[1]:

```
from __future__ import division, print_function, unicode_literals
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
np.random.seed(2)
```

GD Momentum ¶

[3]:

```
X = np.random.rand(1000, 1)
y = 4 + 3 * X + .2*np.random.randn(1000, 1)
```

[4]:

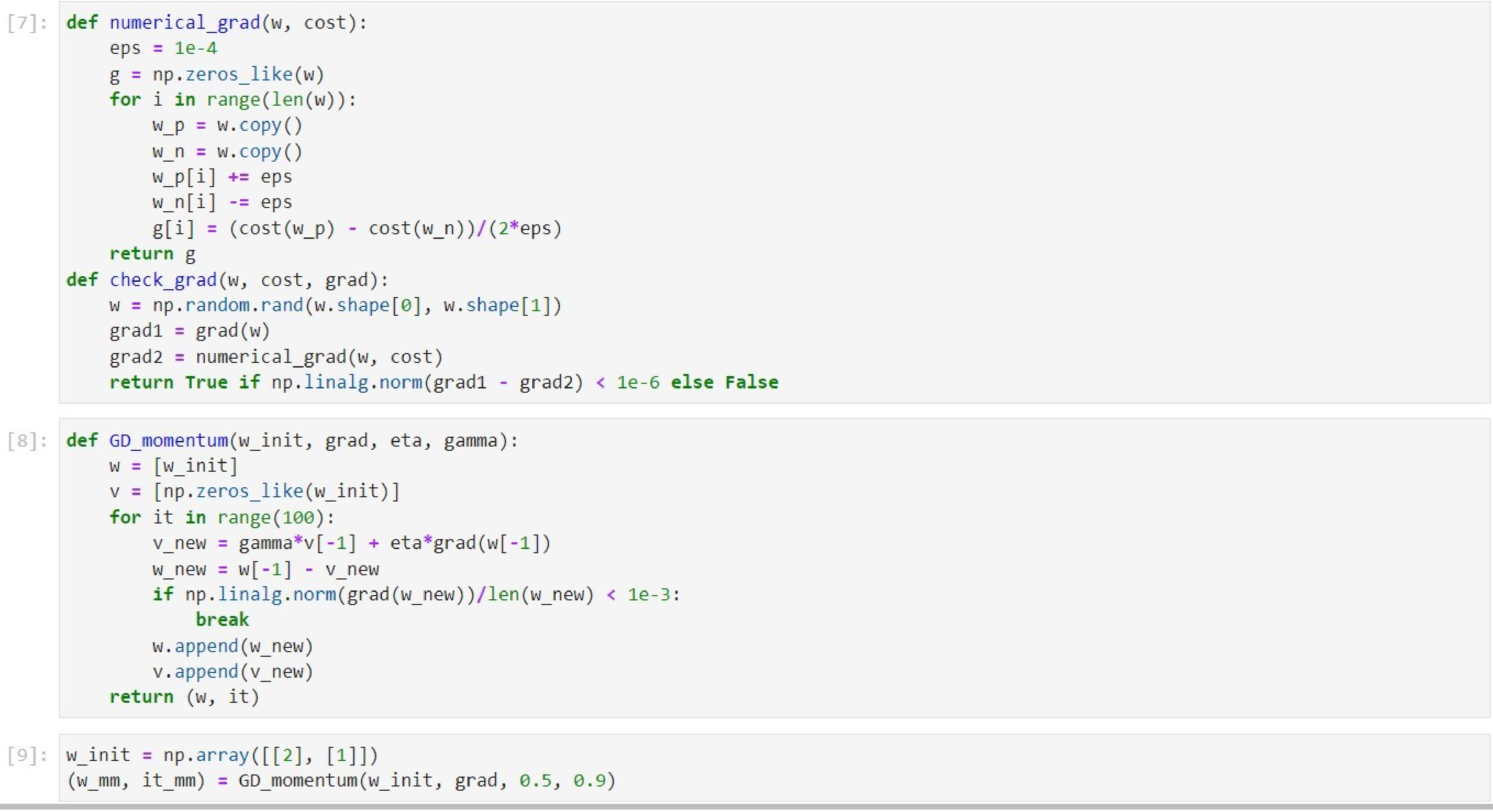
```
one = np.ones((X.shape[0],1))
Xbar = np.concatenate((one, X), axis = 1)
```

