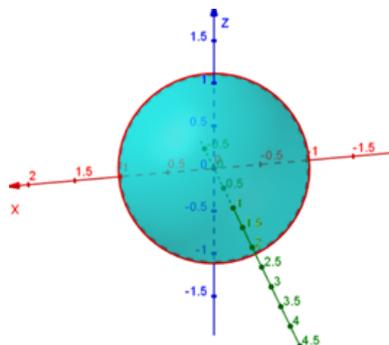


图 2.15

2.2.9 Name. 名称

指令

名称 (< 对象 >)

说明: 返回一个对象的名称 (文本格式)

示例:

名称(A) >>>"A"

注意: 在用映射或者序列批量获取对象的名称时, 对象的类型要一致, 否则无法获取正确的名称.

示例: 已知点 $A, B, C, D, E, F, l1 = \{A, B, C, D, E\}$, 要想获得这些点的名称, 则输入:

```
序列(名称(l1(i)),i,1,5)    >>> {"l1(i)", "l1(i)", "l1(i)", "l1(i)", "l1(i)"}
```

```
序列(名称(元素(l1,i)),i,1,5)  >>>{"A", "B", "C", "D", "E"}
```

```
映射(名称(p),p,l1)    >>> {"p", "p", "p", "p", "p"}
```

可见批量获取名称时, 只有第二种是符合要求的, 读者在使用时, 需要注意.

2.2.10 SetConstructionStep. 设置作图步骤

指令

设置作图步骤 (< 数字 >)

说明: 跳转到作图的第 n 步, 其中 n 由指定的数字决定. 您可以使用此命令创建用于替换或增强导航栏的按钮.

2.2.11 ToPoint. 转换为点

指令

转换为点 (< 复数 >)

说明: 将给定的一个复数转换为点.

Tips: 虚数单位 i 由组合键 Alt+i 输入

2.2.12 ToComplex. 转换为复数

指令

转换为复数 (< 向量或点 >)

说明: 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.

2.2.13 ToPolar. 转换为极坐标形式

指令

转换为极坐标形式 (< 复数 >|< 向量 >).

说明: 将给定的一个向量(点、复数) 转换为极坐标值.

2.2.14 ConstructionStep. 作图步序

指令

作图步序()
作图步序 (< 对象 >)

说明: 以数字形式返回当前已完成的操作步骤数和返回指定对象的步骤顺序

2.3 Financial. 财务

与财务有关的指令很少且很不常用, 故本节只很简略的罗列这些指令语法.

2.3.1 Rate. 利率

指令

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 >)

说明: 返回年金的每期利率。

2.3.2 Payment. 每期付款额

指令

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 >)

说明: 返回固定每期付款额和固定利率的投资期

2.3.3 Periods. 期数

指令

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 >)

说明: 返回基于固定每期付款额和固定利率的定期投资期数。

2.3.4 FutureValue. 未来值

指令

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 >)

说明: 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。

2.3.5 PresentValue. 现值

指令

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 >)

说明: 返回一个投资的付款总计。

2.4 代数指令

2.4.1 Cross. 叉积

指令

叉积 (< 向量 1 >, < 向量 2 >)

说明: 计算向量 1 和向量 2 的向量积。向量积也称外积、叉积。

示例:

叉积((1,3,2),(0,3,-2))} >>>(-12,2,3)

 **注意:** 叉积在 cas 区使用时, 可以计算含参的叉积.

CAS 指令

叉积 ((a,b,c),(d,e,f))
 $\Rightarrow (bf - ce, cd - af, ae - bd)$

 **注意:** 若 $u = (x_1, y_1), v = (x_2, y_2)$, 则得到的结果为: $x_1y_2 - x_2y_1$, 是一个数量, 即 u, v 为三维向量时叉积为向量, u, v 为二维向量时叉积为数量. 根据叉积的定义, 很容易知道叉积 $(u, v) = -\text{叉积}(v, u)$. 二维向量的叉积可用于计算平行四边形的面积, 三维向量的叉积可用于计算平面的法向量.

2.4.2 Division. 除法

指令

除法 (< 被除数 >, < 除数 >)

说明: 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.

指令

除法 (< 被除式 _ 多项式 >, < 除式 _ 多项式 >).

说明: 给出两个多项式相除的商和余式.

示例:

除法 (16,3) >>>{5,1}
 除法 ($x^2+3x+1, x-1$) >>>{x+4,5}

 **注意:** 在代数区支持单变量除法, 在 cas 区, 还支持多变量除法

示例:

CAS 指令

除法 ($x^2+y^2, x+y$)
 $\Rightarrow \{x - y, 2y^2\}$

2.4.3 Dot. 点积

指令

点积 (< 向量 u >, < 向量 v >)

说明: 返回两个向量的点积 (内积), 点积也可以直接用 $u * v$ 计算.

2.4.4 CompleteSquare. 顶点式

指令

顶点式 (< 二次函数 >)

说明: 返回二次函数的顶点式 $a(xh)^2 + k$.

示例:

CAS 指令

$$\begin{aligned} & \text{顶点式 (} a x^2 + b x + c \text{)} \\ \Rightarrow & \left(x - \frac{-b}{2a}\right)^2 + \frac{4ac - b^2}{4a} \end{aligned}$$

此处 a, b, c 都未定义过。

2.4.5 CommonDenominator. 公分母**指令**

公分母 (< 分式 1>, < 分式 2>)

说明: 返回两个表达式最小公分母的方程格式函数。

示例:

公分母(3/(2x+1), 3/(4x^2+4x+1)) >>> $f(x) = 4x^2 + 4x + 1$

2.4.6 NextPrime. 后一质数**指令**

后一质数 (< 数值 >)

说明: 返回比输入数值更大的最小质数。

2.4.7 Simplify. 化简**指令**

化简 (< 函数 >)

说明: 尽可能简化所给函数的项 (合并同类项), 在 cas 区使用时, 可以保留未定义的变量 (参数)。

CAS 指令

化简 ("< 文本 >")

说明: 通过清除重复的符号而整理表达式文本等等。

示例:

CAS 指令

$$\begin{aligned} & \text{化简 (} 3 * x + 4 * x + a * x \text{)} \\ \Rightarrow & x(a + 7) \end{aligned}$$

其中 a 未定义

2.4.8 Solutions. 解集**指令**

解集 (< 方程 > | < 不等式 >)

说明: 给出以 x, y, z 为未知数的方程 (不等式) 或方程组的解列表.

 **注意:** 官网并没有指出解集指令可以用来解不等式, 但实质上可以解不含参的简单不等式

示例:

```
解集(x^2=1)      >>>{-1,1}
解集(x^2-2x>0)  >>>{x<0,x>2}
解集({x + y = 1, x - y = 0}) >>>( 1/2  1/2 )
```

在 cas 中使用时, 可以声明主元.

指令

解集 (< 方程 >, < 变量 >). 解集 (< 方程列表 >, < 变量列表 >).

说明: 给定未知变量方程 (组) 且返回解列表, 变量可以指定, 可以不是 x, y, z .

示例:

CAS 指令

```
解集(2a^2+5a+3=b,a+b=3,a,b)
⇒ ( 0  3 )
   (-3 6 )
```

2.4.9 NSolve. 近似解

指令

近似解 (< 方程 >).

说明: 找出方程的近似解,

其他用法与解集的用法基本一致, 此处不再赘述. 类似的指令还有近似解集, 精确解, 方法都基本一致.

2.4.10 Nsolutions. 近似解集

参见2.4.8.

2.4.11 Solve. 精确解

参见Solutions. 解集.

2.4.12 PreviousPrime. 前一质数

指令

前一质数 (< 数值 >)

说明: 返回比输入数值小的最大质数.

2.4.13 Div. 商式

指令

商式 (< 被除数 >, < 除数 >).

说明: 返回两个数值的整数商 (得数结果的整数部分, 向下取整), 即 $\lfloor \frac{\text{被除数}}{\text{除数}} \rfloor$.

指令

商式 (< 被除式 >, < 除式 >).

说明: 返回两个多项式的商.

示例:

商式(-3.1,2) >>>-2

$\therefore -3.1 \div 2 = -1.55$

$\therefore \lfloor -1.55 \rfloor = -2$

因此商式 = -2

Tips: 除法, 余式, 商式这 3 个指令的关系: 被除数 = 除数 \times 商式 + 余式, 除法 = 商式, 余式.

编者注:

在除数或被除数不是整数的情况下, 出现了代数区和 cas 区商式结果不同的情形, 代数区会对商取整, 而 cas 区则直接等于被除数/除数, 笔者认为这个属于 bug.

商式(-2.1,1) >>>-3

CAS 指令

商式(-2.1,1) ->-2.1

2.4.14 IFactor. 实数域因式分解**指令**

实数域因式分解 (< 多项式 >)

说明: 在可能的情况下, 分解出多项式含有无理数的因式.

2.4.15 IsPrime. 是否为质数**指令**

是否为质数 (< 数字 >)

说明: 根据数值是否为质数得出 true 或 false.

2.4.16 IsFactored. 是否已分解**指令**

是否已分解 (< 多项式 >)

说明: 根据多项式是否因式分解得出 true 或 false.

示例:

是否已分解(x^2-2x) >>>false

2.4.17 Factor. 因式分解

指令

因式分解 (< 多项式 >)

说明: 将多项式因式分解, 在 cas 区可以指定对哪个变量进行分解:

CAS 指令

因式分解 (< 表达式 >, < 变量 >)

示例:

CAS 指令

因式分解 ($a x - 2a, a$) $\Rightarrow a(x - 2)$

2.4.18 Divisors. 因数个数

指令

因数个数 (< 正整数 >)

说明: 计算全部正因数的数量, 包括指定数值本身.

2.4.19 DivisorsSum. 因数和

指令

因数和 (正整数)

说明: 计算全部的正因数之和, 包括指定数值本身.

2.4.20 DivisorsList. 因数列表

指令

因数列表 (< 正整数 >)

说明: 给出给定正整数的全部正因数的列表, 包括本身.

2.4.21 RightSide. 右边

指令

右边 (< 方程 >).

说明: 以多元函数的形式给出化简后方程式的右边

示例:

右边 ($x+y=1$) $\gg a(x, y) = 1$

 **注意:** 假如有方程 $a : x^2 + y^2 = 1$, 则输入:

$x+y >$ 右边 (a) $\gg x + y > 1$

Tips: 右边指令经常与左边指令合起来用, 比如有圆 $c:x^2 + y^2 = 1$, 则

左边(c)>右边(c) $\gg x^2 + y^2 > 1$

#即圆的外部这个平面区域

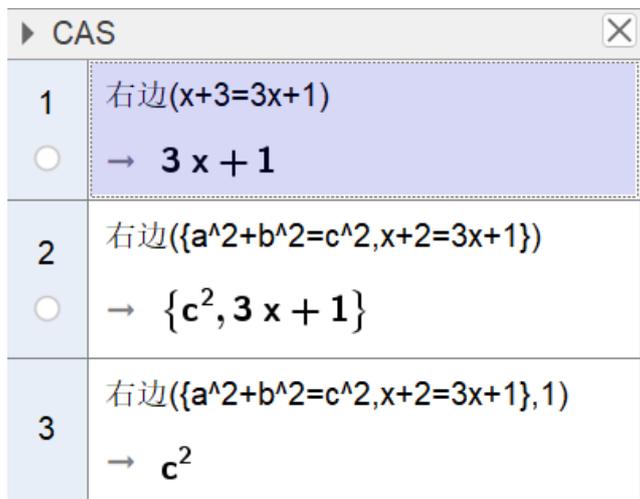


图 2.16

CAS 指令

右边 (< 方程 >)

右边 (< 方程列表 >)

右边 (< 方程列表 >, < 索引 n >)

说明: 分别返回方程, 方程组, 即指定方程组的第 n 个方程的右边, 示例见图 2.16

2.4.22 Mod. 余式

指令

余式 (< 被除数 >, < 除数 >)

余式 (< 被除式 >, < 除式 >)

说明: 返回余数或余式. 余数的定义为 $Mod(x, y) = y \cdot (\frac{x}{y} - \lfloor \frac{x}{y} \rfloor)$, 即 $x - y \lfloor \frac{x}{y} \rfloor$.

注意: 说明: 取余 (余式) 运算中除数和被除数可以是任何实数 (除数不为 0).

示例:

余式(-2,-3) $\gg 1$

余式(3.1,2) $\gg 1.1$

示例:

序列(余式(n-1,4)+1,n,1,10) $\gg \{1,2,3,4,1,2,3,4,1,2\}$

余式指令经常用来做与周期性变化有关的课件, 例如物体的滚动等, 更多的案例见笔者在官网的[课件](#).

注意: Mod 指令在当前版本中在 cas 中的计算有 bug, 比如 $mod(3.1, 2)$ 在 cas 区中得到的结果为 0, 这显然是错误的 (官网已确认该 bug, 但不会对其进行修复, 建议大家使用整数作为除数和被除数, 或用 $f(x, y) = x - y \lfloor \frac{x}{y} \rfloor$ 代替 MOD).

2.4.23 Expand. 展开

指令

展开 (< 代数式 >)

说明: 将代数式展开, 在绘图区会同时绘制出函数的图像

示例:

展开 $((2x-1)^2+2x+3)$ $\ggg f(x) = 4x^2 - 2x + 4$

2.4.24 PrimeFactors. 质因数

指令

质因数 (< 数字 >).

说明: 返回质因数列表, 其中元素的乘积等于指定数值.

CAS 指令

质因数 (100) \Rightarrow {2, 2, 5, 5}

2.4.25 ToBase. 转换进制

指令

转换进制 (< 十进制数 >, < 进制 (基数)_2-36 >).

说明: 以文本格式将指定数值转化为不同的进制 (基数), 数值必须为整数, 进制从 2-36.

示例:

转换进制 (1024, 16) \ggg "400"

2.4.26 FromBase. 转换为十进制

指令

转换为十进制 (< 指定进制型数字 >, < 进制 (基数) 2 36 >)

说明: 将文本格式的给定基数的进制数字转换为 10 进制的数字, 基数的范围 2-36

示例:

转换为十进制 ("1234", 10) \ggg 1234
转换为十进制 ("101001", 2) \ggg 41

2.4.27 GCD. 最大公约数

指令

最大公约数 (< 整数 1 >, < 整数 2 >)

说明: 返回 2 个整数的最大公约数, 在 cas 区中语法一致.

指令

最大公约数 (< 整数列表 >)

说明: 返回列表中所有整数的最大公约数, 在 cas 区中语法一致.

另外在 cas 区中还可以对多项式求最大公因式.

CAS 指令

最大公约数 (< 多项式 1>, < 多项式 2>)

最大公约数 (< 多项式列表 >)

示例:

CAS 指令

最大公约数 ($x^2+4x+4, x^2-x-6, x^3-4x^2-3x+18$)

$\Rightarrow x+2$

2.4.28 Max. 最大值

指令

最大值 (< 区间 >)

说明: 返回区间的最大值 (包含开区间)

指令

最大值 (< 数字 1>, < 数字 2>)

说明: 返回两个给定数值的最大者.

注意: 最大值指令无法给出两函数的最大值. 比如最大值 ($x^2, x+1$) 无效.

Tips: 要想输出函数 $f(x)$ 和 $g(x)$ 的最大值, 可以采用: $\frac{f(x)+g(x)-|f(x)-g(x)|}{2}$ 或者采用如果指令: 如果 (f>g,f,g)

指令

最大值 (< 数字列表 >)

说明: 返回列表中数值的最大值.

指令

最大值 (< 数据列表 >, < 频数列表 >)

说明: 返回数据列表中频数大于 0 的数字的最大值.

示例:

最大值({1, 2, 3, 4, 5}, {5, 3, 4, 2, 0}) $\ggg 4$

说明: 频数大于 0 的数字有 1,2,3,4, 其中最大的是 4, 因此指令返回 4.

指令

最大值 (< 函数 >, < x- 起始值 >, < x- 终止值 >)

说明: 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.

2.4.29 LCM. 最小公倍数

指令

最小公倍数 (< 整数列表 >)

说明: 计算数列表中全部元素的最小公倍数.

指令

最小公倍数 (< 多项式列表 >)

说明: 计算多项式列表中各个多项式的最小公倍数.

2.4.30 Min. 最小值

最小值指令的用法基本和最大值指令一致, 参见 **Max. 最大值**

Tips:

最小值(1,2) >>> 2
 $(f+g-|f-g|)/2$ >>> 函数f和g较小的部分
 如果($f < g$, f, g) >>> 函数f和g较小的部分

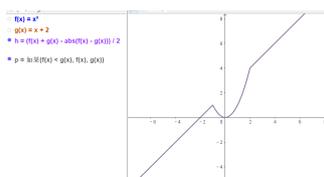


图 2.17

示例: 滑动条分裂技术: a 为 0-5 滑动条, 滑动 a 时, 观察以下指令得到的列表中的元素变化.

序列(最小值(最大值(a-i, 0), 1), i, 0, 4)
 >>> 可以观察到5个元素都在0-1之间变化.

2.4.31 LeftSide. 左边

参见 **RightSide. 右边**

2.5 Probability. 概率

概率方面笔者只熟悉高中及以下的相关指令, 涉及到大学概率统计中的各种分布, 笔者只能依据官网所给的英文帮助, 翻译成中文, 难免有错.

说明: GeoGebra 中的分布大体都分为密度分布函数 (PDF)、累计分布函数 (CDF) 以及相应的变量值所对应的函数值, 一般参数 *true* 表示累计函数, *false* 表示密度分布函数, 因此为了节省篇幅, 这里仅对 *F* 分布做详细的解释, 其他的分布可以参考 *F* 分布.

2.5.1 FDistribution.F 分布

释义: **F 分布** 是 1924 年英国统计学家 Ronald.A.Fisher 爵士提出, 并以其姓氏的第一个字母命名的. 它是两个服从卡方分布的独立随机变量各除以其自由度后的比值的抽样分布, 是一种非对称分布.

指令

F 分布 (< 分子自由度 >, < 分母自由度 >, < 变量值 v >).



说明: 计算 F 分布累积概率函数 (n 是分子自由度, d 是分母自由度) 在给定变量值 v 所对应的函数值, 即 $P(x \leq v)$.

指令

F 分布 (< 分子自由度 >, < 分母自由度 >, 变量值, < 是否累积?true|false>)



说明: 如果 *true*, 计算 F 分布累积概率函数值, 即 $P(x \leq v)$, 否则计算 F 分布概率密度函数值 $P(x = v)$.

2.5.2 nCr

指令

nCr(正整数 n, 正整数 r)



说明: nCr 指令目前没有对应的中文指令, 其含义为组合数, $nCr(n,r) = \frac{n!}{(n-r)!r!}$ 和指令 **Binomial**. 二项式系数的功能一致.

类似的 nPr 指令, 在指令帮助中没有这个指令, 但 nPr 实质上可以使用, 且在 [在线帮助](#) 中有该指令的说明, 语法为 nPr(整数 n, 整数 r), 含义为排列数: $A_n^r = \frac{n!}{(n-r)!}$, 在代数区和运算区都可以使用.

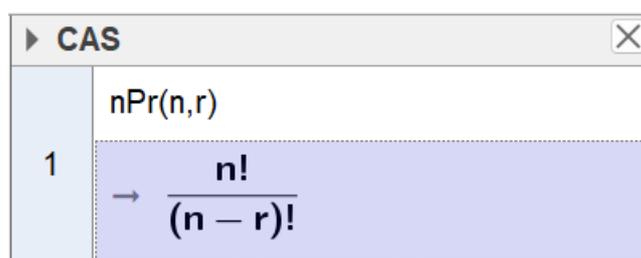


图 2.18

2.5.3 TDistribution.t 分布

在概率论和统计学中, t-分布 (t-distribution) 用于根据小样本来估计呈正态分布且方差未知的总体的均值. 如果总体方差已知或者在样本数量足够多时, 则应该用正态分布来估计总体均值. 密度函数为:

$$f_Z(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi} \Gamma\left(\frac{n}{2}\right)} \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}$$

指令

t 分布 (< 自由度 >, < 变量值 >)

t 分布 (< 自由度 >, < 变量值 >, < 是否累积?true|false>)

t 分布 (< 自由度 >, x, < 是否累积?true|false>)



2.5.4 Erlang. 爱尔朗分布

爱尔朗分布有两个参数, 阶数 k 和比率 λ , 具有阶数 k 的爱尔朗过程被称为 k 阶爱尔朗.

指令

爱尔朗分布 (< 形状参数 k >, < 比率参数 λ >, < 变量值 >)

爱尔朗分布 (< 形状参数 k >, < 比率参数 λ >, < 变量值 >, < 是否累积? true|false >)

爱尔朗分布 (< 形状参数 k >, < 比率参数 λ >, x , < 是否累积? true|false >)



2.5.5 BetaDist. 贝塔分布

释义: 贝塔分布 (Beta Distribution) 是一个作为伯努利分布和二项式分布的共轭先验分布的密度函数, 在机器学习和数理统计学中有重要应用. 在概率论中, 贝塔分布, 也称 B 分布, 是指一组定义在 (0,1) 区间的连续概率分布. 概率密度函数为:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1} du} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

指令

贝塔分布 (< 形状参数 α >, < 尺度参数 β >, < 变量值 >)

贝塔分布 (< 形状参数 α >, < 尺度参数 β >, < 变量值 >, < 是否累积? true|false >)

贝塔分布 (< 形状参数 α >, < 尺度参数 β >, x , < 是否累积? true|false >)



说明: 官网无这个指令的介绍, 其用法和其他分布基本一致.

2.5.6 Bernoulli. 伯努利分布

释义: 伯努利分布即 0-1 分布, p 为成功概率.

指令

伯努利分布 (< 概率 p >, < 是否累积? true| false >)



说明: 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.

 注意: 在本版本中, 当累计为 true 时, 该条形图无法正常显示, 它的值为 ∞ .

2.5.7 Poisson. 泊松分布

指令

泊松分布 (< 平均数 >)



说明: 返回定平均数的泊松分布条形图

指令

泊松分布 (< 平均数 >, < 是否累积? true|false >)



说明: true 返回累计分布阶梯图, false 返回分布条形图

指令

泊松分布 (< 平均数 >, < 变量值 >, < 是否累积? true|false >)



说明: 即在第二种用法中, 代入具体的变量值,true 计算 $P(x \leq v)$, false 计算 $P(x=v)$.

指令

泊松分布 (< 平均数 >, < 数字 u >...< 数字 v >)

说明: 计算 $P(u \leq x \leq v)$

2.5.8 RandomPoisson. 泊松分布随机数**指令**

泊松分布随机数 (< 平均数 >)

说明: 由一个指定平均数 (均值) 的泊松分布生成一个随机数.

2.5.9 HyperGeometric. 超几何分布

释义: 超几何分布是统计学上一种离散概率分布. 它描述了从有限 N 个物件 (其中包含 M 个指定种类的物件) 中抽出 n 个物件, 成功抽出该指定种类的物件的次数 (不放回), 称为超几何分布. 在人教版的教材中, 一般是 N 件产品, 正品有 M 件, 从中抽取 n 件产品, 这样的分布称为超几何分布在 Geogebra 中, 超几何分布的几个参数的含义分别是:

- 总体容量: 产品总数 N
- 成功次数: 正品数 M
- : 样本容量: 抽取的样本数 n

指令

超几何分布 (< 总体容量 >, < 成功次数 >, < 样本容量 >, < true|false >)

说明: 第 4 个参数不加或者选 false 得到超几何分布的条形图, 选 true 则得到超几何分布累计图 (阶梯图).

指令

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 >)

说明: 同其他分布, true 计算 $P(x \leq v)$, false 计算 $P(x = v)$.

示例: 已知有 20 件产品, 合格品有 15 件, 抽取 5 件产品, 则合格的产品为 3 件的概率以及合格产品数 ≤ 3 的概率分别为:

```
超几何分布(20,15,5,3,false)      >>>0.29
超几何分布(20,15,5,3,true)       >>>0.37
```

Tips: 在 cas 区可以计算精确值.

2.5.10 LogNormal. 对数正态分布

释义: 数正态分布 (logarithmic normal distribution) 是指一个随机变量的对数服从正态分布, 则该随机变量服从对数正态分布. 对数正态分布从短期来看, 与正态分布非常接近. 但长期来看, 对数正态分布向上分布的数值更多一些.

指令

对数正态分布 (< 平均数 >, < 标准差 >, < 变量值 >)
 对数正态分布 (< 平均数 >, < 标准差 >, < 变量值 >, < 是否累积? true|false >)
 对数正态分布 (< 平均数 >, < 标准差 >, x, < 是否累积? true|false >)

说明: 用法同其他分布

2.5.11 BinomialDist. 二项分布

释义: 在 n 次独立重复的伯努利试验中, 设每次试验中事件 A 发生的概率为 p . 用 x 表示 n 重伯努利试验中事件 A 发生的次数, 则 x 的可能取值为 $0, 1, \dots, n$, 且对每一个 $k(0 \leq k \leq n)$, 事件 $x = k$ 即为“ n 次试验中事件 A 恰好发生 k 次”, 随机变量 x 的离散概率分布即为二项分布, 记作 $x \sim B(n, p)$, 其中 p 称为成功概率, n 为试验次数.

指令

二项分布 (< 试验次数 >, < 成功概率 >, < 是否累积? true|false >)

说明: 第三个参数不选或选 false, 则产生二项分布条形图, 选 true 产生累计分布图 (阶梯图)

指令

二项分布 (< 试验次数 >, < 成功概率 >, < 变量值 >, < 是否累积? true|false >)

说明: true 计算 $P(x \leq v)$, false 计算 $P(x = v)$.

示例: 抛硬币 5 次, 3 次正面朝上的概率为:

二项分布(5, 1/2, 3, false) >>> 0.31

Tips: 在 cas 区可得精确值 $\frac{5}{16}$.

2.5.12 Binomial. 二项式系数

指令

二项式系数 (< 数值 n >, < 数值 r >)

说明: 即组合数 C_n^r , 等同于指令 nCr.

2.5.13 Uniform. 均匀分布

指令

均匀分布 (< 下界 >, < 上界 >, < 变量值 >)
 均匀分布 (< 下界 >, < 上界 >, < 变量值 >, < 是否累积? true|false >)
 均匀分布 (< 下界 >, < 上界 >, x, < 是否累积? true|false >)

2.5.14 RandomUniform. 均匀分布随机数

指令

均匀分布随机数 (< 最小值 >, < 最大值 >)
 均匀分布随机数 (< 最小值 >, < 最大值 >, < 样本数量 >)

说明: 产生任意连续区间上的均匀随机数, 可以选择产生随机数的个数.**RandomBetween.** 区间随机数只能产生整数随机数.

示例: 产生 $0-2\pi$ 上的均匀随机数 10 个:

```
均匀分布随机数(0,2pi,10)    >>>{3.1, 0.17, 4.4, 4.02, 0.81, 5.1, 4.51, 1.87, 5.5, 0.01}
```

2.5.15 ChiSquared. 卡方分布

指令

卡方分布 (< 自由度 >, < 变量值 >)

说明: 计算卡方分布 CDF 在自变量 x 取自由度 v 时的概率 $P(x \leq v)$.

指令

卡方分布 (2, x , true|false)

说明: 第三个参数选 true 或不加第三个参数表示 CDF, false 输出 PDF

2.5.16 Cauchy. 柯西分布

指令

柯西分布 (< 中位数 >, < 尺度参数 λ >, < 变量值 >)

说明: 计算 CDF 在 $x=v$ 处的概率, 或给定 x 坐标左侧的柯西分布曲线下面积.

指令

柯西分布 (< 中位数 >, < 尺度参数 λ >, < 变量值 >, < 是否累积? true|false >)

说明: 第三个参数选 true 或不加第三个参数表示柯西分布的 CDF, false 输出柯西分布的 PDF

2.5.17 RandomDiscrete. 离散随机数

指令

离散随机数 (< 数字列表 >, < (相对) 概率列表 >)

说明: 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, 这两个列表的长度必须一致.

2.5.18 Logistic. 逻辑分布

指令

逻辑分布 (< 平均数 >, < 尺度参数 λ >, < 变量值 >)

逻辑斯特分布 (< 平均数 μ >, < 尺度参数 λ >, < 变量值 >, < 是否累积? true|false >)

说明: 类似其他分布, 分别返回函数值 $P(x \leq v)$, CDF, PDF

2.5.19 InverseFDistribution. 逆 F 分布

指令

逆 F 分布 (< 分子自由度 >, < 分母自由度 >, < 概率 p >)

说明: 计算 F 分布中使得 $P(xv) = p$ 的 v , 即 F 分布的逆运算.

2.5.20 InverseTDistribution. 逆 t 分布

指令

逆 t 分布 (< 自由度 >, < 概率 p >)

说明: 计算 t 分布中使得 $P(xv) = p$ 的 v .

2.5.21 InverseBeta. 逆贝塔分布

指令

逆贝塔分布 (< 形状参数 α >, < 尺度参数 β >, < 概率 p >)

说明: 计算 Beta 分布中使得 $P(xv) = p$ 的 v .

2.5.22 InversePoisson. 逆泊松分布

指令

逆泊松分布 (< 平均数 >, < 概率 >)

说明: 返回最小整数 n , 使得 $P(xnp)$, 其中 p 是概率, x 是 Poisson 随机变量, 其中均值 λ .

2.5.23 InverseHyperGeometric. 逆超几何分布

指令

逆超几何分布 (< 总体容量 >, < 成功次数 >, < 样本容量 >, < 概率 >)

说明: 返回最小整数 n , 使得 $P(x \leq n) = p$ 其中 p 是概率, x 是由总体大小、成功次数和样本大小给出的超几何随机变量.

2.5.24 InverseLogNormal. 逆对数正态分布

指令

逆对数正态分布 (< 平均数 >, < 标准差 >, < 概率 >)

说明: 当给定均值 μ 和标准偏差 σ 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 $P(x \leq t) = p$ 的 t . 概率 p 必须在 (0,1) 内.

2.5.25 InverseBinomial. 逆二项分布

指令

逆二项分布 (< 试验次数 >, < 成功概率 >, < 概率 >)

说明: 返回最小整数 n , 使得 $P(x \leq n)$, 其中 p 是概率, x 是由试验次数、成功概率给出的二项分布随机变量.

2.5.26 InverseBinomialMinimumTrials. 逆二项最小实验

指令

逆二项最小实验 (< 累积概率 >, < 成功概率 >, < 成功次数 >)

说明: 返回获得给定成功次数的最小试验数 n

2.5.27 InverseChiSquared. 逆卡方分布

指令

逆卡方分布 (< 自由度 >, < 概率 >)

说明: 当卡方分布 p 由自由度 d 给定时, 计算概率 p 的卡方分布累积分布函数的逆值. 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.

2.5.28 InverseCauchy. 逆柯西分布

指令

逆柯西分布 (< 中位数 >, < 尺度参数 λ >, < 概率 >)

说明: 当柯西分布 p 由中位数 m 和尺度参数 λ 给定时, 计算概率 p 的柯西分布累积分布函数的逆值. 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.

2.5.29 InverseLogistic. 逆逻辑分布

指令

逆逻辑分布 (< 平均数 >, < 尺度参数 λ >, < 概率 >)

说明:

当逻辑分布 p 由平均数 μ 和尺度参数 λ 给定时, 计算概率 p 的逻辑分布累积分布函数的逆值. 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.

2.5.30 InversePascal. 逆帕斯卡分布

指令

逆帕斯卡分布 (< 成功次数 >, < 成功概率 >, < 概率 >)

说明: 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 .

2.5.31 InverseZipf. 逆齐普夫分布

指令

逆齐普夫分布 (< 元素数量 >, < 指数 >, < 概率 >)



说明: 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 .

2.5.32 InverseWeibull. 逆威布尔分布

指令

逆威布尔分布 (< 形状参数 k >, < 尺度参数 λ >, < 概率 >).



说明: 当威布尔分布 p 由形状参数 k 和尺度参数 λ 给定时, 计算概率 p 的威布尔分布累积分布函数的逆值. 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.

