Design of a Contoured Nozzle through Two-Dimensional Method of Characteristics with Chemical Reactions

A Project

Presented to

The Faculty of the Department of Aerospace Engineering

San Jose State University

In partial fulfilment of the requirements for the degree

Master of Science in Aerospace Engineering

By

Michael Sandoval

approved by

Dr. Fabrizio Vergine

Faculty Advisor

SJSU SAN JOSÉ STATE UNIVERSITY

Table of Contents

List of Figures

List of Tables

Nomenclature

a_f	=	Vibrationally frozen speed of sound
$[C]$, $[\dot{C}]$	=	Concentration, concentration with respect to time
ΔC	=	Distance along characteristic curve
$C^{+/-}$	=	Positive or negative characteristic line
x	=	Horizontal cartesian coordinate
y	=	Vertical cartesian coordinate
η	=	Unit vector normal to streamline
ξ	=	Unit vector parallel to streamline
θ	=	Inclination angle
μ	=	Mach angle
ν'	=	Reactant stoichiometric constant
ν''	=	Product stoichiometric constant
i	=	Species index
h	=	Enthalpy
$Y, \dot{Y}$	=	Mass fraction, mass fraction with respect to time
X, $\dot{X}$	=	Mole fraction, mole fraction with respect to time
$\mathcal{M}$	=	Molecular weight
k_f	=	Forward reaction rate
k_b	=	Rearward reaction rate
P	=	Pressure
ρ	=	Density
CEA	=	Chemical Equilibrium Application

1. Motivation

1.1 Motivation

Research and development on hypersonic vehicles have been increasing over the last few years. Namely, hypersonic missiles with nuclear capabilities have all the major superpowers in the world racing to develop arsenals of these weapons. These missiles achieve hypersonic speeds with scramjet or ramjet engines. Access to space is also another motivation to develop hypersonic airbreathing engines. For example, a paper by Martin [1] analyzed concepts for single-stage-to-orbit vehicles where a ramjet engine with various configurations was analyzed and compared to traditional rocket engines. The paper found that some configurations of ramjets can carry payloads comparable to medium sized liquid rocket engines,

The focus of this paper is the convergent-divergent nozzle design, but it is worth understanding the functionality of a ramjet and scramjet. **Figure *1*** shows as schematic [2] of a ramjet engine. At the inlet of the ramjet, the flow decelerates and compresses through a series of oblique shockwaves in a diffuser. At the end of the diffuser, a normal shock is formed, and supersonic flow becomes subsonic flow as it passes through a divergent duct into the combustion chamber. Within the combustion chamber, fuel is injected, mixed, and burned with the composition of air. Finally, the flow is accelerated again through a convergent-divergent nozzle and typically reaches Mach 3-6. Scramjets, however, are effective above Mach 6. At these Mach numbers it is not desirable to decelerate the flow like a ramjet, and it uses supersonic combustion, this is where the name "scramjet" derives from. The vehicle body itself is used for the scramjet operation, with its supersonic combustion process, the entire engine geometry is convergent up to the exhaust nozzle as shown in **Figure *2***.

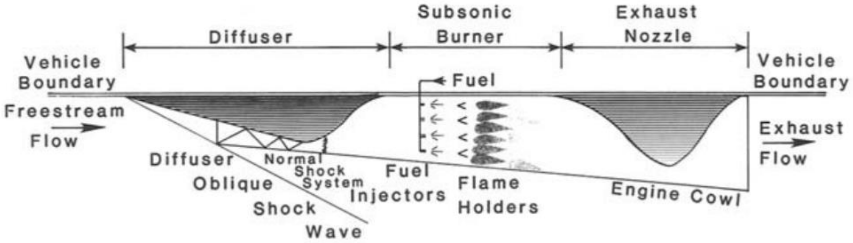


Figure 1: Ramjet engine schematic [2].

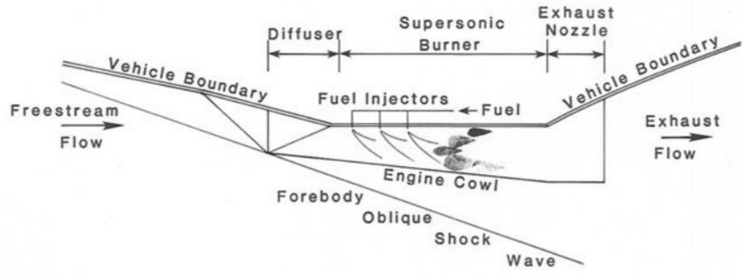


Figure 2: Scramjet engine schematic [2].

The main challenges hypersonic airbreathing engines face relate to stable combustion processes, structural integrity for reusability, and development of analytical tools to predict engine behavior. This paper will focus on nozzle wall development for supersonic and hypersonic flow. The nozzle contour is critical for optimum thrust operation, and this shape can be determined analytically with the method of characteristics in calorically perfect gas (CPG), thermally perfect gas (TPG), thermo-chemical equilibrium (TCE), and non-equilibrium. Below is a definition of each of these states of flow:

- Calorically perfect gas (CPG): the specific heat capacity is constant and is not a function of temperature.
- Thermally perfect gas (TPG): the specific heat capacity is a function of temperature only.
- Thermo-chemical equilibrium (TCE): the state variables are a function of temperature and pressure. It is a multispecies, chemically reacting gas with no intermolecular forces. Each species obeys the perfect gas law. Recombination and dissociation rates are equal.
- Non-equilibrium: Gas composition changes as a function of time, requires reaction mechanisms to solve reaction rates. The state variables are a function of temperature and moles of all species. Each species still obeys perfect gas law. Recombination and dissociation rates are not equal.

There is a well-defined way for determining the nozzle contour with MoC in CPG (discussed in section 1.5) however in TCE or non-equilibrium, it becomes difficult to model and is dependent on available thermodynamic data for the species being considered. In CPG, the nozzle is assumed to be steady, inviscid, isentropic, and adiabatic. This allows the assumption that total enthalpy is conserved and constant throughout the entire nozzle. Looking at the energy equation below, we can see that the exit velocity is proportional to the square root of the difference of enthalpies. This equation is derived in the Fundamentals of Aerodynamics textbook [3]. In the combustion chamber (CC) we will have near zero velocity (assumed to be zero), and the equation is rearranged in terms of exit velocity of the nozzle shown below. Clearly, a large difference in enthalpies will result in higher exit velocity, the goal is to maximize this difference.

$$h_{CC} + \frac{V_{CC}^2}{2} = h_{exit} + \frac{V_{exit}^2}{2} \Longrightarrow V_{exit} = \sqrt{2(h_{CC} - h_{exit})} \tag{1}$$

As explained in the Hypersonics and High-Temperature Gas Dynamics textbook, chapter 14 [4], the temperature of the gas is higher than CPG due to recombination of gas molecules in the divergent section of the nozzle. The recombination effect turns chemical energy into translational energy, which means the static temperature is higher than in the CPG case. **Figure *3*** below compares temperature axially along the nozzle for equilibrium and CPG assumptions. This figure also illustrates the importance of analyzing nozzle flows in TCE and non-equilibrium due to the large difference in static temperatures from CPG to TCE.

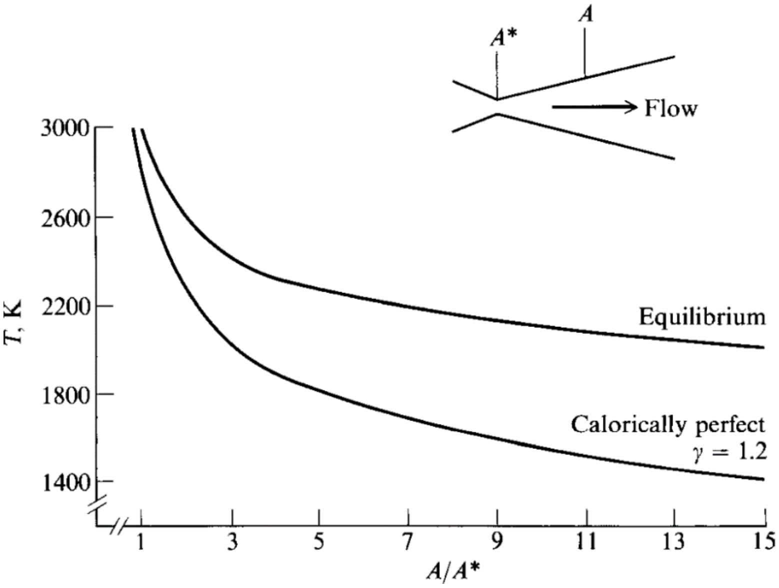


Figure 3: Temperature distribution in CPG and equilibrium along divergent section of a C-D nozzle [4].

A central issue with hypersonic nozzles is the species recombination and disassociation reaction rates. Long nozzles with large expansion ratios can be assumed to be steady, isentropic, quasi-1D flow, with CPG assumptions. Whereas for short nozzles, the nozzle length might not be sufficient to reach equilibrium. Shorter nozzle lengths mean that the molecular species can eject so fast out of the engine that recombination may not happen fast enough. This means the recombination and dissociation rates are not equal; hence the flow is in non-equilibrium. Reaction rates application to MoC is explained more in detail in Chapter 2, and the derivation is shown in Appendix A.

Thrust, boundary layer, and active cooling systems are all affected by the nozzle wall, species make-up of the flow, and the reaction rates of those species. Utilizing MoC, with CPG assumptions, to generate a nozzle geometry is a good first approximation that can be further analyzed in a CFD solver. The same idea can be applied to nozzle contours generated with MoC in TCE or non-equilibrium which is desirable for ramjet or scramjet nozzle designs. A first approximation nozzle contour can reduce the computation time for CFD solvers and include more in-depth analysis of more complex flow problems such as turbulence and boundary layer. Currently, there is a lack of an application that creates a nozzle contour in TCE or non-equilibrium for airbreathing engines, and this is the purpose of this project. Combining tools such as MATLAB and Chemical Equilibrium Application (CEA), this project will utilize MATLAB's calling functionality to extract important thermodynamic properties from CEA that in turn will calculate the MoC compatibility equations (derived in Appendix A) using an finite difference method approach. The compatibility equations solve the characteristic mesh which dictates the nozzle contour. Development of a tool to quickly design two-dimensional nozzles in TCE or non-equilibrium is helpful for scramjet and ramjet geometries. For example, the Lockheed X-7, shown in ***Figure 4***, was a hypersonic ramjet missile with an axisymmetric engine design. According to [2] designs with an axisymmetric nozzle required struts to keep it secure in place, and it did not make efficient use of the airframe. It was found that frames holding the engines causes significant amount of drag and wall effects which canceled out the uninstalled thrust. Now, most ramjet and scramjet vehicles have a two-dimensional engine design. This is because the body of the aircraft is a useful compression surface for the inlet of the engine, and the aft of the airframe can be used as a nozzle wall expansion surface shown below in ***Figure 5***.

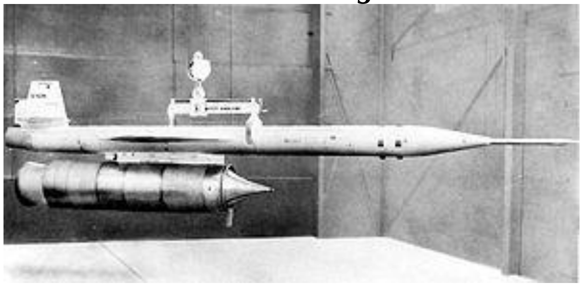


Figure 4: Lockheed X-7, *Source:* *https://en.wikipedia.org/wiki/Lockheed_X-7*.

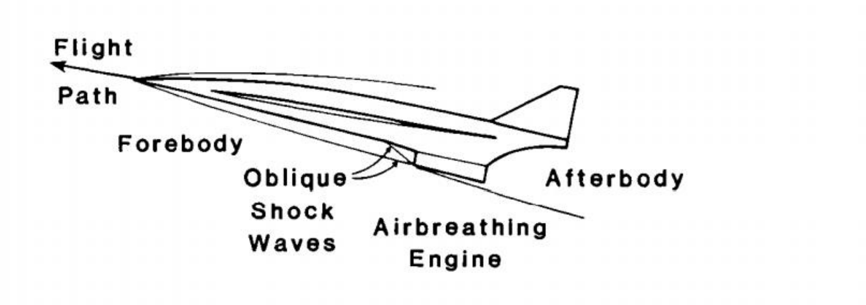


Figure 5: Hypersonic aircraft airflow schematic [2].

1.2 Literature Review

1.2.1 Research Articles and Technical Reports

The technical report, A General Method for Automatic Computation of Equilibrium Compositions and Theoretical Rocket Performance of Propellants [5], explains how their computer program calculates important thermodynamic variables for a rocket engine in chemical equilibrium. This report is vital for implementing mathematical techniques for chemical reacting flow into a MATLAB code. The program includes calculating a nozzle in thermo-chemical equilibrium from either a constant pressure and an adiabatic combustion problem or an isentropic expansion problem. The program carries out the calculations with equations relating to total enthalpy, entropy, Dalton’s law, and concentration of atomic species while assuming that the flow is in local thermo-chemical equilibrium. The Newton-Raphson method is an iteration technique to solve the nonlinear equations that define the flow field. The iteration technique is well explained and is referenced in the main technical report [6] used for this project. The report also shows how to calculate equivalency ratios, what molecular species to consider in the calculations, and calculate it in the code routine. Overall, the report gives more insight into chemical reactions, specifically for nozzles, that the Hypersonic and High-Temperature Gas Dynamics book [4] does not discuss.