[5]:

```
A = np.dot(Xbar.T, Xbar)
b = np.dot(Xbar.T, y)
w_exact = np.dot(np.linalg.pinv(A), b)
```

[6]:

```
def grad(w):
    N = Xbar.shape[0]
    return 1/N * Xbar.T.dot(Xbar.dot(w) - y)

def cost(w):
    N = Xbar.shape[0]
    return .5/N*np.linalg.norm(y - Xbar.dot(w), 2)**2;
```



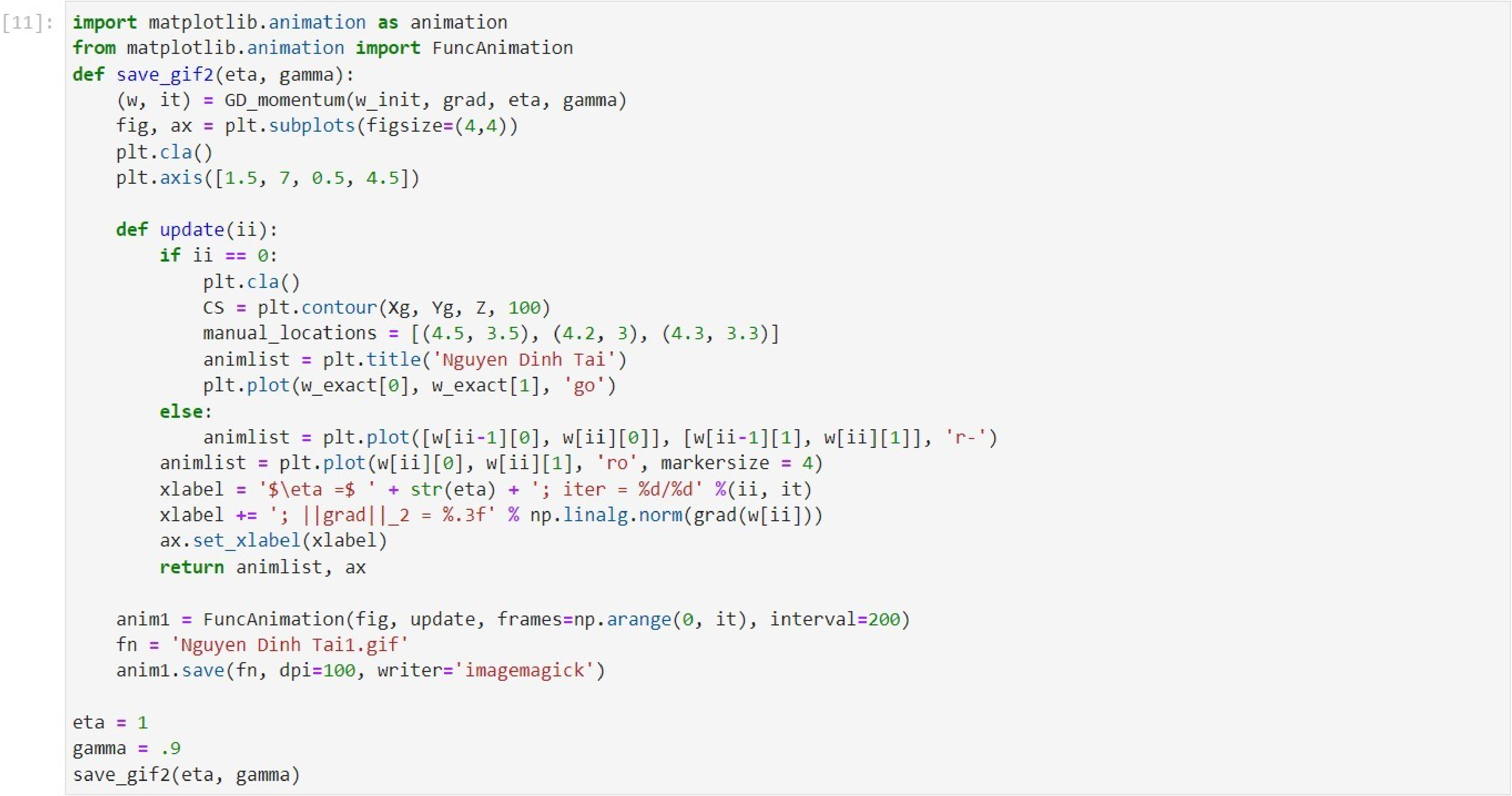
```
[7]: def numerical_grad(w, cost):
         eps = 1e-4
         g = np.zeros_like(w)
         for i in range(len(w)):
             w_p = w.copy()
             w_n = w.copy()
             w_p[i] += eps
             w_n[i] -= eps
             g[i] = (cost(w_p) - cost(w_n))/(2*eps)
         return g
     def check_grad(w, cost, grad):
         w = np.random.rand(w.shape[0], w.shape[1])
         grad1 = grad(w)
         grad2 = numerical_grad(w, cost)
         return True if np.linalg.norm(grad1 - grad2) < 1e-6 else False
```

```
[8]: def GD_momentum(w_init, grad, eta, gamma):
         w = [w_init]
         v = [np.zeros_like(w_init)]
         for it in range(100):
             v_new = gamma*v[-1] + eta*grad(w[-1])
             w_new = w[-1] - v_new
             if np.linalg.norm(grad(w_new))/len(w_new) < 1e-3:
                 break
             w.append(w_new)
             v.append(v_new)
         return (w, it)
```

```
[9]: w_init = np.array([[2], [1]])
     (w_mm, it_mm) = GD_momentum(w_init, grad, 0.5, 0.9)
```

