On the Convergence of Iterative Methods

Because the solution of linear systems is so easy with Matlab's backslash operator, we tend to use this direct elimination method for most small and medium-sized applications requiring this capability. However, it needs to be emphasized that, in contrast, most large simulation tools (requiring the solution of thousands of coupled equations that simulate PDE-based models) use iterative techniques to solve the resultant system of equations. Then again, an introductory numerical methods course (like this one) is not the best place to address these type of simulation tools (these are more appropriate in upper-level applications-oriented courses). Thus, we have somewhat of a dilemma on how to introduce students to iterative methods since, for most of the problems we will face in this course, Matlab's **x** = **A\b** capability is the way to go and, at this point, we are not ready to tackle large PDE-based models.

Thus, with no explicit practical application readily available, we will "make up" a series of problems that illustrate some key aspects concerning the convergence properties of the Gauss Seidel method with successive relaxation (SR method). We have already discussed these methods in the main Lesson #6 Lecture Notes and the **srdemo_lesson6.m** example clearly shows the key relationships associated with convergence rate, the spectral radius of the iteration matrix, and the relaxation parameter, α. However, for large systems, the spectral radius is not practical to compute, so we often simply rely on a ***rule of thumb*** that involves ***diagonal dominance*** to estimate whether an iterative method will converge for the given system.

To explore the relationship of diagonal dominance (or near diagonal dominance) and convergence more closely, let's consider the following three cases:

Case 1:

$$\begin{bmatrix} 9 & 3 & 1 & 0 & 0 & 0 \\ 3 & 9 & 3 & 1 & 0 & 0 \\ 1 & 3 & 9 & 3 & 1 & 0 \\ 0 & 1 & 3 & 9 & 3 & 1 \\ 0 & 0 & 1 & 3 & 9 & 3 \\ 0 & 0 & 0 & 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \\ 0 \\ 20 \end{bmatrix}$$

Case 2: Same as Case 1 except use 6 along the diagonal of the matrix.

Case 3: Same as Case 1 except use 3 along the diagonal of the matrix.

Now, for a matrix to be diagonally dominant, we must have

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \qquad \text{for all i} \tag{1}$$

Applying this condition to the above three cases gives the following observations:

1. Case 1 is diagonally dominant. The condition in eqn. (1) is satisfied for all rows. In particular, the inequality test gives 9 > 4, 9 > 7, 9 > 8, 9 > 8, 9 > 7, and 9 > 4, for rows 1 through 6 respectively.
2. Case 2 is **not** diagonally dominant. The condition in eqn. (1) is not satisfied for rows 2 through 5, where the inequality test gives 6 < 7, 6 < 8, 6 < 8, and 6 < 7, respectively. However, we should note that, although the formal test in eqn. (1) failed, the sum of the

magnitudes of the off-diagonal elements does not greatly exceed the magnitude of the diagonal value (here the worse case ratio of these terms is 8/6 = 1.333). Sometimes this is referred to as "nearly diagonally dominant", but this is surely a subjective description of the current situation since "near" and "far away" are only qualitative descriptions, not quantitative relationships.

3. Case 3 is also **not** diagonally dominant. The condition in eqn. (1) is not satisfied for any row, where the inequality test gives 3 < 4, 3 < 7, 3 < 8, 3 < 8, 3 < 7, and 3 < 4, respectively for rows 1 through 6. Here, every row failed, and the maximum ratio of the sum of the magnitudes of the off-diagonal elements to the magnitude of the diagonal element is 8/3 = 2.667. In this case, one might say that this system is "farther away" from satisfying the diagonal dominance condition.

So, what do these observations mean? Well, as a start, we can say for sure that Case 1 will converge using the Gauss Seidel method (see the discussion on pages 19-20 of the Lesson #6 Lecture Notes). However, for Cases 2 and 3, we do not know anything -- based solely on the statement that ***diagonal dominance of the original A matrix is a sufficient (but not necessary) condition for convergence of the Gauss Seidel method***. Nevertheless, experience has shown that nearly diagonally systems are likely to converge and that a system that is far removed from satisfying the diagonal dominance condition is more likely not to converge. Note that phrases like "nearly diagonally dominant", "far removed", "likely or not likely to converge" are purposely rather vague because these are simply ***rules of thumb*** that have been developed from practical experience, but that have no formal mathematical basis.

