A primer on matrices

Stephen Boyd

August 14, 2007

These notes describe the notation of matrices, the mechanics of matrix manipulation, and how to use matrices to formulate and solve sets of simultaneous linear equations.

We *won't* cover

- linear algebra, *i.e.*, the underlying mathematics of matrices

- numerical linear algebra, *i.e.*, the algorithms used to manipulate matrices and solve linear equations

- software for forming and manipulating matrices, *e.g.*, Matlab, Mathematica, or Octave

- how to represent and manipulate matrices, or solve linear equations, in computer languages such as C/C++ or Java

- applications, for example in statistics, mechanics, economics, circuit analysis, or graph theory

1 Matrix terminology and notation

Matrices

A *matrix* is a rectangular array of numbers (also called *scalars*), written between square brackets, as in

$$A = \begin{bmatrix} 0 & 1 & -2.3 & 0.1 \\ 1.3 & 4 & -0.1 & 0 \\ 4.1 & -1 & 0 & 1.7 \end{bmatrix}.$$

An important attribute of a matrix is its *size* or *dimensions*, *i.e.*, the numbers of *rows* and *columns*. The matrix A above, for example, has 3 rows and 4 columns, so its size is 3×4. (Size is always given as rows × columns.) A matrix with m rows and n columns is called an $m \times n$ matrix.

~~An $m \times n$ matrix is called *square* if $m = n$, *i.e.*, if it has an equal number of rows and columns. Some authors refer to an $m \times n$ matrix as *fat* if $m < n$ (fewer rows than columns), or *skinny* if $m > n$ (more rows than columns). The matrix A above is fat.~~

The *entries* or *coefficients* of a matrix are the values in the array. The i, j entry is the value in the ith row and jth column, denoted by double subscripts: the i, j entry of a matrix C is denoted C_{ij} (which is a number). The positive integers i and j are called the (row and column, respectively) *indices*. For our example above, $A_{13} = -2.3$, $A_{32} = -1$. The row index of the bottom left entry (which has value 4.1) is 3; its column index is 1.

Two matrices are *equal* if they are the same size and all the corresponding entries (which are numbers) are equal.

Vectors and scalars

A matrix with only one column, *i.e.*, with size $n \times 1$, is called a *column vector* or just a *vector*. Sometimes the size is specified by calling it an *n-vector*. The entries of a vector are denoted with just one subscript (since the other is 1), as in a_3. The entries are sometimes called the *components* of the vector, and the number of rows of a vector is sometimes called its *dimension*. As an example,

$$ v = \begin{bmatrix} 1 \\ -2 \\ 3.3 \\ 0.3 \end{bmatrix} $$

Ignore this cross-out (not sure why it's there)

is a 4-vector (or 4×1 matrix, or vector of dimension 4); its third component is $v_3 = 3.3$.

Similarly, a matrix with only one row, *i.e.*, with size $1 \times n$, is called a *row vector*. As an example,

$$ w = \begin{bmatrix} -2.1 & -3 & 0 \end{bmatrix} $$

is a row vector (or 1×3 matrix); its third component is $w_3 = 0$.

Sometimes a 1×1 matrix is considered to be the same as an ordinary number. In the context of matrices and scalars, ordinary numbers are often called *scalars*.

Notational conventions for matrices, vectors, and scalars

Some authors try to use notation that helps the reader distinguish between matrices, vectors, and scalars (numbers). For example, Greek letters (α, β, ...) might be used for numbers, lower-case letters (a, x, f, ...) for vectors, and capital letters (A, F, ...) for matrices. Other notational conventions include matrices given in bold font ($\mathbf{G}$), or vectors written with arrows above them ($\vec{a}$).

Unfortunately, there are about as many notational conventions as authors, so you should be prepared to figure out what things are (*i.e.*, scalars, vectors, matrices) despite the author's notational scheme (if any exists).

Zero and identity matrices

The zero matrix (of size $m \times n$) is the matrix with all entries equal to zero. Sometimes the zero matrix is written as $0_{m \times n}$, where the subscript denotes the size. But often, a zero matrix is denoted just 0, the same symbol used to denote the number 0. In this case you'll

have to figure out the size of the zero matrix from the context. (More on this later.) When a zero matrix is a (row or column) vector, we call it a zero (row or column) vector.

