

# Bookdown with Sage Cells

Author Name

7/6/2020

# Contents

<b>Preface</b>	<b>3</b>
<b>1 Reduced Row-Echelon Form</b>	<b>4</b>
1.1 Matrix and Vector Notation for Systems of Equations . . . . .	4
1.2 Row Operations . . . . .	9

# Preface

This is an example of a Bookdown project with SageMathCell. Sample content is provided by Beezer, R. (2015). A first course in linear algebra under the terms of the GNU Free Documentation License.

# Chapter 1

## Reduced Row-Echelon Form

By Robert Beezer *Here is the PreTeXt version of the same chapter*

After solving a few systems of equations, you will recognize that it does not matter so much what we call our variables, as opposed to what numbers act as their coefficients. A system in the variables  $x_1, x_2, x_3$  would behave the same if we changed the names of the variables to  $a, b, c$  and kept all the constants the same and in the same places. In this section, we will isolate the key bits of information about a system of equations into something called a matrix, and then use this matrix to systematically solve the equations. Along the way we will obtain one of our most important and useful computational tools.

### 1.1 Matrix and Vector Notation for Systems of Equations

**Definition M: Matrix** An  $m \times n$  is a rectangular layout of numbers from *complexes* having  $m$  rows and  $n$  columns. We will use upper-case Latin letters from the start of the alphabet  $A, B, C, \dots$  to denote matrices and squared-off brackets to delimit the layout. Many use large parentheses instead of brackets — the distinction is not important. Rows of a matrix will be referenced starting at the top and working down (i.e. row 1 is at the top) and columns will be referenced starting from the left (i.e. column 1 is at the left). For a matrix  $[A]$ , the notation  $[Aij]$  will refer to the complex number in row  $i$  and column  $j$  of  $A$ .

Be careful with this notation for individual entries, since it is easy to think that  $[Aij]$  refers to the *whole* matrix. It does not. It is just a *number*, but

is a convenient way to talk about the individual entries simultaneously. This notation will get a heavy workout once we get to Chapter M.

**Example AM: A Matrix**

$$B = \begin{bmatrix} -1 & 2 & 5 & 3 \\ 1 & 0 & -6 & 1 \\ -4 & 2 & 2 & -2 \end{bmatrix}$$

is a matrix with  $m = 3$  rows and  $n = 4$  columns. We can say that  $[B2, 3 = -6]$  while  $[B3, 4 = -2]$ .

**Sage M: Matrices**

Matrices are fundamental objects in linear algebra and in Sage, so there are a variety of ways to construct a matrix in Sage. Generally, you need to specify what types of entries the matrix contains (more on that in a minute), the number of rows and columns, and the entries themselves. First, let us dissect an example:

QQ is the set of all rational numbers (fractions with an integer numerator and denominator), 2 is the number of rows, 3 is the number of columns. Sage understands a list of items as delimited by brackets  $[.]$  and the items in the list can again be lists themselves. So  $[[1, 2, 3], [4, 5, 6]]$  is a list of lists, and in this context the inner lists are rows of the matrix.

There are various shortcuts you can employ when creating a matrix. For example, Sage is able to infer the size of the matrix from the lists of entries.

Or you can specify how many rows the matrix will have and provide one big grand list of entries, which will get chopped up, row by row, if you prefer.

It is possible to also skip specifying the type of numbers used for entries of a matrix, however this is fraught with peril, as Sage will make an informed guess about your intent. Is this what you want? Consider when you enter the single character “2” into a computer program like Sage. Is this the integer 2, the rational number  $\frac{2}{1}$ , the real number 2.00000, the complex number  $2 + 0i$ , or the polynomial  $p(x) = 2$ ? In context, us humans can usually figure it out, but a literal-minded computer is not so smart. It happens that the operations we can perform, and how they behave, are influenced by the type of the entries in a matrix. So it is important to get this right and our advice is to be explicit and be in the habit of always specifying the type of the entries of a matrix you create.

Mathematical objects in Sage often come from sets of similar objects. This set is called the “parent” of the element. We can use this to learn how Sage deduces the type of entries in a matrix. Execute the following three compute cells in the Sage notebook, and notice how the three matrices are constructed to have entries from the integers, the rationals and the reals.