Using these rather vague terms, clearly Case 2 is more "nearly diagonally dominant" than Case 3, and Case 3 is clearly "farther removed" from satisfying the formal diagonal dominance condition. Based on this observation, one could argue that Case 2 is "more likely" to converge than Case 3 (but there is no formal mathematical basis for this statement).

Well, as a specific test of this practical experience, we attempt to solve all three of the above cases in file **conv_demo_1.m** (see Table 1), first using Matlab's backslash operator and then using the **sr.m** routine discussed in the Lesson #6 notes (recall the **sr.m** does not perform any partial pivoting -- but the equations for this demo have been written so that this is not needed here).

The results of the **x** = **A\b** operation for the three cases are:

$$x_1 = \begin{bmatrix} 1.2510 \\ -0.4526 \\ 0.0988 \\ 0.0236 \\ -0.8536 \\ 2.5041 \end{bmatrix} \qquad x_2 = \begin{bmatrix} 2.3434 \\ -1.5354 \\ 0.5455 \\ 0.5455 \\ -2.6465 \\ 4.5657 \end{bmatrix} \qquad x_3 = \begin{bmatrix} -7.7778 \\ 11.1111 \\ 0.0000 \\ -10.0000 \\ 4.4444 \\ 5.5556 \end{bmatrix}$$

Now, when the **sr.m** routine is used with $\alpha = 1.0$ for the standard Gauss Seidel method, the first two cases converge to the above results in 12 and 20 iterations, respectively, and Case 3 did not converge at all! However, based on the above discussion of diagonal dominance (or near diagonal dominance), this result is not unexpected. Case 1 converged in the fewest iterations, Case 2, which in this case may be considered as a nearly diagonally dominant system, did converge, but it took several more iterations and, Case 3, which is far removed from being diagonally dominant, did not converge -- and all this is consistent with the behavior that was expected based on previous experience with the Gauss Seidel iteration method.

As a final note here, we computed the spectral radius of the iteration matrix for the above three cases (with $\alpha = 1.0$) where

$$\rho_1 = 0.238 \qquad \rho_2 = 0.474 \qquad \text{and} \qquad \rho_3 = 1.668$$

Clearly, since $\rho_3 > 1$, we would expect this case to diverge (as it did). Also showing a consistent trend is that $\rho_1 < \rho_2$, which supports the observation that Case 1 converged more rapidly than Case 2 -- but, since both ρ_1 and ρ_2 are less than unity, both cases did indeed converge.

Well, the point of real interest here is that ***full diagonally dominant systems always converge, and nearly diagonally dominant systems usually converge using the Gauss Seidel and Successive Relaxation methods***. This is important since many linear systems derived from physical steady state balance equations of the form

production rate – loss rate = 0

are often either fully or nearly diagonally dominant.

In particular, when setting up such models, the loss rate often shows up along the diagonal of the matrix and the production terms occur in both the off-diagonal elements and on the right hand side of the equations. Thus, the loss term is usually on the same order of magnitude as the sum of the off-diagonal terms in each equation (because of the original balance equations). This condition is what gives many physical systems their fully or nearly diagonally dominant behavior -- and this is why the Gauss Seidel method usually works for these systems!

Note: As an example of how the setup of a real system appears in matrix form, you should go back and run the **rect1d_fin_2.m** program from the Lesson #4 Illustrative Applications with N = 7 and print out the **A** matrix. I did this, and the resultant matrix is presented below:

```
A =
   -2.0258    1.0000         0         0         0         0         0
    1.0000   -2.0258    1.0000         0         0         0         0
         0    1.0000   -2.0258    1.0000         0         0         0
         0         0    1.0000   -2.0258    1.0000         0         0
         0         0         0    1.0000   -2.0258    1.0000         0
         0         0         0         0    1.0000   -2.0258    1.0000
         0         0         0         0    1.0000         0   -1.0872
```