Note that zero matrices of different sizes are different matrices, even though we use the same symbol to denote them (*i.e.*, 0). In programming we call this *overloading*: we say the symbol 0 is overloaded because it can mean different things depending on its context (*i.e.*, the equation it appears in).

An identity matrix is another common matrix. It is always square, *i.e.*, has the same number of rows as columns. Its *diagonal* entries, *i.e.*, those with equal row and column index, are all equal to one, and its off-diagonal entries (those with unequal row and column indices) are zero. Identity matrices are denoted by the letter I. Sometimes a subscript denotes the size, as in I_4 or $I_{2\times 2}$. But more often the size must be determined from context (just like zero matrices). Formally, the identity matrix of size n is defined by

$$I_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Perhaps more illuminating are the examples

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

which are the 2×2 and 4×4 identity matrices. (Remember that both are denoted with the same symbol, namely, I.) The importance of the identity matrix will become clear later.

Unit and ones vectors

A vector with one component one and all others zero is called a *unit vector*. The ith unit vector, whose ith component is 1 and all others are zero, is usually denoted e_i. As with zero or identity matrices, you usually have the figure out the dimension of a unit vector from context. The three unit 3-vectors are:

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Note that the n columns of the $n \times n$ identity matrix are the n unit n-vectors. Another term for e_i is ith *standard basis vector*. Also, you should watch out, because some authors use the term 'unit vector' to mean a vector of length one. (We'll explain that later.)

Another common vector is the one with all components one, sometimes called the *ones vector*, and denoted $\mathbf{1}$ (by some authors) or e (by others). For example, the 4-dimensional ones vector is

$$\mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

2   Matrix operations

Matrices can be combined using various operations to form other matrices.

Matrix transpose

If A is an $m \times n$ matrix, its *transpose*, denoted A^T (or sometimes A'), is the $n \times m$ matrix given by $\left(A^T\right)_{ij} = A_{ji}$. In words, the rows and columns of A are transposed in A^T. For example,

$$\begin{bmatrix} 0 & 4 \\ 7 & 0 \\ 3 & 1 \end{bmatrix}^T = \begin{bmatrix} 0 & 7 & 3 \\ 4 & 0 & 1 \end{bmatrix}.$$

Transposition converts row vectors into column vectors, and vice versa. If we transpose a matrix twice, we get back the original matrix: $\left(A^T\right)^T = A$.

Matrix addition

Two matrices *of the same size* can be added together, to form another matrix (of the same size), by adding the corresponding entries (which are numbers). Matrix addition is denoted by the symbol $+$. (Thus the symbol $+$ is overloaded to mean scalar addition when scalars appear on its left and right hand side, and matrix addition when matrices appear on its left and right hand sides.) For example,

$$\begin{bmatrix} 0 & 4 \\ 7 & 0 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 6 \\ 9 & 3 \\ 3 & 5 \end{bmatrix}$$

A pair of row or column vectors of the same size can be added, but you cannot add together a row vector and a column vector (except when they are both scalars!).

Matrix subtraction is similar. As an example,

$$\begin{bmatrix} 1 & 6 \\ 9 & 3 \end{bmatrix} - I = \begin{bmatrix} 0 & 6 \\ 9 & 2 \end{bmatrix}.$$

Note that this gives an example where we have to figure out what size the identity matrix is. Since you can only add (or subtract) matrices of the same size, we conclude that I must refer to a 2×2 identity matrix.

Matrix addition is commutative, *i.e.*, if A and B are matrices of the same size, then $A + B = B + A$. It's also associative, *i.e.*, $(A + B) + C = A + (B + C)$, so we write both as $A + B + C$. We always have $A + 0 = 0 + A = A$, *i.e.*, adding the zero matrix to a matrix has no effect. (This is another example where you have to figure out the exact dimensions of the zero matrix from context. Here, the zero matrix must have the same dimensions as A; otherwise they could not be added.)

