Data Visualization: Plots and Images (2)

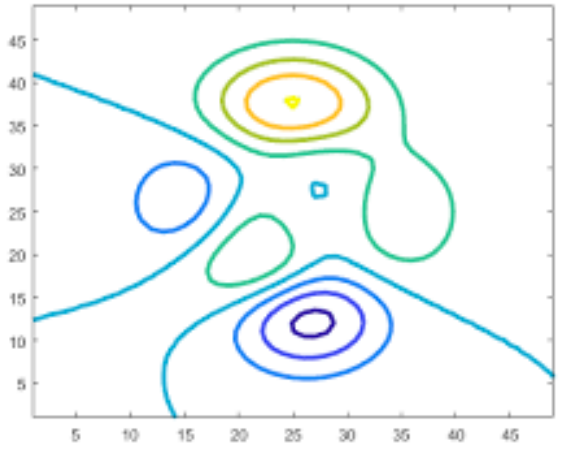
Plotting 2-D Functions

We can plot 2-D functions in the form of $f(x,y)$ by sampling it on a 2-D grid. The sampling grid can be created with:

- Function **`meshgrid`**:
  - Specific for 2-D.
  - Intuitive ordering of **x** and **y** coordinates: **`[X,Y]=meshgrid(x,y)`**, with **`x`** and **`y`** being vectors.
- Function **`ndgrid`**:
  - For 2-D or more dimensions.
  - Following MATLAB ordering of dimensions: **`[A,B]=ndgrid(a,b)`**, with **`a`** and **`b`** being vectors.
  - For 2-D cases, the created matrices are transposes of the matrices created with **`meshgrid`**.

2-D Contour Plots

- Purpose: To display a function **`Z=f(X,Y)`** in 2-D
- Function: **`contour(X,Y,Z)`**
  - Basic form: **`contour(X,Y,Z)`**
  - Specifying the **`Z`** values to draw the contour lines: **`contour(X,Y,Z,v)`**, where **`v`** is a monotonically increasing vector.
  - Specifying the number of contour lines: **`contour(X,Y,Z,n)`**, where **`n`** is an integer.



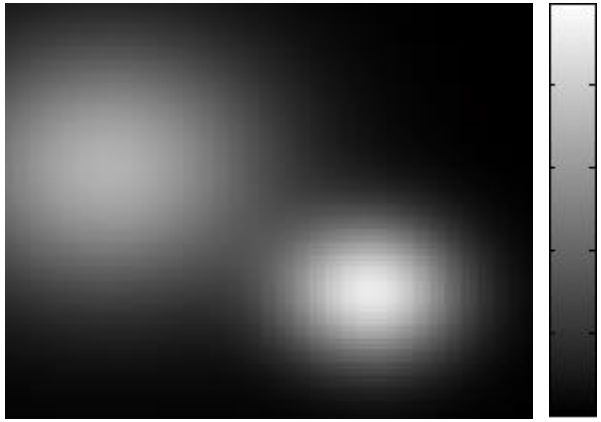
2-D Contour Plots

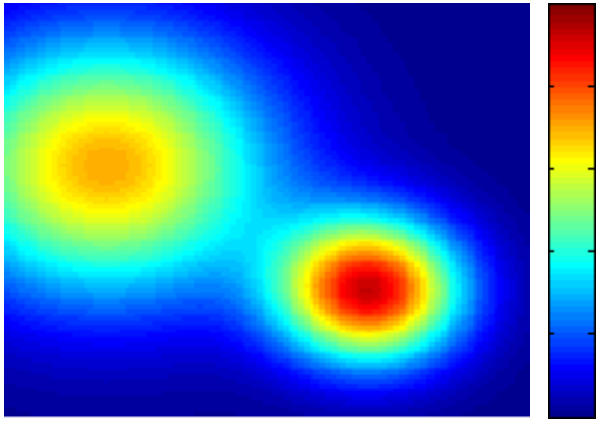
Additional features:

- Showing contour labels: Function **`clabel`**
  - Basic form: **`clabel(C)`** or **`clabel(C,h)`**
  - Labeling specific contours only:
  - Specifying text properties with name-value pairs:
  - Using returned text handles to change text properties (including the text string):
- Filled contour plots: **`contourf(X,Y,Z)`**
  - The use of **colormaps**:

An Introduction to Colormaps

- A mapping from scalar values (positive integers) to color values.
- Used to enhance displays (also called **pseudocolor** processing).





- Colormaps in MATLAB are `Mx3` arrays, each row being a RGB color (values are 0-1).
- Specifying a colormap: `colormap(map)`
  - The specified colormap applies to the whole figure.

An Introduction to Colormaps

- Predefined colormaps in MATLAB:
- Getting a predefined colormap:
  - Example: `jet(32)`