2.5.33 InverseNormal. 逆正态分布

指令

逆正态分布 (< 平均数 >, < 标准差 >, < 概率 >)



说明: 计算函数 $\Phi^{-1}(p) \cdot \sigma + \mu$, 给定概率 p 、均值 μ 和标准偏差 σ , 其中 Φ^{-1} 是 $N(0, 1)$ 的累积分布函数 Φ 的倒数.

2.5.34 InverseExponential. 逆指数分布

指令

逆指数分布 (< 率参数 λ >, < 概率 >)



说明: 当指数分布 p 由率参数 λ 给定时, 计算概率 p 的指数分布累积分布函数的逆值. 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.

2.5.35 InverseGamma. 逆伽玛分布

指令

逆伽玛分布 (< 形状参数 α >, < 尺度参数 β >, < 概率 >)



说明: 当伽玛分布 p 由参数 α 和参数 β 给定时, 计算概率 p 的伽玛分布累积分布函数的逆值. 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.

2.5.36 Pascal. 帕斯卡分布

指令

帕斯卡分布 (< 成功次数 >, < 成功概率 >)

帕斯卡分布 (< 成功次数 >, < 成功概率 >, < 是否累积? true|false >)

帕斯卡分布 (< 成功次数 >, < 成功概率 >, < 变量值 >, < 是否累积? true|false >)



说明: 参见二项分布等其他分布

2.5.37 Zipf. 齐普夫分布

指令

齐普夫分布 (< 元素数量 >, < 指数 >)

说明: 返回齐普夫条形图. 参数的含义:

- 元素数量: 研究元素的数量
 - 指数: 分布的指数特性.
- 其他用法参见其他分布

2.5.38 RandomBetween. 区间随机数

指令

区间随机数 (< 最小整数 >, < 最大整数 >)

说明: 生成一个在最小值与最大值之间的整数(含端点), 可以加第三个参数 true|false, true 随机数不更新, false 随机数更新.

Tips: 英文 RandomBetween 指令在代数区也可以用 Random 输入.

2.5.39 Triangular. 三角形分布

释义: 在概率论与统计学中, 三角形分布是低限为 a 、众数为 c 、上限为 b 的连续概率分布. 密度分布分布函数 (PDF) 的定义是:

$$f(x | a, b, c) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)} & \text{for } a \leq x \leq c, \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{for } c < x \leq b. \end{cases}$$

指令

三角形分布 (< 下界 >, < 上界 >, < 众数 >, < 变量值 >, true|false)

说明: true 计算 $P(x \leq v)$, false 计算 $P(x=v)$.

指令

三角形分布 (< 下界 >, < 上界 >, < 模式 >, x , < 是否累积? true|false >).

说明: 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.

 注意: 软件将英文参数中的 Mode 翻译为模式, 让人难以理解, 笔者认为这里应当翻译为众数.

2.5.40 RandomPolynomial. 随机多项式

指令

随机多项式 (< 次数 >, < 最小系数 >, < 最大系数 >)

说明: 按照给定的相关数值随机产生一个多项式.

2.5.41 RandomBinomial. 随机二项分布数

指令

随机二项分布数 (< 试验次数 >, < 概率 >)

说明: 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.

2.5.42 RandomPointIn. 随机内点

指令

随机内点 (< 区域 >)

随机内点 (< 点列 >)

随机内点 (< x 最小值 >, < x 最大值 >, < y 最小值 >, < y 最大值 >)

说明: 分别在区域 (多边形, 二次曲线)、点列、限定横坐标的最小最大值, 纵坐标的最小最大值的长方形区域内产生随机的内点.

2.5.43 Weibull. 威布尔分布

指令

威布尔分布 (< 形状参数 k >, < 尺度参数 λ >, < 变量值 >)

威布尔分布 (< 形状参数 k >, < 尺度参数 λ >, < 变量值 >, < 是否累积? true/false >)

威布尔分布 (< 形状参数 k >, < 尺度参数 λ >, x , < 是否累积? true/false >)

2.5.44 Normal. 正态分布

正态分布 $x \sim N(\mu, \sigma)$

指令

正态分布 (< 平均数 >, < 标准差 >, < 变量值 v >)

说明: 计算 $P(x \leq v)$

指令

正态分布 (< 平均数 >, < 标准差 >, x , < 是否累积? true/false >).

说明: 如果累积是 true, 创建主变量 x 给定参数平均数 μ 和标准差 σ 的正态分布累积概率函数 $erf(x)$, 否则创建正态分布概率函数, 默认为 true.

示例: 创建平均值为 0, 方差为 4 的正态分布密度曲线: 由于方差为 4, 所以标准差为 2, 故输入指令:

正态分布(0,2,x,false)

>>>生成密度曲线

2.5.45 RandomNormal. 正态分布随机数

指令

正态分布随机数 (< 平均数 >, < 标准差 >)

说明: 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.

2.5.46 Exponential. 指数分布

指令

指数分布 (< 率参数 λ >, < 变量值 >)

指数分布 (< 率参数 λ >, < 变量值 >, < 是否累积? true|false >)

指数分布 (< 率参数 λ >, x , < 是否累积? true|false >)

2.5.47 Gamma. 伽玛分布

指令

伽玛分布 (< 形状参数 α >, < 尺度参数 β >, < 变量值 >)

伽玛分布 (< 形状参数 α >, < 尺度参数 β >, < 变量值 >, < 是否累积? true|false >)

伽玛分布 (< 形状参数 α >, < 尺度参数 β >, x , < 是否累积? true|false >)

2.6 Function 函数与微积分

2.6.1 PartialFractions. 部分分式

指令

部分分式 (< 函数 >).

说明: 如果可能的话, 给出给定函数的变量的部分分式. 函数的图形会在图形视图中绘制出来

2.6.2 ParametricDerivative. 参数导数

指令

参数导数 (< 参数曲线 >).

说明: 返回一条新的参数曲线, $\left(x(t), \frac{y'(t)}{x'(t)}\right)$

示例:

参数导数(曲线(2t, t², t, 0, 10))

>>> 曲线(2t, 2t/2, t, 0, 10)

解释: $y'(t) = 2t$, $x'(t) = 2$, 故结果为 $(2t, 2t/2)$. 在 cas 中会消去参数 t , 得到 x, y 的表达式, 由于 $x = 2t, y = t$, 故得到结果为 $x - 2y = 0$.

CAS 指令

参数导数 (曲线 $(2t, t^2, t, 0, 10)$)
 $\gg 2x - y = 0$



2.6.3 Derivative. 导数

指令

导数 (函数)



说明: 返回函数的导函数, 也可以用 f' 求导数.

指令

导数 (< 函数 >, < 阶数 n >)



说明: 返回函数的 n 阶导数.

指令

导数 (< 参数曲线 >)



说明: 返回参数曲线的导数 $(x'(t), y'(t))$

示例:

导数 (曲线 $(\cos(t), t \sin(t), t, 0, \pi)$) \gg 曲线 $(-\sin(t), \sin(t) + t \cos(t), t, 0, \pi)$

2.6.4 Iteration. 迭代

指令

迭代 (< 函数 f >, < 起始值 >, < 迭代次数 >)



说明: 使用给定的起始值迭代函数 n 次, 得到最后一次迭代得到的对象, 类似几何画板中的最终迭代, 不保留中间迭代的象, 如果要保留所有产生的迭代象, 则使用 `IterationList`. 迭代列表.

示例:

$f(x) = x/2$
 迭代 ($f, 16, 4$) \gg 1

说明: 16 为初象, 按照每次除以 2 的函数迭代, 即 $16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$.

指令

迭代 (< 表达式 >, < 变量 >, < 起始值 >, < 迭代次数 >)



说明: 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)

示例:

迭代 ($p^2, p, \{2\}, 4$) \gg 65536

说明: 起始值需要写在里面, 迭代的变量尽量避免使用 x, y, z .



注意: 每次迭代, 都有输入对象和输出对象, 这两者必须有相同的类型 (点和点, 数和数, 多边形和多边形...), 即输入的对象是点, 则输出对象也必须为点, 因为这次迭代的输出对象要作为下一次迭代的输入对象, 只有两者的类型

相同,才能把迭代进行下去,比较特殊的是输入对象和输出对象都是列表的情况,做分形的时候经常就要用到这种情况.

说明: 迭代也可以有多个迭代变量,即所谓的多元迭代,比较典型的有斐波那契数列的迭代,即 $a_{n+2} = a_n + a_{n+1}$.

```
迭代(a_n + a_{n+1}, a_n, a_{n+1}, {1, 1}, 5)    >>>8
```

为了帮助大家理解,这个指令中我们采用了 a_n, a_{n+1} 作为变量(读者理解了后可以用 p,q,s,t 之类简化的名称).

- 第 k 次迭代得到的象是 a_{k+1} , 第 0 次迭代的结果是 $a_1=1$, 第 1 次迭代的结果为 $a_2=1$, 第 2 次迭代 $=a_1+a_2 = 2$, 依次类推, 第 5 次迭代的结果为 $a_6=8$, 迭代的具体过程见下表

第 k 次迭代	表达式	输出象	变量 1	变量 2	输出象
0	a_1	a_1	1		1
1	a_2	a_2	1	1	1
2	$a_1 + a_2$	a_3	1	1	2
3	$a_2 + a_3$	a_4	1	2	3
4	$a_3 + a_4$	a_5	2	3	5
5	$a_4 + a_5$	a_6	3	5	8

如果将指令改为迭代列表,则输出所有的象.

```
迭代列表(p+q,p,q,{1,1},5)    >>>{1,1,2,3,5,8}
```

说明: 更复杂的迭代,含有序号的迭代,请读者自行查阅资料.

Tips: 在 GeoGebra 中,只有用到递推的问题需要用迭代,知道通项公式的数列,点列等问题可以使用序列和映射指令,部分迭代的功能也可以使用总和指令实现.

2.6.5 IterationList. 迭代列表

语法和迭代一致,区别是迭代列表保留所有的迭代象.如果迭代的次数为 n ,迭代列表得到的列表中元素的个数为 $n+1$,因此想要得到 n 个元素,迭代的次数应当设为 $n-1$.

示例: 求斐波那契数列的前 10 项.

```
迭代列表(p+q,p,q,{1,1},9)    >>>{1, 1, 2, 3, 5, 8, 13, 21, 34, 55}
```

2.6.6 NIntegral. 定积分

指令

定积分 (< 函数 f >)

说明: 计算函数 f 的不定积分 $F(x) + c$, 其中常数项 c 取 0, 并绘制出 $F(x)$ 的函数.

指令

定积分 (< 函数 >, <x-起始值 a>, <x-终止值 b>)

说明: 求函数在起始值和终止值之间的定积分 $\int_a^b f(x)dx$, 并绘制出图形(可用于涂色): 函数图像与 x 轴以及 $x=a, x=b$, 的图形.

指令

定积分 (< 函数 >, <x-起始值 >, <y 起始值 >, <x-终止值 >)

说明: 绘制出 $f(x)$ 的不定积分 $F(x) + c$ 在 x 的起始值和终止值之间的函数, 其中 c 由起始的 x 值和 y 值确定.

官网给出的

示例:

定积分(sin(x)/x, , 1, 2)

说明: 先计算 $\frac{\sin x}{x}$ 的不定积分 $F(x) = Si(x) + c$, 即正弦积分函数加 1 个常数项 c , 接下来代入 x, y 的初始值 π 和 1, 算出 c 的值, 并绘制 $F(x)$ 在 $[\pi, 2\pi]$ 上的图像.

笔者再提供 1 个简单的

示例:

积分(x,1,1,2)

先计算 $f(x) = x$ 的不定积分 $F(x) = \frac{x^2}{2} + c$, 由于 x 的初始值为 1, y 的初始值也为 1, 即 $F(1) = 1$, 解得 $c = 0.5$, 由于 x 的终止值为 2, 故画出的图像即 $F(x) = \frac{x^2}{2} + 0.5, 1 \leq x \leq 2$, 两者的图像是重合的.

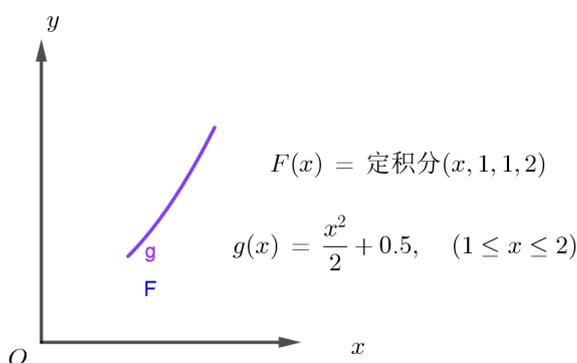


图 2.19

2.6.7 Degree. 多项式次数

2.6.8 Polynomial. 多项式函数

指令

多项式函数 (< 函数 >)

说明: 得出多项式函数的展开式.

指令

多项式函数 (< 点列 >)

说明: 通过给定的 n 个点的点列创建 $n-1$ 阶的插值多项式, 此时等同于多项式拟合指令 (fitpoly).

2.6.9 NInvert. 反函数

指令

反函数 (< 函数 >)

说明: 返回函数的反函数并画出图像, 但不给出反函数的解析式. 如果要给出解析式, 请使用逆反指令.

注意:

1. 不考虑函数的定义域和治愈, 比如 $y = x^2, y = \sin x$.
2. 这个指令对于只有 1 个 x 的函数运行速度更快, 比如 $y = (x + 1)^3$ 要优于 $y = x^3 + 3x^2 + 3x + 1$, 并且某些函数展开和不展开求得的反函数也可能不同, 如图 2.20.

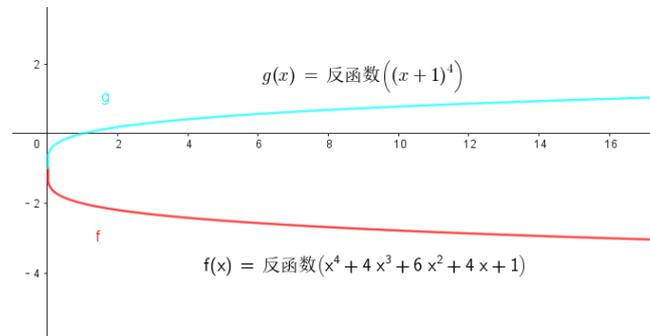


图 2.20

2.6.10 Denominator. 分母

指令

分母 (< 数字 >)

说明: 对于一个有理数返回它的分母, 由于采用了数值计算的方法, 把分母限制在较小的数字. 对于无理数, 则等于其连分式的分母.

示例: $\pi \approx 3 + \frac{1}{7 + \frac{1}{15 + \frac{1}{1 + \frac{1}{292}}}} = \frac{103993}{33102}$, 故

分母(pi) >>>33102

指令

分母 (< 函数 >)

说明: 返回一个函数的分母.

参见 [Numerator. 分子](#) 和 [ContinuedFraction. 连分式](#).

2.6.11 Numerator. 分子

指令

分子 (< 函数 >)

说明: 返回 1 个函数的分子.

指令

分子 (< 数 >)

说明: 对于一个有理数返回它的分子, 由于采用了数值计算的方法, 把分母限制在较小的数字. 对于无理数, 则等于其连分式的分子 (精度为 10^{-8}).

如 [Denominator. 分母](#) 示例中分析, π 的分子为 103993.

2.6.12 ComplexRoot. 复数根

指令

复数根 (< 多项式 >)

说明: 计算多项式关于 x 的复数根, 绘图区创建对应的点.

2.6.13 TurningPoint. 拐点

指令

拐点 (< 多项式 >)

说明: 返回多项式的所有拐点.

2.6.14 Normalize. 归一化

指令

归一化 (< 数字列表 >)

说明: 返回一个列表, 其中包含给定数字的规范化形式

归一化的公式: 假设列表中的数最大值为 x_{Max} , 最小值为 x_{Min} , 则其中的某个数 x 所对应的公式为:

$$x \rightarrow \frac{x - x_{Min}}{x_{Max} - x_{Min}}$$

指令

归一化 (< 点列 >)

说明: 返回给定归一化格式的点列. 横坐标, 纵坐标分别归一化处理.

示例:

```
归一化({(1,5),(2,4),(3,3),(4,2),(5,1)}) >>>{(0,1),(0.25,0.75),(0.5,0.5),(0.75,0.25),(1,0)}
```

2.6.15 Function. 函数

指令

函数 (< 数字列表 >)

说明: 列表前两个数值确定了初始的 x 值与终止的 x 值, 即函数的定义域, 剩余的数值是等距横坐标所对应的 y 值. 返回经过由列表产生的点的一个折线型函数, 并显示为手绘函数.

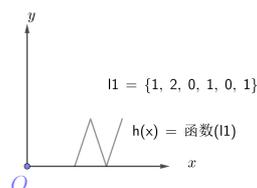


图 2.21

示例:

函数 [1, 2, 0, 1, 0, 1] >>> freehand(x)

说明: 列表的前 2 个元素是 1 和 2, 因此定义域为 [1,2], 后面有 0,1,0,1 这 4 个数, 因此对应应有 4 个点, 把区间 [1,2] 分成 4 个点, 即 $1, \frac{4}{3}, \frac{5}{3}, 2$, 对应的纵坐标分别为 0,1,0,1, 故由 4 个点 $(1,0), (\frac{4}{3},1), (\frac{5}{3},0), (2,0)$, 再返回过这 4 个点的折线函数 (图 2.21).

指令

函数 (< 函数 f>, <x-起始值 a>, <x-终止值 b>)

说明: 生成函数 f 在数值区间上的函数图象, 等同于指令:

如果 ($a \leq x \leq b, f(x)$)

指令

函数 (< 代数式 >, < 参变量 1>, < 起始值 >, < 终止值 >, < 参变量 2>, < 起始值 >, < 终止值 >)

说明: 在 3D 区绘制出二元函数的部分图像, 范围由给定参变量的起始值和终止值决定.

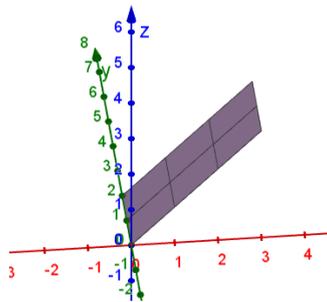


图 2.22

示例:

函数 (u,u,0,3,v,0,2) >>> a(u,v)=u

说明: 返回函数 $z = x + 0y$, ($0 \leq x \leq 3, 0 \leq y \leq 2$) 的图像

2.6.16 Integral. 积分

指令

积分 (< 函数 >)

说明: 出函数关于主变量的不定积分

示例:

积分 (x^3) >>> $\frac{1}{4}x^4$, 并绘制出图像

指令

积分 (< 函数 >, < 变量 >)

说明: 给出函数关于指定变量的偏积分.

示例:

积分($x^3 + 3x y, x$) $\gg>\frac{1}{4}x^4 + \frac{3}{2}x^2y$, 并在3D区绘制图像.

指令

积分(<函数>, <x-积分下限>, <x-积分上限>, <是否给出积分值?true|false>)

说明: 给出函数的定积分, 并选择是否给出积分值, 不选择则默认为给出积分.

CAS 指令

积分(<函数>, <变量>, <起始值>, <终止值>)

说明: 在 cas 区可以指定其他字母为变量.

2.6.17 IntegralBetween. 积分介于**指令**

积分介于(<函数 1>, <函数 2>, <x-积分下限>, <x-积分上限>).

说明: 给出两函数之差的定积分, 并绘制出两函数之间的阴影部分图像, 可以加第五个参数 true|false 选择是否显示积分值.

示例:

积分($\sin(x), \cos(x), 0, \pi$) $\gg>>2$

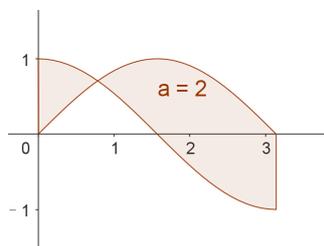


图 2.23

2.6.18 Limit. 极限**指令**

极限(<函数>, <数值>)

说明: 计算指定主函数变量趋向某值时函数极限.

Tips: cas 区可以计算含有参数的极限. 另外, 不是所有极限都有能力求出.

2.6.19 Extremum. 极值点**指令**

极值点(<多项式>)

说明: 得出多项式函数的所有极值点

指令

极值点 (< 函数 >, < x-起始值 >, < x-终止值 >)

说明:

给出连续函数在开区间上的极值点

2.6.20 Asymptote. 渐近线

指令

渐近线 (< 对象 >)

说明: 如果对象是双曲线, 则给出双曲线的 2 条渐近线, 如果对象是函数或隐式曲线, 则尝试以列表的形式, 给出函数的所有渐近线.

示例:

渐近线($x^2 - y^2 / 4 = 1$) $\ggg -2x + y = 0, 2x + y = 0$
 渐近线($(x^3 - 2x^2 - x + 4) / (2x^2 - 2)$) $\ggg \{y = 0.5x - 1, x = -1, x = 1\}$
 渐近线($x^3 + y^3 + y^2 - 3x = 0$) $\ggg \{x + y = -0.33\}$

2.6.21 SolveODE. 解常微分方程

指令

解常微分方程 (< f(x,y) >)

说明: 尝试求一阶常微分方程的精确解

示例:

解常微分方程 ($2x/y$) $\ggg f(x) = \sqrt{2}\sqrt{-c_1 + x^2}$
 并生成 1 个滑动条 $c_1 = 1$

指令

解常微分方程 (< f(x,y) >, < f 上的点 >)

说明: 尝试寻找一阶常微分方程精确的解, 并使其解通过指定的点

示例:

解常微分方程 ($y/x, (1, 2)$) $\ggg y = 2x$

指令

解常微分方程 (< f'(x,y) >, < 起始 x >, < 起始 y >, < 终止 x >, < 步长 >)

说明:

由给出 x 的起点、终点和步长求解一阶常微分方程

示例:

解常微分方程 ($-x*y, x(A), y(A), 5, 0.1$) \ggg 使用 A 点作为起点求解常微分方程, 返回一条轨迹.

 注意:

1. 用长度 [\langle 轨迹 \rangle], 可以获取轨迹中点的个数.
2. 最前元素 (\langle 轨迹 \rangle , \langle 数字 $n \rangle$) 可以提取轨迹的前 n 个点, 提取所有的点, 一般用最前元素 (\langle locus \rangle , 长度 (\langle locus \rangle)-1).
3. 要找到“反向”解决方案, 只需为结束 x 输入一个负值, 例如: 解常微分方程 ($-x*y, x(A), y(A), -5, 0.1$)

指令

解常微分方程 ($\langle b(x) \rangle, \langle c(x) \rangle, \langle f(x) \rangle, \langle$ 起始 $x \rangle, \langle$ 起始 $y \rangle, \langle S$ 起始 $y' \rangle, \langle$ 终止 $x \rangle, \langle$ 步长 \rangle)

说明: 解二阶常微分方程 $y'' + b(x)y' + c(x)y = f(x)$

注意: 返回的结果都是轨迹, 当前的算法是基于 **Runge-Kutta** 数值方法
在 cas 区的使用方法, 见下面几例.

CAS 指令

解常微分方程 ($y' = y/x$)

$\gg y = c_2 x$

说明: 对于 y 的一阶导数和二阶导数, 您可以分别使用 y' 和 y'' .

CAS 指令

解常微分方程 ($y' = y/x, (1, 2)$)

$\gg y = 2x$

CAS 指令

解常微分方程 ($y'' - 3y' + 2 = x, (2, 3), (1, 2)$) $\gg y = \frac{-9x^2 e^3 + 30x e^3 - 32(e^3)^2 + 138e^3 + 32e^{3x}}{54e^3}$

说明: 语法是: 解常微分方程 (\langle 方程 \rangle, \langle f 上的点 \rangle, \langle f' 上的点 \rangle), 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

CAS 指令

解常微分方程 ($v' = v/w, v, w, (1, 2)$)

$\gg v = 2w$

说明: 语法是: 解常微分方程 (\langle 方程 \rangle, \langle 因变量 \rangle, \langle 自变量 \rangle, \langle f 上的点 \rangle), 尝试寻找一或二阶过给定点的常微分方程 (ODE) 的精确解.

CAS 指令

解常微分方程 (\langle 方程 \rangle, \langle 因变量 \rangle, \langle 自变量 \rangle, \langle f 上的点 \rangle, \langle f' 上的点 \rangle)

示例:

CAS 指令

解常微分方程 ($v' = v/w, v, w, (1, 2), (0, 2)$)

$\gg v = 2w$

2.6.22 NSolveODE. 解常微分方程组

指令

解常微分方程组 (< 导数列表 >, < x 坐标初值 >, < y 坐标初值列表 >, < x 坐标终止值 >)

说明: 解 (用数值求解方法) 常微分方程组.

示例:

```
f'(t, f, g, h) = g
g'(t, f, g, h) = h
h'(t, f, g, h) = -t h + 3t g + 2f + t
解常微分方程组({f', g', h'}, 0, {1, 2, -2}, 10)
```

```
x1'(t, x1, x2, x3, x4) = x2
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)
```

示例: 动态钟摆 (可在按钮中输入代码, 并单击按钮完成作图)

```
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(numericalIntegral1)
c = 滑动条(0, 1, 1 / len, 1, 100, false, true, true, false)
x1 = l sin(y(描点(numericalIntegral1, c)))
y1 = -l cos(y(描点(numericalIntegral1, c)))
A = (x1, y1)
线段((0, 0), A)
启动动画()
```

2.6.23 RectangleSum. 矩形法则

指令

矩形法则 (< 函数 >, < x-起始值 >, < x-终止值 >, < 矩形数量 >, < 矩形起始位置 0-左和 1-右和 >)

说明: 用 n 个矩形计算函数在区间内的矩形 (面积) 和, 矩形的左上顶点由一个分数 (小数) d ($0 \leq d \leq 1$) 控制, 如图 (). 当 $d=0$ 时等于 **LeftSum**. **左和** 指令, 而当 $d=1$ 时则计算指定函数的 “右和 (上和)”.

注意: 1. 命令还会绘制左侧总和的矩形. 2. 此命令设计为视觉辅助工具, 因此如果矩形数量过多, 则不会给出准确的答案.

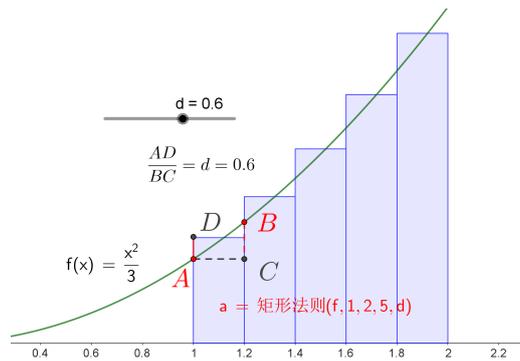


图 2.24

参见LowerSum. 下和 LeftSum. 左和 UpperSum. 上和

2.6.24 RemovableDiscontinuity. 可去间断点

指令

可去间断点(<函数>)

说明: 返回 1 个函数的可去间断点, 释义: 可去间断点指的是左右极限存在且相当的间断点.

示例:

```
可去间断点((3-x)/(2 x^(2)-6 x))    >>>(3,-0.17)
```

2.6.25 Root. 零点

指令

零点(<多项式>)

说明: 得出多项式全部的根, 也就是函数图象与 x 轴的交点.

示例:

```
零点(0.1*x^2 - 1.5*x + 5)          >>>A=(5,0),B=(10,0)
```

指令

零点(<函数>,<x-初值>)

说明: 以 x-初值作为初始点, 用数值迭代法求函数的 1 个零点.

指令

零点(<函数>,<x-初值>,<x-终止值>)

说明: 在区间内用数值迭代法求函数的零点.

注意: 在 cas 区, 这个指令是Solve. 精确解的特例.

2.6.26 Roots. 零值点

指令

零值点 (<函数>, <x-起始值>, <x-终止值>)

说明: 计算给定区间内函数的零点, 要求该函数在区间内必须是连续的. 因为此算法是数值式的, 在某些情况下可能无法找到所有零点.

参见 **Root. 零点**.

2.6.27 RootList. 零值点列

指令

零值点列 (<数字列表>)

说明: 将数字转换成 x 轴上的点.

示例:

零值点列({3, 4, 5, 2, 1, 3}) >>>{(3, 0), (4, 0), (5, 0), (2, 0), (1, 0), (3, 0)}

等同于指令:

(<列表>, 0)

2.6.28 PathParameter. 路径值

指令

路径值 (<路径上的点>)

说明: 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.

在 GeoGebra 里面, 尽管路径值的定义形式多样, 但大多都是基于以下两个函数来定义的:

$$f(x) = \frac{x}{1 + |x|}$$

$$\phi(X, A, B) = \frac{\overrightarrow{AX} \cdot \overrightarrow{AB}}{|AB|^2}$$

当 X, A, B 三点共线时, $\phi(X, A, B)$, 即仿射比 $\lambda(A, B, X)$.

直线 AB	$\frac{f(\phi(X, A, B))+1}{2}$
射线 AB	$f(\phi(X, A, B))$
线段 AB 或二维向量 \overrightarrow{AB}	$\phi(X, A, B)$
以 C 为圆心, r 为半径的圆	点 $X = C + (r \cdot \cos(\alpha), r \cdot \sin(\alpha))$, $\alpha \in (-\pi, \pi)$ 时, 路径值为 $\frac{\alpha+\pi}{2\pi}$
以 C 为中心, 以 \vec{a}, \vec{b} 为轴的椭圆	点 $X = C + \vec{a} \cdot \cos(\alpha) + \vec{b} \cdot \sin(\alpha)$, 当 $\alpha \in (-\pi, \pi)$ 时, 路径值为: $\frac{\alpha}{2\pi}$
双曲线	点 $X = C \pm \vec{a} \cdot \cosh(t) + \vec{b} \cdot \sinh(t)$ 路径值为: $\frac{f(t)+1}{4}$ or $\frac{f(t)+3}{4}$
顶点为 V 对称轴为 \vec{v} 的抛物线	点 $V + \frac{1}{2}p \cdot t^2 \cdot \vec{v} + p \cdot t \cdot \vec{v}^\perp$ 路径值为: $\frac{f(t)+1}{2}$.
折线 $A_1 \dots A_n$	如果 X 在 $A_k A_{k+1}$ 上, 则 X 的路径值为: $\frac{k-1+\phi(X, A_k, A_{k+1})}{n-1}$
多边形 $A_1 \dots A_n$	如果 X 在 $A_k A_{k+1}$ 上, (假定 $A_{n+1} = A_1$), X 路径值为: $\frac{k-1+\phi(X, A_k, A_{k+1})}{n}$
路径列表 $L = p_1 \dots, p_n$	如果 X 在 p_k 上, 且 X 相对于 p_k 的路径值为: t_r , 则 X 的路径值为: $\frac{k-1+t_r}{n}$
点列 $L = A_1, \dots, A_n$	A_k 的路径值为: $\frac{k-1}{n}$. 指令: 描点 $[L, t]$ 返回点 $A_{[tn]+1}$
参数曲线	$f(X)$ 是参数曲线 $f(t)$, $a \leq t \leq b$ 上一点, 则路径值为: $\frac{X-a}{b-a}$
隐式多项式, 轨迹	没有可用的公式.
圆弧	\widehat{AB} 上一点 X 的路径值为: $\frac{\widehat{AX}}{\widehat{AB}}$

 **注意:** 本表格来自官网, 笔者对其进行了翻译, 并做了少许修改, 比如折线和多边形的路径值, 与原网页有不同, 部分参考了肖建伟老师的 <GeoGebra 几何对象路径值 (PathParameter) 的理解与应用> 一文. GeoGebra 的路径值与几何画板的点值在定义上有很大不同, 您在使用中需要注意区别.

2.6.29 OsculatingCircle. 密切圆

指令

密切圆 (< 点 >, < 对象 >)

说明: 得出对象 (函数、参数曲线、二次曲线) 在指定点上的密切圆.

指令

密切圆 (< 点 >, < 函数 >)

说明: 得出函数在指定点上的密切圆.

 **注意:** 自 GeoGebra5, 这个指令也适于二次曲线.

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 (Osculating circle) 或曲率圆. 曲线在 P 点的曲率, 便是密切圆半径的倒数.

