数值代数实验报告 4

Chase Young

2023 年 11 月 21 日

1 上机习题 1

1.1 问题描述

考虑两点边值问题

$$\begin{cases} \epsilon\dfrac{d^2y}{dx^2}+\dfrac{dy}{dx}=a, 0<a<1, \\ y(0)=0, y(1)=1 \end{cases} \tag{1}$$

容易知道它的精确解为

$$y=\frac{1-a}{1-e^{-\frac{1}{\epsilon}}}\left(1-e^{-\frac{x}{\epsilon}}\right)+ax \tag{2}$$

为了把微分方程离散化，把 [0, 1] 区间 n 等分，令 $h=\frac{1}{n}, x_i=ih, i=1,\cdots,n-1$, 得到差分方程

$$\epsilon\frac{y_{i-1}-2y_i+y_{i+1}}{h^2}+\frac{y_{i+1}-y_i}{h}=a$$

简化为

$$(\epsilon+h)y_{i+1}-(2\epsilon+h)y_i+\epsilon y_{i-1}=ah^2$$

离散化后得到线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$, 其中

$$\boldsymbol{A}=\begin{bmatrix} -(2\epsilon+h) & (\epsilon+h) & 0 & 0 & \cdots & 0 \\ \epsilon & -(2\epsilon+h) & (\epsilon+h) & 0 & \cdots & 0 \\ 0 & \epsilon & -(2\epsilon+h) & (\epsilon+h) & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & \epsilon & -(2\epsilon+h) & (\epsilon+h) \\ 0 & 0 & \cdots & 0 & \epsilon & -(2\epsilon+h) \end{bmatrix}$$

对 $\epsilon=1, a=1/2, n=100$, 分别用 Jacobi 迭代法,G-S 迭代法和 SOR 迭代法求线性方程组的解，要求 4 位有效数字，然后比较迭代次数，运行时间与精确解的误差。迭代法终止条件为 $\|x_{k+1}-x_k\|<10^{-6}$。

对 $\epsilon=0.1, 0.01, 0.0001$, 考虑同样的问题。要求输出计算结果，收敛所需要的迭代次数和运行时间。

1.2 程序介绍

本题将 2 阶 ODE 的两点边值问题(1)转化为求解一个稀疏的线性方程组。为保持稀疏性，分别采用 Jacobi 迭代法、Gauss-Seidel 迭代法、SOR 迭代法求解，并将求解结果与真实解(2)比较，计算在 2 范数意义下的误差。

代数方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 中，对比等式两边并结合两点边值，可得右端项为

$$\boldsymbol{b} = \left(ah^2, ah^2, \ldots, ah^2, ah^2 - \epsilon - h\right)^{\mathrm{T}}$$

具体实现的函数有：

- vector<double> Jacobi_iteration(vector<double> x, vector<vector<double> > B, vector<double> g);

  Jacobi 迭代法求解线性方程组，计算并返回迭代 1 步后的解；其中 x 是迭代前的解，B 是迭代矩阵，g 是迭代的常数向量

- void Jacobi_preparation(vector<vector<double> >& A, vector<double>& b);

  Jacobi 迭代法求解线性方程组的准备工作，计算迭代矩阵 B 和常数向量 b，分别保存在系数矩阵 A 和右端项 b 中

- vector<double> Gauss_Seidel_iteration(vector<double> x, vector<vector<double> > B, vector<double> g);

  Gauss-Seidel 迭代法求解线性方程组，计算并返回迭代 1 步后的解；其中 x 是迭代前的解，B 是迭代矩阵，g 是迭代的常数向量

- vector<double> SOR_iteration(vector<double> x, vector<vector<double> > A, vector<double> b, double omega);

  SOR 迭代法求解线性方程组，计算并返回迭代 1 步后的解；其中 x 是迭代前的解，A 是系数矩阵，b 是右端项，omega 是松弛因子

- vector<double> vec_sub(vector<double> a, vector<double> b);

