

Practicing Gaussian Elimination in a Java Applet

J.C. Bugner

Valley City State University
Valley City, ND 58072
John.Bugner@vcsu.edu

Curt Hill

Department of Mathematics
Valley City State University
Valley City, ND 58072
Curt.Hill@vcsu.edu

Abstract

Gaussian elimination is an essential skill in mathematician's repertoire. It is especially important for students to practice this skill to become proficient in dealing with matrices. Since many students struggle with this, we made an interactive program that helps them learn the process by computing the tedious arithmetic for the student, but requiring the student to indicate what step that needs to be done next. This allows them to focus on learning how to apply the operations properly to solve the matrix.

Introduction

Algebra students solving simultaneous linear equation systems with Gaussian elimination often experience problems. They often forget to apply the multiplication to every term in a row or add every column in a pair of rows. This can lead to confidence problems that plague them the rest of the semester. Therefore, a Java applet was designed to aid students with Gaussian Elimination. There are many such systems that solve these kinds of problems so this was not the intent. Rather give them an environment where they can focus on learning how to use the row operations to solve the problem. It does the various kinds of row operations, but at their direction. It also reduces the tedious portion of calculating numbers and thus eliminating many errors that would be made. This applet was attached to the class web page so that it would be available anytime a student would choose to study. Valley City State University is a laptop institution, so students live in wired dormitories and always have access to the web page. The next section will give a detailed view of how a student might use this applet.

Solving a Problem

Initially a student would navigate to the class web page. The URL is: http://community.vcsu.edu/facultypages/curt.hill/My_Webpage/m102home.htm The class has moved beyond this topic, however, the applet still exists on the web page. Figure 1 shows the initial display of the web page. (Instead of showing how the applet would show in the page, we have used screen shots using Applet Viewer.)

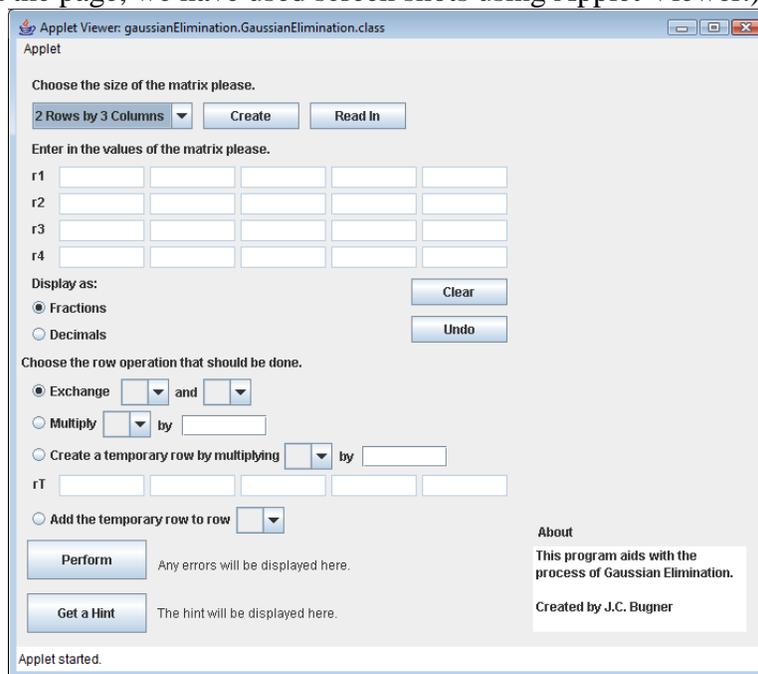


Figure 1: Starting Screen

Once a student arrives at the web page, he or she must enter the problem that needs to be solved. This is started by using the drop down box to select the size of the problem and

then pressing the Create button. When this is done, the values are typed into the cells of the grid. The cells that may be typed into are highlighted in the grid. The highlighting may be seen in Figure 2 as an outline on the grid cells. This gives the student a visual indication of where to type. The student may click on the cell and enter a value. A tab will take the student to the next cell. Figure 3 displays the applet after the Read In button has been clicked. Notice that the outlining around the cells has disappeared. This indicates that the problem solution process may begin.

