Scientific Computation HW4

Michael Nameika

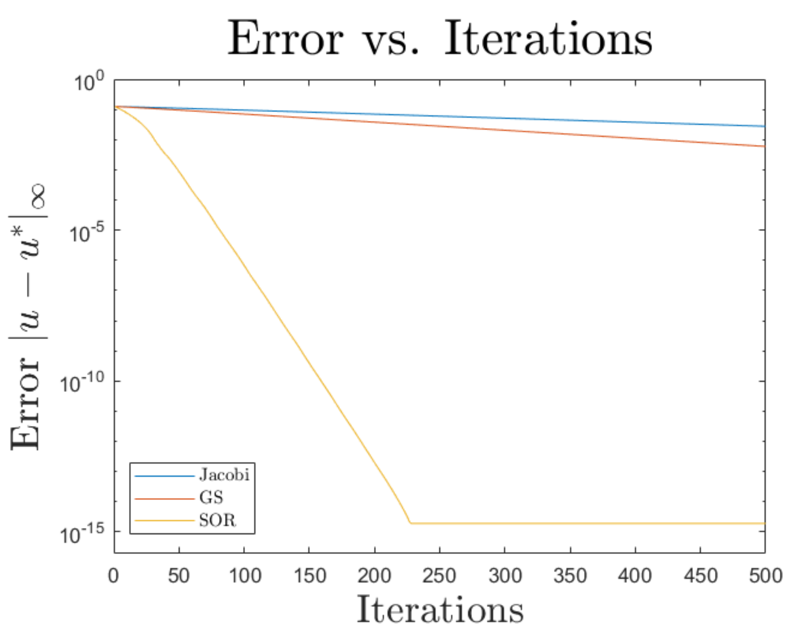
October 2022

Exercise 4.1 *(Convergence of SOR)*

The m-file `iter_bvp_Asplit.m` implements the Jacobi, Gauss-Seidel, and SOR matrix splitting methods on the linear system arising from the boundary value problem $u''(x) = f(x)$ in one space dimension.

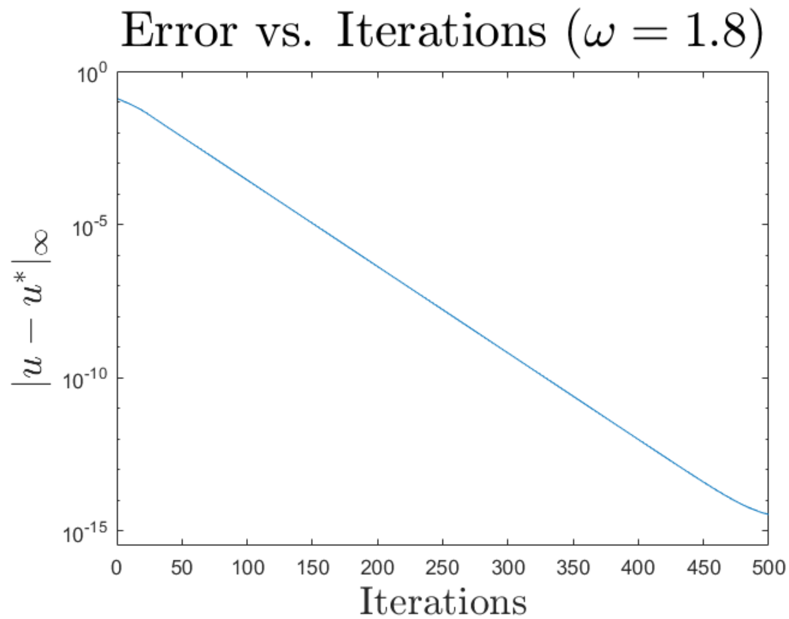
(a) Run this program for each method and produce a plot similar to Figure 4.2.

