本页对应的英文页面已更新，但尚未翻译。 若要查看最新内容，请点击此处访问英文页面。

Axes 属性

坐标区的外观和行为

Axes 属性可控制 Axes 对象的外观和行为。通过更改属性值，您可以修改坐标区的特定方面。

从 R2014b 开始，您可以使用圆点表示法查询和设置属性。

```
ax = gca;
c = ax.Color;
ax.Color = 'blue';
```

如果您使用的是早期版本，请改用 get 和 set 函数。

字体

全部折叠

FontName - 字体名称

支持的字体名称 | 'FixedWidth'

字体名称，指定为支持的字体名称或 'FixedWidth'。要正确显示和打印文本，您必须选择系统支持的字体。默认字体取决于您的操作系统和区域设置。

要使用在任何区域设置中都有较好显示效果的等宽字体，请使用 'FixedWidth'。等宽字体依赖于根 FixedWidthFontName 属性。设置根 FixedWidthFontName 属性会导致立即更新显示方式以使用新字体。

FontWeight - 字符粗细

'normal' （默认） | 'bold'

字符粗细，指定为 'normal' 或 'bold'。

MATLAB® 使用 FontWeight 属性从系统提供的字体中选择一种字体。并非所有字体都有加粗字体。因此，指定加粗字体仍可能得到普通字体。

FontSize - 字体大小

数值标量

字体大小，指定为数值标量。字体大小会影响标题、轴标签和刻度标签。还会影响与坐标区关联的任何图例或颜色栏。默认字体大小取决于具体操作系统和区域设置。默认情况下，字体大小以磅为单位进行测量。要更改单位，请设置 FontUnits 属性。

MATLAB 会自动将某些文本缩放为坐标区字体大小的百分比。

- 标题和轴标签 - 默认情况下为坐标区字体大小的 110%。要控制缩放比例，请使用 TitleFontSizeMultiplier 和 LabelFontSizeMultiplier 属性。
- 图例和颜色栏 - 默认情况下为坐标区字体大小的 90%。要指定不同的字体大小，请设置 Legend 或 Colorbar 对象的 FontSize 属性。

示例： ax.FontSize = 12

FontSizeMode - 字体大小的选择模式

'auto' （默认） | 'manual'

字体大小的选择模式，指定为下列值之一：

- 'auto' - 由 MATLAB 指定的字体大小。如果将坐标区的大小调整为小于默认大小，则字体大小可能会缩小以改善可读性和布局。
- 'manual' - 手动指定的字体大小。不随坐标区大小改变而缩放字体大小。要指定字体大小，请设置 FontSize 属性。

FontAngle - 字符倾斜

'normal' （默认） | 'italic'

字符倾斜，指定为 'normal' 或 'italic'。

并非所有字体都有两种字体样式。因此，斜体可能看上去和常规字体一样。

`LabelFontSizeMultiplier` - 标签字体大小的缩放因子

`1.1` （默认） | 大于 0 的数值

标签字体大小的缩放因子，指定为大于 0 的数值。此缩放因子应用于 `FontSize` 属性的值，以确定 *x* 轴、*y* 轴和 *z* 轴标签的字体大小。

示例： `ax.LabelFontSizeMultiplier = 1.5`

`TitleFontSizeMultiplier` - 标题字体大小的缩放因子

`1.1` （默认） | 大于 0 的数值

标题字体大小的缩放因子，指定为大于 0 的数值。此缩放因子应用于 `FontSize` 属性的值，以确定标题的字体大小。

示例： `ax.TitleFontSizeMultiplier = 1.75`

`TitleFontWeight` - 标题字符的粗细

`'bold'` （默认） | `'normal'`

标题字符的粗细，指定为下列值之一：

- `'bold'` - 字符轮廓比普通粗
- `'normal'` - 特定字体定义的默认粗细

示例： `ax.TitleFontWeight = 'normal'`

`FontUnits` - 字体大小单位

`'points'` （默认） | `'inches'` | `'centimeters'` | `'normalized'` | `'pixels'`

字体大小单位，指定为下列值之一。

Units	说明
`'points'`	磅。一磅等于 1/72 英寸。
`'inches'`	英寸。
`'centimeters'`	厘米。
`'normalized'`	按坐标区高度的一定比例来解释字体大小。如果您调整坐标区的大小，则字体大小也会相应地修改。例如，如果 `FontSize` 为 `0.1`（归一化单位），则文本为坐标区属性 `Position` 中存储的高度值的 1/10。
`'pixels'`	像素。 从 R2015b 开始，以像素为单位的距离不再依赖 Windows® 和 Macintosh 系统上的系统分辨率。 • 在 Windows 系统上，一个像素是 1/96 英寸。 • 在 Macintosh 系统上，一个像素是 1/72 英寸。 • 在 Linux® 系统上，一个像素的大小由系统分辨率确定。

要在单个函数调用中设置字体大小和字体单位，首先必须设置 `FontUnits` 属性，以便 `Axes` 对象能够正确解释指定的字体大小。

`FontSmoothing` - 字符平滑处理

`'on'` （默认） | `'off'`

字符平滑处理，指定为 `'on'` 或 `'off'`。

值	说明	结果
`'on'`	使用消除锯齿功能使文本在屏幕上看上去更平滑。 **示例：** `ax.FontSmoothing = 'on'`	S

值	说明	结果
'off'	不使用消除锯齿功能。在文本看似模糊时使用此设置。 **示例：** ax.FontSmoothing = 'off'	S

刻度

全部折叠

XTick, YTick, ZTick - 刻度值

[] （默认） | 由递增值组成的向量

刻度值，指定为由递增值组成的向量。如果您不希望沿坐标轴显示刻度线，请指定空向量 []。刻度值是坐标轴上显示刻度线的位置。刻度标签是您在每个刻度线旁边看到的标签。使用 XTickLabels、YTickLabels 和 ZTickLabels 属性指定关联的标签。

示例： ax.XTick = [2 4 6 8 10]

示例： ax.YTick = 0:10:100

也可以使用 xticks、yticks 和 zticks 函数指定刻度值。有关示例，请参阅指定坐标轴刻度值和标签。

数据类型： single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64 | categorical | datetime | duration

XTickMode, YTickMode, ZTickMode - 刻度值的选择模式

'auto' （默认） | 'manual'

刻度值的选择模式，指定为下列值之一：

- 'auto' - 根据坐标轴的数据范围自动选择刻度值。
- 'manual' - 手动指定刻度值。要指定值，请设置 XTick、YTick 或 ZTick 属性。

示例： ax.XTickMode = 'auto'

XTickLabel, YTickLabel, ZTickLabel - 刻度标签

'' （默认） | 字符向量元胞数组 | 字符串数组 | 分类数组

刻度标签，指定为字符向量元胞数组、字符串数组或分类数组。如果您不希望显示刻度标签，请指定空元胞数组 {}。如果您没有为所有刻度值指定足够多的标签，将会重复使用标签。

刻度标签支持 TeX 和 LaTeX 标记。有关详细信息，请参阅 TickLabelInterpreter 属性。

如果将此属性指定为分类数组，MATLAB 将使用数组中的值，而不是类别。

作为设置此属性的替代方法，您还可以使用 xticklabels、yticklabels 和 zticklabels 函数。有关示例，请参阅指定坐标轴刻度值和标签。

示例： ax.XTickLabel = {'Jan','Feb','Mar','Apr'}

XTickLabelMode, YTickLabelMode, ZTickLabelMode - 刻度标签的选择模式

'auto' （默认） | 'manual'

刻度标签的选择模式，指定为下列值之一：

- 'auto' - 自动选择刻度标签。
- 'manual' - 手动指定刻度标签。要指定标签，请设置 XTickLabel、YTickLabel 或 ZTickLabel 属性。

示例： ax.XTickLabelMode = 'auto'

TickLabelInterpreter - 刻度标签的解释

'tex' （默认） | 'latex' | 'none' | x

刻度标签的解释，指定为下列值之一：

- 'tex' - 使用 TeX 标记子集解释标签。

- `'latex'` - 使用 LaTeX 标记子集解释标签。
- `'none'` - 显示字面字符。

TeX 标记

默认情况下，MATLAB 支持一部分 TeX 标记。使用 TeX 标记可添加下标和上标，修改文本类型和颜色，并在文本中包括特殊字符。

下表列出了 `TickLabelInterpreter` 属性设置为 `'tex'`（默认值）时支持的修饰符。修饰符的作用范围一直到文本结束，但上标和下标除外。对于上标和下标，仅修改下一个字符或花括号 `{}` 中的文本。

修饰符	说明	示例
`^{ }`	上标	`'text^{superscript}'`
`_{ }`	下标	`'text_{subscript}'`
`\bf`	粗体	`'\bf text'`
`\it`	斜体	`'\it text'`
`\sl`	伪斜体（极少使用）	`'\sl text'`
`\rm`	常规字体	`'\rm text'`
`\fontname{specifier}`	将 `specifier` 设置为字体系列的名称以更改字体样式。您可以将此修饰符与其他修饰符结合使用。	`'\fontname{Courier} text'`
`\fontsize{specifier}`	将 `specifier` 设置为数值标量以更改字体大小。	`'\fontsize{15} text'`
`\color{specifier}`	将 `specifer` 设置为下列颜色之一：`red`、`green`、`yellow`、`magenta`、`blue`、`black`、`white`、`gray`、`darkGreen`、`orange` 或 `lightBlue`。	`'\color{magenta} text'`
`\color[rgb] {specifier}`	将 `specifier` 设置为包含三个元素的 RGB 三元组以更改字体颜色。	`'\color[rgb] {0,0.5,0.5} text'`