2.6.30 SVD. 奇异值分解

指令

奇异值分解 (< 矩阵 >)

说明: 返回矩阵的分解值 (包含 3 个矩阵的列表)

示例:

```
奇异值分解({{3, 1, 1}, {-1, 3, 1}}) >>>{{{0.70711, -0.70711}, {0.70711, 0.70711}}, {{3.4641, 0}, {0, 3.16228}}, {{0.40825, -0.89443}, {0.8165, 0.44721}, {0.40825, 0}}}
```

2.6.31 PlotSolve. 求解绘图

指令

求解绘图 (x 的方程)

说明: 求解主变量的给定方程, 并返回所有解的列表和图形视图中的图形输出.

示例:

```
求解绘图(x^2 = 4x) >>>{(0,0),(4,0)}
```

2.6.32 Curvature. 曲率

指令

曲率 (< 点 >, < 函数 | 曲线 >)

说明: 计算函数 (曲线) 在给定点处的曲率.

示例:

```
曲率((0,0),x^2) >>>2
曲率((0,0),曲线(cos(t),sin(2t),t,0,)) >>>0
```

2.6.33 CurvatureVector. 曲率向量

指令

曲率向量 (< 点 >, < 对象 >)

说明: 得出对象 (函数、曲线、圆锥曲线) 在给定点处的曲率向量.

示例:

```
曲率向量((0,0),x^2) >>> 向量(0,2)
曲率向量((0,0),Curve(cos(t),sin(2t),t,0,)) >>> 向量(0,0)
曲率向量((-1,0),Conic({1,1,1,2,2,3})) >>> 向量(0,-2)
```

2.6.34 CurveCartesian. 曲线

指令

曲线 (<x(t)>,<y(t)>,< 参数值 t>,<t-起始值>,<t-终止值>)

说明: CurveCartesian 也可缩写为 Curve, 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) 内的参数曲线.

说明: 曲线指令和??Surface 曲面] 联系密切. 曲面有 2 个参数, 曲线有 1 个参数, 曲线的表达式写法, 既可以用直角坐标, 也可以用极坐标也可以用球坐标, 以下都是合规的写法.

```
曲线(s,0,s,0,1)
曲线((1;s),s,0,2pi)
曲线(A+s u,s,0,1) 其中u是向量
曲线((1;t;t),t,0,pi)
曲线(a(s,1),s,0,2pi) 其中a是一个曲面.
```

 **注意:** <t-终止值> 必须大于或等于 <t-起始值>, 而且都必须是有有限值 (不可以用无穷大作为范围). 注: x、y 和 z 不允许作为参数变量.

2.6.35 TrigCombine. 三角合并

指令

三角合并 (<表达式>)

说明: 将三角项的乘积合并为和, 或将三角项的和合并为仅包含三角函数的表达式.

指令

三角合并 (<代数式>, <目标函数>)

说明: 将三角表达式的项合并为一个等效表达式, 仅包含给定的目标函数.

示例:

三角合并($\sin(x) + \cos(x)$, $\sin(x)$) $\gg \sqrt{2} \sin(x + \frac{1}{4}\pi)$

2.6.36 TrigSimplify. 三角化简

指令

三角合并 (<表达式>)

说明: 简化给定的三角函数表达式.

 **注意:** 该指令使用与代数区和 cas 区, 但计算能力较弱, 需谨慎使用.

2.6.37 TrigExpand. 三角展开

指令

三角展开 (<表达式>)

说明: 将变量之和的三角函数展开为单个变量的三角函数, 或将三角函数的乘积展开为线性表达式.

示例:

三角展开($\tan(x + y)$) $\gg \frac{\frac{\sin(x)}{\cos(x)} + \frac{\sin(y)}{\cos(y)}}{1 - \frac{\sin(x)}{\cos(x)} \cdot \frac{\sin(y)}{\cos(y)}}$

指令

三角展开 (<表达式>, <目标函数>)

说明: 将变量之和的三角函数扩展为包含 (如果可能) 单个变量的给定目标函数的表达式.

2.6.38 UpperSum. 上和

指令

上和 (<函数>, <x-起始值>, <x-终止值>, <矩形数量>)

说明: 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

2.6.39 IsVertexForm. 是否为顶点式

指令

是否为顶点式 (< 函数 >)

说明: 判断 1 个函数是否为顶点式, 是返回 true, 否返回 false.

示例:

是否为顶点式 $((x+2/3)^2-2/3^2)$ `>>>true`

2.6.40 NDerivative. 数值导数

指令

数值导数 (< 函数 >)

说明: 用数值计算得出给定函数的一阶导数.

2.6.41 TaylorSeries. 泰勒公式

指令

泰勒公式 (< 函数 >, < x 值 >, < 阶数 n >)

说明: 创建在给定 x 值时的 n 阶泰勒展开式.

注意: 阶数必须为自然数.

CAS 指令

泰勒公式 (< 表达式 >, < 变量 >, < 变量值 >, < 阶数 >)

说明: 在 cas 区可以指定哪个变量为主变量.

示例:

CAS 指令

泰勒公式 $(x^3 \sin(y), x, 3, 2) \Rightarrow 27 \sin(y) + 27 \sin(y) (x - 3) + 9 \sin(y) (x - 3)^2$
 泰勒公式 $(x^3 \sin(y), y, 3, 2) \Rightarrow x^3 \sin(3) + x^3 \cos(3) (y - 3) - x^3 \cdot \frac{\sin(3)}{2} (y - 3)^2$

2.6.42 TrapezoidalSum. 梯形法则

指令

梯形法则 (< 函数 >, < x-起始值 >, < x-终止值 >, < 梯形数量 >)

说明: 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.

2.6.43 Coefficients. 系数列表

指令

系数列表 (< 多项式 >)

说明: 得出多项式所有系数组成的列表.

指令

系数列表 (<圆锥曲线>)

说明: 返回圆锥曲线 $ax^2 + by^2 + c + dxy + ex + fy = 0$ 的系数列表 $\{a, b, c, d, e, f\}$

 注意: 无法用系数列表来获取直线一般式的系数, 但可以用 `x(<直线>)` 这样的方式去获取系数. 可以用来获取平面方程 $Ax + By + Cz + D = 0$ 的系数 $\{A, B, C, D\}$.

2.6.44 LowerSum. 下和

指令

下和 (<函数>, <x-起始值>, <x-终止值>, <矩形数量>)

说明: 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.

2.6.45 SlopeField. 斜率场

指令

斜率场 (< $f(x, y)$ >)

说明: 绘制微分方程 $\frac{dy}{dx} = f(x, y)$ 的斜率场.

指令

斜率场 (< $f(x, y)$ >, <数值 n >)

说明:

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 (40×40 个网格).

指令

斜率场 (< $f(x, y)$ >, <数值 n >, <长度倍增器 a >)

说明: 绘制微分方程 $\frac{dy}{dx} = f(x, y)$ 的斜率场. a 决定了每格中的线段长.

指令

斜率场 (< $f(x, y)$ >, <数值 n >, <长度倍增器 a >, < x 最小>, < y 最小>, < x 最大>, < y 最大>)

说明: 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

 注意: 使用“移动视窗”、“放大”和“缩小”工具, 观察变化.

2.6.46 Spline. 样条曲线

释义: 是一种分段 n 次多项式函数, 用于以平滑的方式连接数据点.

指令

样条曲线 (<点列>)

说明: 创建穿过所有点的三次样条.

指令

样条曲线 (< 点列 >, < 次数 ≥ 3 >)

说明: 创建按给定次数穿过所有点的样条曲线.

指令

样条曲线 (< 点列 >, < 阶数 ≥ 3 >, < 权重函数 >)

说明: 创建按给定次数穿过所有点的样条曲线. 权重函数 $w(x, y)$ 描述了相邻两个点 P_i, P_{i+1} 之间的曲线段应该如何过渡, 其中 $(x, y) = P_{i+1} - P_i$, 常用的权重函数有: $\sqrt{x^2 + y^2}, |x| + |y|$ 等

2.6.47 Factors. 因式

指令

因式 (< 多项式 >)

说明: 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. 因式按指数的次数升序排列.

示例:

因式 (x^8-1) $\ggg\{\{x^4+1, 1\}, \{x^2+1, 1\}, \{x+1, 1\}, \{x-1, 1\}\}$

指令

因式 (< 数值 >)

说明: 给出形式为素数, 指数的列表 (矩阵), 素数按升序排序. 在 CAS 中, 可以返回参数的因式.

CAS 指令

因式 (a^8-1)

$\Rightarrow: \begin{pmatrix} a-1 & 1 \\ a+1 & 1 \\ a^2+1 & 1 \\ a^4+1 & 1 \end{pmatrix}$, 其中 a 为未定义参数.

2.6.48 ImplicitCurve. 隐式曲线

指令

隐式曲线 (< 点列 >)

说明: 创建通过由点集所指定点的隐式曲线. 列表的长度必须是 $\frac{n(n+3)}{2}$, 以得出次数为 n 的隐式曲线. 比如圆锥曲线是 2 次的隐式曲线, 需要 5 个点去确定.

指令

隐式曲线 ($f(x,y)$)

说明: 创建隐式曲线 $f(x, y) = 0$, $f(x,y)$ 必须为多项式函数.

2.6.49 ImplicitDerivative. 隐式微分

指令

隐式微分 (<f(x,y)>)

说明: 给出指定表达式的隐式微分

隐式微分(x+2y) >>> a(x,y)=-0.5

CAS 指令

隐式微分 (<表达式>,<因变量>,<自变量>)

示例:

CAS 指令

隐式微分(x²+y²,y,x) ⇒ $-\frac{x}{y}$

2.6.50 LimitAbove. 右极限

指令

右极限 (<函数>,<数值>)

说明: 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.

 注意: GeoGebra 不是所有的极限都能计算出来, 算不出来或没有极限的将返回未定义.

示例:

右极限(1/x,0) >>>∞

在 CAS 区可以指定其他字母为变量.

CAS 指令

右极限 (<表达式>,<变量>,<数值>)

2.6.51 LeftSum. 左和

指令

左和 (<函数>,<x-起始值>,<x-终止值>,<矩形数量 n>)

说明: 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)

2.6.52 LimitBelow. 左极限

指令

左极限 (<函数>,<数值>)

说明: 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.

参见LimitAbove. 右极限

2.7 Geometry. 几何

2.7.1 PenStroke

指令

```
PenStroke(< 点 >, ..., < 点 >)
```

说明: 没有中文翻译, 输入的点必须以坐标的形式, 不可以是已存在的点的名称, 返回一条画笔的轨迹, 具体是什么曲线, 仍需研究.

2.7.2 Radius. 半径

指令

```
半径(< 圆 >)
```

说明: 返回圆的半径.

2.7.3 Envelope. 包络

指令

```
包络(< 路径 >, < 点 >)
```

说明: 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.

示例: 一个梯子靠在墙上滑下来

注意: 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.

另请参阅 [Locus. 轨迹](#) 和 [LocusEquation. 轨迹方程](#) 指令.

2.7.4 Difference. 差集

指令

```
差集(< 多边形 1 >, < 多边形 2 >)
```

说明: 找出两个多边形的不同. 只有不是自相交的多边形, 此命令才有效.

2.7.5 Length. 长度

指令

```
长度(< 对象 >)
```

说明:

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).

2.7.6 OrthogonalLine. 垂线

英文也可以用 PerpendicularLine.

指令

垂线 (< 点 >, < 直线 | 线段 | 向量 >)

说明: 过一点做直线 (线段, 向量) 的垂线

3D 区指令

垂线 (点, < 平面 >)

说明: 过点作一个平面的垂线

 **注意:** 如果要做 xoz 平面的垂线, 用指令 “垂线 (< 点 >, $y=0$)” 时, 必须打开 3d 区视图, 并用鼠标在 3d 区点一下, 这样 $y=0$ 才会被识别平面, $y=0$ 会被认为是 x 轴.

指令

垂线 (< 直线 1 >, < 直线 2 >)

说明:

创建两条给定直线的公垂线 (与两直线都相交, 两直线不可以是平行线).

指令

垂线 (< 点 >, < 方向 1 >, < 方向 2 >)

说明: 过一点画与给定方向 (向量或线) 都垂直的直线.

3D 区指令

垂线 (< 点 >, < 直线 >, < 平面 xOy | 3D space >)

说明: 创建过给定点垂直于给定直线且平行于给定平面的直线.

2.7.7 Vertex. 顶点

指令

顶点 (< 圆锥曲线 | 不等式 | 多边形 >)

说明: 返回圆锥曲线, 不等式和多边形的顶点 (如果不能返回所有顶点, 建议在指令外面加上 {}).

指令

顶点 (< 线段 | 多边形 >, < 索引 n >)

说明: 返回线段或多边形的第 n 个顶点.

2.7.8 Polygon. 多边形

指令

多边形 (< 点列 >)

说明: 返回由点列中的点定义的多边形.

指令

多边形 (< 点 1>, ..., < 点 n>)

说明: 返回一个由给定点所确定的多边形.

指令

多边形 (< 点 1>, < 点 2>, < 顶点数 n>)

说明: 创建由点 1 和点 2 确定的正 n 边形.

3D 区指令

多边形 (< 点 1>, < 点 2>, < 顶点数 n>, < 方向 >)

说明: 创建 1 个由点 1 和点 2 确定且与给定方向平行的正 n 边形. 这里的方向建议采用平面或者平面的单位法向量.

示例:

```
A=(0,0,0)
B=(2,0,0)
p:y+z=3
poly1=多边形(A,B,4,p)
poly2=多边形(A,B,4,-单位法向量(p))
poly3=多边形(A,B,4,向量(单位法向量(p)))
```

多边形
 • poly1 = 多边形(A, B, 4, p)
 • poly2 = 多边形(A, B, 4, 向量(-单位法向量(p)))
 • poly3 = 多边形(A, B, 4, 向量(单位法向量(p)))
 平面
 • p: y + z = 3
 按钮
 • button1
 点
 线段

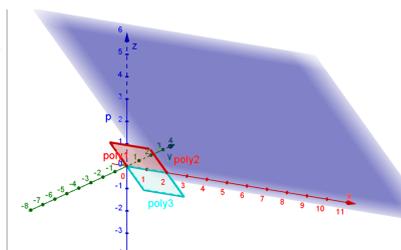


图 2.25

注意: 采用单位法向量更容易控制多边形的方向, 如果不用单位法向量而用法向量作为方向, 则可能出现不是正多边形情况. 另外要注意的是, 在指令中 < 向量 > 和 < 点 > 一般要严格区分的, 因此如果输入:”多边形 (A,B,4,(0,0,1))” 会出现错误的提示, 正确的指令是:”多边形 (A,B,4, 向量 ((0,0,1)))”

2.7.9 Direction. 方向向量

指令

方向向量 (< 直线 | 射线 | 线段 >)

说明: 得出直线 | 射线 | 线段的方向向量.

Tips: 直线 $Ax + By = C$ 的方向向量是 $(B, -A)$, 而斜截式 $y = kx + b$ 的方向向量为 $(1, k)$. 这里要注意一次函数和直线区别, 看清楚对象的类型到底是直线还是函数.

2.7.10 AffineRatio. 仿射比 λ

指令

仿射比 λ (\langle 点 A \rangle , \langle 点 B \rangle , \langle 点 C \rangle)

说明: 返回共线三点 ABC 的仿射比 λ , 其中 $\overrightarrow{AC} = \lambda \overrightarrow{AB}$, 其中 $\lambda \in \mathbb{R}$. 如果三点不共线, 则返回未定义.

2.7.11 RigidPolygon. 刚体多边形

指令

刚体多边形 (\langle 多边形 \rangle)

说明: 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.

指令

刚体多边形 (\langle 多边形 \rangle , \langle x 偏移量 \rangle , \langle y 偏移量 \rangle)

说明:

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

指令

刚体多边形 (\langle 自由点 1 \rangle , ..., \langle 自由点 n \rangle)

说明: 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.

 **注意:** 这个刚体多边形的形状和源多边形绑定, 源多边形的任何改动都会导致刚体多边形的形状发生变化. 如果要改变副本的形状, 只能更改其来源多边形.

2.7.12 Locus. 轨迹

指令

轨迹 (\langle 构造轨迹线的点 Q \rangle , \langle 点 P \rangle)

说明: 返回从属于点 P 的轨迹点 Q 的轨迹.

 **注意:** 点 P 必须是对象上 (线段上, 直线上, 圆锥曲线上等) 的点, 不可以为自由点和内点.

指令

轨迹 (\langle 构造轨迹线的点 \rangle , \langle 滑动条 \rangle)

说明:

返回从属于采用滑动条值的点 Q 的轨迹曲线.

指令

轨迹 (\langle 斜率场 \rangle , \langle 点 \rangle)

说明: 返回相当于过给定点的斜率场的轨迹曲线。

指令

轨迹 (<f(x,y)>,<点>)

说明: 用数值方法计算在给定点处微分方程 $\frac{dy}{dx} = f(x,y)$ 的解。

说明: 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,“Perimeter. 周长周长”指令和“First. 最前元素最前元素”指令。

 **注意:** 要避免使用 2 次及以上路径值和描点指令去产生轨迹点,否则将导致轨迹为未定义。

Tips:

1. 软件没有提供轨迹与其他对象交点的功能,只能用其他方法得到近似值。
2. 不可以直接对轨迹进行变换操作(例如旋转),如果要对轨迹进行变换,可以先对轨迹点进行变换,再创建这个点的轨迹。

2.7.13 LocusEquation. 轨迹方程**指令**

轨迹方程 (<轨迹>)

说明: 计算轨迹的方程并以隐式曲线表示。

指令

轨迹方程 (<轨迹点>,<动点>).

说明: 使用轨迹点和一个动点计算轨迹方程并以隐式曲线表示。

示例: 抛物线的方程:

```
F=(1,0)
l:x=-1
P=描点(l)
f=垂线(P,l)
g=中垂线(F,P)
Q=交点(f,g)
eq1:轨迹方程(Q,P) >>> eq1: 4x - y^2 = 0 并绘制出轨迹的图像.
```

指令

轨迹方程 (<布尔表达式>,<自由点>)

说明: 计算自由点满足布尔表达式时的轨迹。

示例: 椭圆的方程

```
F_1=(-2,0)
F_2=(2,0)
P=(1,2)
##P点为随意自由点
f=线段(P,F_1)
g=线段(P,F_2)
```

轨迹方程 ($f+g=8, P$)

$\ggg eq1 : 3x^2 + 4y^2 = 48$

注意:

1. 轨迹必须由自由点 (非滑动条) 产生
2. 仅适用于一组受限制的几何轨迹, 即使用点、线、圆、圆锥曲线 (线段和射线都被视作无限的)
3. 如果轨迹太复杂, 则返回未定义对象 (显示为?)
4. 如果没有轨迹, 则隐式曲线是空集 $0=1$
5. 如果轨迹是整个平面, 则隐式曲线就是方程 $0=0$
6. 如果计算是用 **Gröbner bases 方法** 完成的, 那么有时候会出现曲线的额外分支, 而这些分支原本不在原曲线中
7. 更多的信息和例子请参考 [官网](#), 此外还有 [隐式曲线轨迹的例子合集](#)

Tips: 该指令不太强, 很多轨迹方程是求不出来的, 请读者降低期望.

2.7.14 Arc. 弧线

指令

弧线 (< 圆 | 椭圆 >, < 点 1 >, < 点 2 >)

说明: 返回圆 (椭圆) 的定向弧线 (逆时针), 端点为给定的两个点. 注: 弧线的值即弧长.

指令

弧线 (< 圆 >, < 参数值 1 >, < 参数值 2 >)

说明: 返回圆 (椭圆) 的弧线, 端点由两参数值确定. 其中圆的参数方程为 $(r;t)$, 椭圆的参数方程为 $(a \cos t, b \sin t)$, 若两参数值为 t_1, t_2 , 则先确定弧线的两端点, 再逆时针绘制弧线.

示例:

弧线 ($x^2+y^2=1, \pi, 0$)

圆的半径为 1, 第一个端点为 $A(1; \pi)$, 即点 $(-1, 0)$, 第二个点为 $B(1; 0)$, 即点 $(1, 0)$, 从 A 到 B 按逆时针画弧, 如图 2.26. 从这个例子可以看出, 参数值 1 与参数值 2 之间没有大小关系. 此外, 弧线指令得到的弧线是圆或者椭圆的一部分, 它的值总是小于等于周长.

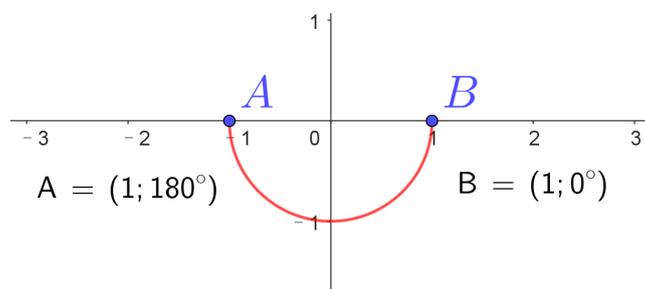


图 2.26

2.7.15 CrossRatio. 交比

指令

交比 (< 点 1 >, < 点 2 >, < 点 3 >, < 点 4 >)

说明: 计算四个共线点 A、B、C 和 D 的交比 λ , 其中: $\lambda = \frac{\text{仿射比}\lambda(B,C,D)}{\text{仿射比}\lambda(A,C,D)}$.
参见 [AffineRatio](#). 仿射比 λ

2.7.16 Intersect. 交点

指令

交点 (< 对象 1>, < 对象 2>)

说明: 得出两个对象的交点.

示例:

```
交点(-3x + 7y = -10, x^2+2y^2=8)          >>> A = (-1.02, -1, 87), B = (2.81, -0.22)
交点(y = x + 3, 曲线(t, 2t, t, 0, 10))    >>> A=(3, 6)
交点(曲线(2s, 5s, s, -10, 10), 曲线(t, 2t, t, -10, 10)) >>> A=(0, 0)
```

 **注意:** 如果用交点指令没有得出所有交点, 可以尝试在指令外面加上大括号, 即 `交点 (< 对象 1>, < 对象 2>)`.

指令

交点 (< 对象 1>, < 对象 2>, < 交点索引 n>)

说明: 得出两个对象的第 n 个交点

指令

交点 (< 对象 1>, < 对象 2>, < 起点 >)

说明: 从起点出发用数值迭代法计算两对象的交点

指令

交点 (< 函数 1>, < 函数 2>, < x-起始值 >, < x-终止值 >)

说明:

得出两个函数在指定数值区间上的全部交点.

指令

交点 (< 曲线 1>, < 曲线 2>, < 参数 1>, < 参数 2>)

说明: 从指定参数开始以迭代法寻找两条曲线的一个交点.

 **注意:** 交点指令在很多时候等同于 [IntersectPath](#). 相交路径.

2.7.17 Angle. 角度

指令

角度 (< 对象 >)

说明: 根据对象的类型不同, 返回的对象和值见下表.

示例:

```
角度(x^2/4+y^2/9=1)          >>> 90°
角度((1, 1))                  >>> 45°
角度(20)                       >>> 65.92°
```

对象类型	返回对象	是否画出角的图形
圆锥曲线	主轴与 x 轴的偏转角	是
向量	向量与 x 轴非负半轴的夹角	是
点 (复数)	极角 (辅角)	是
多边形	顶点逆 (顺) 时针, 范围所有内 (外) 角	是
数字	弧度转换成角度, 并对 360° 取余	否

由弧度制化角度值公式: $20rad = 20 \cdot \frac{180^\circ}{\pi} = 1145.92^\circ$, 再对 360° 取余, 即 $1145.92^\circ - 3 \cdot 360^\circ = 65.92^\circ$
 角度(多边形((4, 1), (2, 4), (1, 1))) >>>56.31°, 52.13°和 71.57°
 角度(多边形(4,1), (1,1), (2,4)) >>>303.69°, 288.43°, 307.87°

指令

角度(< 向量 1>,< 向量 2>)

说明: 返回两个向量之间的夹角 (在平面上按逆时针, 范围 0° 到 360°)

指令

角度(< 直线 1>,< 直线 2>)

说明: 返回两条直线方向向量之间的角度, 可以参考 **Direction. 方向向量**.

指令

角度(< 直线 >,< 平面 >)

说明: 返回线面角.

指令

角度(< 平面 >,< 平面 >)

说明: 返回 2 个平面的夹角 (法向量夹角的补角, 范围 0° - 180°).

指令

角度(< 点 >,< 顶点 >,< 点 >)

说明: 返回由三个点确定的角的角度值 (0° 到 360°), 其中第二个点为角的顶点.

指令

角度(< 点 >,< 顶点 >,< 角度 | 弧度 >).

说明: 返回指定点、按顶点旋转指定角度后的位置点以及角.

示例:

角度((0, 0), (3, 3), 30°) >>>返回 30° 角, 以及点((1.9, -1.1))
 ##即点(0,0)绕着点(3,3)逆时针旋转 30°

指令

角度(< 点 1>,< 点 2>,< 点 3>,< 方向 >)

说明: 返回由点和给定方向定义的角度, 方向可以是一条线或一个平面或向量 (结果为 $[0,360^\circ]$ 或 $[0,2\pi]$ 取决于默认角度单位).

示例:

角度((1, -1, 0), (0, 0, 0), (-1, -1, 0), z轴) >>>270°
 角度((-1, -1, 0), (0, 0, 0), (1, -1, 0), z轴) >>>90°

 **注意:** 如果不使用方向, 那么角度 (A,B,C) 和角度 (C,B,A) 在 3D 区显示的角度是一样的, 你可以使用角度范围控制角度在 0°-180° 或者 180° 到 360°. 如果使用方向, 角度 (A,B,C, 平面 (A,B,C)) 和角度 (C,B,A, 平面 (A,B,C)) 返回的默认角度是不一样的.

2.7.18 AngularBisector. 角平分线

指令

角平分线 (< 直线 1>, < 直线 2>).

说明: 返回由两条直线确定的全部 (两条) 角平分线.

指令

角平分线 (< 点 1>, < 顶点 2>, < 点 3>).

说明: 返回由三个点确定的角的一条角平分线. 第 2 个点为角的顶点

2.7.19 Distance. 距离

指令

距离 (< 点 >, < 对象 >)

说明: 得出一个点与一个对象之间的最短距离.

示例:

距离(((2, 1), $x^2 + (y - 1)^2 = 1$)) >>>1
 距离((2, 1, 2), (1, 3, 0)) >>>3
 距离(A, f) >>>f 是函数, A 为点, 则返回 A 与点 (x(A), f(x(A))) 的距离

 **注意:** 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.

指令

距离 (< 直线 1>, < 直线 2>)

说明: 得出两条直线之间的距离 (适用于平行线, 异面直线), 如果用于两平行线, 则距离为 0.

指令

距离 (< 平面 1>, < 平面 2>)

说明:

返回两平面之间的距离 (相交平面的距离为 0)

2.7.20 Type. 类型

指令

类型 (< 对象 >)

说明: 对于二次曲线或二次曲面返回一个数值. 下表表示二次曲线 (曲面) 的类型和值的对应.

值	二次曲线的类型	二次曲面的类型
1	单独的点	单独的点
2	相交线	
3	椭圆	椭球
4	圆	球面
5	双曲线	
6	空	空
7	双线	
8	平行线	
9	抛物线	抛物面
10	单线	单线
30		圆锥
31		圆柱
33		平面
34		平行平面
35		相交平面
36		单叶双曲面
37		双叶双曲面
38		抛物柱面
39		双曲柱面
40		双曲抛物面

2.7.21 Area. 面积

指令

面积 (< 圆 | 椭圆 | 多边形 >)

说明: 计算圆 (椭圆) 和多边形的面积

 **注意:** 如果是自相交多边形, 计算面积会有异常结果, 要计算函数与函数围成的面积, 可以参考 [IntegralBetween](#). 积分介于

2.7.22 Point. 描点

指令

描点 (< 对象 >)

说明: 返回对象上的点. 结果点能在路径上移动, 一般要结合 [PathParameter](#). 路径值使用.

指令

描点 (< 对象 >, < 参数 >)

说明: 按照路径值在对象上描点.

指令

描点 (< 点 >, < 向量 >)

说明: 创建一个按向量将指定点平移之后的点.

示例:

```
描点((1,1),向量((2,1)))      >>>(3,2)
## 等同于 (1,1)+(2,1)
```

指令

描点 (< 列表 >)

说明:

把只含有 2 个数字的列表转换成 2 维点, 3 个数字的列表转换成 3 维点, 1 个数字或者超过 3 个数字的列表将转换成 1 个自由点, 默认的坐标为原点. 如果你输入的是点列或者几何对象的列表, 则参见描点的第一种用法.

示例:

```
描点({1,2})      >>>(1,2)
描点({1,2,3})    >>>(1,2,3)
```

2.7.23 PointIn. 内点

指令

内点 (< 区域 >)

说明: 返回一个限定在指定区域内的点. 区域可以是多边形、二次曲线、不等式、平面等、

2.7.24 InteriorAngles. 内角

指令

内角 (< 多边形 >)

说明: 返回给定多边形的所有内角值.

2.7.25 Tangent. 切线

指令

切线 (< 点 >, < 圆锥曲线 >)

说明: 创建圆锥曲线过指定点的 (全部) 切线. 该指令用法也适用于部分隐式曲线, 例如 $y - x^3 = 0$.

指令

切线 (< 点 >, < 函数 >)

说明: 若给定点为 A, 则画出函数在 $(x(A), f(x(A)))$ 处的切线, 即 $y - f(x(A)) = f'(x(A))(x - x(A))$

 **注意:** 不管 A 在不在函数图像上, 都是画函数在 $(x(A), f(x(A)))$ 处的切线, 故切线只有 1 条, 这个切线不是过 A 点的切线.

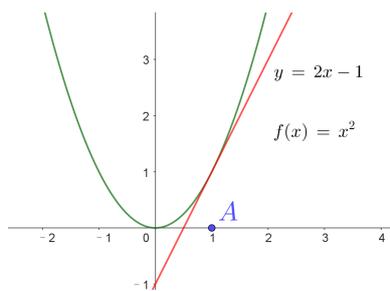


图 2.27

指令

切线 (< 曲线上的点 >, < 曲线 >)

说明:

创建过参数曲线 (样条曲线) 上的指定点并与曲线相切的切线.

指令

切线 (< 横坐标 x 值 >, < 函数 >)

说明: 求函数在 $x =$ 给定横坐标处的切线**指令**

切线 (< 直线 >, < 圆锥曲线 >)

说明: 创建圆锥曲线与指定直线平行的 (全部) 切线.**示例:**切线 ($y=4, x^2+y^2=4$) $\ggg y=2, y=-2$ **指令**

切线 (< 圆锥曲线 1 >, < 圆锥曲线 2 >)

说明:

创建 2 条圆锥曲线的所有公切线.

示例:切线 ($x^2 + y^2 = 4, (x - 6)^2 + y^2 = 4$) $\ggg y = -2, y = 2, 1.33x - 1.49y = 4, 1.33x + 1.49y = 4$ **Tips:** 对一般的函数来说, 要求过一点的切线是比较困难的, 也没有现成的工具或指令, 需要通过解方程等手段去求解切线, 但对于多项式函数等简单的函数, 则可以由以下案例中采用的方法去求切线。**示例:** 已知函数 $f(x) = x^3 - 3x^2$, 求 $f(x)$ 过点 $A = (1, 1)$ 的切线方程. $f=x^3-3x^2$ $A=(1, 1)$ $eq1:y=f=0$ **##**将 f 转换成隐式曲线 $eq1$ 切线 ($A, eq1$) $\ggg -1.06586x + 1.14471y = 0.07886$

2.7.26 ClosestPointRegion. 区域内最近点

指令

区域内最近点 (< 区域 >, < 点 >)

说明: 区域内构建一个距离给定点最近的点.

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

表 2.1

2.7.27 Cubic. 三次曲线

指令

三次曲线 (< 点 >, < 点 >, < 点 >, < 数字 >)

说明: 根据给定的序数值和三角形, 输出三次隐式曲线. 序数值和三次曲线的对应关系见表 2.1:

这个指令仍然在开发中, 支持的序数还在变化.