  计算并返回向量 a, b 的差，要求向量 a, b 相同大小

1.3 实验结果

对 $a = 1/2, n = 100$，取 $\epsilon = 1$，运行结果如图 1、2、3 所示。

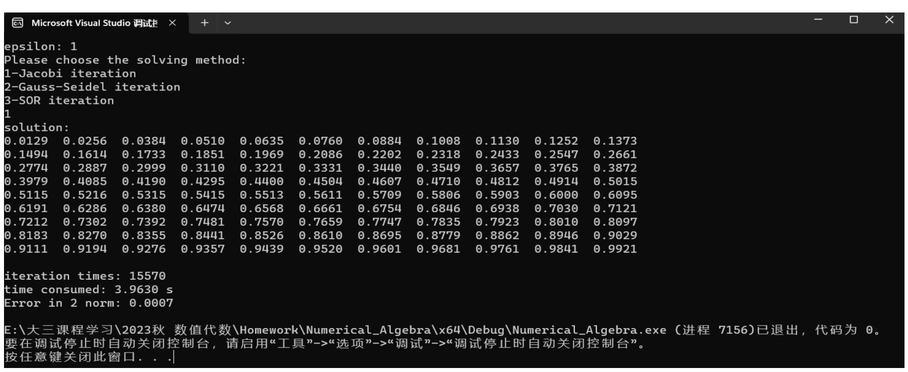
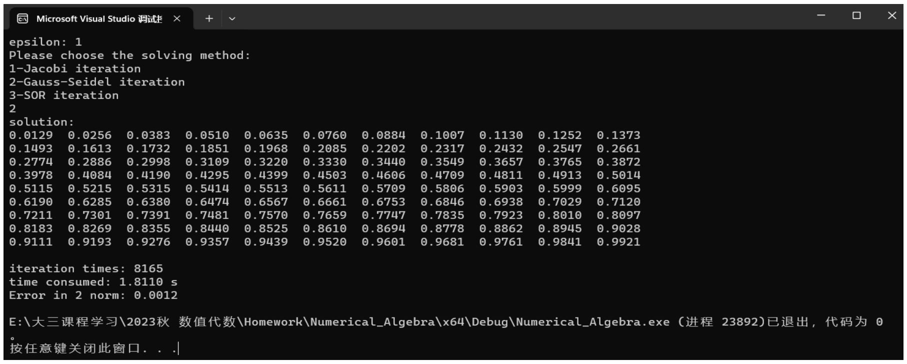


图 1: $\epsilon = 1$，Jacobi 迭代法

其中，与精确解的误差为 2 范数意义下的误差，即

$$error = \|\boldsymbol{y_{sol}} - \boldsymbol{y_{exact}}\|_2$$

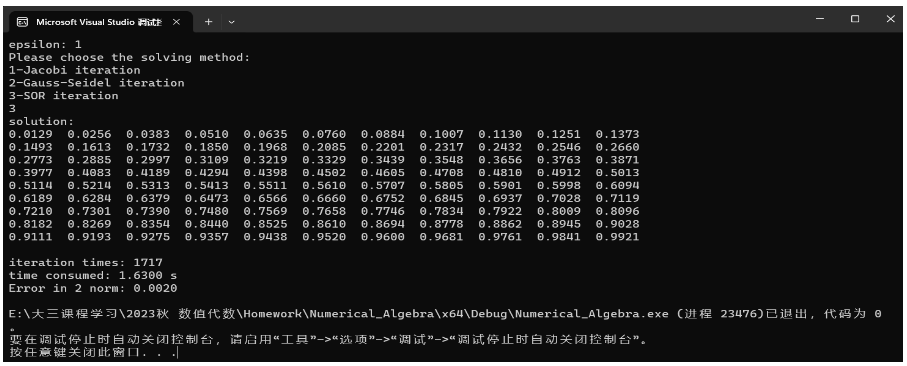


```
Microsoft Visual Studio 调试
epsilon: 1
Please choose the solving method:
1-Jacobi iteration
2-Gauss-Seidel iteration
3-SOR iteration
2
solution:
0.0129  0.0256  0.0383  0.0510  0.0635  0.0760  0.0884  0.1007  0.1130  0.1252  0.1373
0.1493  0.1613  0.1732  0.1851  0.1968  0.2085  0.2202  0.2317  0.2432  0.2547  0.2661
0.2774  0.2886  0.2998  0.3109  0.3220  0.3330  0.3440  0.3549  0.3657  0.3765  0.3872
0.3978  0.4084  0.4190  0.4295  0.4399  0.4503  0.4606  0.4709  0.4811  0.4913  0.5014
0.5115  0.5215  0.5315  0.5414  0.5513  0.5611  0.5709  0.5806  0.5903  0.5999  0.6095
0.6190  0.6285  0.6380  0.6474  0.6567  0.6661  0.6753  0.6846  0.6938  0.7029  0.7120
0.7211  0.7301  0.7391  0.7481  0.7570  0.7659  0.7747  0.7835  0.7923  0.8010  0.8097
0.8183  0.8269  0.8355  0.8440  0.8525  0.8610  0.8694  0.8778  0.8862  0.8945  0.9028
0.9111  0.9193  0.9276  0.9357  0.9439  0.9520  0.9601  0.9681  0.9761  0.9841  0.9921

iteration times: 8165
time consumed: 1.8110 s
Error in 2 norm: 0.0012

E:\大三课程学习\2023秋 数值代数\Homework\Numerical_Algebra\x64\Debug\Numerical_Algebra.exe (进程 23892)已退出，代码为 0
。
按任意键关闭此窗口. . .
```

