Grafica tridimensionale

Oggetti grafici

Graphics3D[primitives, options] represents a three-dimensional graphical image. Default options: AmbientLight -> GrayLevel[0.], AspectRatio -> Automatic, Axes -> False, AxesEdge -> Automatic, AxesLabel -> None, AxesStyle -> Automatic, Background -> Automatic, Boxed -> True, BoxRatios -> Automatic, BoxStyle -> Automatic, ColorOutput -> Automatic, DefaultColor -> Automatic, Epilog -> {}, FaceGrids -> None, Lighting -> True, LightSources -> {{{1., 0., 1.}, RGBColor[1, 0, 0]}, {{1., 1., 1.}, RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}}, PlotLabel -> None, PlotRange -> Automatic, PlotRegion -> Automatic, Plot3Matrix -> Automatic, PolygonIntersections -> True, Prolog -> {}, RenderAll -> True, Shading -> True, SphericalRegion -> False, Ticks -> Automatic, ViewCenter -> Automatic, ViewPoint -> {1.3, -2.4, 2.}, ViewVertical -> {0., 0., 1.}, DefaultFont :> $DefaultFont, DisplayFunction :> $DisplayFunction.

SurfaceGraphics[array] is a representation of a three-dimensional plot of a surface, with heights of each point on a grid specified by values in array.
SurfaceGraphics[array, shades] represents a surface, whose parts are shaded according to the array shades.

Primitive grafiche

Cuboid[{xmin, ymin, zmin}] is a three-dimensional graphics primitive that represents a unit cuboid, oriented parallel to the axes.
Cuboid[{xmin, ymin, zmin}, {xmax, ymax, zmax}] specifies a cuboid by giving the coordinates of opposite corners.

Line[{pt1, pt2, ...}] is a graphics primitive which represents a line joining a sequence of points.

Point[coords] is a graphics primitive that represents a point.

Polygon[{pt1, pt2, ...}] is a graphics primitive that represents a filled polygon.

Text[expr, coords] is a graphics primitive that represents text corresponding to the printed form of expr, centered at the point specified by coords.

Scaled[{x, y, z}] gives the position of a graphical object in terms of coordinates scaled to run from 0 to 1 across the whole plot in each direction.
Scaled[{dx, dy, dz}, {x0, y0, z0}] gives a position obtained by starting at absolute coordinates {x0, y0, z0}, then moving by a scaled offset {dx, dy, dz}.

Direttive grafiche

AbsoluteDashing[{d1, d2, ...}] is a two-dimensional graphics directive which specifies that lines which follow are to be drawn dashed, with successive segments having absolute lengths d1, d2, ... (repeated cyclically). The lengths di are given in points.

AbsolutePointSize[d] is a graphics directive which specifies that points which follow are to be shown if possible as circular regions with absolute radius d. The radius d is given in points.

AbsoluteThickness[d] is a graphics directive which specifies that lines which follow are to be drawn with absolute thickness d. The thickness d is given in points.

CMYKColor[cyan, magenta, yellow, black] is a graphics directive which specifies that graphical objects which follow are to be displayed in the color given.

Dashing[{r1, r2, ...}] is a two-dimensional graphics directive which specifies that lines which follow are to be drawn dashed, with successive segments of lengths r1, r2, ... (repeated cyclically). The ri are given as a fraction of the total width of the graph.

EdgeForm[g] is a three-dimensional graphics directive which specifies that edges of polygons are to be drawn using the graphics directive or list of graphics directives g.

FaceForm[gf, gb] is a three-dimensional graphics directive which specifies that the front faces of polygons are to be drawn with the graphics primitive gf, and the back faces with gb.

GrayLevel[level] is a graphics directive which specifies the gray-level intensity with which graphical objects that follow should be displayed.

Hue[h] is a graphics directive which specifies that graphical objects which follow are to be displayed, if possible, in a color corresponding to hue h.
Hue[h, s, b] specifies colors in terms of hue, saturation and brightness.