The technical report, Applying the Method of Characteristics to Analyze the Flow Field of a Chemically Reacting Gas in a Two-Dimensional or an Axisymmetric Nozzle [6] outlines the derivation of equations of 2D and axisymmetric MoC with reacting flows. The author created a computer program for one reaction system, N4O2 + N2, and another for H2 + Air reaction mechanism broken down into fourteen elementary reactions. The computer program calculates the species reaction rates and the characteristic net for optimal rocket nozzle wall geometry in local thermo-chemical equilibrium or non-equilibrium. The data calculated consisted of pressure, temperature, mass concentration, and mole fraction along the two-dimensional rocket nozzle's symmetry line. The data were compared against experimental results with the same conditions, demonstrating that the computer program closely matched the experimental results. When the article was published, thermodynamic data required to calculate the reaction rates, such as reward reaction rate, species sensible enthalpy, and entropy, was not widely known for a wide range of combinations of fuels and oxidizers.

The previous article [6] presents the equations needed to solve the compatibility equation for the MoC calculations. Those equations used for 2D and axisymmetric MoC with reacting flow in nozzles apply to blunt bodies. The paper, Inviscid Flow of a Reacting Mixture of Gases Around a Blunt Body [7], explains these equations in more detail, such as simplifying the Navier-stokes equations inviscid reacting flow and the effects recombination rates and dissociation rates have on the flow properties. The paper also presents non-dimensional equations of the freestream properties that define the non-equilibrium flow with reasonable accuracy. Overall, the paper clarifies techniques to tackle local equilibrium and non-equilibrium flows. It also highlights the effects of recombination rates and dissociation rates effects on overall nozzle performance.

The technical note, Performance of Several Method-of-Characteristics Exhaust Nozzles [8], analyzes multiple MoC nozzles, and a fixed conical nozzle at various pressures and expansion ratios. The results showed that the MoC method could increase the thrust by 1%, and the paper demonstrates it can approximately double the payload weight of a missile. This paper explains how the coefficient of moment of an engine as a design parameter for MoC nozzles. The ratio, coefficient of thrust over the ideal coefficient of thrust, is a function of the coefficient of moment. The paper correlates the coefficient of moment with the pressure ratio along the nozzle horizontal axis with isentropic assumptions. There is uniform axial flow exiting the nozzle with these assumptions, and the pressure and temperature ratios are a function of Mach number. From the summary of results, the author states that the MoC is not sufficient for low-pressure ratios within the nozzle will, instead of the fixed divergent angle nozzle design will suffice. Low-pressure ratio nozzle designs may benefit from fixed divergent angle nozzle because an MoC contour would create high separation pressures, causing structural damage and a severe decrease in thrust.

The technical report, Recombination of Hydrogen-Air Combustion Products in an Exhaust Nozzle [9], can be a benchmark for data generated by the MATLAB code. The report analyzes a ramjet with a hydrogen-air system with a combustion temperature of 1833 Kelvin and a stoichiometric system. The report offers finite difference method techniques for solving reaction rates only in equilibrium and locating the freezing point along the nozzle axis. Furthermore, the report extensively discusses a ramjet nozzle experimental setup in an arc-jet facility and calibrations needed for pyrometers.

Supersonic Nozzle Flow with a Reacting Gas Mixture [10] analyzes the supersonic nozzle flow of nitrogen tetroxide and dioxide in chemical equilibrium and frozen flow. The authors approach the problem using pressure, area, and the defined species' reaction mechanisms to define the flow field. Craig [6] references this paper regarding the frozen speed of sound and the effect of vibrational energy on the flow. Enthalpies of the species are datasets required to calculate the reaction rates. With this data and known nozzle geometry, the paper gives three methods that fully define the flow: frozen flow, chemical equilibrium flow, and flow with known static pressure.

Design Optimization of Rocket Nozzles in Chemically Reacting Flows [11] provides another look into a method of calculating reacting flows within a rocket nozzle. The mathematical techniques used are Newton's method and the finite rate chemistry model. The author states that the code can reduce computational time for finding the correct nozzle shape for a CFD simulation, thus reducing design time and cost. Another exciting benefit of optimizing a nozzle with reactions is that the species composition data can minimize exhaust plumes' infrared detection. The optimization code calculation uses an implicit Newton's solution method (analytical) in tandem with the Jacobian matrix for the flow analysis (numerical). The author compares the two datasets based on performance and design optimization. The reacting flow model presented in the article closely resembles the equation set up in Craig's technical report [6]; however, it explains the reaction calculation in more detail and includes reaction mechanisms with third body efficiencies. The numerical method presented does not utilize thermodynamic species data to calculate the given species' reaction rates. Instead, the author uses empirical equations to solve the reactions, such as the Arrhenius equation, and a fourth-order polynomial to solve the specific heat constant, entropy, and enthalpy for all species present in the flow. The authors' results are analyzed along with three different nozzle geometries: conical nozzle, optimized nozzle, and the Taurus-II nozzle.

Scramjets are also an important study for nozzle designs. These airbreathing engines require supersonic flow to operate as it has no moving parts, and the geometry of the inlet is specific for a small range of Mach numbers. Oblique shockwaves form in the inlet, thus increasing the pressure and slowing down the flow. The flow is moving so fast that the only feasible geometry requires supersonic combustion. The paper, Development and Verification of a Supersonic Nozzle with a Rectangular Cross Section at a Mach Number of 2.8 for a Scramjet Model Combustor [12], explains a numerical method that calculates the geometry of a supersonic two-dimensional nozzle wall with boundary layer corrections. The method used to design the nozzle contour was Foelsch's method, which calculated the geometry needed to cease the expansion waves forming in the nozzle, like MoC. The article outlines how to approximate the boundary layer thickness by assuming a straight line connects the throat area and exit area. The authors conclude that the methodology used is easier to apply than others and still provides an optimized contour with boundary layer thickness.

Non-equilibrium is an essential study for ramjet and scramjet nozzles. Mach numbers in the exit section are enormous, that the species stay time is very small. If the molecules do not have enough time to react with other molecules within the nozzle, the reactions rates between recombination and dissociation are not equal; therefore, the flow is non-equilibrium. The article, Numerical Analysis of Thermo-chemical Non-equilibrium Flows in a Model Scramjet Engine [13], explains how the Navier-Stokes equations with non-equilibrium implements into an open-source solver of the two-temperature model. The two-temperature kinetic model is discussed in chapter 13.4.1 of the Hypersonic and High-Temperature Gas Dynamics textbook [4]. The solver, hy2Foam, due to its high fidelity of solving non-equilibrium flows. The article can help understand what types of CFD results need analysis for a non-equilibrium or equilibrium flow and if it is feasible to solve non-equilibrium flows for this project.

Thermal protection systems (TPS) are just as critical as the engine because hypersonic vehicles produce a large amount of heat in the combustion chamber. These extreme temperatures test today's technological limits and are an essential topic of study for ramjets and scramjet engines. The Research Progress on Active Thermal Protection for Hypersonic Vehicles [14] article's primary purpose is to overview all the current active cooling methods for hypersonic vehicles. The author states that active cooling such as regenerative, transpiration, or film cooling is necessary for high heat loads present in these engines. The article provides insight into each cooling method's main issues, heat transfer and flow mechanisms for calculations, and current developments. The results and analysis section compares 3D simulations results of a model scramjet with each active cooling method type.

The Thermo-chemical Non-Equilibrium Scale Effect of The High Enthalpy Nozzle [15] paper explains the importance of accurately calculated nozzle geometries with high enthalpy nozzle flows. Much like the previous articles mention, this article also applies Navier-Stokes equations with the non-equilibrium flow to a numerical model. Vibrational energy is calculated using methods explained in chapter 13 of the Hypersonic book [4]. It can apply with other theories such as Gupta-Yos viscosity, Wilkes Law, and a "translational-vibrational energy relaxation time (TVERT)" mathematical model, which fully defines the flow field. This numerical method is combined with a CFD solver to solve for unsteady flow. The article goes into depth with frozen vibrational energy and the freezing point's scaling effects in a high enthalpy nozzle. The article also compares the expansion ratio with the mass fractions of species for three different mechanisms. Each mechanism increases in complexity with more species; the models presented were 5, 7, and 11-species for air.

1.2.2 Textbooks and Dataset References

The NASA technical paper, Thermodynamic Data for Fifty Reference Elements [16], contains tabulated data for 50 elements, including specific heat at constant pressure, sensible enthalpies, and sensible entropies. These variables are a function of temperature in a range of 200 Kelvin to 6000 Kelvin in 100 Kelvin intervals and are crucial in calculating the reaction rates. The Hypersonic and High-Temperature Gas Dynamics textbook [4], specifically chapters 13, 14, and

15, derives and explains equilibrium flow, non-equilibrium flow, and MoC in calorically perfect gases.

1.3 Project Proposal

Hypersonic vehicles such as ramjets and scramjets are the new frontier of airbreathing propulsion. With an increased need to develop and analyze these high enthalpy engines, programs need further development to provide important thermodynamic data and a useful nozzle geometry for a user-defined problem. This data is incredibly valuable for analyzing a vehicle's thermal protection system, boundary layer formation, and separation. Due to the extreme conditions the nozzle experiences at hypersonic speeds, chemical reactions play an important role in performance losses. The ability to quickly iterate nozzle designs and produce data that closely match the physical flow is required for rapid prototype development and cost savings.

The objective of this project is to design a tool that can create two-dimensional nozzle wall contours in CPG, TCE, and non-equilibrium for hypersonic airbreathing engine applications. This project will focus on a hydrogen-air system where the non-equilibrium nozzle will require a reaction mechanism of this system. First, the MoC in CPG will be applied to generate a nozzle contour which will be used to obtain data such as nozzle shape, temperature, and pressure distribution, etc. Once this is achieved, a calling function for CEA into MATLAB will be required to do analysis on TCE and non-equilibrium nozzle flows with an iterative technique, described in Chapter 2. CEA is useful for this application because it calculates thermodynamic properties in equilibrium for a variety of oxidizer and fuel mixtures. The throat thermodynamic data provided by CEA is needed to begin any MoC calculation.

1.4 Methodology

Semester 1: Problem definition and MATLAB script development of MoC in CPG

1. Background research into ramjets and scramjets
2. Background research into MoC in CPG, TCE, and non-equilibrium
3. Develop a calling function for CEA into a MATLAB script
4. Develop functionality for CPG MoC calculations in MALTAB
5. Determine required inputs for CPG, TCE, and non-equilibrium MoC
6. Determine reaction mechanism for non-equilibrium MoC
7. Collect Arrhenius law and third body efficiency data
8. Begin TCE MoC calculation with integration of CEA in a MATLAB script

Semester 2: Development of TCE and Non-equilibrium MoC Nozzle

1. Develop a MATLAB script to iteratively calculate TCE MoC
2. Validate the TCE MoC data with data from [6] as a benchmark

3. Develop a MATLAB script to iteratively calculate non-equilibrium MoC
4. Validate the non-equilibrium MoC data with [6] as a benchmark
5. Further develop the generated nozzle contour in a CFD solver

1.5 MoC with Calorically Perfect Gas Assumption

From chapter 11.4, *Determination of the Compatibility Equations* [17], the compatibility equations define the flow angle and velocity magnitude. The eqn. (5) below is the Prandtl-Meyer Expansion Angle derived using calorically perfect gas assumptions. For the case of reacting flow in nozzles, multiple articles such as [6] utilize the vibrationally frozen speed of sound for calculations in non-equilibrium and local chemical equilibrium flows.

$$K^{+} = \theta - \upsilon(M) \tag{2}$$

$$K^{-} = \theta + \upsilon(M) \tag{3}$$

$$\theta_{max} = \frac{\upsilon(M)_e}{2} \tag{4}$$

$$\upsilon(M) = \sqrt{\frac{\gamma+1}{\gamma-1}}\left(\tan^{-1}\left(\sqrt{\frac{\gamma-1}{\gamma+1}(M^2-1)}\right)\right) - \tan^{-1}\sqrt{M^2-1} \tag{5}$$

$$\mu = \sin^{-1}\left(\frac{1}{M}\right) \tag{6}$$

$$d\theta = \sqrt{(M^2-1)}\frac{dV}{V} = \theta_{max}/N \tag{7}$$

The problem is assumed to be isentropic with calorically perfect gas to calculate a preliminary nozzle wall with MoC. Another analysis will be done for a fixed divergent angle nozzle wall, which does not prevent the expansion waves from forming. From the technical note, Performance of Several Method-of-Characteristics Exhaust Nozzles [8], it was discovered that low-pressure ratios create severe pressure separation in contoured nozzle walls. Therefore, the code needs to be able to calculate fixed divergent wall contours. Shown below in **Figure *6***, the fixed divergent angle nozzle wall does not prevent expansion waves from forming, as seen by the blue's characteristic net. Due to the continuous expansion of the flow, streamlines do not exit radially. The exit Mach number is lower, creating a decrease in thrust. However, these losses outweigh the risk for low-pressure nozzles. Furthermore, the contoured nozzle wall terminates the expansion waves generated from the supersonic flow, as shown in **Figure *7***. Doing so straightens the flow radially; therefore, the expansion ratio can be a function of exit Mach number. Setting the inclination angle at the wall equal to the previous node's inclination angle will prevent the reflection of the Mach line.

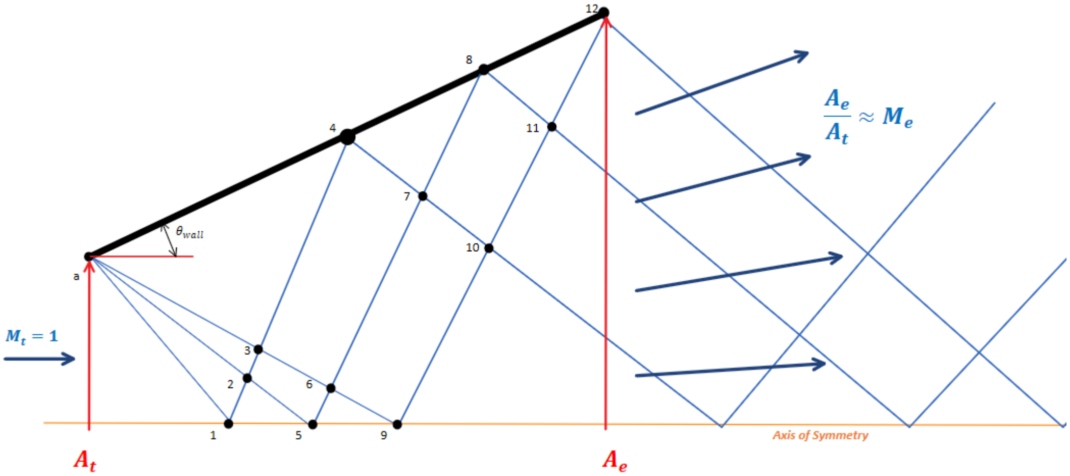


Figure 6: Fixed wall angle nozzle.

