MA/CSC 580, Homework #4
Due October 31, 2018
Lindsay Eddy

1. Let A be a symmetric positive definite matrix, so it has the Cholesky decomposition $A = LL^T$. Show that

   (a): $0 \le l_{kk} \le \sqrt{a_{kk}}$, $k = 1, 2, \cdots, n$.

   By definition, $l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2}$. Since $\sum_{j=1}^{k-1} l_{kj}^2$ is the sum of squares it is greater than or equal to zero. Thus $l_{kk} \le \sqrt{a_{kk}}$, $k = 1, 2, \ldots, n$. Also, $0 \le \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2} = l_{kk}$.

   (b): From (a) to derive $\max_{1 \le j \le i \le n} |l_{ij}| \le \sqrt{\max_{1 \le i,j \le n} |a_{ij}|}$. That is, the Cholesky decomposition is a stable algorithm.

   By definition, $l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2}$. We can rearrange this to get $\sum_{j=1}^{k-1} l_{kj}^2 = a_{kk} - l_{kk}^2$. Note that this is the same as $\sum_{j=1}^{i-1} l_{ij}^2 = a_{ii} - l_{ii}^2$.

   Choose any i, and let $j \le i$. Then

$$
\begin{aligned}
0 \le l_{ij}^2 \le \sum_{j=1}^{i-1} l_{ij}^2 = a_{ii} - l_{ii}^2 &\le |a_{ii}| \quad \text{(by part a)} \\
&\le \max |a_{ij}|.
\end{aligned}
$$

   Since for all $1 \le j \le i \le n$, $l_{ij}^2 \le \max |a_{ij}|$, then

$$\max_{1 \le i \le j \le n} l_{ij}^2 \le \max_{1 \le i \le j \le n} |a_{ij}|.$$

   That is,

$$\max_{1 \le i \le j \le n} |l_{ij}| \le \sqrt{\max_{1 \le i \le j \le n} |a_{ij}|}.$$

   (c): Do the Jacobi, Gauss-Seidel, and SOR(ω) iterative methods converge?

   The SOR(ω) method converges for $0 < \omega < 2$, since A is SPD.

   The Jacobi and Gauss-Seidel methods may or may not converge. It depends on the particular SPD matrix.

2. Consider the Poisson equation

$$\begin{aligned} u_{xx} + u_{yy} &= xy, \quad (x,y) \in \Omega \\ u(x,y)|_{\partial\Omega} &= 0, \end{aligned}$$

where Ω is the unit square. Using the finite difference method, we can get a linear system of equations

$$\frac{U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{ij}}{h^2} = f(x_i, y_j), \quad 1 \le i,j \le 3, \tag{1}$$

where $h = 1/4$, $x_i = ih$, $y_j = jh$, $i,j = 0,1,2,3,4$, and U_{ij} is an approximation of $u(x_i, y_j)$. Write down the coefficient matrix and the right hand side using the *red-black* orderings given in the right diagram below. What is the dimension of the coefficient matrix? How many nonzero entries and how many zeros? Generalize your results to general case when $0 \le i,j \le n$ and $h = 1/n$. Write down the component form of the $SOR(\omega)$ iterative method. Does the $SOR(\omega)$ iterative method depend on the ordering? From your analysis, explain whether you prefer to use Gaussian elimination method or an iterative method.

Using red-black orderings, the coefficient matrix is

$$A = \frac{1}{h^2}\begin{bmatrix} -4 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & -4 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & -4 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & -4 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -4 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & -4 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & -4 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & -4 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & -4 \end{bmatrix}$$

$$= \begin{bmatrix} -64 & 0 & 0 & 0 & 0 & 16 & 16 & 0 & 0 \\ 0 & -64 & 0 & 0 & 0 & 16 & 0 & 16 & 0 \\ 0 & 0 & -64 & 0 & 0 & 16 & 16 & 16 & 16 \\ 0 & 0 & 0 & -64 & 0 & 0 & 16 & 0 & 16 \\ 0 & 0 & 0 & 0 & -64 & 0 & 0 & 16 & 16 \\ 16 & 16 & 0 & 0 & 0 & -64 & 0 & 0 & 16 \\ 16 & 0 & 16 & 16 & 0 & 0 & -64 & 0 & 0 \\ 0 & 16 & 16 & 0 & 16 & 0 & 0 & -64 & 0 \\ 0 & 0 & 16 & 16 & 16 & 0 & 0 & 0 & -64 \end{bmatrix}$$

