**Functions for Reading, Writing, and Displaying Images**

* Images are essentially two-dimensional matrices, so many MATLAB functions can operate on and display images. The following lists the most useful ones. The sections that follow describe these functions in more detail.

| **Function** | **Purpose** | **Function Group** |
| --- | --- | --- |
| [axis](http://in.mathworks.com/help/matlab/ref/axis.html) | Plot axis scaling and appearance. | Display |
| [image](http://in.mathworks.com/help/matlab/ref/image.html) | Display image (create image object). | Display |
| [imagesc](http://in.mathworks.com/help/matlab/ref/imagesc.html) | Scale data and display as image. | Display |
| [imread](http://in.mathworks.com/help/matlab/ref/imread.html) | Read image from graphics file. | File I/O |
| [imwrite](http://in.mathworks.com/help/matlab/ref/imwrite.html) | Write image to graphics file. | File I/O |
| [imfinfo](http://in.mathworks.com/help/matlab/ref/imfinfo.html) | Get image information from graphics file. | Utility |
| [ind2rgb](http://in.mathworks.com/help/matlab/ref/ind2rgb.html) | Convert indexed image to RGB image. | Utility |

# axis

Set axis limits and aspect ratios

## Syntax

* axis(limits)
* axis style
* axis mode
* axis y direction
* axis visibility
* lim = axis
* [m,v,d] = axis('state')
* **\_\_\_** = axis(ax,**\_\_\_**)

## Description

* axis ([limits](http://in.mathworks.com/help/matlab/ref/axis.html#inputarg_limits)) specifies the limits for the current axes. Specify the limits as vector of four, six, or eight elements.
* axis [style](http://in.mathworks.com/help/matlab/ref/axis.html#inputarg_style) uses a predefined style to set the limits and scaling. For example, specify the style as equal to use equal data unit lengths along each axis.
* axis [mode](http://in.mathworks.com/help/matlab/ref/axis.html#inputarg_mode) sets whether the axes automatically chooses the limits or not. Specify the mode as manual, auto, or one of the semiautomatic options, for example, 'auto x'.
* axis [ydirection](http://in.mathworks.com/help/matlab/ref/axis.html#inputarg_ydirection) controls the placement of the coordinate system origin and the direction of increasing y values. Set ydirection to ij to place the origin at the upper left corner of the axes. The y values increase from top to bottom. Set y direction to xy to place the origin at the lower left corner. The y values increase from bottom to top. This is the default value.
* axis [visibility](http://in.mathworks.com/help/matlab/ref/axis.html#inputarg_visibility) controls the visibility of the axes or polar axes background. Set visibility to off to turn off the display of the axes background. Plots in the axes still display. Set visibility to on to display the axes background. This is the default value
* [lim](http://in.mathworks.com/help/matlab/ref/axis.html#outputarg_lim) = axis returns the x-axis and y-axis limits for the current axes. For 3-D axes, it also returns the z-axis limits. For polar axes, it returns the theta-axis and r-axis limits.
* [m,v,d] = axis('state') returns the current settings for the axis limit selection, the axes visibility, and the y-axis direction.

|  |
| --- |
|  |

* **\_\_\_** = axis([ax](http://in.mathworks.com/help/matlab/ref/axis.html" \l "inputarg_ax),**\_\_\_**) uses the axes or polar axes specified by ax instead of the current axes. Specify ax as the first input argument. You can include additional input or output arguments only if the original syntax supported them. Use single quotes around input arguments that are character vectors, for example, axis(ax,'equal').

## Examples

Create a figure with two subplots. Plot a sine wave in each subplot. Then, set the axis limits for the subplots to the same values.

x1 = linspace(0,10,100);

y1 = sin(x1);

ax1 = subplot(2,1,1);

plot(ax1,x1,y1)