Now, recall that each node in this 1-D discretized fin heat transfer problem has conduction heat transfer to/from the nearest neighbor to the right and left, and convection heat transfer to the ambient fluid. Thus, all the interior nodes (the end nodes are special and incorporate case-specific boundary conditions) have conduction losses to two nearest neighbors and convection losses to the ambient fluid. However, from the production perspective, only two production paths (conduction from its neighbors) show up in the **A** matrix, since the production from convection includes the ambient temperature, T_∞, and this term shows up in the right-hand side **b** vector. Thus, in this case, the magnitude of the diagonal elements (the loss terms) is slightly greater than the sum of the magnitudes of the off diagonal terms (the production terms) -- as seen in the above N×N **A** matrix for N = 7. Thus, in this case, the system is indeed diagonally dominant -- although more often than not, the discrete system is only nearly diagonally dominant. However, in either case, the resultant linear equations will often converge using the

Gauss Seidel/SR method. This discussion illustrates why the Gauss Seidel/SR method (and variants thereof) is employed in most large-scale simulation tools…

Table 1 Listing of the conv_demo_1.m program.

```
%
%   CONV_DEMO_1.M   Study the convergence of the Gauss Seidel method for three
%              different systems (with varying degrees of diagonal dominance)
%
%   This program solves three systems of equations using the Gauss Seidel/SR method
%   (with the relaxation parameter set at 1.0) as part of a demonstration in the
%   Lesson 6 Lecture Notes of what we mean by near diagonal dominance.
%
%   This script file calls the SR.M routine to apply the SR method.
%
%   File written by J. R. White, UMass-Lowell (last update:  Nov. 2017)
%

     clear all,  close all
     format compact
%
%   set parameters for Gauss Seidel method (same for all cases)
     tol = 1e-5;   M = 250;  alf = 1;  xo = zeros(6,1);
%
%   Case 1: fully diagonally dominant
     A = [ 9 3 1 0 0 0; 3 9 3 1 0 0; 1 3 9 3 1 0;
           0 1 3 9 3 1; 0 0 1 3 9 3; 0 0 0 1 3 9];
     b = [10 0 0 0 0 20]';
     disp('Case 1 solution via x = A\b: ');  x = A\b
     disp('Case 1 solution via Gauss Seidel: ');  [x,k] = sr(A,b,xo,alf,tol,M)
     if k == M, disp(' *** Warning -- Case not converged ***'); end
     L = tril(A,-1);  D = diag(diag(A));  U = triu(A,1);
     B = inv(alf*L + D)*((1-alf)*D - alf*U);   ev = eig(B);
     disp('Case 1 spectral radius: ');   p = max(abs(ev))
%
%   Case 2: "nearly" diagonally dominant
     A = [ 6 3 1 0 0 0; 3 6 3 1 0 0; 1 3 6 3 1 0;
           0 1 3 6 3 1; 0 0 1 3 6 3; 0 0 0 1 3 6];
     b = [10 0 0 0 0 20]';
     disp('Case 2 solution via x = A\b: ');  x = A\b
     disp('Case 2 solution via Gauss Seidel: ');  [x,k] = sr(A,b,xo,alf,tol,M)
     if k == M, disp(' *** Warning -- Case not converged ***'); end
     L = tril(A,-1);  D = diag(diag(A));  U = triu(A,1);
     B = inv(alf*L + D)*((1-alf)*D - alf*U);   ev = eig(B);
     disp('Case 2 spectral radius: ');   p = max(abs(ev))
%
%   Case 3: "far removed" from being diagonally dominant
     A =  [3 3 1 0 0 0; 3 3 3 1 0 0; 1 3 3 3 1 0;
           0 1 3 3 3 1; 0 0 1 3 3 3; 0 0 0 1 3 3];
     b = [10 0 0 0 0 20]';
     disp('Case 3 solution via x = A\b: ');  x = A\b
     disp('Case 3 solution via Gauss Seidel: ');  [x,k] = sr(A,b,xo,alf,tol,M)
     if k == M, disp(' *** Warning -- Case not converged ***'); end
     L = tril(A,-1);  D = diag(diag(A));  U = triu(A,1);
     B = inv(alf*L + D)*((1-alf)*D - alf*U);   ev = eig(B);
     disp('Case 3 spectral radius: ');   p = max(abs(ev))
%
%   end of program
```