Running the iter_bvp_Asplit.m for the Jacobi, Gauss-Seidel, and Successive Overrelaxation methods, we find the following plot for each method's errors over the iterations:

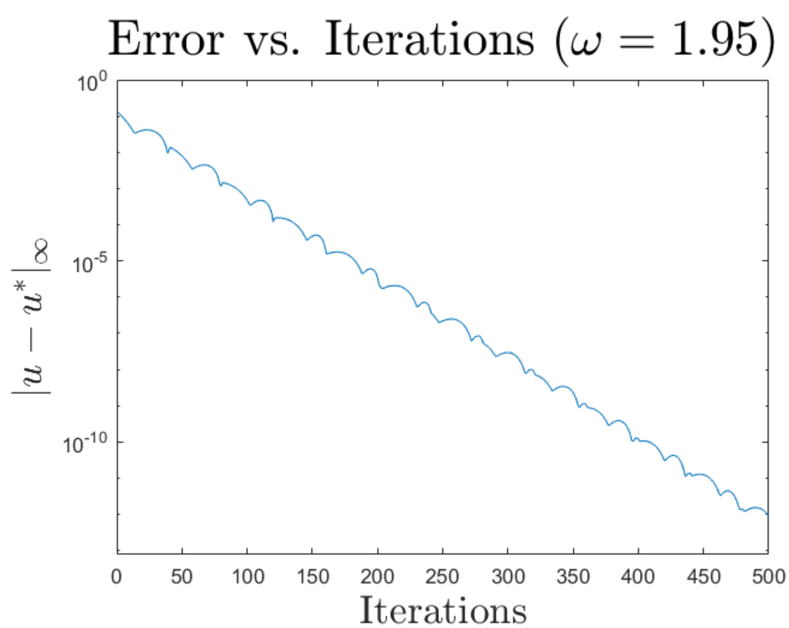


(b) The convergence behavior of SOR is very sensitive to the choice of ω (`omega` in the code). Try changing from the optimal ω to $\omega = 1.8$ or 1.95.

Changing the value of ω to 1.85 in the SOR method, we find the following plot of the error versus number of iterations:

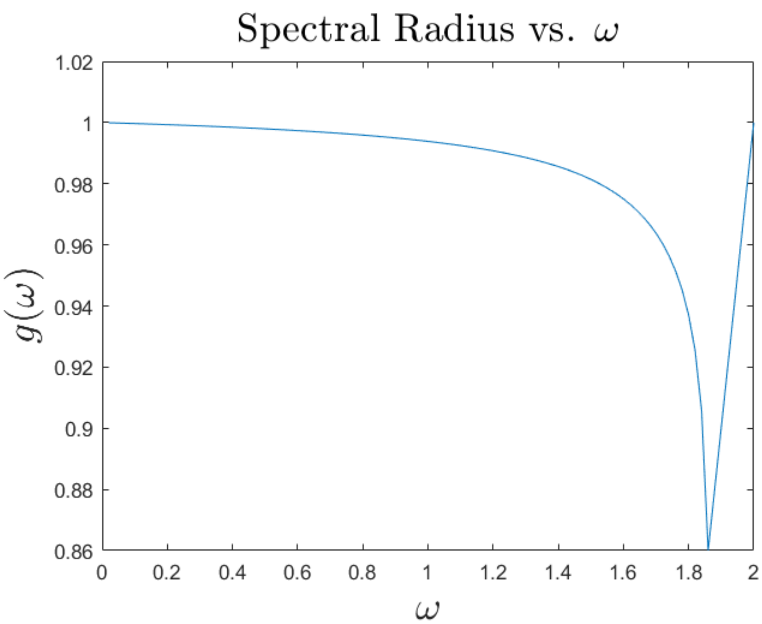


And similarly for $\omega = 1.95$:



(c) Let $g(\omega) = \rho(G(\omega))$ be the spectral radius of the iteration matrix G for a given value of ω. Write a program to produce a plot of $g(\omega)$ for $0 \leq \omega \leq 2$.

