Porblem 2 (30 points)

Problem Description

In this problem you will use linear least squares to fit a linear function to a 3D temperature field, with x,y,z locations and an associated temperature T.

Fill out the notebook as instructed, making the requested plots and printing necessary values.

You are welcome to use any of the code provided in the lecture activities.

Summary of deliverables:

Results:

- Predicted temperature `T(5,5,5)` using a hand-coded LLS squares model with a linear function
- Direction of travel from `(5,5,5)` to experience the greatest decrease in temperature

Discussion:

- Reasoning for how we can use our fitted function to determine the direction of greatest decrease in temperature

Imports and Utility Functions:

In [87]:
```
import numpy as np
```

Load the data

The data is contained in `tempfield.npy` and can be loaded with `np.load(tempfield.npy)`. The first three columns correspond to the `x`, `y`, and `z` locations of the data points, and the 4th column corresponds to the temperature `T` at the respective point. Store the data as you see fit.

In [88]:
```
# YORU CODE GOES HERE
tempfield = np.load('tempfield.npy')
print(tempfield)
```

```
[[ 0  0  0 10]
 [ 8  6  1 15]
 [ 5  2  8 20]
 [ 8  2  6 22]
 [ 5  1  2 16]
 [ 3  3  3 23]
 [ 9  8  2 18]
 [ 3  6  5 19]
 [ 4  6  9 25]
 [ 1  8  2 20]
 [ 1  1  2 28]
 [ 6  4  2 27]]
```

LLS Regression in 3D

Now fit a linear function to the data using the closed form of LLS regression. Use your fitted function to report the predicted temperature at `x = 5, y = 5, z = 5`. You are free to add regularization to your model, but this is not required and will not be graded.

In [89]:
```python
# YOUR CODE GOES HERE
x = tempfield[:,0]
y = tempfield[:,1]
z = tempfield[:,2]
t = tempfield[:,3]

import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

X = np.vstack([x,y,z,np.ones_like(x)]).T
w = (np.linalg.inv(X.T @ X) @ X.T @ t).reshape(4,1)
Xreg = np.array([5,5,5,1])
# Treg = Xreg @ w

def grad(w):
    # YOUR CODE GOES HERE
    # Return the gradient of the objective function
    w_grad = 2*X.T @ (X @ w - t.reshape(-1,1))
    return w_grad

eta = 0.00025
for i in range(101):

    # YOUR CODE GOES HERE
    # Update w
    w = w - eta*grad(w)
    if 0 == i%20:
        Treg = Xreg @ w

print(Treg)
```

```
[21.38907338]
```

Gradient Intuition

Using the function you fit in the previous part, in which direction should one move from the point `p = (5,5,5)` to experience the largest decrease in temperature in the immediate neighborhood of the point? Report the specific direction, along with your reasoning.

Your answer goes here

In [90]:
```
print(-w[:3])
```

```
[[ 0.13421269]
 [-0.11741862]
 [-0.72677374]]
```

Given by the results above, going toward [0.134,-0.117,-0.726] would result in the greatest temperature drop.