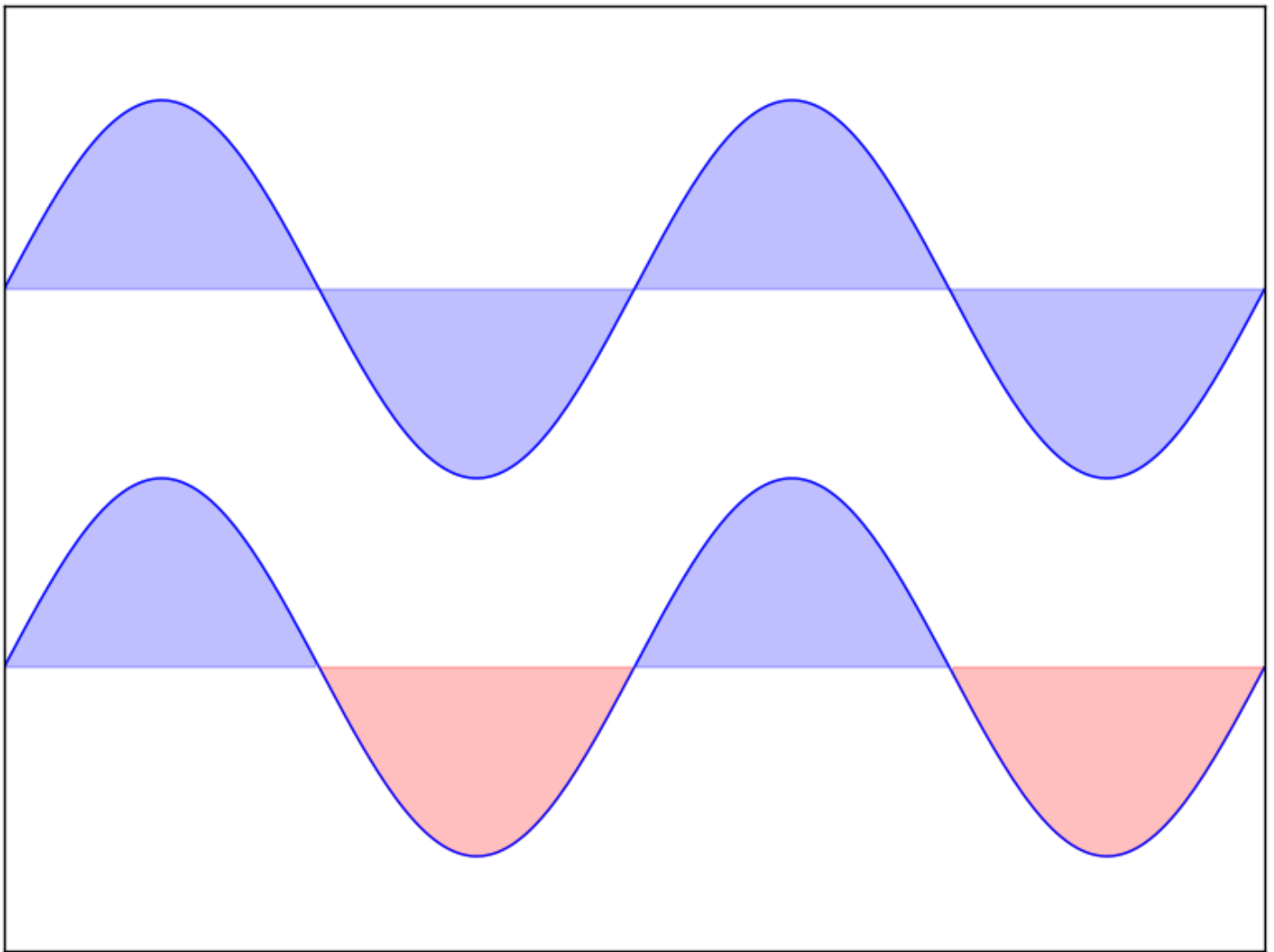


# python matplotlib plotting examples and exercises

---

10 April 2015

## 1. regular plots



```
import numpy as np
import pylab as pl

n = 256
X = np.linspace(-np.pi, np.pi, n, endpoint=True)
Y = np.sin(2 * X)

pl.axes([0.025, 0.025, 0.95, 0.95])

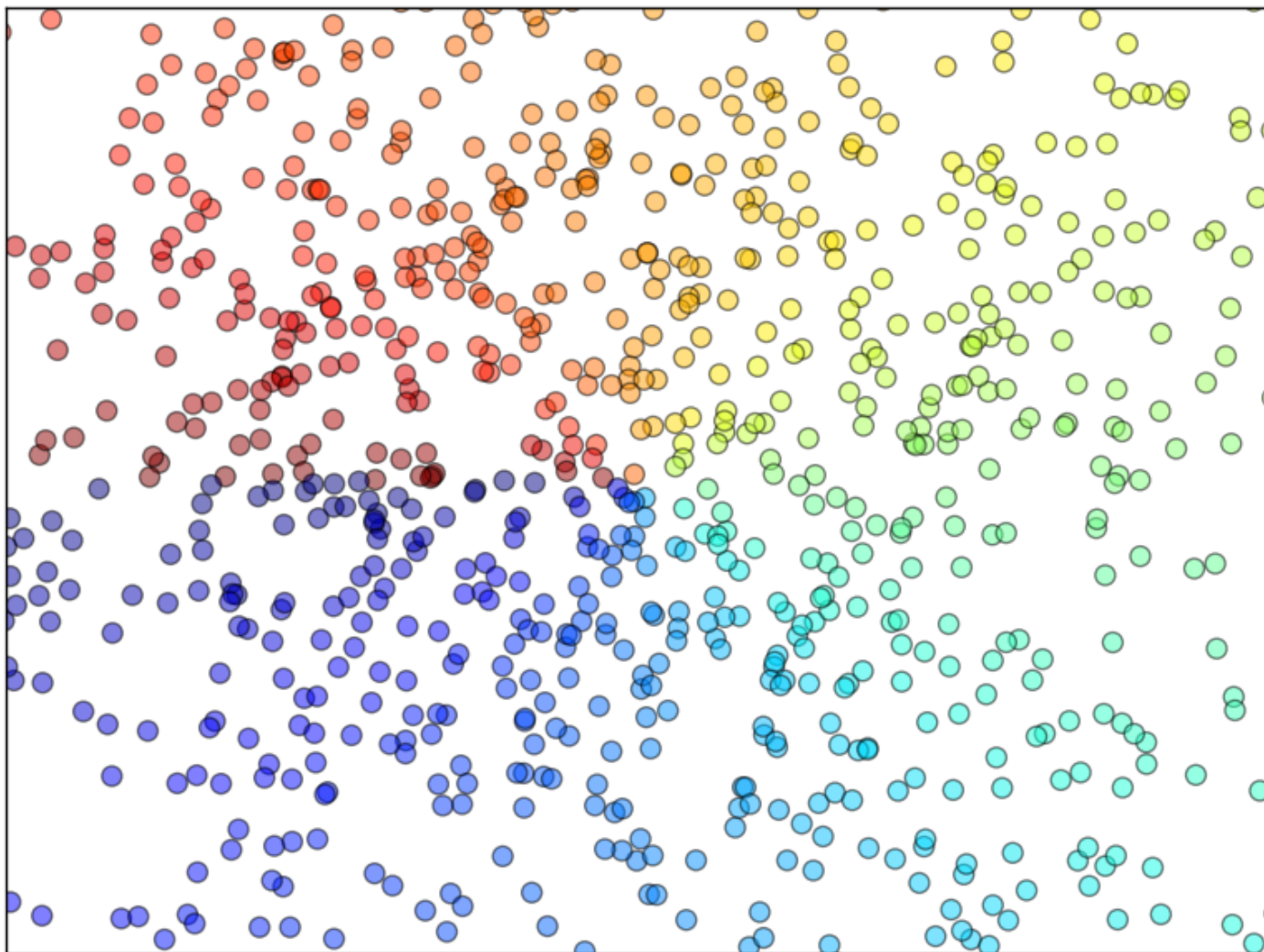
pl.plot(X, Y + 1, color='blue', alpha=1.00)
pl.fill_between(X, 1, Y + 1, color='blue', alpha=.25)

pl.plot(X, Y - 1, color='blue', alpha=1.00)
pl.fill_between(X, -1, Y - 1, (Y - 1) > -1, color='blue', alpha=.25)
pl.fill_between(X, -1, Y - 1, (Y - 1) < -1, color='red', alpha=.25)

pl.xlim(-np.pi, np.pi)
pl.xticks(())
pl.ylim(-2.5, 2.5)
pl.yticks(())

pl.show()
```