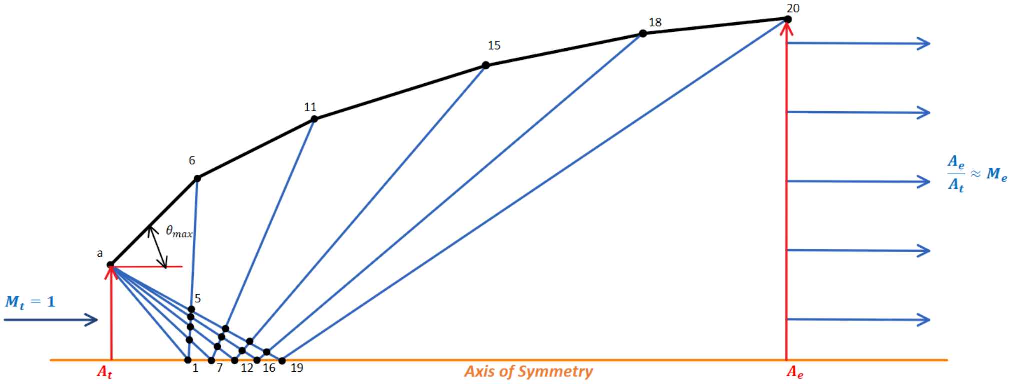


Figure 7: Contoured nozzle.

1.6 MoC with Reactions: General Equations

Outlined in Craig's article [6], the 2D and axisymmetric flow field of a nozzle is calculated by the Navier-Stokes equations and reaction rate equations for the streamline coordinate system shown in **Figure *8***. The streamline coordinate system (ξ,η) is shown in light blue. The cartesian coordinate system (x,y) is shown in dark blue. The streamline coordinates are related to the characteristic lines with the Mach angle. The angle between the horizontal and the streamline direction is the inclination angle. These relationships between coordinate systems are necessary for the derivation of the system of equation.

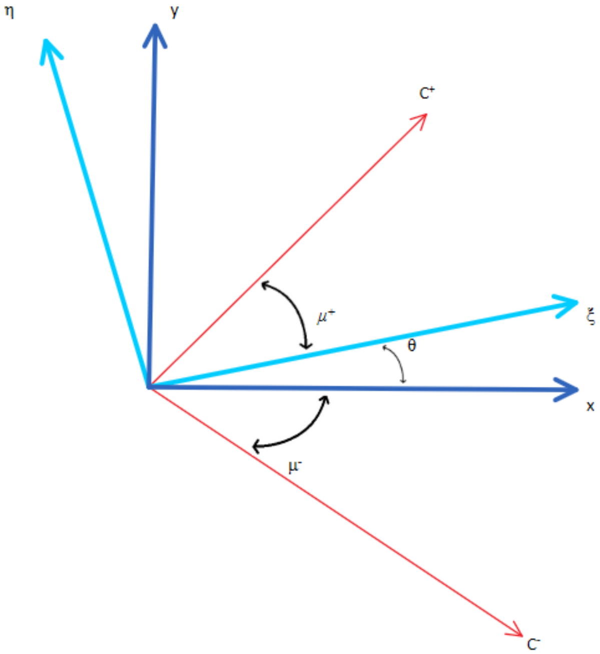


Figure 8: MoC with reactions coordinate system

Equations (8), (9), (10), and (11) represent the continuity, momentum, and energy equations, respectively, and correlate to the streamline coordinate system. Manipulating the system of equations to reduce the number of variables in each equation, one can find the unique solution of the system of unknowns ($\frac{dP}{d\xi}, \frac{dP}{d\eta}, \frac{d\theta}{d\xi}, \frac{d\theta}{d\eta}$). Cramer's rule is applied to find a solution where the determinant is zero.

Continuity:

$$\rho\frac{\partial V}{\partial \xi} + V\frac{\partial \rho}{\partial \xi} + \rho V\frac{\partial \theta}{\partial \eta} = 0 \qquad (8)$$

Momentum:

$$\rho V \frac{\partial V}{\partial \xi} + \frac{\partial P}{\partial \xi} = 0 \tag{9}$$

Momentum:

$$\rho V^2 \frac{\partial \theta}{\partial \xi} + \frac{\partial P}{\partial \eta} = 0 \tag{10}$$

Energy:

$$\frac{\partial h}{\partial \xi} + V \frac{\partial V}{\partial \xi} = 0 \tag{11}$$

Rate of Species Change:

$$V \frac{\partial Y_i}{\partial \xi} = \dot{Y} \tag{12}$$

General Net Reaction Rate:

$$[\dot{C}] = (\nu_i' - \nu_i'')\left\{k_f \prod_{i=1}^{N}([C_i])^{\nu_{i'}} \; - \; k_b \prod_{i=1}^{N}([C_i])^{\nu_{i''}}\right\} \tag{13}$$

Total Enthalpy Equation:

$$H_{Total} = h + \frac{V^2}{2} \tag{14}$$

Below are the final equations required to solve the 2D MoC in reacting flow. On the right-hand side of equation (13), the forward and rearward reaction rates ($k_f \; and \; k_b$) require thermodynamic data to solve. This data can be extracted from CEA through MATLAB by user input of chamber pressure, mixture ratio, fuel and oxidizer types, and temperature. To initiate the characteristic net calculation, MATLAB would pull data calculated by CEA to define the chamber conditions, throat conditions, and the species present in the flow at each of those locations.

Along Characteristics:

$$\frac{\Delta \mathrm{x}}{\Delta \mathrm{y}} = tan\left[\bar{\theta} \pm \overline{\mu_f}\right] \tag{15}$$

$$\frac{\Delta \mathrm{P}}{\bar{\rho}\bar{V}^2 tan\overline{\mu_f}} \pm \Delta\theta \; = \; \frac{\overline{a_f}}{\bar{V}} \Delta \mathrm{c}\left[-\overline{\sum_{\iota=1}^{n} \frac{h_\iota}{C_p T V} \dot{Y}} + \overline{\sum_{\iota=1}^{n} \frac{\mathcal{M}}{\mathcal{M}_\iota V} \dot{Y}}\right] \tag{16}$$

Along Streamlines:

$$\bar{V}\Delta \mathrm{V} + \frac{\Delta \mathrm{P}}{\rho} = 0 \tag{17}$$

$$\frac{\Delta \mathrm{x}}{\Delta \mathrm{y}} = \frac{\bar{v}}{\bar{u}} \tag{18}$$

$$\Delta\xi = \sqrt{(x_{n-1} - x_n)^2 + (y_{n-1} - y_n)^2} = \Delta \mathrm{c} \cos\mu_f \tag{19}$$

The next steps require implementing these equations into a code specifically written for the Hydrogen-Air reaction mechanism. In equations (15), (16), (17), and (18), the average between two points is taken across the state variables in the flow that is not a derivative. The delta symbol, Δ, represents the difference between the two points.

Referring to **Figure *7***, the characteristic net consists of nodes within the nozzle. Point A has properties of the flow at the throat, which can be obtained from CEA. Both point A and point one are required to solve for point two. The equations presented above require knowledge of the state variables at point two, even though it is unknown. The iteration technique converges point two by solving it roughly five times when the previous iteration is within 0.1% agreement. The finite difference method does not suffice to solve for reaction rates. The article [6] recommends integrating on the streamline between two points, as seen in **Figure 9** below.

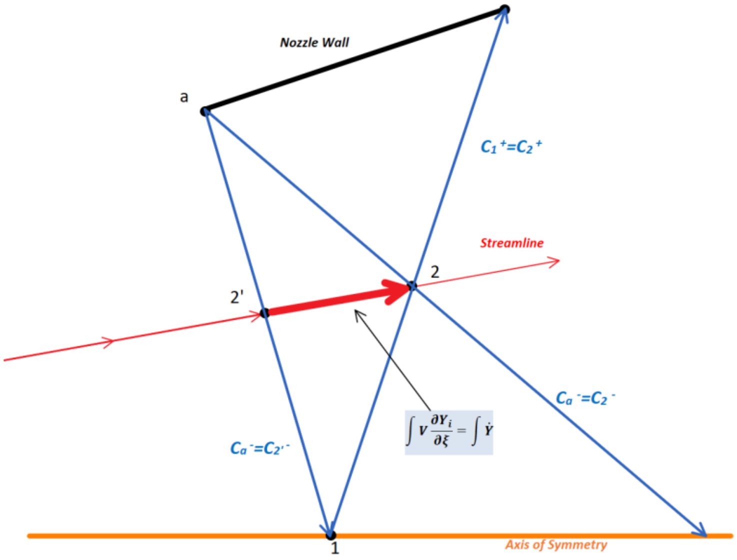


Figure 9: Streamline integration. *This is a close-up image of the characteristic net either from figure 1 or 2. To solve for the reaction rate, eq. (12) is integrated along the streamline between points 2' and 2. CEA will be utilized for this step.*

2. Method of Characteristics in TCE and Non-Equilibrium

2.1 MoC Approach with Chemical Equilibrium and Non-Equilibrium

From Craig's article [6], it was found that vibrationally frozen speed of sound was sufficient for TCE and non-equilibrium MoC. If we assume vibrationally frozen flow at the nozzle throat, this will allow for simplifications in the derivation of MoC with reaction rates. Utilizing eqns. (2-6) the frozen Mach angle can be calculated. The turn angle will be determined by eqn. (7) with the number of characteristic lines, N. In **Figure 9**, the (x,y) coordinate of point 2 is found with the following equations:

$$x_2 = \frac{y_a - y_1 - x_a\, tan\left(\overline{\theta - \mu_f}\right)_{2-a} + x_1\, tan\left(\overline{\theta + \mu_f}\right)_{2-1}}{tan\left(\overline{\theta + \mu_f}\right)_{2-1} - tan\left(\overline{\theta - \mu_f}\right)_{2-a}} \tag{20}$$

$$y_2 = y_1 + (x_2 - x_1)\, tan\left(\overline{\theta + \mu_f}\right)_{2-1} \tag{21}$$

In addition to eqns. (2-6) and eqns. (20-21), the remaining equations are required for nozzles to solve for chemical equilibrium and non-equilibrium.

$$P_2 = \frac{B_1 + B_a + \theta_1 - \theta_a + \frac{P_1}{A_1} + \frac{P_a}{A_a}}{\left(\frac{1}{A_1} + \frac{1}{A_a}\right)} \tag{22}$$

$$\theta_2 = B_1 + \theta_1 - \left(\frac{P_2 - P_1}{A_1}\right) \tag{23}$$

$$A_1 = \left(\overline{\rho V^2 \tan\mu_f}\right)_{2-1} \tag{24}$$

$$A_a = \left(\overline{\rho V^2 \tan\mu_f}\right)_{2-a} \tag{25}$$

$$B_1 = \sin\overline{\mu_f}_{2-1} \Delta C_{2-1} \left\{ \left(\sum_i^n \overline{\frac{\mathcal{M}_{mix}}{\mathcal{M}_i V}}\, \dot{Y}_i \right)_{2-1} - \left(\sum_i^n \overline{\frac{h_i}{C_{pi} T V}}\, \dot{Y}_i \right)_{2-1} \right\} \tag{26}$$

$$B_a = \sin \overline{\mu_f}_{2-a} \Delta C_{2-a} \left\{ \left(\sum_i^n \overline{\frac{\mathcal{M}_{m\iota x}}{\mathcal{M}_\iota V} \dot{Y}_\iota} \right)_{2-a} - \left(\sum_i^n \overline{\frac{h_\iota}{C_{p\iota} T V} \dot{Y}_\iota} \right)_{2-a} \right\} \tag{27}$$

$$\Delta C_{2-1} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{28}$$

$$\Delta C_{2-a} = \sqrt{(x_2 - x_a)^2 + (y_2 - y_a)^2} \tag{29}$$

To begin the calculation of eqns. (24-24), the density and velocity are assumed to be the same as the previous point for the first iteration, i.e.:

- $\rho_{2-1}(1) = \rho_1$and $\rho_{2-a}(1) = \rho_a$
- $V_{2-1}(1) = V_1$and $V_{2-a}(1) = V_a$

The following iterations will be calculated with values with subscript 2 are from the previous iteration. These values are plugged into eqns. (24-25):

$$\rho_{2-1} = \frac{\rho_2 + \rho_1}{2} \tag{30}$$

$$\rho_{2-a} = \frac{\rho_2 + \rho_a}{2} \tag{31}$$

$$V_{2-1} = \frac{V_2 + V_1}{2} \tag{32}$$

$$V_{2-a} = \frac{V_2 + V_a}{2} \tag{33}$$

Once pressure at point 2 is calculated, the velocity at point 2 can be obtained with eqn. 10. Expanded out to relate to point 2:

$$V_2 = V_{2'} - \left(\frac{P_2 - P_{2'}}{\overline{V_{2-2'} \rho_{2-2'}}} \right) \tag{34}$$

Where $V_{2'}$and $P_{2'}$ are found through linear interpolation of point a and 1. The values, $V_{2-2'}$ and $\rho_{2-2'}$, are found through taking the average of points a and 1.

