Final Project Report

Jack Ehlert and Ben Ebel

Me and Ben's original idea for this project was to include our major in some aspect for our idea and building of this code. As chemical engineers, a lot of the information we learned this year, and what will be doing in the future is based on equations and derivations. Therefore, since we were building an algorithm for this project that would be able to take in a certain set of inputs and provide a fully accurate output/solution. The equation that we based our program on was called the Van der Waals equation. Quinn looked at and approved of our idea to use this equation. The van der Waals equation calculates a certain pressure in a container/body based on critical temperature of a gas, temperature of the body, specific volume of the gas, and critical pressure of the gas. Also, we calculated the a and b values through callback functions which are values needed to find the final pressure. These a and b values are also calculated based off of the 4 given inputs. Therefore, our inputs in this program were the 4 aspects that the final pressure was based off of. We thought that this was relatively achievable since we were working off a known equation which means that as long as we perfected the base functions then we wouldn't have to worry about not getting the right output every time.

We started out by creating a main function called finalproject(). This main function's main purpose was controlling all aspects of the GUI including labels, layout, and UI control elements. We all defined a global variable GUI in this main function. By making this variable global we would be able to utilize it in our future functions and be able to update it progressively. This function would solely be responsible for our user interface and none of the actual equations and calculations that occur with our actual inputs and equation. The three GUI styles that were produced from this code and appear in our actual UI are editable text boxes, static text labels, and push buttons. The static text label displays the actual known Van der Waals equation that this program is based off of. The 4 editable text boxes are the inputs for this program. The user inputs numbers in each one of these editable textboxes to obtain a calculated final pressure. The

4 textboes(inputs) are the critical pressure, specific volume , temperature of the body, and the critical temperature. The 3 push buttons are the calculated outputs and they include the a value, b value, and the calculated pressure.

Our second function was called calculatePressure(). This was the function where we would take in the user inputs from the GUI and use them to calculate the given outputs and the values that went along with them. There are multiple other nested functions inside of this calculatedPressure() function though. Our first nested function is called calculateTermOne(). This function calculated the entire first part of the VDW equation (RT/(V-b)). We had another nested function inside of this function called calculateBVariable that would find the b value based off of the given inputs which would then update the b variable and allow us to get a number for the term one function. We then had another function called calculateTermTwo() which was responsible for calculating the second part of the equation (A/V^2). Finally we had two functions that would calculate the a and b values and display them if the push box on the GUI was clicked.

Throughout every single one of these functions we updated our variables based off of the numbers input in the actual GUI. A difficulty we did run into originally is that we got the output to be not a number whenever we tried to produce an output with the updated variables. This is because we had to type .String at the end of each one of our GUI commands inside of the nested functions. This is because with the str2double function it takes a string and then converts to a double precision value. Therefore this meant that we had to have the input into the GUI to be read as the string of the uicontrol before the str2double command, which is why the .string was important. After we got this down, we were able to produce numerical outputs in our calculation functions. The next problem we ran into had to do with the message box. The main problem with the message box was trying to get the box to project the calculated number along with the string that told the user which number it was that was calculated. The solution to this problem was using the same format for the message box as lab 10 but adding in the strcat function to put two strings together. This was done by taking the number calculated in the functions and then turning it back into a string with num2str. The double quotes in strcat made it so that there was a space in between the equals sign and the number. This was only because not having a space bothered us because it looked bad. However, using these two techniques, we were able to display the messages we wanted with no problems in the final solution.

Though me and Ben are both chemical engineers, there are definitely ways that we may be able to implement GUI's in the future. If we participate in research projects where we collect data and have to present it to multiple people, we could create a program that portrays our data in a visual way that is very easy to comprehend to people. We could achieve this using the multiple different UI styles that are available in ways that are most suitable to the results/data that we are trying to portray. All in all, considering the power and functionality of matlab and its UI control, there is a lot of room for usage and functionality in the future as me and Ben continue on our chemical engineering careers.