

Grafica bidimensionale

Oggetti grafici

Graphics[primitives, options] represents a two-dimensional graphical image.
Default options: AspectRatio -> GoldenRatio⁽⁻¹⁾, Axes -> False, AxesLabel -> None, AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic, ColorOutput -> Automatic, DefaultColor -> Automatic, Epilog -> {}, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic, FrameTicks -> Automatic, GridLines -> None, PlotLabel -> None, PlotRange -> Automatic, PlotRegion -> Automatic, Prolog -> {}, RotateLabel -> True, Ticks -> Automatic, DefaultFont := \$DefaultFont, DisplayFunction := \$DisplayFunction.

ContourGraphics[array] is a representation of a contour plot.

DensityGraphics[array] is a representation of a density plot.

Primitive grafiche

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}] 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}, {x0, y0}] gives a position obtained by starting at absolute coordinates {x0, y0}, then moving by a scaled offset {dx, dy}.

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.

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.

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

Plot[*f*, {*x*, *xmin*, *xmax*}] generates a plot of *f* as a function of *x* from *xmin* to *xmax*.

Plot[{*f*₁, *f*₂, ...}, {*x*, *xmin*, *xmax*}] plots several functions *f*_{*i*}.

Default options: AspectRatio -> GoldenRatio⁽⁻¹⁾, Axes -> Automatic, AxesLabel -> None, AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic, ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic, Epilog -> {}, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic, FrameTicks -> Automatic, GridLines -> None, MaxBend -> 10., PlotDivision -> 20., PlotLabel -> None, PlotPoints -> 25, PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic, Prolog -> {}, RotateLabel -> True, Ticks -> Automatic, DefaultFont :> \$DefaultFont, DisplayFunction :> \$DisplayFunction.

ParametricPlot[{*fx*, *fy*}, {*t*, *tmin*, *tmax*}] produces a parametric plot with *x* and *y* coordinates *fx* and *fy* generated as a function of *t*.

ParametricPlot[{{*fx*, *fy*}, {*gx*, *gy*}, ...}, {*t*, *tmin*, *tmax*}] plots several parametric curves.

Default options: AspectRatio -> GoldenRatio⁽⁻¹⁾, Axes -> Automatic, AxesLabel -> None, AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic, ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic, Epilog -> {}, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic, FrameTicks -> Automatic, GridLines -> None, MaxBend -> 10., PlotDivision -> 20., PlotLabel -> None, PlotPoints -> 25, PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic, Prolog -> {}, RotateLabel -> True, Ticks -> Automatic, DefaultFont :> \$DefaultFont, DisplayFunction :> \$DisplayFunction.

ContourPlot[*f*, {*x*, *xmin*, *xmax*}, {*y*, *ymin*, *ymax*}] generates a contour plot of *f* as a function of *x* and *y*.

DensityPlot[*f*, {*x*, *xmin*, *xmax*}, {*y*, *ymin*, *ymax*}] makes a density plot of *f* as a function of *x* and *y*.

ListContourPlot[*array*] generates a contour plot from an array of height values.

ListDensityPlot[*array*] generates a density plot from an array of height values.

ListPlot[{*y*₁, *y*₂, ...}] plots a list of values. The *x* coordinates for each point are taken to be 1, 2, ...

ListPlot[{{*x*₁, *y*₁}, {*x*₂, *y*₂}, ...}] plots a list of values with specified *x* and *y* coordinates.

Visualizzazione di oggetti grafici

Show[*graphics*, *options*] displays two-dimensional graphics, using the options specified.

Show[*g*₁, *g*₂, ...] shows several plots combined.

Opzioni grafiche

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, True} draws a *y* axis but no *x* axis.

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 *y* axis. AxesLabel -> {*xlabel*, *ylabel*} 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*}} specifies that axes should use graphics directives *xstyle* and *ystyle*.

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.

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.

Frame is an option for two-dimensional graphics functions. With Frame -> True, a frame is drawn around the plot. With Frame -> False, no frame is drawn. Frame -> {True, False, True, False} draws just the top and bottom of the frame.

FrameLabel is an option for two-dimensional graphics functions that specifies labels to be placed on the edges of a frame around a plot. FrameLabel -> {xlabel, ylabel} specifies labels for the bottom and left-hand edges of the frame. FrameLabel -> {xlabel, ylabel, xlabel, ylabel} specifies labels for each of the edges of the frame. With FrameLabel -> None, no labels are given.

FrameStyle is an option for two-dimensional graphics functions. With FrameStyle -> Automatic, the edges of a frame are rendered using a default style. With FrameStyle -> style, the specified graphics directive, or list of graphics directives is used. FrameStyle -> {{xmstyle}, {ymstyle}} specifies different styles for different edges of the frame.

FrameTicks is an option for two-dimensional graphics functions. With FrameTicks -> Automatic, tick marks for the edges of a frame are chosen automatically. With FrameTicks -> None, no tick marks are drawn. FrameTicks -> {xmticks, ymticks} specifies tick mark options separately for each edge.

GridLines is an option for two-dimensional graphics functions. With GridLines -> None, no grid lines are drawn. GridLines -> Automatic places the grid lines automatically. GridLines -> {xgrid, ygrid} specifies the position of grid lines in each direction.

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 y. PlotRange -> {{xmin, xmax}, {ymin, ymax}} 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.

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.

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.

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} specifies tick mark options separately for each axis.