4

Scalar multiplication

Another operation is *scalar multiplication*: multiplying a matrix by a scalar (*i.e.*, number), which is done by multiplying every entry of the matrix by the scalar. Scalar multiplication is usually denoted by juxtaposition, with the scalar on the left, as in

$$(-2)\begin{bmatrix} 1 & 6 \\ 9 & 3 \\ 6 & 0 \end{bmatrix} = \begin{bmatrix} -2 & -12 \\ -18 & -6 \\ -12 & 0 \end{bmatrix}.$$

Sometimes you see scalar multiplication with the scalar on the right, or even scalar division with the scalar shown in the denominator (which just means scalar multiplication by one over the scalar), as in

$$\begin{bmatrix} 1 & 6 \\ 9 & 3 \\ 6 & 0 \end{bmatrix} \cdot 2 = \begin{bmatrix} 2 & 12 \\ 18 & 6 \\ 12 & 0 \end{bmatrix}, \quad \frac{\begin{bmatrix} 9 & 6 & 9 \\ 6 & 0 & 3 \end{bmatrix}}{3} = \begin{bmatrix} 3 & 2 & 3 \\ 2 & 0 & 1 \end{bmatrix},$$

but I think these look pretty ugly.

Scalar multiplication obeys several laws you can figure out for yourself, *e.g.*, if A is any matrix and α, β are any scalars, then

$$(\alpha + \beta)A = \alpha A + \beta A.$$

It's a useful exercise to identify the symbols appearing in this formula. The $+$ symbol on the left is addition of scalars, while the $+$ symbol on the right denotes matrix addition.

Another simple property is $(\alpha\beta)A = (\alpha)(\beta A)$, where α and β are scalars and A is a matrix. On the left hand side we see scalar-scalar multiplication ($\alpha\beta$) and scalar-matrix multiplication; on the right we see two cases of scalar-matrix multiplication.

Note that $0 \cdot A = 0$ (where the lefthand zero is the scalar zero, and the righthand zero is a matrix zero of the same size as A).

Matrix multiplication

It's also possible to multiply two matrices using *matrix multiplication*. You can multiply two matrices A and B provided their dimensions are *compatible*, which means the number of columns of A (*i.e.*, its *width*) equals the number of rows of B (*i.e.*, its *height*). Suppose A and B are compatible, *i.e.*, A has size $m \times p$ and B has size $p \times n$. The product matrix $C = AB$, which has size $m \times n$, is defined by

$$C_{ij} = \sum_{k=1}^{p} a_{ik}b_{kj} = a_{i1}b_{1j} + \cdots + a_{ip}b_{pj}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.$$

This rule looks complicated, but there are several ways to remember it. To find the i, j entry of the product $C = AB$, you need to know the ith row of A and the jth column of B. The

summation above can be interpreted as 'moving left to right along the ith row of A' while moving 'top to bottom' down the jth column of B. As you go, you keep a running sum of the product of the coresponding entries from A and B.

As an example, let's find the product $C = AB$, where

$$A = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 4 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & -3 \\ 2 & 1 \\ -1 & 0 \end{bmatrix}.$$

First, we check that they are compatible: A has three columns, and B has three rows, so they're compatible. The product matrix C will have two rows (the number of rows of A) and two columns (the number of columns of B). Now let's find the entries of the product C. To find the $1, 1$ entry, we move across the first row of A and down the first column of B, summing the products of corresponding entries:

$$C_{11} = (1)(0) + (2)(2) + (3)(-1) = 1.$$

To find the $1, 2$ entry, we move across the first row of A and down the second column of B:

$$C_{12} = (1)(-3) + (2)(1) + (3)(0) = -1.$$

In each product term here, the lefthand number comes from the first row of A, and the righthand number comes from the first column of B. Two more similar calculations give us the remaining entries C_{21} and C_{22}:

$$\begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 4 \end{bmatrix} \begin{bmatrix} 0 & -3 \\ 2 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -4 & 3 \end{bmatrix}.$$

At this point, matrix multiplication probably looks very complicated to you. It is, but once you see all the uses for it, you'll get used to it.