## 2. scatter plots



```
import numpy as np
import pylab as pl

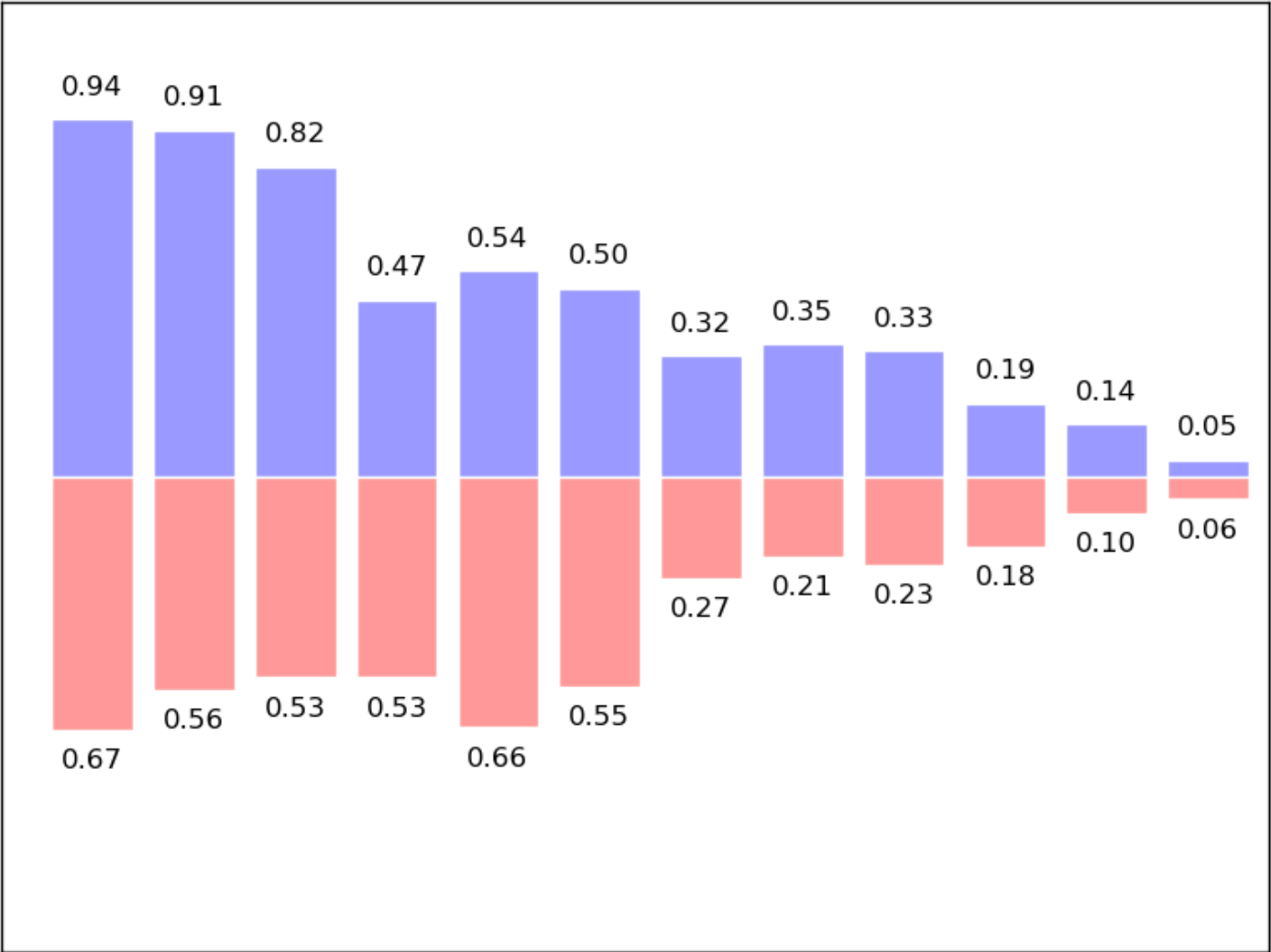
n = 1024
X = np.random.normal(0, 1, n)
Y = np.random.normal(0, 1, n)
T = np.arctan2(Y, X)

pl.axes([0.025, 0.025, 0.95, 0.95])
pl.scatter(X, Y, s=75, c=T, alpha=.5)

pl.xlim(-1.5, 1.5)
pl.xticks(())
pl.ylim(-1.5, 1.5)
pl.yticks(())

pl.show()
```

