Article

June 5, 2014

Contour Plots

Illustrate simple contour plotting, contours on an image with a color-bar for the contours, and labelled contours.

Examples from `http://matplotlib.org/gallery.html`

Generating Data

Here's how we generate the data we will use to plot:

```
>>> delta = 0.025
>>> x = np.arange(-3.0, 3.0, delta)
>>> y = np.arange(-2.0, 2.0, delta)
>>> X, Y = np.meshgrid(x, y)
>>> Z1 = mlab.bivariate_normal(X, Y, 1.0, 1.0, 0.0, 0.0)
>>> Z2 = mlab.bivariate_normal(X, Y, 1.5, 0.5, 1, 1)
>>> # difference of Gaussians
... Z = 10.0 * (Z2 - Z1)
>>> X
array([[-3.   , -2.975, -2.95 , ...,  2.925,  2.95 ,  2.975],
       [-3.   , -2.975, -2.95 , ...,  2.925,  2.95 ,  2.975],
       [-3.   , -2.975, -2.95 , ...,  2.925,  2.95 ,  2.975],
       ...,
       [-3.   , -2.975, -2.95 , ...,  2.925,  2.95 ,  2.975],
       [-3.   , -2.975, -2.95 , ...,  2.925,  2.95 ,  2.975],
       [-3.   , -2.975, -2.95 , ...,  2.925,  2.95 ,  2.975]])
>>> Y
array([[-2.   , -2.   , -2.   , ..., -2.   , -2.   , -2.   ],
       [-1.975, -1.975, -1.975, ..., -1.975, -1.975, -1.975],
       [-1.95 , -1.95 , -1.95 , ..., -1.95 , -1.95 , -1.95 ],
       ...,
       [ 1.925,  1.925,  1.925, ...,  1.925,  1.925,  1.925],
       [ 1.95 ,  1.95 ,  1.95 , ...,  1.95 ,  1.95 ,  1.95 ],
       [ 1.975,  1.975,  1.975, ...,  1.975,  1.975,  1.975]])
```

Graphing Defaults

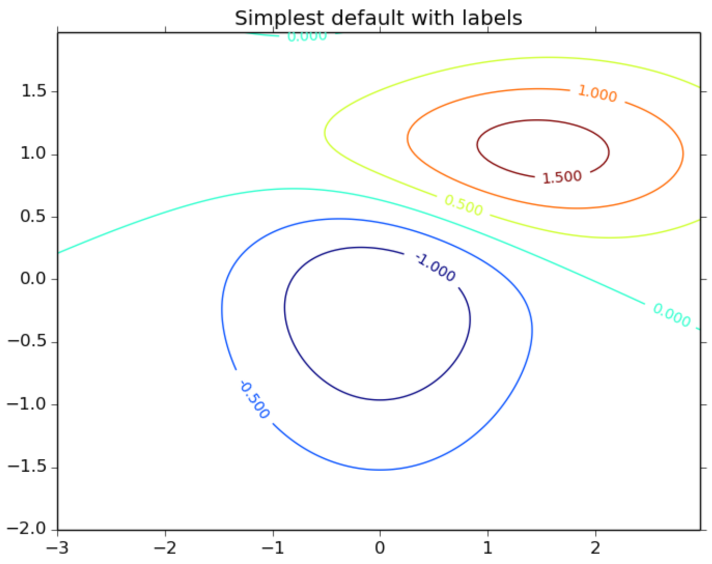
We set up some Matplotlib defaults before we start making plots:

```
matplotlib.rcParams['xtick.direction'] = 'out'
matplotlib.rcParams['ytick.direction'] = 'out'
```

Simple Contour Plot

Create a simple contour plot with labels using default colors. The inline argument to clabel will control whether the labels are draw over the line segments of the contour, removing the lines beneath the label

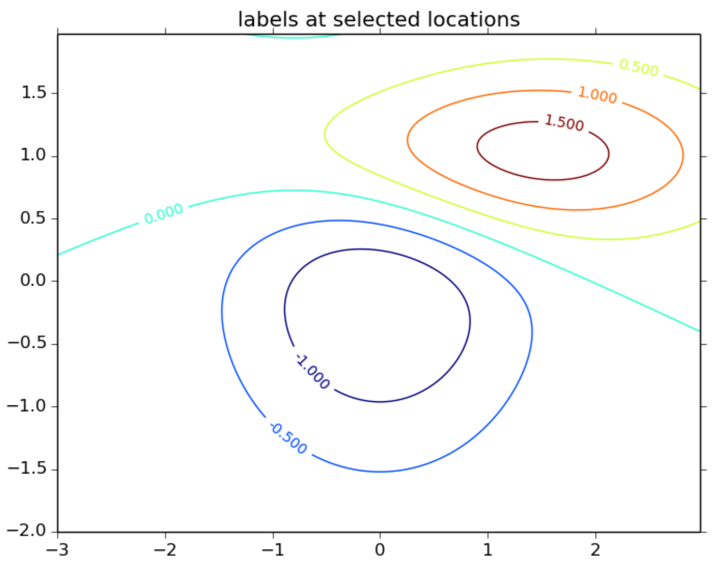
```
plt.figure()
CS = plt.contour(X, Y, Z)
plt.clabel(CS, inline=1, fontsize=10)
plt.title('Simplest default with labels')
with open("simple-contour-plot.pdf", "wb") as f:
    plt.savefig(f)
```