since $h = 1/4$. The right hand side vector is

$$F = \begin{bmatrix} f_{11} - \frac{u_{01}+u_{10}}{h^2} \\ f_{31} - \frac{u_{41}+u_{30}}{h^2} \\ f_{22} \\ f_{13} - \frac{u_{03}+u_{14}}{h^2} \\ f_{33} - \frac{u_{43}+u_{34}}{h^2} \\ f_{21} - \frac{u_{20}}{h^2} \\ f_{12} - \frac{u_{02}}{h^2} \\ f_{32} - \frac{u_{42}}{h^2} \\ f_{23} - \frac{u_{24}}{h^2} \end{bmatrix} = \begin{bmatrix} f_{11} \\ f_{31} \\ f_{22} \\ f_{13} \\ f_{33} \\ f_{21} \\ f_{12} \\ f_{32} \\ f_{23} \end{bmatrix}$$

since $u(x,y)\big|_{\partial\Omega} = 0$. Here, $f_{ij} = f(x_i, y_j)$.

The coefficient matrix is 9×9. There are 33 non-zero entries and 48 zeros.

In the general case where $0 \le i, j \le n$ and $h = 1/n$, the component form of the $SOR(\omega)$ iterative method is:

$$U_{ij}^{(k+1)} = U_{ij}^{(k)}$$

$$\begin{aligned} U_{ij}^{(k+1)} &= (1-\omega)U_{ij}^{(k+1)} - \frac{\omega h^2}{4} f(x_i, y_j) + \frac{\omega}{4}\left(U_{i-1,j}^{(k+1)} + U_{i+1,j}^{(k+1)} + U_{i,j-1}^{(k+1)} + U_{i,j+1}^{(k+1)}\right) \\ &= (1-\omega)U_{ij}^{(k+1)} - \frac{\omega}{4n^2} f(x_i, y_j) + \frac{\omega}{4}\left(U_{i-1,j}^{(k+1)} + U_{i+1,j}^{(k+1)} + U_{i,j-1}^{(k+1)} + U_{i,j+1}^{(k+1)}\right) \end{aligned}$$

for $1 \le i, j \le n-1$.

The method does depend on the ordering because the ordering determines whether an unknown value is updated by the time in is used to compute $U_{ij}^{(k+1)}$. For example, depending on the ordering, in the $(k+1)$st iteration, $U_{i-1,j}^{(k+1)}$ may or may not be updated by the time it is used in computing the updated $U_{ij}^{(k+1)}$. However, as long as the same ordering is used for equations and unknowns, the above component form of the $SOR(\omega)$ iterative method remains the same.

For large n, the coefficient matrix will have many more zeros than non-zero entries. Each row will have a maximum of 5 non-zero entries, and it will have $(n-1)^2$ total entries. Thus, A is a sparse matrix and it is better to use an iterative method than Gaussian elimination in order to avoid "fill-ins".

3. Given the following linear system of equations:

$$\begin{aligned} 3x_1 - x_2 + x_3 &= 3 \\ 2x_2 + x_3 &= 2 \\ -x_2 + 2x_3 &= 2 \end{aligned}$$

(a) With $x^{(0)} = [1, -1, 1]^T$, find the *first* iteration of the Jacobi, Gauss-Seidel, and SOR ($\omega = 1.5$) methods.