2.2.1 Example Reaction Rate Mechanism

This section illustrates an example reaction mechanism for non-equilibrium flow, the reaction mechanism for H_2-Air will be discussed further in chapter 4. To begin the calculation of eqns. (26-27), the reaction rates must be known at point 2 and point 1. With the same iterative process to calculate eqns. (24-25), the reaction rate is assumed to equal the previous point for the first iteration. Utilizing eqn. (13), the reaction rates can be derived for each species, an example of a simple reaction mechanism is demonstrated below.

An example reaction mechanism from [17] is:

- Reaction 1: $H + O_2 \leftrightharpoons OH + O$
- Reaction 2: $O + H_2 \leftrightharpoons OH + H$
- Reaction 3: $OH + H_2 \leftrightharpoons H_2O + H$
- **Overall Reaction:** $\boldsymbol{2H_2 + O_2 \leftrightharpoons OH + H + H_2O}$

Adding up reactions 1, 2 and 3 will result in the overall reaction taking place within the nozzle. The arrows represent recombination and dissociation reaction rate coefficients, k_b and k_f respectively. The reaction rates of each species are derived from eqn. (13) and an example reaction rates of species [OH] and [H] are shown:

$$\frac{d[OH]}{dt} = \left\{k_{f1}[H][O_2] - k_{b1}[OH][O]\right\} + \left\{k_{f2}[O][H_2] - k_{b2}[OH][H]\right\} - \left\{k_{f3}[OH][H_2] - k_{b3}[H_2O][H]\right\}$$

$$\frac{d[H]}{dt} = -\left\{k_{f1}[H][O_2] - k_{b1}[OH][O]\right\} + \left\{k_{f2}[O][H_2] - k_{b2}[OH][H]\right\} + \left\{k_{f3}[OH][H_2] - k_{b3}[H_2O][H]\right\}$$

The subscript numbers on k_b and k_f represent one of the 3 reactions in the reaction mechanism shown above, Reactions 1-3. k_f can be obtained through a statistical equation called the Arrhenius equation. The Arrhenius equation can be found in Chapter 13.3, *Chemical Non-equilibrium: The Chemical Rate Equation* [4].

$$k_f = C_f T^{n_f} e^{\frac{-E}{\mathcal{R}T}} \tag{35}$$

The recombination rate constant, k_b, can be found with the equilibrium constants, K_c and k_p:

$$\frac{k_f}{k_b} = K_c = \left(\frac{1}{\mathcal{R}T}\right)^{\Delta n} k_p \tag{36}$$

$$k_{p1} = \frac{P_{OH}P_O}{P_H P_{O_2}} \tag{37}$$

$$k_{p2} = \frac{P_{OH}P_H}{P_O P_{H_2}} \tag{38}$$

$$k_{p3} = \frac{P_{H_2O}P_H}{P_{OH}P_{H_2}} \tag{39}$$

$$\chi_i = \frac{P_i}{P_{mix}} \tag{40}$$

With the reaction rates defined, the pressure at point 2 can be calculated as eqns. (22-29) are all known after calculating the first iteration. For the second iteration onward, CEA will be needed to calculate the mole fractions. To do this, the previous iteration of temperature, pressure, and mole fractions at point two will be put into the CEA (t, p) problem to calculate the next iteration mole fraction. According to the article [6] assuming that the specific heat at constant pressure and the molecular weight of the mixture are the same at both points, then the temperature at point 2 can be calculated as:

$$T_2 = T_1 + \frac{V_1^2 - V_2^2}{2C_p} \tag{41}$$

The iterative process of calculating eqns. (22) and (23) continues until the pressure calculated between each iteration agrees within 0.1%. Once this is achieved, point 2 is successfully calculated, then point 3 will be calculated next with information from point 2.

3. Reaction Rate Mechanism and CPG MoC Initial Results

3.1 Full Hydrogen-Air Reaction Rate Mechanism for Non-equilibrium MoC

The reaction mechanism for H_2-Air is obtained from reference [6]. In **Table *1***, the elementary reactions are listed. Reactions 1-5, and 12-14 are bimolecular reactions, whereas Reactions 6-11 are trimolecular reactions with a third body, M. The third body can be any of the species within the mechanism, which means there are multiple reaction rates for those trimolecular reactions.

n	R_n
1	$K_{f_1}[H_2][O_2] - K_{b_1}[OH][OH]$
2	$K_{f_2}[H][O_2] - K_{b_2}[OH][O]$
3	$K_{f_3}[O][H_2] - K_{b_3}[OH][H]$
4	$K_{f_4}[OH][H_2] - K_{b_4}[H_2O][H]$
5	$K_{f_5}[OH][OH] - K_{b_5}[H_2O][O]$
6*	$K_{f_6}[H][H][M] - K_{b_6}[H_2][M]$
7*	$K_{f_7}[H][OH][M] - K_{b_7}[H_2O][M]$
8*	$K_{f_8}[H][O][M] - K_{b_8}[OH][M]$
9*	$K_{f_9}[O][O][M] - K_{b_9}[O_2][M]$
10*	$K_{f_{10}}[N][N][M] - K_{b_{10}}[N_2][M]$
11*	$K_{f_{11}}[N][O][M] - K_{b_{11}}[NO][M]$
12	$K_{f_{12}}[N][O_2] - K_{b_{12}}[NO][O]$
13	$K_{f_{13}}[N_2][O] - K_{b_{13}}[NO][N]$
14	$K_{f_{14}}[N_2][O_2] - K_{b_{14}}[NO][NO]$

Table 1: Reaction Rates for H2-Air Mechanism. *The * represents that the equation changes with each species, M. Furthermore, for n = 6-11, reaction rate coefficients have multiple third body efficiencies, η_b, tabulated in table 2.*

The reaction rates require either k_f or k_b to be calculated with the Arrhenius equation. Reactions 6-9 are all recombination reactions that have an activation energy of 0. Therefore, eqn. (36) is modified for recombination:

$$K_b = C_b T^{n_b}$$

Where C_b is a constant and η_b is the recombination efficiency which changes depending on the third body species, most species have an efficiency of 1. Reactions 10, 11, 13, 14 have forward reaction rate data from reference [19].

Reaction No.	Chemical Reaction	C_f, C_b*	E	n_f	n_b $[H_2,\ H_2O,\ N_2]$	Ref.
1	$H_2 + O_2 \leftrightharpoons OH + OH$	1.7e13	48,000	0	-	[4]
2	$H + O_2 \leftrightharpoons OH + O$	2.2e14	16,800	0	-	[4]
3	$O + H_2 \leftrightharpoons OH + H$	1.8e10	8,900	1	-	[4]
4	$OH + H_2 \leftrightharpoons H_2O + H$	2.2e13	5150	0	-	[4]
5	$OH + OH \leftrightharpoons H_2O + O$	6.3e12	1090	0	-	[4]
6	$H + H + M \leftrightharpoons H_2 + M$	1.0e18*	0	0	[2, 6, 1]	[4],[20]
7	$H + OH + M \leftrightharpoons H_2O + M$	2.2e22*	0	0	[0.73, 3.65, 1]	[4],[20]
8	$H + O + M \leftrightharpoons OH + M$	5.0e07*	0	0	[1, 5, 1]	[4],[20]
9	$O + O + M \leftrightharpoons O_2 + M$	1.2e17*	0	0	[2.4, 15.4, 1]	[4]
10+	$N + N + M \leftrightharpoons N_2 + M$	5.4e13	225,000	0	-	[19]
11+	$N + O + M \leftrightharpoons NO + M$	2.4e15	149,000	0	-	[19]
12	$N + O_2 \leftrightharpoons NO + O$	6.4e09	63000	1	-	[19]
13+	$N_2 + O \leftrightharpoons NO + N$	1.6e13	0	0	-	[19]
14+	$N_2 + O_2 \leftrightharpoons 2NO$	1.3e12	63,800	0	-	[19]

Table 2: Reaction mechanism for H2-Air. *The + represents data that was taken for dissociation, k_f. The third body efficiency, η_b, had values from both [4] and [20]. The most recent data is from [20] so the most recent η_b is utilized for calculations.*

This reaction mechanism has nine species: $H_2,\ O_2,\ N_2,\ H_2O,\ OH,\ H,\ O,\ N,\ NO$. Since the third body reactions can be any species, reactions 6-11 each have 9 reaction rates for each species. Luckily, most of the species have an efficiency of 1, so most of the reaction rate coefficients will not change, but the mole fractions must be considered.

Below are the common equations found in the reactions rates and are simplified into a variable, R_n, where n represents the reaction number in **Table 2**.

$$\frac{d[H_2]}{dt} = -R_1 - R_3 - R_4 + \boldsymbol{R_6}([\boldsymbol{M}], \boldsymbol{n_b}) \tag{42}$$

$$\frac{d[O_2]}{dt} = -R_1 - R_2 + \boldsymbol{R_9}([\mathbf{M}], \boldsymbol{n_b}) - R_{12} - R_{14} \tag{43}$$

$$\frac{d[OH]}{dt} = R_1 + R_2 + R_3 - -2R_5 - \boldsymbol{R_7}([\mathbf{M}], \boldsymbol{n_b}) - \boldsymbol{R_8}([\mathbf{M}], \boldsymbol{n_b}) \tag{44}$$

$$\frac{d[H]}{dt} = -R_2 + R_3 + R_4 - 2\boldsymbol{R_6}([\boldsymbol{M}], \boldsymbol{n_b}) - \boldsymbol{R_7}([\mathbf{M}], \boldsymbol{n_b}) - \boldsymbol{R_8}([\mathbf{M}], \boldsymbol{n_b}) \tag{45}$$

$$\frac{d[O]}{dt} = R_2 - R_3 + R_5 - \boldsymbol{R_8}([\mathbf{M}], \boldsymbol{n_b}) - 2\boldsymbol{R_9}([\mathbf{M}], \boldsymbol{n_b}) - \boldsymbol{R_{11}}([\mathbf{M}], \boldsymbol{n_b}) + R_{12} - R_{13} \tag{46}$$

$$\frac{d[H_2O]}{dt} = R_5 + \boldsymbol{R_7}([\boldsymbol{M}], \boldsymbol{n_b}) \tag{47}$$

$$\frac{d[N]}{dt} = -2\boldsymbol{R_{10}}([\boldsymbol{M}], \boldsymbol{n_b}) - \boldsymbol{R_{11}}([\boldsymbol{M}], \boldsymbol{n_b}) - R_{12} + R_{13} \tag{48}$$

$$\frac{d[N_2]}{dt} = \boldsymbol{R_{10}}([\boldsymbol{M}], \boldsymbol{n_b}) - R_{13} - R_{14} \tag{49}$$

$$\frac{d[NO]}{dt} = \boldsymbol{R_{11}}([\boldsymbol{M}], \boldsymbol{n_b}) + \boldsymbol{R_{12}}([\boldsymbol{M}], \boldsymbol{n_b}) + R_{13} + 2R_{14} \tag{50}$$

The reactions rates for each species are shown above in simplified form. Keep note that the variables $R_6, R_7, R_8, R_9, R_{10},\ and\ R_{11}$ are all a function of the species being considered, M, and the corresponding third body efficiencies, η_b. With the reaction mechanism and the reaction rates for each species, the non-equilibrium MoC can be calculated in MATLAB using the iterative technique explained in chapter 2.

3.2 Chemical Equilibrium MoC

As discussed in chapter 2, the reaction rates are zero for equilibrium conditions. This means that the dissociation and recombination occur at equal rates, therefore the rate at equilibrium is equal to the forward or backward reaction rate.

With this simplification, the information required to solve the problem is the mass fractions for each species and the density of the mixture, shown in eqns. (24) and (25). The mass fractions can be determined from using the CEA (t,p) problem. The inputs into CEA would be temperature, pressure, and mole fractions of each species. The output will include enthalpy of the mixture, each mole/mass fraction of the species, and density of the mixture.

A MATLAB function developed by Edgar Aguilar [21], extrapolates data from CEA for a rocket problem. The script will need to be modified slightly to call CEA with a (t,p) problem, and the code is shown in Appendix C.

3.3 CPG MoC Results

The code (see Appendix B) which will eventually include the ability to solve the MoC in thermo-chemical equilibrium, is currently capable to calculate the nozzle contour with the basic CPG assumption and using the specific heat ratio given by CEA. The inputs for CEA are below:

- Chamber Pressure = 10 atm
- Supersonic Area Ratio = 5
- Infinite Contraction Ratio
- Equilibrium
- Fuel = $H_2(L)$
    - 100% rel. wt.
    - Injection Temp. = 20.17 K
- Oxidizer = Air
    - 100% rel. wt.
    - Injection Temp. = 800 K

The injection temperature for air is dependent on the conditions upstream of the airbreathing engine. However, for liquids, the injection temperature is usually known.

The outputs from CEA are inputs for the MoC calculation such as:

- Exit Mach Number = 2.839
- Throat Gamma = 1.1983
- Chamber Temperature = 2597.87 K
- Chamber Density = 1.1436 $\frac{kg}{m^3}$
- Throat Temperature = 2381.98 K
- Throat Pressure = 5.6447 atm
- Throat Sonic Velocity = 984.3 m/s

The remaining inputs are the throat height and number of characteristic lines for the calculation which is user defined. Below in **Figure *10*** to **Figure *14*** is the output of MoC in CPG with 100 characteristic lines. This data can be used to compare to TCE and non-equilibrium flows when the functionality is developed.