### 3. bar plots



```
import numpy as np
import pylab as pl

n = 12
X = np.arange(n)
Y1 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)
Y2 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)

pl.axes([0.025, 0.025, 0.95, 0.95])
pl.bar(X, +Y1, facecolor='#9999ff', edgecolor='white')
pl.bar(X, -Y2, facecolor='#ff9999', edgecolor='white')

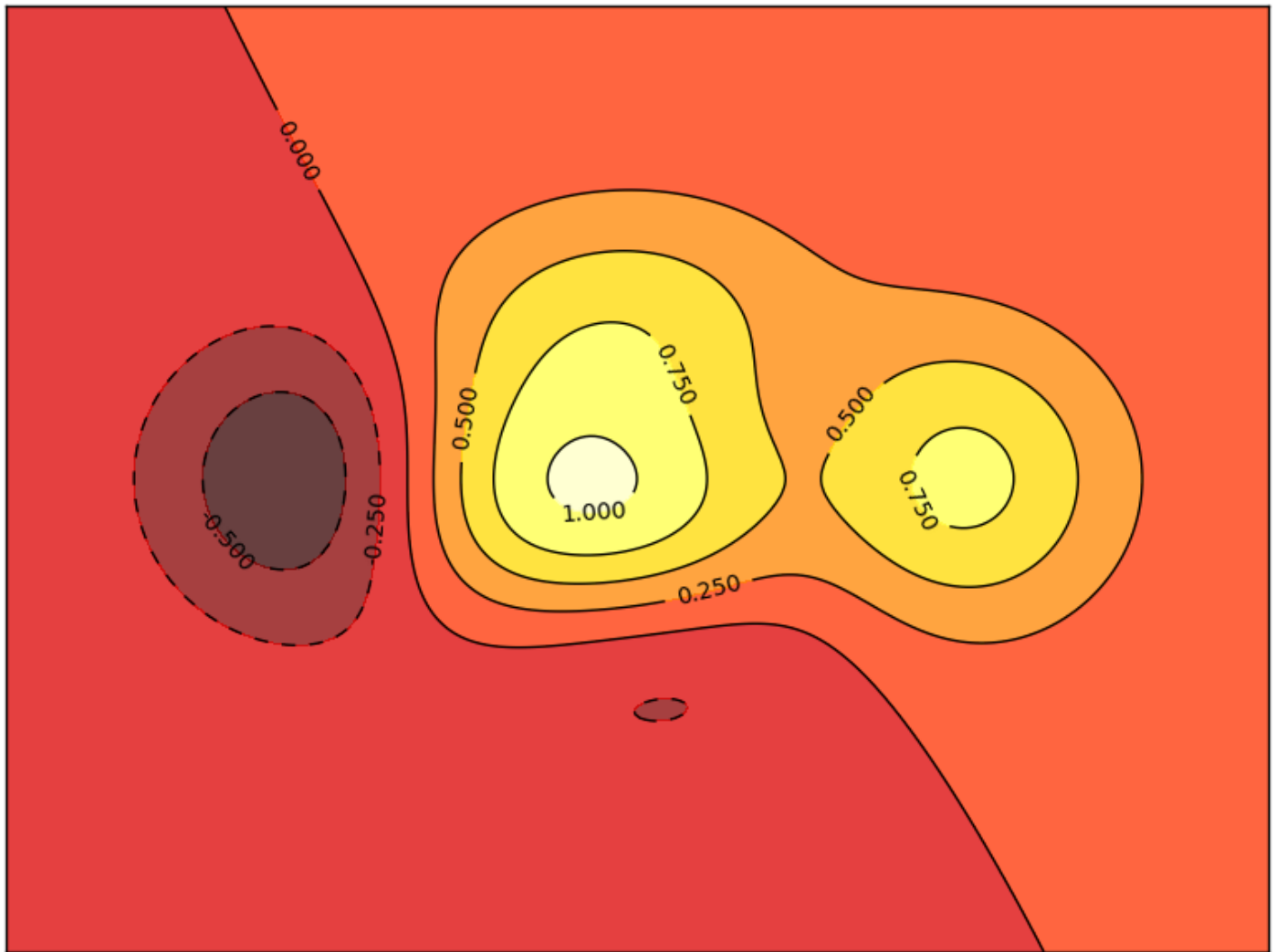
for x, y in zip(X, Y1):
    pl.text(x + 0.4, y + 0.05, '%.2f' % y, ha='center', va= 'bottom')

for x, y in zip(X, Y2):
    pl.text(x + 0.4, -y - 0.05, '%.2f' % y, ha='center', va= 'top')

pl.xlim(-.5, n)
pl.xticks(())
pl.ylim(-1.25, 1.25)
pl.yticks(())

pl.show()
```

## 4. Contour plots



```
import numpy as np
import pylab as pl

def f(x,y):
    return (1 - x / 2 + x**5 + y**3) * np.exp(-x**2 -y**2)

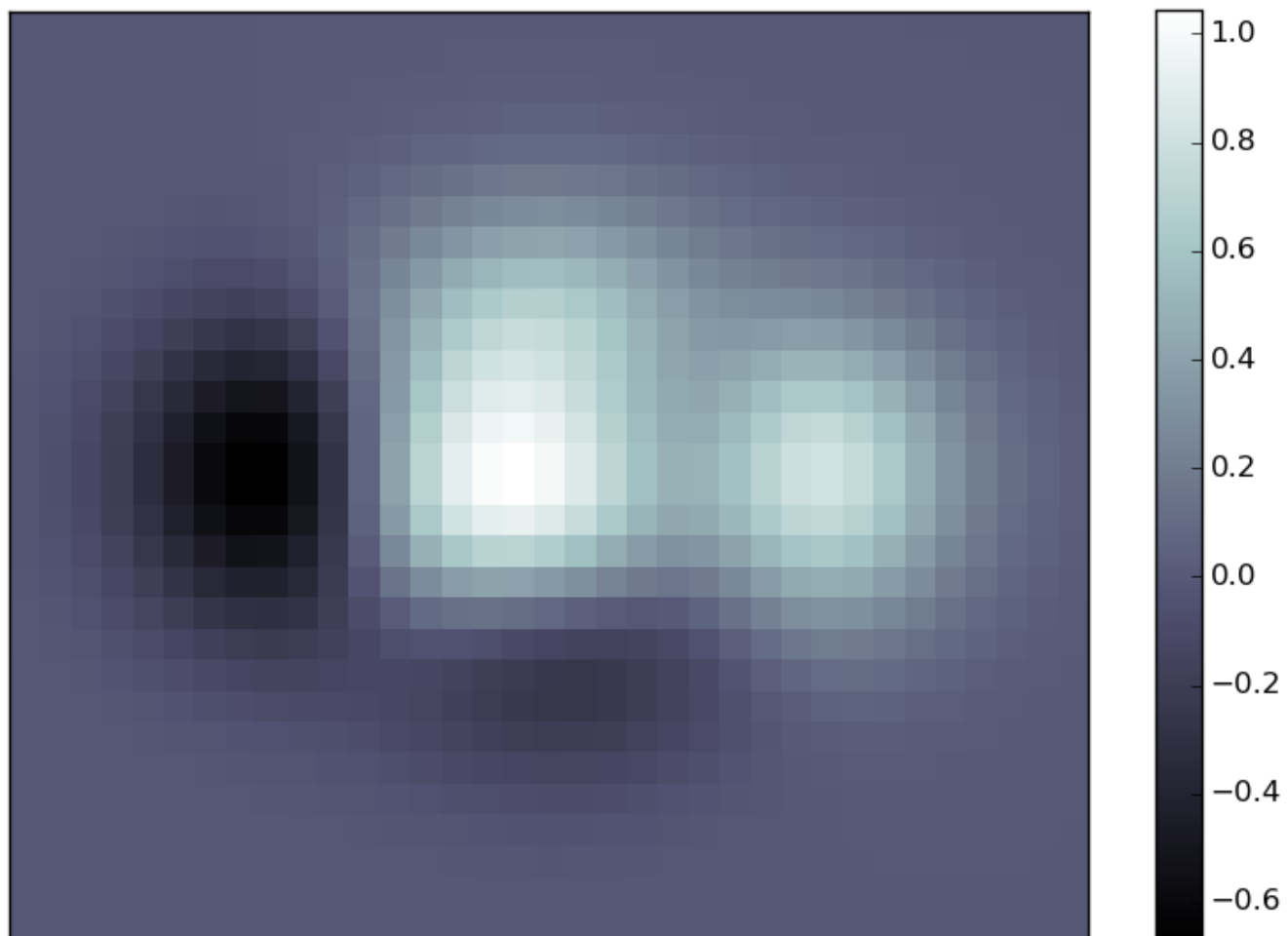
n = 256
x = np.linspace(-3, 3, n)
y = np.linspace(-3, 3, n)
X,Y = np.meshgrid(x, y)

pl.axes([0.025, 0.025, 0.95, 0.95])

pl.contourf(X, Y, f(X, Y), 8, alpha=.75, cmap=pl.cm.hot)
C = pl.contour(X, Y, f(X, Y), 8, colors='black', linewidth=.5)
pl.clabel(C, inline=1, fontsize=10)

pl.xticks(())
pl.yticks(())
pl.show()
```