Jacobi:

$$\begin{aligned} x^{(1)} &= D^{-1}(L+U)x^{(0)} + D^{-1}b \\ &= \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/2 \\ 1/2 \end{bmatrix} \end{aligned}$$

Gauss-Seidel:

$$\begin{aligned} x^{(1)} &= (D-L)^{-1}Ux^{(0)} + (D-L)^{-1}b \\ &= \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 1/4 & 1/2 \end{bmatrix} \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 1/4 & 1/2 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/2 \\ 5/4 \end{bmatrix} \end{aligned}$$

SOR ($\omega = 1.5$):

$$x_i^{(1)} = (-0.5)x_i^{(0)} + 1.5\left(\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(1)} - \sum_{j=i+1}^{3} a_{ij}x_j^{(0)}\right)/a_{ii}\right)$$

Therefore

$$\begin{aligned}
x_1^{(1)} &= (-0.5)1 + 1.5\Big(\big(3-(-1(-1)+1(1))\big)/3\Big) = -\frac{1}{2}+\frac{3}{2}\left(\frac{1}{3}\right) = 0\\
x_2^{(1)} &= (-0.5)(-1) + 1.5\Big(\big(2-0-1(1)\big)/2\Big) = \frac{1}{2}+\frac{3}{2}\left(\frac{1}{2}\right) = \frac{5}{4}\\
x_3^{(1)} &= (-0.5)1 + 1.5\Big(\big(2-0-(-1)(5/4)\big)/2\Big) = \frac{31}{16}
\end{aligned}$$

(b) Write down the Jacobi, Gauss-Seidel, and Seidel iteration matrices R_J , R_{GS}, and SOR(ω).

$$R_J = D^{-1}(L+U) = \begin{bmatrix} 1/3 & 0 & 0\\ 0 & 1/2 & 0\\ 0 & 0 & 1/2 \end{bmatrix}\begin{bmatrix} 0 & 1 & -1\\ 0 & 0 & -1\\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1/3 & -1/3\\ 0 & 0 & -1/2\\ 0 & 1/2 & 0 \end{bmatrix}$$

$$R_{GS} = (D-L)^{-1}U = \begin{bmatrix} 3 & 0 & 0\\ 0 & 2 & 0\\ 0 & -1 & 2 \end{bmatrix}^{-1}\begin{bmatrix} 0 & 1 & -1\\ 0 & 0 & -1\\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1/3 & -1/3\\ 0 & 0 & -1/2\\ 0 & 0 & -1/4 \end{bmatrix}$$

$$\begin{aligned}
R_{SOR(\omega)} &= (D-\omega L)^{-1}\big((1-\omega)D+\omega U\big) = \begin{bmatrix} 3 & 0 & 0\\ 0 & 2 & 0\\ 0 & -\omega & 2 \end{bmatrix}^{-1}\begin{bmatrix} 3(1-\omega) & \omega & -\omega\\ 0 & 2(1-\omega) & -\omega\\ 0 & 0 & 2(1-\omega) \end{bmatrix}\\
&= \begin{bmatrix} 1/3 & 0 & 0\\ 0 & 1/2 & 0\\ 0 & \omega/4 & 1/2 \end{bmatrix}\begin{bmatrix} 3(1-\omega) & \omega & -\omega\\ 0 & 2(1-\omega) & -\omega\\ 0 & 0 & 2(1-\omega) \end{bmatrix} = \begin{bmatrix} 1-\omega & \omega/3 & -\omega/3\\ 0 & 1-\omega & -\omega/2\\ 0 & \omega(1-\omega)/2 & -\omega^2/4+1-\omega \end{bmatrix}
\end{aligned}$$

(c) Do the Jacobi and Gauss-Seidel iterative methods converge? Why?
The Jacobi and Gauss-Seidel methods converge because

$$A = \begin{bmatrix} 3 & -1 & 1\\ 0 & 2 & 1\\ 0 & -1 & 2 \end{bmatrix}$$

is strictly row diagonal dominant.

4. Explain when we want to use *iterative* methods to solve linear system of equations $Ax = b$ instead of *direct* methods.

Also if $||R|| = 1/10$, then the iterative method $x^{(k+1)} = R\,x^{(k)} + c$ converges to the solution x^*, $x^* = Rx^* + c$. How many iterations are required so that $||x^{(k)} - x^*|| \le 10^{-6}$? Suppose $||x^{(0)} - x^*|| = O(1)$.

We want to use iterative methods (instead of direct methods) when matricies are large (and we cannot store all the entries of the matrix) or when matrices are spares, and using direct methods such as Gaussian elimination will destroy the structure of the matrix, causing “fill-in”s – that is, turning a sparse matrix with many zeros into a dense matrix with few zeros.