图 2: $\epsilon = 1$，Gauss-Seidel 迭代法



```
Microsoft Visual Studio 调试
epsilon: 1
Please choose the solving method:
1-Jacobi iteration
2-Gauss-Seidel iteration
3-SOR iteration
3
solution:
0.0129  0.0256  0.0383  0.0510  0.0635  0.0760  0.0884  0.1007  0.1130  0.1251  0.1373
0.1493  0.1613  0.1732  0.1850  0.1968  0.2085  0.2201  0.2317  0.2432  0.2546  0.2660
0.2773  0.2885  0.2997  0.3109  0.3219  0.3329  0.3439  0.3548  0.3656  0.3763  0.3871
0.3977  0.4083  0.4189  0.4294  0.4398  0.4502  0.4605  0.4708  0.4810  0.4912  0.5013
0.5114  0.5214  0.5313  0.5413  0.5511  0.5610  0.5707  0.5805  0.5901  0.5998  0.6094
0.6189  0.6284  0.6379  0.6473  0.6566  0.6660  0.6752  0.6845  0.6937  0.7028  0.7119
0.7210  0.7301  0.7390  0.7480  0.7569  0.7658  0.7746  0.7834  0.7922  0.8009  0.8096
0.8182  0.8269  0.8354  0.8440  0.8525  0.8610  0.8694  0.8778  0.8862  0.8945  0.9028
0.9111  0.9193  0.9275  0.9357  0.9438  0.9520  0.9600  0.9681  0.9761  0.9841  0.9921

iteration times: 1717
time consumed: 1.6300 s
Error in 2 norm: 0.0020

E:\大三课程学习\2023秋 数值代数\Homework\Numerical_Algebra\x64\Debug\Numerical_Algebra.exe (进程 23476)已退出，代码为 0
。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口. . .
```

图 3: $\epsilon = 1$，SOR 迭代法，$\omega = 1.7$

对于 $\epsilon = 0.1, 0.01, 0.0001$，运行结果截图在附录 A 中展示。这里将上述迭代方法的迭代次数、运行时间、与精确解的误差列表如表 1、2、3、4 所示。

迭代方法	迭代次数	运行时间 (s)	与精确解的误差
Jacobi 迭代	15570	3.9630	0.0007
Gauss-Seidel 迭代	8165	1.8110	0.0012
SOR 迭代 ($\omega = 1.7$)	1717	1.6300	0.0020

表 1: $\epsilon = 1$，三种迭代方法的比较

1.4 结果分析

观察上述结果，可以发现对于 $\epsilon = 1, 0.1, 0.01$，Jacobi 迭代的次数、运行时间均高于 G-S 迭代和 SOR 迭代法，但求解精度略高于其余两种方法 (总体精度差别不大)；对于 $\epsilon = 0.0001$，SOR 迭代法的迭代次数和运行时间反而高于 Jacobi 迭代法和 G-S 迭代法，并且此时松弛因子 ω 不能太大 (实践表明 $\omega > 1.15$ 时算法不能收敛)。

迭代方法	迭代次数	运行时间 (s)	与精确解的误差
Jacobi 迭代	5460	1.6790	0.0378
Gauss-Seidel 迭代	2964	0.7600	0.0381
SOR 迭代 ($\omega = 1.7$)	623	0.5310	0.0383

表 2: $\epsilon = 0.1$，三种迭代方法的比较

迭代方法	迭代次数	运行时间 (s)	与精确解的误差
Jacobi 迭代	503	0.1180	0.0988
Gauss-Seidel 迭代	304	0.1250	0.0988
SOR 迭代 ($\omega = 1.7$)	160	0.1760	0.0988

表 3: $\epsilon = 0.01$，三种迭代方法的比较

2 上机习题 2

2.1 问题描述

考虑偏微分方程

$$-\Delta u + g(x,y)u = f(x,y), (x,y) \in [0,1]\times[0,1] \tag{3}$$

在 $[0,1]\times[0,1]$ 边界上 $u = 1$. 沿 x 方向和 y 方向均匀剖分 N 等份，令 $h = 1/N$, 并设应用中心差分离散化后得到差分方程的代数方程组为

$$-u_{i-1,j} - u_{i,j-1} + (4 + h^2 g(ih,jh))u_{i,j} - u_{i+1,j} - u_{i,j+1} = h^2 f(ih,jh) \tag{4}$$