## 5. imshow



```
import numpy as np
import pylab as pl

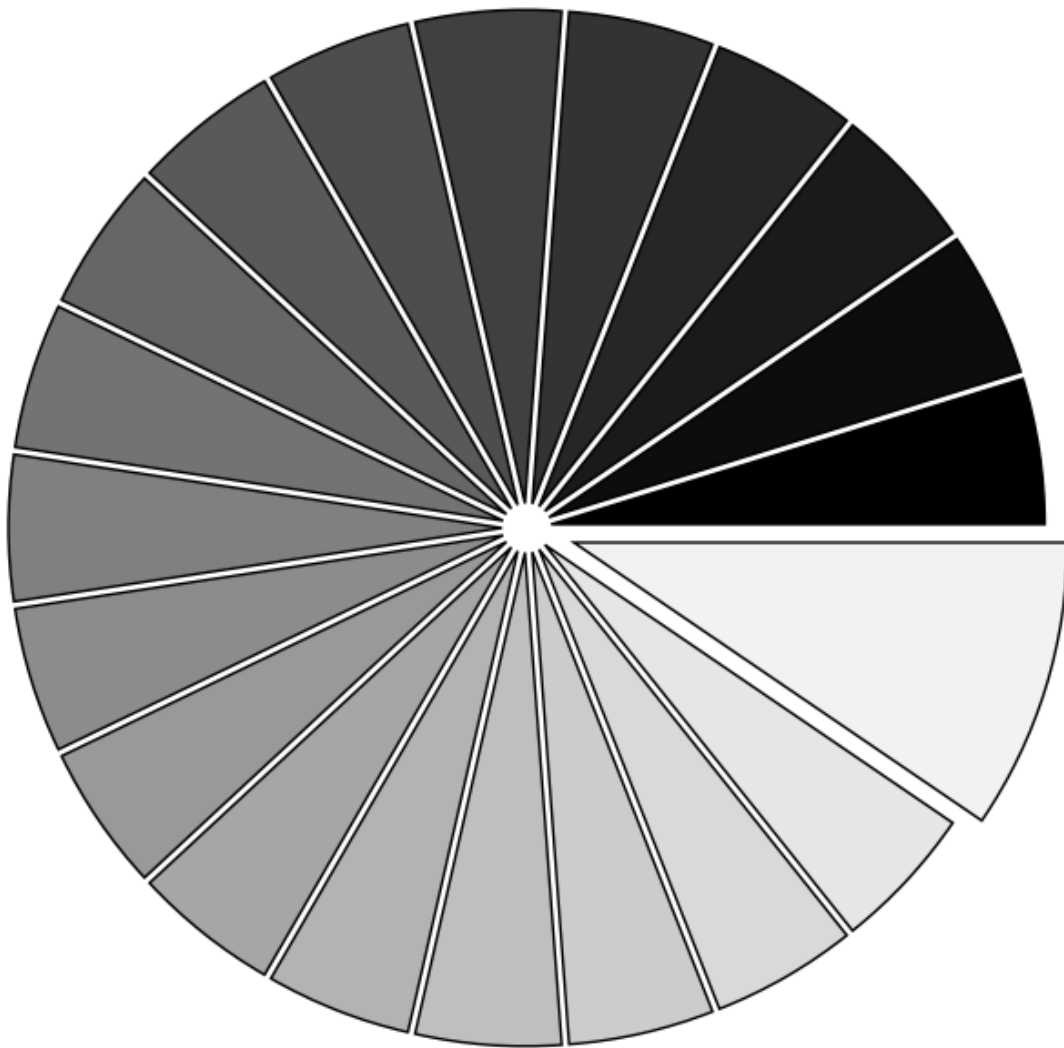
def f(x, y):
    return (1 - x / 2 + x ** 5 + y ** 3) * np.exp(-x ** 2 - y ** 2)

n = 10
x = np.linspace(-3, 3, 3.5 * n)
y = np.linspace(-3, 3, 3.0 * n)
X, Y = np.meshgrid(x, y)
Z = f(X, Y)

pl.axes([0.025, 0.025, 0.95, 0.95])
pl.imshow(Z, interpolation='nearest', cmap='bone', origin='lower')
pl.colorbar(shrink=.92)

pl.xticks(())
pl.yticks(())
pl.show()
```