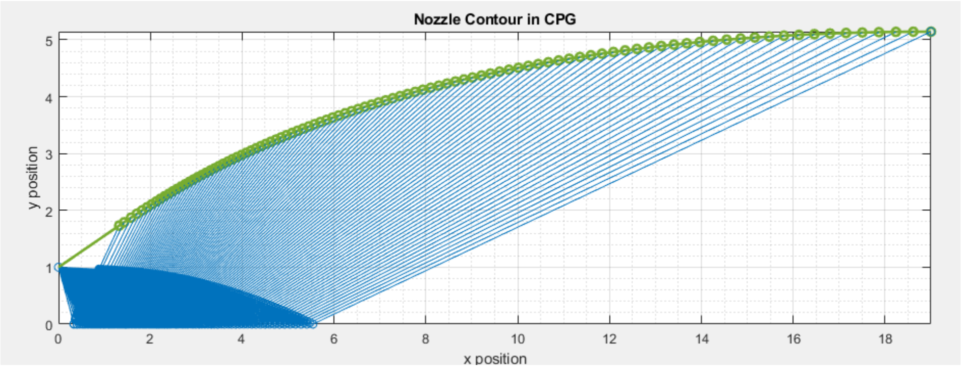


Figure 10: CPG MoC characteristic mesh with 100 characteristic lines

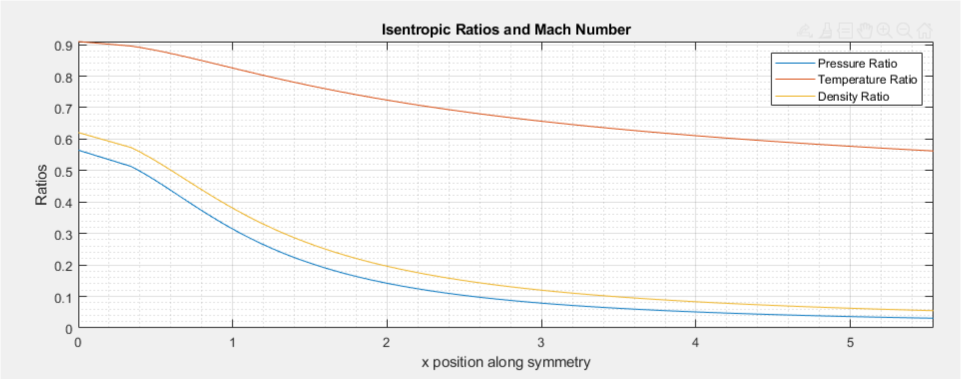


Figure 11: Isentropic ratios for pressure, temperature, and density along the nozzle symmetry line.

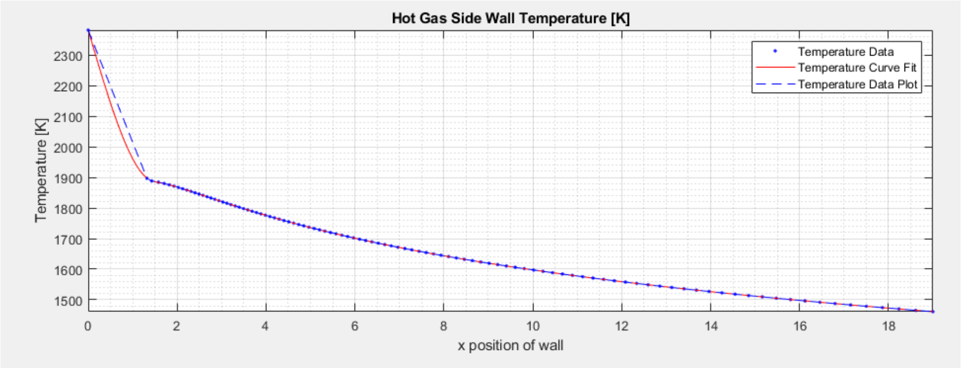


Figure 12: Gas side wall temperature.

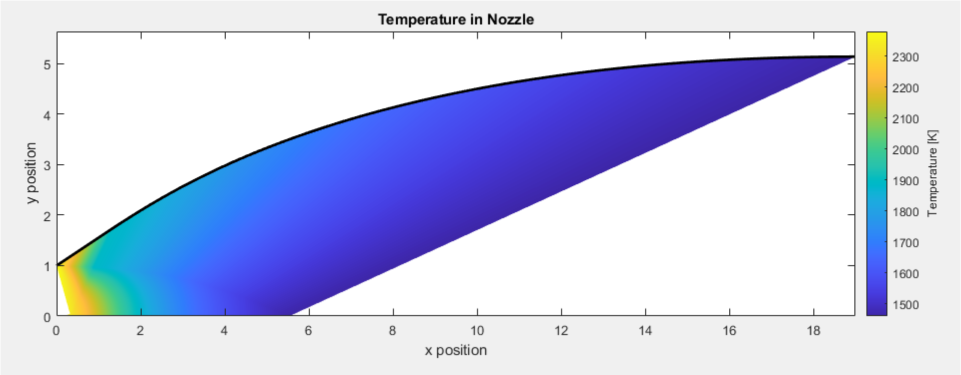


Figure 13: Two-dimensional temperature contour.

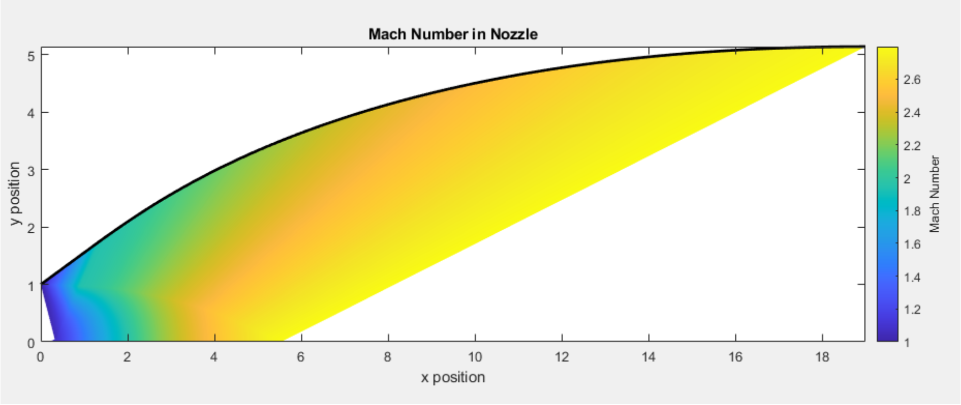


Figure 14: Mach number within the nozzle.

4. Thermochemical Equilibrium Nozzle Design Methodology

4.1 Four-Step Reduced H_2-Air Reaction Mechanism and Reaction Rate Constants

The nozzle design in thermochemical equilibrium requires a reaction mechanism for the propellants used. An example of a reaction mechanism is presented in chapter 3 above, however for TCE, this reaction mechanism is too lengthy, and it is required to calculate the reaction rate of each species. However according to an article called “Complex CSP for Chemistry Reduction and Analysis” [22], a detailed reaction mechanism for H2/Air with 9 species can be reduced to a 4-step reaction mechanism with 6 species. This 4-step reaction mechanism will be used for the TCE calculation within pressure limits of 0.2 atm to 20 atm and equivalence ratios of 0.7 to 2.

The 4-step reduced reaction mechanism is as follows:

- **Reaction 1:** $\boldsymbol{H + O_2 \leftrightharpoons OH + O}$
- **Reaction 2:** $\boldsymbol{O + H_2 \leftrightharpoons OH + H}$
- **Reaction 3:** $\boldsymbol{OH + H_2 \leftrightharpoons H_2O + H}$
- **Reaction 4:** $\boldsymbol{H + H + M \leftrightharpoons H_2 + M}$

These four reactions are also within the detailed reaction mechanism shown in **Table *2*** as 2,3,4, and 6th reactions. For each of these reactions, there is a forward rate constant (k_f) required to calculate the reaction rates of each species, the data for these constants are also shown in **Table *2***.

Below are the forward reaction rate equations for each reaction:

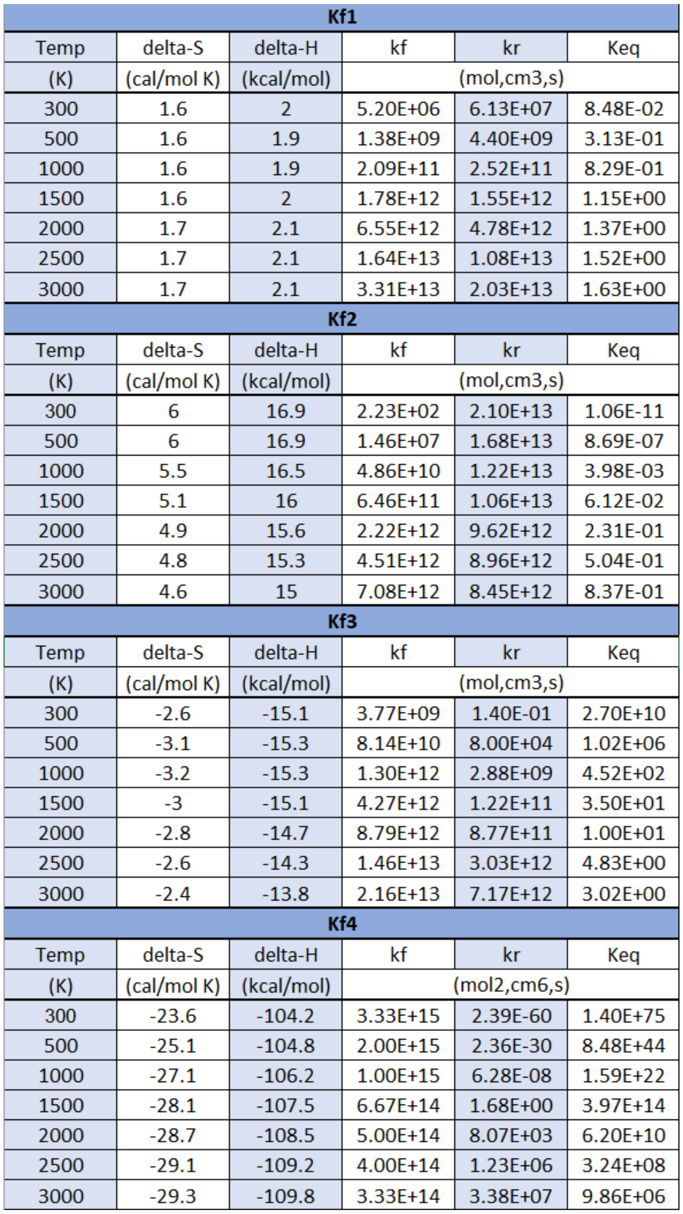
$$\text{Reaction 1: } k_{f1} = 3.87x10^4 T^{2.7} e^{\frac{-6260\frac{cal}{mol}}{\Re T}} [\frac{cm^3}{mol*s}] \qquad (51)$$

$$\text{Reaction 2: } k_{f2} = 2.65x10^{16} T^{-0.6707} e^{\frac{-17041\frac{cal}{mol}}{\Re T}} [\frac{cm^3}{mol*s}] \qquad (52)$$

$$\text{Reaction 3: } k_{f3} = 2.16x10^8 T^{1.51} e^{\frac{-3430\frac{cal}{mol}}{\Re T}} [\frac{cm^3}{mol*s}] \qquad (53)$$

$$\text{Reaction 4: } k_{f4} = \frac{1x10^{18}}{T} \ [\frac{cm^6}{mol^2*s}] \qquad (54)$$

**Note: each equation is converted from centimeters to meters when calculating for reaction rate, and $\Re$, the universal gas constant, is converted into [calories/mol*K] to match the units of the activation energy.*



Kf1					
Temp	delta-S	delta-H	kf	kr	Keq
(K)	(cal/mol K)	(kcal/mol)	(mol,cm3,s)		
300	1.6	2	5.20E+06	6.13E+07	8.48E-02
500	1.6	1.9	1.38E+09	4.40E+09	3.13E-01
1000	1.6	1.9	2.09E+11	2.52E+11	8.29E-01
1500	1.6	2	1.78E+12	1.55E+12	1.15E+00
2000	1.7	2.1	6.55E+12	4.78E+12	1.37E+00
2500	1.7	2.1	1.64E+13	1.08E+13	1.52E+00
3000	1.7	2.1	3.31E+13	2.03E+13	1.63E+00

Kf2					
Temp	delta-S	delta-H	kf	kr	Keq
(K)	(cal/mol K)	(kcal/mol)	(mol,cm3,s)		
300	6	16.9	2.23E+02	2.10E+13	1.06E-11
500	6	16.9	1.46E+07	1.68E+13	8.69E-07
1000	5.5	16.5	4.86E+10	1.22E+13	3.98E-03
1500	5.1	16	6.46E+11	1.06E+13	6.12E-02
2000	4.9	15.6	2.22E+12	9.62E+12	2.31E-01
2500	4.8	15.3	4.51E+12	8.96E+12	5.04E-01
3000	4.6	15	7.08E+12	8.45E+12	8.37E-01

Kf3					
Temp	delta-S	delta-H	kf	kr	Keq
(K)	(cal/mol K)	(kcal/mol)	(mol,cm3,s)		
300	-2.6	-15.1	3.77E+09	1.40E-01	2.70E+10
500	-3.1	-15.3	8.14E+10	8.00E+04	1.02E+06
1000	-3.2	-15.3	1.30E+12	2.88E+09	4.52E+02
1500	-3	-15.1	4.27E+12	1.22E+11	3.50E+01
2000	-2.8	-14.7	8.79E+12	8.77E+11	1.00E+01
2500	-2.6	-14.3	1.46E+13	3.03E+12	4.83E+00
3000	-2.4	-13.8	2.16E+13	7.17E+12	3.02E+00

Kf4					
Temp	delta-S	delta-H	kf	kr	Keq
(K)	(cal/mol K)	(kcal/mol)	(mol2,cm6,s)		
300	-23.6	-104.2	3.33E+15	2.39E-60	1.40E+75
500	-25.1	-104.8	2.00E+15	2.36E-30	8.48E+44
1000	-27.1	-106.2	1.00E+15	6.28E-08	1.59E+22
1500	-28.1	-107.5	6.67E+14	1.68E+00	3.97E+14
2000	-28.7	-108.5	5.00E+14	8.07E+03	6.20E+10
2500	-29.1	-109.2	4.00E+14	1.23E+06	3.24E+08
3000	-29.3	-109.8	3.33E+14	3.38E+07	9.86E+06

Table 3: Reaction Rate Data. Above is data tabulated for each reaction in the 4-step reduced reaction mechanism. The Arrhenius rate law equations where benchmarked with this data provided by reference [20].