2.7.28 TriangleCurve. 三角曲线

笔者注: 这个指令应该是 GeoGeBra 中最难的指令之一, 笔者对其的理解也是皮毛.

指令

三角曲线 (< 点 1 >, < 点 2 >, < 点 3 >, < 重心坐标 (A,B,C) 满足的方程 >)

说明: 返回点 (x, y) 的重心坐标 (A, B, C) 满足第四个参数中的方程时所对应的隐式曲线方程. 在 GGB 中, A, B, C 的几何意义如图 2.28.

释义: **重心坐标的定义**: 已知三个点 P, Q, R , 若一个点的坐标 (x, y) 满足 $(x, y) = A \cdot P + B \cdot Q + C \cdot R$, 则把 (A, B, C) 称为该点的重心坐标. GeoGeBra 中对重心坐标进行了归一化, 即 $A + B + C = 1$. 比较常见的三角形的重心对应的重心坐标为 $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, 如果三角形的三条边为 a, b, c , 周长为 p , 则内心的重心坐标为 $(\frac{a}{p}, \frac{b}{p}, \frac{c}{p})$.

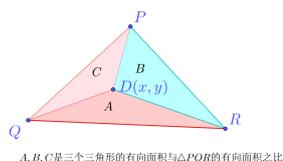


图 2.28

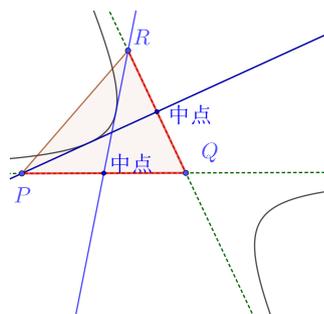


图 2.29

示例:

三角曲线(P,Q,R,B=C)

>>>QR边上的中线

解释: $B = C$ 这个方程表示 $S_{\triangle PDR} = S_{\triangle PDQ}$, 已知 D 落在 RQ 边上的中线上, 因此该指令返回 QR 边上的中线所在的直线.

三角曲线(P,Q,R,A*C=1/8)

>>>以PQ,QR为渐近线,且过P或R的切线把三角形面积平分的双曲线

见图2.29

说明: 若 $\overrightarrow{PD} = \lambda \overrightarrow{PQ} + \mu \overrightarrow{PR}$, 则 $B = \lambda, C = \mu$, 如我们把满足 $\lambda + \mu = k$ 的点 D 所在的直线称为等和线, 对应的指令为:

三角曲线(P,Q,R,B+C=k)

2.7.29 TriangleCenter. 三角形中心

指令

三角形中心(<点>,<点>,<点>,<数字>)

说明: 给出三角形 ABC 的第 n 个中心. $N < 3054$. 常见的三角形中心如下表 (旧版本的指令提示中有下表中的 9 个中心的提示, 新版本中被删除)

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

更多的三角形中心的定义和介绍可以参考网页 <https://faculty.evansville.edu/ck6/encyclopedia/ETC.html>

2.7.30 Trilinear. 三线坐标点

指令

三线坐标点(<点 A>,<点 B>,<点 C>,<数值 a>,<数值 b>,<数值 c>)

说明: 创建由给定数值确定的给定点三角形的三线坐标点.

释义: 三线坐标中的 a, b, c 表示点到三角形三条边的相对距离, 这 3 个数同时乘以 1 个数 μ , 三线坐标点保持不变.

示例:

三线坐标点(A,B,C,1,1,1)

>>>内心

三线坐标点(A,B,C,2,2,2)

>>>内心

常见的三线坐标点见下表 上表中外心可简记为 $\cos A$, 重心简记为 $\frac{1}{a}$, 垂心记为 $\sec A$, 内心记为 1.

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	$\cos B \cos C$	$\cos A \cos C$	$\cos A \cos B$

2.7.31 Sector. 扇形

指令

扇形 (< 圆锥曲线 >, < 点 1 >, < 点 2 >)

说明: 得出一个圆锥曲线 (圆或者椭圆) 上两点之间的扇形.

指令

扇形 (< 圆锥曲线 >, < 参数值 >, < 参数值 >)

说明: 得出一个圆锥曲线上、两个参数值之间的圆锥扇形.

 注意: 本指令只适用于圆和椭圆.

2.7.32 Ray. 射线

指令

射线 (< 起点 >, < 点 >)

说明: 创建一条起于一个点并穿过另一个点的射线.

指令

射线 (< 起点 >, < 方向向量 >)

说明:

创建一条起于给定起点并具有指定方向向量的射线.

2.7.33 ArePerpendicular. 是否垂直

指令

是否垂直 (< 直线 1 >, < 直线 2 >)

说明: 返回布尔值 true/false, 判断直线是否垂直, 真为 true, 假为 false, 不可用于空间直线.

2.7.34 AreConcurrent. 是否共点

指令

是否共点 (< 直线 1>,< 直线 2>,< 直线 3>)

说明: 返回布尔值 true/false, 判断是否三线共点, 不可用于空间直线.

注意: 平行线被认为交点为无限远, 因此如果是三条平行线, 会被判定为共点.

2.7.35 AreCollinear. 是否共线

指令

是否共线 (< 点 1>,< 点 2>,< 点 3>)

说明: 返回布尔值 true/false, 判断是否三点共线.

2.7.36 AreConcyclic. 是否共圆

指令

是否共圆 (< 点 1>,< 点 2>,< 点 3>,< 点 4>)

说明: 返回布尔值 true/false, 判断四点是否共圆

2.7.37 AreParallel. 是否平行

指令

是否平行 (< 直线 1>,< 直线 2>.

说明: 返回布尔值 true/false, 判断两直线是否平行.

注意: 平行包含重合的情况.

2.7.38 AreCongruent. 是否全等

指令

是否全等 (< 几何对象 1>,< 几何对象 2>).

说明: 返回布尔值 true/false, 判断对象是否全等.

2.7.39 AreEqual. 是否相等

指令

是否相等 (< 几何对象 1>,< 几何对象 2>)

说明: 返回布尔值 true/false, 判断对象是否相等.

注意:

```
是否相等(线段((1,2),(3,4)),线段((3,4),(1,6))) >>>false
      线段((1,2),(3,4))  线段((3,4),(1,6)) >>>>true
```

后者只考虑值得大小, 即线段的长度.

2.7.40 IsTangent. 是否相切

指令

是否相切 (< 直线 >, < 圆锥曲线 >)

说明: 返回布尔值 true|false, 判断给定直线是否与圆锥曲线相切.

2.7.41 CircumcircleArc. 外接圆弧

指令

外接圆弧 (< 点 >, < 点 >, < 点 >)

说明: 返回过 3 点的圆弧.

2.7.42 CircumcircleSector. 外接圆扇形

指令

外接圆扇形 (< 点 >, < 点 >, < 点 >)

说明: 返回过三点的圆扇形.

2.7.43 Segment. 线段

指令

线段 (< 点 1 >, < 点 2 >)

说明: 创建一条两点之间的线段.

指令

线段 (< 点 >, < 长度 >)

说明: 创建一条以指定点为起点, 长为指定长度的线段. 线段的终点可以绕着起点转动.

 注意: 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.

2.7.44 IntersectPath. 相交路径

指令

相交路径 (< 直线 | 多边形 >, < 多边形 >)

说明: 创建两对象的公共部分

指令

相交路径 (< 平面 >, < 多边形 >)

说明: 创建给定的平面和多边形相交形成的相交路径.

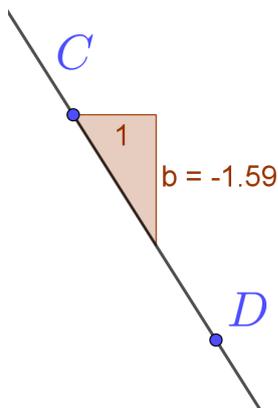


图 2.30

指令

相交路径 (< 平面 >, < 二次曲面 >)

说明:

创建的平面和二次曲面的相交路径, 此时等同于相交曲线

2.7.45 Slope. 斜率**指令**

斜率 (< 直线 | 射线 | 线段 >)

说明: 返回对象的斜率, 并绘制出一个以第一个点为起点的直角三角形, 水平的直角边长为 1, 竖直的边长为斜率 (有向线段), 如图 2.30

2.7.46 Centroid. 形心**指令**

形心 (< 多边形 >)

说明: 返回多边形的形心.

释义: 形心: 1 个不自交的多边形 $A_1 \dots A_n$, 记 $A_{n+1} = A_1$, 有向面积为 A , 则形心的公式为:

$$\frac{1}{6A} \sum_{i=1}^n (A_i + A_{i+1})(A_i \times A_{i+1})$$

2.7.47 CircleArc. 圆弧**指令**

圆弧 (< 圆心 >, < 点 1 >, < 点 2 >)

说明: 以圆心为中心点在另外两点之间逆时针创建一条圆弧线. 第二个点可以不在圆弧上. 英文也写作: **CircleArc**.

2.7.48 CircleSector. 圆扇形

指令

圆扇形 (< 圆心 >, < 点 1 >, < 点 2 >).

说明: 以圆心为中心点在另外两点之间逆时针创建一个扇形. 第二个点可以不在圆弧上.

2.7.49 PolyLine. 折线

指令

折线 (< 点列 >) 折线 (< 点 1 >, < 点 2 >...< 点 n >)

说明:

创建一个开放的多边形链 (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)

注意:

1. 折线的长度会在代数区显示
2. 绘图区有折线的工具, 3d 绘图区需要在工具栏定制中加入折线工具后, 才可以在 3D 区使用折线工具
3. 可以创建不连续折线, 即点列中有未定义的点.

示例:

```
折线((1,3),(4,3),(?,?),(6,2),(4,-2),(2,-2))    >>>f=9.47
## 同时在绘图区绘制出图形
```

Tips: 用折线指令类似于多边形的封闭折线, 只需首尾是相同字母. 例如: 折线 (A,B,C,A).

2.7.50 Prove. 证明

指令

证明 (< 布尔表达式 >)

说明: 返回给定布尔表达式总体上是 true 或 false. 通常, GeoGebra 通过使用数字来确定布尔表达式是否为真计算. 但是, Prove 命令使用符号方法来确定一个陈述总体上是真是假. 如果 GeoGebra 无法确定答案, 则结果是未定义.

示例:

```
A=(1,2)
B=(3,4)
C=(5,6)
Prove(AreCollinear(A,B,C))    >>>true
```

2.7.51 ProveDetails. 证明过程

指令

证明过程 (< 布尔表达式 >)

说明: 返回自动证明结果的一些细节.

2.7.52 Line. 直线

指令

直线 (< 点 1>, < 点 2>).

说明: 创建一条过点 A 和 B 的直线.

指令

直线 (< 点 >, < 平行线 | 方向 >).

说明:

过点创建与线或者方向平行的直线. 这里的线包含直线线段和射线.

2.7.53 LineBisector. 中垂线

指令

中垂线 (< 线段 >)

说明: 得出一条线段的中垂线.

指令

中垂线 (< 点 1>, < 点 2>)

说明: 得出两点间的线段的中垂线.

3D 区指令

中垂线 (< 点 1>, < 点 2>, < 方向 >).

说明: 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, 也可以是平面.

2.7.54 Midpoint. 中点

指令

中点 (< 线段 >)

说明: 返回线段的中点.

指令

中点 (< 圆锥曲线 | 二次曲面 >)

说明:

返回圆锥曲线 (二次曲面) 的中心, 此时等同中心.

指令

中点 (区间)

说明: 返回区间的中点 (数值对象)

指令

中点 (< 点 1>, < 点 2>)

说明:

返回 A,B 两点的中心点, 即两点连线的中点, 可以用 $\frac{A+B}{2}$ 计算

2.7.55 Barycenter. 重心

指令

重心 (< 点列 >, < 权重列表 >)

说明: 计算点列的重心 (即对点的横纵坐标取加权平均数), 设点列为 l1 中的元素为 A_1, A_2, \dots, A_n , 权重列表为 l2, l2 中的元素为 p_1, p_2, \dots, p_n 具体的计算公式如下:

$$\frac{A_1 p_1 + A_2 p_2 + \dots + A_n p_n}{p_1 + p_2 + \dots + p_n}$$

用 ggb 的指令来写即:

```
sum(l1*l2)/sum(l2)
```

示例:

```
重心({(2, 0), (0, 2), (-2, 0), (0, -2)}, {2, 1, 1, 1}) >>>(0.4, 0)
```

```
#计算方法:  $\frac{(2,0)*2+(0,2)*1+(-2,0)*1+(0,-2)*1}{2+1+1+1} = (0.4, 0)$ 
```

2.7.56 Perimeter. 周长

指令

周长 (< 多边形 | 圆锥曲线 | 轨迹 >)

说明: 返回多边形, 椭圆, 圆和轨迹的周长. 轨迹的周长往往计算不正确, 不建议大家使用.

2.7.57 Circumference. 周界长

指令

周界长 (< 圆锥曲线 >)

说明: 如果是圆或者椭圆, 返回周长, 其他的圆锥曲线返回未定义对象.

2.7.58 ClosestPoint. 最近点

指令

最近点 (< 路径 >, < 点 >)

说明: 返回一条路径上最接近所选点的一个新点. 一般来说 2 个参数可以互换.

指令

最近点 (< 直线 1>, < 直线 2>)

说明: 返回第一条直线上最接近于第二条直线的点 (可以用于 3D 区).

2.8 Transformation. 几何变换

2.8.1 Mirror. 对称

指令

对称 (< 几何对象 >, < 对称中心点 >)

说明: 将几何对象以指定点为对称点进行镜像反射。

指令

对称 (< 几何对象 >, < 对称轴直线 | 射线 | 线段 >)

说明: 以指定直线为对称轴，将对象（例如图片）进行镜像反射。

3D 区指令

对称 (< 几何对象 >, < 对称平面 >)

说明: 关于平面反射几何对象

指令

对称 (< 几何对象 >, < 反演基圆 >)

说明: 几何对象关于指定圆的反演图像。



注意:

1. 反演、轴对称、中心对称、平面对称这些工具得到的指令都是对称。
2. 线段、多边形等对象对称后会产生对应的顶点。

2.8.2 Translate. 平移

指令

移 (< 几何对象 >, < 向量 >)

说明: 按向量移动几何对象 (使用指令时向量和点没有严格区分), 线段、多边形等对象对称后会产生对应的顶点。

指令

平移 (< 向量 >, < 起点 >)

说明:

将向量移动到新的起点。

2.8.3 Shear. 切变

指令

切变 (< 几何对象 >, < 直线 | 射线 | 线段 >, < 比 >)

说明: 释义: 切变是一种变换, 其中沿给定线的所有点保持固定, 而其他点平行移至与它们到线的距离为 d 倍 (给定的比) 的位置, 点在线的两侧时, 平移的方向相反. 切变平面图形不会改变其面积. 具体的计算方法如下:

已知直线 AB (线段, 射线同理), 则任意一点 C , 以 d 为比率作切变后的坐标:

$$C' = C + d \frac{(C - A) \otimes (C - B)}{(B - A)^2} (B - A)$$

示例: 如图 2.31, $\triangle CDE$ 和 $\triangle CDF$ 沿着线段 AB , 以比例 $d = \frac{1}{2}$ 做切变. 由于 C, D 在线段 AB 上, 因此切变后的位置不变, 而 E, F 在线段两侧, 沿着线段 AB 方向平移的方向相反, 平移的距离由点到线段的距离乘以比率 $d = \frac{1}{2}$, 因此可见: $EE' = \frac{1}{2}EF, GG' = \frac{1}{2}GH$, 切变后得到 2 个新的和原三角形面积相等的三角形 (含新产生的顶点)

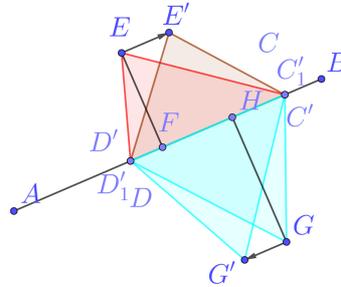


图 2.31

2.8.4 Stretch. 伸缩

指令

伸缩 (< 几何对象 >, < 向量 >)

说明: 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)。

指令

伸缩 (< 几何对象 >, < 直线 | 射线 | 线段 >, < 比 >)

说明:

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)。

Tips: 伸缩可以用于 2d 区模拟 3D 区的绕轴旋转的效果, 例如: 伸缩 (f, y 轴, a), f 为一个对象, a 是 -1 到 1 的滑动条。

2.8.5 Dilate. 位似

指令

位似 (< 几何对象 >, < 位似比例 >, < 位似中心点 >)

说明: 以指定的位似中心将对象按给定的位似比例进行缩放. 如果不指定位似中心, 则默认为原点。

注意: 位似比可以为负数, 若位似中心为 O , 对象上的任意一点 A , 位似比为 μ , 则 A 点对应的象 A' 满足 $\overrightarrow{OA'} = \mu \overrightarrow{OA}$

2.8.6 Rotate. 旋转

指令

旋转 (< 几何对象 >, < 角度 | 弧度 >, < 旋转中心 >)

说明: 将几何对象围绕给定点旋转指定角度, 不指定旋转中心则默认为原点.

3D 区指令

旋转 (< 几何对象 >, < 角度 | 弧度 >, < 旋转轴 >)

说明: 将几何对象绕给定的旋转轴旋转指定的角度.

指令

旋转 (< 对象 >, < 度 | 弧度 >, < 轴上的点 >, < 轴方向或平面 >)

说明: 按给定点定义的轴和轴方向或平面围绕轴的角度旋转几何对象.



注意:

1. 向量不能围绕坐标原点旋转, 但可以绕起点旋转;
2. 当多边形、线段、弧线等旋转时, 也会显示顶点/端点和边的图像 (如果是多边形) 被创建.
3. 可以用于旋转图片
4. 旋转文本需要使用 `RotateText`. 旋转文本

2.9 script. 脚本

本章所有的指令与其他的指令区别在于: 脚本类指令在输入栏输入后, 通常不会在代数区留下定义, 通常用于赋值以及设置对象的属性或执行一个动作等, 脚本类指令通常也不能用于序列或映射中用于批量操作.

2.9.1 Button. 按钮

指令

按钮 ()

说明: 创建一个按钮

指令

按钮 ("< 标题 >")

说明:

创建给定标题的按钮. 标题必须为文本格式.

示例:

```
按钮("开始")           >>> 创建1个标题为"开始"的按钮,按钮上文字显示为开始,如果不设定标题,则显示为按钮的名称.
% 注如非特殊规定,按钮的名字一般为button1,button2...
```

2.9.2 PlaySound. 播放声音

指令

播放声音 (< 网址 >)

说明: 播放给定网址上(网址放在""里面)的 mp3 文件.

示例:

播放声音("https://test.geogebra.org/~mike/welcome-to-geogebra-markus.mp3")
播放声音("#J2sQQfwQ") >>>播放上传到GeoGeBra 网站上的声音.

指令

播放声音 (< 是否播放? true|false >)

说明: 暂停或重新播放. 播放声音 (true)= 播放, 播放 (false)= 暂停.

指令

播放声音 (< 函数 >, < 最小值 >, < 最大值 >)

说明: 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.

示例:

播放声音($\sin(440 \cdot 2\pi \cdot x)$, 0, 1) >>>播放一个纯正弦波音调, 频率为440赫兹(音乐中的A调), 持续一秒钟.

指令

播放声音 (< 函数 >, < 最小值 >, < 最大值 >, < 采样率 >, < 样本深度 >)

说明: 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.cai'yang'lv 采样率可以为 8000、11025、16000、22050 或 44100, 样本深度可以为 8 和 16.

指令

播放声音 (< 音符 >, < 持续时间 >, < 乐器 >)

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				

表 2.2

编号	乐器	编号	乐器	编号	乐器	编号	乐器	编号	乐器
1	原声三角钢琴	27	爵士电吉他	53	合唱团 Aahs	79	哨子	105	锡塔尔琴
2	明亮的原声钢琴	28	电吉他(干净)	54	声音哎呀	80	陶笛	106	班卓琴
3	电三角钢琴	29	电吉他(静音)	55	合成器语音	81	lead1(方形)	107	三味线
4	Honky-tonk 钢琴	30	过载吉他	56	管弦乐队热门	82	lead2(锯齿)	108	江东
5	电钢琴 1	31	失真吉他	57	小号	83	Lead3(calliope)	109	卡林巴
6	电钢琴 2	32	吉他泛音	58	长号	84	lead4(chiff)	110	袋管
7	大键琴	33	原声贝斯	59	大号	85	Lead5(charang)	111	小提琴
8	克拉维	34	电贝司(手指)	60	静音小号	86	Lead6(声乐)	112	沙奈
9	钢片琴	35	电贝司(拨片)	61	法国号	87	lead(7 五度)	113	叮叮当
10	钟琴	36	无品贝斯	62	铜管部分	88	Lead8(低主音)	114	阿戈戈
11	八音盒	37	巴掌贝斯 1	63	合成铜管 1	89	Pad1(新时代)	115	钢桶
12	颤音琴	38	巴掌贝斯 2	64	合成铜管 2	90	Pad2(暖)	116	木刻
13	马林巴琴	39	合成贝斯 1	65	高音萨克斯	91	Pad3 复音合成器	117	太鼓鼓
14	木琴	40	合成贝斯 2	66	中音萨克斯	92	Pad4(合唱团)	118	旋律唢呐
15	管状铃铛	41	小提琴	67	次中音萨克斯	93	pad5(弓形)	119	合成器鼓
16	扬琴	42	中提琴	68	男中音萨克斯	94	Pad6(金属)	120	反向钹
17	拉杆风琴	43	大提琴	69	双簧管	95	Pad7(光环)	121	吉他品格噪声
18	敲击风琴	44	低音提琴	70	英国管	96	Pad8(扫描)	122	呼吸噪声
19	摇滚风琴	45	颤音琴弦	71	巴松管	97	FX1(雨)	123	海滨
20	教堂管风琴	46	Pizzicato 琴弦	72	单簧管	98	FX2(原声带)	124	鸟啾
21	簧片风琴	47	管弦竖琴	73	短笛	99	FX3(水晶)	125	电话铃声
22	手风琴	48	定音鼓	74	长笛	100	FX4(大气)	126	直升机
23	口琴	49	弦乐合奏 1	75	录音机	101	FX5(亮度)	127	掌声
24	Tango 手风琴	50	弦乐合奏 2	76	排笛	102	FX6(哥布林)	128	枪声
25	原声吉他(尼龙)	51	合成弦 1	77	吹瓶	103	FX7(回声)		
26	原声吉他(钢)	52	合成弦 2	78	尺八	104	FX8(科幻)		

表 2.3

说明:

播放一个音符. 音符是 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. 详见表 2.2 和表 2.3

2.9.3 ExportImage. 导出图片**指令**

导出图片(< 属性 >, < 数值 >, < 属性 >, < 数值 >, ...)

说明: 导出当前活动视图(或由“view”参数指定的视图)的图像.

属性必须使用英文文本, 常用的有:

1. view: 视图. 其中与“view”有关的值 1 表示绘图区, 2 表示绘图区 2, -1 表示 3d 绘图区.
2. scale: 比例, scalecm: 每单位几厘米, dpi: 每英寸点数
3. filename: 文件名
4. type: 文件类型, 可选 pdf, svg, png 等
5. width: 宽度, height: 高度
6. loop: 循环, time: 时间
7. rotate: 旋转
8. corner: 角点

示例:

导出图片("scale", 5)

>>> 显示当前视图的弹出窗口, 以便用户可以右键单击

导出图片("filename", "image.png")

```

>>>保存当前视图的名为 “image.png” 的文件
导出图片("filename", "image.png", "view", 2)
>>>保存图形视图 2 中名为 “image.png” 的文件
导出图片("filename", "image.png", "view", -1)
>>>保存名为 “image.png” 的 3D绘图区的文件
导出图片("filename", "image.png", "dpi", 300)
>>>以 300 dpi (每英寸点数) 存储当前视图的名为 “image.png” 的文件
导出图片("filename", "image.png", "scale", 2)
>>>以比例 2 (即两倍标称屏幕分辨率) 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 600 dpi 的比例以 1 单位 = 2 厘米的比例保存当前视图的名为 “image.png” 的文件
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)
>>>保存名为 “image.png” 的透明 PNG 文件
将 “transparent” 属性设置为 false 也会保存背景图像.
导出图片("filename", "image.svg", "type", "svg")
>>>将当前视图的名为 “image.svg” 的文件保存为 SVG 格式
导出图片("filename", "image.gif", "type", "gif", "slider", a, "loop", true, "time", 200, "width", 400)
>>>保存由滑块 “a” 控制的当前视图的循环动画 GIF, 帧之间为 200 毫秒.保持 width 和 小滑块步数
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” 控制的当前视图的循环动画 GIF, 帧之间间隔 200 毫秒, 并旋转视图 动画期间为 360°.保持视图
的大小和步长数较小
导出图片("type", "pdf", "filename", "test.pdf")
>>>创建当前视图的 PDF (如果图形视图 2 已打开, 则创建 2 页的 PDF)
导出图片("type", "pdf", "filename", "test.pdf", "slider", n)
>>>创建当前视图的多页 PDF, 其中每个页面对应于滑块 “n” 的一个步骤
pic1 = 导出图片("view", 2, "corner", A, "corner2", B)
>>>创建 绘图区2 的 GeoGebra 图像, 并将其放置在视图中, 其位置由 和 定义AB

```

 **注意:** 在 GeoGebra Classic 5 中, 如果没有参数, 图像将被复制到剪贴板. 在 Chrome 中 browser 中, 您可以指定 filename="clipboard", true 要导出部分区域的图像, 可以用右键框选, 再在菜单中导出图片或者截图. GeoGebra 5 Classic 不支持某些语法. 在 Chrome 中, 您可以尝试实验性语法以获得比使用更快且更小的导出. 动画 GIF "type", "webm" 对于 2D 图形视图, 您可以尝试此语法来获得单色导出 (仅限 PNG 格式) "grayscale", true

对于 3D 图形视图, 仅支持位图 (例如 png、gif). 在 GeoGebra Classic 5 中, 您可以尝试此操作以获得高分辨率输出. 将 <username> 更改为您的 Windows 用户名. 导出图片 ("filename", "c:\Users \<username> \AppData \image.png", "view", -1, "width", 2000)

2.9.4 ZoomIn. 放大

指令

放大 ()

说明: 重置绘图区到标准视图

指令

放大 (< 放大倍数 n>)

说明: 放大当前视图 n 倍

小技巧: 放大 (1) 可以清除踪迹

指令

放大 (< 缩放因子 >, < 视图中心点 >)

说明: 第一个参数指定放大倍数, 第二个参数指定缩放的中心点.

指令

放大 (< x 最小值 >, < y 最小值 >, < x 最大值 >, < y 最大值 >)

说明: 放大视图到由 x 最小值, y 最小值, x 最大值, y 最大值确定的矩形区域**注意:** 如果这些参数任何一个是从属的数或者任何已定义的量, 那么视图的边界将由这个变量控制, 而不能通过鼠标滚轮或者拖动进行控制了. 为避免这种情况行为中, 请使用 CopyFreeObject(复制自由对象) 命令.

3D 区指令

放大 (< x 最小值 >, < y 最小值 >, < z 最小值 >, < x 最大值 >, < y 最大值 >, < z 最大值 >)

说明:放大视图到由 x 最小值, y 最小值, z 最小值, x 最大值, y 最大值, z 最大值确定的长方体区域

2.9.5 SetValue. 赋值

指令

赋值 (< 布尔值 >, < 0=false|1=true >)

说明: 给一个布尔值对象/复选框赋值: 1=true, 0=false.

指令

赋值 (< 对象 1>, < 对象 2>)

说明: 把对象 2 的值赋值给对象 1. 需要注意以下几点:

1. 是把对象 2 的值赋予对象 1, 值: 包括数值, 点值, 方程值, 函数式, 列表等
2. 两对象的值的类型要一致, 点值赋值给点, 数值赋值给数, 方程赋值给方程. 赋值不会改变对象 1 的类型 (复数, 点, 向量被认为都是点型).
3. 对象 1 需要为自由对象或半自由对象, 不可以为附属对象, 否则赋值无效, 比如多边形, 线段无法进行赋值.
4. 路径上的点进行赋值时, 不能脱离路径, 对象 2 不在路径上时,
5. 要区别赋值和等号的区别, 赋值后, 对象 2 的改变不会影响对象 1.

指令

赋值 (< 列表 >, < 数字 n>, < 对象 1>)

说明: 将自由列表的第 n 个元素赋值为对象 1, n 最大值为列表的长度 +1**示例:**

```
l1={1,2,3}
l2={1,2,3}
```

```
赋值(l1,2,4)          >>>l1={1,4,3}
赋值l2,4,-1)         >>>l2={1,2,3,-1}
##l1,l2必须为已存在的列表,赋值后名称不变,列表中的数值改变,赋值({1,2,3},4,-1)这个指令不会产生新的列表.
```

指令

SetValue (< 下拉列表 >, < 数字 n >)

说明: 下拉列表中将所选元素的索引设置为 n

示例: 已知下拉列表 l1=1,2,3,5

```
赋值(l1,4)  >>>所选元素变为5
```

指令

赋值(对象,?)

说明: 把 1 个自由对象赋值为未定义对象, 显示为?

2.9.6 Checkbox. 复选框

指令

复选框()

说明: 创建一个复选框

指令

复选框(< 标题 >)

说明:

创建 1 个给定标题的复选框(注: 是标题, 而不是对象名, 标题必须为文本)

指令

复选框(< 列表 >)

说明: 创建一个与列表中对象关联的复选框, 当复选框没被选中时, 隐藏列表中的对象.

指令

复选框(标题,< 列表 >)

说明: 创建一个与列表中对象关联的给定标题的复选框, 当复选框没被选中时, 隐藏列表中的对象.



注意: 复选框关联对象的本质即把对象的显示条件设置为复选框对应的布尔值, 复选框勾选时, 布尔值为 true, 不勾选时为 false. 用复选框工具可以关联多个对象, 但是确定后无法再用工具增加关联对象, 但是可以用设置显示条件的方式增加.

示例:

```
复选框({A,B,C})  >>>创建1个关联对象A,B,C的复选框
```

2.9.7 CopyFreeObject. 复制自由对象

指令

复制自由对象 (< 对象 >)

说明: 创建 1 个自由对象的副本, 该副本是个独立的对象, 不会随着原来对象的变化而变化.

注意: 不能复制多边形等对象.

2.9.8 AttachCopyToView. 附加副本

指令

附加副本 (< 对象 >, < 视图 0|1|2 >)

说明: 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.

指令

附加副本 (< 对象 >, < 视图值 0-创建副本 |1-创建从属副本 |2-创建从属副本 >, < 点 1 >, < 点 2 >, < 屏幕点 1 >, < 屏幕点 2 >)

说明:

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.

2.9.9 UpdateConstruction. 更新作图

指令

更新作图 ()

说明: 重算全部的对象 (随机数重新生成) 同于“F9”键或“Ctrl+R”

Tips: 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, 可能还需要首先使用激活视图 (1) 或激活视图 (2).

2.9.10 Turtle. 海龟

指令

海龟 ()

说明: 构造一只海龟卡通图. 初始的海龟在 origin 位置 (一般以 turtle* 为名字), 且在绘图区的左下角出现一个操作海龟的按钮.

