Activity 17

Setup

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
def prxgraddescent_l2(X,y,tau,lam,w_init,it):

## compute it iterations of L2 proximal gradient descent starting at w1
## w_{k+1}= (w_k - tau*X'*(X*w_k - y)/(1+lam*tau)
## step size tau
    W = np.zeros((w_init.shape[0], it+1))
    Z = np.zeros((w_init.shape[0], it+1))
    W[:,[0]] = w_init
    for k in range(it):
        Z[:,[k+1]] = W[:,[k]] - tau * X.T @ (X @ W[:,[k]] - y);
        W[:,[k+1]] = Z[:,[k+1]]/(1+lam*tau)

    return W,Z
```

In [3]:
```
## Proximal gradient descent trajectories
## Least Squares Problem
U = np.array([[1, 0], [0, 1], [0, 0], [0, 0]])
S = np.array([[1, 0], [0, 0.5]])
Sinv = np.linalg.inv(S)
V = 1/np.sqrt(2)*np.array([[1, 1], [1, -1]])
y = np.array([[np.sqrt(2)], [0], [1], [0]])

X = U @ S @ V.T

### Find Least Squares Solution
w_ls = V @ Sinv @ U.T @ y
c = y.T @ y - y.T @ X @ w_ls

### Find values of f(w), the contour plot surface for
w1 = np.arange(-1,3,.1)
w2 = np.arange(-1,3,.1)
fw = np.zeros((len(w1), len(w2)))
for i in range(len(w2)):
    for j in range(len(w1)):
        w = np.array([ [w1[j]], [w2[i]] ])
        fw[i,j] = (w-w_ls).T @ X.T @ X @ (w-w_ls) + c

# 1/ ||A||op**2 = 1/sigma1**2
```

```
C:\Users\ftstc\AppData\Local\Temp\ipykernel_7724\2784363303.py:22: DeprecationWarning: Conversion of an array with ndim > 0 to a scalar is deprecated, and will e
rror in future. Ensure you extract a single element from your array before performing this operation. (Deprecated NumPy 1.25.)
  fw[i,j] = (w-w_ls).T @ X.T @ X @ (w-w_ls) + c
```

Question 3a)

What is the maximum value for the step size τ that will guarantee convergence?

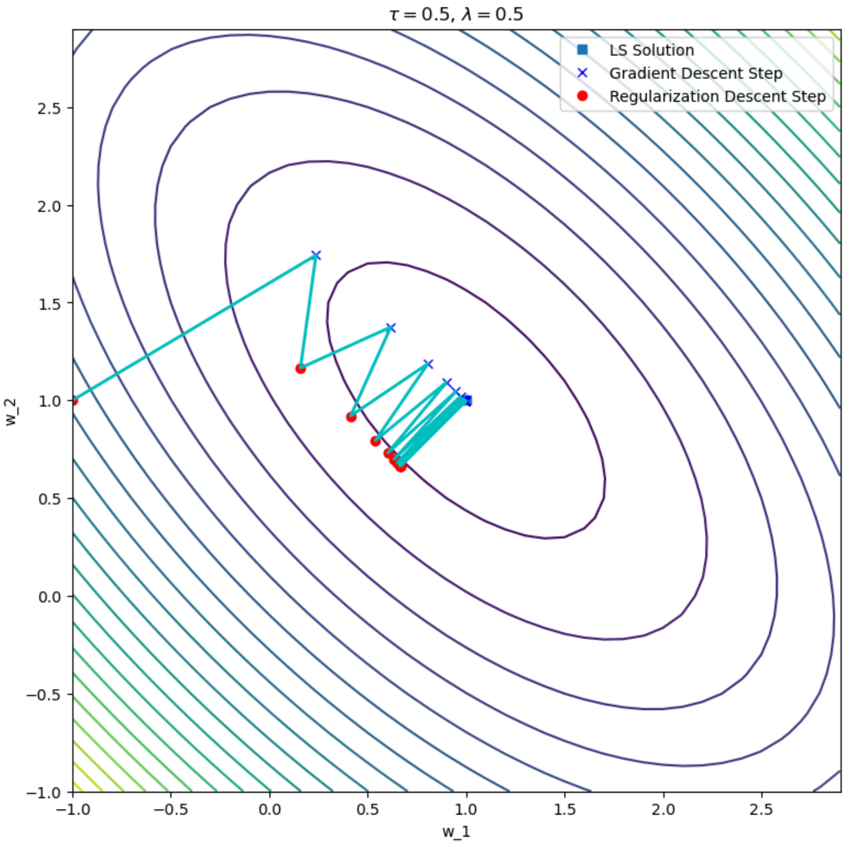
τ should be less than 1/sigma1**2 = 1.

In [23]:
```
## Find and display weights generated by gradient descent
w_init = np.array([[-1],[1]])
lam = 0.5;
it = 20
# tau = 0.5
# tau = 1
tau = 0.99
W,Z = prxgraddescent_l2(X,y,tau,lam,w_init,it)