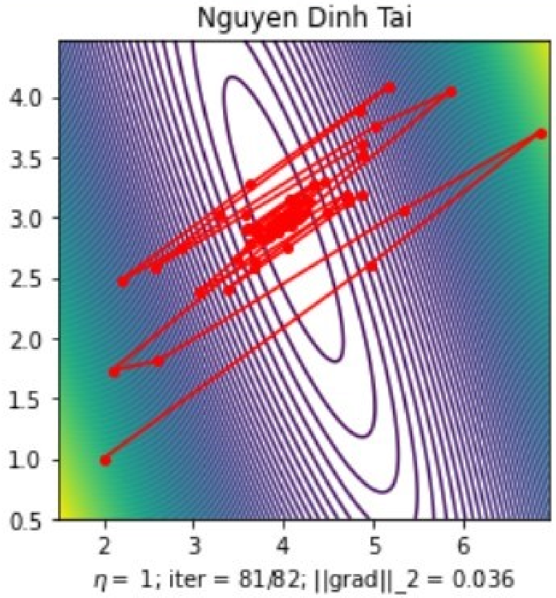
[10]:

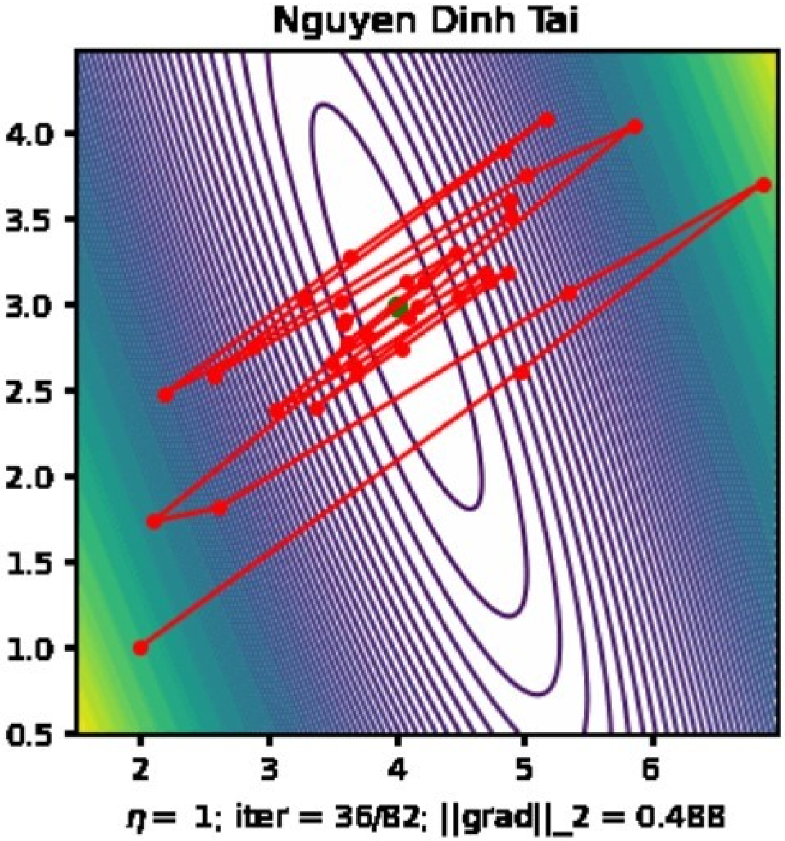
```
N = X.shape[0]
a1 = np.linalg.norm(y, 2)**2/N
b1 = 2*np.sum(X)/N
c1 = np.linalg.norm(X, 2)**2/N
d1 = -2*np.sum(y)/N
e1 = -2*X.T.dot(y)/N

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

delta = 0.025
xg = np.arange(1.5, 7.0, delta)
yg = np.arange(0.5, 4.5, delta)
Xg, Yg = np.meshgrid(xg, yg)
Z = a1 + Xg**2 +b1*Xg*Yg + c1*Yg**2 + d1*Xg + e1*Yg
```