Some properties of matrix multiplication

~~Now we can explain why the identity has its name: if A is any $m \times n$ matrix, then $AI = A$ and $IA = A$, i.e., when you multiply a matrix by an identity matrix, it has no effect. (The identity matrices in the formulas $AI = A$ and $IA = A$ have different sizes — what are they?)~~

One very important fact about matrix multiplication is that it is (in general) *not commutative*: we *don't* (in general) have $AB = BA$. In fact, BA may not even make sense, or, if it makes sense, may be a different size than AB (so that equality in $AB = BA$ is meaningless). For example, if A is 2×3 and B is 3×4, then AB makes sense (the dimensions are compatible) but BA doesn't even make sense (much less equal AB). Even when AB and BA both make sense and are the same size, *i.e.*, when A and B are square, we don't (in general) have $AB = BA$. As a simple example, consider:

$$\begin{bmatrix} 1 & 6 \\ 9 & 3 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} -6 & 11 \\ -3 & -3 \end{bmatrix}, \qquad \begin{bmatrix} 0 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 6 \\ 9 & 3 \end{bmatrix} = \begin{bmatrix} -9 & -3 \\ 17 & 0 \end{bmatrix}.$$

Matrix multiplication *is* associative, *i.e.*, $(AB)C = A(BC)$ (provided the products make sense). Therefore we write the product simply as ABC. Matrix multiplication is also associative with scalar multiplication, *i.e.*, $\alpha(AB) = (\alpha A)B$, where α is a scalar and A and B are matrices (that can be multiplied). Matrix multiplication distributes across matrix addition: $A(B + C) = AB + AC$ and $(A + B)C = AC + BC$.

Matrix-vector product

A very important and common case of matrix multiplication is $y = Ax$, where A is an $m \times n$ matrix, x is an n-vector, and y is an m-vector. We can think of matrix vector multiplication (with an $m \times n$ matrix) as a function that transforms n-vectors into m-vectors. The formula is

$$y_i = A_{i1}x_1 + \cdots + A_{in}x_n, \quad i = 1, \ldots, m$$

Very Important! Also read: https://www.varsitytutors.com/ hotmath/hotmath_help/topics/ multiplying-vector-by-a-matrix

Inner product (Usually called "dot product")

Another important special case of matrix multiplication occurs when v is an row n-vector and w is a column n vector. Then the product vw makes sense, and has size 1×1, *i.e.*, is a scalar:

$$vw = v_1w_1 + \cdots + v_nw_n.$$

This occurs often in the form $x^T y$ where x and y are both n-vectors. In this case the product (which is a number) is called the *inner product* or *dot product* of the vectors x and y. Other notation for the inner product is $\langle x, y \rangle$ or $x \cdot y$. If x and y are n-vectors, then their inner product is

$$\langle x, y \rangle = x^T y = x_1y_1 + \cdots + x_ny_n.$$

But remember that the matrix product xy doesn't make sense (unless they are both scalars).

Also read https://en.wikipedia.org/wiki/Dot_product#Coordinate_definition (just the "Coordinate Definition" section)

Matrix powers

When a matrix A is square, then it makes sense to multiply A by itself, *i.e.*, to form $A \cdot A$. We refer to this matrix as A^2. Similarly, k copies of A multiplied together is denoted A^k.

(Non-integer powers, such as $A^{1/2}$ (the matrix squareroot), are pretty tricky — they might not make sense, or be ambiguous, unless certain conditions on A hold. This is an advanced topic in linear algebra.)

By convention we set $A^0 = I$ (usually only when A is invertible — see below).

~~Matrix inverse~~

~~If A is square, and there is a matrix F such that $FA = I$, then we say that A is *invertible* or *nonsingular*. We call F the *inverse* of A, and denote it A^{-1}. We can then also define $A^{-k} = (A^{-1})^k$. If a matrix is not invertible, we say it is *singular* or *noninvertible*.~~

It's important to understand that not all square matrices are invertible, *i.e.*, have inverses. (For example, a zero matrix never has an inverse.) As a less obvious example, you might try to show that the matrix

$$\begin{bmatrix} 1 & -1 \\ 2 & 2 \end{bmatrix}$$

does not have an inverse.

As an example of the matrix inverse, we have

$$\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}^{-1} = \frac{1}{3}\begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}$$

(you should check this!).

When a matrix is invertible, the inverse of the inverse is the original matrix, *i.e.*, $(A^{-1})^{-1} = A$. A basic result of linear algebra is that $AA^{-1} = I$. In other words, if you multiply a matrix by its inverse on the *right* (as well as the left), you get the identity.