We require a k such that

$$\|x^{(k)} - x^*\| = \|e^{(k)}\| \leq \frac{\|R\|^k}{1 + \|R\|}\|x^{(1)} - x^{(0)}\| \leq 10^{-6}.$$

Since $\|R\| = 1/10$, this becomes

$$\frac{(1/10)^k}{11/10}\|x^{(1)} - x^{(0)}\| \leq 10^{-6}.$$

We can rewrite this to get

$$\|x^{(1)} - x^{(0)}\|10^{-k} \leq \left(\frac{10}{11}\right)10^{-6}.$$

Thus, the number of iterations needed is

$$k = 6 + \log_{10}\left(\|x^{(1)} - x^{(0)}\|\right).$$

Indeed, we see that

$$\|x^{(1)} - x^{(0)}\|10^{\left(-6-\log_{10}(\|x^{(1)}-x^{(0)}\|)\right)} = \|x^{(1)} - x^{(0)}\|\left(10^{-6}\right)\|x^{(1)} - x^{(0)}\|^{-1} = 10^{-6} \leq \left(\frac{10}{11}\right)10^{-6}.$$

5. Judge whether the iterative method $x^{(k+1)} = R\,x^{(k)} + c$ converges or not.

$$(a): \quad R = \begin{bmatrix} e^{-1} & -e^{1} & -1 & -1 & -10 \\ 0 & \sin\pi/4 & 10^4 & -1 & -1 \\ 0 & 0 & -0.1 & -1 & 1 \\ 0 & 0 & 0 & 1-e^{-2} & -1 \\ 0 & 0 & 0 & 0 & 1-\sin(\alpha\pi) \end{bmatrix}, \qquad (b): \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 0.3 & -0.7 \\ 0 & 0.69 & 0.2999 \end{bmatrix}$$

(a) Since R is an upper triangular matrix, its eigenvalues are its diagonal entries. Thus, $\rho(R)$ is equal to the maximum of R's diagonal entries. Since the absolute values of the first four diagonal entries are less than one, the spectral radius of R is less than one when $\left|1 - \sin(\alpha\pi)\right|$ is less than one. This is the case when:

$$-1 < 1 - \sin(\alpha\pi) < 1$$

$$-2 < -\sin(\alpha\pi) < 0$$

$$0 < \sin(\alpha\pi) < 2.$$

This holds when

$$2k < \alpha < 2k + 1, \quad k \in \mathbb{Z}.$$

Thus, the iterative method converges for the above values of α, and does not converge for all other values of α.

(b) $\|R\|_1 = 0.9999 < 1$, so the iterative method converges.

6. Determine the convergence of the Jacobi and Gauss-Seidel method applied to the system of equations $Ax = b$, where

$$(a): \quad A = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & -2 & 1 \end{bmatrix}, \qquad (b): \quad A = \begin{bmatrix} 3 & -1 & 0 & 0 & \cdots & \cdots & 0 \\ 2 & 3 & -1 & 0 & \cdots & \cdots & 0 \\ 0 & 2 & 3 & -1 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & 0 & 2 & 3 & -1 \\ 0 & \cdots & \cdots & \cdots & 0 & 2 & 3 \end{bmatrix}$$

(a) <u>Jacobi:</u>

$$R_J = D^{-1}(L+U) = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 2 & 0 \end{bmatrix}$$

To find the eigenvalues of R_J:

$$\det(R_J - \lambda I) = \begin{vmatrix} -\lambda & 0 & 0 \\ 0 & -\lambda & -2 \\ 0 & 2 & -\lambda \end{vmatrix} = -\lambda \begin{vmatrix} -\lambda & -2 \\ 2 & -\lambda \end{vmatrix} = -\lambda(\lambda^2 + 4) = 0$$

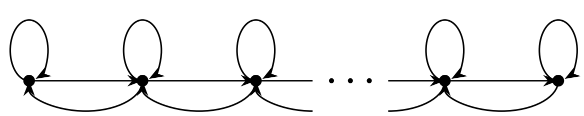
Thus $\lambda_1 = -2i$, $\lambda_2 = 2i$, and $\lambda_3 = 0$. Then $|\lambda_1| = \sqrt{0^2 + (-2)^2} = 2$, $|\lambda_2| = \sqrt{0^2 + 2^2} = 2$, and $|\lambda_3| = 0$. Therefore, $\rho(R_J) = 2 > 1$, and the Jacobi method does not converge.

