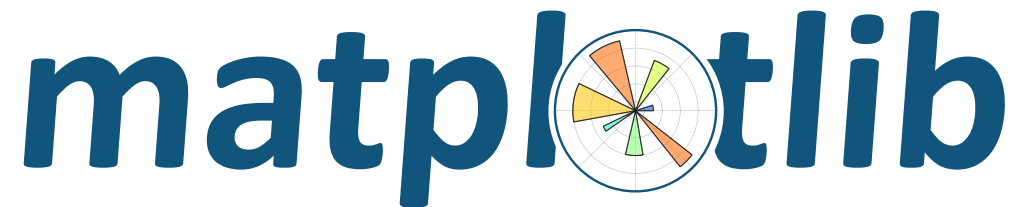


We're updating the default styles for Matplotlib 2.0

Learn what to expect in the [new updates](#)



Fork me on GitHub

[home](#) | [examples](#) | [gallery](#) | [pyplot](#) | [docs](#) » [Matplotlib Examples](#) »

[previous](#) | [next](#) | [modules](#) | [index](#)

[pylab\\_examples Examples](#) »

## pylab\_examples example code: contour\_image.py

([Source code](#), [png](#), [hires.png](#), [pdf](#))

Depsy 100th percentile

Travis-CI: build passing

### Related Topics

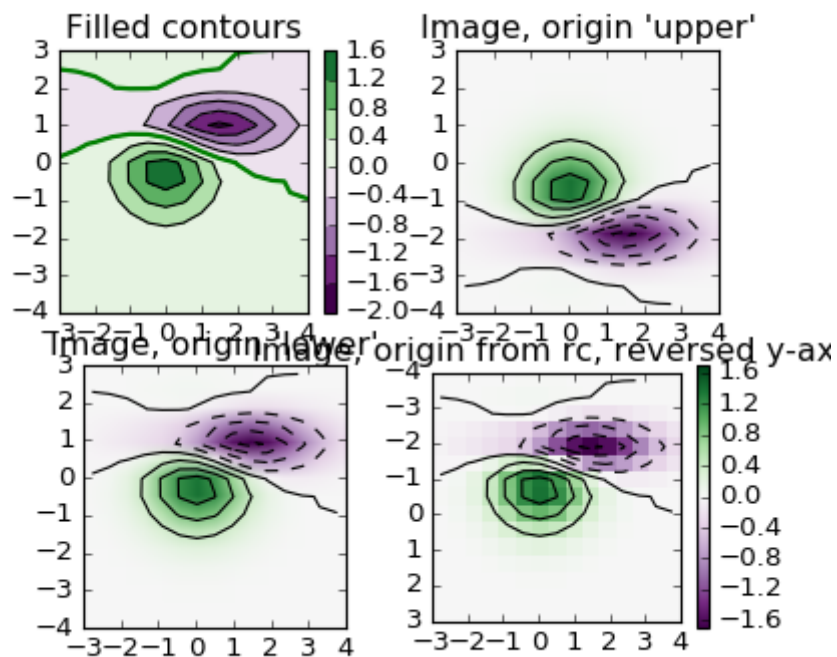
#### Documentation overview

- [Matplotlib Examples](#)
  - [pylab\\_examples Examples](#)
    - Previous:  
[pylab\\_examples example code: contour\\_demo.py](#)
    - Next: [pylab\\_examples example code: contour\\_label\\_demo.py](#)

### This Page

[Show Source](#)

### Quick search



```
#!/usr/bin/env python
'''
Test combinations of contouring, filled contouring, and image plotting.
For contour labelling, see contour_demo.py.
```

```
The emphasis in this demo is on showing how to make contours register
correctly on images, and on how to get both of them oriented as
desired. In particular, note the usage of the "origin" and "extent"
keyword arguments to imshow and contour.
```

```
'''
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import mlab, cm
```

```
# Default delta is large because that makes it fast, and it illustrates
# the correct registration between image and contours.
delta = 0.5
```

```
extent = (-3, 4, -4, 3)
```

Go

Enter search terms or a module, class  
or function name.

```
x = np.arange(-3.0, 4.001, delta)
y = np.arange(-4.0, 3.001, 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)
Z = (Z1 - Z2) * 10

levels = np.arange(-2.0, 1.601, 0.4) # Boost the upper limit to avoid truncation errors.

norm = cm.colors.Normalize(vmax=abs(Z).max(), vmin=-abs(Z).max())
cmap = cm.PRGN

plt.figure()

plt.subplot(2, 2, 1)

cset1 = plt.contourf(X, Y, Z, levels,
                    cmap=cm.get_cmap(cmap, len(levels) - 1),
                    norm=norm,
                    )
# It is not necessary, but for the colormap, we need only the
# number of levels minus 1. To avoid discretization error, use
# either this number or a large number such as the default (256).

# If we want lines as well as filled regions, we need to call
# contour separately; don't try to change the edgecolor or edgewidth
# of the polygons in the collections returned by contourf.
# Use levels output from previous call to guarantee they are the same.
cset2 = plt.contour(X, Y, Z, cset1.levels,
                   colors='k',
                   hold='on')
# We don't really need dashed contour lines to indicate negative
# regions, so let's turn them off.
for c in cset2.collections:
    c.set_linestyle('solid')

# It is easier here to make a separate call to contour than
# to set up an array of colors and linewidths.
# We are making a thick green line as a zero contour.
# Specify the zero level as a tuple with only 0 in it.
```

```
cset3 = plt.contour(X, Y, Z, (0,)),
                colors='g',
                linewidths=2,
                hold='on')
plt.title('Filled contours')
plt.colorbar(cset1)
#hot()

plt.subplot(2, 2, 2)

plt.imshow(Z, extent=extent, cmap=cmap, norm=norm)
v = plt.axis()
plt.contour(Z, levels, hold='on', colors='k',
            origin='upper', extent=extent)
plt.axis(v)
plt.title("Image, origin 'upper'")

plt.subplot(2, 2, 3)

plt.imshow(Z, origin='lower', extent=extent, cmap=cmap, norm=norm)
v = plt.axis()
plt.contour(Z, levels, hold='on', colors='k',
            origin='lower', extent=extent)
plt.axis(v)
plt.title("Image, origin 'lower'")

plt.subplot(2, 2, 4)

# We will use the interpolation "nearest" here to show the actual
# image pixels.
# Note that the contour lines don't extend to the edge of the box.
# This is intentional. The Z values are defined at the center of each
# image pixel (each color block on the following subplot), so the
# domain that is contoured does not extend beyond these pixel centers.
im = plt.imshow(Z, interpolation='nearest', extent=extent, cmap=cmap, norm=norm)
v = plt.axis()
plt.contour(Z, levels, hold='on', colors='k',
            origin='image', extent=extent)
plt.axis(v)
ylim = plt.get(plt.gca(), 'ylim')
plt.setp(plt.gca(), ylim=ylim[:-1])
```

```
plt.title("Image, origin from rc, reversed y-axis")  
plt.colorbar(im)  
  
plt.show()
```

Keywords: python, matplotlib, pylab, example, codex (see [Search examples](#))

© Copyright 2002 - 2012 John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the matplotlib development team; 2012 - 2014 The matplotlib development team. Last updated on Jun 18, 2016. Created using [Sphinx](#) 1.3.5.