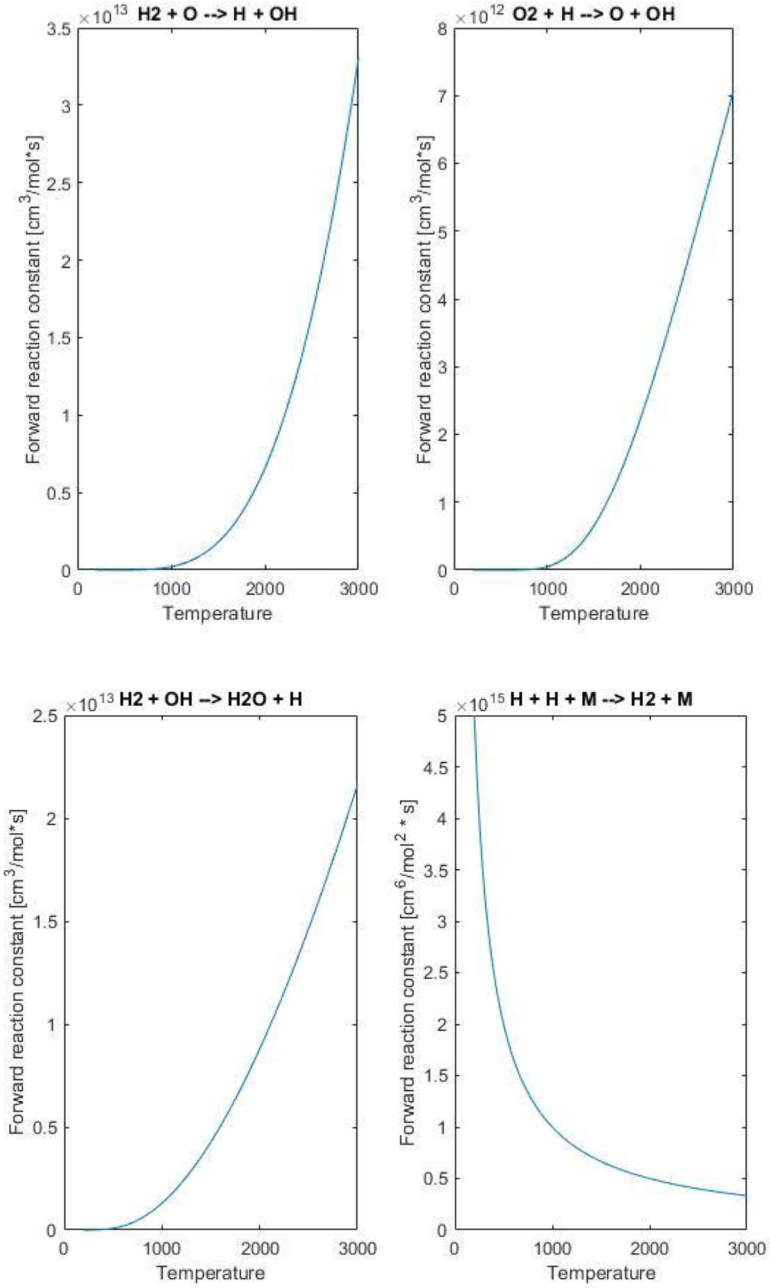


Figure 15: Forward reaction rate constants calculated with Arrhenius rate law.

4.2 Reaction Rates in Equilibrium

Recalling equations 13, 26, and 27 (repeated here), we focus attention on the reaction rate, $\dot{Y}_\imath$, (highlighted in red) for each of the six species being considered in the flow.

$$[\dot{C}] = (\nu_i' - \nu_i'')\{k_f \prod_{i=1}^{N}([C_i])^{\nu_i'} \ - \ k_b \prod_{i=1}^{N}([C_i])^{\nu_i''}\} \tag{13}$$

$$B_1 = \sin \overline{\mu_f}_{2-1} \Delta C_{2-1} \left\{ \left(\sum_i^n \overline{\frac{\mathcal{M}_{m\imath x}}{\mathcal{M}_\imath V} \boldsymbol{\dot{Y}_\imath}} \right)_{2-1} - \left(\sum_i^n \overline{\frac{h_\imath}{C_p T V} \boldsymbol{\dot{Y}_\imath}} \right)_{2-1} \right\} \tag{26}$$

$$B_a = \sin \overline{\mu_f}_{2-a} \Delta C_{2-a} \left\{ \left(\sum_i^n \overline{\frac{\mathcal{M}_{m\imath x}}{\mathcal{M}_\imath V} \boldsymbol{\dot{Y}_\imath}} \right)_{2-a} - \left(\sum_i^n \overline{\frac{h_\imath}{C_p T V} \boldsymbol{\dot{Y}_\imath}} \right)_{2-a} \right\} \tag{27}$$

Equation 13 needs to be in terms of mass fraction with respect to time. To convert into mass fraction with respect to time, the following ideal gas with species equations are applied:

$$[C_i] = \frac{X_i P_{mix}}{\Re T} \rightarrow X_i = \frac{[C_i]\Re T}{P_{mix}} \tag{55}$$

$$Y_i = \frac{X_i \mathcal{M}_i}{\mathcal{M}_{mix}} \rightarrow Y_i = \frac{[C_i]\Re T \mathcal{M}_i}{\mathcal{M}_{mix} P_{mix}}; \text{ where } \frac{P_{mix}}{\mathcal{M}_{mix} R T} = \frac{\rho_{mix}}{\mathcal{M}_{mix}}$$

$$\boldsymbol{Y_i = \frac{[C_i]\mathcal{M}_i}{\rho_{mix}}} \tag{56}$$

$$[\dot{Y}] = \frac{d[Y]}{dt} = \frac{\mathcal{M}_i}{\rho_{mix}} (\nu_i' - \nu_i'') \left\{ k_f \prod_{i=1}^{N} \left(\frac{Y_i \rho_{mix}}{\mathcal{M}_i} \right)^{\nu_i'} \ - \ k_b \prod_{i=1}^{N} \left(\frac{Y_i \rho_{mix}}{\mathcal{M}_i} \right)^{\nu_i''} \right\}$$

Furthermore, recall equation 36, which related the forward reaction rate constant to the rearward reaction rate constant.

$$\frac{k_f}{k_b} = K_c = \left(\frac{1}{\Re T} \right)^{\Delta n} k_p, \text{ where } \Delta n = \nu_i'' - \nu_i' \tag{36}$$

For reactions 1, 2, and 3, the Δn is equal to zero because the stoichiometric coefficients are equal (products minus reactants). For reaction 4, the Δn= -1 due to third body reactant, [M], which can be any of the six species colliding with hydrogen. The reaction rate in terms of mass fraction is further simplified in terms of mole fractions which will be convenient for CEA calculations.

$$\left[\dot{Y}\right]_{NET} = \frac{d[Y]}{dt}_{NET} = \frac{\mathcal{M}_i}{\rho_{mix}}\left(\nu_i' - \nu_i''\right)\left\{k_f \prod_{i=1}^{N}\left(\frac{X_i\rho_{mix}}{\mathcal{M}_{mix}}\right)^{\nu_i'} - \frac{1}{\left(\frac{1}{\mathcal{R}T}\right)^{\Delta n} k_{p_i}}\prod_{i=1}^{N}\left(\frac{X_i\rho_{mix}}{\mathcal{M}_{mix}}\right)^{\nu_i''}\right\} \tag{57}$$

Below are the full reaction rates for each species with the given 4-step reduced reaction mechanism.

$$[\dot{Y}_{H_2}]_{NET} = \left(\frac{\mathcal{M}_{H_2}\rho_{mix}}{\mathcal{M}_{mix}^2}\right)\left\{-k_{f_1}\left(X_{H_2}X_O - \frac{X_H X_{OH}}{k_{p_1}}\right) - k_{f_3}\left(X_{H_2}X_{OH} - \frac{X_{H_2O}X_H}{k_{p_3}}\right) + \frac{k_{f_4}}{\mathcal{R}T}\left(X_H^2 P_{mix} - \frac{X_{H_2}}{k_{p_4}}\right)\right\} \tag{58}$$

$$[\dot{Y}_{O_2}]_{NET} = \left(\frac{\mathcal{M}_{O_2}\rho_{mix}}{\mathcal{M}_{mix}^2}\right)\left\{-k_{f_2}\left(X_{O_2}X_H - \frac{X_O X_{OH}}{k_{p_2}}\right)\right\} \tag{59}$$

$$[\dot{Y}_O]_{NET} = \left(\frac{\mathcal{M}_O\rho_{mix}}{\mathcal{M}_{mix}^2}\right)\left\{-k_{f_1}\left(X_{H_2}X_O - \frac{X_H X_{OH}}{k_{p_1}}\right) + k_{f_2}\left(X_{O_2}X_H - \frac{X_O X_{OH}}{k_{p_2}}\right)\right\} \tag{60}$$

$$\begin{aligned}[\dot{Y}_H]_{NET} = \left(\frac{\mathcal{M}_H\rho_{mix}}{\mathcal{M}_{mix}^2}\right)\Bigg\{&k_{f_1}\left(X_{H_2}X_O - \frac{X_H X_{OH}}{k_{p_1}}\right) - k_{f_2}\left(X_{O_2}X_H - \frac{X_O X_{OH}}{k_{p_2}}\right) + k_{f_3}\left(X_{H_2}X_{OH} - \frac{X_{H_2O}X_H}{k_{p_3}}\right)\\ &+ \frac{2k_{f_4}}{\mathcal{R}T}\left(X_H^2 P_{mix} - \frac{X_{H_2}}{k_{p_4}}\right)\Bigg\}\end{aligned} \tag{61}$$

$$[\dot{Y}_{OH}]_{NET} = \left(\frac{\mathcal{M}_{OH}\rho_{mix}}{\mathcal{M}_{mix}^2}\right)\left\{k_{f_1}\left(X_{H_2}X_O - \frac{X_H X_{OH}}{k_{p_1}}\right) + k_{f_2}\left(X_{O_2}X_H - \frac{X_O X_{OH}}{k_{p_2}}\right) - k_{f_3}\left(X_{H_2}X_{OH} - \frac{X_{H_2O}X_H}{k_{p_3}}\right)\right\} \tag{62}$$

$$[\dot{Y}_{H_2O}]_{NET} = \left(\frac{\mathcal{M}_{H_2O}\rho_{mix}}{\mathcal{M}_{mix}^2}\right)\left\{k_{f_3}\left(X_{H_2}X_{OH} - \frac{X_{H_2O}X_H}{k_{p_3}}\right)\right\} \tag{63}$$

Equations (58-63) represent the full reaction rate for each species which is needed for non-equilibrium flows; however, in equilibrium the net reaction rate is equal to zero. This means the finite rates of both the forward and rearward reactions are equal.

$$Net\ Rate = Forward\ Rate - Rearward\ Rate = 0 \implies Forward\ Rate = Rearward\ Rate$$

The net reaction rates for equilibrium are shown below. Equations (64-69) will be needed for the calculation of equations (26) and (27).

$$[\dot{Y}_{H_2}]_{NET} = \left(\frac{\mathcal{M}_{H_2}\rho_{mix}}{\mathcal{M}_{mix}^2}\right)\left\{-k_{f_1}\left(X_{H_2}X_O\right) - k_{f_3}\left(X_{H_2}X_{OH}\right) + \frac{k_{f_4}}{\mathcal{R}T}\left(X_H^2 P_{mix}\right)\right\} \tag{64}$$

$$[\dot{Y}_{O_2}]_{NET} = \left(\frac{\mathcal{M}_{O_2}\rho_{mix}}{\mathcal{M}_{mix}^2}\right)\left\{-k_{f_2}\left(X_{O_2}X_H\right)\right\} \tag{65}$$

$$[\dot{Y}_O]_{NET} = \left(\frac{\mathcal{M}_O\rho_{mix}}{\mathcal{M}_{mix}^2}\right)\left\{-k_{f_1}\left(X_{H_2}X_O\right) + k_{f_2}\left(X_{O_2}X_H\right)\right\} \tag{66}$$

$$[\dot{\boldsymbol{Y}}_{\boldsymbol{H}}]_{\boldsymbol{NET}} = \left(\frac{\mathcal{M}_H \rho_{mix}}{\mathcal{M}_{mix}^2}\right)\left\{k_{f_1}(X_{H_2}X_O) - k_{f_2}(X_{O_2}X_H) + k_{f_3}(X_{H_2}X_{OH}) + \frac{2k_{f_4}}{\mathcal{R}T}(X_H^2 P_{mix})\right\} \tag{67}$$

$$[\dot{\boldsymbol{Y}}_{\boldsymbol{OH}}]_{\boldsymbol{NET}} = \left(\frac{\mathcal{M}_{OH} \rho_{mix}}{\mathcal{M}_{mix}^2}\right)\{k_{f_1}(X_{H_2}X_O) + k_{f_2}(X_{O_2}X_H) - k_{f_3}(X_{H_2}X_{OH})\} \tag{68}$$

$$[\dot{\boldsymbol{Y}}_{\boldsymbol{H_2O}}]_{\boldsymbol{NET}} = \left(\frac{\mathcal{M}_{H_2O} \rho_{mix}}{\mathcal{M}_{mix}^2}\right)\{k_{f_3}(X_{H_2}X_{OH})\} \tag{69}$$

4.3 Absolute Enthalpy for Each Species

Looking back at equations (26) and (27), the last unknown to solve for is the absolute enthalpy of each species (highlighted in red below).

$$B_1 = \sin \overline{\mu_f}_{2-1} \Delta C_{2-1} \left\{\left(\sum_i^n \frac{\overline{\mathcal{M}_{m\imath x}}}{\mathcal{M}_\imath V}\dot{Y}_\imath\right)_{2-1} - \left(\sum_i^n \frac{\overline{\boldsymbol{h_\imath}}}{C_p T V}\dot{Y}_\imath\right)_{2-1}\right\} \tag{26}$$

$$B_a = \sin \overline{\mu_f}_{2-a} \Delta C_{2-a} \left\{\left(\sum_i^n \frac{\overline{\mathcal{M}_{m\imath x}}}{\mathcal{M}_\imath V}\dot{Y}_\imath\right)_{2-a} - \left(\sum_i^n \frac{\overline{\boldsymbol{h_\imath}}}{C_p T V}\dot{Y}_\imath\right)_{2-a}\right\} \tag{27}$$

Absolute enthalpy is defined as the sensible enthalpy obtained from statistical thermodynamics added with the heat of formation at zero kelvin.

$$h_i = \frac{(H - E_0)_i + \left(\Delta H_f^0\right)_i}{\mathcal{M}_i}$$

The term, $\frac{(H-E_0)_i}{\mathcal{M}_i}$, is the sensible enthalpy per unit mass. Furthermore, according to reference [4], chapter 11, the sensible enthalpy can be calculated directly from the internal energies of the molecules or atoms.