It's very useful to know the general formula for a 2×2 matrix inverse:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

provided $ad - bc \neq 0$. (If $ad - bc = 0$, the matrix is not invertible.) There are similar, but much more complicated, formulas for the inverse of larger (invertible) square matrices, but they are not used in practice.

The importance of the matrix inverse will become clear when we study linear equations.

Useful identities

We've already mentioned a handful of matrix identities, that you could figure out yourself, *e.g.*, $A + 0 = A$. Here we list a few others, that are not hard to derive, and quite useful. (We're making no claims that our list is complete!)

- transpose of product: $(AB)^T = B^T A^T$

- transpose of sum: $(A + B)^T = A^T + B^T$

- inverse of product: $(AB)^{-1} = B^{-1}A^{-1}$ provided A and B are square (of the same size) and invertible

- products of powers: $A^k A^l = A^{k+l}$ (for $k, l \geq 1$ in general, and for all k, l if A is invertible)

Block matrices and submatrices

In some applications it's useful to form matrices whose entries are themselves matrices, as in

$$\begin{bmatrix} A & B & C \end{bmatrix}, \qquad \begin{bmatrix} F & I \\ 0 & G \end{bmatrix},$$

where A, B, C, F, and G are matrices (as are 0 and I). Such matrices are called *block matrices*; the entries A, B, etc. are called 'blocks' and are sometimes named by indices. Thus, F is called the $1,1$ block of the second matrix.

Of course the block matrices must have the right dimensions to be able to fit together: matrices in the same (block) row must have the same number of rows (*i.e.*, the same 'height'); matrices in the same (block) column must have the same number of columns (*i.e.*, the same 'width'). Thus in the examples above, A, B and C must have the same number of rows (*e.g.*, they could be 2×3, 2×2, and 2×1). The second example is more interesting. Suppose that F is $m \times n$. Then the identity matrix in the $1,2$ position must have size $m \times m$ (since it must have the same number of rows as F). We also see that G must have m columns, say, dimensions $p \times m$. That fixes the dimensions of the 0 matrix in the $2,1$ block — it must be $p \times n$.

As a specific example, suppose that

$$C = \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 2 & 3 \\ 5 & 4 & 7 \end{bmatrix}.$$

Then we have

$$\begin{bmatrix} D & C \end{bmatrix} = \begin{bmatrix} 0 & 2 & 3 & 2 & 2 \\ 5 & 4 & 7 & 1 & 3 \end{bmatrix}$$

Continuing this example, the expression

$$\begin{bmatrix} C \\ D \end{bmatrix}$$

doesn't make sense, because the top block has two columns and the bottom block has three. But the block expression

$$\begin{bmatrix} C \\ D^T \end{bmatrix}$$

does make sense, because now the bottom block has two columns, just like the top block.

You can also divide a larger matrix (or vector) into 'blocks'. In this context the blocks are sometimes called *submatrices* of the big matrix. For example, it's often useful to write an $m \times n$ matrix as a $1 \times n$ block matrix of m-vectors (which are just its columns), or as an $m \times 1$ block matrix of n-row-vectors (which are its rows).

Block matrices can be added and multiplied as if the entries were numbers, provided the corresponding entries have the right sizes (*i.e.*, 'conform') and you're careful about the order of multipication. Thus we have

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} AX + BY \\ CX + DY \end{bmatrix}$$

~~provided the products AX, BY, CX, and DY makes sense.~~

3 Linear equations and matrices

Linear functions

Suppose that f is a function that takes as argument (input) n-vectors and returns (as output) m-vectors. We say f is *linear* if it satisfies two properties:

- scaling: for any n-vector x and any scalar α, $f(\alpha x) = \alpha f(x)$

- superposition: for any n-vectors u and v, $f(u + v) = f(u) + f(v)$

It's not hard to show that such a function can always be represented as matrix-vector multiplication: there is an $m \times n$ matrix A such that $f(x) = Ax$ for all n-vectors x. (Conversely, functions defined by matrix-vector multiplication are linear.)

We can also write out the linear function in explicit form, *i.e.*, $f(x) = y$ where

$$y_i = \sum_{j=1}^{n} A_{ij} x_j = A_{i1} x_1 + \cdots + A_{in} x_n, \quad i = 1, \ldots, m$$

This gives a simple interpretation of A_{ij}: it gives the coefficient by which y_i depends on x_j.