Contour Labels

Contour labels can be placed manually by providing list of positions (in data coordinate).

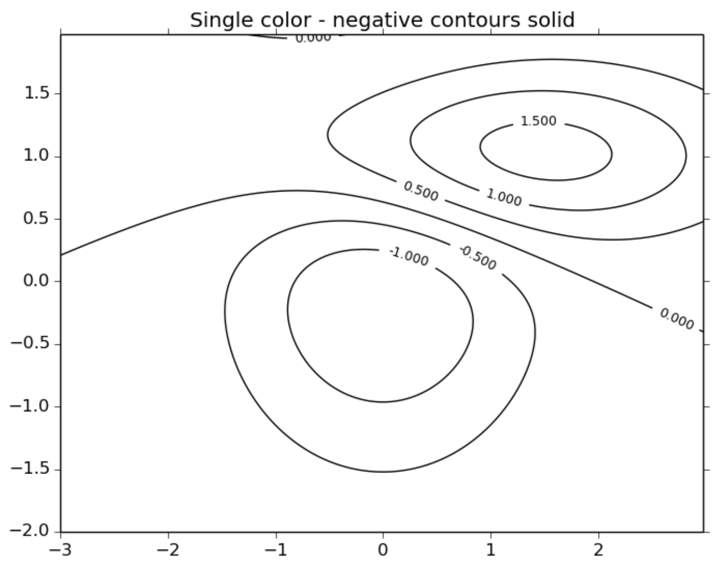
```
plt.figure()
CS = plt.contour(X, Y, Z)
manual_locations = [(-1, -1.4), (-0.62, -0.7), (-2, 0.5), (1.7, 1.2), (2.0, 1.4), (2.4, 1.7)]
plt.clabel(CS, inline=1, fontsize=10, manual=manual_locations)
plt.title('labels at selected locations')
with open("contour-plot-manual-labels.pdf", "wb") as f:
    plt.savefig(f)
```



Negative Contours

You can set negative contours to be solid instead of dashed:

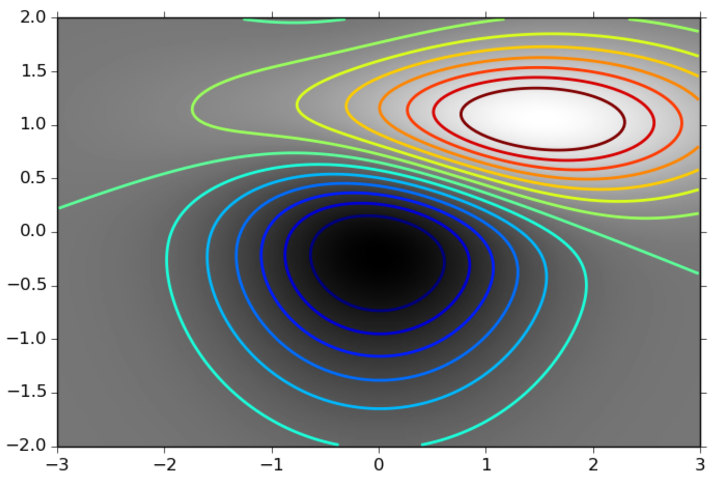
```
matplotlib.rcParams['contour.negative_linestyle'] = 'solid'
plt.figure()
CS = plt.contour(X, Y, Z, 6,
                 colors='k', # negative contours will be dashed by default
                 )
plt.clabel(CS, fontsize=9, inline=1)
plt.title('Single color - negative contours solid')
with open("negative-contours-solid.pdf", "wb") as f:
    plt.savefig(f)
```



