Solving Linear Systems in MATLAB

Leigh J. Little SUNY Brockport October 6, 2021

Contents

1 Linear Systems of Equations

You have seen linear systems of equations in the past. Consider the problem below:

$$u + v + w = 3 \quad (1)$$
$$2u + v + w = 0 \quad (2)$$
$$2u + v - w = 1. \quad (3)$$

This is called a 3×3 linear system of equations because there are 3 equations and 3 unknown quantities (u, v and w). The equations are linear because all of the unknowns appear to the first power (*i.e.*, no higher powers, roots or divisions by the unknowns).

You have solved these equations like this in the past. The process for doing this is straightforward. First, solve Equation (1) for u

$$u = 3 - v - w \quad (4)$$

and substitute this into Equations (2) and (3). For Equation (2), this gives

$$2(3 - v - w) + v + w = 0 \quad (5)$$
$$6 - 2v - 2w + v + w = 0 \quad (6)$$
$$-v - w = -6. \quad (7)$$

For Equation (3), this gives

$$2(3 - v - w) + v - w = 1 \quad (8)$$
$$6 - 2v - 2w + v - w = 1 \quad (9)$$
$$-v - 3w = -5. \quad (10)$$

This is referred to as eliminating u from the system.

Now repeat this process. Solve Equation (7) for v. This gives

$$v = 6 - w. \quad (11)$$

Substitute this into Equation (10). This gives

$$-(6 - w) - 3w = -5 \quad (12)$$
$$-6 + w - 3w = -5 \quad (13)$$
$$-2w = 1. \quad (14)$$
$$(15)$$

This can be solved for w to give

$$w = -\frac{1}{2}.$$

Substitute this into Equation (11) to get v

$$v = 6 - w = 6 - (-\frac{1}{2}) = \frac{13}{2}. \tag{16}$$

Finally, substitute the values for v and w into Equation (4) to get u

$$u = 3 - v - w = 3 - \frac{13}{2} - (-\frac{1}{2}) = -3. \tag{17}$$

The values $u = -3,\ v = \frac{13}{2}$ and $w = -\frac{1}{2}$ are the solution to the linear system of equations.

The fact that the process is straightforward doesn't make it any less tedious. There are also many ways the situation could be made worse. For example, the system could be 100 equations with 100 unknowns, the coefficients could be fractions, or the coefficients could be irrational numbers. Try solving

$$\sqrt{2}x + \sqrt{3}y = 2 \tag{18}$$

$$2x + \sqrt{5}y = 4. \tag{19}$$

by hand and see how messy the process gets.

Fortunately, this is exactly the type of problem MATLAB was originally designed to solve (and it only takes 3 steps to do it).

2 Using MATLAB to Solve the Problem

In order to get MATLAB to solve the problem, it is necessary to write the problem in a slightly different way.

2.1 Example 1

Consider the original problem:

$$u + v + w = 3 \tag{20}$$

$$2u + v + w = 0 \tag{21}$$

$$2u + v - w = 1 \tag{22}$$

and examine the first equation more closely. This equation can be decomposed as a dot product between two vectors; in particular, we can write

$$u + v + w = 3 \tag{23}$$

as

$$\begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} u \\ v \\ w \end{pmatrix} = 3. \tag{24}$$

Similarly, we can write the second equation

$$2u + v + w = 0 \tag{25}$$

as

$$\begin{pmatrix} 2 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} u \\ v \\ w \end{pmatrix} = 0 \tag{26}$$

and the third equation

$$2u + v - w = 1 \tag{27}$$

as

$$\begin{pmatrix} 2 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} u \\ v \\ w \end{pmatrix} = 1. \tag{28}$$

Combining these three results, we can write the original problem as the matrix-vector equation

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 1 & -1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} \tag{29}$$

This is normally written as

$$A\mathbf{x} = \mathbf{b} \tag{30}$$

where

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 1 & -1 \end{pmatrix} \tag{31}$$

is called the *coefficient matrix*,

$$\mathbf{x} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{32}$$

is called the *solution vector* and

$$\mathbf{b} = \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix} \tag{33}$$

is called the *right-hand side* vector.