Using the modified code to run for values of ω between 0 and 2 using 101 mesh points, we find the following plot:

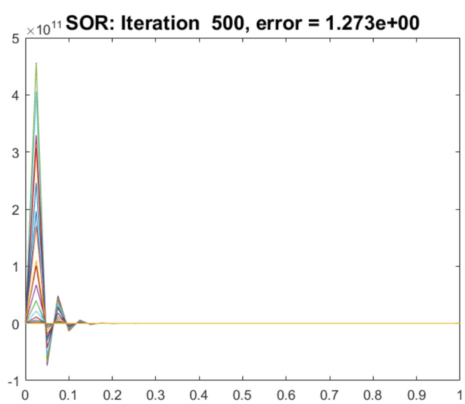


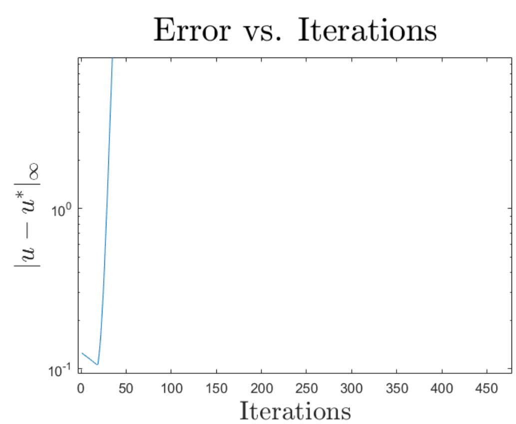
(d) From equations (4.22) one might be tempted to try to implement SOR as

```
for iter=1:maxiter
   uGS = (DA - LA) \ (UA*u + rhs);
   u = u + omega * (uGS - u);
   end
```

where the matrices have been defined as in `iter_bvp_Asplit.m`. Try this computationally and observe that it does not work well. Explain what is wrong with this and derive the correct expression (4.24).

Implementing the above into the script, we find the following error plot and solution plot:





Clearly, this modification does not work. The reason this change does not work is because it does not account for the implicit steps. That is, we are assuming the Gauss-Seidel method is an explicit method for the SOR algorithm. This is clearly not the case, since Gauss-Seidel is implicit by definition! Let us derive the formula given by (4.24).

Recall the definition of the SOR method:

$$u_{ij}^{[k+1]} = u_{ij}^{[k]} + \omega(u_{ij}^{GS} - u_{ij}^{[k]})$$

where

$$u_{ij}^{GS} = \frac{1}{4}(u_{i-1,j}^{[k+1]} + u_{i+1,j}^{[k]} + u_{i,j-1}^{[k+1]} + u_{i,j+1}^{[k]}) - \frac{h^2}{4} f_{ij}$$

Substituting this into our definition for the SOR method, we find (after simplifying):

$$u_{ij}^{[k+1]} - \frac{\omega}{4} u_{i-1,j}^{[k+1]} - \frac{\omega}{4} u_{i,j-1}^{[k+1]} = (1-\omega) u_{ij} + \frac{\omega}{4} u_{i+1,j}^{[k]} + \frac{\omega}{4} u_{i,j+1}^{[k]} - \frac{\omega h^2}{4} f_{ij}$$

Writing in matrix form, we find

$$-\frac{4}{\omega h^2} \begin{bmatrix} 1 & & & & \\ -\omega/4 & 1 & & & \\ & -\omega/4 & 1 & & \\ & & \ddots & \ddots & \\ & & & -\omega/4 & 1 \end{bmatrix} U^{[k+1]} = -\frac{4}{\omega h^2} \begin{bmatrix} 1-\omega & \omega/4 & & & \\ & 1-\omega & \omega/4 & & \\ & & \ddots & \ddots & \\ & & & 1-\omega & \omega/4 \\ & & & & 1-\omega \end{bmatrix} U^{[k]} + f_{ij}$$

which leads us to

$$M = \frac{1}{\omega}(D - \omega L) \ ; \ N = \frac{1}{\omega}((1-\omega)D + \omega U)$$

where

$$D = -\frac{4}{h^2} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \ ; \ L = -\frac{4}{h^2} \begin{bmatrix} 0 & & & \\ \omega/4 & 0 & & \\ & \ddots & \ddots & \\ & & \omega/4 & 0 \end{bmatrix} \ ; \ U = -\frac{4}{h^2} \begin{bmatrix} 0 & \omega/4 & & \\ & \ddots & \ddots & \\ & & 0 & \omega/4 \\ & & & 0 \end{bmatrix}$$

Which is what we wished to show.