取 $g(x,y) = \exp(xy), f(x,y) = x + y$, 分别用 Jacobi 迭代法，G-S 迭代法和 SOR 迭代法求解上述代数方程组，要求输出解的最小分量，并比较 $N = 20, 40, 60$ 时收敛所需要的迭代次数和运行时间，迭代终止条件为 $\|x_{k+1} - x_k\|_2 < 10^{-7}$。

在用 SOR 迭代法求解的过程中，请对不同的 N 使用合适的松弛因子 ω, 并在程序输出中打印松弛因子的值。观察运行结果后选取合适的。

2.2 程序介绍

2.2.1 Jacobi 迭代格式的推导

用 $\boldsymbol{u}^{(k)} \in \mathbb{R}^{(N-1)\times(N-1)}$ 表示第 k 步迭代后得到的解，记 $\boldsymbol{v} \in \mathbb{R}^{(N-1)^2}$ 表示将 $\boldsymbol{u}^{(k)}$ 展平得到的向量。根据 Jacobi 迭代的表达式，有

$$\boldsymbol{D}\boldsymbol{v}^{(k+1)} = (\boldsymbol{L} + \boldsymbol{U})\,\boldsymbol{v}^{(k)} + \boldsymbol{b}$$

其中 $\boldsymbol{D}, \boldsymbol{L}, \boldsymbol{U}$ 满足线性方程组

$$\boldsymbol{A}\boldsymbol{v} = \boldsymbol{b}, \quad \boldsymbol{A} = \boldsymbol{D} - \boldsymbol{L} - \boldsymbol{U}$$

从而

$$\boldsymbol{D}_{ii}\boldsymbol{v}_i^{(k+1)} = \boldsymbol{L}_{i1}\boldsymbol{v}_1^{(k)} + \cdots + \boldsymbol{L}_{i,i-1}\boldsymbol{v}_{i-1}^{(k)} + \boldsymbol{U}_{i,i+1}\boldsymbol{v}_{i+1}^{(k)} + \cdots + \boldsymbol{U}_{in}\boldsymbol{v}_n^{(k)} + \boldsymbol{b}_i$$

迭代方法	迭代次数	运行时间 (s)	与精确解的误差
Jacobi 迭代	112	0.0790	0.0050
Gauss-Seidel 迭代	106	0.0580	0.0050
SOR 迭代 ($\omega = 1.14$)	156	0.1560	0.0050

表 4: $\epsilon = 0.0001$，三种迭代方法的比较

回到矩阵表示，即为

$$\left(4 + h^2 g\left(ih, jh\right)\right) \boldsymbol{u}_{i,j}^{(k+1)} = \boldsymbol{u}_{i-1,j}^{(k)} + \boldsymbol{u}_{i,j-1}^{(k)} + \boldsymbol{u}_{i+1,j}^{(k)} + \boldsymbol{u}_{i,j+1}^{(k)} + h^2 f(ih, jh) \tag{5}$$

此即 Jacobi 迭代的迭代格式。

2.2.2 Gauss-Seidel 迭代格式的推导

用 $\boldsymbol{u}^{(k)} \in \mathbb{R}^{(N-1)\times(N-1)}$ 表示第 k 步迭代后得到的解，记 $\boldsymbol{v} \in \mathbb{R}^{(N-1)^2}$ 表示将 $\boldsymbol{u}^{(k)}$ 展平得到的向量。根据 Gauss-Seidel 迭代的表达式，有

$$\boldsymbol{D}\boldsymbol{v}^{(k+1)} = \boldsymbol{L}\boldsymbol{v}^{(k+1)} + \boldsymbol{U}\boldsymbol{v}^{(k)} + \boldsymbol{b}$$

其中 $\boldsymbol{D}, \boldsymbol{L}, \boldsymbol{U}$ 满足线性方程组

$$\boldsymbol{A}\boldsymbol{v} = \boldsymbol{b}, \quad \boldsymbol{A} = \boldsymbol{D} - \boldsymbol{L} - \boldsymbol{U}$$

