运用雅克比和高斯赛德尔公式的求解方程组

21 级计算机科学与技术 陈岳阳

2022 年 12 月 3 日

目录

1 线性方程组的迭代法

线性方程组的解法可分为直接法和迭代法。迭代法的算法简单，编制程序比较容易，但是对于系数矩阵有特殊要求。

通常，迭代法使用新值和老值的偏差刻画精度，且设置最大迭代次数防止迭代过程发散或收敛过于缓慢。

1.1 雅可比迭代法

考察一般形式的线性方程组

$$\sum_{j=1}^{n} a_{ij}x_j = b_i,\ k = 1, 2, \ldots, n$$

分离出变量 x_i，将其改写为：

$$x_i = \frac{1}{a_{ii}}(b_i - \sum_{j=1,j\neq n}^{n} a_{ij}x_j),\ i = 1, 2, \ldots, n$$

由此可以建立出被称为雅可比迭代公式的迭代式：

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1,j\neq n}^{n} a_{ij}x_j^{(k)}),\ i = 1, 2, \ldots, n$$

1.2 高斯-塞德尔迭代

由于我们使用雅可比迭代法得到的新值比老值更准确，因此可以将其替代老值用于下一步的迭代，这就是高斯-塞德尔迭代的思想。

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)},\ i = 1, 2, \ldots, n$$

1.3 迭代法的收敛性

1.3.1 迭代公式的矩阵表示

线性方程组可以用矩阵记号简记为 Ax=b，其中系数矩阵 A 可以分解为对角矩阵、上三角矩阵和下三角矩阵，即：

$$A = D - L - U$$

采用上述记号，雅可比迭代公式可表示为：

$$x^{(k+1)} = (I - D^{-1}A)x^{(k)} + D^{-1}b$$

而高斯-塞德尔公式的矩阵形式是：

$$x^{(k+1)} = (D - L)^{-1}Ux^{(k)} + (D - L)^{-1}b$$

1.3.2 迭代公式的收敛性判定

可以看出，上面两种迭代方法的矩阵形式都是：

$$x = Gx + b$$

```
[25]: def Jacobi(A, B, n=100, *, epsilon=1e-6):
          #雅可比迭代法
          L, U, D = getL(A), getU(A), getD(A)
          InvD = np.linalg.inv(D)
          x = np.zeros((A.shape[0], 1)).reshape(A.shape[0], 1)
          #迭代矩阵
          Co = np.matmul(InvD, L+U)
          #判断收敛性
          if SR(Co) >= 1:
              print("谱半径为%f, 不满足收敛性条件"%SR(Co))
              return
          InvDB = np.matmul(InvD, B)
          for i in range(n):
              x1 = np.matmul(Co, x) + InvDB
              flag = False
              for j in range(x.shape[0]):
                  if abs(x1[j][0] - x[j][0]) > epsilon:
                      flag = True
                      break
              if flag == False:
                  print("迭代次数: %d"%i)
                  return x1
              x = x1
          print("迭代次数过多，收敛过于缓慢")
```

图 1: Jacobi 迭代法代码

若 $\|G\|<1$，则迭代公式对于一切初值均收敛。此条件的成立性可以使用类似压缩映射定理收敛速率的证明方法证明。

此外，上迭代公式收敛当且仅当 G 的谱半径 $\rho(G)<1$。此条件的充分性与上一条件等价。

如果仅考虑系数矩阵 A，当 A 是对角占优矩阵时，即满足

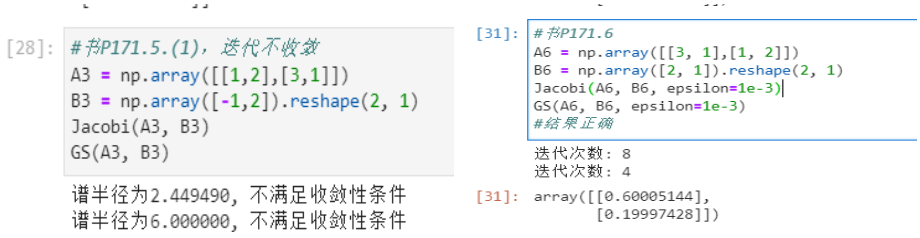
$$\sum_{j=1,j\neq i}^{n} |a_{ij}| < |a_{ii}|,\ i = 1, 2, \ldots, n$$

时，上述两种迭代方法的迭代过程对于任意初值收敛。对于两种迭代方法，该条件可以推出 $\|G\|_\infty<1$，因此正确性成立。

2 实验代码及解决问题

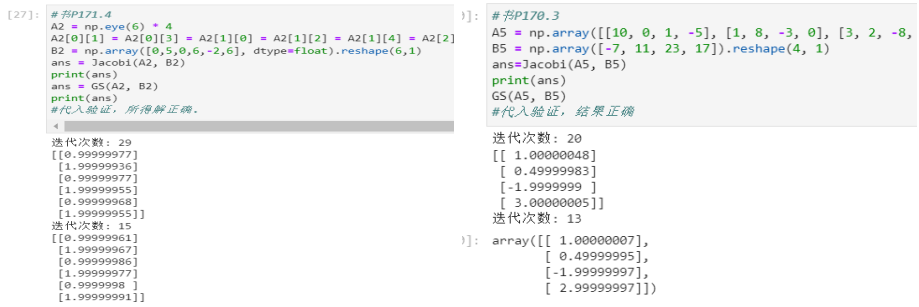
实验代码及详细注释在附件“陈岳阳-第五章实验.md”中，使用雅可比迭代法和高斯-塞德尔迭代法计算了书 P156. 例 1，P170.3、4、5、6 题。

两种迭代方法都先通过 DLU 矩阵计算迭代矩阵 Co(书上是 G)，然后通过谱半径判断收敛性 (谱半径和逆矩阵都通过 numpy 自带函数进行计算)。若迭代过程收敛，则对于给定的增广矩阵 (A,B)，进行最多 n 次迭代或迭代达到精度 epsilon 停止。若未达到最大迭代次数，则输出当前迭代次数，并返回解向量。否则，认为收敛过程过于缓慢。



(a) P171.5.(1) 测试结果　　(b) P171.6 测试结果

图 2: 收敛性的改变



(a) P170.3 测试结果　　(b) P171.4 测试结果

图 3: 收敛速率的不同

2.1 收敛性判定及收敛速率的比较

实验中，使用迭代矩阵的谱半径 $\rho(G)$ 判断迭代过程的收敛性。书 P171.5.(1) 和 P171.6 使用的方程组相同，仅交换了其中两行，却有不相同的谱半径，迭代过程收敛性也不同。这说明可以通过交换方程组的某些行改变迭代过程的收敛性。

而通过书 P170.3 和 P171.4 可以看出，通常雅可比迭代的收敛速率要慢于高斯-塞德尔方法。

3 实验体会

这次实验，我对高斯-塞德尔迭代法和雅可比迭代法有了更深入的了解，知道如何判断雅可比迭代和高斯-塞德尔迭代的收敛性，并用两种迭代方法求线性方程组的近似解。此外，这次试验还加强了我的代码能力，并加深了对 numpy 中矩阵操作的了解。