Sage knows a wide variety of sets of numbers. These are known as “rings” or “fields” (see Section F), but we will call them “number systems” here. Examples

include: `ZZ` is the integers, `QQ` is the rationals, `RR` is the real numbers with reasonable precision, and `CC` is the complex numbers with reasonable precision. We will present the theory of linear algebra over the complex numbers. However, in any computer system, there will always be complications surrounding the inability of digital arithmetic to accurately represent all complex numbers. In contrast, Sage can represent rational numbers exactly as the quotient of two (perhaps very large) integers. So our Sage examples will begin by using `QQ` as our number system and we can concentrate on understanding the key concepts.

Once we have constructed a matrix, we can learn a lot about it (such as its parent). Sage is largely object-oriented, which means many commands apply to an object by using the “dot” notation. `A.parent()` is an example of this syntax, while the constructor `matrix([[1, 2, 3], [4, 5, 6]])` is an exception. Here are a few examples, followed by some explanation:

The number of rows and the number of columns should be apparent, `.base_ring()` gives the number system for the entries, as included in the information provided by `.parent()`.

Computer scientists and computer languages prefer to begin counting from zero, while mathematicians and written mathematics prefer to begin counting at one. Sage and this text are no exception. It takes some getting used to, but the reasons for counting from zero in computer programs soon becomes very obvious. Counting from one in mathematics is historical, and unlikely to change anytime soon. So above, the two rows of `A` are numbered 0 and 1, while the columns are numbered 0, 1 and 2. So `A[1,2]` refers to the entry of `A` in the second row and the third column, i.e. 6

There is much more to say about how Sage works with matrices, but this is already a lot to digest. Use the space below to create some matrices (different ways) and examine them and their properties (size, entries, number system, parent).

When we do equation operations on system of equations, the names of the variables really are not very important. Use  $x_1, x_2, x_3$ , or  $a, b, c$ , or  $x, y, z$ , it really does not matter. In this subsection we will describe some notation that will make it easier to describe linear systems, solve the systems and describe the solution sets. Here is a list of definitions, laden with notation.

**Definition CV: Column Vector.** A *column vector* of size  $m$  is an ordered list of  $m$  numbers, which is written in order vertically, starting at the top and proceeding to the bottom. At times, we will refer to a column vector as simply a *vector*. Column vectors will be written in bold, usually with lower case Latin letter from the end of the alphabet such as  $\{u\}, \{v\}, \{w\}, \{x\}, y, z$ . Some books like to write vectors with arrows, such as  $\vec{u}$ . Writing by hand, some like to put arrows on top of the symbol, or a tilde underneath the symbol, as in  $\tilde{u}$ . To refer to the *entry* or *component* of vector  $v$  in location  $i$  of the list, we write  $[v]_i$ .

### 1.1. MATRIX AND VECTOR NOTATION FOR SYSTEMS OF EQUATIONS 7

Be careful with this notation. While the symbols  $v_i$  might look somewhat substantial, as an object this represents just one entry of a vector, which is just a single complex number.

**Definition ZCV Zero Column Vector** The zero vector of size  $m$  is the column vector of size  $m$  where each entry is the number zero,

$$0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

or defined much more compactly,  $[0]_i$  for  $1 \leq i \leq m$ .

#### Coefficient Matrix

For a system of linear equations,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

the *coefficient matrix* is the  $m \times n$  matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & & & & \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

#### Vector of Constants

For a system of linear equations,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

**Solution Vector**

For a system of linear equations,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

The solution vector may do double-duty on occasion. It might refer to a list of variable quantities at one point, and subsequently refer to values of those variables that actually form a particular solution to that system.

**Matrix Representation of a Linear System.** If  $A$  is the coefficient matrix of a system of linear equations and  $b$  is the vector of constants, then we will write  $\mathcal{LS}(A, b)$  as a shorthand expression for the system of linear equations, which we will refer to as the *matrix representation* of the linear system.