# Concatenate gradient and regularization steps to display trajectory
G = np.zeros((2,0))
print(G.shape)
for i in range(it):
    G = np.hstack((G,np.hstack((W[:,[i]],Z[:,[i+1]]))))

plt.figure(figsize=(9,9))
plt.contour(w1,w2,fw,20)
plt.plot(w_ls[0],w_ls[1],"s", label="LS Solution")
plt.plot(Z[0,1::],Z[1,1:],'bx',linewidth=2, label="Gradient Descent Step")
plt.plot(W[0,:],W[1,:],'ro',linewidth=2, label="Regularization Descent Step")
plt.plot(G[0,:],G[1,:],'-c',linewidth=2)
plt.legend()
plt.xlabel('w_1')
plt.ylabel('w_2')
plt.title('$\\tau = $'+str(.5)+', $\lambda = $'+str(lam));
print(G.shape)
```

```
(2, 0)
(2, 40)
```



Question 3b)

Start proximal gradient descent from the point w = [−1 1] using a step size of τ = 0.5 and tuning parameter λ = 0.5. How do you explain the trajectory the weights take toward the optimum, e.g., why is it shaped this way? What direction does each iteration move in the regularization step

--> In each iteration, 2 operations happen, Gradient descent (z(k) calculation blue) and Regularization descent (w(k+1) calculation red), and the line goes through the 2 points also goes through origin, w(k+1) is shrinked from z(k).?

Question 3c)

Repeat the previous case with λ = 0.1 What happens? How does λ affect each iteration and why?

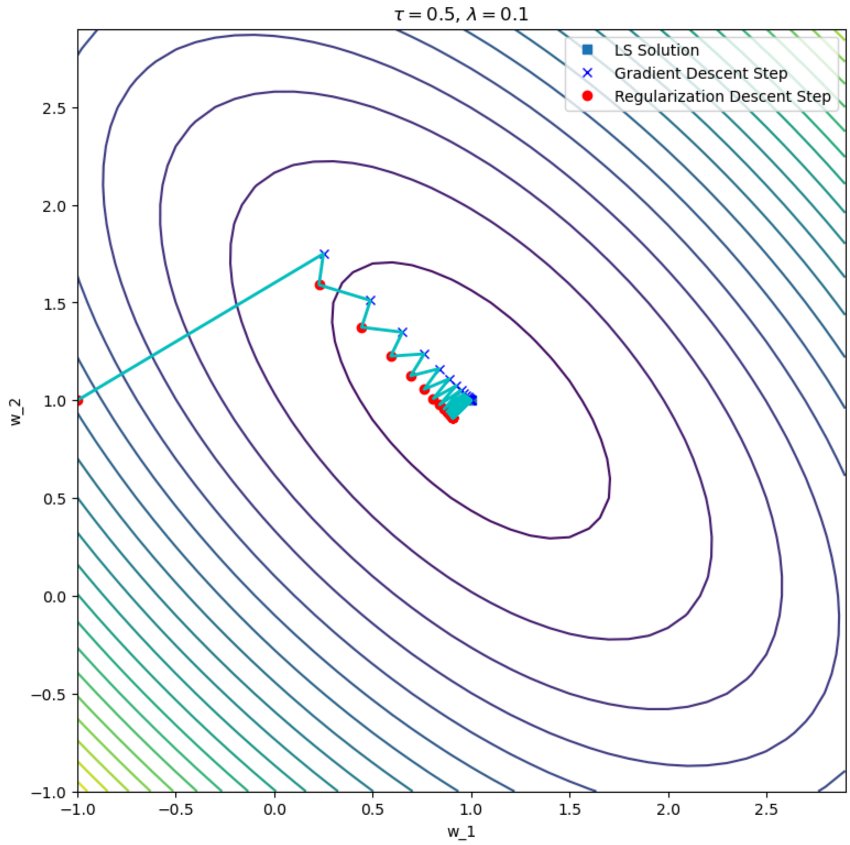
--> In each iteration, the step distance turns smaller.

In [15]:
```
## Find and display weights generated by gradient descent

w_init = np.array([[-1],[1]])
lam = 0.1
it = 20
tau = 1
W,Z = prxgraddescent_l2(X,y,tau,lam,w_init,it)