Colormap

Or you can use a colormap to specify the colors; the default colormap will be used for the contour lines

```
plt.figure()
im = plt.imshow(Z, interpolation='bilinear', origin='lower',
                cmap=cm.gray, extent=(-3,3,-2,2))
levels = np.arange(-1.2, 1.6, 0.2)
CS = plt.contour(Z, levels,
                 origin='lower',
                 linewidths=2,
                 extent=(-3,3,-2,2))
with open("colormap.pdf", "wb") as f:
    plt.savefig(f)
```



Source Code

LaTeX

Here is the raw LaTeX source code of this document:

```
\section{Contour Plots}

Illustrate simple contour plotting, contours on an image with
a colorbar for the contours, and labelled contours.

Examples from \url{http://matplotlib.org/gallery.html}

\subsection{Generating Data}

Here's how we generate the data we will use to plot:

<< d['contour_demo.py|idio|pycon|pyg|l']['generate-data'] >>

\subsection{Graphing Defaults}

We set up some Matplotlib defaults before we start making plots:

<< d['contour_demo.py|idio|l']['graphing-defaults'] >>

\newpage

\subsection{Simple Contour Plot}

Create a simple contour plot with labels using default colors.  The
inline argument to clabel will control whether the labels are draw
over the line segments of the contour, removing the lines beneath
the label

<< d['contour_demo.py|idio|l']['simple-contour-plot'] >>

\includegraphics{simple-contour-plot.pdf}

\newpage

\subsection{Contour Labels}

Contour labels can be placed manually by providing list of positions (in data coordinate).

<< d['contour_demo.py|idio|l']['contour-labels'] >>
```

```
\includegraphics{contour-plot-manual-labels.pdf}

\newpage

\subsection{Negative Contours}

You can set negative contours to be solid instead of dashed:

<< d['contour_demo.py|idio|l']['negative-contours'] >>

\includegraphics{negative-contours-solid.pdf}

\newpage

\subsection{Colormap}

Or you can use a colormap to specify the colors; the default
colormap will be used for the contour lines

<< d['contour_demo.py|idio|l']['colormap'] >>

\includegraphics{colormap.pdf}

\newpage

\section{Source Code}

\subsection{LaTeX}

Here is the raw \LaTeX\ source code of this document:

<< d['article.tex|pyg|l'] >>

\subsection{Python}

Here is the whole Python script:

<< d['contour_demo.py|pyg|l'] >>
```

Python

Here is the whole Python script:

```
### "imports"
import matplotlib
matplotlib.use("Agg")
import numpy as np
import matplotlib.cm as cm
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt

### "graphing-defaults"
matplotlib.rcParams['xtick.direction'] = 'out'
matplotlib.rcParams['ytick.direction'] = 'out'

### "generate-data"
delta = 0.025
x = np.arange(-3.0, 3.0, delta)
y = np.arange(-2.0, 2.0, delta)
X, Y = np.meshgrid(x, y)
Z1 = mlab.bivariate_normal(X, Y, 1.0, 1.0, 0.0, 0.0)
Z2 = mlab.bivariate_normal(X, Y, 1.5, 0.5, 1, 1)
# difference of Gaussians
Z = 10.0 * (Z2 - Z1)
X
Y

### "simple-contour-plot"
plt.figure()
CS = plt.contour(X, Y, Z)
plt.clabel(CS, inline=1, fontsize=10)
plt.title('Simplest default with labels')
with open("simple-contour-plot.pdf", "wb") as f:
    plt.savefig(f)

### "contour-labels"
plt.figure()
CS = plt.contour(X, Y, Z)
manual_locations = [(-1, -1.4), (-0.62, -0.7), (-2, 0.5), (1.7, 1.2), (2.0, 1.4), (2.4, 1.7)]
plt.clabel(CS, inline=1, fontsize=10, manual=manual_locations)
plt.title('labels at selected locations')
with open("contour-plot-manual-labels.pdf", "wb") as f:
    plt.savefig(f)

### "negative-contours"
matplotlib.rcParams['contour.negative_linestyle'] = 'solid'
plt.figure()
```

```
CS = plt.contour(X, Y, Z, 6,
                 colors='k', # negative contours will be dashed by default
                 )
plt.clabel(CS, fontsize=9, inline=1)
plt.title('Single color - negative contours solid')
with open("negative-contours-solid.pdf", "wb") as f:
    plt.savefig(f)

### "colormap"
plt.figure()
im = plt.imshow(Z, interpolation='bilinear', origin='lower',
                cmap=cm.gray, extent=(-3,3,-2,2))
levels = np.arange(-1.2, 1.6, 0.2)
CS = plt.contour(Z, levels,
                 origin='lower',
                 linewidths=2,
                 extent=(-3,3,-2,2))
with open("colormap.pdf", "wb") as f:
    plt.savefig(f)
```