线性方程组的迭代解法

实验内容

1. 编写高斯-塞德尔迭代和 SOR 迭代的通用程序。
   输入：矩阵 A 和向量 b, 迭代初值 x^0, 迭代最大步数 K, 误差控制 ϵ。对于超松弛迭代，还需输入松弛因子 ω。
   输出：迭代步数及方程 $Ax = b$ 的根值 x^*。
   (1) 要求根据给定的矩阵和向量进行测试，取初值 $x^(0)$，误差控制 $\epsilon = 10^{-8}$，打印出两种迭代方法的输出。
   (2) 取松弛因子 $\omega = \frac{i}{50} i = 1, 2, \ldots, 99$，打印迭代步数，并给一个最佳值。

算法实现

高斯-塞德尔算法实现如下所示：

```
# 高斯-塞德尔迭代
def gauss_seidel(self, x0, delta):
    x = x0
    count = 0
    while True:
        count += 1
        max_err = 0.0
        for i in range(self.n):
            old_sum = 0.0
            new_sum = 0.0
            for j in range(0, i):
                new_sum += self.matrix[i][j] * x[j]
            for j in range(i + 1, self.n):
                old_sum += self.matrix[i][j] * x[j]
            old_xi = x[i]
            x[i] = (self.vector[i] - old_sum - new_sum)/self.matrix[i][i]
            err = abs(old_xi - x[i])
            max_err = max(max_err, err)
        if max_err < delta:
            break
        max_err = 0
    return (x, count)
```

在高斯-塞德尔迭代法中，我们首先将初始向量赋值给 x，随后进行迭代，直到计算的0范数小于给定的误差限时返回计算的向量和迭代次数。

SOR 迭代法的实现如下所示：

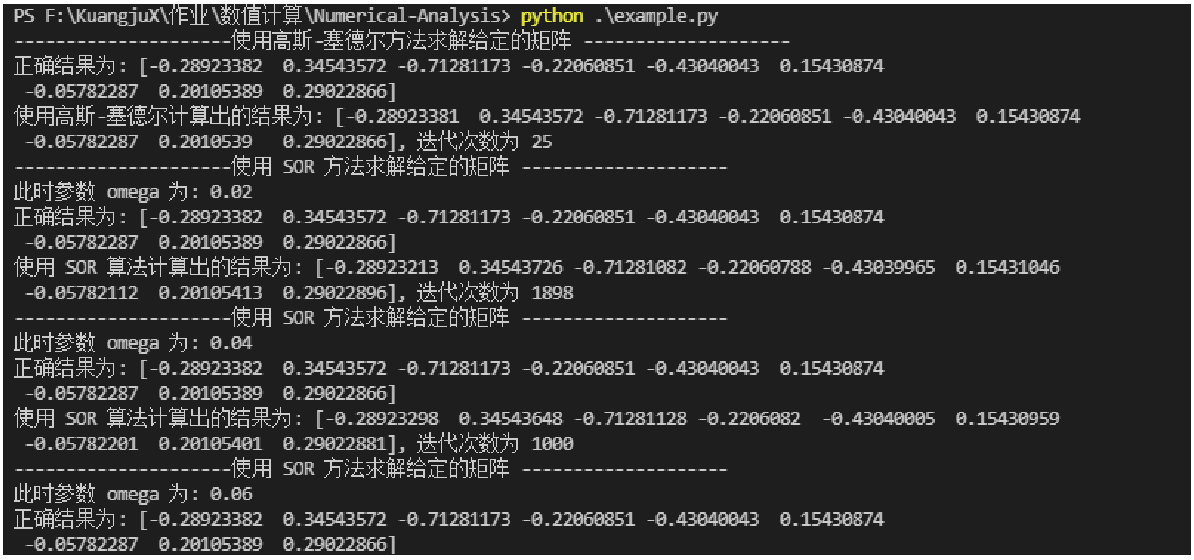
```
def sor(self, x0, omega, delta):
    x = x0
    count = 0
    while True:
        count += 1
        max_err = 0.0
        for i in range(self.n):
            old_sum = 0.0
```

```python
                new_sum = 0.0
                for j in range(0, i):
                    new_sum += self.matrix[i][j] * x[j]
                for j in range(i, self.n):
                    old_sum += self.matrix[i][j] * x[j]
                old_xi = x[i]
                x[i] = x[i] + omega * (self.vector[i] - old_sum -
new_sum)/self.matrix[i][i]
                err = abs(x[i] - old_xi)
                max_err = max(max_err, err)
            if max_err < delta:
                break
            max_err = 0
        return (x, count)
```

大体上与 高斯-塞德尔 迭代相同，只不过每次迭代时都要乘上一个给定的系数 ω。

实验结果

高斯-塞德尔：



```
PS F:\KuangjuX\作业\数值计算\Numerical-Analysis> python .\example.py
--------------------使用高斯-塞德尔方法求解给定的矩阵 --------------------
正确结果为: [-0.28923382  0.34543572 -0.71281173 -0.22060851 -0.43040043  0.15430874
 -0.05782287  0.20105389  0.29022866]
使用高斯-塞德尔计算出的结果为: [-0.28923381  0.34543572 -0.71281173 -0.22060851 -0.43040043  0.15430874
 -0.05782287  0.2010539   0.29022866], 迭代次数为 25
--------------------使用 SOR 方法求解给定的矩阵 --------------------
此时参数 omega 为: 0.02
正确结果为: [-0.28923382  0.34543572 -0.71281173 -0.22060851 -0.43040043  0.15430874
 -0.05782287  0.20105389  0.29022866]
使用 SOR 算法计算出的结果为: [-0.28923213  0.34543726 -0.71281082 -0.22060788 -0.43039965  0.15431046
 -0.05782112  0.20105413  0.29022896], 迭代次数为 1898
--------------------使用 SOR 方法求解给定的矩阵 --------------------
此时参数 omega 为: 0.04
正确结果为: [-0.28923382  0.34543572 -0.71281173 -0.22060851 -0.43040043  0.15430874
 -0.05782287  0.20105389  0.29022866]
使用 SOR 算法计算出的结果为: [-0.28923298  0.34543648 -0.71281128 -0.2206082  -0.43040005  0.15430959
 -0.05782201  0.20105401  0.29022881], 迭代次数为 1000
--------------------使用 SOR 方法求解给定的矩阵 --------------------
此时参数 omega 为: 0.06
正确结果为: [-0.28923382  0.34543572 -0.71281173 -0.22060851 -0.43040043  0.15430874
 -0.05782287  0.20105389  0.29022866]
```

SOR 迭代法：

```
最少迭代次数为: 14, 此时 omega 为: 1.18
```

实验结果如上图所示，SOR 迭代法的最佳值为 $/omega$ 取 1.18 的时候，此时最小迭代次数为 14