下表列出了当 `Interpreter` 属性设置为 `'tex'` 时支持的特殊字符。

字符序列	符号	字符序列	符号	字符序列	符号
`\alpha`	α	`\upsilon`	υ	`\sim`	~
`\angle`	∠	`\phi`	ϕ	`\leq`	≤
`\ast`	*	`\chi`	χ	`\infty`	∞
`\beta`	β	`\psi`	ψ	`\clubsuit`	♣
`\gamma`	γ	`\omega`	ω	`\diamondsuit`	♦
`\delta`	δ	`\Gamma`	Γ	`\heartsuit`	♥
`\epsilon`	ϵ	`\Delta`	Δ	`\spadesuit`	♠
`\zeta`	ζ	`\Theta`	Θ	`\leftrightarrow`	↔
`\eta`	η	`\Lambda`	Λ	`\leftarrow`	←
`\theta`	θ	`\Xi`	Ξ	`\Leftarrow`	⇐
`\vartheta`	ϑ	`\Pi`	Π	`\uparrow`	↑
`\iota`	ι	`\Sigma`	Σ	`\rightarrow`	→
`\kappa`	κ	`\Upsilon`	ϒ	`\Rightarrow`	⇒
`\lambda`	λ	`\Phi`	Φ	`\downarrow`	↓
`\mu`	μ	`\Psi`	Ψ	`\circ`	°
`\nu`	ν	`\Omega`	Ω	`\pm`	±
`\xi`	ξ	`\forall`	∀	`\geq`	≥
`\pi`	π	`\exists`	∃	`\propto`	∝
`\rho`	ρ	`\ni`	∋	`\partial`	∂
`\sigma`	σ	`\cong`	≅	`\bullet`	•
`\varsigma`	ς	`\approx`	≈	`\div`	÷
`\tau`	τ	`\Re`	ℜ	`\neq`	≠
`\equiv`	≡	`\oplus`	⊕	`\aleph`	ℵ
`\Im`	ℑ	`\cup`	∪	`\wp`	℘
`\otimes`	⊗	`\subseteq`	⊆	`\oslash`	∅
`\cap`	∩	`\in`	∈	`\supseteq`	⊇
`\supset`	⊃	`\lceil`	⌈	`\subset`	⊂

字符序列	符号	字符序列	符号	字符序列	符号
\int	∫	\cdot	·	\o	ο
\rfloor	⌋	\neg	¬	\nabla	∇
\lfloor	⌊	\times	x	\ldots	…
\perp	⊥	\surd	√	\prime	′
\wedge	∧	\varpi	ϖ	\0	∅
\rceil	⌉	\rangle	〉	\mid	|
\vee	∨	\langle	〈	\copyright	©

LaTeX 标记

要使用 LaTeX 标记，请将 `TickLabelInterpreter` 属性设置为 `'latex'`。使用美元符号将文本括起来，例如，对于内联模式，使用 `'$\int_1^{20} x^2 dx$'`；对于显示模式，使用 `'$$\int_1^{20} x^2 dx$$'`。

显示的文本将使用默认的 LaTeX 字体样式。要更改字体样式，请在文本中使用 LaTeX 标记。`FontName`、`FontWeight` 和 `FontAngle` 属性不起作用。

用于 LaTeX 解释器的文本最多可以包含 1200 个字符。对于多行文本，此限制会在此数量的基础上每行减少约 10 个字符。有关 LaTeX 系统的详细信息，请访问 LaTeX Project 网站：www.latex-project.org。

`XTickLabelRotation`, `YTickLabelRotation`, `ZTickLabelRotation` - 刻度标签的旋转

`0` （默认） | 以度为单位的数值

刻度标签的旋转，指定为以度为单位的数值。正值将导致按逆时针旋转。负值产生顺时针旋转。

示例： `ax.XTickLabelRotation = 45`

示例： `ax.YTickLabelRotation = 90`

也可以使用 `xtickangle`、`ytickangle` 和 `ztickangle` 函数。

`XMinorTick`, `YMinorTick`, `ZMinorTick` - 次刻度线

`'off'` | `'on'`

次刻度线，指定为下列值之一：

- `'off'` - 不显示次刻度线。此值是使用线性刻度的坐标轴的默认值。
- `'on'` - 在坐标轴的主刻度线之间显示次刻度线。主刻度线的间距决定次刻度线的数目。此值是使用对数刻度的坐标轴的默认值。

示例： `ax.XMinorTick = 'on'`

`TickDir` - 刻度线方向

`'in'` （默认） | `'out'` | `'both'`

刻度线方向，指定为下列值之一：

- `'in'` - 刻度线从轴线指向内部。（二维视图的默认值）
- `'out'` - 刻度线从轴线指向外部。（三维视图的默认值）
- `'both'` - 刻度线以轴线为中心。

示例： `ax.TickDir = 'out'`

`TickDirMode` - `TickDir` 的选择模式

`'auto'` （默认） | `'manual'`

`TickDir` 属性的选择模式，指定为下列值之一：

- `'auto'` - 根据当前视图自动选择刻度方向。
- `'manual'` - 手动指定刻度方向。要指定刻度方向，请设置 `TickDir` 属性。

示例： `ax.TickDirMode = 'auto'`

TickLength - 刻度线长度

[0.01 0.025] （默认） | 二元素向量

刻度线长度，指定为 [2Dlength 3Dlength] 形式的二元素向量。第一个元素是二维视图中的刻度线长度，第二个元素是三维视图中的刻度线长度。指定值是以可见的 *x* 轴、*y* 轴或 *z* 轴中最长线条为基准进行归一化的值。

示例： ax.TickLength = [0.02 0.035]

标尺

全部折叠

XLim, YLim, ZLim - 最小和最大坐标轴范围

[0 1] （默认） | [min max] 形式的二元素向量

最小和最大范围，指定为 [min max] 形式的二元素向量，其中 max 大于 min。您可以将范围指定为数字、分类、日期时间或持续时间值。但您指定的值类型必须与坐标轴上的值类型匹配。

您可以指定这两个范围，也可以指定一个范围而让坐标区自动计算另一个范围。对于自动计算的最小值或最小值范围，分别使用 -inf 或 inf 来表示。

示例： ax.XLim = [0 10]

示例： ax.YLim = [-inf 10]

示例： ax.ZLim = [0 inf]

也可以使用 xlim、ylim 和 zlim 函数设置这些范围。有关示例，请参阅指定坐标轴范围。

数据类型： single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64 | datetime | duration

XLimMode, YLimMode, ZLimMode - 坐标轴范围的选择模式

'auto' （默认） | 'manual'

坐标轴范围的选择模式，指定为下列值之一：

- 'auto' - 根据绘制的数据（即坐标区中显示的所有对象的 XData、YData 或 ZData 的总体范围）自动选择坐标轴范围。
- 'manual' - 手动指定坐标轴范围。要指定坐标轴范围，请设置 XLim、YLim 或 ZLim 属性。

示例： ax.XLimMode = 'auto'

XAxis, YAxis, ZAxis - 轴标尺

标尺对象

轴标尺，以标尺对象的形式返回。标尺控制 *x* 轴、*y* 轴或 *z* 轴的外观和行为。通过访问相关的标尺并设置标尺属性，可以修改坐标轴的外观和行为。MATLAB 为每个轴创建的标尺类型取决于绘制的数据。有关 Axes 对象支持的标尺属性的列表，请参阅：

- NumericRuler 属性
- DatetimeRuler 属性
- DurationRuler 属性
- CategoricalRuler 属性

例如，通过 XAxis 属性访问 *x* 轴的标尺。然后，更改标尺的 Color 属性，使 *x* 轴的颜色为红色。同样，将 *y* 轴的颜色更改为绿色。

```
ax = gca;
ax.XAxis.Color = 'r';
ax.YAxis.Color = 'g';
```

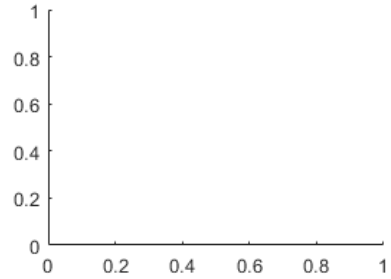
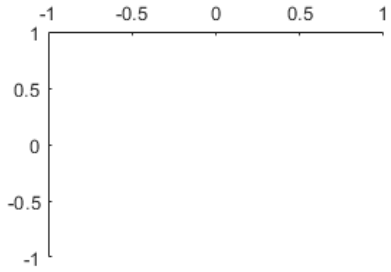
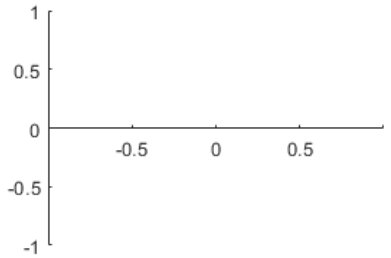
如果 Axes 对象有两个 y 轴，则 YAxis 属性存储两个标尺对象。

XAxisLocation - x 轴位置

'bottom' （默认） | 'top' | 'origin'

x 轴位置，指定为下表中的值之一。此属性仅应用于二维视图。

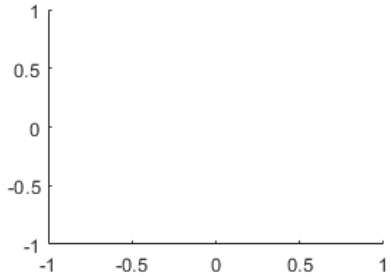
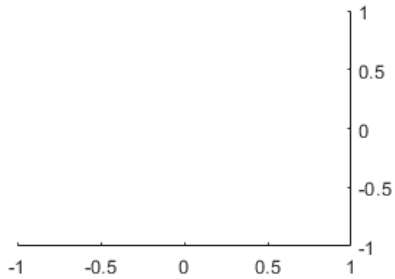
值	说明	结果

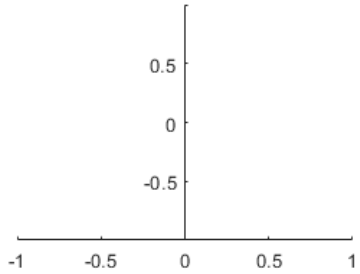
值	说明	结果
`'bottom'`	坐标区的底部。 **示例：** `ax.XAxisLocation = 'bottom'`	
`'top'`	坐标区的顶部。 **示例：** `ax.XAxisLocation = 'top'`	
`'origin'`	穿过原点 (0,0)。 **示例：** `ax.XAxisLocation = 'origin'`	

`YAxisLocation` - y 轴位置

`'left'` （默认） | `'right'` | `'origin'`

y 轴位置，指定为下表中的值之一。此属性仅应用于二维视图。

值	说明	结果
`'left'`	坐标区的左侧。 **示例：** `ax.YAxisLocation = 'left'`	
`'right'`	坐标区的右侧。 **示例：** `ax.YAxisLocation = 'right'`	

值	说明	结果
`'origin'`	穿过原点 (0,0)。 **示例：** `ax.YAxisLocation = 'origin'`	

`XColor`, `YColor`, `ZColor` - 轴线、刻度值和标签的颜色

`[0.15 0.15 0.15]` （默认） | RGB 三元组 | 十六进制颜色代码 | `'r'` | `'g'` | `'b'` | ...

