Jarod Harris and Wade Slepicka

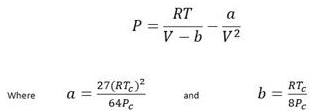
CSCE 155N

Professor Quinn Lanik

5/1/2021

Design Document for the Van der Waal Equation Solver

Approved by Laurel Hilger

 The Van der Waal equation is a very common but tedious equation that can be found throughout the entirety of the freshman year Chemical Engineering curriculum. Figure 1 shows the notation for the equation, as well as the calculation for gas constants ‘a’ and ‘b.’

**Figure 1:**

During the short time we have spent as Chemical Engineering majors, this equation has been a thorn in our side, often requiring more than a page of scratch paper for all of the conversions and calculations. From our pain, the seed was planted to code a calculator to do all of the work for us.

The project was a lot more involved than we initially predicted. Our first issue arose when we came to the realization that while we may be able to code one way to solve the equation, our professor tends to ask the same question multiple ways, and only being able to solve for one variable would be useless in that context. Thus, the calculator has two branches, one solving for pressure and the other solving for temperature. The second issue came from the fact that ‘a’ and ‘b’ are not explicitly defined in the literature our Chemical Engineering professor dictates we use. Therefore, edit boxes needed to be used to calculate ‘a’ and ‘b’ from the critical temperature and critical pressure, which is the most typical metric listed in the literature. After drafting the end result by hand, the coding process began, and for the most part we ran into only a few issues.

Initially, the code used a dropdown menu to select the appropriate gas, and then set the value of the criticalTemperature and criticalPressure variables to the preset value of each gas. However, once again, this process was too narrow to be able to solve the equations that our professor would throw at us. There were too many gasses listed in literature to include every value in our switch statements, and beyond that, our professor also would use gases like Gas Z or Gas Q with arbitrary critical temperature and pressure values. With all of this stacked against us, we swapped to edit boxes to allow any value to be entered.

The second issue was that the code would crash if invalid values were put into each edit box, as math cannot be done with strings. We had to read into the str2num function a bit in order to find out that str2num does not convert strings to ASCII values like we first thought. A series of if statements was implemented to check for valid values, and the code stop crashing due to incompatible values.

Another issue arose when we attempted to insert an image of the Van der Waal equation into our figure. Basically, we discovered that MATLAB images are very funky, messy, and cannot be resized easily. It is one of the first pitfalls of the language that we have found throughout this class.

There were also countless times that variables were not displaying correctly or being misassigned. Many of those errors were solved in one of two ways: either checking that our variable names matched and were case-sensitive, or the difference between %f and %d in our fprintf statements made enough of a difference to completely remove our displayed values.

Overall, this project was eye-opening to the potential that our coding knowledge has. While in class, these kinds of problems were a thorn in our side, constantly wasting hours of our time for basic calculation. It is certain that we will run into even more tedious equations in the future of our major, and it will be beneficial to be able to code user-friendly GUIs in order to make these kinds of problems cake walks in the future.