Condition Numbers and Iterative Solvers

Susanne Brenner and Li-Yeng Sung
(modified by Douglas B. Meade)
Department of Mathematics

Overview

This investigation into iterative solvers for $\mathbf{Ax} = \mathbf{b}$ concludes with a comparison of the convergence properties and computational complexity of the Jacobi and Gauss–Seidel methods.

Recall that each Jacobi iteration requies $O(n)$ flops and each Gauss–Seidel iteration requires $O(n^2)$ flops. In order for the Gauss–Seidel method to be "cheaper" than the Jacobi method for a particular problem, the iterates will need to converge to the exact solution in fewer iterations.

But, it must also be remembered that the Jacobi and Gauss–Seidel methods have different conditions under which they are assured to converge to the exact solution. It is important to check if a method converges before applying it to a problem.

There are no new `MATLAB` commands introduced in this lab. However, the `Jacobi` and `GS` M-files created in Lab 10 are needed to complete this lab.

Part I

For many systems $\mathbf{Ax} = \mathbf{b}$ the Gauss–Seidel method performs better than the Jacobi method. To understand this, write $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$ and observe that the lower triangular part of $\mathbf{A}$, i.e., $\mathbf{L} + \mathbf{D}$, contains more of the information in $\mathbf{A}$ than the diagonal of $\mathbf{A}$, i.e., $\mathbf{D}$. From another point of view, it is seen that the Gauss-Seidel method is the Jacobi method with immediate updates. That is, for the Gauss–Seidel method component k of $\mathbf{x}^{\mathbf{new}}$ is computed using the first $k-1$ components of $\mathbf{x}^{\mathbf{new}}$ and the last $n-k$ components of $\mathbf{x}^{\mathbf{old}}$ whereas the Jacobi method uses only $\mathbf{x}^{\mathbf{old}}$.

Example 1

Consider the system $\begin{bmatrix} 5 & 0 & 1 \\ -1 & 5 & 1 \\ 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -3 \\ 14 \end{bmatrix}$. The solution to this system is $\mathbf{x} = \begin{bmatrix} 1 \\ -1 \\ 3 \end{bmatrix}$.

In a `MATLAB` session, create the matrix $\mathbf{A}$ and right-hand side vector $\mathbf{b}$ for this system. Then enter the following commands:

```
>> format long g
>> tol = 1e-8;                          % relative error tolerance
>> x0 = rand(3,1);                      % random initial guess
>> [x,rel,niter] = Jacobi(A,b,x0,tol,50)
>> [x,rel,niter] = GS(A,b,x0,tol,50)
```

Use these results to answer the following questions:

- Did both methods converge?
- Do they have the same limit?
- Which method converged faster?
- How do the flops counts compare? (Requires some additional commands.)
- How do these results depend on the initial vector?

Example 2

Consider the system $\begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$. The solution to this system is $\mathbf{x} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$.

Create the matrix **A** and right-hand side vector **b** for this system. Then enter the following commands:

```
>> x0 = rand(3,1);                    % random initial guess
>> [x,rel,niter] = Jacobi(A,b,x0,tol,50)
>> [x,rel,niter] = GS(A,b,x0,tol,50)
```

Use these results to answer the following questions:

- Did both methods converge?
- Do they have the same limit?
- Which method converged faster?
- How do the flops counts compare? (Requires some additional commands.)
- How do these results depend on the initial vector?

Example 3

Consider the system $\begin{bmatrix} 2 & 1 & 2 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 3 \\ 6 \end{bmatrix}$. The solution to this system is $\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$.

Create the matrix **A** and right-hand side vector **b** for this system. Then enter the following commands:

```
>> x0 = rand(3,1);                    % random initial guess
>> [x,rel,niter] = Jacobi(A,b,x0,tol,50)
>> [x,rel,niter] = GS(A,b,x0,tol,50)
```

Use these results to answer the following questions:

- Did both methods converge?
- Do they have the same limit?
- Which method converged faster?
- How do the flops counts compare? (Requires some additional commands.)
- How do these results depend on the initial vector?

Concluding Remarks

Notice that the only coefficient matrix that is strictly row diagonally dominant is Example 1. The convergence of both the Jacobi and Gauss–Seidel methods in Example 1 is consistent with this observation. In Examples 2 and 3 the coefficient matrices are not strictly row diagonally dominant. The convergence theorems do not say anything about these cases. In particular, it is not surprising when one (or both) of the methods does not converge.

In Example 2 you should have obverved that the Jacobi method solves this problem in exactly five steps! (Actually, the exact solution is reached after four iterations but the estimated relative error after the fourth iteration still exceeds the tolerance.)

Clear all variables before you begin to work on Part II.