x、*y* 或 *z* 方向的轴线、刻度值和标签的颜色，指定为 RGB 三元组、十六进制颜色代码、颜色名称或短名称。您指定的颜色还会影响网格线，除非您使用 `GridColor` 或 `MinorGridColor` 属性指定网格线颜色。

对于自定义颜色，请指定 RGB 三元组或十六进制颜色代码。

- RGB 三元组是包含三个元素的行向量，其元素分别指定颜色中红、绿、蓝分量的强度。强度值必须位于 `[0,1]` 范围内，例如 `[0.4 0.6 0.7]`。
- 十六进制颜色代码是字符向量或字符串标量，以井号 (`#`) 开头，后跟三个或六个十六进制数字，范围可以是 `0` 到 `F`。这些值不区分大小写。因此，颜色代码 `'#FF8800'` 与 `'#ff8800'`、`'#F80'` 与 `'#f80'` 是等效的。

此外，还可以按名称指定一些常见的颜色。下表列出了命名颜色选项、等效 RGB 三元组和十六进制颜色代码。

颜色名称	短名称	RGB 三元组	十六进制颜色代码	外观
`'red'`	`'r'`	`[1 0 0]`	`'#FF0000'`	
`'green'`	`'g'`	`[0 1 0]`	`'#00FF00'`	
`'blue'`	`'b'`	`[0 0 1]`	`'#0000FF'`	
`'cyan'`	`'c'`	`[0 1 1]`	`'#00FFFF'`	
`'magenta'`	`'m'`	`[1 0 1]`	`'#FF00FF'`	
`'yellow'`	`'y'`	`[1 1 0]`	`'#FFFF00'`	
`'black'`	`'k'`	`[0 0 0]`	`'#000000'`	
`'white'`	`'w'`	`[1 1 1]`	`'#FFFFFF'`	
`'none'`	不适用	不适用	不适用	无颜色

以下是 MATLAB 在许多类型的绘图中使用的默认颜色的 RGB 三元组和十六进制颜色代码。

RGB 三元组	十六进制颜色代码	外观
`[0 0.4470 0.7410]`	`'#0072BD'`	
`[0.8500 0.3250 0.0980]`	`'#D95319'`	
`[0.9290 0.6940 0.1250]`	`'#EDB120'`	
`[0.4940 0.1840 0.5560]`	`'#7E2F8E'`	
`[0.4660 0.6740 0.1880]`	`'#77AC30'`	
`[0.3010 0.7450 0.9330]`	`'#4DBEEE'`	
`[0.6350 0.0780 0.1840]`	`'#A2142F'`	

示例： `ax.XColor = [1 1 0]`

示例： `ax.YColor = 'y'`

示例： `ax.ZColor = 'yellow'`

示例： `ax.ZColor = '#FFFF00'`

`XColorMode` - 用于设置 x 轴网格颜色的属性

`'auto'` （默认） | `'manual'`

用于设置 x 轴网格颜色的属性，指定为 `'auto'` 或 `'manual'`。模式值仅影响 x 轴网格颜色。无论在什么模式下，x 轴线、刻度值和标签始终使用 `XColor` 值。

x 轴网格颜色取决于 `XColorMode` 属性和 `GridColorMode` 属性，如下表所示。

XColorMode	GridColorMode	x 轴网格颜色
`'auto'`	`'auto'`	`GridColor` 属性
	`'manual'`	`GridColor` 属性
`'manual'`	`'auto'`	`XColor` 属性
	`'manual'`	`GridColor` 属性

x 轴次网格颜色取决于 `XColorMode` 属性和 `MinorGridColorMode` 属性，如下表所示。

XColorMode	MinorGridColorMode	x 轴次网格颜色
`'auto'`	`'auto'`	`MinorGridColor` 属性
	`'manual'`	`MinorGridColor` 属性
`'manual'`	`'auto'`	`XColor` 属性
	`'manual'`	`MinorGridColor` 属性

`YColorMode` - 用于设置 y 轴网格颜色的属性

`'auto'` （默认） | `'manual'`

用于设置 y 轴网格颜色的属性，指定为 `'auto'` 或 `'manual'`。模式值仅影响 y 轴网格颜色。无论在什么模式下，y 轴线、刻度值和标签始终使用 `YColor` 值。

y 轴网格颜色取决于 `YColorMode` 属性和 `GridColorMode` 属性，如下表所示。

YColorMode	GridColorMode	y 轴网格颜色
`'auto'`	`'auto'`	`GridColor` 属性
	`'manual'`	`GridColor` 属性
`'manual'`	`'auto'`	`YColor` 属性
	`'manual'`	`GridColor` 属性

y 轴次网格颜色取决于 `YColorMode` 属性和 `MinorGridColorMode` 属性，如下表所示。

YColorMode	MinorGridColorMode	y 轴次网格颜色
`'auto'`	`'auto'`	`MinorGridColor` 属性
	`'manual'`	`MinorGridColor` 属性
`'manual'`	`'auto'`	`YColor` 属性
	`'manual'`	`MinorGridColor` 属性

`ZColorMode` - 用于设置 z 轴网格颜色的属性

`'auto'` （默认） | `'manual'`

用于设置 z 轴网格颜色的属性，指定为 `'auto'` 或 `'manual'`。模式值仅影响 z 轴网格颜色。无论在什么模式下，z 轴线、刻度值和标签始终使用 `ZColor` 值。

z 轴网格颜色取决于 `ZColorMode` 属性和 `GridColorMode` 属性，如下表所示。

ZColorMode	GridColorMode	z 轴网格颜色
`'auto'`	`'auto'`	`GridColor` 属性
	`'manual'`	`GridColor` 属性
`'manual'`	`'auto'`	`ZColor` 属性

ZColorMode	GridColorMode	z 轴网格颜色
	'manual'	GridColor 属性

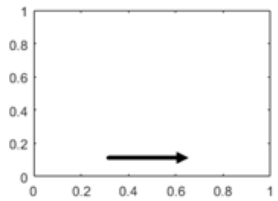
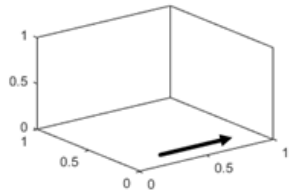
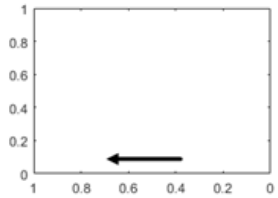
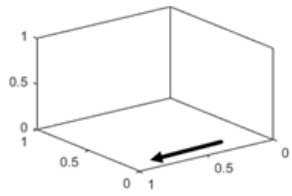
z 轴次网格颜色取决于 ZColorMode 属性和 MinorGridColorMode 属性，如下表所示。

ZColorMode	MinorGridColorMode	z 轴次网格颜色
'auto'	'auto'	MinorGridColor 属性
	'manual'	MinorGridColor 属性
'manual'	'auto'	ZColor 属性
	'manual'	MinorGridColor 属性

XDir - x 轴方向

'normal' （默认） | 'reverse'

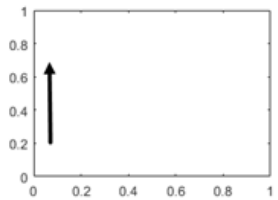
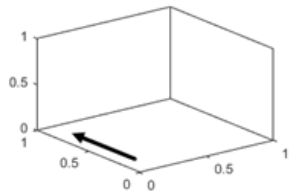
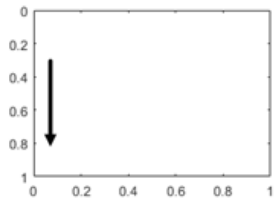
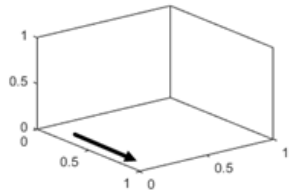
x 轴方向，指定为下列值之一。

值	说明	生成二维视图	生成三维视图
'normal'	值从左向右逐渐增加。 **示例：** ax.XDir = 'normal'		
'reverse'	值从右向左逐渐增加。 **示例：** ax.XDir = 'reverse'		

YDir - y 轴方向

'normal' （默认） | 'reverse'

y 轴方向，指定为下列值之一。

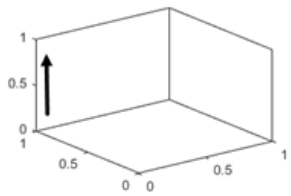
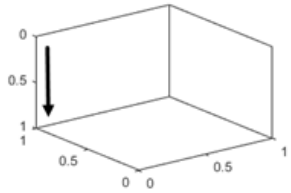
值	说明	生成二维视图	生成三维视图
'normal'	值从下向上（二维视图）或从前向后（三维视图）逐渐增加。 **示例：** ax.YDir = 'normal'		
'reverse'	值从上向下（二维视图）或从后向前（三维视图）逐渐增加。 **示例：** ax.YDir = 'reverse'		

ZDir - z 轴方向

'normal' （默认） | 'reverse'

z 轴方向，指定为下列值之一。

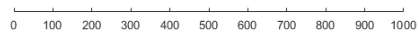
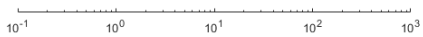
值	说明	生成三维视图

值	说明	生成三维视图
'normal'	值按从内向外（二维视图）或从下向上（三维视图）逐渐增加。 **示例：** ax.ZDir = 'normal'	
'reverse'	值按从外向内（二维视图）或从上向下（三维视图）逐渐增加。 **示例：** ax.ZDir = 'reverse'	

XScale, YScale, ZScale - 值沿坐标轴的标度

'linear' （默认） | 'log'

坐标轴刻度，指定为下列值之一。

值	说明	结果
'linear'	线性刻度 **示例：** ax.XScale = 'linear'	
'log'	对数刻度 **示例：** ax.XScale = 'log'	

网格

全部折叠

XGrid, YGrid, ZGrid - 网格线

'off' （默认） | 'on'

网格线，指定为下列值之一：

- 'off' - 不显示网格线。
- 'on' - 坐标轴正交的网格线，例如，沿着常量 *x*、*y* 或 *z* 值对应的线条。

还可以使用 grid on 或 grid off 命令将所有三个属性分别设置为 'on' 或 'off'。有关详细信息，请参阅 grid。

示例： ax.XGrid = 'on'

Layer - 网格线和刻度线的位置

'bottom' （默认） | 'top'

网格线和刻度线相对于图形对象的位置，指定为下列值之一：

- 'bottom' - 在图形对象下方显示刻度线和网格线。
- 'top' - 在图形对象上方显示刻度线和网格线。

此属性仅影响二维视图。

示例： ax.Layer = 'top'

GridLineStyle - 网格线的线型

'-' （默认） | '--' | ':' | '-.' | 'none'

网格线的线型，指定为下表中的线型之一。

线型	说明	表示的线条

线型	说明	表示的线条
'-'	实线	————————
'--'	虚线	- — — — —
':'	点线	.....
'-.'	点划线	—·—·—·—·-
'none'	无线条	无线条

要显示网格线，请使用 grid on 命令或将 XGrid、YGrid 或 ZGrid 属性设置为 'on'。

示例： ax.GridLineStyle = '--'

GridColor - 网格线的颜色

[0.15 0.15 0.15] （默认） | RGB 三元组 | 十六进制颜色代码 | 'r' | 'g' | 'b' | ...