<u>Gauss-Seidel:</u>

$$R_{GS} = (D-L)^{-1}U = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & -4 \end{bmatrix}$$

Note that R_{GS} is and upper-triangular matrix, so $\lambda_1 = -4$, $\lambda_2 = 0$, and $\lambda_3 = 0$. Thus $\rho(R_{GS}) = 4$, so the Gauss-Seidel method does not converge.

(b) Both the Gauss-Seidel and Jacobi methods converge. To prove this, first note that A is weakly row diagonal dominant. Also, in the first and last rows, the magnitude of the diagonal element is strictly greater than the sum of the magnitudes of the off-diagonal elements.
We will now show that A is irreducible. Note that if the digraph of A is strongly connected, then A is irreducible. The digraph of A looks like:



This graph is strongly connected, thus A is irreducible.
Thus, the Jacobi and Gauss-Seidel methods converge for A.

7. Modify the Matlab code *poisson_drive.m* and *poisson_sor.m* to solve the following diffusion and convection equation:

$$u_{xx} + u_{yy} + au_x - bu_y = f(x,y), \quad 0 \le x, y \le 1,$$

Assume that solution at the boundary $x = 0$, $x = 1$, $y = 0$, $y = 1$ are given (Dirichlet boundary conditions). The central-upwinding finite difference scheme is

$$\frac{U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{ij}}{h^2} + a\frac{U_{i+1,j} - U_{ij}}{h} - b\frac{U_{i,j} - U_{i,j-1}}{h} = f_{ij}$$

(a) Assume the exact solution is $u(x,y) = e^{2y}\sin(\pi x)$, find $f(x,y)$.

(b) Use the $u(x,y)$ above for the boundary condition and the $f(x,y)$ above for the partial differential equation. Let $a = 1$, $b = 2$, and $a = 100$, $b = 2$, solve the problem with $n = 20$, 40, 80, and $n = 160$. Try $\omega = 1$, the best ω for the Poisson equation discussed in the class, the optimal ω by testing, for example $\omega = 1.9, 1.8, \cdots, 1$.

(c) Tabulate the error, the number of iterations for $n = 20$, 40, 80, and $n = 160$ with your tested optimal ω, compare the number of iterations with the Gauss-Seidel method.

(d) Plot the solution and the error for $n = 40$ with your tested optimal ω. Label your plots as well.

(a)

$$
\begin{aligned}
u_x &= e^{2y}\pi\cos(\pi x)\\
u_{xx} &= -e^{2y}\pi^2\sin(\pi x)\\
u_y &= 2e^{2y}\sin(\pi x)\\
u_{yy} &= 4e^{2y}\sin(\pi x)
\end{aligned}
$$

Thus

$$
\begin{aligned}
f(x,y) &= -e^{2y}\pi^2\sin(\pi x) + 4e^{2y}\sin(\pi x) + ae^{2y}\pi\cos(\pi x) - 2be^{2y}\sin(\pi x)\\
&= e^{2y}\left[4\sin(\pi x) - \pi^2\sin(\pi x) = a\pi\cos(\pi x) - 2b\sin(\pi x)\right]
\end{aligned}
$$

(b) See Matlab code, files `Q7.m`, `getData.m`, `optimalw.m`, and `poisson_mod_sor.m`.
The best ω for the Poisson equation discussed in class was calculated using the formula

$$\omega_{\text{opt}} = \frac{2}{1 + \frac{\pi}{n+1}}.$$

The optimal ω by testing was defined as the ω in $\omega = 1.9, 1.8, \ldots, 1$ which required the smallest number of iterations.

(c) The data is reported in Figures 1 and 2.