RGBColor[red, green, blue] is a graphics directive which specifies that graphical objects which follow are to be displayed, if possible, in the color given.

PointSize[r] is a graphics directive which specifies that points which follow are to be shown if possible as circular regions with radius r. The radius r is given as a fraction of the total width of the graph.

SurfaceColor[dcol] is a three-dimensional graphics directive which specifies that the polygons which follow should act as diffuse reflectors of light with a color given by dcol.
SurfaceColor[dcol, scol] specifies that a specular reflection component should be included, with a color given by scol.
SurfaceColor[dcol, scol, n] specifies that the reflection should occur with specular exponent n.

Thickness[r] is a graphics directive which specifies that lines which follow are to be drawn with a thickness r. The thickness r is given as a fraction of the total width of the graph.

Generazione di oggetti grafici

Plot3D[f, {x, xmin, xmax}, {y, ymin, ymax}] generates a three-dimensional plot of f as a function of x and y.
Plot3D[{f, s}, {x, xmin, xmax}, {y, ymin, ymax}] generates a three-dimensional plot in which the height of the surface is specified by f, and the shading is specified by s.
Default options: AmbientLight -> GrayLevel[0], AspectRatio -> Automatic, Axes -> True, AxesEdge -> Automatic, AxesLabel -> None, AxesStyle -> Automatic, Background -> Automatic, Boxed -> True, BoxRatios -> {1, 1, 0.4}, BoxStyle -> Automatic, ClipFill -> Automatic, ColorFunction -> Automatic, ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic, Epilog -> {}, FaceGrids -> None, HiddenSurface -> True, Lighting -> True, LightSources -> {{{1., 0., 1.}, RGBColor[1, 0, 0]}, {{1., 1., 1.}, RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}}, Mesh -> True, MeshStyle -> Automatic, PlotLabel -> None, PlotPoints -> 15, PlotRange -> Automatic, PlotRegion -> Automatic, Plot3Matrix -> Automatic, Prolog -> {}, Shading -> True, SphericalRegion -> False, Ticks -> Automatic, ViewCenter -> Automatic, ViewPoint -> {1.3, -2.4, 2.}, ViewVertical -> {0., 0., 1.}, DefaultFont :> $DefaultFont, DisplayFunction :> $DisplayFunction.

ParametricPlot3D[{fx, fy, fz}, {t, tmin, tmax}] produces a three-dimensional space curve parameterized by a variable t which runs from tmin to tmax.
ParametricPlot3D[{fx, fy, fz}, {t, tmin, tmax}, {u, umin, umax}] produces a three-dimensional surface parametrized by t and u.
ParametricPlot3D[{fx, fy, fz, s}, ...] shades the plot according to the color specification s.
ParametricPlot3D[{{fx, fy, fz}, {gx, gy, gz}, ...}, ...] plots several objects together.
Default options: AmbientLight -> GrayLevel[0.], AspectRatio -> Automatic, Axes -> True, AxesEdge -> Automatic, AxesLabel -> None, AxesStyle -> Automatic, Background -> Automatic, Boxed -> True, BoxRatios -> Automatic, BoxStyle -> Automatic, ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic, Epilog -> {}, FaceGrids -> None, Lighting -> True, LightSources -> {{{1., 0., 1.}, RGBColor[1, 0, 0]}, {{1., 1., 1.}, RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}}, PlotLabel -> None, PlotPoints -> Automatic, PlotRange -> Automatic, PlotRegion -> Automatic, Plot3Matrix -> Automatic, PolygonIntersections -> True, Prolog -> {}, RenderAll -> True, Shading -> True, SphericalRegion -> False, Ticks -> Automatic, ViewCenter -> Automatic, ViewPoint -> {1.3, -2.4, 2.}, ViewVertical -> {0., 0., 1.}, DefaultFont :> $DefaultFont, DisplayFunction :> $DisplayFunction.

