2-D special linear transformations

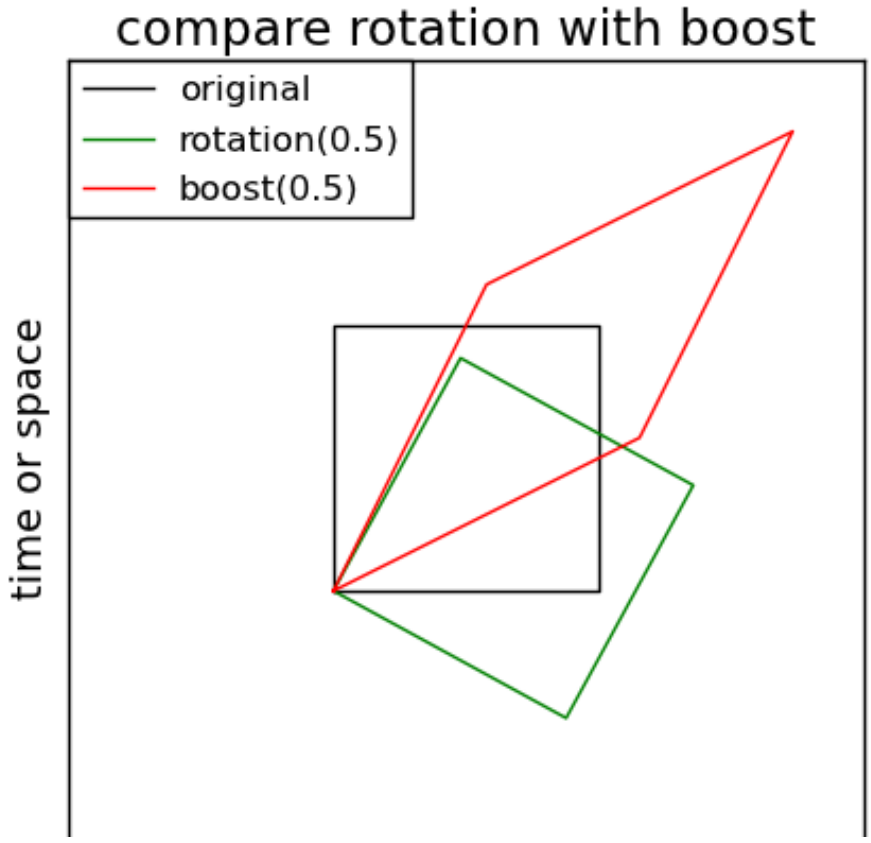
This compares boost and rotation transformations of the unit square. Both are elements of a special linear group, with determinant 1.0. This guarantees that the area is maintained, as the Jacobian is just this determinant. See the wikipedia discussion

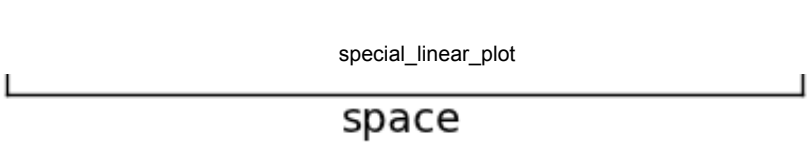
The following shows how python, with matplotlib and numpy, makes this easy to show.

Additionally, the IPython 0.12 notebook allows nice geneneration of a document

In [1]:
```
def R(theta):
    return matrix([[cos(theta), -sin(theta)],[sin(theta),cos(theta)]])
def B(beta):
    return matrix([[1,beta],[beta,1]])/sqrt(1-beta**2)
def plot_square(T, **kwargs):
    unitsquare = array([[0,0], [0,1], [1,1], [1,0], [0,0]])
    t = unitsquare*T
    plot(t[:,0], t[:,1], '-', **kwargs)
```

In [2]:
```
fig=figure(1, figsize=(5,5)); clf()
plot_square(R(0),   color='k', label='original')
plot_square(R(0.5), color='g', label='rotation(0.5)')
plot_square(B(0.5), color='r', label='boost(0.5)')
setp(gca(), xlim=(-1,2), ylim=(-1,2), xlabel='space', ylabel='time or space',
     xticks=[],yticks=[], title='compare rotation with boost')
legend(loc='upper left')
display(fig)
```




space