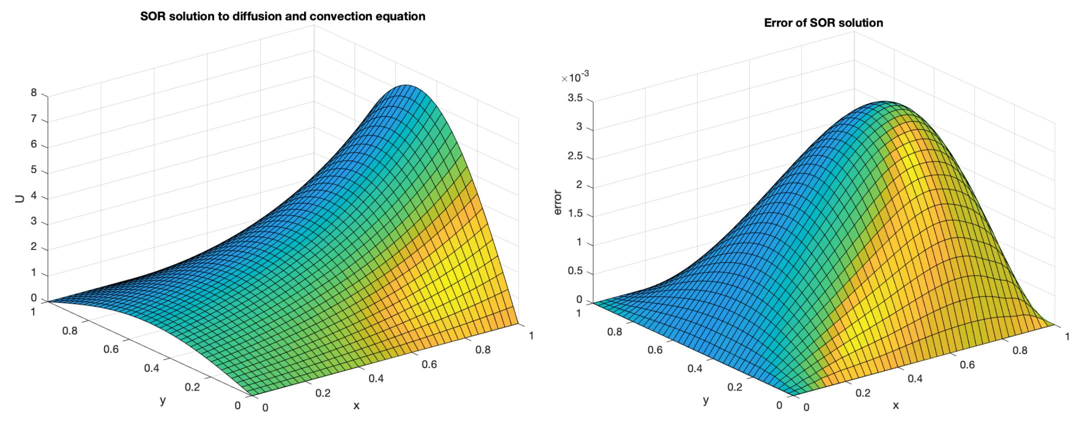
n	20	40	80	160
ω_{opt}	1.7	1.8	1.9	1.9
error	5.6964×10^{-3}	2.3247×10^{-2}	3.0018×10^{-2}	1.6988×10^{-1}
# iterations with opt ω	28	52	92	137
# iterations with Gauss-Seidel	71	123	155	156

Figure 1: Data where $a = 1$ and $b = 2$.

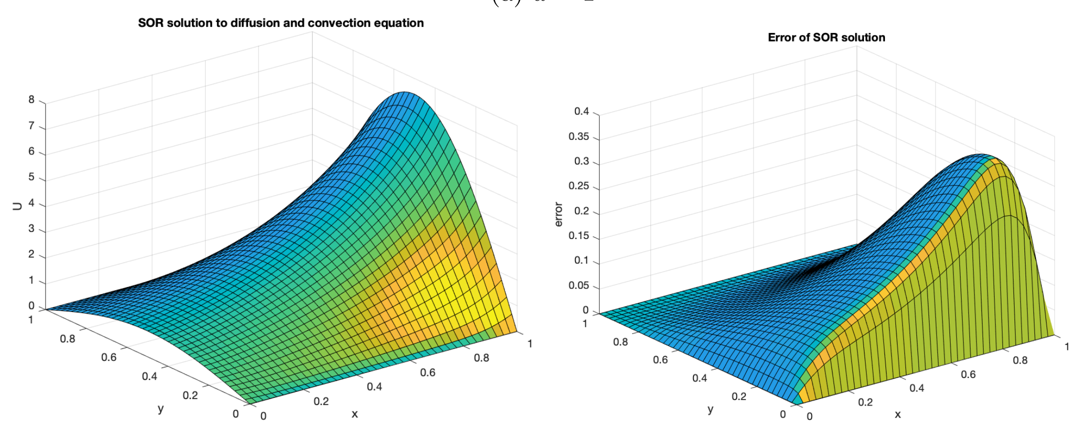
n	20	40	80	160
ω_{opt}	1.2	1.3	1.5	1.7
error	1.0112×10^{-1}	5.2739×10^{-2}	2.6550×10^{-2}	1.3194×10^{-2}
# iterations with opt ω	24	52	101	191
# iterations with Gauss-Seidel	33	77	205	568

Figure 2: Data where $a = 100$ and $b = 2$.

(d) The plots are reported in Figure . The pointwise error was defined as $|u(x_i, y_j) - U_{ij}|$.



(a) $a = 1$



(b) $a = 100$

Figure 3: Solution grid, U, obtained using the SOR method with tested optimal ω, as well as the error, for $a = 1$ and $a = 100$.