从而

$$\boldsymbol{D}_{ii}\boldsymbol{v}_i^{(k+1)} = \boldsymbol{L}_{i1}\boldsymbol{v}_1^{(k+1)} + \cdots + \boldsymbol{L}_{i,i-1}\boldsymbol{v}_{i-1}^{(k+1)} + \boldsymbol{U}_{i,i+1}\boldsymbol{v}_{i+1}^{(k)} + \cdots + \boldsymbol{U}_{in}\boldsymbol{v}_n^{(k)} + \boldsymbol{b}_i$$

回到矩阵表示，即为

$$\left(4 + h^2 g\left(ih, jh\right)\right) \boldsymbol{u}_{i,j}^{(k+1)} = \boldsymbol{u}_{i-1,j}^{(k+1)} + \boldsymbol{u}_{i,j-1}^{(k+1)} + \boldsymbol{u}_{i+1,j}^{(k)} + \boldsymbol{u}_{i,j+1}^{(k)} + h^2 f(ih, jh) \tag{6}$$

此即 Jacobi 迭代的迭代格式。

2.2.3 SOR 迭代格式的推导

用 $\boldsymbol{u}^{(k)} \in \mathbb{R}^{(N-1)\times(N-1)}$ 表示第 k 步迭代后得到的解。注意到 SOR 迭代格式和 Gauss-Seidel 迭代格式之间的关系为

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^{(k)} + \omega(\boldsymbol{u}_{GS}^{(k+1)} - \boldsymbol{u}^{(k)})$$

其中 $\boldsymbol{u}_{GS}^{(k+1)}$ 表示对 $\boldsymbol{u}^{(k)}$ 用 1 步 Gauss-Seidel 迭代得到的矩阵。结合(6)可得 SOR 迭代法的迭代格式为

$$\left(4 + h^2 g\left(ih, jh\right)\right) \boldsymbol{u}_{ij}^{(k+1)} = (1-\omega)\left(4 + h^2 g\left(ih, jh\right)\right) \boldsymbol{u}_{ij}^{(k)} + \omega\left(\boldsymbol{u}_{i-1,j}^{(k+1)} + \boldsymbol{u}_{i,j-1}^{(k+1)} + \boldsymbol{u}_{i+1,j}^{(k)} + \boldsymbol{u}_{i,j+1}^{(k)} + h^2 f\left(ih, jh\right)\right) \tag{7}$$

其中 $\omega \geq 1$ 是松弛因子。

2.2.4 具体实现

本题主要实现了如下函数：

- double mat_flatten_2_norm(vector<vector<double> > A);

  将矩阵 A 展平成向量，求出并返回展平后的向量 2 范数

- vector<vector<double> > pde2D_Jacobi(int n, double eps);

  使用 Jacobi 迭代法求解 PDE(3)的数值解；其中 n 是网格线数目，eps 是停机条件；返回求出的解矩阵

- double find_min(vector<vector<double> > A, int& ii, int& jj);

  找出并返回矩阵 A 中的最小元素，记录对应的下标 ii, jj

- vector<vector<double> > pde2D_GS(int n, double eps);

  使用 Gauss-Seidel 迭代法求解 PDE(3)的数值解；其中 n 是网格线数目，eps 是停机条件；返回求出的解矩阵

- vector<vector<double> > pde2D_SOR(int n, double eps, double omega);

  使用 SOR 迭代法求解 PDE(3)的数值解；其中 n 是网格线数目，eps 是停机条件，omega 是松弛因子；返回求出的解矩阵

2.3 实验结果

取 $N = 20$，分别使用 Jacobi 迭代法、Gauss-Seidel 迭代法和 SOR 迭代法，求解结果如图 4、5、6 所示。

图 4: $N = 20$，Jacobi 迭代法

图 5: $N = 20$，Gauss-Seidel 迭代法

为了简介起见，$N = 40, 80$ 的运行结果见附录 B。

将上述结果列表如表 5、6、7 所示。

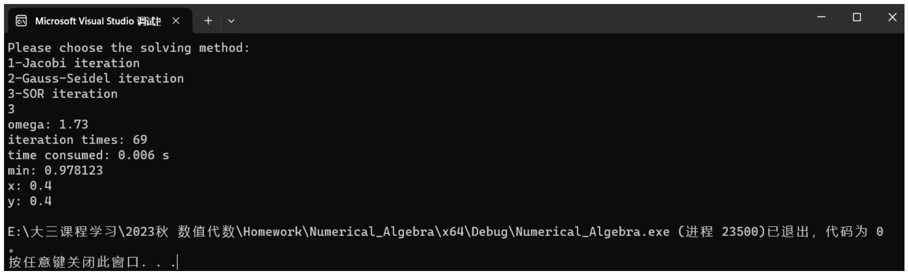


图 6: $N=20$，SOR 迭代法，$\omega=1.73$

迭代方法	迭代次数	运行时间 (s)	解的最小分量
Jacobi 迭代	1104	0.094	$u\,(0.4, 0.4) = 0.978122$
Gauss-Seidel 迭代	580	0.062	$u\,(0.4, 0.4) = 0.978123$
SOR 迭代 ($\omega = 1.73$)	69	0.006	$u\,(0.4, 0.4) = 0.978123$

