Convergence of matrix-splitting methods

1 Introduction

We would like to solve a system of linear algebraic equations

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is a given non-singular square matrix, and $b \in \mathbb{R}^n$ is a given vector. The idea of matrix-splitting methods is to split the matrix into two parts:

$$A = M - N, \tag{2}$$

where M is non-singular and has some simple structure (for example diagonal or triangular), such that linear systems with this matrix are easily solvable. Then we have

$$Ax = Mx - Nx = b, \qquad \text{or} \qquad Mx = Nx + b. \tag{3}$$

Thus our algorithm becomes: chose a starting guess $x^{(0)} \in \mathbb{R}^n$, and then iterate

$$Mx^{(k+1)} = Nx^{(k)} + b. \tag{4}$$

Thus at every iteration, we need to solve a linear system with matrix M in the left hand side.

Example 1. Jacobi algorithm corresponds to the choice $M = \operatorname{diag}(A)$.

Example 2. Gauss–Seidel algorithm is obtained by selecting M to be either upper or lower triangular part of A (including the diagonal).

For the purpose of the notational convenience we will now write M^{-1}, but in algorithms this still means "solve a linear system with matrix M in the left hand side". We can write

$$x^{(k+1)} = M^{-1}[(M - A)x^{(k)} + b] = x^{(k)} + M^{-1}r^{(k)}, \tag{5}$$

where $r^{(k)} := b - Ax^{(k)}$ is the *residual* of our linear system at $x^{(k)}$. Thus the computations at every iterations are: (i) compute residual $r^{(k)}$, which only requires a matrix-vector product $Ax^{(k)}$ and not necessarily the matrix itself; (ii) solve the linear system $M\Delta x^{(k)} = r^{(k)}$; (iii) update the solution $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$.

Example 3. Consider applying Jacobi method to the system with the tri-diagonal structure

$$A = \begin{pmatrix} 2+\varepsilon & 1 & 0 & 0 & \dots & 0 \\ 1 & 2+\varepsilon & 1 & 0 & \dots & 0 \\ 0 & 1 & 2+\varepsilon & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & 2+\varepsilon & 1 \\ 0 & \dots & 0 & 0 & 1 & 2+\varepsilon \end{pmatrix},$$

where $\epsilon > 0$ is a given number.

Note that the matrix-vector product $y = Ax$ in this case requires only $O(n)$ operations and not $O(n^2)$ as for dense matrices:

$$y = (2+\varepsilon)x + \begin{pmatrix} x_2 \\ x_3 \\ \vdots \\ x_n \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{pmatrix} \tag{6}$$

Further, $M^{-1} = (2+\varepsilon)^{-1} I$, where I is the identity matrix. Therefore (5) reduces to:

$$x^{(k+1)} = x^{(k)} + (2-\varepsilon)^{-1} \left[b - (2+\varepsilon)x^{(k)} + \begin{pmatrix} x_2^{(k)} \\ x_3^{(k)} \\ \vdots \\ x_n^{(k)} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_{n-1}^{(k)} \end{pmatrix} \right] = (2-\varepsilon)^{-1} \left[b + \begin{pmatrix} x_2^{(k)} \\ x_3^{(k)} \\ \vdots \\ x_n^{(k)} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_{n-1}^{(k)} \end{pmatrix} \right],$$

which still requires $O(n)$ computations per iteration.

Example 4. Let us consider the example 3, but apply Gauss–Seidel method instead. Thus equation (4) reduces to

$$(2+\varepsilon)x^{(k+1)} + \begin{pmatrix} 0 \\ x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_{n-1}^{(k+1)} \end{pmatrix} = - \begin{pmatrix} x_2^{(k)} \\ x_3^{(k)} \\ \vdots \\ x_n^{(k)} \\ 0 \end{pmatrix} + b.$$

In other words, we can compute $x^{(k+1)}$ component-by-component:

$$\begin{aligned} x_1^{(k+1)} &= (2+\varepsilon)^{-1}[b_1 - x_2^{(k)}], \\ x_2^{(k+1)} &= (2+\varepsilon)^{-1}[b_2 - x_1^{(k+1)} - x_3^{(k)}], \\ &\vdots \\ x_{n-1}^{(k+1)} &= (2+\varepsilon)^{-1}[b_{n-1} - x_{n-2}^{(k+1)} - x_n^{(k)}], \\ x_n^{(k+1)} &= (2+\varepsilon)^{-1}[b_n - x_{n-1}^{(k+1)}]. \end{aligned}$$

Thus the update, unlike in the case of Jacobi, cannot be done in the arbitrary order any longer, which means the algorithm cannot be so easily parallelized.

Alternatively, we can write the iteration as follows:

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b =: Gx^{(k)} + c, \tag{7}$$

where the *iteration matrix* $G = M^{-1}N$ has been introduced. Note that the solution x to (1) also satisfies the equation

$$x = Gx + c. \tag{8}$$

Let us now introduce the *error* vector $e^{(k)} = x^{(k)} - x$. Then

$$e^{(k+1)} = Ge^{(k)} = G^2 e^{(k-1)} = \cdots = G^{k+1} e^{(0)}. \tag{9}$$

Thus the behaviour of the error in our iterative algorithm is completely determined by the iteration matrix G.

2 Convergence of matrix-splitting algorithms

Recall that vector and matrix norms are called compatible if they satisfy the inequality

$$\|Ax\| \le \|A\|\|x\|, \quad \forall A \in \mathbb{R}^{n\times n}, x \in \mathbb{R}^n.$$

For the rest of this note we assume that compatible norms are utilized.

Theorem 1. *Assume that the matrix splitting $A = M - N$ is such that the resulting iteration matrix $G = M^{-1}N$ satisfies the inequality $\|G\| < 1$. Then the matrix-splitting algorithm converges from an arbitrary starting point.*

Proof.

$$\|e^{(k)}\| = \|G^k e^{(0)}\| = \|G(G^{k-1}e^{(0)})\| \le \|G\|\|G^{k-1}e^{(0)}\| \le \cdots \le \|G\|^k \|e^{(0)}\| \to 0, \qquad (10)$$

as $k \to \infty$ when $\|G\| < 1$ for an arbitrary initial error $e^{(0)}$. □

This theorem is very easy to prove, but it is not so easy to apply. Namely, it is not quite clear how to relate the condition $\|G\|$ to the coefficients of the original matrix A and its splitting $A = M - N$.

Definition 1. The matrix A is called *strictly diagonally dominant* if

$$|A_{ii}| > \sum_{j \ne i} |A_{ij}|, \qquad \forall i = 1, \ldots, n.$$

In particular, strictly diagonally dominant matrices have non-zero elements on the diagonal.

Example 5. The matrix in example 3 is strictly diagonally dominant for any $\varepsilon > 0$.

Theorem 2. *Suppose that the matrix A is strictly diagonally dominant. Then Jacobi iteration converges from an arbitrary starting point.*

Proof.

$$G = \begin{pmatrix} A_{11}^{-1} & 0 & 0 & 0 & \ldots & 0 \\ 0 & A_{22}^{-1} & 0 & 0 & \ldots & 0 \\ 0 & 0 & A_{33}^{-1} & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & 0 & A_{n-1,n-1}^{-1} & 0 \\ 0 & \ldots & 0 & 0 & 0 & A_{nn}^{-1} \end{pmatrix} \begin{pmatrix} 0 & -A_{12} & -A_{13} & -A_{14} & \ldots & -A_{1n} \\ -A_{21} & 0 & -A_{23} & -A_{24} & \ldots & -A_{2n} \\ -A_{31} & -A_{32} & 0 & -A_{34} & \ddots & -A_{3n} \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ -A_{n-1,1} & \ldots & \ldots & -A_{n-1,n-2} & 0 & -A_{n-1,n} \\ -A_{n1} & \ldots & \ldots & -A_{n,n-2} & -A_{n,n-1} & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & -A_{12}/A_{11} & -A_{13}/A_{11} & -A_{14}/A_{11} & \ldots & -A_{1n}/A_{11} \\ -A_{21}/A_{22} & 0 & -A_{23}/A_{22} & -A_{24}/A_{22} & \ldots & -A_{2n}/A_{22} \\ -A_{31}/A_{33} & -A_{32}/A_{33} & 0 & -A_{34}/A_{33} & \ddots & -A_{3n}/A_{33} \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ -A_{n-1,1}/A_{n-1,n-1} & \ldots & \ldots & -A_{n-1,n-2}/A_{n-1,n-1} & 0 & -A_{n-1,n}/A_{n-1,n-1} \\ -A_{n1}/A_{n,n} & \ldots & \ldots & -A_{n,n-2}/A_{n,n} & -A_{n,n-1}/A_{n,n} & 0 \end{pmatrix}$$

Thus

$$\|G\|_\infty = \max_i \sum_{j \ne i} \frac{|A_{ij}|}{|A_{ii}|} < 1, \qquad (11)$$

because the matrix is strictly diagonally dominant. Finally, Theorem 1 implies the desired result. □

Example 6. Let us measure the norm $\|G\|_\infty$ given by (11) for the matrix in example 3:

$$\|G\|_\infty = \max\left\{\frac{1}{2+\varepsilon}, \frac{2}{2+\varepsilon}\right\} = \frac{2}{2+\varepsilon} = \frac{1}{1+\varepsilon/2}.$$

Thus, in the worst case, the error after k iterations of Jacobi iteration satisfies $\|e^{(k)}\|_\infty \le (1 + \varepsilon/2)^{-k}\|e^{(0)}\|_\infty$. Suppose we are interested in reducing the error such that $\|e^{(k)}\|_\infty / \|e^{(0)}\|_\infty \le \delta$. How many iterations do we need?

$$\log(1 + \varepsilon/2)^{-k} = -k \log(1 + \varepsilon/2) \approx -k\varepsilon/2 \le \log \delta, \qquad (12)$$

(where the approximation is owing to the first order series expansion of log around 1) and therefore

$$k \geq -\frac{2\log\delta}{\varepsilon}$$

We will now try to establish a result, showing that the convergence of Gauss–Seidel method is at least as good (or as bad, depending on your point of view) as that of Jacobi, and is often faster. In the following we assume that M is set to be the lower triangular part of the matrix A, including the diagonal entries (the second possibility is to use the upper triangular part including the diagonal).

Theorem 3. *Assume that the matrix A is strictly diagonally dominant. Then after one iteration of Gauss–Seidel method we have*

$$\|e^{(k+1)}\|_\infty \leq \max_i \frac{\sum_{j>i}|A_{ij}|}{|A_{ii}| - \sum_{j<i}|A_{ij}|}\|e^{(k)}\|_\infty. \tag{13}$$

Proof. Let us write down the equation $e^{(k+1)} = Ge^{(k)} = M^{-1}Ne^{(k)}$, or $Me^{(k+1)} = Ne^{(k)}$ for the Gauss-Seidel method. For the component i of the error we have:

$$\sum_{j\leq i} A_{ij}e_j^{(k+1)} = -\sum_{j>i} A_{ij}e_j^{(k)}.$$

We will now estimate the left and the right hand sides of this equation. In the right hand side we have:

$$\left|-\sum_{j>i} A_{ij}e^{(k)}\right| \leq \sum_{j>i}|A_{ij}||e_j^{(k)}| \leq \sum_{j>i}|A_{ij}|\|e^{(k)}\|_\infty, \qquad \forall i. \tag{14}$$

To estimate the left hand side we first choose an index i such that $|e_i^{(k+1)}| = \max_i|e_i^{(k+1)}| = \|e^{(k+1)}\|_\infty$. For this particular index we can write:

$$\left|\sum_{j\leq i} A_{ij}e_j^{(k+1)}\right| \geq |A_{ii}||e_i^{(k+1)}| - \sum_{j<i}|A_{ij}||e_j^{(k+1)}| \geq |A_{ii}|\|e^{(k+1)}\|_\infty - \sum_{j<i}|A_{ij}|\|e^{(k+1)}\|_\infty. \tag{15}$$

Finally, because A is strictly diagonally dominant we have the inequality $|A_{ii}| - \sum_{j<i}|A_{ij}| > 0$, and therefore we can combine (14) and (15) into

$$\|e^{(k+1)}\|_\infty \leq \frac{\sum_{j>i}|A_{ij}|}{|A_{ii}| - \sum_{j<i}|A_{ij}|}\|e^{(k)}\|_\infty \leq \max_i \frac{\sum_{j>i}|A_{ij}|}{|A_{ii}| - \sum_{j<i}|A_{ij}|}\|e^{(k)}\|_\infty.$$

□

Does Theorem 3 even imply convergence of Gauss–Seidel method for diagonally dominant matrices? Yes it does. Indeed, let

$$0 \leq \alpha := \max_i \sum_{j\neq i}\frac{|A_{ij}|}{|A_{ii}|} < 1, \qquad \text{and} \qquad 0 \leq \delta_i := \sum_{j<i}\frac{|A_{ij}|}{|A_{ii}|} \leq \alpha$$

Then

$$\max_i \frac{\sum_{j>i}|A_{ij}|}{|A_{ii}| - \sum_{j<i}|A_{ij}|} = \max_i \frac{\sum_{j\neq i}\frac{|A_{ij}|}{|A_{ii}|} - \delta_i}{1-\delta_i} \leq \max_i \frac{\alpha-\delta_i}{1-\delta_i} =: \max_i g(\delta_i).$$

Then $g(0) = \alpha$, $g(\alpha) = 0$, and

$$g'(\delta) = \frac{-(1-\delta) - (\alpha-\delta)(-1)}{(1-\delta)^2} = \frac{\alpha-1}{(1-\delta)^2} < 0.$$

Thus g is monotonically decreasing for all $0 \le \delta \le \alpha < 1$, and the largest possible value is $g(0) = \alpha < 1$.

As a result, the reduction after one step of Jacobi iteration (in the worst case) is $\|e^{(k+1)}\|_\infty / \|e^{(k)}\|_\infty \le \alpha$, whereas for Gauss–Seidel we have $\|e^{(k+1)}\|_\infty / \|e^{(k)}\|_\infty \le \max_i g(\delta_i) \le \alpha$. Note that if all $\delta_i > 0$ (i.e. all rows have some non-zero sub-diagonal elements) then $\max_i g(\delta_i) < \alpha$ and Gauss–Seidel will typically reduce error faster than Jacobi iteration.

Example 7. Let us look at the matrix from Example 3 again. Here we have

$$\max_i \frac{\sum_{j>i} |A_{ij}|}{|A_{ii}| - \sum_{j<i} |A_{ij}|} = \max \left\{ \frac{1}{2+\varepsilon}, \frac{1}{2+\varepsilon-1}, \frac{0}{2+\varepsilon-1} \right\} = \frac{1}{1+\varepsilon}.$$

A computation similar to (12) shows that in order to reduce the error $\|e^{(k)}\|_\infty / \|e^{(0)}\|_\infty \le \delta$ we would need

$$k \ge -\frac{\log \delta}{\varepsilon}, \tag{16}$$

or approximately half of the iterations needed by Jacobi's method.