For atoms, internal energy per unit mass:

$$e = \frac{3}{2}RT + e_{elec}$$

For molecules, internal energy per unit mass:

$$e = e_{translational} + e_{rotational} + e_{vibrational} + e_{electronic}$$

or

$$e = \frac{3}{2}RT + RT + \frac{\frac{\hbar v}{KT}}{\mathrm{e}^{\frac{\hbar v}{KT}} - 1}RT + e_{elec}$$

Where the vibrational energy term contains plank’s constant ($\hbar$), fundamental vibrational frequency of a molecule (v), and Boltzmann’s constant (K).

Recall the second law of thermodynamics:

$$h = e + pV \longrightarrow h = e + RT$$

The sensible enthalpy for atoms becomes:

$$h = \frac{5}{2}RT + e_{elec}$$

Sensible enthalpy for molecules:

$$h = RT\left(\frac{7}{2} + \frac{\frac{\hbar v}{KT}}{\mathrm{e}^{\frac{\hbar v}{KT}} - 1}\right) + e_{elec}$$

It would be extremely cumbersome to include and calculate the sensible enthalpy taking into consideration each molecule’s fundamental vibrational frequency; luckily the JANAF tables provide the sensible and heat of formation data for each species being considered. The tables are provided in Appendix D. Finally, equations (26) and (27) are now known and can be solved to determine the pressure at each point in the characteristic mesh.

References

[1] Martin, J., “Ramjet Propulsion for Single-Stage-to-Orbit Vehicles,” Journal of Spacecraft and Rockets, Vol. 15, No. 5, NASA Langley Research Center, Hampton Va., October 1978.

[2] Heiser, W., Pratt, D., “Hypersonic Airbreathing Propulsion,” 5th ed., AIAA, Washington, DC.

[3] Anderson, J. D., “Fundamentals of Aerodynamics,” McGraw-Hill Education, 6th ed., New York, NY, 2017.

[4] Anderson, J. D., “Hypersonic and High-Temperature Gas Dynamics,” AIAA Education Series, 2nd edition, Washington, D.C., 2006.

[5] Gordon S., Zeleznik, F. J., Huff, V. N., “A General Method for Automatic Computation of Equilibrium Compositions and Theoretical Rocket Performance of Propellants,” NASA Technical Note, NASA TN D-132, Washington, D.C., October 1959.

[6] Craig, R. R. “Applying the Method of Characteristics to Analyze the Flow Field of a Chemically Reacting Gas in A Two-Dimensional or an Axisymmetric Nozzle”, NASA Technical Report, AFAPL-TR-65-20, April 1965.

[7] Lick, W., “Inviscid Flow of a Reacting Mixture of Gases Around a Blunt Body,” Pierce Hall, Harvard University, Cambridge, Mass, April 1959.

[8] Farley, J. M., Campbell, C. E., “Performance of Several Method-of-Characteristics Exhaust Nozzles,” Lewis Research Center, NASA TN D-293, Cleveland, Ohio, October 1960.

[9] Lezberg, E., “Recombination of Hydrogen-Air Combustion Products,” Lewis Research Center, NASA TN D-1052, Cleveland, Ohio, August 1961.

[10] Wegner, P.P., “Supersonic Nozzle Flow with a Reacting Gas Mixture,” *AIP Physics of Fluids,* Vol. 2, No. 264 (2012).

[11] Yumusak, M., “Design Optimization of Rocket Nozzles in Chemically Reacting Flows,” *Elsevier*, 65 (2012), pp. 25-34.

[12] Yamaguchi, T. Hizawa, T., Ichikawa, T., Kudo T., “Development and verification of a supersonic nozzle with rectangular cross section at a Mach number of 2.8 for a scramjet model combustor,” *Institute of Fluid Science,* Tohoku university, November 2018.

[13] Han, S. “Numerical Analysis of Thermo-chemical Non-equilibrium Flows in a Model Scramjet Engine,” *MDPI Energies,* January 2020.

[14] Zhang, S. Li, X., “Research progress on active thermal protection for hypersonic vehicles,” *Elsevier,* Progress in Aerospace Science, No 119 (2020).

[15] Shen, J. Lu, “The thermo-chemical non-equilibrium scale effects of the high enthalpy nozzle,” *Advances in Aerodynamics,* China Academy of Aerospace and Aerodynamics Beijing, China, 2020.

[16] McBride, B.J., Gordon, S., Reno, M.A., “Thermodynamic Data for Fifty Reference Elements,” NASA Technical Paper 3287, January 1993.

[17] Anderson, J. D., “Modern Compressible Flow with Historical Perspective,” McGraw-Hill, 3rd ed., New York, New York, 2003.

[18] Wang, H., “Combustion Chemistry,” Princeton-CEFRC Summer School on Combustion, Stanford University, June 2015.

[19] Ludwig, C. B., Sulzmann, K. G. P., “Chemical Kinetics in Shock-Heated Air,” Armed Services Technical Information Agency, Arlington, Virginia, AD 257-971, February 1961.

[20] Serauskas, B. ”GRI-3.0 Program,” Gas Research Institute, obtained from: combustion.berkeley.edu/gri-mech/data/frames.html

[21] Aguilar, E., MATLAB CEA Calling Function, San Jose State University, 2019.

[22] Tianfeng, L., Yiguang J., Chung, L. K., “Complex CSP for Chemistry Reduction and Analysis,” The Combustion Institute, Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, 2001.

Appendix A: Method of Characteristics with Reaction Rates Derivation

Note that this derivation is the same in [6] but shown here with more detailed steps and clarifications. Beginning with the continuity, momentum, and energy equations in terms of coordinate system shown below:

- Continuity Equation: $\rho \frac{\partial V}{\partial \xi} + V \frac{\partial \rho}{\partial \xi} + \rho V \frac{\partial \theta}{\partial \eta} = 0$

- Momentum Equation: $\rho V \frac{\partial V}{\partial \xi} + \frac{\partial P}{\partial \xi} = 0$

- Momentum Equation: $\rho V^2 \frac{\partial \theta}{\partial \xi} + \frac{\partial P}{\partial \eta} = 0$

- Energy Equation: $\frac{\partial h}{\partial \xi} + V \frac{\partial V}{\partial \xi} = 0$

Mass fractions with respect to streamline:

$$\frac{\partial Y_i}{\partial \xi} = \frac{\dot{Y}}{V}$$

Reaction rate equation in terms of mole fractions, this form is needed for CEA:

$$\dot{\chi} = (\nu_i' - \nu_i'')\left\{k_f \prod_{i=1}^{N} (\chi_i)^{\nu_i{}'} \; - \; k_b \prod_{i=1}^{N} (\chi_i)^{\nu_i{}''}\right\}$$

From the perfect gas law, we have the general partial derivative of density with respect to the streamline from general chain rule:

$$\frac{\partial \rho}{\partial \xi} = \frac{\partial \rho}{\partial P}\frac{\partial P}{\partial \xi} + \frac{\partial \rho}{\partial T}\frac{\partial T}{\partial \xi} + \frac{\partial \rho}{\partial Y_i}\frac{\partial Y_i}{\partial \xi}$$

Below are some useful transformations of the perfect gas law:

$$\rho = \frac{P}{\mathcal{R}T\frac{\sum_i Y_i}{\sum_i \mathcal{M}_i}} \Rightarrow P = \frac{\rho\mathcal{R}T}{\mathcal{M}_{mix}}, where\ \mathcal{M}_{mix} = \frac{\sum_i \mathcal{M}_i}{\sum_i Y_i} \Rightarrow \rho = \frac{P\mathcal{M}_{mix}}{\mathcal{R}T}$$

Taking the partial derivative against each state variable, we have the following simplifications for $\frac{\partial \rho}{\partial \xi}$:

$$\frac{\partial \rho}{\partial P} = \frac{\mathcal{M}_{mix}}{\mathcal{R}T};\quad \frac{\partial \rho}{\partial T} = \frac{-\rho}{T};\quad \frac{\partial \rho}{\partial Y_i} = \frac{-\rho}{\sum_i Y_i} = \frac{-\rho\left(\frac{1}{\sum \mathcal{M}_i}\right)}{\left(\sum \frac{Y_i}{\mathcal{M}_i}\right)}$$

$$\frac{\partial \rho}{\partial \xi} = \frac{\mathcal{M}_{mix}}{\mathcal{R}T}\frac{\partial P}{\partial \xi} - \frac{\rho}{T}\frac{\partial T}{\partial \xi} - \frac{\rho\left(\frac{1}{\sum \mathcal{M}_i}\right)}{\left(\sum \frac{Y_i}{\mathcal{M}_i}\right)}\frac{\partial Y_i}{\partial \xi}$$

The above equation is substituted into the continuity equation highlighted in red below:

$$\rho\frac{\partial V}{\partial \xi} + V\frac{\partial \rho}{\partial \xi} + \rho V\frac{\partial \theta}{\partial \eta} = 0$$

$$\rho\frac{\partial V}{\partial \xi} + V\left(\frac{\mathcal{M}_{mix}}{\mathcal{R}T}\frac{\partial P}{\partial \xi} - \frac{\rho}{T}\frac{\partial T}{\partial \xi} - \frac{\rho\left(\frac{1}{\sum \mathcal{M}_i}\right)}{\left(\sum \frac{Y_i}{\mathcal{M}_i}\right)}\frac{\partial Y_i}{\partial \xi}\right) + \rho V\frac{\partial \theta}{\partial \eta} = 0$$

After simplification of the above equation, now must simplify $\frac{\partial T}{\partial \xi}$. The goal is to reduce the number of variables in the equation below:

$$\rho\frac{\partial V}{\partial \xi} + \rho V\left(\frac{1}{P}\frac{\partial P}{\partial \xi} - \frac{1}{T}\frac{\partial T}{\partial \xi} - \frac{1}{\sum Y_i}\frac{\partial Y_i}{\partial \xi}\right) + \rho V\frac{\partial \theta}{\partial \eta} = 0$$

Evaluating the partial derivative of the energy equation along a streamline:

$$\frac{\partial h}{\partial \xi} = \sum_i h_i\frac{\partial Y_i}{\partial \xi} + \sum_i Y_i\frac{\partial h_i}{\partial \xi}$$

Shifting focus to $\frac{\partial h_i}{\partial \xi}$, to obtain $\frac{\partial T}{\partial \xi}$:

$$\frac{\partial h_i}{\partial \xi} = \frac{\partial h_i}{\partial T}\frac{\partial T}{\partial \xi} = C_{p_i}\frac{\partial T}{\partial \xi}$$

*Note that $\frac{\partial h_i}{\partial T}$ is by definition $\left(\frac{\partial h_i}{\partial T}\right)_{P=const.} = C_{p_i}$
Substituting $\frac{\partial h_i}{\partial \xi}$ simplification into the energy equation:

$$\frac{\partial h}{\partial \xi} = \sum_i h_i \frac{\partial Y_i}{\partial \xi} + \sum_i Y_i\, C_{p_i}\frac{\partial T}{\partial \xi}$$

Then rearranging the energy equation in terms of $\frac{\partial T}{\partial \xi}$:

$$\frac{\partial h}{\partial \xi} + V\frac{\partial V}{\partial \xi} = 0 \implies \sum_i h_i \frac{\partial Y_i}{\partial \xi} + \sum_i Y_i\, C_{p_i}\frac{\partial T}{\partial \xi} + V\frac{\partial V}{\partial \xi} = 0$$

$$\frac{\partial T}{\partial \xi} = -\frac{\sum_i h_i}{C_{p_{mix}}}\frac{\partial Y_i}{\partial \xi} - \frac{V}{C_{p_{mix}}}\frac{\partial V}{\partial \xi}$$

Substituting $\frac{\partial T}{\partial \xi}$ into the continuity equation and combining like terms:

$$\rho\frac{\partial V}{\partial \xi} + \rho V\left(\frac{1}{P}\frac{\partial P}{\partial \xi} + \frac{1}{T}\left(\frac{\sum_i h_i}{C_{p_{mix}}}\frac{\partial Y_i}{\partial \xi} + \frac{V}{C_{p_{mix}}}\frac{\partial V}{\partial \xi}\right) - \frac{1}{\sum Y_i}\frac{\partial Y_i}{\partial \xi}\right) + \rho V\frac{\partial \theta}{\partial \eta} = 0$$

$$\left(\rho + \frac{\rho V^2}{TC_{p_{mix}}}\right)\frac{\partial V}{\partial \xi} + \frac{\rho V}{P}\frac{\partial P}{\partial \xi} + \rho V\left(\frac{\sum_i h_i}{TC_{p_{mix}}} - \frac{1}{\sum Y_i}\right)\frac{\partial Y_i}{\partial \xi} + \rho V\frac{\partial \theta}{\partial \eta} = 0$$