表 5: $N=20$，三种迭代方法的比较

迭代方法	迭代次数	运行时间 (s)	解的最小分量
Jacobi 迭代	4216	1.155	$u\,(0.4, 0.4) = 0.978078$
Gauss-Seidel 迭代	2214	0.581	$u\,(0.4, 0.4) = 0.978078$
SOR 迭代 ($\omega = 1.85$)	141	0.048	$u\,(0.4, 0.4) = 0.978079$

表 6: $N=40$，三种迭代方法的比较

迭代方法	迭代次数	运行时间 (s)	解的最小分量
Jacobi 迭代	16028	13.161	$u\,(0.3875, 0.3875) = 0.978056$
Gauss-Seidel 迭代	8437	6.439	$u\,(0.3875, 0.3875) = 0.978058$
SOR 迭代 ($\omega = 1.92$)	309	0.253	$u\,(0.3875, 0.3875) = 0.978059$

表 7: $N=80$，三种迭代方法的比较

2.4 结果分析

观察上述结果，可以发现：

- 随着 N 的增大，三种迭代方法的迭代次数、运行时间均上升；
- 对于固定的 N，从迭代次数和运行时间来看，SOR 迭代法优于 G-S 迭代法优于 Jacobi 迭代法；
- 对于固定的 N，三种方法得到的解的最小分量一致，这也验证了算法的正确性。