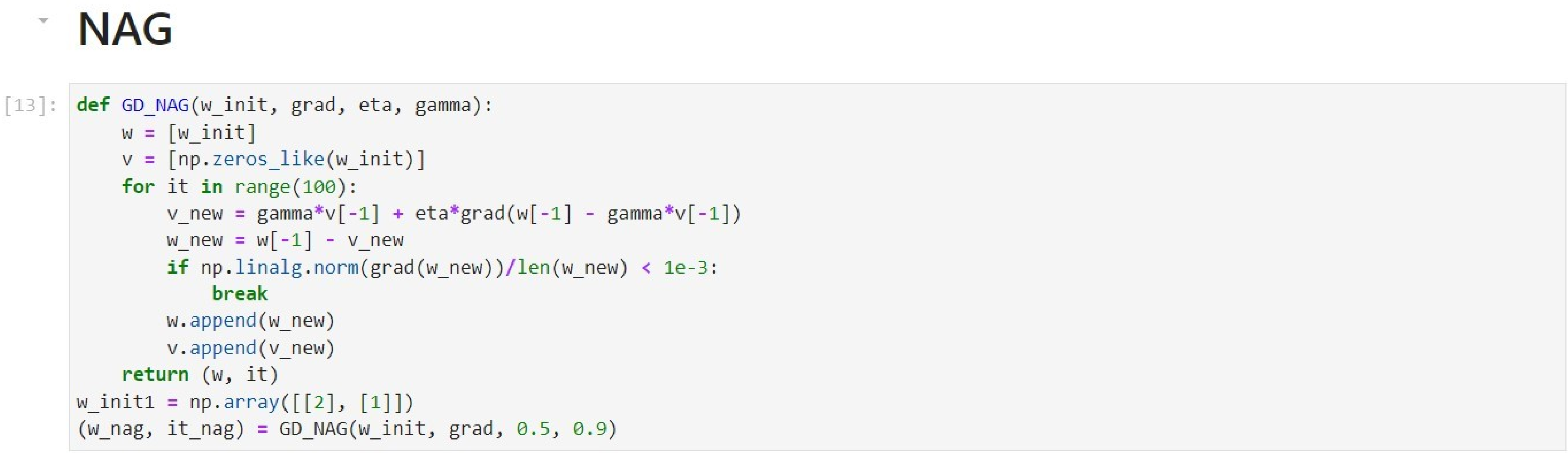


```
[11]: import matplotlib.animation as animation
      from matplotlib.animation import FuncAnimation
      def save_gif2(eta, gamma):
          (w, it) = GD_momentum(w_init, grad, eta, gamma)
          fig, ax = plt.subplots(figsize=(4,4))
          plt.cla()
          plt.axis([1.5, 7, 0.5, 4.5])

          def update(ii):
              if ii == 0:
                  plt.cla()
                  CS = plt.contour(Xg, Yg, Z, 100)
                  manual_locations = [(4.5, 3.5), (4.2, 3), (4.3, 3.3)]
                  animlist = plt.title('Nguyen Dinh Tai')
                  plt.plot(w_exact[0], w_exact[1], 'go')
              else:
                  animlist = plt.plot([w[ii-1][0], w[ii][0]], [w[ii-1][1], w[ii][1]], 'r-')
              animlist = plt.plot(w[ii][0], w[ii][1], 'ro', markersize = 4)
              xlabel = '$\eta =$ ' + str(eta) + '; iter = %d/%d' %(ii, it)
              xlabel += '; ||grad||_2 = %.3f' % np.linalg.norm(grad(w[ii]))
              ax.set_xlabel(xlabel)
              return animlist, ax

          anim1 = FuncAnimation(fig, update, frames=np.arange(0, it), interval=200)
          fn = 'Nguyen Dinh Tai1.gif'
          anim1.save(fn, dpi=100, writer='imagemagick')

      eta = 1
      gamma = .9
      save_gif2(eta, gamma)
```

```
MovieWriter imagemagick unavailable; using Pillow instead.
```