2.9.11 TurtleBack. 后退

指令

后退 (< 海龟 >, < 路程 >)

说明: 海龟按照路程后退.

```
后退(turtle1,3) >>>绘图区左下角出现一个播放按钮,单击播放按钮,海龟的动画才会启动,且同时画出运动的轨迹
```

 **注意:** 类似的指令还有:TurtleForward. 前进,TurtleDown. 落笔,TurtleLeft. 左转,TurtleRight. 右转,TurtleUp. 抬笔

2.9.12 Slider. 滑动条

指令

```
滑动条(< 最小值>, < 最大值>, < 增量>, < 速度>, < 宽度(px)>, < 角度? true|false>, < 水平? true|false>, < 启动动画? true|false>, < 随机? true|false>)
```

说明: 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 2 个参数是必须的, 最大值必须比最小值大, 否则会出现滑动条无法滑动的情况, 再关闭 ggb 文件再重新打开时可能会出错.
- 增量: 默认为 0.1
- 速度: 默认为 1, 速度为 1 表示播放 1 次动画需要 10s 左右, 如果速度设为负值, 则动画时, 会和“重复”参数规定的方向相反, 比如重复设定为递增 1 次, 速度设为-1, 则滑动条的滑动方式为递减 1 次.
- 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.
- 启动动画: 设置滑动条是否自动播放动画—默认为: false.
- 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.

 **注意:** 这些参数只有前 2 个是必须的, 后面的参数可以不输入.

2.9.13 ParseToFunction. 解析为函数

指令

```
解析为函数(< 函数>,< 字符串>)
```

说明: 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

示例:

```
f(x)=2x
解析为函数(f,"ln(x)") >>>f(x)=ln(x)
```

指令

```
解析为函数(< 字符串>)
```

说明: 创建 1 个由字符串定义的函数, 这种用法指令帮助中没有提及

```
h(x)=解析为函数("x+1/x") >>>f(x)=x+1/x
```

指令

```
解析为函数("二元函数字符串", 变量字符串的列表)
```

说明: 将字符串解析为二元函数

示例:

```
解析为函数("2u+3v", {"u", "v"})          >>>创建1个以u,v为变量的二元函数a(u,v)=2u+3v
```

2.9.14 ParseToNumber. 解析为数

指令

解析为数 (<数值>, <字符串>)

说明: 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.

注意: 这是一个脚本指令, 只作用 1 次, 不会随着字符串的改变动态更新. 要想将字符串中的数字变成动态更新的数字, 可以使用: [转换为十进制 \(text1,10\)](#)

指令

解析为数 (<字符串>)

说明: 解析文本并将结果存储为数字. 指令提示中没有这种用法.

示例:

```
a=解析为数("1")          >>>a=1
```

2.9.15 StartRecord. 开始记录

指令

开始记录 ()

说明: 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.

指令

开始记录 (<true|false>)

说明: 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)

2.9.16 TurtleDown. 落笔

指令

落笔 (<海龟>)

说明: 命令给定的海龟可留下踪迹.

2.9.17 StartAnimation. 启动动画

指令

启动动画 ()

说明: 将所有暂停的动画启动, 注意该指令仅对启动过动画的动点, 滑动条起作用.

指令

启动动画 (< 布尔值 >)

说明:

如果布尔值是 false, 暂停所有动画, 否则重启动画.

指令

启动动画 (< 滑动条 | 点 >, < 滑动条 | 点 >, ...)

说明:

启动给定的点和滑动条动画, 点必须是在路径上.

指令

启动动画 (< 滑动条 | 点 >, < 滑动条 | 点 >, ..., < true | false >)

说明: 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.

注意: 如果想做 1 个控制动画暂停和启动的按钮, 先新建 1 个布尔值 a, 动画滑动条为 b, 新建 1 个按钮, 在单击脚本中写:

```
a=!a
启动动画(b,a)
```

2.9.18 TurtleForward. 前进

指令

前进 (< 海龟 >, < 路程 >)

说明: 海龟按照路程前进, 路程可以为负值.

2.9.19 Delete. 删除

指令

删除 (< 对象 >)

说明: 删除对象和他的附属对象. 如果对象不存在, 删除指令也不会报错.

2.9.20 SetBackgroundColor. 设置背景颜色

指令

设置背景颜色 (< 对象 >, < 红色 0-1 >, < 绿色 0-1 >, < 蓝色 0-1 >)

说明: 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 $2|\frac{t}{2} - \text{round}(\frac{t}{2})|$ 映射.

指令

设置背景颜色 (< 对象 >, "< 颜色 >")

说明: 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. 如果在 ExeCute 指令中使用, 必须使用英文颜色名. 部分颜色的 16 进制和 RGB 值及颜色的英文名如下表, 在指令栏或者按钮的脚本中使用, 颜色名可以使用中文.

颜色	英文	16 进制	R	G	B	颜色	英文	16 进制	R	G	B
水蓝	aqua	BCD4E6	0.74	0.83	0.9	青色	cyan	00FFFF	0.00	1.00	1.00
黑色	black	000000	0	0	0	深蓝	darkblue	1C39BB	0.11	0.22	0.73
蓝色	blue	0000FF	0	0	1	深灰	darkgray	7A7777	0.48	0.47	0.47
灰色	gray	808080	0.5	0.5	0.5	深绿	darkgreen	254117	0.15	0.25	0.09
绿色	green	008000	0	0.5	0	金黄	gold	D4A017	0.83	0.63	0.09
黄绿	lime	00FF00	0	1	0	靛蓝	indigo	307D7E	0.19	0.49	0.49
褐红	maroon	800000	0.5	0	0	浅蓝	lightblue	ADDFFF	0.68	0.87	1.00
紫色	purple	800080	0.5	0	0.5	浅绿	lightgreen	CCFFCC	0.80	1.00	0.80
红色	red	FF0000	1	0	0	浅灰	lightgray	A8A8A8	0.66	0.66	0.66
银灰	silver	C0C0C0	0.75	0.75	0.75	浅黄	lightyellow	FFFEDC	1.00	1.00	0.86
青色	teal	008080	0	0.5	0.5	洋红	magenta	FF00FF	1.00	0.00	1.00
白色	white	FFFFFF	1	1	1	橙色	orange	F87A17	0.97	0.48	0.09
黄色	yellow	FFFF00	1	1	0	粉红	pink	FAAFBE	0.98	0.69	0.75
棕色	brown	980516	0.6	0.02	0.09	蓝绿	turquoise	43C6DB	0.26	0.78	0.86
深红	crimson	E41B17	0.89	0.11	0.09	紫蓝	violet	8D38C9	0.55	0.22	0.79

表 2.4

 **注意:** 如果不指定对象, 则设置的是当前活动视图的背景颜色.

2.9.21 SetDecoration. 设置标记

指令

设置标记 (< 线段 | 角 >, < 标记数字代码 0—6 | 7 >)

说明: 设置线段或角的标记. 线段的数字代码和标记的样式从 0-6, 如下图

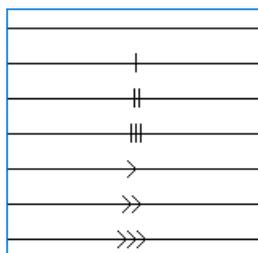


图 2.32

角的标记样式从 0-6 如下图:

填充样式从 0-7 如下表:

2.9.22 SetLabelMode. 设置标签模式

指令

设置标签模式 (< 对象 >, < 数字 >)

说明:

数字与模式的对应如下表

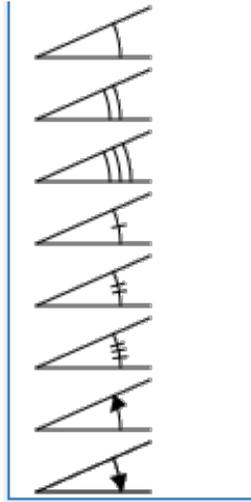


图 2.33

数字	样式
0	标准
1	斜线
2	网格
3	棋盘
4	圆点
5	蜂巢
6	砖形
7	交织

表 2.5

2.9.23 SetCaption. 设置标题

指令

设置标题 (< 对象 >, < 文本 >).

说明: 更改指定对象的标题，文本必须用一对英文的双引号""括住。

2.9.24 SetPointSize. 设置点径

指令

设置点径 (< 对象 >, < 数字 >)

说明: 更改点或者对象顶点的大小，对象是任何有顶点的对象，比如多边形、多面体等 (不包含线段, 折线等)

数值	模式
0	名称
1	名称与数值
2	数值
3	标题
4	标题与数值

表 2.6

2.9.25 SetPointSize. 设置点型

指令

设置点型 (< 点 >, < 数字 >)

说明: 更改点型为指定样式, 数字和点型的对照见图2.34.

数字	样式	符号
0	有轮廓的实心点	●
1	交叉型	×
2	空心点	○
3	十字形	+
4	实心菱形	◆
5	空心菱形	◇
6	朝上的三角形	▲
7	朝下的三角形	▼
8	朝右的三角形	►
9	朝左的三角形	◄
10	无轮廓的实心点	•

图 2.34

2.9.26 SetDynamicColor. 设置动态颜色

指令

设置动态颜色 (< 对象 >, < 红色值 0-1 >, < 绿色值 0-1 >, < 蓝色值 0-1 >)

说明: 设置对象的动态颜色. 三种颜色的范围为 0-1, 对应 RGB 颜色的 0-255

指令

设置动态颜色 (< 对象 >, < 红色值 0-1 >, < 绿色值 0-1 >, < 蓝色值 0-1 >, < 虚实 0-1 >)

说明:

设置对象的动态颜色和虚实度. 注意: 多边形, 圆锥曲线等有虚实度, 点, 线等没有虚实度.

2.9.27 SetPerspective. 设置格局

指令

设置格局 (< 文本 >)

说明: 文本和格局的对照见下表.

示例:

设置格局("G")	>>>显示绘图区
设置格局("AG")	>>>并排显示绘图区和代数区, 代数区在绘图区的左边
设置格局("S/G")	>>>显示数据区和绘图区, 数据区在绘图区的上面
设置格局("S/(GA)")	>>>显示数据区和代数区和绘图区, 数据区在绘图区和代数区的上面, 绘图区在代数区的左边

 注意: 可以通过 +-号增加视图或减少视图.

字母	视图
A	N 代数区
B	▲ 概率计算器
C	↳ CAS
D	▲ 绘图区2
G	▲ 绘图区
L	☰ 作图过程
P	⚙ 设置
R	数据分析 (Desktop only)
S	▣ 数据区
T	▲ 3D 区

图 2.35

设置格局("+D") >>>在当前视图的右边加上绘图区2
 设置格局("-A") >>>关闭代数区

 **注意:** 指令的文本也可以用数字文本, 文本和字母的对应如下表:

1	AG
2	G
3	SG
4	CG
5	AT
6	B

2.9.28 SetTrace. 设置跟踪

指令

设置跟踪 (< 对象 >, < true | false >)

说明: 打开或者关闭指定对象的追踪

 **注意:** 追踪产生的踪迹, 可以通过滚动鼠标滚轮, 鼠标拖动绘图区, 指令: "放大 (1)" 等多种方式清除.

2.9.29 SetTooltipMode. 设置工具提示模式

指令

设置工具提示模式 (< 数字 >)

说明: 根据下表更改指定对象的工具提示的模式 (超出范围的视为 0):

0	自动
1	开
2	关闭
3	标题
4	下一个 (右侧的) 单元格

2.9.30 SetFixed. 设置固定

指令

设置固定 (< 对象 >, <true|false>)

说明: true 固定对象, false 不固定对象. 要删除固定的对象, 需要先解除固定, 固定的对象不可以移动.

指令

设置固定 (< 对象 >, <true | false>, <true | false>)

说明: 第一个布尔值设置对象是否固定, 第二个布尔值设置对象是否允许选择.

2.9.31 SetActiveView. 设置活动视图

指令

设置活动视图 (< 视图编号 1 或 "G"-绘图区 | 2 或 "D"-绘图区 2|-1 或 "T"-3D 绘图区 | "A"-代数区 | "S"-数据区 | "C"-CAS >)

说明: 更改视图的布局和视觉效果, 字母代表的视图见图 2.35

指令

设置活动视图 (< 平面 >)

说明:

激活指定平面的附加视图.

2.9.32 SetVisibleInView. 设置可见性

指令

设置可见性 (< 对象 >, < 视图编号 1-绘图区 | 2-绘图区 2 >, < 布尔值 >)

说明: 在指定的绘图区 (1 或 2) 显示 (true) 或隐藏 (false) 对象.

2.9.33 SetViewDirection. 设置视图方向

指令

设置视图方向 ()

说明: 返回 3d 视图的默认方向

3D 区指令

设置视图方向 (< 方向 >)

说明: 方向可以是向量, 平面和线段等. 默认有旋转视图方向的动画, 如果第二个参数加上 false, 则视图直接转向指定方向, 没有转向动画.

注意: 如果你执行设置视图方向 ($x + y + z = 1$) 两次, 则有两种可能的结果, 第二种将视图旋转 180° . 为避免歧义, 请使用: "设置视图方向 (向量((0, 0, 1)))", 切不可省略向量两字.

Tips: 可以利用设置视图方向指令, 制作三视图, 其中正视图的方向为 (0,1,0), 侧视图为 (1,0,0), 俯视图为 (0,0,-1).

pause	
play	▶
stop	■
replay	↺
skip_next	▶▶
skip_previous	◀◀
loop	↺↻
zoom_in	🔍
zoom_out	🔍

图 2.36

3D 区指令

设置视图方向 (向量 ((1,0,0))) ⇒ 侧视图, 从左往右看.

**2.9.34 SetFilling. 设置填充****指令**

设置填充 (< 对象 >, < 数字 >)



说明: 更改对象的虚实度, 范围是 0-1, 其中 0 表示透明, 1 表示 100

2.9.35 SetLayer. 设置图层**指令**

设置图层 (< 数字 >)



说明: 给指定对象设置图层, 范围是 0-9 的整数, 越在上层的图层对应的数字越大, 即 0 为底层, 9 为最高层.

2.9.36 SetImage. 设置图片**指令**

设置图片 (< 对象 >, < 图片 >)



说明: 给指定对象填充指定的图片

指令

设置图片 (< 按钮 >, < 文本 >)



说明: 给按钮填充图片, 图片的名字必须加上"', GeoGebra 内置的图片的名字 (英文) 见下图

2.9.37 SetLevelOfDetail. 设置细节级别**指令**

设置细节级别 (< 曲面 >, < 细节级别 0|1 >)



说明: 设置曲面的细节级别, 0 为速度, 1 为质量.

close	×
arrow_up	↑
arrow_down	↓
arrow_forward	→
arrow_back	←
fast_forward	▶▶
fast_rewind	◀◀
zoom_to_fit	🔍
center_view	🔄
help	?
settings	⚙️

图 2.37

2.9.38 SetConditionToShowObject. 设置显示条件

指令

设置显示条件 (< 对象 >, < 条件 >)

说明: 设置对象的显示条件, 显示条件一般指的是布尔值, 例如 $(a>1, a==1, a<>1)$, 或者复选框).

Tips: 要使得某对象始终显示, 不受其他任何操作的影响, 可以把其显示条件设置为 true.

2.9.39 SetLineThickness. 设置线径

指令

设置线径 (< 对象 >, < 数字 >)

说明: 将给定对象的线条的线径设定为 $\frac{N}{2}$ 像素粗, N 是给定的数字, 范围是 0-13 的自然数. 多数对象的线径最小值为 1, 少数对象例如不等式、多边形等的线径最小值为 0, 当线径设为 0 时, 则线不显示.

2.9.40 SetLineStyle. 设置线型

指令

设置线型 (< 对象 >, < 数字 >)

说明: 将给定对象的线型设定为给定数字代表的样式. 数字和线型的对照关系如下: 0-实线, 1-长虚线, 2-短虚线, 3-点虚线, 3. 虚线-点.

2.9.41 SetColor. 设置颜色

指令

设置颜色 (< 对象 >, < Color >)

说明: 改变给定对象的颜色, 颜色用颜色名 (文本) 或者 16 进制字符串 (以 # 开头), 见表 2.4, 字符串可以是 #AARRGGBB 格式或者 #AARRGGBB 格式, 其中 AA 表示透明度 (00 表示完全透明, FF 表示不透明, FF 在十进制中表示 255, 80 在十进制表示 128, 故 80 代表半透明), RRGGBB 分别代表红色绿色和蓝色的比例, 例如 "#80FF0000", 表示透明度为 50%, 红色为 100%, 绿色和蓝色为 0% 的颜色, 即透明度为 50% 的红色.

指令

设置颜色 (< 对象 >, < 红色值 0-1 >, < 绿色值 0-1 >, < 蓝色值 0-1 >)

说明: 更改给定对象的颜色. 红、绿、蓝三色表示对应颜色分量的量, 0 表示最小值和 1 最大值, 超过范围的值用公式: $2|\frac{t}{2} - \text{round}(\frac{t}{2})|$ 映射

2.9.42 SetSeed. 设置种子

指令

设置种子 (< 整数 >)

说明: 为随机数生成器设定种子, 以便后续的随机数将由种子确定. 种子相当于随机数的一个编号, 保证在此种子编号下运行得到的随机数是一致的.

```
设置种子(1)
random() >>>0.21
#在第二次再输入同样的指令,产生的随机数仍然是0.21
```

为了方便使用, 建议该指令放在按钮中的脚本中和随机数指令结合使用.

2.9.43 SetSpinSpeed. 设置转速

指令

设置转速 (< 速度值 >)

说明: 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.

2.9.44 SetCoords. 设置坐标

指令

设置坐标 (< 对象 >, < 横坐标 >, < 纵坐标 >)

说明: 将 2D 视图中自由对象的笛卡尔坐标设置为给定坐标. 此命令使用坐标值, 而不是它们的定义, 因此对象任然保持自由. 3D 区的使用方法类似.

- 这也适用于路径和区域中的点. 该点将被移动到最近的可能位置.
- 此命令也适用于滑块、按钮、复选框、输入框和图像.
- 如果选择了“绝对屏幕位置”选项, 则 x、y 和 z 以屏幕像素为单位.

2.9.45 SetAxesRatio. 设置坐标轴比例

指令

设置坐标轴比例 (< 数值 1 >, < 数值 2 >)

说明: 更改 x 轴:y 轴的值. 3D 区的指令类似.

2.9.46 Textfield. 输入框

指令

输入框 (< 链接对象 >)

说明: 创建一个新的输入框并将链接对象与它关联, 英文名也写做 InputBox.

2.9.47 DataFunction. 数据函数

指令

数据函数 (< 数字列表 1>, < 数字列表 2>)

说明: 生成一个连接点 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, 其中 $x_1, \dots, x_n, y_1, \dots, y_n$ 是输入列表. 在这些点之间, 使用线性插值拟合成函数, 两个列表的长度要保持一致.

注意: 数据函数和折线图的功能基本是一致的, 但折线图 (I1, I2) 是 I1, I2 的附属对象, 即 I1 或 I2 的变化会导致函数的变化, 但数据函数 (I1, I2) 生成函数后, 得到的是一个自由对象, 不随 I1, I2 的变化而变化.

2.9.48 ZoomOut. 缩小

指令

缩小 (< 缩放因子 >)

说明: 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

指令

缩小 (< 缩放因子 >, < 中心点 >)

说明: 基于当前比例将绘图区视图以指定缩放因子以指定的缩放中心点进行缩小.

注意: 缩小 (A) 等同于放大 (1/A).

2.9.49 TurtleUp. 抬笔

指令

抬笔 (< 海龟 >)

说明: 指示给定的海龟停止追踪踪迹.

2.9.50 GetTime. 系统时间

指令

系统时间 ()

说明: 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)、星期几 (文本格式)、星期几数 (1= 周日、2= 周一, 依次类推).

指令

系统时间 ("< 格式 >")

说明: 使用格式作为模板创建文本, 替换以下任何以反斜杠为前缀的字符 (\): 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<https://www.php.net/manual/en/function.date.php>

示例: 系统时间 ("当前的日期为 \l\j\S\F\Y") 返回的结果为:"前的日期为星期一 21 Oct. 2024"

 **注意:** 和其他脚本类指令一样, 仅在执行脚本时有用, 不会随着时间的更新而更新.

2.9.51 ShowLabel. 显示标签

指令

显示标签 (<对象>, <true|false>)

说明: 在绘图区显示或隐藏给定对象的标签.

2.9.52 ShowLayer. 显示图层

指令

显示图层 (<数值>)

说明: 使所有指定图层上的对象显示.

2.9.53 ShowGrid. 显示网格

指令

显示网格 ()

说明: 使当前活动视图显示网格.

指令

显示网格 (<true|false>)

说明:

在激活绘图区显示/隐藏网格.

指令

显示网格 (<视图 1|2>, <true|false>)

说明: 显示/隐藏由数字 1 或 2 或 3 代表的视图的网格 (1: 绘图区, 2: 绘图区 2, 3: 3d 绘图区)

2.9.54 ShowAxes. 显示坐标轴

指令

显示坐标轴 () 显示坐标轴 (true|false)

说明: 同显示网格 (略)

指令

显示坐标轴 (<视图编号 1 或"G"-绘图区 | 2 或"D"-绘图区 2|-1 或"T"-3D 绘图区 | "A"-代数区 | "S"-数据区 | "C"-CAS>, <布尔值>)

说明: 根据布尔值隐藏或显示指定视图, 视图编号和视图的对应关系参照图2.35.

2.9.55 SelectObjects. 选择

指令

选择 ()

说明: 取消所有选定对象.

指令

选择 (< 对象 1>,< 对象 2>,...)

说明: 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.

2.9.56 Pan. 移动视图

指令

移动视图 (<x>, <y>)

说明: 将活动视图向右移动 x 像素, 向上移动 y 像素.

3D 区指令

移动视图 (<x>, <y>, <z>)

说明: 如果活动视图是 3D 视图, 则移动活动视图 (x, y, z) 像素, 如果是 2D 视图, 则移动 (x, y) 像素

2.9.57 HideLayer. 隐藏图层

指令

隐藏图层 (< 图层编号 (0-9)>)

说明: 使给定图层中的所有对象不可见. 该操作不会覆盖对象的显示条件设置, 即有显示条件设置的对象, 不受该指令的影响.

2.9.58 TurtleRight. 右转

指令

右转 (< 海龟 >,< 角度 >)

说明: 海龟按照角度右转.

2.9.59 ReadText. 阅读文本

指令

阅读文本 ("< 文本 >")

说明: 指示屏幕阅读器立即阅读给定文本.

=====



注意: 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.

2.9.60 RunClickScript. 运行单击脚本

指令

运行单击脚本 (< 对象 >)

说明: 运行与对象相关联的“单击时”脚本 (如果有的话).

示例: 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.

2.9.61 RunUpdateScript. 运行更新脚本

指令

运行更新脚本 (< 对象 >).

说明: 运行对象的“更新时”脚本 (如果有的话).

2.9.62 Execute. 执行

指令

执行 (< 文本列表 >)

说明: 执行以文本形式输入在列表中的指令.

注意: 储存在文本中的指令必须为英文指令, 不管程序使用什么语言.

Tips: 这个指令是笔者非常喜爱的指令, 如果熟练使用, 可以完成很多花样. 例如批量设置显示条件, 生成渐变色, 批量重命名等序列和映射无法完成的批量脚本类操作.

示例:

执行{"A=(1,1)","B=(3,3)","C = Midpoint(A, B)"} >>>生成A(1,1), B(3,3), 及AB中点C

示例:

执行(合并({"f_{1}=1","f_{2}=1"},序列("f_{"+(i+2)+"}=f_{"+(i+1)+"}+f_{"+i+"}",i,1, 10))) >>>
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

说明: 序列 ("f_{"+(i+2)+"}=f_{"+(i+1)+"}+f_{"+i+"}",i,1, 10) 这个指令先生成一个文本列表: $f_3 = f_2 + f_1$, " $f_4 = f_3 + f_2$ ", " $f_5 = f_4 + f_3$ ", " $f_6 = f_5 + f_4$ ", " $f_7 = f_6 + f_5$ ", " $f_8 = f_7 + f_6$ ", " $f_9 = f_8 + f_7$ ", " $f_{10} = f_9 + f_8$ ", " $f_{11} = f_{10} + f_9$ ", " $f_{12} = f_{11} + f_{10}$ ", 然后与 {"f_1 = 1", "f_2 = 1"} 合并, 再执行这些指令, 即可得到斐波那契数列的前 12 项.

示例:

执行(序列("SetConditionToShowObject(pic"+i+",n>"+i+")",i,1,100)) >>>将pic1到的显示条件设置为n > 1, pic为n > 2, 依次类推, 其中n为滑动条.

注意: 执行指令要用好, 需要对文本的操作十分熟悉, 比如如何生成动态文本, 如何生成序列文本等.

指令

执行 (< 文本列表 >, < 参数 1 >, < 参数 2 >, ...)

说明: 用%1 代表参数 1, %2 代表参数 2, 依次类推, 最多可以有 9 个参数,

2.9.63 CenterView. 中心定位

指令

中心定位 (< 视图中心设置坐标 (x,y)| 视图中心点 >)

说明: 把绘图区的中心移动到指定点.

2.9.64 Repeat. 重复

指令

重复 (< 数值 >, < 脚本指令 1 >, < 脚本指令 2 >, ...)

说明: 重复脚本指令 n 次, 注意必须为脚本类指令, 即本节出现的指令.

示例: 假如 $a=1$, button1 的单击时脚本为 $a=a+1$, 输入指令: 重复 (10, 运行单击脚本 (button1)), 相当于点击 button1 十次, 得到的结果为 $a=11$.

2.9.65 Rename. 重命名

指令

重命名 (< 对象 >, < 名称 >)

说明: 将指定对象的名称更改为指定名称, 名称必须为文本格式, 且不能使用已存在的名称.

示例:

重命名 (a, "b") >>> 把 a 重命名为 b

2.9.66 TurtleLeft. 左转

指令

左转 (< 海龟 >, < 角度 >)

说明: 海龟按照给定角度左转.

2.10 离散数学

2.10.1 DelauneyTriangulation.Delaunay 三角网

指令

Delaunay 三角网 (< 点列 >)

说明: 创建给定点列的三角网. 返回的对象类型是轨迹. delaunay 三角网的定义见: <https://mathworld.wolfram.com/DelaunayTriangulation.html>

示例:

Delaunay 三角网 ({A, B, C, D, E, F}) >>> 如图 2.38 生成点列的三角网, 可用于分割多边形 .

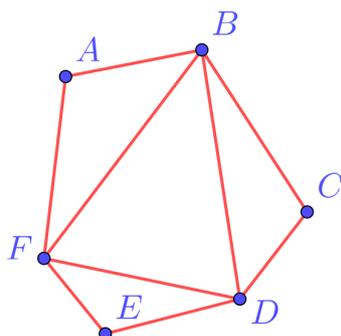


图 2.38

2.10.2 Voronoi.Voronoi 图

指令

Voronoi 图 (< 点列 >)



说明: 由给定点列生成 Voronoi 图, 它的属性是轨迹. Delaunay 三角网和 Voronoi 图是彼此对偶的.

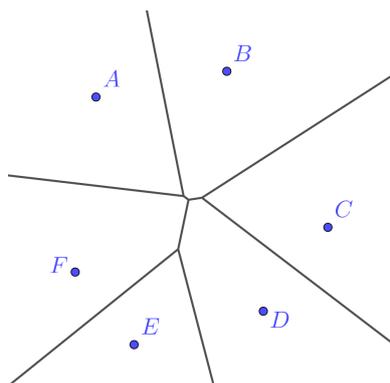


图 2.39

2.10.3 TravelingSalesman. 旅行商问题

指令

旅行商问题 (< 点列 >)



说明: 返回恰好通过每个点一次的最短闭合路径. 返回的对象是一个轨迹. 常可用于封闭图形的涂色.

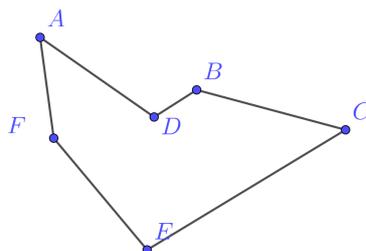


图 2.40

2.10.4 ConvexHull. 凸包

指令

凸包 (< 点列 >)



说明: 创建给定点集的凸包. 返回的对象是一个轨迹. 凸包的定义: 在平面上能包含所有给定点的最小凸多边形叫做凸包.

说明: 凸包的长度等于最小凸多边形的顶点数加 1, 可以通过最前元素 (< 凸包 >, < 长度 >-1) 的指令, 获取多边形的顶点.

2.10.5 ShortestDistance. 最短距离

指令

最短距离 (< 线段列表 >, < 起始点 >, < 终止点 >, < 是否加权? true|false >)



说明: 在由线段列表给定的图形中查找起点和终点之间的最短路径. 如果加权为 false, 则每条边的权重应为 1 (即我们正在寻找边数最少的路径), 否则它是给定线段的长度 (我们是寻找几何最短路径).

2.10.6 MinimumSpanningTree. 最小生成树

指令

最小生成树 (< 点列 >)



说明: 返回给定顶点上完整图的最小生成树, 其中边的权重 w_{uv} 为 u 和 v 之间的欧几里得距离. 生成的对象是一个轨迹.

2.11 列表

2.11.1 Flatten. 扁平列表

指令

扁平列表 (< 列表 >)



说明: 将所有列表合成 1 个 (只留下 1 个大括号)

```
扁平列表({2,3,{5,1},{2,1,{3}}}) >>>{2,3,5,1,2,1,2}
```

2.11.2 Union. 并集

指令

并集 (< 列表 >, < 列表 >)



说明: 求两个列表的并集 (列表的元素满足互异性, 重复元素只保留 1 个), 并集中元素的顺序和原列表一致, 有重复的以第一次出现的为准.

指令

并集 (< 多边形 1>, < 多边形 2>)

说明: 查找两个多边形的并集. 仅适用于多边形不自相交且并集是单个多边形.

2.11.3 Insert. 插入

指令

插入 (< 列表 >, < 列表 >, < 序数位置 >)

说明: 将列表 1 中的全部元素插入到列表 2 中的指定位置, 如果序数为负数, 则序数从列表的右边开始数.

示例:

```
插入({11, 12}, {1, 2, 3, 4, 5}, 3)      >>>{1.2.11.12.3.4.5}
插入({11, 12}, {1, 2, 3, 4, 5}, -2)   >>>{1,2,3,4,11,12,5}
```

注意: 列表 1 的元素被插入到列表 2 中, 而不再作为一个列表整体被插入, 如果要作为 1 个整体插入, 可以见下面的示例.

示例: 假如 $l1=\{2,3\}$, $m1=\{\{1,2\}\{-1,1\},\{0,-1\}\}$, 将 $l1$ 插入到 $l2$ 的第 2 个位置, 分别输入

```
插入(l1, m1, 2)      >>>l2={\{1, 2\}, 2, 3, {\{2, 1\}, {\{0, -1\}\}}
插入({l1}, m1, 2)   >>>m2= {\{1, 2\}, {\{2, 3\}, {\{2, 1\}, {\{0, -1\}\}}
```

读者可以看到这两者之间的区别.

指令

插入 (< 对象 >, < 列表 >, < 序数位置 >)

说明: 将对象插入到列表中的指定位置., 如果序数为负, 则从列表的右边开始计数.

2.11.4 Product. 乘积

指令

乘积 (< 原始数据列表 >)

说明: 计算列表中所有数值的乘积

示例:

```
乘积({2,5,8})      >>>80
```

指令

乘积 (< 数字列表 >, < 最前元素数量 >)

说明: 计算列表中前 n 个元素的乘积

示例:

```
乘积({1, 2, 3, 4}, 3)      >>>6
```

指令

乘积 (< 数据列表 >, < 频数列表 >)

说明: 计算第 1 个列表中的每个元素的相应频数次方的乘积**示例:** 乘积 (20, 40, 50, 60, 4, 3, 2, 1) \gg 1536000000000000即 $20^4 \times 40^3 \times 50^2 \times 60^1 = 1536000000000000$

指令

乘积 (< 表达式 >, < 变量 >, < 起始值 >, < 终止值 >)

说明: 计算通过将给定变量替换为从给定开始值到给定结束值的每个整数来获取的表达式乘积**示例:**乘积(x+1,x,2,3) \gg 12

即 (2+1) (3+1)=12

 **注意:** 在 cas 区, 列表中的元素可以为未定义的变量.

2.11.5 PointList. 点列

指令

点列 (< 列表 >)

说明: 从双元素列表创建点列表.**示例:**点列({{1,2},{3,4}}) \gg {(1,2),(3,4)}

2.11.6 Join. 合并

指令

合并 (< 列表的列表 >)

说明: 将多个子列表合并为一个较长列表.(减少一层大括号)**示例:**合并({{1,2,3},{3,4},{8,7}}) \gg {1,2,3,3,4,8,7} **注意:** 和并集区别的是, 合并指令不会去重. 合并后的列表不对元素进行重新排序.