A 附录：上机习题 1 中 $\epsilon = 0.1, 0.01, 0.0001$ 的运行结果

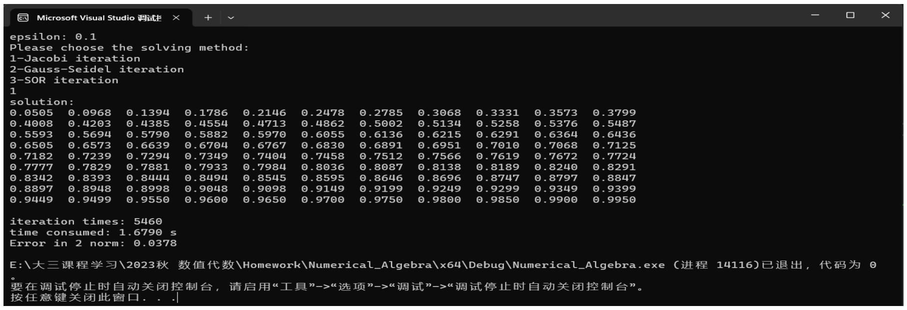


图 7: $\epsilon = 0.1$，Jacobi 迭代法

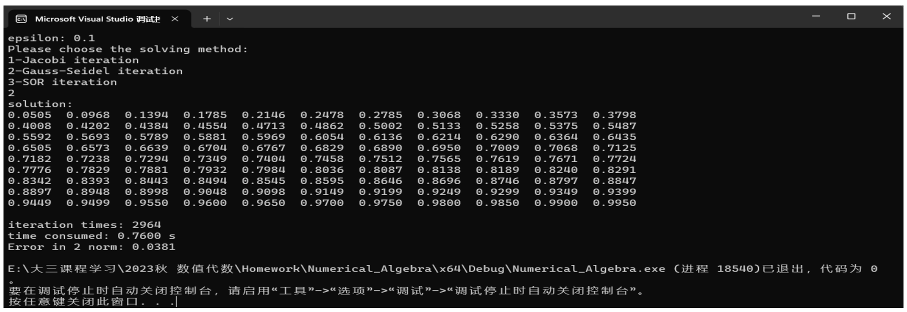


图 8: $\epsilon = 0.1$，Gauss-Seidel 迭代法

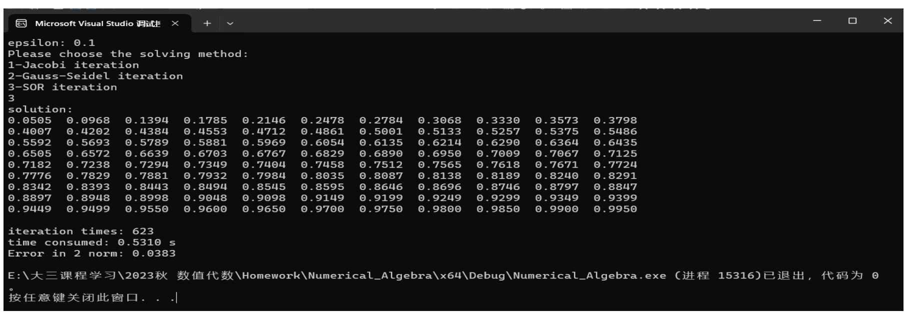


图 9: $\epsilon = 0.1$，SOR 迭代法，$\omega = 1.7$

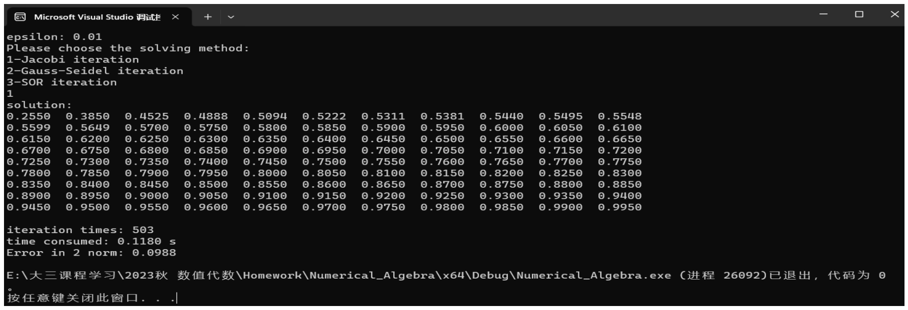


图 10: $\epsilon = 0.01$，Jacobi 迭代法

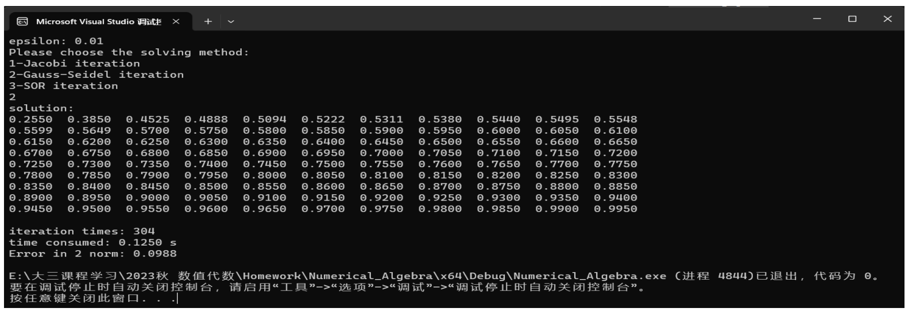


图 11: $\epsilon = 0.01$，Gauss-Seidel 迭代法

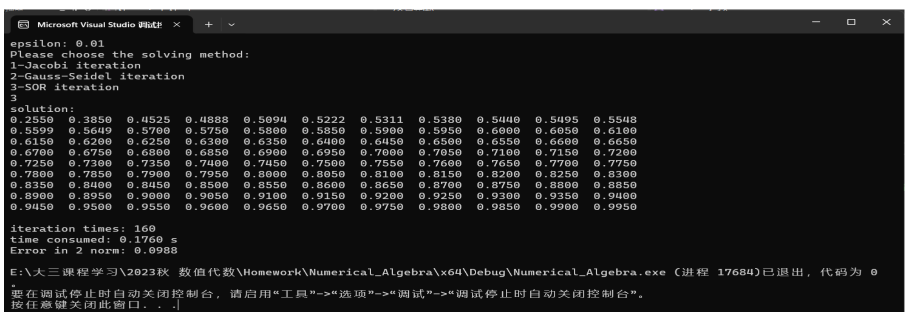


图 12: $\epsilon = 0.01$，SOR 迭代法，$\omega = 1.7$

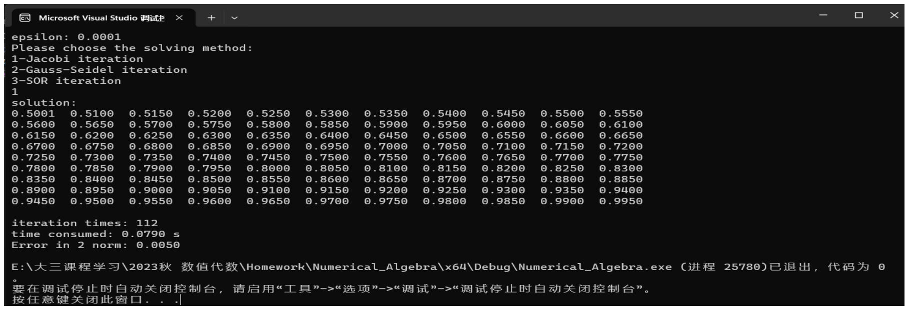


图 13: $\epsilon = 0.0001$，Jacobi 迭代法

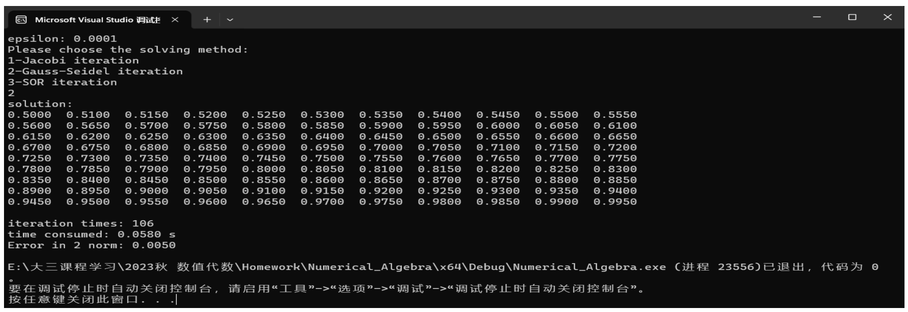


图 14: $\epsilon = 0.0001$，Gauss-Seidel 迭代法

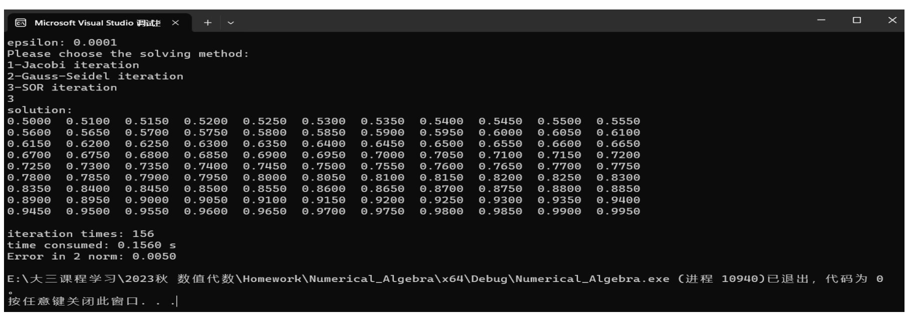


图 15: $\epsilon = 0.0001$，SOR 迭代法，$\omega = 1.14$

B 附录：上机习题 2 中 $N = 40, 80$ 的运行结果

```
Please choose the solving method:
1-Jacobi iteration
2-Gauss-Seidel iteration
3-SOR iteration
1
iteration times: 4216
time consumed: 1.155 s