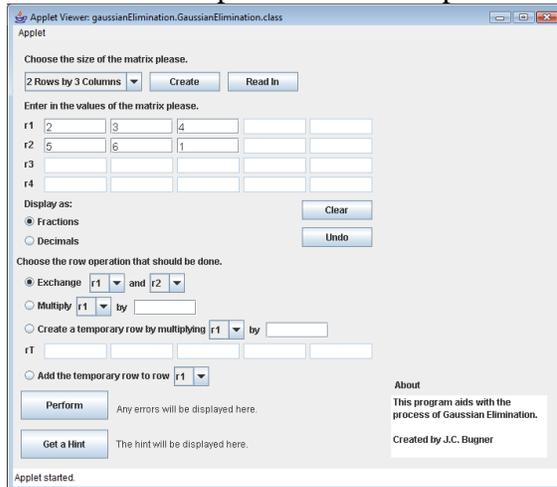


Figure 2: Entering numbers

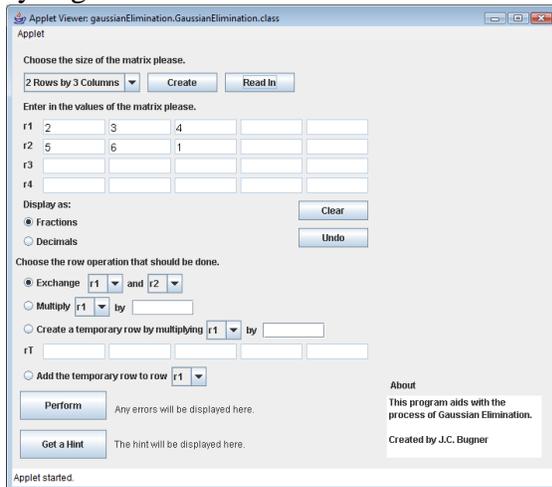


Figure 3: Read In button has been clicked

Notice in the previous figures that the Perform button near the bottom of the panel and the radio buttons above it. The problem will be solved by cycling through the following process:

1. Select the operation identified by the radio button.
2. Fill in the extra parameters required for that operation.
3. Click the Perform button.

In addition, a hint may be obtained when the student is not sure how to proceed. The data is almost always entered in decimal format. Algebra problems typically use integers for all the coefficients. However, in the process of solving the problem it is common to have either decimals or fractions appear. Most calculators prefer a decimal representation, while students without a calculator generally use fractions. There is another radio button that chooses how to display these intermediate results. In the example displayed in the figures, the first row should be divided by two to force a one into the cell in the first row and first column. This is done by setting the appropriate values in the Multiply radio button line. The setup is shown in Figure 4 and the results of clicking the Perform button in Figure 5.

