Iterative Methods
for Solution of Linear Equations

We introduce the basic concepts and components of iterative methods for the solution of a system of linear equations. Consider the solution of linear equation system

\[ Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad b \in \text{span}(A) \]  

(1)

Assume that the null space of \( A \) has no nonzero vector. Denote by \( x_\ast \) the unique solution. Define

\[ r = r(x) = b - Ax, \]

as the residual at \( x \) with respect to the system \( (A, b) \). The residual at the exact solution \( x_\ast \) is zero.

The fixed-point iteration

We develop an iterative method as follows. First, a system of linear equations of (1) can be written in many mathematically equivalent forms, we consider a particular form involving the residual in order to make corrections,

\[ x = g(x) \overset{\text{def}}{=} x + Br(x), \quad B \text{ is nonsingular}. \]  

(2)

We refer to this form as the form of fixed-point equations. We shall see shortly the role of matrix \( B \).

Next, starting with an initial guess \( x_0 \), we obtain iteratively new guesses,

\[ x_{k+1} = g(x_k) = x_k + B_k \cdot r(x_k), \quad k = 0, 1, \ldots. \]  

(3)

The iteration may or may converge, depending on the selection of matrix \( B \). The error in \( x_k \) is \( e_k = x_k - x_\ast \). The iteration converges to the true solution, \( x_k \to x_\ast \) if and only if \( e_k \to 0 \).

Convergence analysis

In convergence analysis, we obtain the error propagation equations first from (2) and (3)

\[ e_{k+1} = e_k + Br(x_k) = (I - BA)e_k, \]  

(4)
The matrix \((I - BA)\) is the error propagation matrix, it relates the errors in two successive iteration steps, and hence relates the error at every step \(k\) to the initial one, \(e_k = (I - BA)^ke_0\).

It is easy to check that the necessary and sufficient condition for the errors \(e_k\) to converge to zero with arbitrary initial guess is
\[
\rho(I - BA) < 1,
\]
where \(\rho(M) = \max |\lambda(M)|\) denotes the spectrum radius of matrix \(M\). The smaller \(\rho(I - BA)\) is, the faster the convergence. The extreme case is \(\rho(I - BA) = 0\), which means \(B = A^{-1}\) and we are brought back to where we start. We are left to consider the the remaining case
\[
0 < \rho(I - BA) < 1. 
\tag{5}
\]
In this case, matrix \(B\) may be viewed as the inverse of a portion of the matrix \(A\) as we shall see shortly.

**A design approach: matrix splittings**

We choose the matrix \(B\) in the fixed-point iteration to satisfy the convergence criterion (5). Write \(A\) in a split form
\[
A = M - N, \quad M \text{ is nonsingular.} \tag{6}
\]
We refer to matrix \(M\) the splitting matrix. Then, \(A = M(I - M^{-1}N)\). Let \(B = M^{-1}\). Then, \((I - BA) = M^{-1}N\). The convergence criterion (5) becomes the following splitting criterion
\[
0 < \rho(M^{-1}N) < 1. \tag{7}
\]
The error propagation equation can be rewritten in terms of the split matrices
\[
E_{k+1} = M^{-1}NE_k, \quad E_k \overset{\text{def}}{=} x_k - x_* \tag{8}
\]
The iteration procedure (3) takes the following specific form,
\[
Mx_{k+1} = Nx_k + b. \tag{9}
\]
We note that computing the right hand side \(b_k = Nx_k + b\) involves matrix-vector multiplication and vector addition, the computation of \(x_{k+1}\) then involves the solution of \(Mx_{k+1} = b_k\). In splitting, we choose \(M\) so that the system can be solved easily.
Iterations Based on Diagonal-Triangular Splitting

Diagonal systems and triangular systems are easy to solve. Many iteration methods are based on the diagonal-triangular split form of $A$:

$$A = D - L - U,$$

where $D$ is diagonal (not necessarily equal to the diagonal portion of $A$) and nonsingular, and $L$ is strict lower triangular ($L_{ii} = 0$). In particular, $D - L$ may be the lower portion of $A$.

- **Jacobi Iteration:** $M = D = \text{diag}(A)$.

  $$D x_{k+1} = (L + U)x_k + b,$$

  Sufficient condition for convergence: $A$ is diagonally dominant.

- **Gauss-Seidel(GS) Iteration:** $M = D - L = \text{tril}(A)$.

  $$(D - L)x_{k+1} = Ux_k + b$$

  Sufficient condition for convergence: $A$ is diagonally dominant.

- **Successive Over-Relaxation(SOR):** $M = \omega^{-1}(D - \omega L)$, $\omega$ is a scalar.

  $$(D - \omega L)x_{k+1} = [(1 - \omega)D + \omega U]x_k + b.$$  

  When $\omega = 1$, SOR recovers the GS iteration.

  Sufficient condition for convergence: if $A$ is symmetric, positive definite, the SOR iteration converges for any $\omega \in (0, 2)$. There are many studies on determining the optimal parameter $\omega$.

- Symmetric SOR(SSOR):

  $$M = \frac{1}{\omega(2 - \omega)}(D - \omega L)D^{-1}(D - \omega U)$$

  It invokes two successive triangular systems solutions per iteration. Try to write down the computational procedure.

  Is the iteration matrix symmetric when $A$ is spd ?

Remarks.
In the Jacobi iteration, all elements are updated at the same time and the computation can be done in parallel in a trivial way.

In the GS/SOR iteration, the update of the \((i+1)\)th element depends on the update of the \(i\)-th element, as in the substitution procedure. There still exists some extent of parallelism in the computation.

The Jacobi iteration and the GS iteration cost the same in FLOPS per iteration. Try to find out which one converges faster for a given diagonally dominant matrix \(A\).

Matrix-vector norms are used in bounding \(\rho(M^{-1}N)\) in the convergence analysis.