网格线的颜色，指定为 RGB 三元组、十六进制颜色代码、颜色名称或短名称。

对于自定义颜色，请指定 RGB 三元组或十六进制颜色代码。

- RGB 三元组是包含三个元素的行向量，其元素分别指定颜色中红、绿、蓝分量的强度。强度值必须位于 [0,1] 范围内，例如 [0.4 0.6 0.7]。
- 十六进制颜色代码是字符向量或字符串标量，以井号 (#) 开头，后跟三个或六个十六进制数字，范围可以是 0 到 F。这些值不区分大小写。因此，颜色代码 '#FF8800' 与 '#ff8800'、'#F80' 与 '#f80' 是等效的。

此外，还可以按名称指定一些常见的颜色。下表列出了命名颜色选项、等效 RGB 三元组和十六进制颜色代码。

颜色名称	短名称	RGB 三元组	十六进制颜色代码	外观
'red'	'r'	[1 0 0]	'#FF0000'	
'green'	'g'	[0 1 0]	'#00FF00'	
'blue'	'b'	[0 0 1]	'#0000FF'	
'cyan'	'c'	[0 1 1]	'#00FFFF'	
'magenta'	'm'	[1 0 1]	'#FF00FF'	
'yellow'	'y'	[1 1 0]	'#FFFF00'	
'black'	'k'	[0 0 0]	'#000000'	
'white'	'w'	[1 1 1]	'#FFFFFF'	
'none'	不适用	不适用	不适用	无颜色

以下是 MATLAB 在许多类型的绘图中使用的默认颜色的 RGB 三元组和十六进制颜色代码。

RGB 三元组	十六进制颜色代码	外观
[0 0.4470 0.7410]	'#0072BD'	
[0.8500 0.3250 0.0980]	'#D95319'	
[0.9290 0.6940 0.1250]	'#EDB120'	
[0.4940 0.1840 0.5560]	'#7E2F8E'	
[0.4660 0.6740 0.1880]	'#77AC30'	
[0.3010 0.7450 0.9330]	'#4DBEEE'	
[0.6350 0.0780 0.1840]	'#A2142F'	

要为坐标区框轮廓设置颜色，请使用 XColor、YColor 和 ZColor 属性。

要显示网格线，请使用 grid on 命令或将 XGrid、YGrid 或 ZGrid 属性设置为 'on'。

示例： ax.GridColor = [0 0 1]

示例： ax.GridColor = 'b'

示例： ax.GridColor = 'blue'

示例： ax.GridColor = '#0000FF'

GridColorMode - 用于设置网格颜色的属性

`'auto'` （默认） | `'manual'`

用于设置网格颜色的属性，指定为下列值之一：

- `'auto'` - 检查 `XColorMode`、`YColorMode` 和 `ZColorMode` 属性的值，以确定 x、y 和 z 方向的网格线颜色。
- `'manual'` - 使用 `GridColor` 设置所有方向的网格线颜色。

GridAlpha - 网格线透明度

`0.15` （默认） | 范围 `[0,1]` 内的值

网格线透明度，指定为范围 `[0,1]` 中的值。值为 `1` 表示不透明，值为 `0` 表示完全透明。

示例： `ax.GridAlpha = 0.5`

GridAlphaMode - GridAlpha 的选择模式

`'auto'` （默认） | `'manual'`

`GridAlpha` 属性的选择模式，指定为下列值之一：

- `'auto'` - 使用默认透明度值 `0.15`。
- `'manual'` - 手动指定透明度值。要指定值，请设置 `GridAlpha` 属性。

示例： `ax.GridAlphaMode = 'auto'`

XMinorGrid, YMinorGrid, ZMinorGrid - 次网格线

`'off'` （默认） | `'on'`

次网格线，指定为下列值之一：

- `'off'` - 不显示网格线。
- `'on'` - 显示与轴的次刻度线对齐的网格线。不必启用次刻度即可显示次网格线。

或者使用 `grid minor` 命令切换次网格线的可见性。

示例： `ax.XMinorGrid = 'on'`

MinorGridLineStyle - 次网格线的线型

`':'` （默认） | `'-'` | `'--'` | `'-.'` | `'none'`

次网格线的线型，指定为下表中显示的线型之一。

线型	说明	表示的线条
`'-'`	实线	————————
`'--'`	虚线	- — — — —
`':'`	点线	.....
`'-.'`	点划线	—·—·—·—·-
`'none'`	无线条	无线条

要显示次网格线，请使用 `grid minor` 命令或将 `XMinorGrid`、`YMinorGrid` 或 `ZMinorGrid` 属性设置为 `'on'`。

示例： `ax.MinorGridLineStyle = '-.'`

MinorGridColor - 次网格线的颜色

`[0.1 0.1 0.1]` （默认） | RGB 三元组 | 十六进制颜色代码 | `'r'` | `'g'` | `'b'` | ...

次网格线的颜色，指定为 RGB 三元组、十六进制颜色代码、颜色名称或短名称。

对于自定义颜色，请指定 RGB 三元组或十六进制颜色代码。

- RGB 三元组是包含三个元素的行向量，其元素分别指定颜色中红、绿、蓝分量的强度。强度值必须位于 `[0,1]` 范围内，例如 `[0.4 0.6 0.7]`。
- 十六进制颜色代码是字符向量或字符串标量，以井号 (`#`) 开头，后跟三个或六个十六进制数字，范围可以是 `0` 到 `F`。这些值不区分大小写。因此，颜色代码 `'#FF8800'` 与 `'#ff8800'`、`'#F80'` 与 `'#f80'` 是等效的。

此外，还可以按名称指定一些常见的颜色。下表列出了命名颜色选项、等效 RGB 三元组和十六进制颜色代码。

颜色名称	短名称	RGB 三元组	十六进制颜色代码	外观
`'red'`	`'r'`	`[1 0 0]`	`'#FF0000'`	
`'green'`	`'g'`	`[0 1 0]`	`'#00FF00'`	
`'blue'`	`'b'`	`[0 0 1]`	`'#0000FF'`	
`'cyan'`	`'c'`	`[0 1 1]`	`'#00FFFF'`	
`'magenta'`	`'m'`	`[1 0 1]`	`'#FF00FF'`	
`'yellow'`	`'y'`	`[1 1 0]`	`'#FFFF00'`	
`'black'`	`'k'`	`[0 0 0]`	`'#000000'`	
`'white'`	`'w'`	`[1 1 1]`	`'#FFFFFF'`	
`'none'`	不适用	不适用	不适用	无颜色

以下是 MATLAB 在许多类型的绘图中使用的默认颜色的 RGB 三元组和十六进制颜色代码。

RGB 三元组	十六进制颜色代码	外观
`[0 0.4470 0.7410]`	`'#0072BD'`	
`[0.8500 0.3250 0.0980]`	`'#D95319'`	
`[0.9290 0.6940 0.1250]`	`'#EDB120'`	
`[0.4940 0.1840 0.5560]`	`'#7E2F8E'`	
`[0.4660 0.6740 0.1880]`	`'#77AC30'`	
`[0.3010 0.7450 0.9330]`	`'#4DBEEE'`	
`[0.6350 0.0780 0.1840]`	`'#A2142F'`	

要显示次网格线，请使用 `grid minor` 命令或将 `XMinorGrid`、`YMinorGrid` 或 `ZMinorGrid` 属性设置为 `'on'`。

示例： `ax.MinorGridColor = [0 0 1]`

示例： `ax.MinorGridColor = 'b'`

示例： `ax.MinorGridColor = 'blue'`

示例： `ax.MinorGridColor = '#0000FF'`

`MinorGridColorMode` - 用于设置次网格颜色的属性

`'auto'` （默认） | `'manual'`

用于设置次网格颜色的属性，指定为下列值之一：

- `'auto'` - 检查 `XColorMode`、`YColorMode` 和 `ZColorMode` 属性的值，以确定 x、y 和 z 方向的网格线颜色。
- `'manual'` - 使用 `MinorGridColor` 设置所有方向的次网格线颜色。

`MinorGridAlpha` - 次网格线的透明度

`0.25` （默认） | 范围 `[0,1]` 内的值

次网格线透明度，指定为范围 `[0,1]` 中的值。值为 `1` 表示不透明，值为 `0` 表示完全透明。

示例： `ax.MinorGridAlpha = 0.5`

`MinorGridAlphaMode` - `MinorGridAlpha` 的选择模式

`'auto'` （默认） | `'manual'`

MinorGridAlpha 属性的选择模式，指定为下列值之一：

- 'auto' - 使用默认透明度值 0.25。
- 'manual' - 手动指定透明度值。要指定值，请设置 MinorGridAlpha 属性。

示例： ax.MinorGridAlphaMode = 'auto'

标签

全部折叠

Title - 坐标区标题的文本对象

文本对象

坐标区标题的文本对象。要添加标题，请设置文本对象的 String 属性。要更改标题外观，例如字体样式或字体颜色，请设置其他属性。有关完整列表，请参阅 Text 属性。

```
ax = gca;
ax.Title.String = 'My Title';
ax.Title.FontWeight = 'normal';
```

也可以使用 title 函数添加标题并控制外观。

```
title('My Title','FontWeight','normal')
```

注意

此文本对象不包含在坐标区的 Children 属性中，它不能由 findobj 返回，也不使用为文本对象定义的默认值。

XLabel, YLabel, ZLabel - 坐标轴标签的文本对象

文本对象

轴标签的文本对象。要添加轴标签，请设置文本对象的 String 属性。要更改标签外观，例如字号，请设置其他属性。有关完整列表，请参阅 Text 属性。

```
ax = gca;
ax.YLabel.String = 'My y-Axis Label';
ax.YLabel.FontSize = 12;
```

也可以使用 xlabel、ylabel 和 zlabel 函数添加轴标签并控制外观。

```
ylabel('My y-Axis Label','FontSize',12)
```

注意

这些文本对象不包含在坐标区的 Children 属性中，它们不能由 findobj 返回，也不使用为文本对象定义的默认值。

Legend - 与坐标区关联的图例

empty GraphicsPlaceholder （默认） | Legend 对象

此属性为只读

与 Axes 对象关联的图例，指定为 Legend 对象。要向坐标区中添加图例，请使用 legend 函数。然后，您可以使用此属性修改图例。有关完整的属性列表，请参阅 Legend 属性。

```
plot(rand(3))
legend({'Line 1','Line 2','Line 3'},'FontSize',12)
ax = gca;
ax.Legend.TextColor = 'red';
```

还可以使用此属性确定坐标区是否包含图例。

```
ax = gca;
lgd = ax.Legend
if ~isempty(lgd)
    disp('Legend Exists')
end
```

多个绘图

全部折叠

ColorOrder - 色序

七种预定义颜色 （默认） | 由 RGB 三元组组成的三列矩阵

色序，指定为由 RGB 三元组组成的三列矩阵。此属性定义 MATLAB 创建绘图对象（例如 Line、Scatter 和 Bar 对象）时使用的颜色的调色板。数组的每一行都是一个 RGB 三元组。RGB 三元组是包含三个元素的向量，其元素分别指定颜色的红、绿、蓝分量的强度。强度必须在 [0, 1] 范围内。下表列出了默认颜色。

颜色	ColorOrder 矩阵

	`[ 0 0.4470 0.7410`
	`0.8500 0.3250 0.0980`
	`0.9290 0.6940 0.1250`
	`0.4940 0.1840 0.5560`
	`0.4660 0.6740 0.1880`
	`0.3010 0.7450 0.9330`
	`0.6350 0.0780 0.1840]`

MATLAB 根据对象的创建顺序为对象分配颜色。例如，在绘制线条时，第一条线用第一种颜色，第二条线用第二种颜色，以此类推。如果线条比颜色多，则循环使用颜色。

在绘图之前或之后更改色序

从 R2019b 开始，您可以通过下列任一方式更改色序：

- 调用 colororder 函数以更改图窗中所有坐标区的色序。图窗中现有绘图的颜色会立即更新。如果您在图窗中放入其他坐标区，则这些坐标区也会使用新的色序。如果您继续调用绘图命令，这些命令也会使用新的颜色。
- 对坐标区设置 ColorOrder 属性，调用 hold 函数以将坐标区保留状态设置为 'on'，然后调用所需的绘图函数。这类似于调用 colororder 函数，但在本例中是为特定坐标区设置色序，而不是为整个图窗设置。必须将 hold 状态设置为 'on'，以确保后续绘图命令不会将坐标区重置为使用默认色序。

如果您使用 R2019a 或更早版本，则更改色序矩阵不会影响现有绘图。要更改现有绘图中的颜色，您必须设置 ColorOrder 属性，然后在调用任何绘图函数之前将坐标区保留状态设置为 'on'。

此属性的行为在 R2019b 中已更改。有关详细信息，请参阅：

- 更改坐标区的 ColorOrder 或 LineStyleOrder 会立即影响现有绘图
- ColorOrder 和 LineStyleOrder 的索引方案可能会更改绘图的颜色和线型。

ColorOrderIndex - 色序索引

1 （默认） | 正整数

色序索引，指定为正整数。此属性指定当 MATLAB 创建下一个绘图对象（例如 Line、Scatter 或 Bar 对象）时从坐标区 ColorOrder 属性中选择的下一个颜色。例如，如果色序索引值为 1，则添加到坐标区的下一个对象将使用 ColorOrder 矩阵中的第一个颜色。如果索引值超过 ColorOrder 矩阵中的颜色数，则通过索引值对 ColorOrder 矩阵中的颜色数求模来决定下一个对象的颜色。

如果坐标区的 NextPlot 属性设置为 'add'，则每次在坐标区中添加新绘图时色序索引值都会增加。要再次从第一个颜色开始，请将 ColorOrderIndex 属性设置为 1。

此属性的行为在 R2019b 中已更改。有关详细信息，请参阅ColorOrder 和 LineStyleOrder 的索引方案可能会更改绘图的颜色和线型。。

LineStyleOrder - 线型序列

'-' 实线 （默认） | 字符向量 | 字符向量元胞数组 | 字符串数组

线型序列，指定为字符向量、字符向量元胞数组或字符串数组。此属性列出了 MATLAB 在坐标区中显示多个绘图线条时使用的线型。MATLAB 根据创建线条的顺序为它们分配线型。只有在对当前线型使用 ColorOrder 属性中的所有颜色后，才会更改为使用下一线型。默认的 LineStyleOrder 只有一个线型 '-'。

要自定义线型序列，请创建一个字符向量元胞数组或字符串数组。将数组的每个元素指定为下列各表中的一个线条设定符或标记设定符。您可以在一个元素中组合使用线条和标记设定符，如 `'-*'`。

线条设定符	说明
`'-'` （默认值）	实线
`'--'`	虚线
`':'`	点线
`'-.'`	点划线

标记设定符	说明
`'+'`	加号标记
`'o'`	圆形标记
`'*'`	星形标记
`'.'`	点标记
`'x'`	交叉标记
`'s'`	方形标记
`'d'`	菱形标记
`'^'`	上三角标记
`'v'`	下三角标记
`'>'`	右三角标记
`'<'`	左三角标记
`'p'`	五角星（五角形）标记
`'h'`	六角星（六角形）标记

在绘图之前或之后更改线型序列

从 R2019b 开始，您可以于坐标区中绘图之前或之后更改线型序列。当 `LineStyleOrder` 属性设置为新值时，MATLAB 会更新坐标区中所有线条的线型。如果继续在坐标区中绘图，绘图命令会继续使用更新后的列表中的线型。

如果您使用的是 R2019a 或更早版本，则必须在绘图之前更改线型序列。设置 `LineStyleOrder` 属性的值，然后在调用任何绘图函数之前调用 `hold` 函数以将坐标区保留状态设置为 `'on'`。

此属性的行为在 R2019b 中已更改。有关详细信息，请参阅：

- 更改坐标区的 ColorOrder 或 LineStyleOrder 会立即影响现有绘图
- ColorOrder 和 LineStyleOrder 的索引方案可能会更改绘图的颜色和线型。

`LineStyleOrderIndex` - 线型序列索引

`1` （默认） | 正整数

线型序列索引，指定为正整数。此属性指定 MATLAB 在创建下一个绘图线条时从坐标区 `LineStyleOrder` 属性中选择的下一个线型。例如，如果此属性设置为 `1`，则添加到坐标区的下一个绘图线条将使用 `LineStyleOrder` 属性中的第一项。如果索引值超过 `LineStyleOrder` 数组中的线型数，则使用索引值对 `LineStyleOrder` 数组中元素数量求模来决定下一个线条的线型。

当坐标区的 `NextPlot` 属性设置为 `'add'` 时，MATLAB 会在对当前线型用尽 `ColorOrder` 属性中的所有颜色后递增索引值。要再次从第一个线型开始，请将 `LineStyleOrderIndex` 属性设置为 `1`。

此属性的行为在 R2019b 中已更改。有关详细信息，请参阅ColorOrder 和 LineStyleOrder 的索引方案可能会更改绘图的颜色和线型。。

`NextPlot` - 要重置的属性

`'replace'` （默认） | `'add'` | `'replacechildren'` | `'replaceall'`

在向坐标区中添加新绘图时要重置的属性，指定为下列值之一：

- `'add'` - 在现有坐标区上添加新绘图。在显示新绘图之前，不删除现有绘图或重置坐标区属性。
- `'replacechildren'` - 在显示新绘图之前删除现有绘图。可将 `ColorOrderIndex` 和 `LineStyleOrderIndex` 属性重置为 `1`，但不要重置其他坐标区属性。添加到坐标区的下一个绘图基于 `ColorOrder` 和 `LineStyle` 序列属性使用第一个颜色和线型。此值类似于在每个新绘图之前使用 `cla`。
- `'replace'` - 在显示新绘图之前，删除现有绘图并将除 `Position` 和 `Units` 之外的所有坐标区属性重置为默认值。

- `'replaceall'` - 在显示新绘图之前，删除现有绘图并将除 `Position` 和 `Units` 之外的所有坐标区属性重置为默认值。此值类似于在每个新绘图之前使用 `cla reset`。

> **注意**
>
> 对于只有一个 *y* 轴的 `Axes` 对象，`'replace'` 和 `'replaceall'` 属性值是等同的。对于具有两个 *y* 轴的 `Axes` 对象，`'replace'` 值只影响活动侧，而 `'replaceall'` 值会同时影响两侧。

图窗对象还有一个 `NextPlot` 属性。也可以使用 `newplot` 函数准备图窗和坐标区，以执行后续的图形命令。

`SortMethod` - 渲染对象的顺序

`'depth'` | `'childorder'`

渲染对象的顺序，指定为下列值之一：

- `'depth'` - 基于当前视图按从后到前的顺序绘制对象。使用此值可确保前方对象得到正确绘制渲染（相对于后方对象）。
- `'childorder'` - 按图形函数创建对象的顺序绘制这些对象，而不考虑对象在三个维度中的关系。此值可产生更快的渲染速度，特别是在图窗非常大时，但同时可能导致显示对象的深度排序不恰当。