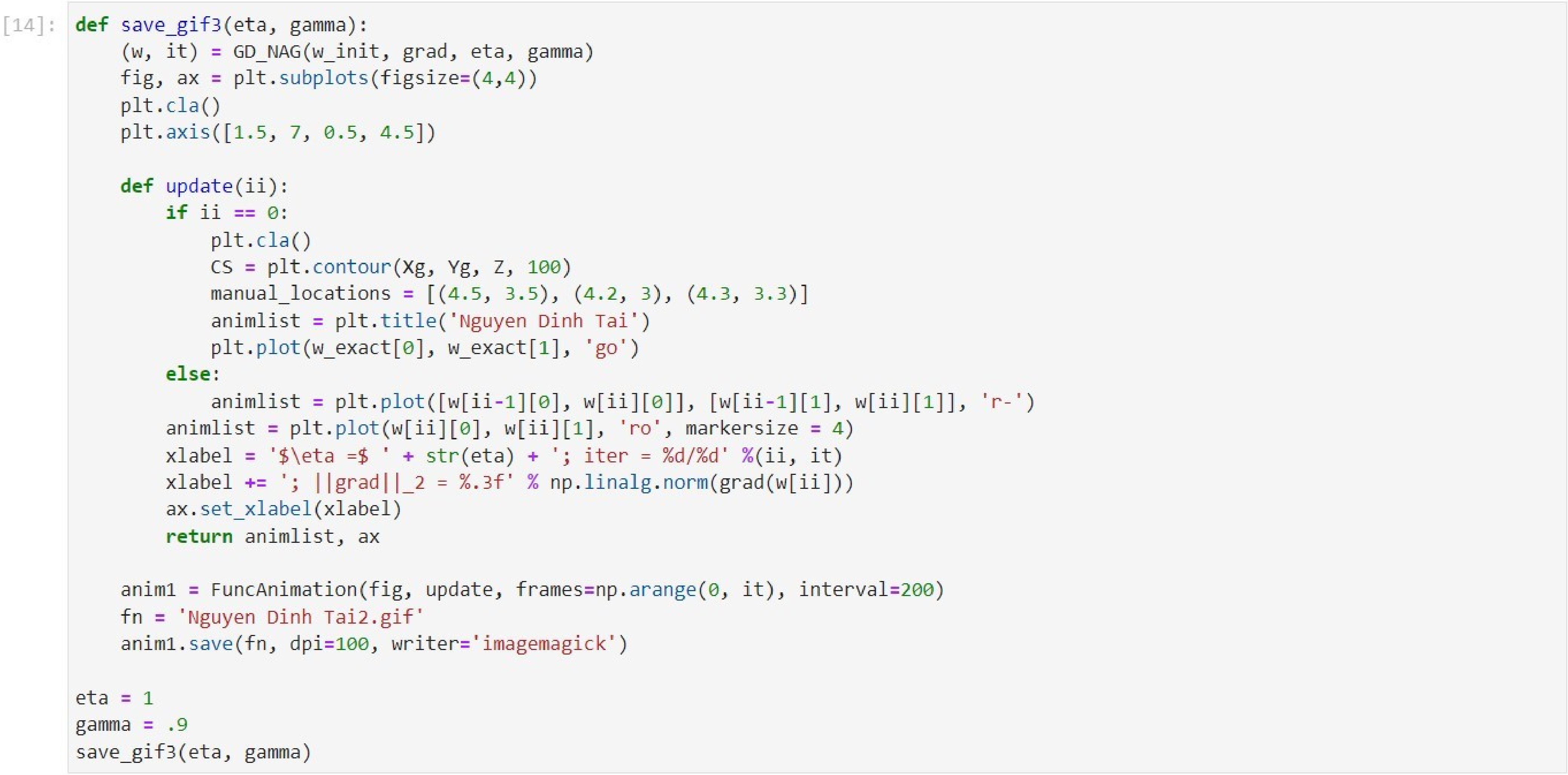

Nguyen Dinh Tai
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
2
3
4
5
6
η= 1; iter = 36/82; ||grad||_2 = 0.488



NAG

[13]:

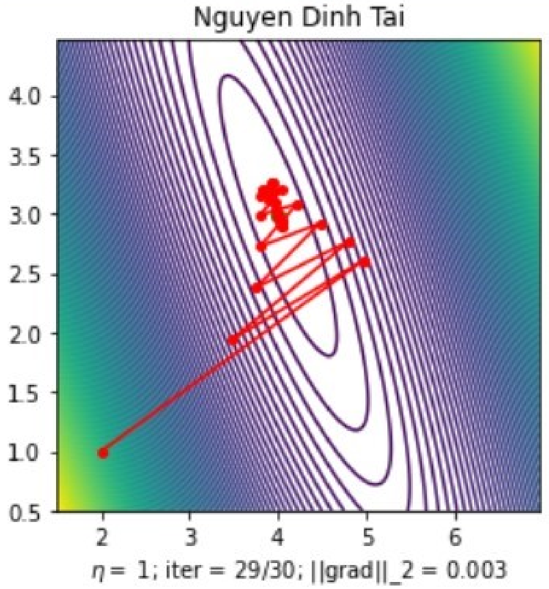
```
def GD_NAG(w_init, grad, eta, gamma):
    w = [w_init]
    v = [np.zeros_like(w_init)]
    for it in range(100):
        v_new = gamma*v[-1] + eta*grad(w[-1] - gamma*v[-1])
        w_new = w[-1] - v_new
        if np.linalg.norm(grad(w_new))/len(w_new) < 1e-3:
            break
        w.append(w_new)
        v.append(v_new)
    return (w, it)
w_init1 = np.array([[2], [1]])
(w_nag, it_nag) = GD_NAG(w_init, grad, 0.5, 0.9)
```