**Notation for systems of linear equations**

The system of linear equations

$$\begin{aligned} 2x_1 + 4x_2 - 3x_3 + 5x_4 + x_5 &= 9 \\ 3x_1 + x_2 + \quad \quad x_4 - 3x_5 &= 0 \\ -2x_1 + 7x_2 - 5x_3 + 2x_4 + 2x_5 &= -3 \end{aligned}$$

has coefficient matrix

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

and vector of constants

$$b = \begin{bmatrix} 9 \\ 0 \\ -3 \end{bmatrix}$$

and so will be referenced as  $\mathcal{LS}(A, b)$

**Augmented Matrix.** Suppose we have a system of  $m$  equations in  $n$  variables, with coefficient matrix  $A$  and vector of constants  $\text{vect}\{b\}$ . Then the *augmented matrix* of the system of equations is the  $m \times (n+1)$  matrix whose first  $n$  columns are the columns of  $A$  and whose last column



$(n+1)$  is the column vector  $\text{\textbf{vec}}\{\mathbf{b}\}$ . This matrix will be written as  $\text{\textbf{augmented}}\{A\}\{\text{\textbf{vec}}\{\mathbf{b}\}\}$ .

The augmented matrix *represents* all the important information in the system of equations, since the names of the variables have been ignored, and the only connection with the variables is the location of their coefficients in the matrix. It is important to realize that the augmented matrix is just that, a matrix, and *not* a system of equations. In particular, the augmented matrix does not have any “solutions,” though it will be useful for finding solutions to the system of equations that it is associated with. (Think about your objects, and review Proof Technique L.) However, notice that an augmented matrix always belongs to some system of equations, and vice versa, so it is tempting to try and blur the distinction between the two. Here is a quick example.

**Augmented matrix for Archetype A**

## 1.2 Row Operations

An augmented matrix for a system of equations will save us the tedium of continually writing down the names of the variables as we solve the system. It will also release us from any dependence on the actual names of the variables. We have seen how certain operations we can perform on equations (Definition EO) will preserve their solutions (Theorem EOPSS). The next two definitions and the following theorem carry over these ideas to augmented matrices.

**Definition RO: Row Operations.** The following three operations will transform an  $m \times n$  matrix into a different matrix of the same size, and each is known as a *row operation*.

1. Swap the locations of two rows.
2. Multiply each entry of a single row by a nonzero quantity.
3. Multiply each entry of one row by some quantity, and add these values to the entries in the same columns of a second row. Leave the first row the same after this operation, but replace the second row by the new values.

We will use a symbolic shorthand to describe these row operations:

$$-2R_1 \rightarrow R_2$$

1.  $R_i \leftrightarrow R_j$ : Swap the location of rows  $i$  and  $j$ .
2.  $\alpha R_i$ : Multiply row  $i$  by the nonzero scalar  $\alpha$ .
3.  $\alpha R_i + R_j$ : Multiply row  $i$  by the scalar  $\alpha$  and add to row  $j$ .

**Row-Equivalent Matrices:** Two matrices,  $A$  and  $B$ , are *row-equivalent* if one can be obtained from the other by a sequence of row operations.

**Example TREM: Two row-equivalent matrices**

The matrices

$$A = \begin{bmatrix} 2 & -1 & 3 & 4 \\ 5 & 2 & -2 & 3 \\ 1 & 1 & 0 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 0 & 6 \\ 3 & 0 & -2 & -9 \\ 2 & -1 & 3 & 4 \end{bmatrix}$$

are row-equivalent as can be seen from

$$\begin{bmatrix} 2 & -1 & 3 & 4 \\ 5 & 2 & -2 & 3 \\ 1 & 1 & 0 & 6 \end{bmatrix}$$

$$R_1 \rightarrow R_3$$

$$\begin{bmatrix} 1 & 1 & 0 & 6 \\ 5 & 2 & -2 & 3 \\ 2 & -1 & 3 & 4 \end{bmatrix}$$

$$-2R_1 \rightarrow R_2$$

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

We can also say that any pair of these three matrices are row-equivalent.

Notice that each of the three row operations is reversible (Exercise RREF.T10), so we do not have to be careful about the distinction between “\$A\$ is row-equivalent to \$B\$” and “\$B\$ is row-equivalent to \$A\$.” (Exercise RREF.T11)

The preceding definitions are designed to make the following theorem possible. It says that row-equivalent matrices represent systems of linear equations that have identical solution sets.