ListPlot3D[array] generates a three-dimensional plot of a surface representing an array of height values.

ListPlot3D[array, shades] generates a plot with each element of the surface shaded according to the specification in shades.

Visualizzazione di oggetti grafici

Show[graphics, options] displays two- and three-dimensional graphics, using the options specified.
Show[g1, g2, ...] shows several plots combined.

Opzioni

AmbientLight is an option to Graphics3D and related functions that gives the level of simulated ambient illumination in a three-dimensional picture. The setting must be a GrayLevel, Hue, or RGBColor directive.

AspectRatio is an option for Show and related functions. With AspectRatio -> Automatic, the ratio of height to width of the plot is determined from the actual coordinate values in the plot. AspectRatio -> r makes the ratio equal to r.

Axes is an option for graphics functions. With Axes -> True, all axes are drawn. Axes -> False draws no axes. Axes -> {False, False, True} draws a z axis but no x or y axis.

AxesEdge is an option for 3D graphics functions. AxesEdge -> {{xdir, ydir}, {xdir, zdir}, {xdir, ydir}} specifies on which three edges of the bounding box axes are drawn. The idir must be either +1 or -1, and specify whether axes are drawn on the edge of the box with a larger or smaller value of coordinate i, respectively. Any pair {idir, jdir} can be replaced by Automatic or None. The default setting is AxesEdge -> Automatic.

AxesLabel is an option for graphics functions. With AxesLabel -> None, no labels are drawn on the axes in the plot. AxesLabel -> label specifies a label for the z axis. AxesLabel -> {xlabel, ylabel, zlabel} specifies labels for different axes.

AxesOrigin is an option for two-dimensional graphics functions. AxesOrigin -> {x, y} specifies that the axes drawn should cross at the point {x, y}. AxesOrigin -> Automatic uses an internal algorithm to determine where the axes should cross.

AxesStyle is an option for graphics functions. AxesStyle -> style specifies that all axes are to be rendered with the specified graphics directive, or list of graphics directives. AxesStyle -> {{xstyle}, {ystyle}, {zstyle}} specifies that axes should use graphics directives xstyle, zstyle.

Background is an option for graphics functions which specifies the background color to use. A setting must be a CMYKColor, GrayLevel, Hue or RGBColor directive. The default setting Background -> Automatic produces a white background on most output devices.

Boxed is an option for Graphics3D. Boxed -> True draws the edges of the bounding box in a three-dimensional picture. With Boxed -> False, no bounding box is drawn.

BoxRatios is an option for Graphics3D and SurfaceGraphics. BoxRatios -> {rx, ry, rz} gives the ratios of side lengths for the bounding box of the three-dimensional picture. BoxRatios -> Automatic determines the ratios using the range of actual coordinate values in the plot.

BoxStyle is an option for three-dimensional graphics functions which specifies how the bounding box should be rendered. BoxStyle can be set to a list of graphics directives such as Dashing, Thickness, GrayLevel and RGBColor. BoxStyle -> Automatic uses a default style.

ColorOutput is an option for graphics functions which specifies the type of color output to produce. ColorOutput -> Automatic uses whatever color directives are given. ColorOutput -> CMYKColor converts to CMYKColor. ColorOutput -> GrayLevel converts to GrayLevel. ColorOutput -> RGBColor converts to RGBColor. ColorOutput -> f converts using the function f.

DefaultColor is an option for graphics functions which specifies the default color to use for lines, points, etc. The setting must be a CMYKColor, GrayLevel, Hue or RGBColor directive. The default setting DefaultColor -> Automatic gives a color complementary to the background specified.

DefaultFont is an option for graphics functions which specifies the default font to use for text. DefaultFont -> {"font", size} specifies the name and size of the font to use. The default setting is DefaultFont :> $DefaultFont.