Pseudocolor 2-D Plots

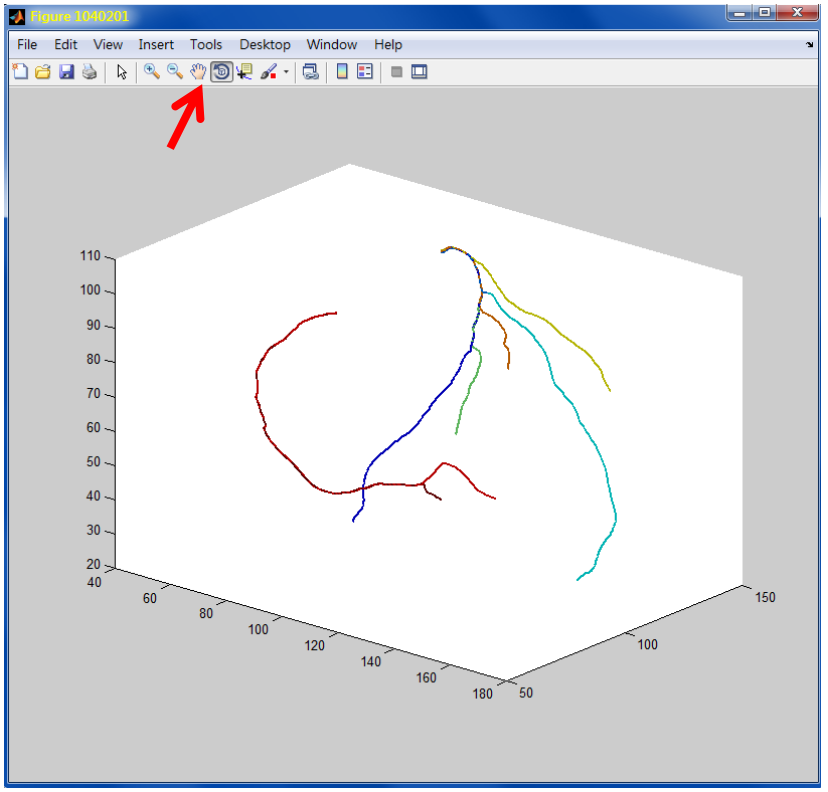
- Function **`imagesc`**:
  - Basic form: **`imagesc(Z)`**, where **`Z`** is a 2-D array.
  - With specified **`x`** and **`y`** values (in vectors): **`imagesc(x,y,Z)`**
  - With specified **`Z`** range: **`imagesc(…,[zmin zmax])`**. If not specified, the full range of data is used.
  - The full range of the current colormap is used.
- Showing the value-color correspondence: Function **`colorbar`** (The "color bar" is displayed in another axes).

3-D Plots

- Line series plots in 3-D: **`plot3(x,y,z,…)`**.
- Scatter plots in 3-D:
  - **`scatter3(x,y,z,…)`**.
  - One can specify the sizes or colors of individual markers: **`scatter3(x,y,z,s,c)`**.
- Many 2-D plotting functions have corresponding 3-D versions, such as **`stem3`**, **`contour3`**, etc. Check them out in the documentation.
- Some plot types that look fancier in 3-D: **`bar3`**, **`pie3`**, etc.

Adjusting Views of 3-D Plots

- Interactive view adjustments:



(In newer versions of MATLAB, select Tools→Rotate 3D, or open the Camera Toolbar to play with the functionalities.)

- View adjustments in programs:
  - Adjusting viewpoints of 3-D plots: function **`view`**.
  - MATLAB provides several functions with names **`cam*`** that controls the "camera" in 3-D views.

Mesh and Surface Plots

- Mesh and surface plots: Showing 2-D functions in 3-D views.
  - **`X`** and **`Y`** should be 2-D grids.
  - Each set of adjacent 2x2 grid points form a patch.
  - Function **`mesh(X,Y,Z,…)`** : Draws only the edges of the patches.
  - Function **`surf(X,Y,Z,…)`** : Draws the patch surfaces.

Image Basics

Basic types of images:

- Binary images, one bit per pixel. In MATLAB: **MxN** arrays of type **logical**. The images are black-and-white.
- Intensity images. In MATLAB: **MxN** arrays. Types can be integers (mostly **uint8**) or floating-point. The images are gray-scale.
- Color images. In MATLAB: **MxNx3** array. Types can be integers (mostly **uint8**) or floating point.
  - The size of the third dimension is the number of **color planes**. Standard color images have 3 planes (R, G, B). Some image formats may use values other than 3, such as 4.

Image Basics

Basic types of images:

- For intensity and color images, when the data type is **uint8**, the values are from 0 to 255. When the data type is floating point, the values are from 0 to 1.
- Indexed images. In MATLAB: An **MxN** array of integers plus a color map.
  - Saves storage space for color images.
  - Makes re-coloring easy.
  - For many processing tasks (such as filtering), it is necessary to convert indexed images to regular (non-indexed) images first.

Loading Images

- Reading images from files: Function **`imread`**:
  - A non-indexed image is always put in an array of type **uint8** (most common) or **unit16**, or **logical** for 1-bit-per-pixel image files.
  - For indexed images, the color map can be retrieved together.

Displaying Images

- Function **`image`**:
  - Used in a way similar to **`plot`**; the image can be displayed as part of an axes using the coordinate system of that axes.
  - If displayed in a new axes, the **`YDir`** property of the axes is set to **`'reversed'`** (same as **`axis ij`**).
  - For indexed images, you need to supply the color map in the call.
- Function **`imshow`**:
  - Similar to **`image`**, but **`axis ij`**, **`axis equal`**, **`axis tight`** , and **`axis off`** are set automatically.

Writing Images

- Function **`imwrite`**:
  - Image file format can be automatically determined from the file name extension.
    - For some file formats, additional properties can be set with name-value pairs.
  - Images in floating-point arrays are always converted to type **uint8** (0→0 and 1→255) in the image files.
  - For indexed images, you need to supply the color map in the call.

Image Type Conversion

- Between data types: Functions **`im2double`**, **`im2uint8`**, **`im2uint16`**, **`im2single`**.
  - Automatic source type checking and scaling.
- Between color and gray-scale images: Function **`rgb2gray`**.
  - You can convert a gray-scale image **`A`** to a RGB image using **`cat(3,A,A,A)`** or **`repmat(A,[1 1 3])`**.
- To and from indexed images: Functions **`rgb2ind`**, **`ind2rgb`**, **`gray2ind`**, and **`ind2gray`**.
  - When converting to an index image, the color map can be supplied or generated automatically.