## 6. pie charts



```
import numpy as np
import pylab as pl

n = 20
Z = np.ones(n)
Z[-1] *= 2

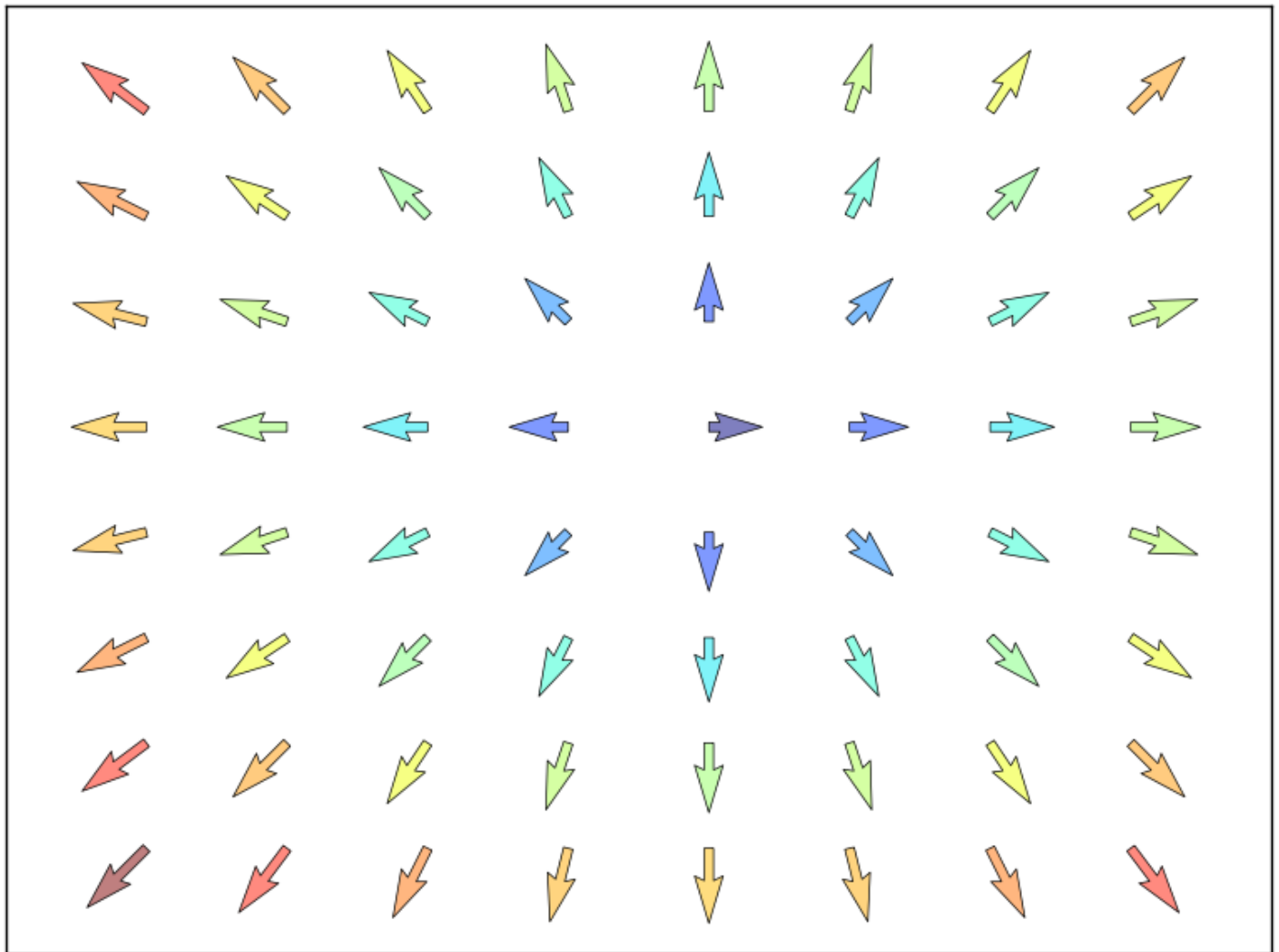
pl.axes([0.025, 0.025, 0.95, 0.95])

pl.pie(Z, explode=Z*.05, colors = ['%f' % (i/float(n)) for i in range(n)])
pl.axis('equal')
pl.xticks(())
pl.yticks()

pl.show()
```

## 7. quiver plots





```
import numpy as np
import pylab as pl

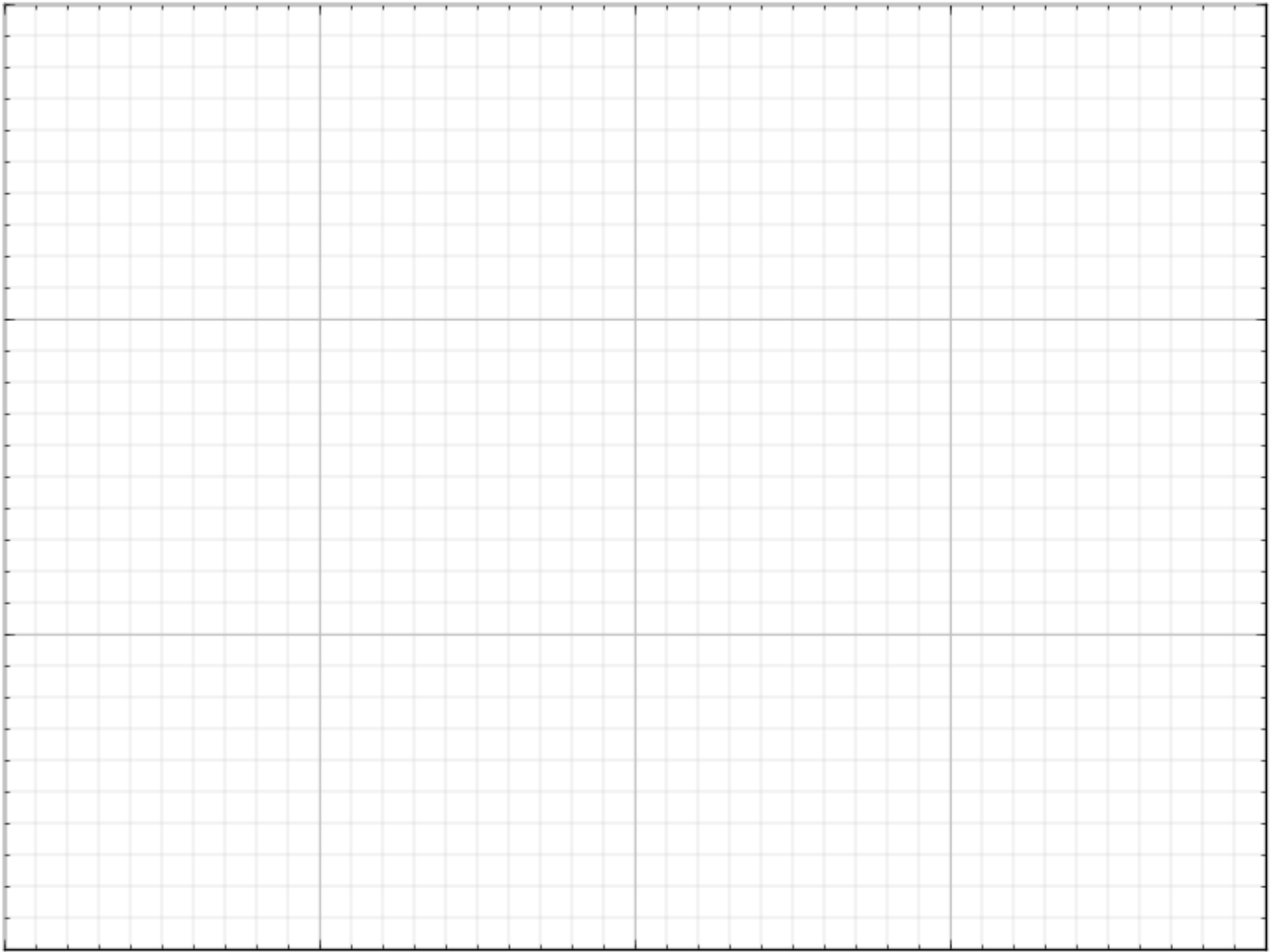
n = 8
X, Y = np.mgrid[0:n, 0:n]
T = np.arctan2(Y - n / 2., X - n / 2.)
R = 10 + np.sqrt((Y - n / 2.0) ** 2 + (X - n / 2.0) ** 2)
U, V = R * np.cos(T), R * np.sin(T)

pl.axes([0.025, 0.025, 0.95, 0.95])
pl.quiver(X, Y, U, V, R, alpha=.5)
pl.quiver(X, Y, U, V, edgecolor='k', facecolor='None', linewidth=.5)

pl.xlim(-1, n)
pl.xticks(())
pl.ylim(-1, n)
pl.yticks(())

pl.show()
```

