

```

import numpy as np

A = np.array([[6, -1, -1],
              [6, 9, 1],
              [-3, 1, 12]])
b = np.array([3, 40, 50])

solution = np.linalg.solve(A, b)
print(f'Solution using built-in function: x = {solution}')

def solve(A, b, x, tol=1e-6, max_iter=10):
    n = len(b)
    iter = 0
    while iter < max_iter:
        previous_x = x.copy() # Make a full copy of the current solution
        for i in range(n):
            sigma = 0
            for j in range(n):
                if i != j:
                    sigma += A[i][j] * x[j]
            x[i] = (b[i] - sigma) / A[i][i] # Gauss-Seidel update
        if max(np.abs(np.subtract(previous_x, x))) < tol:
            return x
        iter += 1

    print("Warning: Max iterations exceeded without convergence")
    return x

initial_guess = np.zeros(len(b))
x = solve(A, b, initial_guess, tol=0.0005, max_iter=20)
print(f'Solution using gauss-seidel: x = {x}')

'''If the equations are not reordered, the algorithm is not guaranteed to converge.
This emanates from updating of the x vector, where dividing a by a small diagonal value
A[i][i] will cause the x value to "blow up"
'''

```