颜色图和透明度图

全部折叠

`Colormap` - 颜色图

parula （默认） | 由 RGB 三元组组成的 `m×3` 数组

颜色图，指定为由 RGB（红色、绿色、蓝色）三元组组成的 `m×3` 数组，这些三元组定义 `m` 种单独的颜色。

示例： `ax.Colormap = [1 0 1; 0 0 1; 1 1 0]` 将颜色图设置为三种颜色：品红色、蓝色和黄色。

MATLAB 按行号访问这些颜色。

也可以使用 `colormap` 函数更改颜色图。

`ColorScale` - 颜色图的刻度

`'linear'` （默认） | `'log'`

颜色图的刻度，指定为下列值之一：

- `'linear'` - 线性刻度。颜色栏上的刻度值也使用线性刻度。
- `'log'` - 对数刻度。颜色栏上的刻度值也使用对数刻度。

示例： `ax.ColorScale = 'log'`

`CLim` - 颜色范围

`[0 1]` （默认） | `[cmin cmax]` 形式的二元素向量

坐标区中使用颜色图的对象的颜色范围，指定为 `[cmin cmax]` 形式的二元素向量。此属性用来确定数据值如何映射到颜色图中的颜色，其中：

- `cmin` 指定映射到颜色图中的第一个颜色的数据值。
- `cmax` 指定映射到颜色图中的最后一个颜色的数据值。

`Axes` 对象根据颜色图中 `cmin` 和 `cmax` 所规定的范围进行数据值插值。此范围外的值使用第一个或最后一个颜色（取最接近的值）。

`CLimMode` - `CLim` 的选择模式

`'auto'` （默认） | `'manual'`

`CLim` 属性的选择模式，指定为下列值之一：

- `'auto'` - 根据坐标区包含的图形对象的颜色数据自动选择范围。
- `'manual'` - 手动指定值。要指定值，请设置 `CLim` 属性。当坐标区子级的范围改变时，这些值不会改变。

`Alphamap` - 透明度图

由从 `0` 到 `1` 的 64 个值组成的数组 （默认） | 由从 `0` 到 `1` 的有限 alpha 值组成的数组

透明度图，指定为从 `0` 到 `1` 线性递增的有限 alpha 值组成的数组。数组大小可以是 m×1 或 1×m。MATLAB 按 alpha 值在数组中的索引访问它们。Alphamap 可以为任意长度。

`AlphaScale` - 透明度图的刻度

`'linear'` （默认） | `'log'`

透明度图的刻度，指定为下列值之一：

- `'linear'` - 线性刻度
- `'log'` - 对数刻度

示例： `ax.AlphaScale = 'log'`

`ALim` - alpha 范围

`[0 1]` （默认） | `[amin amax]` 形式的二元素向量

alpha 范围，指定为 `[amin amax]` 形式的二元素向量。此属性影响图形对象（如曲面、图像和补片对象）的 `AlphaData` 值。此属性决定 `AlphaData` 值如何映射到图窗 alpha 映射，其中：

- `amin` 指定映射到图窗 alpha 映射的第一个 alpha 值的数据值。
- `amax` 指定映射到图窗 alpha 映射的最后一个 alpha 值的数据值。

`Axes` 对象根据图窗的 alpha 映射中 `amin` 和 `amax` 所规定的范围进行数据值插值。此范围外的值使用第一个或最后一个 alpha 映射值（取最接近的值）。

图窗的 `Alphamap` 属性包含 alpha 映射。有关详细信息，请参阅 `alpha` 函数。

`ALimMode` - `ALim` 的选择模式

`'auto'` （默认） | `'manual'`

`ALim` 属性的选择模式，指定为下列值之一：

- `'auto'` - 根据坐标区包含的图形对象的 `AlphaData` 值自动选择范围。
- `'manual'` - 手动指定 alpha 范围。要指定 alpha 范围，请设置 `ALim` 属性。

框样式

全部折叠

`Color` - 背景色

`[1 1 1]` （默认） | RGB 三元组 | 十六进制颜色代码 | `'r'` | `'g'` | `'b'` | ...

背景颜色，指定为 RGB 三元组、十六进制颜色代码、颜色名称或短名称。

对于自定义颜色，请指定 RGB 三元组或十六进制颜色代码。

- RGB 三元组是包含三个元素的行向量，其元素分别指定颜色中红、绿、蓝分量的强度。强度值必须位于 `[0,1]` 范围内，例如 `[0.4 0.6 0.7]`。
- 十六进制颜色代码是字符向量或字符串标量，以井号 (`#`) 开头，后跟三个或六个十六进制数字，范围可以是 `0` 到 `F`。这些值不区分大小写。因此，颜色代码 `'#FF8800'` 与 `'#ff8800'`、`'#F80'` 与 `'#f80'` 是等效的。

此外，还可以按名称指定一些常见的颜色。下表列出了命名颜色选项、等效 RGB 三元组和十六进制颜色代码。

颜色名称	短名称	RGB 三元组	十六进制颜色代码	外观
`'red'`	`'r'`	`[1 0 0]`	`'#FF0000'`	
`'green'`	`'g'`	`[0 1 0]`	`'#00FF00'`	
`'blue'`	`'b'`	`[0 0 1]`	`'#0000FF'`	
`'cyan'`	`'c'`	`[0 1 1]`	`'#00FFFF'`	

颜色名称	短名称	RGB 三元组	十六进制颜色代码	外观
'magenta'	'm'	[1 0 1]	'#FF00FF'	
'yellow'	'y'	[1 1 0]	'#FFFF00'	
'black'	'k'	[0 0 0]	'#000000'	
'white'	'w'	[1 1 1]	'#FFFFFF'	
'none'	不适用	不适用	不适用	无颜色

以下是 MATLAB 在许多类型的绘图中使用的默认颜色的 RGB 三元组和十六进制颜色代码。

RGB 三元组	十六进制颜色代码	外观
[0 0.4470 0.7410]	'#0072BD'	
[0.8500 0.3250 0.0980]	'#D95319'	
[0.9290 0.6940 0.1250]	'#EDB120'	
[0.4940 0.1840 0.5560]	'#7E2F8E'	
[0.4660 0.6740 0.1880]	'#77AC30'	
[0.3010 0.7450 0.9330]	'#4DBEEE'	
[0.6350 0.0780 0.1840]	'#A2142F'	

示例： `ax.Color = [0 0 1];`

示例： `ax.Color = 'b';`

示例： `ax.Color = 'blue';`

示例： `ax.Color = '#0000FF';`

`LineWidth` - 线条宽度

`0.5` （默认） | 正数值

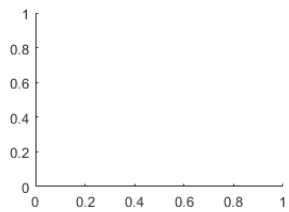
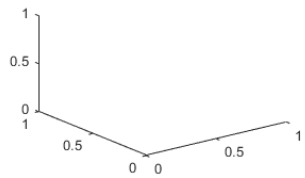
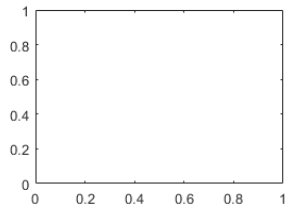
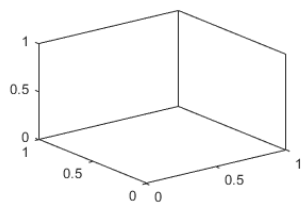
坐标区轮廓、刻度线和网格线的线宽，指定为正数值（以磅为单位）。一磅等于 1/72 英寸。

示例： `ax.LineWidth = 1.5`

`Box` - 框轮廓

`'off'` （默认） | `'on'`

框轮廓，指定为 `'off'` 或 `'on'`。

值	说明	二维结果	三维结果
'off'	不显示坐标区周围的框轮廓。 **示例：** `ax.Box = 'off'`		
'on'	显示坐标区周围的框轮廓。对于三维视图，可以使用 `BoxStyle` 属性更改轮廓的范围。 **示例：** `ax.Box = 'on'`		

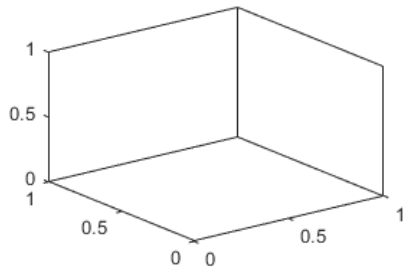
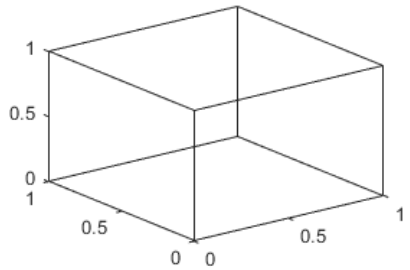
`XColor`、`YColor` 和 `ZColor` 属性控制轮廓的颜色。

示例： `ax.Box = 'on'`

`BoxStyle` - 框轮廓样式

`'back'` （默认） | `'full'`

框轮廓样式，指定为 `'back'` 或 `'full'`。此属性仅影响三维视图。

值	说明	结果
`'back'`	画出三维框的背板轮廓。 **示例：** `ax.BoxStyle = 'back'`	
`'full'`	画出整个三维框的轮廓。 **示例：** `ax.BoxStyle = 'full'`	

`Clipping` - 在坐标区范围内裁剪对象

`'on'` （默认） | `'off'`

按照坐标区范围裁剪对象，指定为 `'on'` 或 `'off'`。`Axes` 对象内某个对象的裁剪行为由 `Axes` 对象的 `Clipping` 属性和该具体对象的 `Clipping` 属性共同决定。`Axes` 对象的属性值具有以下作用：

- `'on'` - 使坐标区内的每个对象基于其 `Clipping` 属性值控制自身的裁剪行为。
- `'off'` - 禁用坐标区内所有对象的裁剪，而不管各个对象的 `Clipping` 属性值如何设置。对象的某些部分可能会显示在坐标区范围之外。例如，如果创建一个绘图，使用 `hold on` 命令冻结轴缩放，然后添加一个大于原始绘图的绘图，则绘图的某些部分可能会显示在范围之外。

下表列出了 `Clipping` 属性值的不同组合结果。

坐标区对象的裁剪属性	单个对象的裁剪属性	结果
`'on'`	`'on'`	裁剪单个对象。其他对象可能会被裁剪，也可能不会。
`'on'`	`'off'`	不裁剪单个对象。其他对象可能会被裁剪，也可能不会。
`'off'`	`'on'`	所有对象都不裁剪。
`'off'`	`'off'`	所有对象都不裁剪。

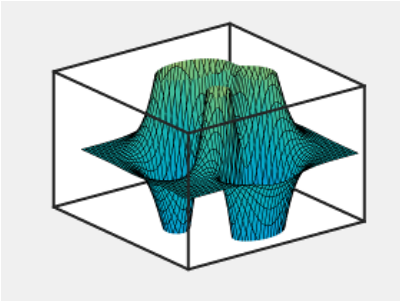
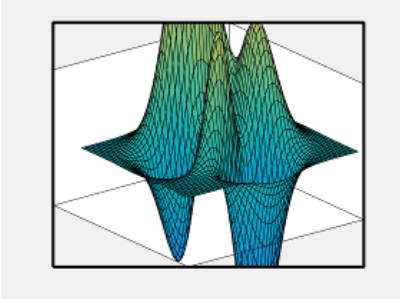
`ClippingStyle` - 裁剪边界

`'3dbox'` （默认） | `'rectangle'`

裁剪边界，指定为下表中的值之一。如果绘图包含标记，则只要数据点位于坐标区范围内，MATLAB 就会绘制整个标记。

如果 `Clipping` 属性设置为 `'off'`，则 `ClippingStyle` 属性无效。

值	说明	边界区域图示

值	说明	边界区域图示
`'3dbox'`	按照坐标轴范围定义的坐标区框的六条边对绘制对象进行裁剪。 粗线可能会显示在坐标区范围之外。	
`'rectangle'`	在任何给定视图中按照包围坐标区的矩形边界对绘制对象进行裁剪。 在坐标区范围处裁剪粗线。	

`AmbientLightColor` - 背景光源颜色

`[1 1 1]` （默认） | RGB 三元组 | 十六进制颜色代码 | `'r'` | `'g'` | `'b'` | ...