## 8. grids



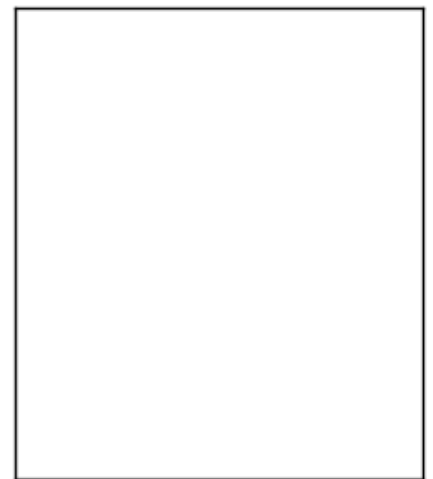
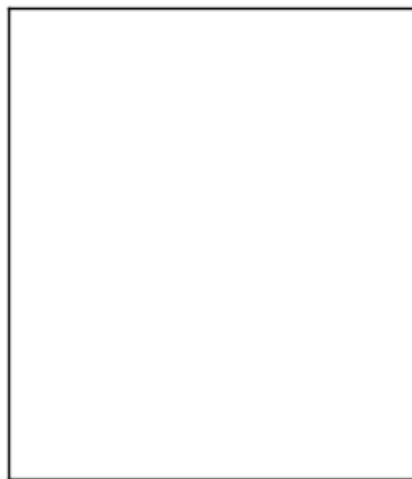
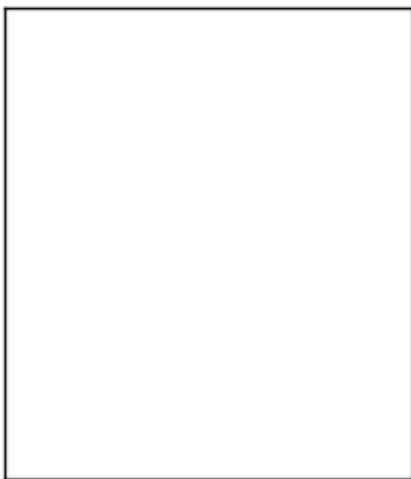
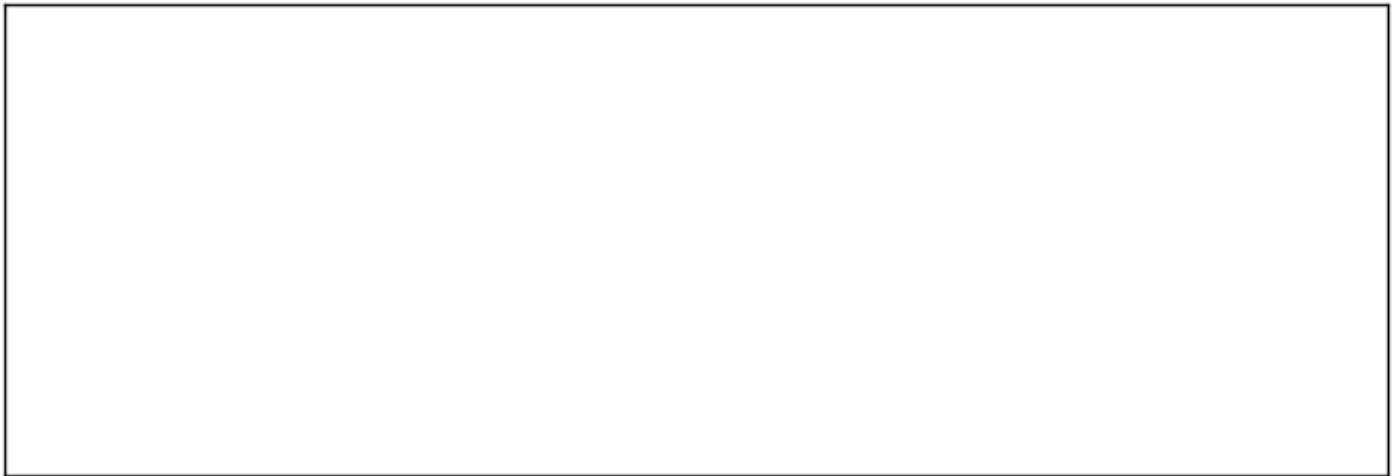
```
import pylab as pl

ax = pl.axes([0.025, 0.025, 0.95, 0.95])

ax.set_xlim(0,4)
ax.set_ylim(0,3)
ax.xaxis.set_major_locator(pl.MultipleLocator(1.0))
ax.xaxis.set_minor_locator(pl.MultipleLocator(0.1))
ax.yaxis.set_major_locator(pl.MultipleLocator(1.0))
ax.yaxis.set_minor_locator(pl.MultipleLocator(0.1))
ax.grid(which='major', axis='x', linewidth=0.75, linestyle='-', color='0.75')
ax.grid(which='minor', axis='x', linewidth=0.25, linestyle='-', color='0.75')
ax.grid(which='major', axis='y', linewidth=0.75, linestyle='-', color='0.75')
ax.grid(which='minor', axis='y', linewidth=0.25, linestyle='-', color='0.75')
ax.set_xticklabels([])
ax.set_yticklabels([])

pl.show()
```