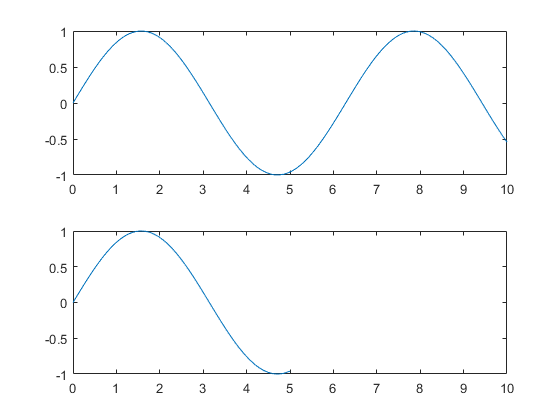
x2 = linspace(0,5,100);

y2 = sin(x2);

ax2 = subplot(2,1,2);

plot(ax2,x2,y2)

axis([ax1 ax2],[0 10 -1 1])



# image

Display image from array

## Syntax

* image(C)
* image(x,y,C)
* image('CData',C)
* image('XData',x,'YData',y,'CData',C)
* image(**\_\_\_**,Name,Value)
* image(ax,**\_\_\_**)
* im = image(**\_\_\_**)

## Description

* image([C](http://in.mathworks.com/help/matlab/ref/image.html#inputarg_C)) displays the data in array C as an image. Each element of C specifies the color for 1 pixel of the image. The resulting image is an m-by-n grid of pixels where m is the number of columns and n is the number of rows in C. The row and column indices of the elements determine the centers of the corresponding pixels.
* image([x](http://in.mathworks.com/help/matlab/ref/image.html#inputarg_x),[y](http://in.mathworks.com/help/matlab/ref/image.html#inputarg_y),[C](http://in.mathworks.com/help/matlab/ref/image.html#inputarg_C)) specifies the image location. Use x and y to specify the locations of the corners corresponding to C(1,1) and C(m,n). To specify both corners, set x and y as two-element vectors. To specify the first corner and let image determine the other, set x and y as scalar values. The image is stretched and oriented as applicable.
* image ('CData',[C](http://in.mathworks.com/help/matlab/ref/image.html#inputarg_C)) adds the image to the current axes without replacing existing plots. This syntax is the low-level version of image(C). For more information, see [High-Level Versus Low-Level Version of Image](http://in.mathworks.com/help/matlab/ref/image.html#buryijq).
* image ('XData',[x](http://in.mathworks.com/help/matlab/ref/image.html#inputarg_x),'YData',[y](http://in.mathworks.com/help/matlab/ref/image.html#inputarg_y),'CData',[C](http://in.mathworks.com/help/matlab/ref/image.html#inputarg_C)) specifies the image location. This syntax is the low-level version of image(x,y,C).
* image (**\_\_\_**,[Name,Value](http://in.mathworks.com/help/matlab/ref/image.html#namevaluepairarguments)) specifies image properties using one or more name-value pair arguments. You can specify image properties with any of the input argument combinations in the previous syntaxes.
* image ([ax](http://in.mathworks.com/help/matlab/ref/image.html#inputarg_ax),**\_\_\_**) creates the image in the axes specified by ax instead of in the current axes (gca). The option ax can precede any of the input argument combinations in the previous syntaxes.
* im = image(**\_\_\_**) returns the image object created. Use im to set properties of the image after it is created. You can specify this output with any of the input argument combinations in the previous syntaxes. For a list of image properties and descriptions, see [Image Properties](http://in.mathworks.com/help/matlab/ref/image-properties.html).

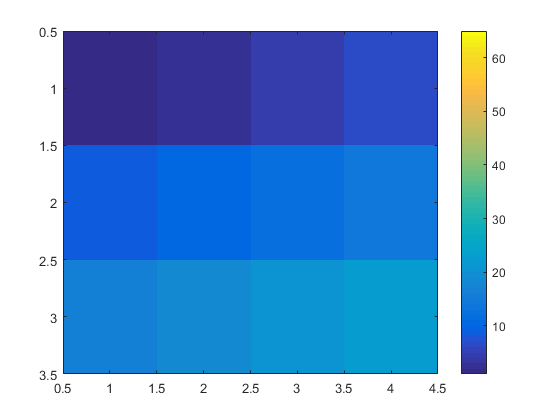
## Examples

Create matrix C. Display an image of the data in C. Add a colorbar to the graph to show the current color map.