背景光颜色，指定为 RGB 三元组、十六进制颜色代码、颜色名称或短名称。背景光是一种无向光，均匀地照射在坐标区内的所有对象上。要添加光源，请使用 `light` 函数。

对于自定义颜色，请指定 RGB 三元组或十六进制颜色代码。

- RGB 三元组是包含三个元素的行向量，其元素分别指定颜色中红、绿、蓝分量的强度。强度值必须位于 `[0,1]` 范围内，例如 `[0.4 0.6 0.7]`。
- 十六进制颜色代码是字符向量或字符串标量，以井号 (`#`) 开头，后跟三个或六个十六进制数字，范围可以是 `0` 到 `F`。这些值不区分大小写。因此，颜色代码 `'#FF8800'` 与 `'#ff8800'`、`'#F80'` 与 `'#f80'` 是等效的。

此外，还可以按名称指定一些常见的颜色。下表列出了命名颜色选项、等效 RGB 三元组和十六进制颜色代码。

颜色名称	短名称	RGB 三元组	十六进制颜色代码	外观
`'red'`	`'r'`	`[1 0 0]`	`'#FF0000'`	
`'green'`	`'g'`	`[0 1 0]`	`'#00FF00'`	
`'blue'`	`'b'`	`[0 0 1]`	`'#0000FF'`	
`'cyan'`	`'c'`	`[0 1 1]`	`'#00FFFF'`	
`'magenta'`	`'m'`	`[1 0 1]`	`'#FF00FF'`	
`'yellow'`	`'y'`	`[1 1 0]`	`'#FFFF00'`	
`'black'`	`'k'`	`[0 0 0]`	`'#000000'`	
`'white'`	`'w'`	`[1 1 1]`	`'#FFFFFF'`	
`'none'`	不适用	不适用	不适用	无颜色

以下是 MATLAB 在许多类型的绘图中使用的默认颜色的 RGB 三元组和十六进制颜色代码。

RGB 三元组	十六进制颜色代码	外观
`[0 0.4470 0.7410]`	`'#0072BD'`	
`[0.8500 0.3250 0.0980]`	`'#D95319'`	
`[0.9290 0.6940 0.1250]`	`'#EDB120'`	
`[0.4940 0.1840 0.5560]`	`'#7E2F8E'`	
`[0.4660 0.6740 0.1880]`	`'#77AC30'`	
`[0.3010 0.7450 0.9330]`	`'#4DBEEE'`	

RGB 三元组	十六进制颜色代码	外观
[0.6350 0.0780 0.1840]	'#A2142F'	

示例： `ax.AmbientLightColor = [1 0 1]`

示例： `ax.AmbientLightColor = 'm'`

示例： `ax.AmbientLightColor = 'magenta'`

示例： `ax.AmbientLightColor = '#FF00FF'`

位置

全部折叠

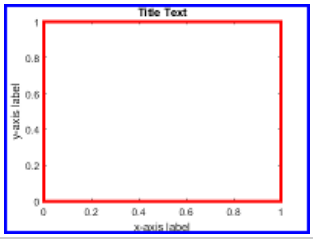
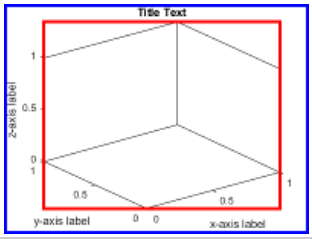
`OuterPosition` - 大小和位置，包括标签和边距

`[0 0 1 1]` （默认） | `[left bottom width height]` 形式的四元素向量

大小和位置，包括标签和边距，指定为 `[left bottom width height]` 形式的四元素向量。默认情况下，MATLAB 按照归一化的容器单位来测量值。要更改单位，请设置 `Units` 属性。`[0 0 1 1]` 的默认值包括容器的整个内部。

- `left` 和 `bottom` 元素定义从容器（通常为图窗、面板或选项卡）左下角到外部位置边界左下角之间的距离。
- `width` 和 `height` 元素是外边界尺寸。

下图显示由 `OuterPosition` 值（蓝色）和 `Position` 值（红色）定义的区域。

二维坐标区视图	三维坐标区视图
	

有关坐标区位置的详细信息，请参阅控制坐标区布局。

> **注意**
>
> 当父容器为 `TiledChartLayout` 时，设置此属性不起作用。

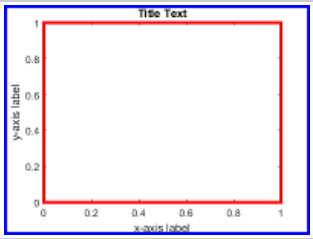
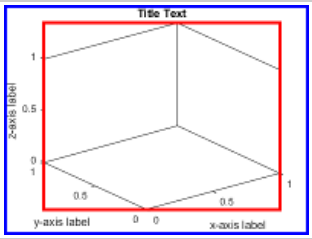
`Position` - 大小和位置，不包括标签边距

`[0.1300 0.1100 0.7750 0.8150]` （默认） | `[left bottom width height]` 形式的四元素向量

大小和位置，不包括标签边距，指定为 `[left bottom width height]` 形式的四元素向量。默认情况下，MATLAB 按照归一化的容器单位来测量值。要更改单位，请设置 `Units` 属性。

- `left` 和 `bottom` 元素定义从容器（通常为图窗、面板或选项卡）左下角到位置边界左下角之间的距离。
- `width` 和 `height` 元素是位置边界维度。对于三维视图中的坐标区，`Position` 属性是包围坐标区的最小矩形。

如果要指定位置并考虑让文本围绕坐标区，请改为设置 `OuterPosition` 属性。下图显示由 `OuterPosition` 值（蓝色）和 `Position` 值（红色）定义的区域。

二维坐标区视图	三维坐标区视图
	

有关坐标区位置的详细信息，请参阅控制坐标区布局。

> **注意**
>
> 当父容器为 `TiledChartLayout` 时，设置此属性不起作用。

`TightInset` - 文本标签的边距

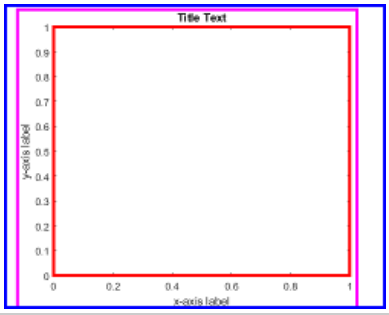
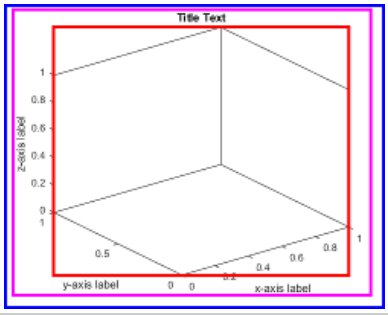
`[left bottom right top]` 形式的四元素向量

此属性为只读

文本标签的边距，指定为 `[left bottom right top]` 形式的四元素向量。默认情况下，MATLAB 按照归一化的容器单位来测量值。要更改单位，请设置 `Units` 属性。

这些元素定义 `Position` 属性的边界与周围文本之间的距离。将 `Position` 值与 `TightInset` 值结合使用可定义包围坐标区和周围文本的最紧凑的边界框。

下面的图显示由 `OuterPosition` 值（蓝色）、`Position` 值（红色）以及由 `TightInset` 值（品红色）外扩的 `Position` 定义的区域。

二维坐标区视图	三维坐标区视图
	

有关详细信息，请参阅控制坐标区布局。

`ActivePositionProperty` - 活动位置属性

`'outerposition'` | `'position'`

在增加、删除或更改装饰元素时保持不变的位置属性，指定为下列值之一：

- `'outerposition'` - 在添加、删除或更改装饰元素（例如标题或轴标签）时，`OuterPosition` 属性保持不变。如需任何位置调整，MATLAB 会调整 `Position` 属性。
- `'position'` - 在添加、删除或更改装饰元素（例如标题或轴标签）时，`Position` 属性保持不变。如需任何位置调整，MATLAB 会调整 `OuterPosition` 属性。

> **注意**
>
> 当父容器为 `TiledChartLayout` 时，设置此属性不起作用。

`Units` - 位置单位

`'normalized'` （默认） | `'inches'` | `'centimeters'` | `'points'` | `'pixels'` | `'characters'`

位置单位，指定为下列值之一：

Units	说明
`'normalized'` （默认值）	根据容器进行归一化，容器通常是图窗或面板。容器的左下角映射到 `(0,0)`，右上角映射到 `(1,1)`。
`'inches'`	英寸。
`'centimeters'`	厘米。
`'characters'`	基于图形根对象的默认 uicontrol 字体： • 字符宽度 = 字母 x 的宽度。 • 字符高度 = 两个文本行的基线之间的距离。
`'points'`	字体磅数。一磅等于 1/72 英寸。

Units	说明
'pixels'	像素。 从 R2015b 开始，以像素为单位的距离不再依赖 Windows 和 Macintosh 系统上的系统分辨率。 • 在 Windows 系统上，一个像素是 1/96 英寸。 • 在 Macintosh 系统上，一个像素是 1/72 英寸。 • 在 Linux 系统上，一个像素的大小由系统分辨率确定。