DisplayFunction is an option for graphics functions that specifies the function to apply to graphics primitives in order to display them. The default setting is $DisplayFunction. A typical setting is DisplayFunction -> (Display[channel, #]&). DisplayFunction -> Identity causes the objects to be returned, but no display to be generated.

Epilog is an option for graphics functions which gives a list of graphics primitives to be rendered after the main part of the graphics is rendered.

FaceGrids is an option for three-dimensional graphics functions. With FaceGrids -> All, grid lines are drawn on all the faces of the bounding box. FaceGrids -> None draws no grid lines. FaceGrids -> {face1, face2, ... } draws grid lines on the specified faces. FaceGrids -> {{face1, {xgrid, ygrid}}, ... } specifies an arrangement of the grid lines.

Lighting is an option to Graphics3D and related functions. With Lighting -> True, simulated illumination is used; with Lighting -> False, it is not.

LightSources is an option to Graphics3D and related functions that specifies the properties of point light sources for simulated illumination. The basic form is LightSources -> {{direction1, color1}, {direction2, color2}, ... }. The direction is specified as {x, y, z}. The color can be specified by GrayLevel, Hue or RGBColor.

PlotLabel is an option for graphics functions that specifies an overall label for a plot. With PlotLabel -> None, no label is given. PlotLabel -> label specifies a label.

PlotRange is an option for graphics functions. With PlotRange -> All, all points are included in the plot. With PlotRange -> Automatic, outlying points are dropped. PlotRange -> {min, max} specifies explicit limits for z. PlotRange -> {{xmin, xmax}, {ymin, ymax}, {zmin, zmax}} gives explicit limits for each coordinate.

PlotRegion is an option for graphics functions. PlotRegion -> {{sxmin, sxmax}, {symin, symax}} specifies the region in scaled coordinates that the plot should fill in the final display area. With PlotRegion -> Automatic, the plot fills the final display area.

PolygonIntersections is an option for Graphics3D and ParametricPlot3D. With PolygonIntersections -> True, intersecting polygons are left unchanged. With PolygonIntersections -> False, Graphics3D objects are modified by breaking polygons into smaller pieces which do not intersect each other.

Prolog is an option for graphics functions which gives a list of graphics primitives to be rendered before the main part of the graphics is rendered.

RenderAll is an option to Graphics3D. When RenderAll -> False, PostScript will be generated only for those polygons or parts of polygons which are visible in the final picture. If RenderAll -> True, PostScript is generated for all polygons.

RotateLabel is an option for two-dimensional graphics functions. With RotateLabel -> True, labels on vertical frame axes are rotated to be vertical. With RotateLabel -> False, they are not.

Shading is an option for SurfaceGraphics. With Shading -> True, surfaces are shaded. With Shading -> False, they are not.

SphericalRegion is an option for three-dimensional graphics functions. With SphericalRegion -> True, the final image is scaled so that a sphere drawn around the three-dimensional bounding box will fit in the display area specified. SphericalRegion -> False scales three-dimensional images to be as large as possible, given the display area specified.

Ticks is an option for graphics functions. With Ticks -> None, no tick marks are drawn on the axes. With Ticks -> Automatic, tick marks are placed automatically. Ticks -> {xticks, yticks, zticks} specifies tick mark options separately for each axis.

ViewCenter is an option for Graphics3D and SurfaceGraphics. With ViewCenter -> Automatic, the whole bounding box is centered in the final image area. With ViewCenter -> {x, y, z}, the point in the three-dimensional bounding box with scaled coordinates x, y, z is placed at the center of the final display area.

ViewPoint is an option for Graphics3D and SurfaceGraphics which gives the point in space from which the objects plotted are to be viewed. ViewPoint -> {x, y, z} gives the position of the view point relative to the center of the three-dimensional box that contains the object being plotted.

ViewVertical is an option for Graphics3D and SurfaceGraphics. ViewVertical -> {x, y, z} specifies the direction in scaled coordinates which should be vertical in the final image.