指令

合并 (< 列表 1>, < 列表 2>, ...)

说明: 把多个列表合并成 1 个列表.**示例:**合并({5, 4, 3}, {1, 2, 3}) \gg {5,4,3,1,2,3}

2.11.7 Unique. 互异

指令

互异 (< 列表 >)



说明: 返回将指定列表中的元素升序排列后的新列表, 重复元素 (不含未定义元素) 只记一次. 可用于数字列表、点列和文本列表等.

示例:

```
互异({1, 2, 4, 1, 4})           >>>{1, 2, 4}
互异({?, ?, 1, 2, 3, 1})       >>>{?, ?, 1, 2, 3}
互异({"a", "b", "Hello", "Hello"}) >>> {"Hello", "a", "b"}
互异({(1,2), (2,1), (1,2)})   >>>{(1,2), (2,1)}
```

2.11.8 Intersection. 交集

指令

交集 (< 列表 1>, < 列表 2>)



说明: 给出包含两个列表中所有共有且不重复的元素的列表.

示例:

```
交集({1,2,3,4}, {3,4,5,6})     >>>{3,4}
交集({1,2,3,3}, {1,3,3})       >>>{1,3}
```

小技巧: 交集 (**ll, ll**), 可以实现对 ll 去重, 且能保持原有元素的排序 (重复的元素以第一次出现的为准), 注意和互异指令的区别.

示例:

```
交集({1,3,3,2,2}, {1,3,3,2,2}) >>>{1,3,2}
```

2.11.9 Reverse. 逆序排列

指令

逆序排列 (< 列表 >)



说明: 将列表的元素倒序排列

示例:

```
逆序排列({(1, 2), (3, 4), (5, 6)}) >>>{(5, 6), (3, 4), (1,2)}
```

2.11.10 Frequency. 频数列表

指令

频数列表 (< 原始数据列表 >)



说明: 输出一个指定的原始数据列表中各个互异值频数的列表

示例:

频数列表({ "a", "a", "x", "x", "x", "b" }) >>>先得到互异的升序排列的列表("a", "b", "x"),再计算频数,因此返回的值为{2,1,3}.

指令

频数列表 (< 是否累积? true|false >,< 原始数据列表 >)

说明: 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.

指令

频数列表 (< 组界列表 >,< 原始数据列表 >).

说明: 即频数分布表, 按照分组, 每组的端点由组界列表确定, 获得每组的频数.

示例:

频数列表({1, 2, 3}, {1, 1, 2, 3}) >>>{2,2}

由组界列表, 分组为 [1,2],[2,3], 除最后一组为闭区间外, 其他区间都是左臂右开. 可知在 [1,2) 上有 2 个数,[2,3] 也有 2 个数, 所以返回值为 2,2

小技巧: 要计算列表 l1 中 1,2,3,4,5,6 这 6 个数的频数为多少, 可以输入指令:

频数列表[1...7,l1]

思考: 频数列表 (l1) 和上述指令的区别是什么?

指令

频数列表 (< 是否累积? true|false >,< 组界列表 >,< 原始数据列表 >).

说明: 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.

指令

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)>)

说明: 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, 即频数/组距的列表.

指令

频数列表 (< 文本列表 1>,< 文本列表 2>)

说明: 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)

示例:

l1= {"a", "b", "b", "c", "c", "c", "c"}

l2={"a", "b", "a", "a", "c", "c", "d"}

频数列表(l1,l2) >>> {{1, 0, 0, 0}, {1, 1, 0, 0}, {1, 0, 2, 1}}

先对 L1, L2 互异得 {"a","b","c"}, {"a","b","c","d"}，第一个列表有 3 个互异元素，对应矩阵有 3 行，第二个列表有 4 个互异元素，对应矩阵有 4 列，因此得到 3×4 的矩阵，行元素之和为第一个列表的频数，列元素之和为第二个列表的频数。

2.11.11 TiedRank. 平秩列表

指令

平秩列表 (< 列表 >)

说明: 返回一个列表，其中的元素是给定列表 L 中元素的序号 (序号由序数列表 (L) 获得). 如果在 L 中有多个相同元素，自动顺延序号，然后相同元素的序号取它们平均值。

示例:

```
平秩列表({4,1,2,3,4,2})      >>>{5.5,1,2.5,4,5.5,2.5}
##先升序排列得{1,2,2,3,4,4},因此元素1的序号为1,2的序号取2和3的平均数2.5,3的序号为4,4的序号取5和6的平均数
5.5,因此得列表 {5.5,1,2.5,4,5.5,2.5}
```

2.11.12 Remove. 去除

指令

去除 (< 列表 1>,< 列表 2>)

说明: 从列表 1 中将列表 2 中的元素去除一次. 如果要去除全部, 使用 $L1 \setminus L2$

示例:

```
去除({1,3,4,4,9},{1,4,5})    >>>{3,4,9}
{1,2,3,3,1}\{1,3}           >>>{2}
```

2.11.13 RemoveUndefined. 去除未定义对象

指令

去除未定义对象 (< 列表 >)

说明: 去除掉列表中所有的未定义对象, 未定义对象在列表中显示为?.

示例:

```
去除未定义对象(序列((-1)^i,i,-3,-1,0.5)) >>>{-1,1,-1}
##在ggb中负数的小数次幂无意义,要和分数指数幂区别.
```

2.11.14 Sort. 升序排列

指令

升序排列 (< 列表 >)

说明: 对数列, 点列, 文本列进行升序排列, 其中文本按照字母顺序, 点按照横坐标的顺序, 数按照数字大小从小到大排列.

示例:

```
升序排列({3, 2, 1})      >>>{1,2,3}
升序排列({"pears","apples","figs"}) >>> {"apples", "figs", "pears"}
升序排列(({3,2),(2,5),(4,1)}) >>>{(2, 5), (3, 2), (4, 1)}
```

 **注意:** 列表在进行升序排列时, 是对值 (value) 进行排列.

指令

升序排列 (< 数值 >, < 关键字列表 >)

说明: 根据相应的第二个列表的关键字对第一个列表的值进行排序.

示例:

```
升序排列(l1,y(l1))      >>>对点列l1,按照纵坐标大小进行排序.
升序排列(l1,abs(l1))    >>>对点列或者数列按照模(绝对值)大小进行升序排列.
升序排列(l1,zip(arg(p,p,l1))) >>>对点列l1按照辅角大小升序排列.
```

Tips: 如果要对列表进行降序排列 (区别逆序排列), 一般有以下方法. 对于数列 l1, 可以输入指令:

```
升序排列(l1, -l1)  ##即按照l1中每个数相反数的大小从小到大排,所以l1中的元素从大到小降序排列.
```

假如 $l1 = \{3, 1, 5\}$, 则 $-l1 = \{-3, -1, -5\}$, 按照 $-l1$ 中元素的大小, 5 的相反数最小, 因此排第一, 1 的相反数最大, 因此排第三, 所以结果为 $\{5, 3, 1\}$. 对于其他列表, 可以用嵌套指令:

```
逆序排列(升序排列(l1))
##先升序排列,再把元素的顺序逆序.
```

2.11.15 RandomElement. 随机元素

指令

随机元素 (< 列表 >)

说明: 返回从列表中随机选择的一个元素 (相同概率). 列表中元素的类型必须相同.

示例:

```
随机元素({3,2,-4,7})      >>>2(4个元素中随机抽取1个)
```

2.11.16 IndexOf. 索引

指令

索引 (< 对象 >, < 列表 >)

说明: 返回对象在列表中第一次出现的位置. 如果对象没有在列表中, 则返回未定义对象.

Tips: 如果要返回对象 a 在列表 l1 中的所有序号 (位置), 使用指令:

```
l1={1,1,2,3,5,1,6}
条件子列(l1(p)==1,p,1...长度(p)) >>>{1,2,6}
```

指令

索引 (< 对象 >, < 列表 >, < 起始索引 >)

说明: 返回对象在列表中起始索引开始第一次出现的位置,

示例:

```
IndexOf(5, {1, 3, 5, 2, 5, 4}, 3)      >>>3
```

指令

索引("< 文本 1>","< 文本 2>")

说明:

返回较短的字符组 (文本 1) 在整个文本 (文本 2) 中第一次出现的位置

指令

索引("< 文本 1>","< 文本 2>",< 起始索引 >)

说明: 返回较短的字符组 (文本 1) 在整个文本 (文本 2) 中起始索引后第一次出现的位置.

示例:

```
索引("Ge", "GeoGebra")      >>>1
##"Ge"是单词的第1位和第2位,取第1个字符"G"的索引1作为整个单词的索引.
索引("Ge", "GeoGebra", 3)    >>>4
```

2.11.17 Take. 提取

指令

提取 (< 列表 | 文本 >,< 起始位置 >)

说明: 返回原列表 (文本) 从起始位置到结尾的元素 (文本).

指令

提取 (< 列表 | 文本 >,< 起始位置 >,< 终止位置 >)

说明: 返回原列表 (文本) 从起始位置到终止位置的元素 (文本).

2.11.18 Sequence. 序列

指令

序列 (< 终止值 >)

说明: 创建从 1 到给定结束值的连续整数列表. 如果结束值小于 1, 返回空列表, 终止值非整数, 则先对终止值取整.

示例:

```
序列(4)      >>>{1,2,3,4}
```

 **注意:** 序列 (k) 等同于 1...k.

指令

序列 (< 起始值 >,< 终止值 >)

说明: 创建从起始值到终止值的连续整数列表 (递增或递减), 序列 (k,n) 等同于 k...n

示例:

```
序列(5,1)          >>>{5,4,3,2,1}
序列(7,13)         >>>{, 8, 9, 10, 11, 12, 13}
```

指令

序列(<起始值>,<终止值>,<增量>)

说明: 创建从最小值到终止值的给定间隔的列表。

示例:

```
序列(7,13,2)       >>>{7,9,11,13}
序列(7,13,4)       >>>{7,11}
序列(13,7,-1.5)    >>>{13, 11.5, 10, 8.5, 7}
```

 **注意:** 列表的最后一个元素不一定等于终止值, 间隔不要求为整数。

指令

序列(<表达式>,<变量>,<起始值>,<终止值>)

说明: 生成使用给定表达式和变量 k 创建的对象列表, k 范围从最小值 a 到最大值 b , 默认间隔(增量)为 1, 且起始值 \leq 终止值, 也可以在第五个参数指定间隔(增量). 表达式中可以没有变量。

示例:

```
序列((2, k), k, 1, 3, 0.5) >>>{(2, 1), (2, 1.5), (2, 2), (2, 2.5), (2, 3)}
序列(x^k, k, 1, 10, 2)    >>>{x, x^3, x^5, x^7, x^9}
序列(random(),k,1,6)      >>>{0.6556, 0.81061, 0.70006, 0.52781, 0.33024, 0.37237}
```

2.11.19 OrdinalRank. 序数列表

指令

序数列表(<列表>)

说明: 返回一个列表, 其元素是指定列表 L 中元素的排序值. 相同的元素按原列表的先后顺序占据不同的序数值

示例:

```
序数列表({4, 1, 2, 3, 4, 2}) >>>{5,1,2,4,6,3}
##先对列表进行升序排列得{1,2,2,3,4,4},因此1的序号为1,2的序号为2,3的序号为4;4的序号为5,6
```

2.11.20 SelectedElement. 选定的元素

指令

选定的元素(<下拉列表>)

说明: 返回一个在列表对应的下拉列表中当前所选择元素. 默认选择的是第一个元素. 另可参阅 [SelectedIndex](#). 选定索引

2.11.21 SelectedIndex. 选定索引

指令

选定索引 (< 下拉列表 >)

说明: 返回一个下拉列表中当前所选择元素的对应索引.

2.11.22 Zip. 映射

指令

映射 (< 表达式 >, < 变量 1 >, < 列表 1 >, < 变量 2 >, < 列表 2 >, ...)

说明: 将指定列表中相应位置的元素依次作为变量代入表达式得到的结果作为新列表的元素. 新列表的长度由给定列表中最短的列表决定, 如果变量数超过列表数, 则最后一个变量对应列表 $1 \dots k$, k 为最短列表的长度.

示例:

```
映射(中点(A, B), A, {P, Q}, B, {R, S}) >>> 创建AR,BS的中点组成的列表
映射[多项式次数(p), p, {x^2, x^3, x^6}] >>> {2, 3, 6}
映射(f(2), f, {x+1, x+3}) >>> {3, 5}
映射(p+q, p, {1, 2, 3}, q, {-1, 1}) >>> {0, 3}
映射(向量(p, q), p, l1, q, 提取(l1, 2)) >>> 用向量连接点列l1.
```

注意: 映射相比于序列, 语句更简洁, 大部分时候可以替代序列的作用. 映射也可以嵌套使用, 比如:

```
映射(映射((p, q), p, 1...2), q, 1...2) >>> {{(1, 1), (2, 1)}, {(1, 2), (2, 2)}}
```

Tips: 用于映射的列表中的元素必须为同一类型的.

2.11.23 Element. 元素

指令

元素 (< 列表 >, < 元素位置 >)

说明: 返回列表中的第 n 个元素.

指令

元素 (< 文本 >, < 文本位置 >)

说明: 返回字符串 (文本) 的第 n 个字符. 该用法在指令提示和在线帮助中都没有提及, 但可以实际使用.

示例:

```
元素("nightrain", 3) >>> "g"
```

指令

元素 (< 矩阵 >, < 行序 n >, < 列序 m >)

说明: 返回矩阵的第 n 行第 m 列的数字

对于 n 维列表 (大括号的层数为 n , 矩阵可以看成是 2 维列表), 索引可以有 n 个.

示例:

```
元素({{1,2},{3,4}},{5,6},{7,8}},1,2,1) >>>3
元素({1,3,2},{0,3,-2},2,3) >>>-2
```

2.11.24 Append. 追加

指令

追加 (< 列表 >, < 对象 >)

说明: 向列表的末尾中追加对象.

指令

追加 (< 对象 >, < 列表 >)

说明:

向列表的开头追加对象.

注意: 如果 l1, l2 都是列表, 则追加 (l1, l2), 表示把 l2 作为元素追加到 l1 的末尾.

```
m1={{1,2},{2,3}}
l1={-1,1}
追加(l1,m1) >>>{-1,1,{1,2},{2,3}}
追加(m1,l1) >>>{{1,2},{2,3},{-1,1}}
```

2.11.25 Classes. 组限

指令

组限 (< 数据列表 >, < 起点 >, < 组的宽度 >)

说明: 给出数据列表等距分组的端点的列表. 由起点和组距确定分组, 组数可以通过 $\text{ceil}(\frac{\text{数据范围}}{\text{组距}})$ 来确定.

示例:

```
组限({0.1,0.2,0.4,1.1},0,1) >>>由于组距为1,起点为0,组数计算得2组,因此区间端点为0,1,2,即返回{0,1,2}
```

指令

组限 (< 数据列表 >, < 组的数量 >)

说明: 给出数据列表的组限 (即分组的端点). 即频率分布直方图中等距分组中, 各分组的端点, 其中组距 = , 组限列表的第一个元素为数据中的最小值, 其他元素成等差数列, 公差等于组距, 最后一个元素为数据中的最大值, 如果组数为 n, 则组限列表的长度为 n+1.

注意: 本指令常结合直方图指令使用. 由组限确定的分组, 除最后一组为闭区间外, 其他都是左闭右开的区间.

2.11.26 Last. 最后元素

指令

最后元素 (< 列表 | 文本 >, < 数字 >)

说明: 由列表 (文本) 的最后 n 个元素 (字符) 构成的列表 (文本), 若不指定 n 则默认为最后一个.

示例:

```
最后元素({1,3,2})      >>>{2}
最后元素({1,2,3},2)   >>>{2,3}
最后元素("GeoGebra")  >>>"a"
最后元素("GeoGebra",3) >>>"bra"
```

参见 [First. 最前元素](#)

2.11.27 First. 最前元素

指令

最前元素 (< 列表 | 文本 >, < 数字 > 🔥

说明: 由列表 (文本) 的最前 n 个元素 (字符) 构成的列表 (文本), 若不指定 n 则默认为第一个.

示例:

```
最前元素({1, 4, 3})    >>>{1}
最前元素({1, 4, 3},2) >>>{1,4}
最前元素("GeoGebra")  >>>"G"
最前元素("GeoGebra",3) >>>"Geo"
```

Tips: 最前元素和最后元素用于列表时返回的结果都是列表, 这 2 个指令同样也适用于轨迹 (凸包之类对象).

参见 [Last. 最后元素](#)

2.12 逻辑

2.12.1 Relation. 关系

指令

关系 (< 列表 >) 🔥

说明: 显示一个消息框, 该消息框提供有关两个或多个 (最多 4 个) 几何对象之间的关系的的信息 (用数值检验).

示例:

关系 ({A,B,C,D})

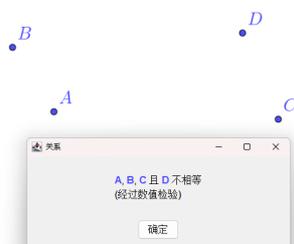


图 2.41

指令

关系 (< 对象 1>,< 对象 2>)

说明:

显示一个消息框, 该消息框提供有关两个对象之间关系的信息. 这个指令可以帮助您了解两个对象是否:

- 两线垂直
- 两条线平行
- 点位于直线或圆锥上
- 直线与圆锥相交或相切
- 三点共线或三线共点
- 四点共圆或四点共线

2.12.2 If. 如果

指令

如果 (< 条件 >,< 结果 >)

说明: 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

示例: 设 $n=3$

```
If(n==3, x + y = 4)      >>>x+y=4
如果(n==4,x+y=4)        >>>不产生任何对象(可能是个bug)
```

指令

如果 (< 条件 >,< 结果 >,< 否则 >)

说明: 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, 注意: 两个对象必须属于同一类型.

```
如果(n==3, x + y = 4, x - y = 4)    >>>n=3时返回x+y=4, 否则返回x-y=4
如果(n==3,1,x+y=1)                  >>>当n=3, 返回1, 否则返回未定义对象?, 因为1和x+y=1类型不同.
```

示例: 限定定义域的函数

```
If(1<=x<=2,x^2)                >>>画出y=x^2在1到2上的图像
```

示例: 限定值域的函数

```
如果(tan(x)<1,tan(x))            >>>画出y<1的正切函数图像
```

但有些指令就是非法的, 比如圆为 $c,l1=\{A,B,C,D,E\}$

```
条件计数(x c,l1)                >>>错误, 参数不符合
```

指令

条件计数 (< 条件 >,< 变量 >,< 列表 >)

说明: 这种用法变量可以用其他的字母, 比如 p,q, 用法更灵活.

示例:

```
条件计数(x(p)<3,p,l1)           >>>输出l1中满足横坐标<3的点的元素个数.
```

参见KeepIf. 条件子列

2.12.3 KeepIf. 条件子列

条件子列和条件计数的用法完全一致, 条件计数返回满足条件的元素个数, 而条件子列返回所有满足条件的元素组成的列表.

指令

条件子列 (< 条件 >, < 列表 >)

说明: 创建一个列表, 包含列表中满足条件的元素, 变量只能是 x,y,z.

注意: 官网或者旧版指令汇编中对于条件的说法, 笔者认为是不对的, 官网的说明是这样的:”用于其它对象类型的列表时, 只能使用 x 常量或 $x \neq$ 常量这两种条件判断的形式之一.”, 而事实上可以用的条件不止这两种.

指令

条件子列 (< 条件 >, < 变量 >, < 列表 >)

说明:

创建一个列表, 包含列表中满足条件的元素. 这种用法变量可以用其他的字母, 比如 p,q, 用法更灵活参加??

2.13 数据区

2.13.1 Cell. 单元格

指令

单元格 (< 列序 >, < 行序 >)

说明: 返回表格区中指定列和行的单元格内容 (可视作该单元格的附属对象, 会随着单元格内容的变化而变化)

示例:

单元格(1,2) >>>返回单元格A2中的对象, 若A2为空, 则返回未定义对象

注意: 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. 一般需要调用的单元格在作图过程中早于这个命令.

2.13.2 CellRange. 单元格区域数值列表

指令

单元格区域数字列表 (< 起始单元格 >, < 终止单元格 >)

说明: 返回从起始单元格到终止单元格区域的对象列表.

注意: 尽管指令名为区域数值列表, 但实质上可以返回的不只是数值列表, 也可以是点列等, 故该翻译有待商榷 (编者注).

示例: 已知 $A1=1, A2=2, A3=3$

单元格区域数字列表(A1, A3) >>>{1, 2, 3}

已知 $A1=(1,1), A2=(0,1), B1=(0,0), B2=(2,3)$

单元格区域数字列表(A1, B2) >>>{(1, 1), (0, 1), (0, 0), (2, 3)}

2.13.3 ColumnName. 列名称

指令

列名称 (< 单元格 >)

说明: 返回单元格的列名称 (以文本格式)

示例:

列名称(A1) >>>"A"

单元格可以是空单元格.

2.13.4 Column. 列序

指令

列序 (< 数据区单元格 >)

说明: 返回单元格的列序 (以数字形式)

示例:

列序(A1) >>>1

2.13.5 FillCells. 填充单元格

指令

填充单元格 (< 单元格区域 >, < 对象 >)

说明: 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, 填充好后, 单元格中的对象为自由对象, 不附属于任何对象.

 **注意:** 区域的写法和 excel 类似, 写作 A1:B3 之类的, 单元格的名称是字母 + 数字的形式, 不允许使用 A:A.

指令

填充单元格 (< 单元格区域 >, < 列表 >)

说明: 将列表中的对象填充到区域中, 填充时先行再列, 生成的单元格是自由对象, 不附属于列表. 如果单元格比列表短, 则依次填充到最后一个单元格为止, 如果单元格比列表长, 则循环填充列表中的对象.

指令

填充单元格 (< 单元格 >, < 列表 >)

说明: 从指定单元格开始向右填充列表中的对象, 生成的单元格为自由对象, 即独立于列表.

Tips: 如果要想下填充, 需要将列表转置为矩阵.

指令

填充单元格 (< 单元格 >, < 矩阵 >)

说明: 将值从矩阵复制到电子表格中. 矩阵的左上角与给定的单元格匹配. 生成的单元格是自由对象, 即独立于矩阵.

2.13.6 FillColumn. 填充列

指令

填充列 (< 列序 >, < 列表 >)



说明: 将列表中的对象填充到指令列中 (从第一行开始), 填充好的对象为自由对象, 即独立于列表.

2.13.7 FillRow. 填充行

指令

填充行 (< 行序 >, < 列表 >)



说明: 将列表中的对象填充到指定行中 (从第一列行开始), 填充好的对象为自由对象, 即独立于列表.

2.13.8 Row. 行序

指令

行序 (< 数据区单元格 >)



说明: 返回单元格的行序 (以数字形式), 单元格可以为空.

Tips: 对于 3 个填充指令, 如果要填充的列表或矩阵不是已存在的列表, 则会自动生成 1 个列表, 并将列表中的元素填充到单元格中, 且单元格为自由对象独立于列表或矩阵.

2.14 统计

2.14.1 Mad.mad

释义: Mad 没有对应的中文指令, 翻译为平均绝对离差, 定义为:

$$\frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$$

是各数据与平均值的离差的绝对值的平均数

指令

mad (< 数列 >)



说明: 计算一系列数的平均绝对离差.

指令

mad (< 数字列表 >, < 频率列表 >)



说明:

计算给定数字的加权平均绝对离差.

2.14.2 Mean

释义: Mean 即平均数, 在当前版本 Mean 不再翻译成中文, 但输入中文平均数也可以使用, 并会自动转换为 mean, 类似的有 Mad 指令.

指令

平均数 (< 原始数据列表 >)



说明: 计算列表中所有元素的算数平均数. 若 mean 在 cas 使用, 数字列表中可以含有未定义的字母.

CAS 指令

`mean(a,b,c) »>(a+b+c)/3`



指令

mean(< 数字列表 >, < 频数列表 >)



说明:

计算加权平均数.

2.14.3 SigmaXX

释义: SigmaXX 即 $\sum x^2$, 即横坐标的平方和.

指令

SigmaXX(< 原始数据列表 >)



说明: 计算给定点列的横坐标的平方和

指令

SigmaXX(< 原始数据列表 >)



说明:

计算所给数据的平方和, 如果要计算方差可以利用公式 $\overline{x^2} - \bar{x}^2$.

示例: 如果数列为 l1, 长度为 n

```
sigmaXX(l1) >>> 返回数据的平方和
sigmaxx(l1)/n-mean(l1)^2 >>> l1 的方差
```

指令

sigmaXX(< 数据列表 >, < 频数列表 >)



说明: 计算加权横坐标的平方和

2.14.4 SigmaXY

释义: SigmaXY: $\sum xy$, 即横纵坐标乘积的平方和.

指令

sigmaXY[< 点列 >]



说明: 计算点列横坐标与纵坐标的乘积之和. 可以利用 $\text{cov}(x,y)=E(xy)-E(x)E(y)$ 公式来计算协方差, 用 ggb 的指令来写为:

```
sigmaXY(l1)^2/n-meanX(l1)meanY(l1) >>> l1 的协方差.
```

指令

sigmaXY(<x 坐标列表>,<y 坐标列表>)



说明: 计算横纵坐标的乘积之和.

2.14.5 SigmaYY

释义: sigmaYY 即 $\sum y^2$, 即纵坐标的平方和.

指令

sigmaYY[<点列>]



说明: 计算纵坐标的平方和.

2.14.6 Spearman.Spearman 秩相关系数

释义: Spearman 秩相关系数被定义成等级变量之间的皮尔逊相关系数. 对于样本容量为 n 的样本, n 个原始数据被转换成等级数据, 相关系数:

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

指令

Spearman 秩相关系数 (<点列>)



说明: 计算点列的横坐标与纵坐标之间的 Spearman 秩相关系数

示例: l1={(-3, 4), (-1, 4), (-2, 3), (1, 3), (2, 2), (1, 5)}

Spearman 秩相关系数(l1) >>>-0.37

指令

Spearman 秩相关系数 (<列表>,<列表>)



说明: 计算两组数之间的 Spearman 秩相关系数

示例:

Spearman 秩相关系数({3, 2, 4, 5, 1, 6, 8, 9},{5, 6, 8, 2, 1, 3, 4, 7}) >>>0.24

2.14.7 stdev

释义: 即样本标准差, 曾用 SampleSD 作为英文指令 (仍可使用), 定义为:

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

指令

stdev(<列表>)



说明: 返回列表的样本标准差.

指令

stdev(< 原始数据列表 >, < 频数列表 >)



说明:

返回加权样本标准差.

示例:

stdev({1, 2, 3}) >>>1

stdev({1, 2, 3, 4}, {1, 1, 1, 2}) >>>1.3

CAS 指令

stdev(1, 2, a)



返回:

$$\frac{\sqrt{a^2 - 3a + 3}}{\sqrt{3}}$$

2.14.8 stdevp

释义: 即标准差, 曾用 SD 作为英文指令 (仍然可用), 定义为:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

指令

stdevp(< 列表 >) stdevp(< 原始数据列表 >, < 频数列表 >)



说明: 计算标准差和加权标准差

2.14.9 Sxx

释义: 在统计学中, S_{xx} .

$$S_{xx} = \sum(x_i - \bar{x})^2 = \sum(x^2) - \frac{(\sum x)^2}{n}$$

指令

Sxx(< 点列 >) Sxx(< 数列 >)



说明: 计算点列的横坐标的离差平方和或数列的离差平方和.

2.14.10 Sxy

释义: sxy 即协方差, 计算方式为:

$$S_{xy} = \sum xy - \frac{(\sum x)(\sum y)}{n}$$

指令

sxy(< 点列 >)



说明: 计算点列的协方差

指令

sxy(< 列表 >, < 列表 >)

说明:

计算两组数的协方差

2.14.11 Syy

释义: Syy 类似 Sxx 可称为纵坐标的离差平方和, 计算公式:

$$Syy = \sum (y_i - \bar{y})^2 = \sum (y^2) - \frac{(\sum y)^2}{n}$$

指令

Syy(< 点列 >)

说明: 计算点列纵坐标的离差平方和.

2.14.12 TTest.t 检验

指令

t 检验 (< 样本数据列表 >, < 假设平均数 >, < 尾部” <”-总体均值小于假设平均数 | ” >”-总体均值大于假设平均数 | ” ≠”-总体均值不等于假设平均数 >)

说明: 使用给定的样本数据列表对总体均值执行单样本 t 检验. 假设均值为在原假设中假设的总体均值. 尾部的可能值为 “<”、“>” 和 “≠”. 这个指定了替代假设如下:

- ”<”= 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

指令

t 检验 (< 样本平均数 >, < 样本标准差 >, < 样本容量 >, < 假设平均数 >, < 尾部” <”-总体均值小于假设平均数 | ” >”-总体均值大于假设平均数 | ” ≠”-总体均值不等于假设平均数 >)

说明: 使用给定的样本统计量对总体均值执行单样本 t 检验. 假设的均值和尾部的定义如上. 结果以列表形式返回为 {可能性, t 检验统计数} 的列表

2.14.13 Percentile. 百分位数

指令

百分位数 (< 数据列表 >, < 百分数 >)

说明: 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 “ $0 < p \leq 1$ ” 内.

 **注意:** GGB 中百分位数的计算方法和人教 A 版 2019 版高中数学教科书中的定义不同, 具体计算方法如下:

- 1 首先获取列表的长度 n , 并且对列表进行升序排列得到列表 l_1 .

2 计算 $p_1 = p \times (n + 1)$

3 计算 $p_2 = \max(1, \text{floor}(p_1)), p_3 = \min(n, \text{ceil}(p_1))$

4 则第 p 百分位数 $= l_1(p_2) + (p_1 - p_2) \times (l_1(p_3) - l_1(p_2))$ (线性插值)

由定义可知, 大多数时候第 75 百分位数不等于第三四分位数, 第一四分位数不等于第 25 百分位数.

示例:

百分位数({1, 2, 3, 4}, 0.25) >>> 1.25

百分位数({1, 2, 3, 4}) >>> 1.5

计算过程如下:

{1, 2, 3, 4} 的长度 $n = 4$, 升序排列后的列表 $l_1 = \{1, 2, 3, 4\}$,

$p_1 = 0.25 \cdot (4 + 1) = 1.25$

$p_2 = \max(1, \text{floor}(1.25)) = 1, p_3 = \min(4, \text{ceil}(1.25)) = 4$

因此第 25 百分位数 $= l_1(1) + (1.25 - 1) \times (l_1(4) - l_1(1)) = 1 + 0.25 \times (4 - 1) = 1.25$

参见 Q1. 第一四分位数和 Q3. 第三四分位数.

2.14.14 ZProportionEstimate. 单比例 z 估计

指令

单比例 Z 估计 (< 样本比例 >, < 样本容量 >, < 置信水平 >)

说明: 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}.

2.14.15 ZProportionTest. 单比例 z 检验

指令

单比例 z 检验 (< 样本比例 >, < 样本容量 >, < 假设比例 >, < 尾部 > <“-总体均值小于假设平均数 |” >“-总体均值大于假设平均数 |” <“≠”-总体均值不等于假设平均数 >)

说明: 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.

2.14.16 TMeanEstimate. 单均值 t 估计

指令

单均值 t 估计 (< 样本数据列表 >, < 置信水平 >)

说明: 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 {置信下限, 置信上限} 的列表中.

指令

单均值 t 估计 (< 样本平均数 >, < 样本标准差 >, < 样本容量 >, < 置信水平 >)

说明:

使用给定的样本统计量和置信度计算总体均值的 t 置信区间估计值水平. 结果以列表形式返回为 {置信下限、置信上限}.

2.14.17 ZMeanEstimate. 单均值 z 估计**指令**

单均值 Z 估计 (< 样本数据列表 >, < 标准差 >, < 置信水平 >)

说明: 使用给定的样本数据 (总体标准) 计算总体均值的 Z 置信区间估计值偏差和置信水平. 结果以列表形式返回为 {置信下限、置信上限}.

