Report on Numerical Linear Algebra Coding Assignment

Chapter 4: Classical Iterations for Linear Systems

Hongyi Zhang, 1500017736

November 19, 2017

1 Problem setting

Consider the boundary value problem

$$\begin{cases} \epsilon \dfrac{\mathrm{d}^2 y}{\mathrm{d}x^2} + \dfrac{\mathrm{d}y}{\mathrm{d}x} = a,\ 0 < a < 1, \\ y(0) = 0,\ y(1) = 1. \end{cases} \tag{*}$$

In order to obtain a numerical solution, divide the interval $[0, 1]$ evenly into n parts. Let $x_i = i/n$ $(1 \le i \le n-1)$, then the problem can be discretized into a difference equation

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

which simplifies to

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

The equation can be written in a matrix form $Ay = b$, which is if written more verbosely

$$\begin{bmatrix} -(2\epsilon+h) & \epsilon+h & & & & \\ \epsilon & -(2\epsilon+h) & \epsilon+h & & & \\ & \ddots & \ddots & \ddots & & \\ & & \epsilon & -(2\epsilon+h) & \epsilon+h & \\ & & & \epsilon & -(2\epsilon+h) & \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-2} \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} ah^2 \\ ah^2 \\ \vdots \\ ah^2 \\ ah^2 - (\epsilon + h) \end{bmatrix}.$$

Set $a = 1/2,\ n = 100$, and try to solve this equation by Jacobi iteration, Gauss-Seidel iteration as well as SOR iteration to obtain a numerical solution with up to 4 significant digits for $\epsilon = 1, 0.1, 0.01, 0.0001$. Compare the error with the accurate solution.

2 Classical iterations for linear systems

To solve the system $Ax = b$ by iteration, we decompose $A = L + D + U$, and formulate the three classical iterations in the following consistent form:

$$Nx_{k+1} = Mx_k + \omega b, \tag{1}$$

where for the Jacobi iteration,

$$N = D, \quad M = -(L + U), \quad \omega = 1,$$

and for Gauss-Seidel ($\omega = 1$) and SOR iterations ($0 < \omega < 2$),

$$N = D + \omega L, \quad M = (1 - \omega)D - \omega U.$$

The optimal relaxation factor for SOR iteration is given by

$$\omega^* = \frac{2}{1 + \sqrt{1 - \rho(B)^2}}, \tag{2}$$

where $B = -D^{-1}(L + U)$ is the Jacobi iteration matrix. This particular factor minimizes the spectral radius of the SOR iteration matrix and yields the fastest possible convergence. Though the computation for this problem can be optimized since A is tridiagonal, we do not take this advantage here and treat it in a more general manner somehow.

In order to achieve four significant digits in the solution, we need to specify an appropriate stopping criterion. The option adopted here is the posterior criterion

$$\|x_k - x^*\| \leq \frac{q}{1 - q}\|x_k - x_{k-1}\| \leq \frac{1}{2} \times 10^{-4},$$

where q is the norm of the iteration matrix. Note that $q < 1$ is a must for the criterion to work, which can be violated when the spectral radius of the iteration matrix is fairly close to 1. If that does happen, we choose to use $\|x_k - x_{k-1}\| < 10^{-8}$ for error approximation.

3 Numerical results

The analytic solution for the problem is

$$y = \frac{1 - a}{1 - e^{-1/\epsilon}}(1 - e^{-x/\epsilon}) + ax, \tag{3}$$

which is monotonically increasing in the interval $[0, 1]$.

Table 1: Numerical results for problem (*) ($N = 100$)

(a) $\epsilon = 1$

	$\rho(B)$	N_{iter}	$\|y - y^*\|_2$
Jacobi	0.9995	23663	0.0022
G-S	0.9990	11504	0.0022
SOR	0.9384	351	0.0022

(b) $\epsilon = 0.1$

	$\rho(B)$	N_{iter}	$\|y - y^*\|_2$
Jacobi	0.9984	9041	0.0384
G-S	0.9967	4475	0.0384
SOR	0.8921	239	0.0384

(c) $\epsilon = 0.01$

	$\rho(B)$	N_{iter}	$\|y - y^*\|_2$
Jacobi	0.9423	657	0.0988
G-S	0.8880	381	0.0988
SOR	0.5483	119	0.0988

(d) $\epsilon = 0.0001$

	$\rho(B)$	N_{iter}	$\|y - y^*\|_2$
Jacobi	0.1951	124	0.0050
G-S	0.0392	113	0.0050
SOR	0.0520	106	0.0050

For the iteration method, the initial value is $x_0 = 0$, and the max number of iterations is set to be 10^5. Numerical results for the differential equation are shown in Table 1, whose notations are explained as follows: $\rho(B)$ is the spectral radius of the iteration matrix, N_{iter} is the number of iterations needed for convergence, $\|y - y^*\|_2$ is the L_2 error of the output from the analytic solution.

As can be seen from Table 1, when ϵ is fixed, Gauss-Seidel enjoys a smaller spectral radius than Jacobi iteration, and thus leading to a convergence roughly twice as fast, as is indicated by the number of iterations. SOR iteration here is performed using the optimal relaxation factor defined by formula (2), and it brings amazing improvement and incredible acceleration on the iteration convergence.

The case when $\epsilon = 0.0001$ seems a bit odd — both G-S and SOR do not improve as much as they do for the other three cases. The reason is that, when ϵ gets too small, the matrix L is not far away from 0, and hence the iteration matrix for Jacobi and G-S can both be approximated by $B \approx -D^{-1}U$, leading to a similar rate of convergence for these two methods. Furthermore, since now B has eigenvalues all around zero, the optimal ω^* is almost 1 as well. Thus, SOR shows no significant faster convergence than G-S either in this situation. As for the larger spectral radius of SOR than G-S, that should be an issue of numerical error in evaluating eigenvalues.

Three methods all give the same L_2 error with up to 4 digits, which means the numerical solution to the linear equation $Ay = b$ is acceptable, and this error basically arises from

Table 2: Numerical results for problem (*) ($N = 200$)

(a) $\epsilon = 1$

	$\rho(B)$	N_{iter}	$\|y - y^*\|_2$
Jacobi	0.9999	97396	0.0016
G-S	0.9997	47272	0.0016
SOR	0.9687	691	0.0015

(b) $\epsilon = 0.1$

	$\rho(B)$	N_{iter}	$\|y - y^*\|_2$
Jacobi	0.9996	36007	0.0275
G-S	0.9992	17664	0.0275
SOR	0.9436	466	0.0275

(c) $\epsilon = 0.01$

	$\rho(B)$	N_{iter}	$\|y - y^*\|_2$
Jacobi	0.9797	2124	0.0779
G-S	0.9598	1164	0.0779
SOR	0.7150	230	0.0779

(d) $\epsilon = 0.0001$

	$\rho(B)$	N_{iter}	$\|y - y^*\|_2$
Jacobi	0.3381	244	0.0098
G-S	0.0948	223	0.0098
SOR	0.0982	215	0.0098

the discretization process of the differential equation. Intuitively, decrease on stride h can help increase precision, and Table 2 shows the results of solving the same equation on a $N = 200$ scale.

Similar as before, G-S converges twice as fast as Jacobi, and SOR exhibits even more powerful reduction on the number of iteration steps. Compared with Table 1, the L_2 error indeed shrinks towards zero a bit, but this advance in accuracy calls for more computational cost due to a larger scale of equation to solve, which is intolerable if we keep increasing the number N of subintervals. The $\epsilon = 0.0001$ case again seems weird, but the analysis on this requires further discussion and we will not delve into it for this part.