C = [0 2 4 6; 8 10 12 14; 16 18 20 22];

image(C)

colorbar



# imagesc

Display image with scaled colors

## Syntax

* imagesc(C)
* imagesc(x,y,C)
* imagesc(**\_\_\_**,clims)
* imagesc('CData',C)
* imagesc('XData',x,'YData',y,'CData',C)
* imagesc(**\_\_\_**,Name,Value)
* imagesc(ax,**\_\_\_**)
* im = imagesc(**\_\_\_**)

## Description

* imagesc([C](http://in.mathworks.com/help/matlab/ref/imagesc.html#inputarg_C)) displays the data in array C as an image that uses the full range of colors in the colormap. Each element of C specifies the color for 1 pixel of the image. The resulting image is an m-by-n grid of pixels where m is the number of columns and n is the number of rows in C. The row and column indices of the elements determine the centers of the corresponding pixels.
* imagesc([x](http://in.mathworks.com/help/matlab/ref/imagesc.html#inputarg_x),[y](http://in.mathworks.com/help/matlab/ref/imagesc.html#inputarg_y),[C](http://in.mathworks.com/help/matlab/ref/imagesc.html#inputarg_C)) specifies the image location. Use x and y to specify the locations of the corners corresponding to C(1,1) and C(m,n). To specify both corners, set x and y as two-element vectors. To specify the first corner and let imagesc determine the other, set x and y as scalar values. The image is stretched and oriented as applicable.
* imagesc(**\_\_\_**,[clims](http://in.mathworks.com/help/matlab/ref/imagesc.html#inputarg_clims)) specifies the data values that map to the first and last elements of the colormap. Specify clims as a two-element vector of the form [cmin cmax], where values less than or equal to cmin map to the first color in the colormap and values greater than or equal to cmax map to the last color in the colormap.
* imagesc('CData',[C](http://in.mathworks.com/help/matlab/ref/imagesc.html#inputarg_C)) adds the image to the current axes without replacing existing plots. This syntax is the low-level version of imagesc(C). For more information, see [High-Level Versus Low-Level Version](http://in.mathworks.com/help/matlab/ref/imagesc.html#bux1nfb-2).
* imagesc('XData',[x](http://in.mathworks.com/help/matlab/ref/imagesc.html#inputarg_x),'YData',[y](http://in.mathworks.com/help/matlab/ref/imagesc.html#inputarg_y),'CData',[C](http://in.mathworks.com/help/matlab/ref/imagesc.html#inputarg_C)) specifies the image location. This syntax is the low-level version of imagesc(x,y,C).
* imagesc(**\_\_\_**,[Name,Value](http://in.mathworks.com/help/matlab/ref/imagesc.html#namevaluepairarguments)) specifies image properties using one or more name-value pair arguments. You can specify image properties with any of the input argument combinations in the previous syntaxes. For a list of image properties and descriptions, see [Image Properties](http://in.mathworks.com/help/matlab/ref/image-properties.html).
* imagesc([ax](http://in.mathworks.com/help/matlab/ref/imagesc.html" \l "inputarg_ax),**\_\_\_**) creates the image in the axes specified by ax instead of in the current axes (gca). The option ax can precede any of the input argument combinations in the previous syntaxes.
* [im](http://in.mathworks.com/help/matlab/ref/imagesc.html#outputarg_im) = imagesc(**\_\_\_**) returns the image object created. Use im to set properties of the image after it is created. You can specify this output with any of the input argument combinations in the previous syntaxes.

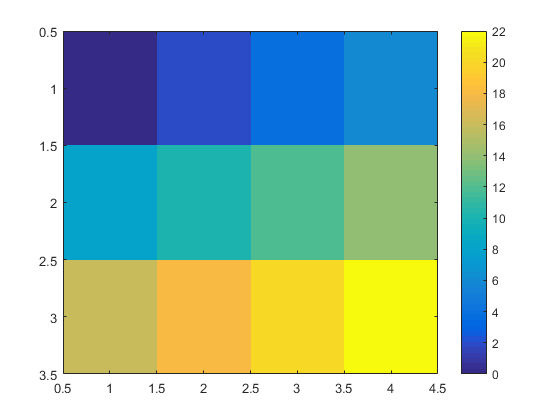
## Examples

Create matrix C. Display an image of the data in C. Add a colorbar to the graph to show the current colormap. By default, imagesc scales the color limits so that image uses the full range of the colormap, where the smallest value in C maps to the first color in the colormap and the largest value maps to the last color.