## 9. multi plots



```
import pylab as pl

fig = pl.figure()
fig.subplots_adjust(bottom=0.025, left=0.025, top = 0.975, right=0.975)

pl.subplot(2, 1, 1)
pl.xticks(), pl.yticks()

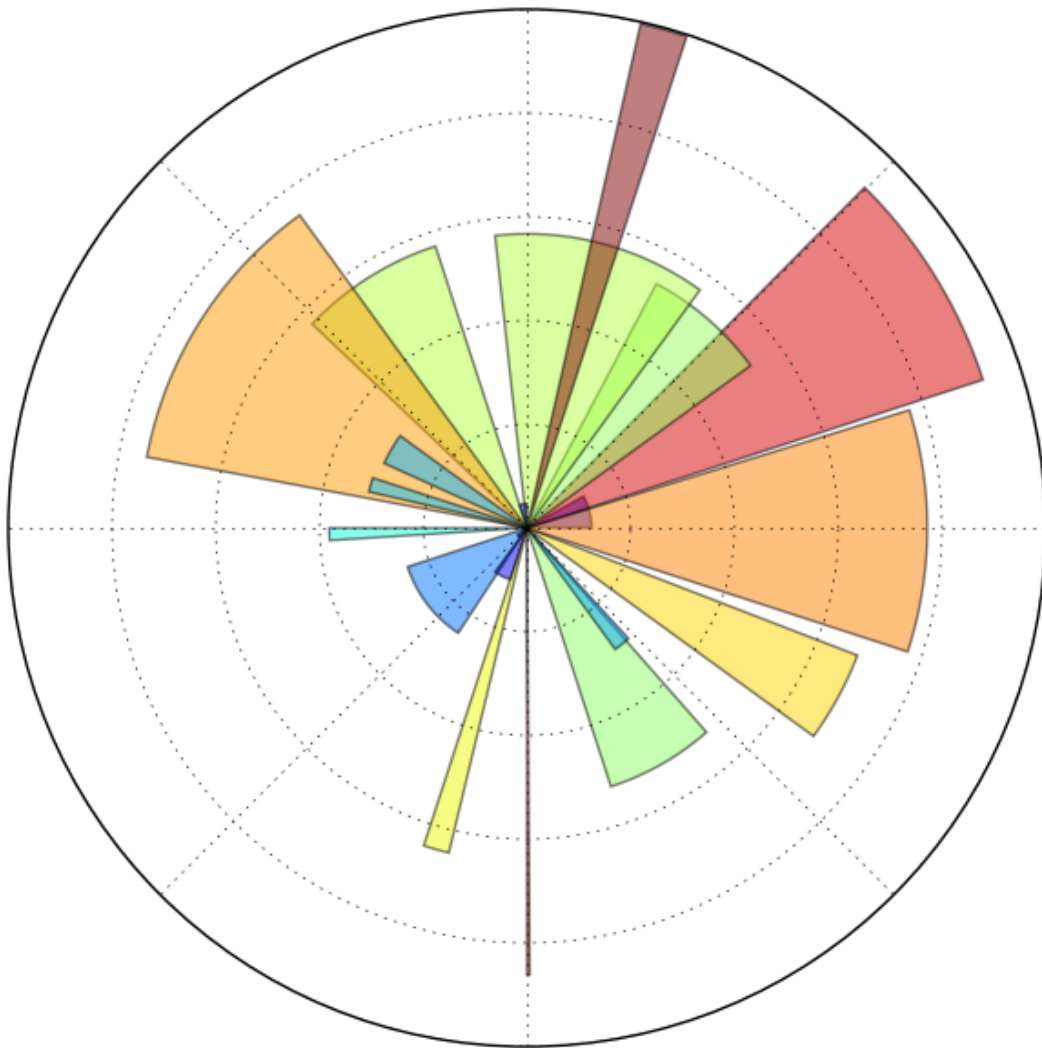
pl.subplot(2, 3, 4)
pl.xticks()
pl.yticks()

pl.subplot(2, 3, 5)
pl.xticks()
pl.yticks()

pl.subplot(2, 3, 6)
pl.xticks()
pl.yticks()

pl.show()
```

## 10. polar axis



```
import numpy as np
import pylab as pl

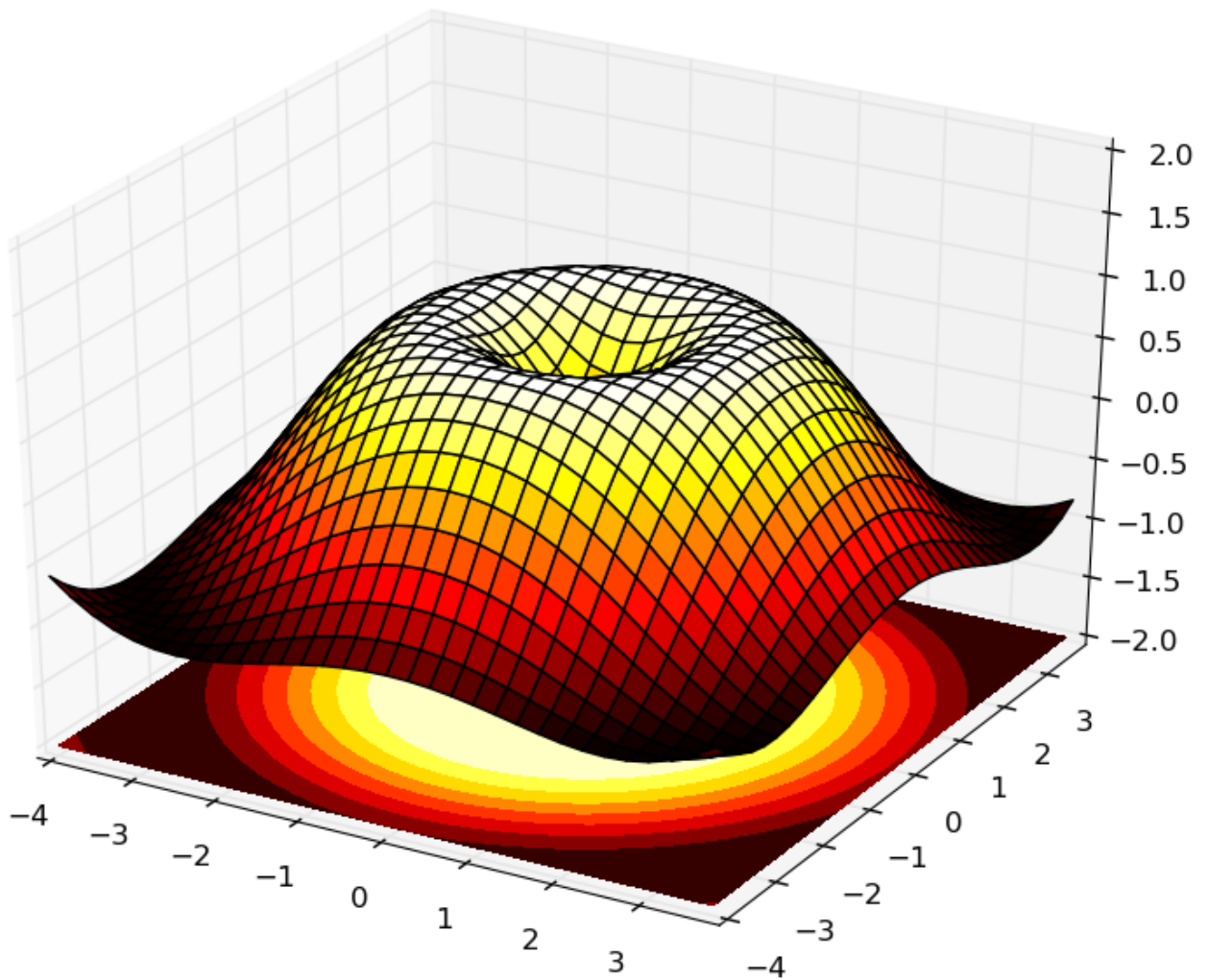
ax = pl.axes([0.025, 0.025, 0.95, 0.95], polar=True)

N = 20
theta = np.arange(0.0, 2 * np.pi, 2 * np.pi / N)
radii = 10 * np.random.rand(N)
width = np.pi / 4 * np.random.rand(N)
bars = pl.bar(theta, radii, width=width, bottom=0.0)

for r, bar in zip(radii, bars):
    bar.set_facecolor(pl.cm.jet(r/10.))
    bar.set_alpha(0.5)

ax.set_xticklabels([])
ax.set_yticklabels([])
pl.show()
```