在对象创建过程中将单位指定为 Name,Value 对组时，您必须先设置 Units 属性，然后再指定要使用这些单位的属性（如 Position）。

DataAspectRatio - 数据单元的相对长度

[1 1 1] （默认） | [dx dy dz] 形式的三元素向量

数据单位沿每个坐标轴的相对长度，指定为 [dx dy dz] 形式的三元素向量。此向量定义相对的 *x*、*y* 和 *z* 数据缩放因子。例如，若将该属性指定为 [1 2 1]，会将 x 方向中的一个数据单位长度设置为等同于 y 方向中的两个数据单位长度和 z 方向中的一个数据单位长度。

也可以使用 daspect 函数更改数据纵横比。

示例： ax.DataAspectRatio = [1 1 1]

数据类型： single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

DataAspectRatioMode - 数据纵横比模式

'auto' （默认） | 'manual'

数据纵横比模式，指定为下列值之一：

- 'auto' - 自动选择能够充分利用可用空间的值。如果 PlotBoxAspectRatioMode 和 CameraViewAngleMode 也设置为 'auto'，将启用“伸展填充”行为。伸展坐标区，使其填满 Position 属性定义的可用空间。
- 'manual' - 禁用“伸展填充”行为并使用手动指定的数据纵横比。要指定值，请设置 DataAspectRatio 属性。

PlotBoxAspectRatio - 每个坐标轴的相对长度

[1 1 1] （默认） | [px py pz] 形式的三元素向量

每个坐标轴的相对长度，指定为 [px py pz] 形式的三元素向量，三个元素分别定义 *x* 轴、*y* 轴和 *z* 轴的相对缩放因子。图框是包含坐标轴范围定义的轴数据区域的框。

也可以使用 pbaspect 函数更改数据纵横比。

如果您指定了坐标轴范围、数据纵横比和图框纵横比，MATLAB 将忽略图框纵横比，而遵守坐标轴范围和数据纵横比。

示例： ax.PlotBoxAspectRatio = [1 0.75 0.75]

数据类型： single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

PlotBoxAspectRatioMode - PlotBoxAspectRatio 的选择模式

'auto' （默认） | 'manual'

PlotBoxAspectRatio 属性的选择模式，指定为下列值之一：

- 'auto' - 自动选择能够充分利用可用空间的值。如果 DataAspectRatioMode 和 CameraViewAngleMode 也设置为 'auto'，将启用“伸展填充”行为。伸展 Axes 对象，使其填满 Position 属性定义的可用空间。
- 'manual' - 禁用“伸展填充”行为并使用手动指定的图框纵横比。要指定值，请设置 PlotBoxAspectRatio 属性。

Layout - 布局选项

空 LayoutOptions 数组 （默认） | TiledChartLayoutOptions 对象

布局选项，指定为 TiledChartLayoutOptions 对象。此属性指定当坐标区是分块图布局的子级时的选项。通过对 TiledChartLayoutOptions 对象设置 Tile 和 TileSpan 属性，使用此属性在布局内定位坐标区。

例如，以下代码将坐标区 ax 放置在分块图布局的第三个图块中。

```
ax.Layout.Tile = 3;
```

要使坐标区占据多个图块，请将 TileSpan 属性指定为二元素向量。例如，此坐标区占据了 2 行和 3 列的图块。

```
ax.Layout.TileSpan = [2 3];
```

如果此坐标区不是分块图布局的子级（例如，如果它是图窗或面板的子级），则此属性为空且不起作用。

视图

全部折叠

View - 视图的方位角和仰角

[0 90] （默认）| [azimuth elevation] 形式的二元素向量

视图的方位角和仰角，指定为以度为单位定义的 [azimuth elevation] 形式的二元素向量。也可以使用 view 函数设置视图。

示例： ax.View = [45 45]

Projection - 二维屏幕上的投影类型

'orthographic' （默认）| 'perspective'

二维屏幕上的投影类型，指定为下列值之一：

- 'orthographic' - 保持图形对象的正确相对维度（就给定点到观察者之间的距离而言），并在屏幕上根据平行数据绘制平行线条。
- 'perspective' - 使用前缩透视法，这可以让您在三维对象的二维表示形式中表现景深。透视投影不会保留对象的相对维度。它改为显示较远的线段短于较近的等长线段。按数据平行的线条在屏幕上可能显示为不平行。

CameraPosition - 照相机位置

[x y z] 形式的三元素向量

照相机位置或视点，指定为 [x y z] 形式的三元素向量。此向量定义照相机位置（即您观察坐标区的点）的坐标区坐标。照相机沿视图轴指定方向，该轴是一条连接照相机位置和照相机目标的直线。有关说明，请参阅照相机图形术语。

如果 Projection 属性设置为 'perspective'，则当您更改 CameraPosition 设置时，透视量也会更改。

也可以使用 campos 函数设置照相机位置。

示例： ax.CameraPosition = [0.5 0.5 9]

数据类型： single | double

CameraPositionMode - CameraPosition 的选择模式

'auto' （默认）| 'manual'

CameraPosition 属性的选择模式，指定为下列值之一：

- 'auto' - 沿观察轴自动设置 CameraPosition。计算位置，使照相机沿当前视图（view 函数返回的视图）指定的方位角和仰角与目标保持固定的距离。诸如 rotate3d、zoom 和 pan 之类的函数会将此模式更改为 'auto' 以执行其操作。
- 'manual' - 手动指定值。要指定值，请设置 CameraPosition 属性。

CameraTarget - 照相机目标点

[x y z] 形式的三元素向量

照相机目标点，指定为 [x y z] 形式的三元素向量。此向量定义点的轴坐标。照相机沿视图轴指定方向，该轴是一条连接照相机位置和照相机目标的直线。有关说明，请参阅照相机图形术语。

也可以使用 camtarget 函数设置照相机目标。

示例： ax.CameraTarget = [0.5 0.5 0.5]

数据类型： single | double

CameraTargetMode - CameraTarget 的选择模式
`'auto'` （默认） | `'manual'`

`CameraTarget` 属性的选择模式，指定为下列值之一：

- `'auto'` - 将照相机目标置于轴图框矩心。
- `'manual'` - 使用手动指定的照相机目标值。要指定值，请设置 `CameraTarget` 属性。

CameraUpVector - 定义向上方向的向量
`[x y z]` 形式的三元素方向向量

定义向上方向的向量，指定为 `[x y z]` 形式的三元素方向向量。对于二维视图，默认值为 `[0 1 0]`。对于三维视图，默认值为 `[0 0 1]`。有关说明，请参阅照相机图形术语。

也可以使用 `camup` 函数设置向上的方向。

示例： `ax.CameraUpVector = [sin(45) cos(45) 1]`

CameraUpVectorMode - CameraUpVector 的选择模式
`'auto'` （默认） | `'manual'`

`CameraUpVector` 属性的选择模式，指定为下列值之一：

- `'auto'` - 对于三维视图，自动将值设置为 `[0 0 1]`，使 z 轴正方向朝上。对于二维视图，将值设置为 `[0 1 0]`，使 y 轴正方向朝上。
- `'manual'` - 手动指定定义向上方向的向量。要指定值，请设置 `CameraUpVector` 属性。

CameraViewAngle - 视野
`6.6086` （默认） | 范围 [0,180) 中的标量角

视野，指定为大于 0 且小于或等于 180 的标量角。更改照相机视角会影响坐标区中显示的图形对象的大小，但不会影响透视变形度。角度越大，视野越大，而场景中显示的对象就越小。有关说明，请参阅照相机图形术语。

示例： `ax.CameraViewAngle = 15`

数据类型： `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64` | `logical`

CameraViewAngleMode - CameraViewAngle 的选择模式
`'auto'` （默认） | `'manual'`

`CameraViewAngle` 属性的选择模式，指定为下列值之一：

- `'auto'` - 自动选择能够捕获整个场景的最小角度作为视野（最高可达 180 度）。
- `'manual'` - 手动指定视野。要指定值，请设置 `CameraViewAngle` 属性。

交互性

全部展开

Toolbar - 数据探查工具栏
`AxesToolbar` 对象 （默认）

Interactions - 交互
由交互对象组成的数组 | `[]`

Visible - 可见性状态
`'on'` （默认） | `'off'`

CurrentPoint - 鼠标指针的位置
2×3 数组

UIContextMenu - 上下文菜单
空 GraphicsPlaceholder 数组 （默认） | ContextMenu 对象

Selected - 选择状态
'off' （默认） | 'on'

SelectionHighlight - 是否显示选择句柄
'on' （默认） | 'off'

回调

全部展开

ButtonDownFcn - 鼠标点击回调
'' （默认） | 函数句柄 | 元胞数组 | 字符向量

CreateFcn - 创建函数
'' （默认） | 函数句柄 | 元胞数组 | 字符向量

DeleteFcn - 删除函数
'' （默认） | 函数句柄 | 元胞数组 | 字符向量

回调执行控件

全部展开

Interruptible - 回调中断
'on' （默认） | 'off'

BusyAction - 回调排队
'queue' （默认） | 'cancel'

PickableParts - 捕获鼠标点击的能力
'visible' （默认） | 'all' | 'none'

HitTest - 响应捕获的鼠标点击
'on' （默认） | 'off'

BeingDeleted - 删除状态
'off' | 'on'

父级/子级

全部展开

Parent - 父容器
Figure 对象 | Panel 对象 | Tab 对象 | TiledChartLayout 对象

Children - 子级
空 GraphicsPlaceholder 数组 | 图形对象的数组

HandleVisibility - 对象句柄的可见性
'on' （默认） | 'off' | 'callback'

标识符

全部展开

> **`Type` - 图形对象的类型**
> `'axes'`

> **`Tag` - 对象标识符**
> `''` （默认） | 字符向量 | 字符串标量

> **`UserData` - 用户数据**
> `[]` （默认） | 数组

兼容性考虑

全部展开

> **更改坐标区的 ColorOrder 或 LineStyleOrder 会立即影响现有绘图**
> *R2019b 中的行为有变化*

> **ColorOrder 和 LineStyleOrder 的索引方案可能会更改绘图的颜色和线型。**
> *R2019b 中的行为有变化*

另请参阅

`axes` | `axis` | `box` | `caxis` | `cla` | `gca` | `grid`

主题

访问属性值
图形对象属性

在 R2006a 之前推出