C = [0 2 4 6; 8 10 12 14; 16 18 20 22];

imagesc(C)

colorbar



# imread

Read image from graphics file

## Syntax

* A = imread(filename)
* A = imread(filename,fmt)
* A = imread(**\_\_\_**,idx)
* A = imread(**\_\_\_**,Name,Value)
* [A,map] = imread(**\_\_\_**)
* [A,map,transparency] = imread(**\_\_\_**)

## Description

* [A](http://in.mathworks.com/help/matlab/ref/imread.html#outputarg_A) = imread([filename](http://in.mathworks.com/help/matlab/ref/imread.html" \l "inputarg_filename)) reads the image from the file specified by filename, inferring the format of the file from its contents. If filename is a multi-image file, then imread reads the first image in the file.
* [A](http://in.mathworks.com/help/matlab/ref/imread.html#outputarg_A) = imread([filename](http://in.mathworks.com/help/matlab/ref/imread.html" \l "inputarg_filename),[fmt](http://in.mathworks.com/help/matlab/ref/imread.html#inputarg_fmt)) additionally specifies the format of the file with the standard file extension indicated by fmt. If imread cannot find a file with the name specified by filename, it looks for a file named *filename.fmt*.
* [A](http://in.mathworks.com/help/matlab/ref/imread.html#outputarg_A) = imread(**\_\_\_**,[idx](http://in.mathworks.com/help/matlab/ref/imread.html#inputarg_idx)) reads the specified image or images from a multi-image file. This syntax applies only to GIF, CUR, ICO, and HDF4 files. You must specify a [filename](http://in.mathworks.com/help/matlab/ref/imread.html#inputarg_filename) input, and you can optionally specify [fmt](http://in.mathworks.com/help/matlab/ref/imread.html#inputarg_fmt).
* [A](http://in.mathworks.com/help/matlab/ref/imread.html#outputarg_A) = imread(**\_\_\_**,[Name,Value](http://in.mathworks.com/help/matlab/ref/imread.html#namevaluepairarguments)) specifies format-specific options using one or more name-value pair arguments, in addition to any of the input arguments in the previous syntaxes.
* [[A](http://in.mathworks.com/help/matlab/ref/imread.html#outputarg_A),[map](http://in.mathworks.com/help/matlab/ref/imread.html#outputarg_map)] = imread(**\_\_\_**) reads the indexed image in filename into A and reads its associated colormap into map. Colormap values in the image file are automatically rescaled into the range [0,1].
* [[A](http://in.mathworks.com/help/matlab/ref/imread.html#outputarg_A),[map](http://in.mathworks.com/help/matlab/ref/imread.html#outputarg_map),[transparency](http://in.mathworks.com/help/matlab/ref/imread.html#outputarg_transparency)] = imread(**\_\_\_**) additionally returns the image transparency. This syntax applies only to PNG, CUR, and ICO files. For PNG files, transparency is the alpha channel, if one is present. For CUR and ICO files, it is the AND (opacity) mask.

## Examples

A = imread('ngc6543a.jpg');

# imwrite

Write image to graphics file

## Syntax

* imwrite(A,filename)
* imwrite(A,map,filename)
* imwrite(**\_\_\_**,fmt)
* imwrite(**\_\_\_**,Name,Value)

## Description

imwrite([A](http://in.mathworks.com/help/matlab/ref/imwrite.html" \l "inputarg_A),[filename](http://in.mathworks.com/help/matlab/ref/imwrite.html#inputarg_filename)) writes image data A to the file specified by filename, inferring the file format from the extension. imwrite creates the new file in your current folder. The bit depth of the output image depends on the data type of A and the file format. For most formats:

* If A is of data type uint8, then imwrite outputs 8-bit values.
* If A is of data type uint16 and the output file format supports 16-bit data (JPEG, PNG, and TIFF), then imwrite outputs 16-bit values. If the output file format does not support 16-bit data, then imwrite returns an error.
* If A is a gray scale or RGB color image of data type double or single, then imwrite assumes that the dynamic range is [0,1] and automatically scales the data by 255 before writing it to the file as 8-bit values. If the data in A is single, convert A to double before writing to a GIF or TIFF file.
* If A is of data type logical, then imwrite assumes that the data is a binary image and writes it to the file with a bit depth of 1, if the format allows it. BMP, PNG, or TIFF formats accept binary images as input arrays.