## 11. 3d plots



```
import numpy as np
import pylab as pl
from mpl_toolkits.mplot3d import Axes3D

fig = pl.figure()
ax = Axes3D(fig)
X = np.arange(-4, 4, 0.25)
Y = np.arange(-4, 4, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X ** 2 + Y ** 2)
Z = np.sin(R)

ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=pl.cm.hot)
ax.contourf(X, Y, Z, zdir='z', offset=-2, cmap=pl.cm.hot)
ax.set_zlim(-2, 2)

pl.show()
```

## 12. text

$$\begin{aligned}
 E &= mc^2 = \sqrt{m_0^2 c^4 + p^2 c^2} \\
 W_{\delta_1 \rho_1 \sigma_2}^{3\beta} &= U_{\delta_1 \rho_1}^{3\beta} + \frac{1}{8\pi^2} \int_{\alpha_2}^{\alpha_2} d\alpha'_2 \left[ \frac{U_{\delta_1 \rho_1}^{2\beta} - \alpha'_2 U_{\rho_1 \sigma_2}^{1\beta}}{U_{\rho_1 \sigma_2}^{0\beta}} \right] \\
 W_{\delta_1 \rho_1 \sigma_2}^{3\beta} &= U_{\delta_1 \rho_1}^{3\beta} + \frac{1}{8\pi^2} \int_{\alpha_2}^{\alpha_2} d\alpha'_2 \left[ \frac{U_{\delta_1 \rho_1}^{2\beta} - \alpha'_2 U_{\rho_1 \sigma_2}^{1\beta}}{U_{\rho_1 \sigma_2}^{0\beta}} \right] \\
 E &= mc^2 = \sqrt{m_0^2 c^4 + p^2 c^2} \\
 F_G &= G \frac{m_1 m_2}{r^2} \\
 E &= mc^2 = \sqrt{m_0^2 c^4 + p^2 c^2} \\
 F_G &= G \frac{m_1 m_2}{r^2} \\
 W_{\delta_1 \rho_1 \sigma_2}^{3\beta} &= U_{\delta_1 \rho_1}^{3\beta} + \frac{1}{8\pi^2} \int_{\alpha_2}^{\alpha_2} d\alpha'_2 \left[ \frac{U_{\delta_1 \rho_1}^{2\beta} - \alpha'_2 U_{\rho_1 \sigma_2}^{1\beta}}{U_{\rho_1 \sigma_2}^{0\beta}} \right] \\
 E &= mc^2 = \sqrt{m_0^2 c^4 + p^2 c^2} \\
 F_G &= G \frac{m_1 m_2}{r^2}
 \end{aligned}$$

```

import numpy as np
import pylab as pl

eqs = []
eqs.append((r"$W^{\{3\beta\}}_{\{\delta_1 \rho_1 \sigma_2\}} = U^{\{3\beta\}}_{\{\delta_1 \rho_1\}} + \frac{1}{8 \pi^2} \int^{\{\alpha_2\}}_{\{\alpha_2\}} d \alpha^{\prime_2} \left[ \frac{U^{\{2\beta\}}_{\{\delta_1 \rho_1\}} - \alpha^{\prime_2} U^{\{1\beta\}}_{\{\rho_1 \sigma_2\}} \right] U^{\{0\beta\}}_{\{\rho_1 \sigma_2\}} \right]$"))
eqs.append((r"$\frac{d\rho}{dt} + \rho \vec{v} \cdot \nabla \vec{v} = -\nabla p + \mu \nabla^2 \vec{v} + \rho \vec{g}$"))
eqs.append((r"$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$"))
eqs.append((r"$E = mc^2 = \sqrt{m_0^2 c^4 + p^2 c^2}$"))
eqs.append((r"$F_G = G \frac{m_1 m_2}{r^2}$"))

pl.axes([0.025, 0.025, 0.95, 0.95])

for i in range(24):
    index = np.random.randint(0, len(eqs))
    eq = eqs[index]
    size = np.random.uniform(12, 32)
    x,y = np.random.uniform(0, 1, 2)
    alpha = np.random.uniform(0.25, .75)
    pl.text(x, y, eq, ha='center', va='center', color="#11557c", alpha=alpha,
            transform=pl.gca().transAxes, fontsize=size, clip_on=True)
pl.xticks(())
pl.yticks(())

pl.show()

```


[python<sup>95</sup> \(/categories.html#python-ref\)](/categories.html#python-ref)

[matplotlib<sup>8</sup> \(/tags.html#matplotlib-ref\)](/tags.html#matplotlib-ref)
[plot<sup>5</sup> \(/tags.html#plot-ref\)](/tags.html#plot-ref)
[pylab<sup>2</sup> \(/tags.html#pylab-ref\)](/tags.html#pylab-ref)
[« Previous \(/python/2015/04/09/python-numpy-and-ipython-scipy2013-part1-the-numpy-array-object\)](/python/2015/04/09/python-numpy-and-ipython-scipy2013-part1-the-numpy-array-object)
[Archive \(/archive.html\)](/archive.html)
[Next » \(/python/2015/04/11/python-matplotlib-image-tutorial\)](/python/2015/04/11/python-matplotlib-image-tutorial)

**0 Comments** **hqlgree2****1 Login** ▾ **Recommend**  **Share****Sort by Best** ▾

Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Be the first to comment.

ALSO ON HQLGREE2

**python instal pymssql on mac**

9 comments • a year ago

**Sherif Eid** — saved the day for me, thanks**mac diskutil command**




1 comment • 3 years ago

**JM** — none of you fuckin twits ever does a fat partition what the fuck is this good for you morons?**apache kylin get started**

1 comment • 2 years ago

**Rejwan Shamsul Kabir** — How did you fix the first problem ?Thanks in advance.**install nagiosgraph**

1 comment • 2 years ago

**Mahmoud Alizadeh** — Hi\$ perl -MCPAN -e shell cpan[1]> install Nagios::Configtrys to download a file from internet ! And my server has no **Subscribe**  **Add Disqus to your site** **Add Disqus** **Add**  **Privacy**

© 2018 hqlgree2 with help from Jekyll Bootstrap (<http://jekyllbootstrap.com>) and Twitter Bootstrap (<http://twitter.github.com/bootstrap/>)