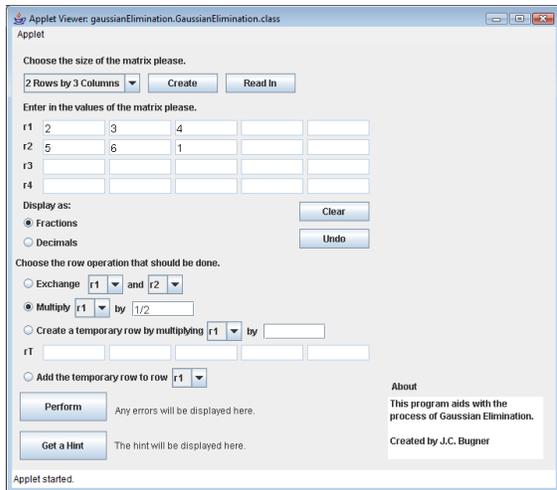


Figure 4: Multiply row set

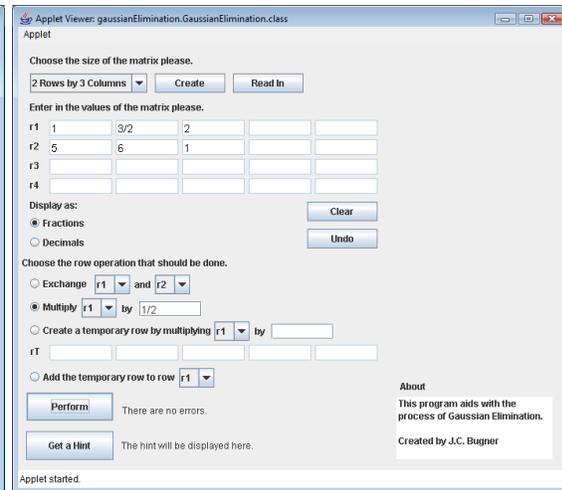


Figure 5: After hitting Perform

Although the next step would be to attempt to zero the rest of the column, the student might not know this. Therefore, a Hint button is given. Figure 6 shows what the matrix looks like when one hits the Get a Hint button to get a tip on how to solve the matrix.

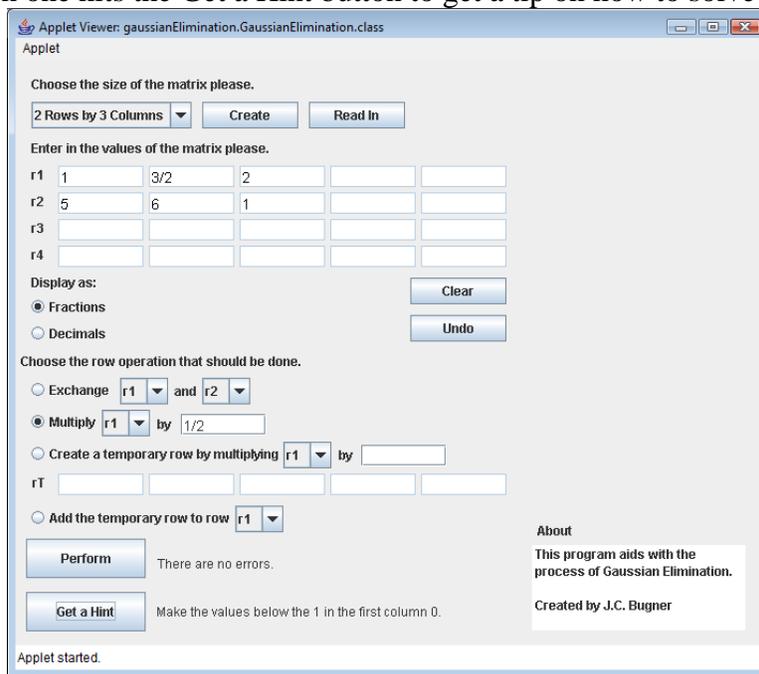


Figure 6: After hitting Get a Hint

The hint tells them to make the value that now has 5 into 0. The process for this is to multiply the first row by -5 and then to add that to the second row. This is a bit complicated so the applet breaks it into two steps. The first is to create a temporary row by multiplying the first row by -5. The second step is to then add the temporary row to the second row. Figure 7 shows choosing the appropriate radio button and entering in -5 for the value to multiply the row by. Figure 8 shows the result.