To get MATLAB to solve the problem we need to create the matrix A and the right-hand side vector $\mathbf{b}$.

```
>> clear
>> format rat         % Turn on rational formatting for easy reading
>> A = [1 1 1 ; 2 1 1; 2 1 -1];
>> b = [3 0 1]';
```

Then, we need to issue the command

```
>> x = A\b
x =
      -3       % first component is u
      13/2     % second component is v
      -1/2     % third component is w
```

We can verify that this is the solution to the problem by computing the *residual vector* $\mathbf{r} = \mathbf{b} - A\mathbf{x}$

```
>> r = b - A*x
r =
      0
      0
      0
```

2.2 Example 2

Suppose we want to solve

$$3u - 5v + 6w = 10 \tag{34}$$

$$u + 2v - 3w = 16 \tag{35}$$

$$10u - 8v - 4w = -13. \tag{36}$$

We can write this in matrix vector form as

$$\begin{pmatrix} 3 & -5 & 6 \\ 1 & 2 & -3 \\ 10 & -8 & -4 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 10 \\ 16 \\ -13 \end{pmatrix} \tag{37}$$

Define the coefficient matrix and right-hand side vector in MATLAB

```
>> clear
>> A = [3 -5 6 ; 1 2 -3; 10 -8 -4];
>> b = [10 16 -13]';
```

then do

```
>> x = A\b
x =
    1447/134   % first component is u
    1607/134   % second component is v
    839/134    % third component is w
```

Again, we can verify that this is the solution by computing the residual vector **r**

```
>> r = b - A*x
r =
      0
      0
      0
```

3 Important Terminology

Solving linear systems of equations is an important area of mathematics called *linear algebra*. It is instructive to describe what we just did using appropriate language.

3.1 The One Equation Form

Consider one linear equation in one unknown variable. For example,

$$3(x-2) - 2(x+1) = 4x. \tag{38}$$

After some manipulation, we can write this as

$$3x = -8 \tag{39}$$

which gives

$$x = -\frac{8}{3}. \tag{40}$$

No matter how a one equation in one variable problem starts, it can always be reduced to the form

$$ax = b \tag{41}$$

which can be solved by dividing both sides by a (assuming a is not zero)

$$x = \frac{b}{a}. \tag{42}$$

If a is zero, then there are two possibilities; the first is an equation of the form

$$0x = b \tag{43}$$

which has no solution. The other possibility is an equation of the form

$$0x = 0 \tag{44}$$

which means that there are an infinite number of solutions because any value of x will satisfy the equation.

3.2 Generalizing to n Equations

The above discussion can be generalized for n equations in n unknowns. We saw that we can write this problem in the matrix-vector form

$$A\mathbf{x} = \mathbf{b} \tag{45}$$

where A is an $n \times n$ coefficient matrix, $\mathbf{x}$ is a column vector of unknowns of length n and $\mathbf{b}$ is the right-hand side column vector of length n. Based on the discussion for one equation in one unknown, it would appear that we want to do

$$\mathbf{x} = \frac{\mathbf{b}}{A} \tag{46}$$

but this is not feasible. Dividing a vector by a matrix makes no sense in linear algebra. However, we can do something equivalent. Return to the one equation in one unknown case

$$ax = b. \tag{47}$$

Instead of dividing by a, think of multiplying both sides by the inverse of a.

$$ax = b \tag{48}$$
$$a^{-1}(ax) = a^{-1}b \tag{49}$$
$$x = a^{-1}b. \tag{50}$$

This is mathematically equivalent to dividing by a.

This is the same idea we want to adopt for the case of n equations in n unknowns.

$$A\mathbf{x} = \mathbf{b} \tag{51}$$
$$A^{-1}(A\mathbf{x}) = A^{-1}\mathbf{b} \tag{52}$$
$$\mathbf{x} = A^{-1}\mathbf{b} \tag{53}$$

This makes more sense because we can multiply a matrix with a vector and a matrix with a matrix. The question is, what is A^{-1}? This matrix is called the *inverse matrix of A*.

3.3 The Identity Matrix, I

In order to gain a sense of what A^{-1} is, we first need to define a special matrix called the *identity matrix*. This matrix is denoted by I. The matrix I is the matrix version of 1. Assuming the dimensions permit the operation, any matrix or vector multiplied with I just gives the original matrix or vector, for example

$$IA = A \tag{54}$$
$$AI = A \tag{55}$$
$$Ix = x. \tag{56}$$

The identity matrix has an easy formula. For the case of a 2×2 matrix, we have

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{57}$$

while a 3×3 identity matrix looks like

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{58}$$

An identity matrix has ones on the main diagonal (*i.e.,* locations where the row and column index are equal) and is zero everywhere else.