指令

单均值 z 估计 (< 样本平均数 >, < 标准差 >, < 样本容量 >, < 置信水平 >)

说明:

使用给定的样本统计量 (总体) 计算总体均值的 Z 置信区间估计值标准差和置信水平. 结果以列表形式返回为 {置信下限、置信上限}.

2.14.18 ZMeanTest. 单均值 z 检验**指令**

单均值 z 检验 (< 样本数据列表 >, < 标准差 >, < 假设平均数 >, < 尾部" <"-总体均值小于假设平均数 | ">"-总体均值大于假设平均数 | "≠"-总体均值不等于假设平均数 >)

说明: 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.

指令

单均值 z 检验 (< 样本平均数 >, < 标准差 >, < 样本容量 >, < 假设平均数 >, < 尾部" <"-总体均值小于假设平均数 | ">"-总体均值大于假设平均数 | "≠"-总体均值不等于假设平均数 >)

说明:

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同上.

2.14.19 Q3. 第三四分位数

参见 `namerefQ1`

指令

第三四分位数 (< 原始数据列表 >)

说明: 求列表的第三四分位数

指令

第三四分位数 (< 原始数据列表 >, < 频数列表 >)

说明: 计算加权第三四分位数

示例:

第三四分位数({1, 2, 3, 4}, {3, 2, 4, 2}) >>>3

由频数列表可知共有11项,故Q3取第9项3

2.14.20 Q1. 第一四分位数

释义:Q1 第一四分位数又做 Quartile1, 在 GGB 中, 计算 Q1 和 Q3 的方法称为 **Moore & McCabe Method**, 同样与人教版教科书上的定义不同. 简略的计算方法如下:

1. 先升序排列给定的列表, 记为 l_1 , 记列表长度为 n ,
2. 先取列表的前一半, 若 n 为偶数, 取前 $\frac{n}{2}$ 项, 若 n 为奇数, 则取前 $\frac{n-1}{2}$ 项, 把项数记作 m ,
3. 再取剩下 m 项的中间项, 如果 m 为奇数, 则取第 $\frac{m+1}{2}$ 项为 Q1, 若 m 为偶数, 则取第 $\frac{m}{2}$ 和第 $\frac{m}{2} + 1$ 项的平均数作为 Q1
4. 同理可求得 Q3.

可知: $n=4k, 4k+1$ 时, Q1 取第 k 和第 $k+1$ 项的平均数, 当 $n=4k+2, 4k+3$ 时, Q1 取第 $k+1$ 项.

$n=4k$ 时, Q3 取 $3k$ 和 $3k+1$ 项的平均数, $n=4k+1$ 时, 取第 $3k+1$ 和 $3k+2$ 项的平均数, $n=4k+2$ 时, Q3 取 $3k+2$ 项, $n=4k+3$ 时, Q3 取 $3k+3$ 项.

指令

第一四分位数 (< 原始数据列表 >)

说明: 求列表的第一四分位数

示例:

Q1{1,2,3,4} >>>1.5

计算方法: 有 4 项, 故 Q1= 第 1 和第 2 项的平均数 1.5.

示例:

第一四分位数({1,2,3,4,5,6}) >>>2

计算方法: 有 6 项, 故 Q1= 第 2 项, 即 Q1=2.

指令

第一四分位数 (< 数据列表 >, < 频数列表 >)

说明: 计算加权第一四分位数.

示例:

第一四分位数({1, 2, 3, 4}, {3, 2, 4, 2}) >>>1

由频数列表可知有 11 项, 即 $4 \times 2 + 3$, 故 Q1 取第 3 项, 即 Q1=1 参见 Q3. [第三四分位数](#)

2.14.21 HarmonicMean. 调和平均数

释义: 数值倒数的平均数的倒数称为调和平均数.

指令

调和平均数 (< 数据列表 >)



说明: 返回给定数据列表的调和平均数.

2.14.22 FitLog. 对数拟合

指令

对数拟合 (< 点列 >)



说明: 计算对数拟合曲线.

示例:

对数拟合($\{(e, 1), (e^2, 4)\}$) $\ggg -2+3\ln(X)$

2.14.23 FitPoly. 多项式拟合

指令

多项式拟合 (< 点列 >, < 多项式次数 >)



说明: 计算点列的多项式回归曲线, 默认的次数为点列长度-1, 点列的输入方式可以用列表的形式输入, 也可以依次输入点.

示例:

多项式拟合($\{(-1, -1), (0, 1), (1, 1), (2, 5)\}, 3)$ $\ggg x^3 - x^2 + x - 1$ 多项式拟合($(-1, -1), (0, 1), (1, 1), (2, 5)$) $\ggg x^3 - x^2 + x - 1$

Tips: 可用于计算过三点的抛物线方程.

指令

多项式拟合 (< 手绘函数 >, < 多项式次数 >)



说明: 计算给定阶数的多项式回归模型, 该模型适合使用手绘形状工具绘制的函数.

2.14.24 Variance. 方差

指令

方差 (< 原始数据列表 >)



说明: 计算数据的方差.

指令

方差 (< 数据列表 >, < 频数列表 >)



说明: 计算加权方差

2.14.25 ANOVA. 方差分析

指令

方差分析 (< 数据列表 1>, < 数据列表 2>, ...)



说明: 对指定的数列表执行单因子方差分析. 结果返回一个形如 {P 值, F 检验统计量} 的列表.

2.14.26 SDX. 横坐标标准差

指令

横坐标标准差 (< 点列 >)



说明: 计算点列的横坐标的标准差.

示例:

横坐标标准差({(1,1), (2,2), (3,1), (3,3), (4,2), (3,-1)}) >>>0.97

2.14.27 MeanX. 横坐标平均数

指令

横坐标平均数 (< 点列 >)



说明: 计算横坐标的平均数.

2.14.28 GeometricMean. 几何平均数

释义: 几何平均数 = $\sqrt[n]{x_1 x_2 \dots x_n}$

指令

几何平均数 (< 数据列表 >)



说明: 计算几何平均数.

示例:

几何平均数(13,7,26,5,19) >>>11.76

2.14.29 RootMeanSquare. 均方根

释义: 也称方均根, 英语简称 RMS, 计算方法为:

$$\text{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$$

指令

均方根 (< 数字列表 >)



说明: 返回给定数据列表的均方根.

2.14.30 ChiSquaredTest. 卡方检验

指令

卡方检验 (< 矩阵 >)

说明: 执行卡方检验, 比较给定的观察计数矩阵与由独立性假设确定的预期计数矩阵的对比. 预期计数矩阵是在内部计算的. 每个预期计数都是从的行和列总计中找到的使用规则的给定观测计数矩阵:

$$\text{预期计数} = \frac{\text{行合计} \times \text{列合计}}{\text{观察到的总数}}$$

结果返回在形式为 { 概率值, 卡方检验统计量 } 的列表中.

示例:

卡方检验({{1,2,1},{3,2,3}}) >>>{0.69, 0.75}

指令

卡方检验 (< 列表 1>,< 列表 2>)

说明: 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.

指令

卡方检验 (< 观察矩阵 >, < 期望矩阵 >)

说明: 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.

2.14.31 RSquare. 可决系数 R 方

指令

可决系数 R 方 (< 点列 >,< 函数 >)

说明: 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 $R^2 = 1 - \frac{SSE}{S_{yy}}$. SSE 即 2.14.46, S_{yy} 即 2.14.11.

示例:

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) >>>0.28

2.14.32 FitLogistic. 逻辑斯蒂曲线拟合

指令

逻辑斯蒂曲线拟合 (< 点列 >)

说明: 以 $\frac{a}{1+be^{-kx}}$ 的形式返回逻辑斯蒂回归曲线. 第一个和最后一个数据点应该非常接近曲线. 列表应至少有 3 个点, 最好是更多.

2.14.33 FitPow. 幂函数拟合

指令

幂函数拟合 (< 点列 >)



说明: 返回形如 ax^b 的拟合曲线.

示例:

幂函数拟合({(1,1),(3,2),(7,4)}) >>>0.97x^{0.71}

2.14.34 Fit. 拟合曲线

指令

拟合曲线 (< 点列 >, < 函数列表 >)



说明: 返回最适合列表中的点的函数的线性组合.

指令

拟合曲线 (< 点列 >, < 函数 >)



说明:

返回一个函数, 该函数拟合列表中相对于指定模型的最小平方误差的点. 给定的模型函数必须依赖于一个或多个滑块, 这些滑块被视为要优化的参数的起始值. 非线性迭代可能不会收敛, 但是将滑块调整到更好的起点可能会有所帮助.

2.14.35 FitLineX. 拟合直线 X

指令

拟合直线 X (< 点列 >)



说明: 计算点横坐标关于纵坐标的回归线.

示例:

拟合直线X({(-1,3),(2,1),(3,4),(5,3),(6,5)}) >>>y=1.1x-0.1

2.14.36 FitLineY. 拟合直线 Y

指令

拟合直线 Y (< 点列 >)



说明: 计算点纵坐标关于横坐标的回归线.

2.14.37 TTestPaired. 配对样本 t 检验

指令

配对样本 t 检验 (< 样本数据 1 列表 >, < 样本数据 2 列表 >, < 尾部 " <" - 总体均值小于假设平均数 | " >" - 总体均值大于假设平均数 | " ≠" - 总体均值不等于假设平均数 >)

说明: 使用给定配对样本数据列表执行一个配对 T 检验. 尾部有 " <"、" >"、" ≠" 可能值提供如下假设选项:

- " <": μ 小于 0
- " >": μ 大于 0
- " ≠": μ 不等于 0.

"(μ 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.

2.14.38 FitGrowth. 生长曲线拟合

指令

生长曲线拟合 (< 点列 >)

说明: 以 ab^x 的形式返回回归方程. 与 2.14.55 类似.

示例:

生长曲线拟合({(0,1), (2,3), (4,3), (6,4)}) $\ggg 1.31 \cdot 1.23^x$

2.14.39 ZProportion2Estimate. 双样本比例 z 估计

指令

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 >)

说明: 使用给定样本计算两个比率之间差值的 Z 置信区间估计值统计数据 and 置信度. 结果以列表形式返回为 {置信下限、置信上限}.

指令

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 >)

说明: 结果以列表形式返回为 {置信下限、置信上限}.

2.14.40 ZProportion2Test. 双样本比例 z 检验

指令

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 >, < 尾部 " <" - 总体均值小于假设平均数 | " >" - 总体均值大于假设平均数 | " ≠" - 总体均值不等于假设平均数 >)

说明: 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.

”<”: 总体比例之差小于 0. 结果返回在是形如 {可能性,Z 检验统计数} 的列表中.

2.14.41 TMean2Estimate. 双样本均值 t 估计

指令

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 >)



说明: 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 {置信下限、置信上限} 的列表中.

指令

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 >)



说明: 使用给定样本计算两个总体均值之间差值的 t 置信区间估计值统计数据 and 置信度. 结果返回在形如 {置信下限、置信上限} 的列表中.

2.14.42 ZMean2Estimate. 双样本均值 z 估计

指令

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 >)



说明: 使用给定样本计算两个比率之间差值的 Z 置信区间估计值统计数据 and 置信度. 结果以列表形式返回为 {置信下限、置信上限}.

指令

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 >)



说明: 使用给定的样本平均数, 标准差, 样本容量计算置信区间, 结果以列表形式返回为 {置信下限、置信上限}.

2.14.43 ZMean2Test. 双样本均值 z 检验

指令

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” <”0|””0|””0 > >).



说明: 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.

指令

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 >)



说明: 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.

2.14.44 TTest2. 双总体 t 检验

指令

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 | ” ≠ ” 0 >, < ?true|false >)

说明: 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}.

指令

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” <” 0 | ” ” 0 | ” ” 0 >, < ?true|false >)

说明:

执行使用给定统计数据的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.

2.14.45 Shuffle. 随机排列

指令

随机排列 (< 列表 >)

说明: 返回相同元素但顺序重新随机排列的新列表.

2.14.46 SumSquaredErrors. 误差平方和

释义: 缩写为 SSE, 定义为: 在统计学和回归分析中, SSE (Sum of Squared Errors, 平方误差和) 是用来衡量模型预测值与实际观测值之间差异的一种指标. 它反映了模型的拟合优度, SSE 的值越小, 表示模型对数据的拟合程度越好.

SSE 的定义可以表示为:

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

指令

误差平方和 (< 点列 >, < 函数 >)

说明: 计算在列表中点的 y 坐标和列表中 x 变量的函数值之间的误差平方和 “SSE” .

2.14.47 CorrelationCoefficient. 相关系数

释义: 相关系数: 英文指令为 CorrelationCoefficient, 曾用名 PMCC, 用于衡量两个变量之间线性关系强度和方向的统计量, 常用小写字母 r 表示, r > 0 表示正相关, r < 0 表示负相关.

指令

相关系数 (< x 坐标列表 >, < y 坐标列表 >)

说明: 用给定的 x 坐标列表、y 坐标列表，计算出相关系数。

指令

相关系数(< 点列 >)

说明:

用给定点的坐标，计算给定点列的相关系数。

示例:

相关系数({1,3,2,1,5,2},{1,6,4,3,3,2}) >>>0.36

相关系数({(1,1),(3,6),(2,4),(1,3),(5,3),(2,2)}) >>>0.36

2.14.48 Covariance. 协方差

释义: 协方差 (Covariance) 是用来衡量两个随机变量之间的共同变动程度的统计量. 它反映了一个变量的变化如何影响另一个变量的变化. 协方差的符号和大小可以告诉我们两个变量之间的关系.

如果有 n 个观测值 $(x_1, x_2, \dots, x_n$ 和 $y_1, y_2, \dots, y_n)$ ，协方差的样本估计可以通过以下公式计算:

$$\text{Cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

其中: \bar{x} 和 \bar{y} 分别是样本 X 和 Y 的样本均值.

指令

协方差(< 数据列表 1>,< 数据列表 2>) 协方差(< 点列 >)

说明: 计算指定列表中元素间的协方差或点列中横坐标和纵坐标之间的相关系数.

示例:

协方差({1,2,3},{1,3,7}) >>>2

协方差({(1,1),(2,3),(3,7)}) >>>2

2.14.49 Sample. 样本

指令

样本(< 列表 >,< 容量 n>,< 是否重复 true|false>)

说明: 从列表中随机选择 n 个元素组成的列表, 如果是否重复为 true, 则每个元素可以多次被重复选择, 如果为 false 则只能被选择一次, 默认为 true.

示例:

样本(1...6,1000,true) >>>每次随机选择1-6, 选出1000个数字(可用以生成1000颗骰子随机试验的点数).

样本(1...1000,100,false) >>>从1-1000个数字中不重复的随机选择100个数字.

Tips: 样本指令可以模拟用于不放回摸球和可放回摸球的实验.

注意:

- 如果布尔值选择 false, 则样本容量必须小于等于列表的长度, 否则会返回未定义对象.
- 列表的元素的类型可以不同, 且列表中的元素也可以是列表.

2.14.50 SampleSDX. 样本点横坐标标准差

指令

样本标准差 (< 原始数据列表 >)



说明: 返回依据给定列表点的 x 坐标样本标准差.

2.14.51 SampleSDY. 样本点纵坐标标准差

指令

样本点纵坐标标准差 (< 点列 >)



说明: 返回依据给定列表点的 y 坐标样本标准差.

2.14.52 SampleVariance. 样本方差

释义:

指令

样本方差 (< 原始数据列表 >)



说明: 返回给定数据列表的样本方差.

指令

样本方差 (< 数据列表 >, < 频数列表 >)



说明:

返回加权样本方差

2.14.53 FitImplicit. 隐函数拟合

指令

隐函数拟合 (< 点列 >, < 次数 >)



说明: 尝试通过点找到阶数为 $n \geq 2$ 的最佳拟合隐式曲线. 您至少需要 $\frac{n(n+3)}{2}$ 个点.

2.14.54 FitSin. 正弦拟合

指令

正弦拟合 (< 点列 >)



说明: 以 $a + b \sin cx + d$ 的形式计算回归曲线.



注意: 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.

2.14.55 FitExp. 指数拟合

指令

指数拟合 (< 点列 >)



说明: 以 ae^{bx} 的形式返回回归曲线. 注意和2.14.38的区别.

2.14.56 Median. 中位数

指令

中位数 (< 原始数据列表 >)



说明: 计算中位数.

指令

中位数 (< 数据列表 >, < 频数列表 >)



说明:

计算加权中位数

示例:

```
中位数({1,2,3}) >>>2
中位数({1,1,8,8}) >>>4.5
中位数({1,2,3},{4,1,3}) >>>1.5
中位数({1,2,3},{4,1,6}) >>>3
```

 **注意:** 求中位数的方法: 先对列表进行升序排列, 如果长度为奇数, 则取中间项, 如果为偶数则取中间两项的平均数.

2.14.57 Mode. 众数

指令

众数 (< 列表 >)



说明: 求列表的众数组成的列表, 如果没有众数, 则返回空列表.

 **注意:** 众数可以有多个, 也可以没有 (当所有数据都是唯一的时候).

示例:

```
众数({1,2,3,4}) >>>{}
众数({1,1,1,2,3,4}) >>>{1}
众数({1,1,2,2,3,3,4}) >>>{1,2,3}
```

2.14.58 Sum. 总和

指令

总和 (< 列表 >)



说明: 求列表所有元素的和.

示例:

总和({1, 2, 3}) >>>a = 6.

总和({x^2, x^3}) >>> f(x) = x² + x³.

总和(序列(i, i, 1, 100)) >>> a = 5050.

总和({(1, 2), (2, 3)}) >>> A = (3, 5).

总和({"a", "b", "c"}) >>> "abc"

指令

总和 (< 列表 >, < 元素数量 >)

说明:

计算列表前 n 个元素的和.

示例:

总和({1, 2, 3, 4, 5, 6}, 4) >>>a=10

指令

总和 (< 列表 >, < 频数列表 >)

说明: 计算列表中元素与其对应频数乘积的总和.

示例:

总和({1,2,3,4,5},{3,2,4,4,1}) >>>40

即总和(11*12)

指令

总和 (< 代数式 >, < 变量 >, < 起始值 >, < 终止值 >)

说明: 在在线帮助中说明这种用法只在 cas 中使用, 实际上在代数区也可以使用. 即求把从变量的起始值到终止值代入到代数式后得到的值的总和. 若代数式为 $f(t)$, 起始值为 a , 终止值为 b , 则总和可以用以下公式表示:

$$\sum_{t=a}^b f(t)$$

在 cas 区使用时终止值为可以为无限大 ∞ , 在代数区使用时必须为有限值. 起始值和终止值应当为整数, 如果输入的值不是整数, 则先进行取整处理.

示例:

总和(n^2, n, 1, 3) >>>14

总和指令在 cas 中可以用于计算数列的前 n 项和公式.

CAS 指令

总和 (r,i,0,n)

返回值: $\frac{r^{n+1}}{r-1} - \frac{1}{r-1}$

CAS 指令

总和 $((1/3)\hat{n}, n, 0, \text{Infinity})$
 $\gg 1.5$



注意: 总和指令可以用于数, 向量, 文本等多种对象的求和, 但要求列表中的对象的类型必须一致.

2.14.59 SDY. 纵坐标标准差

英文指令也写作 SampleSDY.

指令

纵坐标标准差 (< 点列 >)



说明: 计算纵坐标的标准差.

示例:

纵坐标标准差($\{(1, 1), (2, 2), (3, 1), (3, 3), (4, 2), (3, -1)\}$) $\gg 1.25$

2.14.60 MeanY. 纵坐标平均数

指令

纵坐标平均数 (< 点列 >)



说明: 计算纵坐标的平均数.

示例:

纵坐标平均数($\{(0, 0), (3, 2), (5, 1), (2, 1), (2, 4)\}$) $\gg 1.6$

2.15 图表

图表类的指令, 都会在绘图区中绘制相应的图象或图表.

2.15.1 StickGraph. 棒图

指令

棒图 (< 点列 >)



说明: 绘制给定点列的棒图. 从 x 轴作每个点的垂线段.

示例:

棒图($\{(1, 1), (3, 2), (4, 5), (5, 7)\}$)

绘制棒图 2.42, 棒图的值显示为线段的总长度.

指令

棒图 (< 点列 >, < 是否水平? true|false >)



说明: 如果选择 true, 则绘制水平方向棒图, 即从 y 轴作垂线段, 否则绘制垂直方向棒图, 即从 x 轴绘制垂线段.

示例:

棒图($\{(1,1), (3,2), (4,5), (5,7)\}$, true)

绘制水平棒图 2.43 棒图的值显示为线段的总长度。

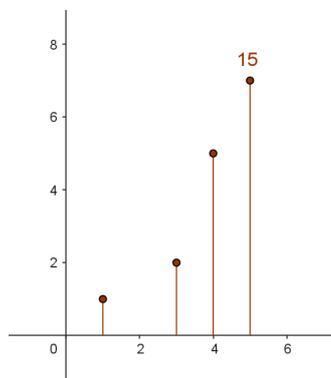


图 2.42

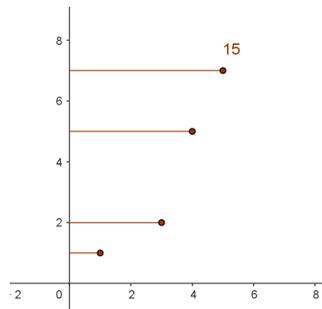


图 2.43

说明: 此外点列也可以以横坐标列表和纵坐标列表分别作为参数的方式输入。

示例:

棒图($\{1,3,4,5\}$, $\{1,2,5,7\}$, true)

2.15.2 ResidualPlot. 残差图

指令

残差图 (< 点列 >, < 函数 >)

说明: 绘制残差图, 所谓残差图即以原点列的横坐标为横坐标, 以残差为纵坐标的点列. 如果原列表中第 n 个元素是点 (a,b) , 那么残差图中第 n 个元素的结果就是 $(a,b-f(a))$.

示例:

残差图($\{(-1,1), (-0.51,2), (0,0.61), (0.51,-1.41), (0.54,1.97), (1.11,0.42), (1.21,2.53), (-0.8,-0.12)\}$, $x^5 + x^4 - x - 1$) $\gg\gg\{(-1,1), (-0.51,2.46), (0,1.61), (0.51,0), (0.54,3.38), (1.11,-0.66), (1.21,0), (-0.8,0)\}$

绘制残差图 2.44.

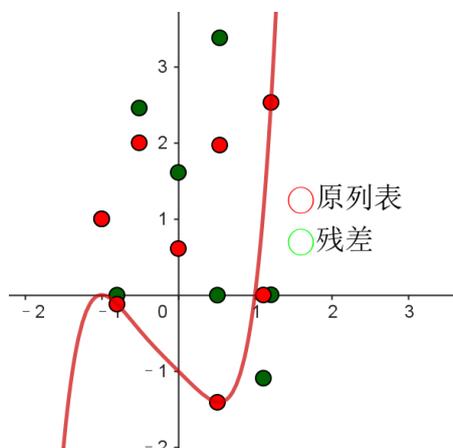


图 2.44

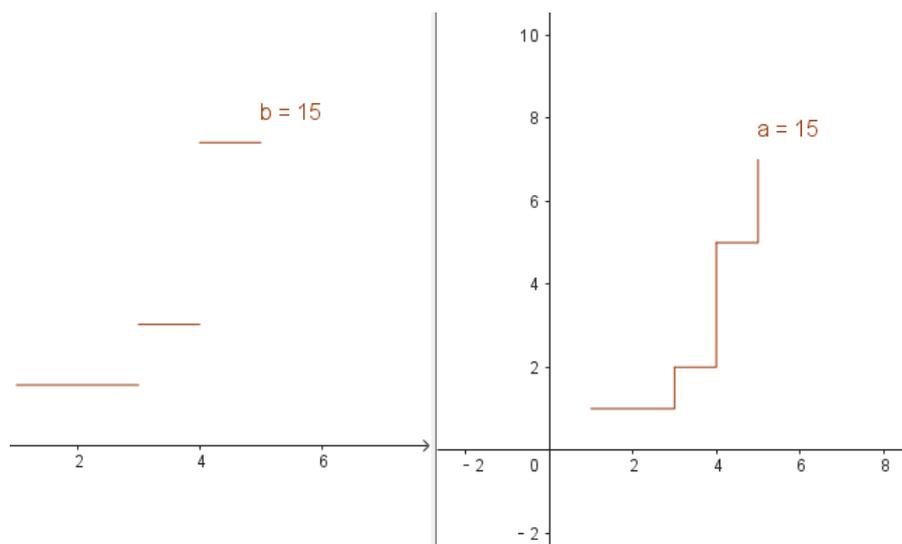


图 2.45

2.15.3 DotPlot. 点阵图

指令

点阵图 (< 原始数据列表 >)



说明: 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 $(n, 1), (n, 2), \dots, (n, k)$.

指令

点阵图 (< 原始数据列表 >, < 堆栈相邻点 (可选) >, < 缩放因子 (可选) >)



说明: 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 $(n, 1), (n, 2), \dots, (n, k)$. 如果选了一个缩放因子 s , 返回点列包括点 $(n, 1s), (n, 2s), \dots, (n, ks)$. 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, 结果等同不选, 即点和点之间是分开状态. 指令点阵图同样适用于文本列表.

示例:

```
点阵图({2,5,3,4,3,5,3}) >>>{(2,1),(3,1),(3,2),(3,3),(4,1),(5,1),(5,2)}
```

```
点阵图({"red","red","red","blue","blue"}) >>>{(1,1),(1,2),(2,1),(2,2),(2,3)}
```

2.15.4 StepGraph. 阶梯图

指令

阶梯图 (< 点列 >, < 是否连接 true|false >)



说明: 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, 如果不选, 则默认为 `false`.

指令

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 >)



说明: 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.

说明: 点列的输入方式可以以横坐标列表和纵坐标列表的方式输入.

2.15.5 StemPlot. 茎叶图

指令

茎叶图 (< 列表 >)

说明: 绘制茎叶图.

指令

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 >)

说明: 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.

2.15.6 ContingencyTable. 列联表

指令

列联表 (< 文本列表 1 >, < 文本列表 2 >)

说明: 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.

示例:

列联表({"男", "女"}, {"近视", "不近视"})

绘制列联表

频数	不近视	近视	总数
女	1	0	1
男	0	2	2
总数	1	2	3

指令

列联表 (< 行数据列表 >, < 列数据列表 >, < 频数表 >)

说明: 用给定的行、列和频数列表绘制列联表.

示例:

列联表({"男", "女"}, {"右撇子", "左撇子"}, {{43, 9}, {44, 4}})

绘制列联表

频数	右撇子	左撇子	总数
男	43	9	52
女	44	4	48
总数	87	13	100

说明: 可以在以上两种用法的基础上加上参数选项, 选项的可能值为: “|”、“_”、“+”、“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.

2.15.7 FrequencyTable. 频数表

指令

频数表 (< 原始数据列表 >, < 缩放因子 (可选) >)

说明: 返回一个表 (作为文本), 第一列为原始数据列表中的唯一元素, 第二列为该元素的频数乘以缩放因子, 默认为 1.

指令

频数表 (< 是否累积? true|false >, < 原始数据列表 >)

说明: 返回一个表 (作为文本), 第一列为原始数据列表中的唯一元素, 第二列如果选 true, 则返回累计的频数, 如果为 false, 则返回频数.

指令

频数表 (< 组界列表 >, < 原始数据列表 >)

说明: 返回频数分布表.

指令

频数表 (< 是否累积? true|false >, < 组界列表 >, < 原始数据列表 >)

说明: 返回累计频数分布表.

指令

频数表 (< 组界列表 >, < 原始数据列表 >, < 是否应用密度 >, < 密度缩放因子 (可选) >)

说明:

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)

指令

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) >)

说明:

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).

2.15.8 FrequencyPolygon. 频数多边形

指令

频数多边形 (< 组界列表 >, < 高度列表 >) 频数多边形 (< 组界列表 >, < 高度列表 >) 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) >)

说明: 绘制对应直方图的频数多边形, 即连接直方图各矩形的终点的折线.

2.15.9 PieChart. 扇形图

指令

扇形图 (< 频数列表 >)

扇形图 (< 频数列表 >, < 点 >, < 半径 >)



说明: 由频数列表按比例创建各扇形切片构成的扇形图, 总比例为 100%, 默认扇形的半径为 3, 圆心为原点, 也可以指定圆心和半径.

示例:

```
PieChart({20, 15, 40, 5, 10, 20})
```

```
PieChart({20, 15, 40, 5, 10, 20}, (3, 3), 5)
```

返回图 2.46.

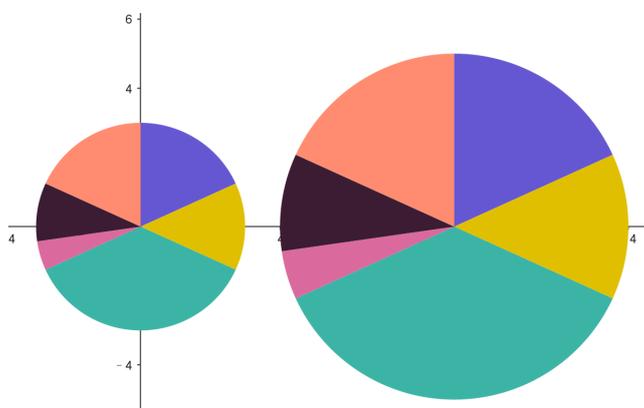


图 2.46

Tips: 饼图的初始基础颜色是 6 个, 如果切片超过 6 个则往这些颜色中掺入一定比例的白色, 得到新的颜色。这些切片的颜色也可以手动调整颜色。

注意: 该指令在旧版的 GGB 中是不存在的, 建议更新到最新版后使用。

2.15.10 BarChart. 条形图

指令

条形图 (< 数据列表 >, < 频数列表 >)



说明: 使用列表中的数据对应的频数创建一个条形图。

注意: 原始数据列表中的数值需要递增排列, 且两个列表的元素个数必须相等。

指令

条形图 (< 原始数据列表 >, < 条形宽度 >)



说明: 创建给定条宽度的原始数据条形图。

示例: 画出 n 个骰子抛掷试验的条形图

```
条形图(样本(1...6, n, true), 0.5)
```

指令

条形图 (< 数据列表 >, < 频数列表 >, < 条形宽度 >).



说明: 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.

指令

条形图 (< 起始值 >, < 终止值 >, < 高度列表 >)

说明: 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.

指令

条形图 (< 起始值 >, < 终止值 >, < 表达式 >, < 变量 >, < 从数值 1 >, < 到数值 2 >)

说明:

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.

指令

条形图 (< 起始值 >, < 终止值 >, < 表达式 >, < 变量 >, < 从数值 1 >, < 到数值 2 >, < 步长 >).

说明:

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.

指令

条形图 (< 原始数据列表 >, < 条形宽度 >, < 竖直缩放因子 (可选) >).

说明: 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

2.15.11 BoxPlot. 箱形图

指令

箱线图 (y 轴方向偏移量, y 轴方向范围, 原始数据列表).

说明: 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.

 **注意:** < 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.

指令

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)

说明:

创建给定统计数据在区间 (初始值, 终止值) 的箱线图.

指令

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 >)

说明:

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.

指令

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>)

说明:

依据频数画数据的箱线图.

2.15.12 LineGraph. 折线图

笔者注: 这个是新增指令中笔者最爱的指令, 可以完成很多花样.

指令

折线图 (<列表 1>,<列表 2>)

说明: 创建一个数据函数, 并画出图像 (用线段连接在列表中定义坐标的点). 列表 1 是包含 x 值的列表列表, 定义为按升序排列的数字, 列表 2 是包含 y 值 (定义为数字) 的列表. 该函数的解析式不会在代数区给出, 但实质上是 $|x - x_i|$ 函数的线性组合.

