MA348 Numerical Analysis, Systems of Equations and Matrices

David Jefts

February 18, 2019

Introduction

The goal of this lab is to use the Gaussian Matrix Elimination algorithm to solve for the values in a System of Equations. For the purposes of this lab, the System of Equations was:

$$\begin{cases} x - y + 3z = -3 \\ -x \qquad - 2z = 1 \\ 2x + 2y + 4z = 0 \end{cases}$$

This was translated into the matrix below to be used by the Gaussian Elimination software.

$$\left[\begin{array}{ccc|c} 1 & -1 & 3 & -3 \\ -1 & 0 & -2 & 1 \\ 2 & 2 & 4 & 0 \end{array}\right]$$

Theory-Analysis

Gaussian Elimination has two major steps, with two optional steps that are discussed at the end of this section. Forward Elimination, which solves the bottom half of the matrix, followed by Backwards Substitution, which solves for the variable values.

By the end of the Forward Elimination step, the bottom left-hand corner of the matrix (below the diagonal) should be all equivalent to 0, as highlighted below.

Numerical Solution

This lab was solved using Fortran code to estimate the roots of the function and gnuplot to plot and tabulate the values. Multiple different methods were used, which as a group are colloquially called "open methods." This name comes from the fact that all of the methods work on an "open" set of numbers as opposed to the "bracketing" methods that work only on a set of numbers within a limited range.

Newton's Method (also known as the Newton-Raphson method is an approximation algorithm that uses the tangent line of a function to estimate the root. Starting with a given value, it iteratively uses the function $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ until a desired accuracy is reached. This method converges quadratically, making it a very desirable method for quickly calculating the root of a function and it is relatively easy to extend this method to high-order equations and functions. The drawbacks to this method are that the derivative of $f(x)$ has to be recalculated each iteration and the method will completely fail if $f'(x) = 0$ at any point during the iteration process.

The Secant Method is an approximation algorithm that uses the secant line of a function to estimate the root. Starting with a given value, it iteratively uses the function $x_{n+1} = x_n - \frac{f(x)\times(x_n - x_{n-1})}{f(x_n)-f(x_{n-1})}$ until a desired accuracy is reached. The Secant Method converges slightly slower than Newton's Method, on the order of $\approx$1.618, the Golden Ratio. This method can be implemented easier than the Newton Method since the derivative of $f(x)$ does not need to be calculated, however this method requires two initial points to start and it is not always guaranteed to converge.

The Modified Secant Method is another approximation algorithm that uses the secant line of a function to estimate the root. This method is a modification of The Secant Method detailed above, taking advantage of Newton's difference quotient function to estimate the derivative. It iteratively uses the function $x_{n+1} = x_n - \frac{f(x_n)\times\delta)}{f(x_n+\delta)-f(x_n)}$ until a desired accuracy is reached. The Modified Secant Method converges at the same rate as The Secant Method, while removing the need to define and track two initial points. However it requires a definition of delta (δ) with constraints- if defined too large it may skip over the root, too small and it may never converge in a reasonable amount of time. For this purposes of this lab $\delta = 0.01$.

The Modified Newton's Method for Roots of Multiplicity is an approximation algorithm that uses the tangent line of a function and its second derivative to estimate the root. This method iteratively uses the function $x_{n+1} = x_n - \frac{f(x_n)\times f'(x_n)}{(f'(x_n))^2 - f(x)\times f''(x_n)}$ until a desired accuracy is reached. Like Newton's Method, this method converges quadratically. Additionally, this method works even if there is a Root of Multiplicity (i.e. the function has two roots of the same value such as $f(x) = x^2 - 9$. However calculating the second derivative of a function is not always easy.

For the comparison test using $P = 10$atm and $T = 100$K and starting numbers of $x_{n+1} = 3, x_n = 4$, and $x_{n-1} = 5$:

Newton's Method converged in 4 iterations
The Secant Method converged in 6 iterations
The Modified Secant Method converged in 5 iterations
The Modified Newton's Method for Roots of Multiplicity converged in 6 iterations

Results and Discussion

Firgure **??** in Appendix A is a graph of the error during each iteration of each algorithm, where each line represents one of the above-mentioned estimation methods, the x-axis is the number of iterations and the y-axis is the error. In this graph scenario $P = 10atm$ and $T = 300K$. This graph seems to indicate that Newton's Method both starts with a smaller error and converges towards the desired value faster. The other methods all converged upon the same value, but were slower, less efficient, and/or unnecessarily complicated (as in the case of the Modified Newton's Method for Roots of Multiplicity since this function does not have any roots with multiplicity). When compared with the Ideal Gas Law however, they all reached a Modal Volume about $0.2\frac{m^3}{mol}$. The cause of this discrepancy is most likely due to the corrections the van der Waal's equation makes to the Ideal Gas Law formula by observing and allowing for the fact that gas molecules attract each other (noted by the variable a in his equation) and the volume of each individual mole of gas (noted by the variable b in his equation).

Conclusions

Newton's Method appears to be the most efficient method without sacrificing any error or simplicity, in the future that may not always be the case though, depending on the function, how it curves, and what its roots are. If this were to be done by hand the most accurate and efficient way would be Newton's Method, however the other methods do not converge at a significantly slower rate when using a tolerance of 10^{-6}. There are many other ways to optimize the estimation this function however, the methods of finding roots detailed in this report are just a handful from a plethora of various solvers and algorithms that accomplish the same task.

Appendix A

Appendix B

Appendix C