3.4 The Definition of A^{-1}

Now that we know about I we can define A^{-1}. The inverse of A is defined to be the matrix that gives an identity when multiplied by A. This means that

$$AA^{-1} = I \tag{59}$$

and

$$A^{-1}A = I. \tag{60}$$

This is not a very satisfying definition. It turns out that computing A^{-1} is like solving the system of linear equations at the beginning of this section. The process is straightforward, but is also very tedious. We won't cover how to compute A^{-1} here, but we can make use of some MATLAB functions to give us some insights.

Consider the coefficient matrix we started with

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 1 & -1 \end{pmatrix}. \tag{61}$$

We can compute the inverse of this matrix in MATLAB using

```
>> clear
>> format rat
>> A = [1 1 1; 2 1 1; 2 1 -1];
>> AI = inv(A)
AI =
     -1              1              0
     2            -3/2            1/2
     0             1/2           -1/2
```

We can verify that `AI` is the inverse of `A` by multiplying these two matrices together and seeing if we get the identity matrix I.

```
>> A*AI
ans =
     1              0              0
     0              1              0
     0              0              1
>> AI*A
ans =
     1              0              0
     0              1              0
     0              0              1
```

Thus, `AI` satisfies the definition of A^{-1}. *It is important to note that the inverse A^{-1} is not obtained by inverting all the elements of A.*

3.5 An Important Rule in Scientific Computing

One way we can solve the original problem is to do

```
>> clear
>> A = [3 -5 6 ; 1 2 -3; 10 -8 -4];
>> b = [10 16 -13]';
>> x = inv(A)*b
```

however, this is not the way this problem is solved in real situations.

One of the most fundamental rules in scientific computing is: **Never compute A^{-1} unless it is absolutely necessary**. This is because it is an expensive computation and can be very sensitive to round-off errors. Instead, when you need to solve a system of linear equations, use the sequence

```
>> clear
>> A = [3 -5 6 ; 1 2 -3; 10 -8 -4];
>> b = [10 16 -13]';
>> x = A\b
```

The operation `x = A\b` performs a calculation that is mathematically equivalent to `x = inv(A)*b` but is cheaper and less affected by round-off errors.

3.6 Example 3

Consider the problem

$$u + v = 1 \tag{62}$$

$$2u + 2v = 2. \tag{63}$$

In MATLAB,

```
>> A = [1 1; 2 2];
>> b = [1 2]'
>> x = A\b
Warning: Matrix is singular to working precision.
x =
    NaN
    NaN
```

In this case, there is no unique solution to the problem. This is because the two equations are actually the same equation (the second is simply the first equation multiplied by 2). Mathematically, the inverse of the matrix A does not exist. It is the matrix equivalent of trying to divide by 0.

In the special case of a 2×2 system, we can interpret the solution of the system as the common point of two intersecting lines. There are three possibilities which are indicated in Figure 1. If the two lines intersect

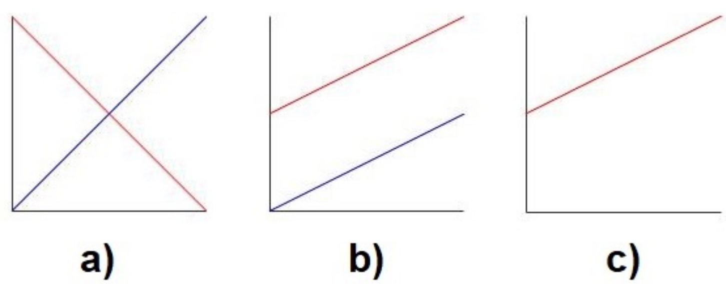

Figure 1: Three possibilities for a 2×2 system of equations: a) the two lines intersect in a single point, b) the lines are parallel, c) the lines are the same.

in a single point, then the system has a one unique solution. If the lines are parallel, there is no intersection point, so there is no solution. If the two lines are the same line, then there are infinite number of solutions because they intersect at an infinite number of points. We have similar interpretations for a 3×3 system, but now the solution represents the common point of intersection of 3 planes in 3-dimensional space.

Although the details are beyond the scope of this course, one of the challenging aspects of solving certain types of problems (for example, fluid flow at high speeds) is that round-off errors in the calculations can make a problem of type a) behave like a problem of type b).