min: 0.978078
x: 0.4
y: 0.4

E:\大三课程学习\2023秋 数值代数\Homework\Numerical_Algebra\x64\Debug\Numerical_Algebra.exe (进程 7692)已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口. . .
```

图 16: $N = 40$，Jacobi 迭代法

```
Please choose the solving method:
1-Jacobi iteration
2-Gauss-Seidel iteration
3-SOR iteration
2
iteration times: 2214
time consumed: 0.581 s
min: 0.978078
x: 0.4
y: 0.4

E:\大三课程学习\2023秋 数值代数\Homework\Numerical_Algebra\x64\Debug\Numerical_Algebra.exe (进程 22840)已退出，代码为 0
。
按任意键关闭此窗口. . .
```

图 17: $N = 40$，Gauss-Seidel 迭代法

```
Please choose the solving method:
1-Jacobi iteration
2-Gauss-Seidel iteration
3-SOR iteration
3
omega: 1.85
iteration times: 141
time consumed: 0.048 s
min: 0.978079
x: 0.4
y: 0.4

E:\大三课程学习\2023秋 数值代数\Homework\Numerical_Algebra\x64\Debug\Numerical_Algebra.exe (进程 27476)已退出，代码为 0
。
按任意键关闭此窗口. . .
```

图 18: $N = 40$，SOR 迭代法，$\omega = 1.85$

图 19: $N = 80$，Jacobi 迭代法

图 20: $N = 80$，Gauss-Seidel 迭代法

图 21: $N = 80$，SOR 迭代法，$\omega = 1.92$