LINEAR SYSTEMS

Consider the following example of a linear system:

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= -5 \\ -x_1 + x_3 &= -3 \\ 3x_1 + x_2 + 3x_3 &= -3 \end{aligned}$$

Its unique solution is

$$x_1 = 1, \quad x_2 = 0, \quad x_3 = -2$$

In general we want to solve n equations in n unknowns. For this, we need some simplifying notation. In particular we introduce *arrays*. We can think of these as means for storing information about the linear system in a computer. In the above case, we introduce

$$A = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 1 \\ 3 & 1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} -5 \\ -3 \\ -3 \end{bmatrix}, \quad x = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}$$

These arrays completely specify the linear system and its solution. We also know that we can give meaning to multiplication and addition of these quantities, calling them *matrices* and *vectors*. The linear system is then written as

$$Ax = b$$

with Ax denoting a matrix-vector multiplication.

The general system is written as

$$\begin{aligned} a_{1,1}x_1 + \cdots + a_{1,n}x_n &= b_1 \\ &\vdots \\ a_{n,1}x_1 + \cdots + a_{n,n}x_n &= b_n \end{aligned}$$

This is a system of n linear equations in the n unknowns $x_1, \ldots, x_n$. This can be written in matrix-vector notation as

$$Ax = b$$

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{bmatrix}, \; b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \; x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

A TRIDIAGONAL SYSTEM

Consider the *tridiagonal* linear system

$$\begin{array}{rcl} 3x_1 - x_2 & = & 2 \\ -x_1 + 3x_2 - x_3 & = & 1 \\ & \vdots & \\ -x_{n-2} + 3x_{n-1} - x_n & = & 1 \\ -x_{n-1} + 3x_n & = & 2 \end{array}$$

The solution is

$$x_1 = \cdots = x_n = 1$$

This has the associated arrays

$$A = \begin{bmatrix} 3 & -1 & 0 & \cdots & 0 \\ -1 & 3 & -1 & 0 & \\ & & \ddots & & \\ \vdots & & -1 & 3 & -1 \\ 0 & \cdots & & -1 & 3 \end{bmatrix}, \ b = \begin{bmatrix} 2 \\ 1 \\ \vdots \\ 1 \\ 2 \end{bmatrix}, \ x = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

SOLVING LINEAR SYSTEMS

Linear systems $Ax = b$ occur widely in applied mathematics. They occur as direct formulations of "real world" problems; but more often, they occur as a part of the numerical analysis of some other problem. As examples of the latter, we have the construction of spline functions, the numerical solution of systems of nonlinear equations, ordinary and partial differential equations, integral equations, optimization problems, graph theory, search algorithms.

There are many ways of classifying linear systems.

Size: Small, moderate, and large. This of course varies with the machine you are using. Most PCs are now being sold with a memory of 1-8 gigabytes (Gb). My seven year old HP Quadcore has 8 Gb of main memory.

For a matrix A of order $n \times n$, it will take $8n^2$ bytes to store it in double precision. Thus a matrix of order 50,000 will need around 20 Gb of storage. The latter would be too large for most present day PCs if the matrix was to be stored in the computer's memory; although one can easily expand a PC to contain much more memory than this.

Sparse vs. Dense. Many linear systems have a matrix A in which almost all the elements are zero. These matrices are said to be *sparse*. For example, it is quite common to work with tridiagonal matrices

$$A = \begin{bmatrix} a_1 & c_1 & 0 & \cdots & 0 \\ b_2 & a_2 & c_2 & 0 & \vdots \\ 0 & b_3 & a_3 & c_3 & \\ \vdots & & & \ddots & \\ 0 & \cdots & & b_n & a_n \end{bmatrix}$$

in which the order is 10^4 or much more. For such matrices, it does not make sense to store the zero elements; and the sparsity should be taken into account when solving the linear system $Ax = b$. Also, the sparsity need not be as regular as in this example.