# Concatenate gradient and regularization steps to display trajectory
G = np.zeros((2,0))
for i in range(it):
    G = np.hstack((G,np.hstack((W[:,[i]],Z[:,[i+1]]))))

plt.figure(figsize=(9,9))
plt.contour(w1,w2,fw,20)
plt.plot(w_ls[0],w_ls[1],"s", label="LS Solution")
plt.plot(Z[0,1::],Z[1,1:],'bx',linewidth=2, label="Gradient Descent Step")
plt.plot(W[0,:],W[1,:],'ro',linewidth=2, label="Regularization Descent Step")
plt.plot(G[0,:],G[1,:],'-c',linewidth=2)
plt.legend()
plt.xlabel('w_1')
plt.ylabel('w_2')
plt.title('$\\tau = $'+str(.5)+', $\lambda = $'+str(lam));
```


$\tau = 0.5$, $\lambda = 0.1$
LS Solution
Gradient Descent Step
Regularization Descent Step
w_1
w_2
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

In [ ]:

1. a) $(A^T A A^T + \lambda A^T) = (A^T A + \lambda I) A^T = A^T (A A^T + \lambda I)$

$\Rightarrow (A^T A + \lambda I)^{-1} A^T (A A^T + \lambda I) = (A^T A + \lambda I)^{-1} \cancel{A^T (A A^T + \lambda I)} \underbrace{(A^T A + \lambda I) A^T}_{I}$

$\Rightarrow (A^T A + \lambda I)^{-1} A^T \underbrace{(A A^T + \lambda I)(A A^T + \lambda I)^{-1}}_{I} = A^T (A A^T + \lambda I)^{-1}$

$\Rightarrow (A^T A + \lambda I)^{-1} A^T = A^T (A A^T + \lambda I)^{-1}$

b) $w = \cancel{(A^T A)^{-1} A^T y}$

$w = (A^T A + \lambda I)^{-1} A^T y \rightarrow (100 \times 100 + \lambda I)^{-1} \; 100 \times 1$ → faster, because of smaller dim

$= A^T (A A^T + \lambda I)^{-1} y \rightarrow (8000 \times 8000 + \lambda I)^{-1} \; 8000 \times 1$

~~8000×100~~

100×8000

A: 8000×100

y: 8000×1

c) g_i & w: 8000×1

$\hat{y}_i = \text{sign}\{g_i^T w\}, \; i = 1, 2 \ldots 1000$

~~the most~~ $\text{rank}(G) \leq 100 \leq \cancel{\text{col}(G)}$ $\leq w: 8000 \times 1$

~~8000~~

G: 8000×100

LS $= (G^T G)^{-1} G^T y = w$

8000×8000

i) $Gw = y$ $\;\;\cancel{100 \times 1}$

$100 \times \cancel{8000} \; \cancel{8000 \times 1}$

→ Not unique sol

Rig-reg:

ii) $G^T (\underline{G G^T} + \lambda I)^{-1} y = w$ $\quad \text{rank}(G) \leq 100 = w: 100 \times 1$

100×100

→ unique

2. a) $\cancel{\sum_{i=1} (z_i - w_i)^2 + \lambda \tau w_i^2}$

$w^{(k+1)} = \arg\min_{w_i, i=1,\ldots,M} \sum_{i=1}^{M} (z_i^{(k)} - w_i)^2 + \lambda \tau w_i^2 \Rightarrow w_i^{(k+1)} = \frac{1}{1 + \lambda \tau} z_i^{(k)}$

b) $w^{(k+1)} = \arg\min_{w_i, i=1\ldots M} \sum_{i=1}^{M} (z_i^{(k)} - w_i)^2 + \lambda \tau w_i$