In [1]:

```
"""
Created on Mon Feb 19 17:00:37 2024

@author: flash
"""

import numpy as np

# Define the data
y = np.array([110, 140, 180, 190])
X = np.array([[180, 150], [150, 175], [170, 165], [185, 210]])

# Initialize theta using Ordinary Least Squares (OLS)
theta = np.linalg.inv(X.T @ X) @ X.T @ y

# Initialize Sigma as identity matrix (simplification for starting point)
Sigma = np.eye(4)

# Iteration parameters
max_iterations = 1000
tolerance = 1e-6
converged = False

for iteration in range(max_iterations):
    # Update Sigma based on current residuals (diagonal with squared resi
    residuals = y - X @ theta
    Sigma = np.diag(residuals**2)

    # Update theta (consider Sigma in the update rule if necessary, here
    # For a more complex model considering Sigma, use generalized least s
    theta_new = np.linalg.inv(X.T @ np.linalg.inv(Sigma) @ X) @ X.T @ np.

    # Check for convergence
    if np.linalg.norm(theta_new - theta) < tolerance:
        converged = True
        theta = theta_new
        break
    else:
        theta = theta_new

# Sigma diagonal (variances) update based on final residuals
final_residuals = y - X @ theta
Sigma_final = np.diag(final_residuals**2)

print(theta)
print( Sigma_final)
print( converged)
print( iteration)
```

```
[-0.53731343  1.37810945]
[[3.46563635e-24 0.00000000e+00 0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 4.23212916e+02 0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 1.93206171e+03 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 5.46068451e-25]]
True
4
```

In [ ]: