This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit

http://creativecommons.org/licenses/by-nc-sa/4.0/

Last updated: 21 July 2015
Disclaimer: Links are followed at own risk when viewing this document electronically.

Chapter 4

SYSTEMS OF LINEAR EQUATIONS

4.1 Introduction

In this chapter we consider the numerical solution of a system of linear equations. The numerical method we describe is an iterative one, known as the *Jacobi method*.

4.2 Solvability

Consider m equations in n unknowns

$$\begin{aligned} a_{11}x_1 + \ a_{12}x_2 + \cdots + \ a_{1n}x_n &= b_1 \\ a_{21}x_1 + \ a_{22}x_2 + \cdots + \ a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

which have the matrix representation

$$\mathsf{A}\boldsymbol{x} = \boldsymbol{b}. \tag{4.1}$$

From linear algebra the following is known regarding the solution of (4.1):

(a) Generally, there are no solutions if $m > n$.

(b) Generally, there are infinitely many solutions if $m < n$.

(c) If $m = n$ then A is a square matrix and possibly solvable.

4.3 Cramer's rule

For $m = n$ we define $D = \det \mathsf{A}$. Cramer's rule differentiates between three cases:

(a) If $D = 0$ and $\boldsymbol{b} \neq \boldsymbol{0}$ then, in general, there are no solutions.

(b) If $\boldsymbol{b} = \boldsymbol{0}$ and $D \neq 0$ then there is only the trivial solution $\boldsymbol{x} = \boldsymbol{0}$. Thus, a necessary condition for nontrivial solutions, if $\boldsymbol{b} = \boldsymbol{0}$, is that $D = 0$.

(c) The general case is when $D \neq 0$ and $\boldsymbol{b} \neq \boldsymbol{0}$. From Cramer's rule it follows that if D_j is the determinant obtained when the jth column of A is replaced by $\boldsymbol{b}$, then the solution is given by

$$x_j = \frac{D_j}{D} \qquad j = 1, 2, \ldots, n. \tag{4.2}$$

Determining (4.2) is equivalent to inverting the matrix equation (4.1), i.e.

$$\boldsymbol{x} = \mathsf{A}^{-1}\boldsymbol{b}.$$

In practice (4.2) is not used because of the large amount of computational effort required: it can be shown that $(n-1)n!$ multiplications and $n!-1$ additions are needed, which is approximately $nn!$ arithmetical operations in all.

4.4 The Jacobi method

If the diagonal elements a_{ii} of A are all nonzero, then we may implement an iterative technique to solve (4.1)), known as the *Jacobi method*.

Let

$$\mathsf{A} = \mathsf{D} + \mathsf{L} + \mathsf{U}$$

where D is the matrix of diagonal entries of A, L is the lower triangular part of A, and U is the upper triangular part of A. This gives

$$\begin{aligned}\mathsf{A}\boldsymbol{x} &= (\mathsf{D}+\mathsf{L}+\mathsf{U})\boldsymbol{x}\\ &= \mathsf{D}\boldsymbol{x} + (\mathsf{L}+\mathsf{U})\boldsymbol{x} = \boldsymbol{b}\end{aligned}$$

so that

$$\boldsymbol{x} = \mathsf{D}^{-1}\big(\boldsymbol{b} - (\mathsf{L}+\mathsf{U})\boldsymbol{x}\big) \tag{4.3}$$

provided that $a_{ii} \neq 0$. Note that the computation of D^{-1} is straightforward; since D is a diagonal matrix, its inverse is obtained simply by taking the reciprocal of its elements (this requires no more than n arithmetical operations). Equation (4.3) suggests the iteration scheme

$$\boldsymbol{x}^{(k)} = \mathsf{D}^{-1}\big(\boldsymbol{b} - (\mathsf{L}+\mathsf{U})\boldsymbol{x}^{(k-1)}\big), \qquad k = 0, 1, 2, \ldots \tag{4.4}$$

where k denotes the iteration count. Clearly, the implementation of this method requires an initial guess $\boldsymbol{x}^{(0)}$. It is interesting to note that (4.4)) is a form of fixed-point iteration, as seen in the previous chapter, although here it is of a multivariable nature (those variables being the components x_i of $\boldsymbol{x}$).

It can be shown that *convergence* to the exact solution $\boldsymbol{x}$ is guaranteed provided

$$|a_{ii}| > \sum_{\substack{j=1\\ i\neq j}}^{n} |a_{ij}| \tag{4.5}$$

for $i = 1, 2, \ldots, n$. In other words, on each row, the magnitude of the diagonal element must exceed the sum of the magnitudes of all the other elements on that row, and this must hold for all rows in A. An indication of the quality of the approximate solution $\boldsymbol{x}^{(k)}$ may be determined by computing the *residual*

$$\boldsymbol{r}^{(k)} = \mathsf{A}\boldsymbol{x}^{(k)} - \boldsymbol{b}$$

and the iteration process is stopped when the magnitude of $\boldsymbol{r}^{(k)}$ is less than some imposed tolerance.

It can be shown that the Jacobi method requires about n^2 arithmetical operations per iteration; in comparison with a computational implementation of Cramer's rule we have, as measure of relative efficiency,

$$\frac{Mn^2}{n \times n!} = \frac{Mn^2}{nn(n-1)!} = \frac{M}{(n-1)!}$$

where M is the number of Jacobi iterations. Clearly, if n is large we would expect the Jacobi method to be more efficient.

Example 4.1. Consider

$$\mathsf{A} = \begin{bmatrix} 5 & 3 & 1 \\ -2 & 3 & 0 \\ 6 & -1 & 8 \end{bmatrix} = \underbrace{\begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 8 \end{bmatrix}}_{\mathsf{D}} + \underbrace{\begin{bmatrix} 0 & 3 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathsf{U}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 6 & -1 & 0 \end{bmatrix}}_{\mathsf{L}}$$

$$\boldsymbol{b} = \begin{bmatrix} 3 \\ 0 \\ -7 \end{bmatrix}$$

Clearly, A satisfies (4.5) so that the Jacobi method may be used. Starting with

$$\boldsymbol{x}^{(0)} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$$

we find, after 40 iterations,

$$\boldsymbol{x}^{(40)} = \begin{bmatrix} 0.61184210526407 \\ 0.40789473683653 \\ -1.28289473683631 \end{bmatrix} \qquad \boldsymbol{r}^{(40)} = \begin{bmatrix} -0.06 \\ -0.18 \\ 0.57 \end{bmatrix} \times 10^{-10}$$

with $\left|\boldsymbol{r}^{(40)}\right| = 6.1 \times 10^{-11}$. The true solution is

$$\boldsymbol{x} = \begin{bmatrix} 0.61184210526316 \\ 0.40789473684210 \\ -1.28289473684211 \end{bmatrix}$$

and we see that the difference between each of these entries and those in $\boldsymbol{x}^{(40)}$ is less than 10^{-11}.

We note that there are other types of iterative methods for solving linear systems, such as the *Gauss-Seidel* method and *successive over-relaxtion* (SOR), but they are similar in spirit to the Jacobi method, which is the simplest of the three. Like the Jacobi method, these other methods also require that the diagonal entries of A must all be nonzero. Furthermore, for large n, all these iterative methods become relatively more efficient (with respect to arithmetic computation) than direct inversion.