BASIC DEFINITIONS & THEORY

A *homogeneous* linear system $Ax = b$ is one for which the right hand constants are all zero. Using vector notation, we say b is the zero vector for a homogeneous system. Otherwise the linear system is call *non-homogeneous*.

Theorem. The following are equivalent statements.
(1) For each b, there is exactly one solution x.
(2) For each b, there is a solution x.
(3) The homogeneous system $Ax = 0$ has only the solution $x = 0$.
(4) $\det(A) \neq 0$.
(5) A^{-1} exists. [The matrix inverse and determinant are introduced in §6.2, but they belong as a part of this theorem.]

EXAMPLE. Consider again the tridiagonal system

$$\begin{array}{rcl} 3x_1 - x_2 & = & 2 \\ -x_1 + 3x_2 - x_3 & = & 1 \\ & \vdots & \\ -x_{n-2} + 3x_{n-1} - x_n & = & 1 \\ -x_{n-1} + 3x_n & = & 2 \end{array}$$

The homogeneous version is simply

$$\begin{array}{rcl} 3x_1 - x_2 & = & 0 \\ -x_1 + 3x_2 - x_3 & = & 0 \\ & \vdots & \\ -x_{n-2} + 3x_{n-1} - x_n & = & 0 \\ -x_{n-1} + 3x_n & = & 0 \end{array}$$

Assume $x \neq 0$, and therefore that x has nonzero components. Let x_k denote a component of maximum size:

$$|x_k| = \max_{1 \leq j \leq n} \left| x_j \right|$$

Consider now equation k, and assume $1 < k < n$. Then

$$\begin{aligned} -x_{k-1} + 3x_k - x_{k+1} &= 0 \\ x_k &= \frac{1}{3}\left(x_{k-1} + x_{k+1}\right) \\ |x_k| &\le \frac{1}{3}\left(|x_{k-1}| + |x_{k+1}|\right) \\ &\le \frac{1}{3}\left(|x_k| + |x_k|\right) \\ &= \frac{2}{3}|x_k| \end{aligned}$$

This implies $x_k = 0$, and therefore $x = 0$. A similar proof is valid if $k = 1$ or $k = n$, using the first or the last equation, respectively.

Thus the original tridiagonal linear system $Ax = b$ has a unique solution x for each right side b.

METHODS OF SOLUTION

There are two general categories of numerical methods for solving $Ax = b$.

Direct Methods: These are methods with a finite number of steps; and they end with the exact solution x, provided that all arithmetic operations are exact. The most used of these methods is *Gaussian elimination*, which we begin with. There are other direct methods, especially for sparse systems, but we do not study them here.

Iteration Methods: These are used in solving all types of linear systems, but they are most commonly used with large sparse systems, especially those produced by discretizing partial differential equations. This is an active area of research.

MATRICES in Matlab

Consider the matrices

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \\ 3 & 3 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

In *MATLAB*, A can be created as follows.

$$A = [1 \quad 2 \quad 3; \quad 2 \quad 2 \quad 3; \quad 3 \quad 3 \quad 3];$$
$$A = [1,\ 2,\ 3;\ 2,\ 2,\ 3;\ 3,\ 3,\ 3];$$
$$\begin{aligned} A = [1 \quad 2 \quad 3 \\ 2 \quad 2 \quad 3 \\ 3 \quad 3 \quad 3]\,; \end{aligned}$$

Commas can be used to replace the spaces. The vector b can be created by

$$b = ones(3, 1);$$

Consider setting up the matrices for the system $Ax = b$ with

$$A_{i,j} = \max\{i, j\}, \qquad b_i = 1, \qquad 1 \le i, j \le n$$

One way to set up the matrix A is as follows:

$$\begin{array}{l} A = zeros(n,n); \\ \textbf{for} \quad i = 1:n \\ \qquad A(i, 1:i) = i; \\ \qquad A(i, i+1:n) = i+1:n; \\ \textbf{end} \end{array}$$

and set up the vector b by

$$b = ones(n,1);$$