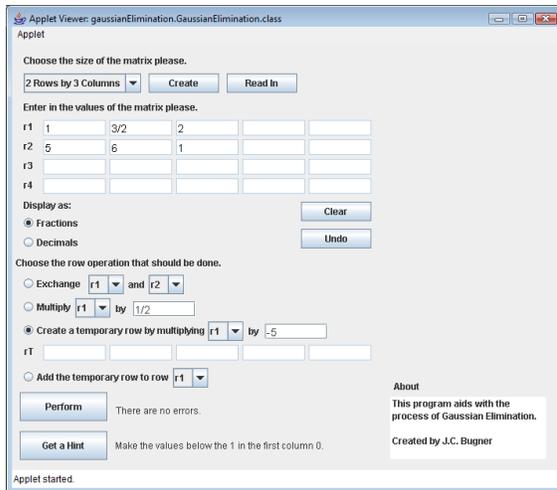


Figure 7: About to create temporary

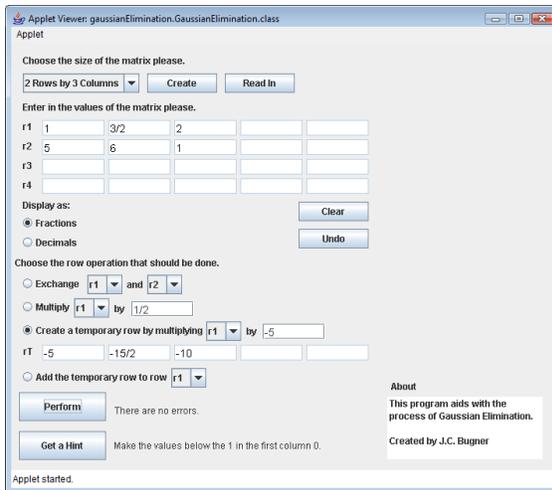


Figure 8: The temporary is created

Figure 8: After creating the temporary row

The second step is to add the temporary row into the second row. Now we will add the temporary row to the second row by choosing the add operation, change the row added to row two, and then hitting the Perform button. Figure 9 shows the matrix with the correct row operation radio button and the correct row chosen.

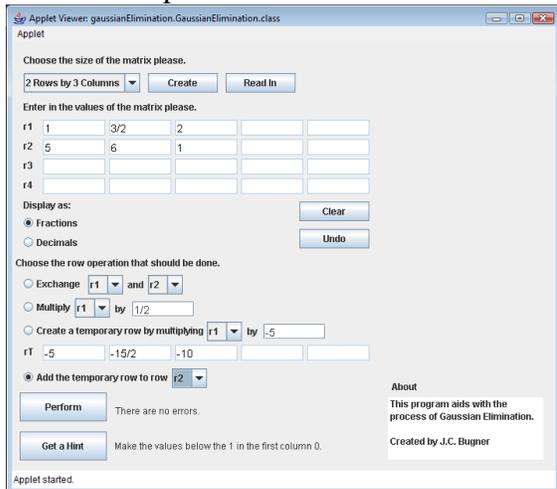


Figure 9: Preparing to add

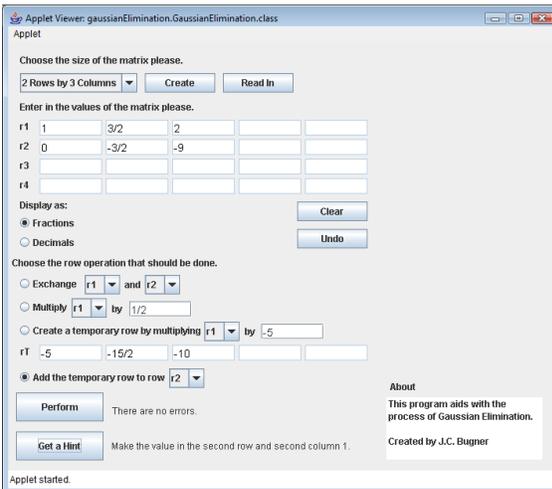


Figure 10: After the add

The process continues in a similar way. The hints that are given vary by the state of the matrix. Figure 10 also shows a subsequent hint that has been given.

Implementation Challenges

An interesting part of implementing this program was the process of creating a fraction class. Although it seems likely that many similar fraction classes have been implemented in Java, none were found that included the source code. Therefore, a limited fraction class

was designed. It was limited in the sense that it did not need every method that a general class might require. For instance, it needed equal and not equal comparisons, but not less than or greater than. It was also interesting in that it required work by the first author that was different than had been seen in previous programming classes.

Designing the graphical user interface was also challenging. Drop down boxes needed intuitive values so only active rows would be available to be chosen. The "Get a Hint" button was especially long to program because of so much checking. The Undo button was interesting to program, since it allows the user to undo their last move. A common error is to click a radio button that selects an operation but forget to change the values prior to clicking the Perform button.

Final Thoughts

This applet did not see much use this year as would be desired. This was due to the small size of the classes it accommodated. Another problem was that the class in question move past the topic after several useful enhancements had been developed. These include several bug fixes as well as the Undo button. However, it seems very likely it will prove useful in the following years because of its easy-to-use interface and its internet availability.