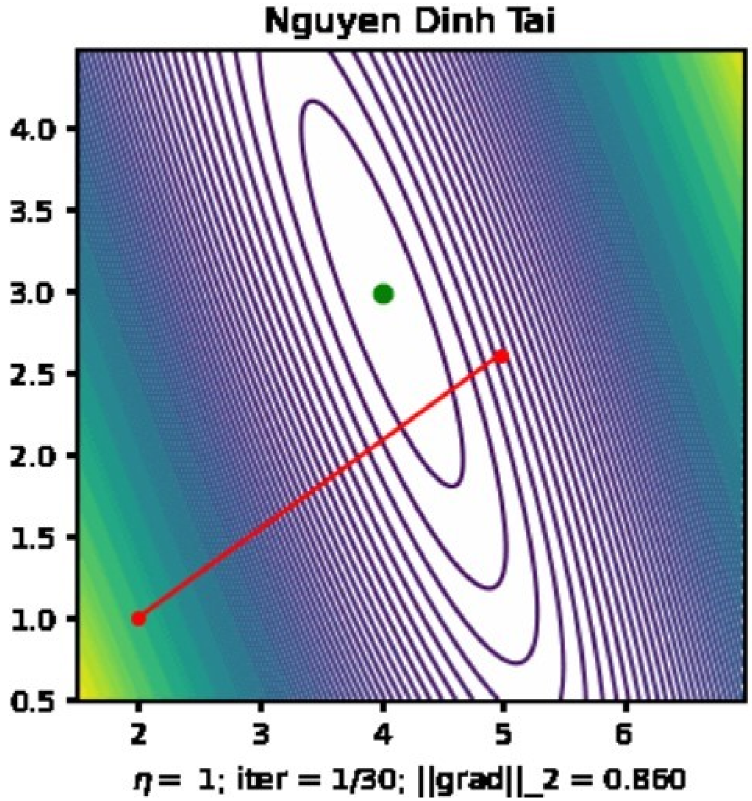


[14]:

```
def save_gif3(eta, gamma):
    (w, it) = GD_NAG(w_init, grad, eta, gamma)
    fig, ax = plt.subplots(figsize=(4,4))
    plt.cla()
    plt.axis([1.5, 7, 0.5, 4.5])

    def update(ii):
        if ii == 0:
            plt.cla()
            CS = plt.contour(Xg, Yg, Z, 100)
            manual_locations = [(4.5, 3.5), (4.2, 3), (4.3, 3.3)]
            animlist = plt.title('Nguyen Dinh Tai')
            plt.plot(w_exact[0], w_exact[1], 'go')
        else:
            animlist = plt.plot([w[ii-1][0], w[ii][0]], [w[ii-1][1], w[ii][1]], 'r-')
        animlist = plt.plot(w[ii][0], w[ii][1], 'ro', markersize = 4)
        xlabel = '$\eta =$ ' + str(eta) + '; iter = %d/%d' %(ii, it)
        xlabel += '; ||grad||_2 = %.3f' % np.linalg.norm(grad(w[ii]))
        ax.set_xlabel(xlabel)
        return animlist, ax

    anim1 = FuncAnimation(fig, update, frames=np.arange(0, it), interval=200)
    fn = 'Nguyen Dinh Tai2.gif'
    anim1.save(fn, dpi=100, writer='imagemagick')

eta = 1
gamma = .9
save_gif3(eta, gamma)
```

```
MovieWriter imagemagick unavailable; using Pillow instead.
```





Nguyen Dinh Tai