Exercise 4.2 *(Forward vs. backward Gauss-Seidel)*

(a) The Gauss-Seidel method for the discretization of $u''(x) = f(x)$ takes the form (4.5) if we assume we are marching forwards across the grid, for $i = 1, 2, \ldots, m$. We can also define a *backwards Gauss-Seidel method* by setting

$$u_i^{[k+1]} = \frac{1}{2}(u_{i-1}^{[k]} + u_{i+1}^{[k+1]} - h^2 f_i), \qquad \text{for } i = m,\ m-1,\ m-2,\ \ldots,\ 1. \tag{E4.2a}$$

Show that this is a matrix splitting method of the type described in Section 4.2 with $M = D - U$ and $N = L$.

Rearranging the above equation, we find the following:

$$-\frac{2}{h^2} u_i^{[k+1]} + \frac{1}{h^2} u_{i+1}^{[k+1]} = f_i - \frac{1}{h^2} u_{i-1}^{[k]}$$

which we may write as

$$-\frac{1}{h^2} \begin{bmatrix} 2 & -1 & & \\ & \ddots & \ddots & \\ & & 2 & -1 \\ & & & 2 \end{bmatrix} U^{[k+1]} = -\frac{1}{h^2} \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} U^{[k]} + f_i$$

Notice that this is a matrix splitting method for $M = D - U$ and $N = L$ where

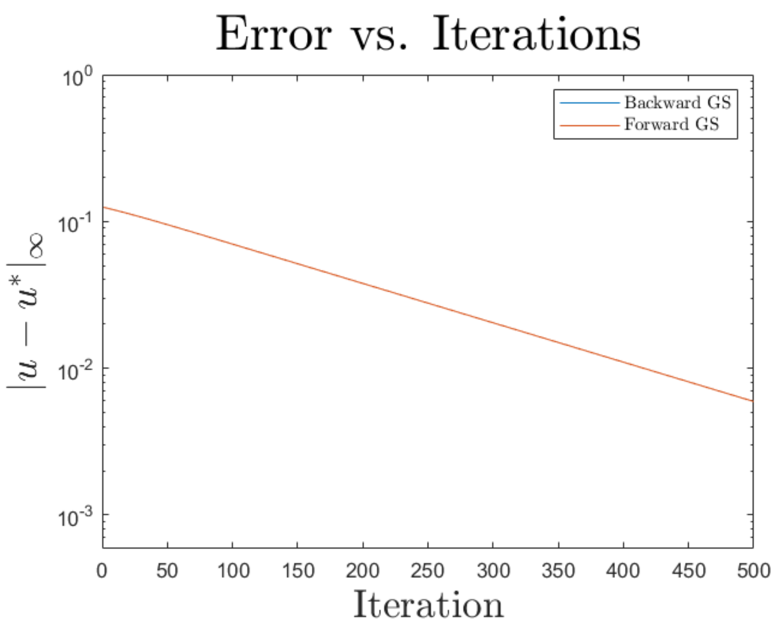
$$D = -\frac{1}{h^2} \begin{bmatrix} 2 & & & \\ & 2 & & \\ & & \ddots & \\ & & & 2 \end{bmatrix} \ ; \ U = -\frac{1}{h^2} \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ & & & 0 \end{bmatrix} \ ; \ L = -\frac{1}{h^2} \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}$$

(b) Implement this method in `iter_bvp_Asplit.m` and observe that it converges at the same rate as forward Gauss-Siedel for this problem.

Swapping the position of LA and UA in the code (see code snippet below)

```
case 'GS'
   M = DA - UA;
   N = LA;
```

and running the forward Gauss-Seidel, we observe the following error plot for the forward and backward Gauss-Seidel methods:



Notice that the error curve for the forward and backward Gauss-Seidel methods are indistinguishable.