示例:

折线图(1...3, {2, 4, 0})

如图 2.47 返回函数 $f(x)$ 并画出图像, 函数解析式不会给出, 但实质上是: $f(x) = 1.5|x - 1| - 3|x - 2| + 2.5|x - 3|$

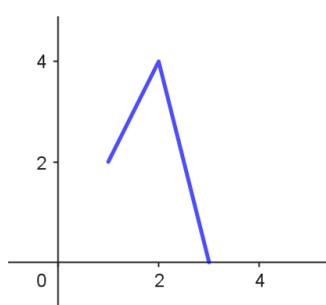


图 2.47

示例: 把任意多边形转换为参数曲线. 令 $A=(1,0), B=(2,1), C=(1,4), D=(0,4), E=(-1,3)$, 将多边形 ABCDE 转换为参数曲线, 如图 2.48

```
l1={A,B,C,D,E,A}
```

```
f=折线图(0...5,x(l1))
```

```
g=折线图(0...5,y(l1))
```

```
曲线(f(t),g(t),t,0,5)
```

注: 如果建立路程和坐标的折线图, 则可实现折线上的匀速.

2.15.13 NormalQuantilePlot. 正态分位数图

指令

正态分位数图 (<原始数据列表>)

说明: 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).

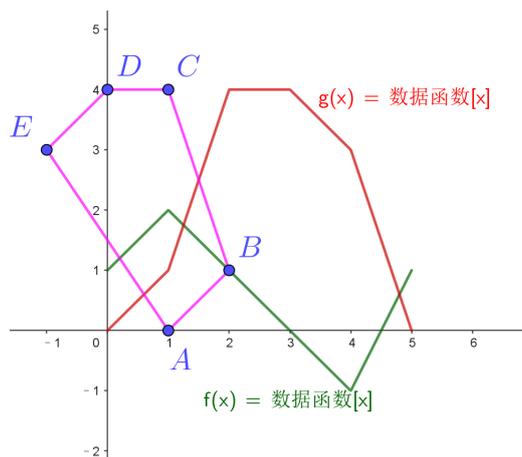


图 2.48

2.15.14 Histogram. 直方图

Tips: 直方图指令往往要和 **Classes**. **组限** 一起使用, 直方图中小矩形的高度可以是频数, $\frac{\text{频数}}{\text{组距}}$, 频率, $\frac{\text{频率}}{\text{组距}}$, 需要结合不同的参数去实现. 另外直方图只有 1 种颜色, 直方图的值为小矩形的面积之和.

指令

直方图 (< 组界列表 >, < 高度列表 >)

说明: 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. 组界列表即每个矩形在 x 轴上的起点和终点, 因此组界列表要比高度列表多 1 个元素. 高度可以是频数, 也可以是频率.

指令

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) >)

说明: 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. 矩形的高度如下规定:

1. 如果后 2 个参数都不加或第三个参数为 true, 缩放因子为 1 或不加, 则高度为每组的 $\frac{\text{频数}}{\text{组距}}$, 即频数分布直方图, 面积之和为 n;
2. 若应用密度为 false, 则高度等于频数;
3. 若应用密度为 true, 缩放因子 = $\frac{1}{n}$, 则高度为 $\frac{\text{频率}}{\text{组距}}$, 即频率分布直方图, 面积之和为 1;
4. 若应用密度为 true, 缩放因子为 $\frac{\text{组距}}{\text{样本容量 } n}$, 则高度为频率.

Tips: 直方图中分组的规定是最后一组为 $[a, b]$, 前面的组为 $[a, b)$, 所有数据必须在分组的范围内.

指令

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) >)

说明: 当是否累积为 true 时, 则频数为累计, 其余的设定和以上的说明保持一致.

2.15.15 HistogramRight. 直方图右和

指令

直方图右和 (< 组界列表 >, < 高度列表 >)

说明: 和直方图指令的意义一样.

 **注意:** 其他的使用方法和规定和直方图都是一致的, 区别是分组的设定上, 分组的规则是 $(a, b]$, 最后一组为 $[a, b]$

2.16 文本

2.16.1 TableText. 表格文本

指令

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 | ...>)



说明: 创建一个由列表对象组成表格的文本对象, 可以控制对齐方式, 表格样式等. 默认情况下, 每个列表都显示在其自己的表行中. 对齐方式还可以指定表格的分隔符, 括号等.

具体的字符 (加双引号) 对应如下:

1. l 左对齐
2. c 居中
3. r 右对齐
4. v 垂直
5. h 水平
6. % 转换为百分比, 并在小数点上对齐
7. . 在小数点对齐
8. a 类似于., 但显示填充的 0
9. p 类似于%, 但显示填充的 0

示例:

表格文本({{2011.56, 2, 3.7, 4}, {1, 4.2, 9, 16.365}}, "v.")

	2011.56	1
创建表格, 数值在小数点数对齐	2	4.2
	3.7	9
	4	16.37

表格文本({{2011.56, 2, 3.7, 4}, {1, 4.2, 9, 16.365}}, "v%")

201156%	100%
200%	420%
370%	900%
400%	1636.5%

表格文本({1, 2}, {3, 4}, "c()")

创建 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

表格文本({1, 2}, {3, 4}, "c|_")

创建

1	2
3	4

表格文本({1, 2}, {3, 4}, "||")

创建 $\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$

表格文本({1, 2}, {3, 4}, "||||")

创建 $\left\| \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} \right\|$

表格文本({{"2x+3y=5"}, {"5x+8y=12"}}, {"v"})

$$\begin{cases} 2x + 3y = 5 \\ 5x + 8y = 12 \end{cases}$$

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"})

返回一个没有虚线和分隔的垂直排列的表格。

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"})

返回一个表格

1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

，其中第 1,2,5 列有竖线，第 1,2,6 行有横线。1 表示有分隔线或框，0 表示没有分隔线或框。

 **注意:** 如果要创建更丰富色彩和样式的表格，可以在表格区设置好表格的样式，再创建表格文本。

此外，还可以在指令的最后，加上 2 个参数，1 个控制表格的最小宽度，一个控制最小高度，以像素为单位。比如：表格 (11,"vc",50,50)，这里的最小高度和宽度和字体的大小有关，小到一定程度就不会再变小。

2.16.2 Split. 拆分

指令

拆分 (< 文本 >, < 要拆分的文本列表 >)

说明: 将给定文本按指定分隔符拆分成文本列表（分隔符不包含在列表中）。

示例:

拆分("3cos(t)cos(2y)", {"cos"}) >>>{"3", "(t)", "(2y)"}

2.16.3 FractionText. 分数文本

指令

分数文本 (< 数字 >)

说明: 转换数值为分数，它以文本 (LaTeX) 形式出现在绘图区。

指令

分数文本 (< 点 >)

说明:

将点的坐标以分数形式显示在绘图区。

指令

分数文本 (< 数字 >, < 布尔值? 是否选择单一分数 >)

说明: 在绘图区中创建并显示包含给定数字的分数形式的 LaTeX 文本。布尔值设置分数的负号的位置。如果为 true，则将显示减号在分子中，如果为 false，则减号将显示在分数前面。

示例:

```
分数文本((1.33,0.8))    >>>(133/100, 4/5)
分数文本(-0.8, true)    >>>-4/5
分数文本(-0.8,false)   >>>-4/5
```

2.16.4 SurdText. 根式文本

指令

根式文本 (< 数字 >)

说明: 以 $\frac{a+b\sqrt{c}}{c}$ 的形式创建数字的根式文本。

指令

根式文本 (< 点 >)

说明: 创建点的根式文本，形式同数字的根式文本。

指令

根式文本 (< 数字 >, < 列表 >)

说明: 创建数字的文本表示形式，改写为列表中常量的组合。如果列表为空，该命令将使用常用的常量列表。

示例:

```
根式文本(2.439230484541326) >>>7+3√3/5
根式文本(3.718281828459045, {exp(1)})>>>e + 1
根式文本(5.382332347441762, {sqrt(2), sqrt(3), sqrt(5)})>>>√5 + √3 + √2
根式文本(1.693147180559945, {ln(2)})>>>ln(2) + 1
```

注意:

1. 如果要在文本工具中使用根式文本，则需要勾选 latex 选项，如果要在文本指令中使用根式文本，则让文本的第四个参数“是否启用 latex 公式”为 true。
2. 由于此命令使用四舍五入的十进制数作为输入对象，因此在某些情况下，结果可能不符合你的预期。
3. 如果找不到数字的合适答案，则会返回数字本身的文本。

2.16.5 LaTeX. 公式文本

英文名又做:FormulaText.

指令

公式文本 (< 对象 >)

说明: 返回对象的 latex 文本, 默认情况下变量都以值代入.

示例:

```
a=2
公式文本(a x^2)>>>"2x^2" (以latex的形式)
```

指令

公式文本 (< 对象 >, < 是否替换变量? true|false >, < 是否显示名称? true|false >)

说明: 将对象以 LaTeX 文本方式返回. 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), 第二个布尔值如果不输入, 则默认为 false.

示例:

```
a=2
f=a x^2
公式文本(f,false) >>>"a x^2"
公式文本(f,true) >>>"2 x^2"
公式文本(f,true,true) >>>"f(x)=2 x^2"
```

Tips: 公式文本可以通过按住鼠标把代数区的对象拖入到绘图区的方式产生, 默认两个布尔值都是 true. 此外可以通过拆分指令从公式文本中提取中对象的名称.

2.16.6 ScientificText. 科学记数法

指令

科学记数法 (< 数字 >)

说明: 创建一个文本, 以科学记数法显示给定的数字.

指令

科学记数法 (< 数字 >, < 有效数字位数 >)

说明: 创建一个文本, 以科学记数法显示给定的数字, 四舍五入为精度指定的有效数字.

2.16.7 ContinuedFraction. 连分式

指令

连分式 (< 数字 >)

说明: 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} .

指令

连分式 (< 数字 >, < 层级 >)

说明:

创建近似于给定数字的连续分数. 商的数量小于或等于层级, 但永远不会超过达到 10^{-8} 的数值精度所需的商数.

指令

连分式 (< 数字 >, < 层级 >, < 速记? true|false >)



说明: 创建近似于给定数字的连续分数. 如果指定了层级, 则商数小于或等于层级, 但永远不会超过达到 10^{-8} 的数值精度所需的商数. 当速记为 true 时, LaTeX 文本使用较短的语法, 并包含连续分数的整数部分列表.

示例:

连分式(5.45) `>>>"5 + $\frac{1}{2 + \frac{1}{4 + \frac{1}{1 + \frac{1}{1}}}}$ "`

连分式(5.45, 3) `>>>"5 + $\frac{1}{2 + \frac{1}{4 + \dots}}$ "`

连分式(5.45, true) `>>>"[5; 2, 4, 1, 1]"`

连分式(5.45, 3, true) `>>>"[5; 2, 4, \dots]"`

2.16.8 VerticalText. 竖排文本

指令

竖排文本 (< 文本 >)



说明: 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.

示例:

竖排文本("ABC") `>>`

ABC

指令

竖排文本 ("< 文本 >", < 点 >)



说明:

将文本按照竖排在固定点显示, 文本随着点移动.

2.16.9 ReplaceAll. 替换所有

指令

替换所有 (< 文本 >, < 要匹配的文本 >, < 要替换的文本 >)



说明: 创建一个新文本, 其中把要匹配的文本替换成要替换的文本.

示例:

替换所有("3cos(t)+cos(2y)", "cos", "sin") `>>>"3sin(t)+sin(2y)"`

2.16.10 UnicodeToText. 统一码转换为文本

指令

统一码转换为文本 (< 统一字符编码整数列表 >)



说明: 转换整数统一码数值到文本.

部分统一码和文本的对照表见 2.7.

示例:

```
统一码转换为文本({104,101,108,108,111}) >>>"hello"
```

2.16.11 UnicodeToLetter. 统一码转换为字母

指令

统一码转换为字母 (< 整数 >)



说明: 把在绘图区中以文本格式显示的整数统一码转换为字母.

Tips: 可结合序列产生连续的字母列表.

```
序列(统一码转换为字母(65+i),i,0,9) >>>{"A", "B", "C", "D", "E", "F", "G", "H", "I", "J"}
```

可以结合字母转换为统一码和序列指令,产生连续字母列表.

```
序列(统一码转换为字母(字母转换为统一码("d") + i), i, 0, 6) >>>{"d", "e", "f", "g", "h", "i", "j"}
```

2.16.12 Text. 文本

指令

文本 (< 对象 >, < 点 >, < 是否替换变量? true|false>, < LaTeX 公式布尔值 >, < 水平对齐 -1|0|1>, < 垂直对齐 -1|0|1>)



说明: 第一个参数是创建对象的文本, 如果不设定第二个参数则默认附着在原点, 第二个参数设定文本的附着点, 第三个参数选择是否替换变量, 如果不设定, 则默认为 true, 第 4 个参数选择是否应用, 如果不设定则默认为 false, 水平对齐和垂直对齐中, -1 表示居左或居下, 0 表示居中, 1 表示居右或居上.

2.16.13 TextToUnicode. 文本转换为统一码

指令

文本转换为统一码 (< 文本 >)



说明: 将文本转换为统一码数字列表, 每个字符一个数字.

示例:

```
文本转换为统一码("Some text") >>>{83,111,109,101,32,116,101,120,116}
```

Tips: 提取数字 a 各位上的数字的方法:

```
a=12357901
文本转换为统一码(a+"")-48>>>{1,2,3,5,7,9,0,1}
```

说明 $a+''''$ 表示把 a 转为文本,0-9 分别对应统一码 48-57, 因此文本转换为统一码 ($a+''''$) 得到的列表再减去 48 即可得到各位上的数字.

2.16.14 Ordinal. 序数

指令

序数 (< 自然数 >)

说明: 把自然数变为文本, 如果输入的不是自然数, 则先对其取整. 序数 (n) 等同于 $n+''''$.

示例:

```
序数(5) >>>"5"
```

2.16.15 RotateText. 旋转文本

指令

旋转文本 (< 文本 >, < 角度 | 弧度 >)

说明: 创建一个新的 LaTeX 文本, 按给定的角度旋转.



注意:

1. 文本需要加双引号
2. 文本绕着文本所在文本框的左上角 (角点 4) 旋转, 并放在坐标原点
3. 默认使用弧度制, 如果要用角度, 则需要添加 $^{\circ}$.

示例:

```
旋转文本("a = 5", 45^{\circ})
```

```
>>>
```

如果要把旋转后的文本固定在某点上, 则需要结合文本指令使用.

2.16.16 LetterToUnicode. 字母转换为统一码

指令

字符转换为统一码 (< 字母 >)

说明: 把字符 (文本形式) 转换为统一码, 部分统一码和字符的对应关系见表 2.7

示例:

```
字母转换为统一码("a") >>>97
```

```
字母转换为统一码("0") >>>48
```

数字	48	49	50	51	52	53	54	55	56	57
字符	0	1	2	3	4	5	6	7	8	9
数字	58	59	60	61	62	63	64	65	66	67
字符	:	;	<	=	>	?	@	A	B	C
数字	68	69	70	71	72	73	74	75	76	77
字符	D	E	F	G	H	I	J	K	L	M
数字	78	79	80	81	82	83	84	85	86	87
字符	N	O	P	Q	R	S	T	U	V	W
数字	88	89	90	91	92	93	94	95	96	97
字符	X	Y	Z	[\]	^	_	'	a
数字	98	99	100	101	102	103	104	105	106	107
字符	b	c	d	e	f	g	h	i	j	k
数字	108	109	110	111	112	113	114	115	116	117
字符	l	m	n	o	p	q	r	s	t	u
数字	118	119	120	121	122	123	124	125	126	127
字符	v	w	x	y	z					

表 2.7

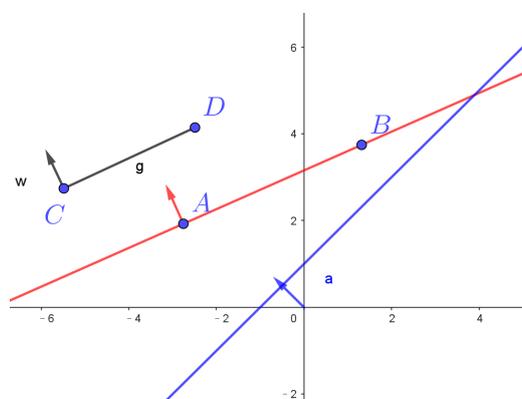


图 2.49

2.17 向量与矩阵

2.17.1 UnitOrthogonalVector. 单位法向量

指令

单位法向量 (< 直线 | 线段 | 向量 | 平面 >)

说明: 绘制出单位法向量, 对于两点确定的直线和线段和向量, 单位法向量的起点在第一个点, 对于由直线方程确定的直线以及向量的法向量, 起点在原点, 如图2.49

2.17.2 Identity. 单位矩阵

指令

单位矩阵 (< 数值 >)

说明: 返回给定阶数的单位矩阵. 若矩阵 A 为 n 阶方阵, 则 A^0 等同于单位矩阵 (n).

示例:

```
单位矩阵(3) >>>{{1,0,0},{0,1,0},{0,0,1}}
```

2.17.3 UnitVector. 单位向量

指令

单位向量 (< 向量 | 直线 | 线段 >)

说明: 返回与给定向量(线段, 直线)方向相同长度为1的向量, 点被视作向量使用. 单位向量起点的规定参见UnitOrthogonalVector. 单位法向量中的说明. $A + a$ 单位向量 (u) 表示 A 沿着指定方向前进 a 个单位, 故常用于动点的作图.

2.17.4 OrthogonalVector. 法向量

指令

法向量 (< 直线 | 线段 | 向量 | 平面 >)

说明: 若直线的方程为 $Ax + By + C = 0$, 则返回的法向量为 (A, B) , 若直线 | 线段 | 向量由两点 $(x_1, y_1), (x_2, y_2)$ 确定, 则法向量为 $(y_1 - y_2, x_2 - x_1)$, 向量的起点为第一个点. 若平面的方程为 $Ax + By + Cz + D = 0$, 则法向量为 (A, B, C) , 起点为原点. 若向量式以坐标 (a, b) 定义, 则法向量为 $(-b, a)$.

2.17.5 ReducedRowEchelonForm. 简化行梯阵式

指令

简化行梯阵式 (< 矩阵 >)

说明: 返回矩阵的简约行梯阵形式

2.17.6 MatrixRank. 矩阵的秩

指令

矩阵的秩 (< 矩阵 >)

说明: 返回给定矩阵的秩

2.17.7 Invert. 逆反

指令

逆反 (矩阵 | 函数)

说明: 返回逆矩阵或反函数. 由逆反产生的反函数, 会给出反函数的解析式, 注意和2.6.9 令的区别.

2.17.8 Dimension. 维度

指令

维度 (< 点 | 向量 | 矩阵 >)

说明: 给出一个向量或是一个矩阵的维度.

示例:

维度({{1, 2}, {3, 4}, {5, 6}}) >>>{3,2}, 即3行2列.
 维度((1,2,3))>>>3

2.17.9 Vector. 向量

指令

向量 (< 终点 (原点为起点)>)

说明: 返回给定点的位置向量.

指令

向量 (< 起点 >, < 终点 >)

说明: 创建一个从起点到终点的向量.

2.17.10 Determinant. 行列式

指令

行列式 (< 矩阵 >)

说明: 给出矩阵的行列式的值.

2.17.11 ApplyMatrix. 应用矩阵

指令

应用矩阵 (< 矩阵 >, < 对象 >)

说明: 设矩阵为 M , 对象为点 P , 则点 P 被当做列矩阵参与运算.

若 M 是 2×2 矩阵, P 是二维点, 则 P 被视为

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

则返回的对象为 MP , 即矩阵之积.

示例:

应用矩阵({{1, 0}, {1, 1}}, (1, 2)) >>>(2,3)

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

若 M 为 3×3 矩阵, P 为三维点, 则返回对象为 MP . 若 M 为 3×3 矩阵, P 为二维点, 则先把 P 映射为 $\begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix}$, 得到

列矩阵 $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$, 则返回的点为 $(\frac{x}{z}, \frac{y}{z})$. 若 M 为 2×2 矩阵, P 为三维点, 则先把 M 矩阵映射为 $\begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{pmatrix}$ 矩阵

后再计算 MP. 对于其他对象, 应用矩阵即对对象上所有的点进行如上规定的映射, 可以应用矩阵的对象有点, 线, 圆锥曲线, 多边形, 图片, 参数曲线等.

Tips: 常用二维旋转矩阵:

$$\begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$$

绕 x 轴旋转矩阵:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}$$

绕 y 轴旋转矩阵::

$$\begin{pmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{pmatrix}$$

绕 z 轴旋转矩阵::

$$\begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

示例:

应用矩阵 $\{\{\cos(30^\circ), -\sin(30^\circ)\}, \{\sin(30^\circ), \cos(30^\circ)\}\}$, f $\gg f'$ (将 f 绕原点逆时针旋转 30°)
 # 先对 f 的两个顶点应用旋转矩阵, 再连接 2 个顶点, 得到 f'

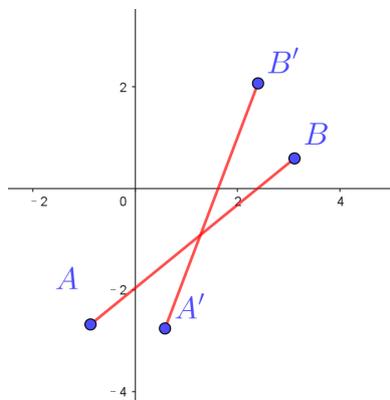


图 2.50

2.17.12 Transpose. 转置

指令

转置 (< 矩阵 >)



说明: 转置矩阵

2.18 优化指令

2.18.1 Maximize. 最大值点

指令

最大值点 (< 因变数 >, < 滑动条 >)



说明: 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.

指令

最大值点 (< 因变数 >, < 界点 >)



说明:

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.

Tips: 最大值点常用于常规作图或尺规作图无法精确作图的情形，用于搜寻满足条件的点，一般一个课件中最多使用一次，如果使用多次，软件会不可避免的出现卡顿. 最小值点的指令也是类似的.

2.18.2 Minimize. 最小值点

指令

最小值点 (< 因变量 >, < 滑动条 >)



说明: 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.

指令

最小值点 (< 因变数 >, < 界点 >)



说明:

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.

参见 [Maximize. 最大值点](#)

2.19 圆锥曲线

 **注意:** 本手册所指的圆锥曲线不作特殊说明，都包含圆.

2.19.1 Excentricity. 半焦距

指令

半焦距 (< 圆锥曲线 >)



说明: 返回圆锥曲线的半焦距 c

2.19.2 Semicircle. 半圆

指令

半圆 (< 点 1>, < 点 2>)

说明: 在两点之间的线段上方 (指法向量方向) 创建一个半圆, 并在代数区视图中显示其长度.

2.19.3 SecondAxisLength. 副半轴长

指令

副半轴长 (< 圆锥曲线 >)

说明: 返回椭圆的短半轴长和双曲线的虚半轴长 b 或圆的半径长

2.19.4 SecondAxis. 副轴

指令

副轴 (< 圆锥曲线 >)

说明: 返回圆锥曲线的副轴方程.

示例:

副轴 ($x^2+y^2-2x=0$)	$\ggg x=1$, 即与 x 轴垂直的直径所在直线
副轴 ($x^2-y^2/4=1$)	$\ggg x=0$, 即双曲线的虚轴所在直线
副轴 ($x^2/4+y^2=1$)	$\ggg x=0$, 即椭圆的短轴所在的直线

2.19.5 Diameter. 共轭直径

指令

共轭直径 (< 向量 | 直线 >, < 圆锥曲线 >)

说明: 返回平行于给定直线 (向量) 的直径的共轭直径所在的直线.

释义: 经过圆锥曲线的中心的弦被称为直径, 所有平行于某直径的弦的中点的连线, 称为共轭直径. 主轴和副轴互为共轭直径, 在斜率存在的直线中, 共轭直径的斜率乘积为 $e^2 - 1$, 圆的离心率 e 视作 0.

示例:

共轭直径 ($-4x+5y=-2$, $x^2+4y^2+2x-8y+1=0$) $\ggg 5x+16y=11$

说明: 椭圆的中心为 $(-1, 1)$, $e^2 - 1 = -\frac{1}{4}$,
 $-4x + 5y = -2$ 的斜率为 $\frac{4}{5}$, 因此共轭直径的斜率为 $-\frac{5}{16}$,
 因此直线方程为: $y - 1 = -\frac{5}{16}(x + 1)$, 化简得: $5x + 16y = 11$.

2.19.6 Polar. 极线

指令

极线 (< 点 >, < 圆锥曲线 >)

说明: 创建给定点关于圆锥曲线的极线.

释义: 如果圆锥曲线切于 AB 两点的切线相交于 P 点, 那么 P 点称为直线 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

指令

焦参数 (< 抛物线 >)



说明: 返回抛物线的焦准距 p

2.19.8 Focus. 焦点

指令

焦点 (< 圆锥曲线 >)



说明: 返回圆锥曲线的焦点 (圆的 2 个焦点为圆心)

2.19.9 Eccentricity. 离心率

指令

离心率 (圆锥曲线)



说明: 返回圆锥曲线的离心率

2.19.10 Incircle. 内切圆

指令

内切圆 (< 点 1>, < 点 2>, < 点 3>)



说明: 返回三个点为顶点的三角形的内切圆

2.19.11 Parabola. 抛物线

指令

抛物线 (< 焦点 >, < 准线 >)



说明: 返回一个给定焦点和准线的抛物线.

2.19.12 Hyperbola. 双曲线

指令

双曲线 (< 焦点 1>, < 焦点 1>, < 主半轴长 | 线段 >)



说明: 创建给定两个焦点和主半轴 (线段) 长度的双曲线. 如果 $2a > 2c$, 则返回的为椭圆.

指令

双曲线 (< 焦点 1>, < 焦点 1>, < 双曲线上一点 >)



说明: 创建给定点为焦点且经过给定点的双曲线.

2.19.13 Ellipse. 椭圆

指令

椭圆 (< 焦点 1>,< 焦点 1>,< 主半轴长 | 线段 >)

说明: 创建给定两个焦点和主半轴 (线段) 长度的椭圆. 如果 $2a < 2c$, 则返回的为双曲线.

指令

椭圆 (< 焦点 1>,< 焦点 1>,< 椭圆上一点 >)

说明:

创建给定点为焦点且经过给定点的椭圆.

2.19.14 Circle. 圆周

指令

圆周 (< 圆心 >,< 半径长度 | 线段 >)

说明: 返回给定圆心, 且以给定半径长 (线段长) 为半径的圆

指令

圆周 (< 圆心 >,< 圆上一点 >)

说明: 得出一个给定圆心过一个给定点的圆.

指令

圆周 (< 点 >,< 点 >,< 点 >)

说明: 创建经过 3 点的圆 (即三角形的外接圆)

3D 区指令

圆周 (< 直线 >,< 点 >)

说明: 创建直线为轴且过点的圆 (也适用于 3D 区)

3D 区指令

圆周 (< 圆心 >,< 半径 >,< 方向 >)

说明: 创建一个指定圆心、半径和轴平行于一条直线、向量或平面的圆.

3D 区指令

圆周 (< 圆心 >,< 圆上一点 >,< 方向 >)

说明: 创建一个指定圆心过点且轴平行于方向的圆.

2.19.15 Conic. 圆锥曲线

指令

圆锥曲线 (< 点 1>, < 点 2>, < 点 3>, < 点 4>, < 点 5>)

说明: 返回一个过给定五点的圆锥曲线.

2.19.16 Center. 中心

指令

中心 (< 圆锥曲线 | 球面 | 二次曲面 >)

说明: 返回圆锥曲线, 球面, 二次曲面的中心 (例如圆锥面), 注意这里的圆锥面指的是用二次方程写出来的曲面, 而非圆锥的面.

示例:

中心($x^2 + 4y^2 + 2x - 8y + 1 = 0$) $\ggg(-1, 1)$

2.19.17 Axes. 轴线

指令

轴线 (< 圆锥曲 | 二次曲面 >)

说明: 返回圆锥曲线或二次曲面的轴线

 **注意:** 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. 需要激活 3D 绘图区, 显示此指令的效果.

2.19.18 FirstAxisLength. 主半轴长

指令

主半轴长 (< 圆锥曲线 >)

说明: 返回圆锥曲线的主半轴长 (主轴一半).

参见 [SecondAxisLength](#). 副半轴长

2.19.19 FirstAxis. 主轴

指令

主轴 (< 主轴 >)

说明: 返回圆锥曲线的主轴直线方程.

参见 [SecondAxis](#). 副轴

2.19.20 Directrix. 准线

指令

准线 (< 圆锥曲线 >)



说明: 返回圆锥曲线 (除圆外) 的准线.

第3章 附录

3.1 中英文查询

网址是:<https://www.geogebra.org/m/fzcxua5>

3.2 部分通用的英文名

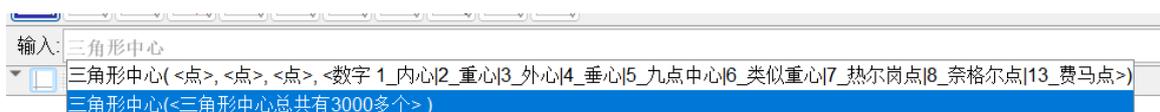
SD=stdevp	PerpendicularLine=OrthogonalLine
SampleSD=stdev	PerpendicularBisector =LineBisector
FormulaText=Latex	Dilate=Enlarge
Q1=Quartile1	InputBox=Textfield
Q3=Quartile3	Isdefined=Defined
InflectionPoint=TurningPoint	CorrelationCoefficient=PMCC
Binomial=BinomialCoefficient	UnitPerpendicularVector=UnitOrthogonalVector
Curve=CurveCartesian	PerpendicularVector=OrthogonalVector
SemiMinorAxisLength =SecondAxisLength	SemiMajorAxisLength= FirstAxisLength
MajorAxis =FirstAxis	ConjugateDiameter=Diameter
MinorAxis =SecondAxis	

3.3 修改指令名和指令提示



```
TriangleCenter.Syntax=[ <点>, <点>, <点>, <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_点> ]\n[三角形中心总共有3000多个]
```

图 3.1



输入: 三角形中心

三角形中心(<点>, <点>, <点>, <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>)

三角形中心(<三角形中心总共有3000多个>)

图 3.2

GeoGebra 中的中文指令和指令提示存储在文件 `geogebra_properties.jar` 中, 修改的方法如下:

1. 解压缩 jar 文件, 笔者使用的是 7zip, 得到 2 个文件夹: org 和 META-INF, 我们要用的是 org 文件夹, 另一个可以丢弃, 建议大家先备份源文件.
2. 打开 org 文件夹, 并一直打开子文件夹, 直到进入 properties 文件夹, 找到 `command_zh_CN.properties` 文件, 用记事本或其他软件打开, 笔者使用的是 vscode.
3. 接下来就可以开始修改:
 - (1) 可以把没翻译的英文翻译, 以 `nCr` 指令为例, 可以看到其没有对应的中文指令, 我们可以在它所在的行(我这里是 1094 行) 的上面加上一句: `nCr= 组合数`;
 - (2) 修改中文名, 以周界长为例, 我们找到第 116 行, 把 `Circumference= 周界长` 修改为 `Circumference= 周界`;
 - (3) 修改或增加提示, 比如三角中心指令, 在第 997 行, 将 `TriangleCenter.Syntax=[<点>, <点>, <点>, <数`

字 >], 修改为 `TriangleCenter.Syntax=[< 点 >, < 点 >, < 点 >, < 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]`, 也可以加上 `\n` 增加一行提示, 如图 3.1 .

4. 修改完, 关闭该文件, 最后将整个 org 文件夹压缩为 zip 格式, 并修改为 `geogebra_properties.jar`, 存放在原来的文件夹, 并覆盖该文件, 这样指令就修改好了. 效果如图 3.2.

 **注意:** 由于 GeoGebra 经常更新, 上面修改的文件会被更新所覆盖, 因此建议大家及时备份. 此外, 该改动只在本机有用, 制作出来的课件分享给他人, 仍会显示官方默认的指令, 因此不会影响使用. 此外, 我们不能通过这种方式凭空增加指令。