Suppose an m-vector y is a linear function of the n-vector x, *i.e.*, $y = Ax$ where A is $m \times n$. Suppose also that a p-vector z is a linear function of y, *i.e.*, $z = By$ where B is $p \times m$. Then z is a linear function of x, which we can express in the simple form $z = By = (BA)x$. So matrix multiplication corresponds to composition of linear functions (*i.e.*, linear functions of linear functions of some variables).

Linear equations

Any set of m linear equations in (scalar) variables $x_1, \ldots, x_n$ can be represented by the compact matrix equation $Ax = b$, where x is a vector made from the variables, A is an $m \times n$ matrix and b is an m-vector. Let's start with a simple example of two equations in three variables:

$$1 + x_2 = x_3 - 2x_1, \quad x_3 = x_2 - 2.$$

The first thing to do is to rewrite the equations with the variables lined up in columns, and the constants on the righthand side:

$$
\begin{array}{rrrcr}
2x_1 & +x_2 & -x_3 & = & -1 \\
0x_1 & -x_2 & +x_3 & = & -2
\end{array}
$$

Now it's easy to rewrite the equations as a single matrix equation:

$$
\begin{bmatrix} 2 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}
=
\begin{bmatrix} -1 \\ -2 \end{bmatrix},
$$

so we can express the two equations in the three variables as $Ax = b$ where

$$A = \begin{bmatrix} 2 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad b = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

Solving linear equations

Now suppose we have n linear equations in n variables $x_1, \ldots, x_n$, written as the compact matrix equation $Ax = b$, where A is an $n \times n$ matrix, and b is an n-vector. Suppose that A is invertible, *i.e.*, the inverse A^{-1} exists. Let's multiply both sides of the matrix equation $Ax = b$ on the left by A^{-1}:

$$A^{-1}(Ax) = A^{-1}b.$$

The lefthand side simplifies to $A^{-1}Ax = Ix = x$, so we have actually solved the simultaneous linear equations: $x = A^{-1}b$.

Now you can see the importance of the matrix inverse: it can be used to solve simultaneous linear equations. Here we should make a comment about matrix notation. The power of matrix notation is that just a few symbols (*e.g.*, A^{-1}) can express a lot. Another (perhaps more pessimistic) way to put this is, a lot of work can be hidden behind just a few symbols (*e.g.*, A^{-1}).

Of course, you can't *always* solve n linear equations for n variables. One or more of the equations might be redundant (*i.e.*, can be obtained from the others), or the equations could be inconsistent as in $x_1 = 1$, $x_1 = 2$. When these pathologies occur, the matrix A is singular, *i.e.*, noninvertible. Conversely, when a matrix A is singular, it turns out the simultaneous linear equations $Ax = b$ are redundant or inconsistent. (These facts are studied in linear algebra.)

From a practical point of view, then, A is singular means that the equations in $Ax = b$ are redundant or inconsistent — a sign that you have set up the wrong equations (or don't have enough of them). Otherwise, A^{-1} exists, and the equations can be solved as $x = A^{-1}b$.

Solving linear equations in practice

When we solve linear equations in practice, (*i.e.*, by computer) we do not first compute the matrix A^{-1}, and then multiply it on the right by the vector b, to get the solution $x = A^{-1}b$ (although that procedure would, of course, work). Practical methods compute the solution $x = A^{-1}b$ directly.

The most common methods for computing $x = A^{-1}b$ (*i.e.*, solving a system of n simultaneous linear equations) require on the order of n^3 basic arithmetic operations. But modern computers are very fast, so solving say a set of 500 equations on a small PC class computer takes only a few seconds. But solving much larger sets of equations (say, 5000) takes much, much longer (on a PC class computer).

In many applications the matrix A has many, or almost all, of its entries equal to zero, in which case we say it is *sparse*. In terms of the associated linear equations, this means each

equation involves only some (often just a few) of the variables. It turns out that such sparse equations can be solved by computer very efficiently, using *sparse matrix techniques*. It's not uncommon to solve for hundreds of thousands of variables, with hundreds of thousands of (sparse) equations. Even on a PC class computer, solving a system of 10000 simultaneous sparse linear equations is feasible, and might take only a few seconds (but it depends on how sparse the equations are).