If A contains indexed image data, you should additionally specify the map input argument.

imwrite([A](http://in.mathworks.com/help/matlab/ref/imwrite.html" \l "inputarg_A),[map](http://in.mathworks.com/help/matlab/ref/imwrite.html#inputarg_map),[filename](http://in.mathworks.com/help/matlab/ref/imwrite.html#inputarg_filename)) writes the indexed image in A and its associated colormap, map, to the file specified by filename.

* If A is an indexed image of data type double or single, then imwrite converts the indices to zero-based indices by subtracting 1 from each element, and then writes the data as uint8. If the data in A is single, convert A to double before writing to a GIF or TIFF file.

imwrite(**\_\_\_**,[fmt](http://in.mathworks.com/help/matlab/ref/imwrite.html#inputarg_fmt)) writes the image in the format specified by fmt, regardless of the file extension in filename. You can specify fmt after the input arguments in any of the previous syntaxes.

imwrite(**\_\_\_**,[Name,Value](http://in.mathworks.com/help/matlab/ref/imwrite.html#namevaluepairarguments)) specifies additional parameters for output GIF, HDF, JPEG, PBM, PGM, PNG, PPM, and TIFF files, using one or more name-value pair arguments. You can specify Name,Value after the input arguments in any of the previous syntaxes.

## Examples

A = rand(50);

imwrite(A,'myGray.png')

# imfinfo

Information about graphics file

## Syntax

* info = imfinfo(filename)
* info = imfinfo(filename,fmt)
* info = imfinfo(URL)

## Description

* [info](http://in.mathworks.com/help/matlab/ref/imfinfo.html#outputarg_info) = imfinfo([filename](http://in.mathworks.com/help/matlab/ref/imfinfo.html#inputarg_filename)) returns a structure whose fields contain information about an image in a graphics file, filename.
* The format of the file is inferred from its contents.
* If filename is a TIFF, HDF, ICO, GIF, or CUR file containing more than one image, then info is a structure array with one element for each image in the file. For example, info(3) would contain information about the third image in the file.
* [info](http://in.mathworks.com/help/matlab/ref/imfinfo.html#outputarg_info) = imfinfo([filename](http://in.mathworks.com/help/matlab/ref/imfinfo.html#inputarg_filename),[fmt](http://in.mathworks.com/help/matlab/ref/imfinfo.html#inputarg_fmt)) additionally looks for a file named filename.fmt, if MATLAB® cannot find a file named filename.
* [info](http://in.mathworks.com/help/matlab/ref/imfinfo.html#outputarg_info) = imfinfo([URL](http://in.mathworks.com/help/matlab/ref/imfinfo.html#inputarg_URL)) returns information about the image at the specified internet resource, URL.

## Examples

info = imfinfo('ngc6543a.jpg');

# ind2rgb

Convert indexed image to RGB image

## Syntax

RGB = ind2rgb(X,map)

## Description

RGB = ind2rgb(X,map) converts the indexed image, X, and the corresponding colormap, map, to the true color image, RGB. The indexed image, X, is an m-by-n array of integers. The colormap, map, is a three-column array of values in the range [0,1]. Each row of map is a three-element RGB triplet that specifies the red, green, and blue components of a single color of the colormap.

* If you specify X as an array of class uint8 or uint16, then the value 0 corresponds to the first color in the colormap.
* If you specify X as an array of class single or double, then the value 1 corresponds to the first color in the colormap.

The true color image output, RGB, is an m-by-n-by-3 array. For more information on image types, see [Image Types](http://in.mathworks.com/help/matlab/creating_plots/image-types.html).