Utilizing the first momentum equation, $\frac{\partial V}{\partial \xi}$ can be simplified in terms of $\frac{\partial P}{\partial \xi}$:

$$\frac{\partial V}{\partial \xi} = -\frac{1}{\rho V}\frac{\partial P}{\partial \xi}$$

Finally, $\frac{\partial Y_i}{\partial \xi}$, can be changed into a constant value:

$$\left(-\frac{1}{V} - \frac{V}{TC_{p_{mix}}} + \frac{\rho V}{P}\right)\frac{\partial P}{\partial \xi} + \rho V\left(\frac{\sum_i h_i}{TC_{p_{mix}}} - \frac{1}{\sum Y_i}\right)\frac{\partial Y_i}{\partial \xi} + \rho V\frac{\partial \theta}{\partial \eta} = 0$$

$$\frac{\partial Y_i}{\partial \xi} = \frac{\dot{Y_\iota}}{V}$$

$$\left(-\frac{1}{V} - \frac{V}{TC_{p_{mix}}} + \frac{\rho V}{P}\right)\frac{\partial P}{\partial \xi} + \rho V \frac{\partial \theta}{\partial \eta} + \rho \dot{Y_\iota}\left(\frac{\sum_i h_i}{TC_{p_{mix}}} - \frac{1}{\sum Y_i}\right) = 0$$

Now manipulating equations for matrix form:

$$\left(-\frac{1}{V} - \frac{V}{TC_{p_{mix}}} + \frac{\rho V}{P}\right)\frac{\partial P}{\partial \xi} + 0\frac{\partial P}{\partial \eta} + 0\frac{\partial \theta}{\partial \xi} + \rho V \frac{\partial \theta}{\partial \eta} = -\rho \dot{Y_\iota}\left(\frac{\sum_i h_i}{TC_{p_{mix}}} - \frac{1}{\sum Y_i}\right)$$

$$A\frac{\partial P}{\partial \xi} + 0\frac{\partial P}{\partial \eta} + 0\frac{\partial \theta}{\partial \xi} + \rho V \frac{\partial \theta}{\partial \eta} = B$$

$$0\frac{\partial P}{\partial \xi} + 1\frac{\partial P}{\partial \eta} + \rho V^2\frac{\partial \theta}{\partial \xi} + 0\frac{\partial \theta}{\partial \eta} = 0$$

$$\partial\xi\frac{\partial P}{\partial \xi} + \partial\eta\frac{\partial P}{\partial \eta} + 0\frac{\partial \theta}{\partial \xi} + 0\frac{\partial \theta}{\partial \eta} = dP$$

$$0\frac{\partial P}{\partial \xi} + 0\frac{\partial P}{\partial \eta} + \partial\xi\frac{\partial \theta}{\partial \xi} + \partial\eta\frac{\partial \theta}{\partial \eta} = d\theta$$

$$\begin{bmatrix} A & 0 & 0 & \rho V \\ 0 & 1 & \rho V^2 & 0 \\ \partial\xi & \partial\eta & 0 & 0 \\ 0 & 0 & \partial\xi & \partial\eta \end{bmatrix}\begin{bmatrix} \frac{\partial P}{\partial \xi} \\ \frac{\partial P}{\partial \eta} \\ \frac{\partial \theta}{\partial \xi} \\ \frac{\partial \theta}{\partial \eta} \end{bmatrix} = \begin{bmatrix} B \\ 0 \\ dP \\ d\theta \end{bmatrix}$$

Need to solve unknown variable, $\frac{\partial P}{\partial \eta}$, with Cramer's rule. The denominator needs to be zero for the solution to be indeterminate and discontinuous, which is also the characteristics lines for non-equilibrium and TCE:

$$\frac{\partial P}{\partial \eta} = \frac{N}{D} = \frac{\begin{bmatrix} A & B & 0 & \rho V \\ 0 & 0 & \rho V^2 & 0 \\ \partial \xi & dP & 0 & 0 \\ 0 & d\theta & \partial \xi & \partial \eta \end{bmatrix}}{\begin{bmatrix} A & 0 & 0 & \rho V \\ 0 & 1 & \rho V^2 & 0 \\ \partial \xi & \partial \eta & 0 & 0 \\ 0 & 0 & \partial \xi & \partial \eta \end{bmatrix}}$$

Solving the determinate for the denominator:

$$det\left(\begin{bmatrix} A & 0 & 0 & \rho V \\ 0 & 1 & \rho V^2 & 0 \\ \partial \xi & \partial \eta & 0 & 0 \\ 0 & 0 & \partial \xi & \partial \eta \end{bmatrix}\right) = 0$$

$$A\begin{bmatrix} 1 & \rho V^2 & 0 \\ \partial \eta & 0 & 0 \\ 0 & \partial \xi & \partial \eta \end{bmatrix} - \rho V \begin{bmatrix} 0 & 1 & \rho V^2 \\ \partial \xi & \partial \eta & 0 \\ 0 & 0 & \partial \xi \end{bmatrix} = 0$$

$$A\left(1\begin{bmatrix} 0 & 0 \\ \partial \xi & \partial \eta \end{bmatrix} - \rho V^2 \begin{bmatrix} \partial \eta & 0 \\ 0 & \partial \eta \end{bmatrix}\right) - \rho V\left(-\partial \xi \begin{bmatrix} 1 & \rho V^2 \\ 0 & \partial \xi \end{bmatrix}\right) = 0$$

$$-A\,\rho V^2 \partial \eta^2 + \rho V\,\partial \xi^2 = 0$$

$$A\,\rho V^2 \partial \eta^2 = \rho V\,\partial \xi^2$$

Rearranging in terms of $\frac{\partial \eta}{\partial \xi}$:

$$\frac{\partial \eta}{\partial \xi} = \pm\sqrt{\frac{1}{A\,V}} = \pm\frac{1}{\sqrt{\frac{\rho V^2}{P} - \frac{V^2}{T C_p} - 1}} = \pm\frac{1}{\sqrt{\frac{C_P \mathcal{M} - \mathcal{R}}{\mathcal{R} T C_p} V^2 - 1}}$$

*Note: definition of frozen speed of sound: $a_f = \frac{\mathcal{R} T C_p}{C_P \mathcal{M} - \mathcal{R}}$

$$\frac{\partial \eta}{\partial \xi} = \pm\frac{1}{\sqrt{M_f^2 - 1}} = \pm\tan \mu_f$$

Now solving numerator in terms of $\frac{\partial P}{\partial \eta}$, using Cramer's rule:

$$\det\left(\begin{bmatrix} A & B & 0 & \rho V \\ 0 & 0 & \rho V^2 & 0 \\ \partial\xi & dP & 0 & 0 \\ 0 & d\theta & \partial\xi & \partial\eta \end{bmatrix}\right) = 0$$

$$A\begin{bmatrix} 0 & \rho V^2 & 0 \\ dP & 0 & 0 \\ d\theta & \partial\xi & \partial\eta \end{bmatrix} + \partial\xi\begin{bmatrix} B & 0 & \rho V \\ 0 & \rho V^2 & 0 \\ d\theta & \partial\xi & \partial\eta \end{bmatrix} = 0$$

$$A\left(-\rho V^2\begin{bmatrix} dP & 0 \\ d\theta & \partial\eta \end{bmatrix}\right) + \partial\xi\left(B\begin{bmatrix} \rho V^2 & 0 \\ \partial\xi & \partial\eta \end{bmatrix} + d\theta\begin{bmatrix} 0 & \rho V \\ \rho V^2 & 0 \end{bmatrix}\right) = 0$$

$$-A\rho V^2 dP\,\partial\eta + B\rho V^2\partial\eta\partial\xi - \rho^2 V^3 d\theta\partial\xi = 0$$

Simplifying further:

$$-A\rho V^2 dP + \rho V^2\left(B\partial\xi - \rho V d\theta\frac{\partial\eta}{\partial\xi}\right) = 0$$

$$\frac{AV}{\rho V^2}dP + d\theta\frac{\partial\eta}{\partial\xi} = \frac{B}{\rho V}\partial\xi$$

*Recall: $\tan^2\mu_f = \frac{1}{AV}$, and $\frac{\partial\eta}{\partial\xi} = \pm\tan\mu_f$

$$\frac{dP}{\rho V^2 \tan^2\mu_f} \pm \frac{d\theta}{\tan\mu_f} = \frac{B}{\rho V}\ \partial\xi$$

Multiplying by $\tan\mu_f$:

$$\frac{dP}{\rho V^2 \tan\mu_f} \pm d\theta = \frac{B\tan\mu_f}{\rho V}\ \partial\xi$$

Where $\partial\xi$=dccosμ_f from coordinate system:

$$\frac{dP}{\rho V^2 \tan\mu_f} \pm d\theta = \frac{B\sin\mu_f}{\rho V}\ dc$$

Also, from geometry, $\sin\mu_f = \frac{1}{M_f} = \frac{a_f}{V}$

$$\frac{dP}{\rho V^2 \tan\mu_f} \pm d\theta = \frac{Ba_f}{\rho V^2}\, dc$$

Arrive at the final equations defining the characteristic lines:

$$\frac{dP}{\rho V^2 \tan\mu_f} \pm d\theta = \frac{a_f}{V} dc \left(\frac{1}{V \sum Y_i} \dot{Y_i} - \frac{\sum_i h_i}{C_{p_{mix}} T V} \dot{Y_i} \right)$$

$$\frac{dx}{dy} = \tan(\theta + \mu_f)$$

Appendix B: MoC Contour Nozzle Code

```
function [Wall_Points, T_wall] = MoC_Analysis(M_e, gamma, Tc, Pc, rc, a_f,
P_t, T_t, C_Lines, Throat_Radius)
%% 2D Method of Characteristics
%% Code Created by Michael Sandoval

%% Inital Conditions
%     ***Nomenclature***
% M_e = Exit Mach Number
% Gamma = Specific Heat Ratio
% Tc = Chamber Total Temperature
% Pc = Chamber Total Pressure
% rc = Chamber Density
% a_f = Throat Speed of Sound
% P_t = Throat Static Pressure
% T_t = Throat Static Temperature
% C_Lines = User defined number of Characteristic Lines
% Throat_Radius = Radius of the throat
% Wall_Points = Wall (x,y) coordinates
% T_wall = Temperature at the wall

% Gamma Constants
y_a = sqrt((gamma+1)/(gamma-1));
y_b = sqrt((gamma-1)/(gamma+1));

%% Solving for divergence angle of the nozzle and initial theta increments
nu_e      = ((y_a)*(atand(y_b*sqrt((M_e^2-1))))-atand(sqrt(M_e^2-1)));
Theta_max = nu_e/2;
DTheta_1  = Theta_max - fix(Theta_max);
% C_input   = input('Number of Characteristic Lines: ');
C         = 1:C_Lines;
dTheta    = fix(Theta_max)/C(end);

%% Solving & Defining MoC Points

% Grid Points
GP = 1:(0.5*(C_Lines^2 + 3*C_Lines));

% Wall Points
WP = zeros(1,length(C));

for i = 1:length(C)
    WP(i) = 0.5*( (2*C(end)+3)*i - i^2 );
end

WP = [0 WP];

% Symmetry Points
SP = zeros(1,length(C));

for i = 1:length(C)
    if i == 1
        SP(i) = GP(1);
    else
        SP(i) = WP(i)+1;
    end
end

WP(1) = [];

% Characteristic node matrix
AA    = zeros(C(end),WP(1));

for i = 1:C(end)
    A = SP(i):WP(i);
    AA(i,i:end) = A;
end

[rows, columns] = size(AA);
SP(1) = [];

% Solving for Theta, Mu, K_p, and K_m
Theta = zeros(C(end),WP(1));
Nu    = zeros(C(end),WP(1));
K_p   = zeros(C(end),WP(1));
K_m   = zeros(C(end),WP(1));
Mu    = zeros(C(end),WP(1));
M     = zeros(C(end),WP(1));

for i = 1:rows
    for ii = 1:columns
        if AA(i,ii) == 0
            Theta(i,ii) = 0;
            Nu(i,ii)    = NaN;
            K_p(i,ii)   = 0;
            K_m(i,ii)   = 0;
            M(i,ii)     = 0;
            Mu(i,ii)    = 0;
        elseif any(SP(:) == AA(i,ii))
            Theta(i,ii)  = 0;
            K_m(i,ii)    = K_m(i-1,ii);
            Nu(i,ii)     = K_m(i,ii) - Theta(i,ii);
            K_p(i,ii)    = Theta(i,ii) - Nu(i,ii);
        elseif any(WP(:) == AA(i,ii))
            Theta(i,ii)  = Theta(i,ii-1);
            K_p(i,ii)    = K_p(i,ii-1);
            Nu(i,ii)     = Theta(i,ii) - K_p(i,ii);
            K_m(i,ii)    = Theta(i,ii) + Nu(i,ii);
        elseif any(C(:) == AA(i,ii))
            if C(1) == AA(i,ii)
                Theta(i,ii) = DTheta_1;
                Nu(i,ii)    = DTheta_1;
                K_p(i,ii)   = 0;
                K_m(i,ii)   = Theta(i,ii)+Nu(i,ii);
            else
                Theta(i,ii) = dTheta*ii;
                Nu(i,ii)    = Theta(i,ii);
                K_p(i,ii)   = 0;
                K_m(i,ii)   = Theta(i,ii)+Nu(i,ii);
            end
        else
            K_p(i,ii)    = K_p(i, ii-1);
            K_m(i,ii)    = K_m(i-1,ii);
            Theta(i,ii)  = 0.5*(K_m(i,ii) + K_p(i,ii));
            Nu(i,ii)     = 0.5*(K_m(i,ii)-K_p(i,ii));
        end
    end
end

% Manipulating Data for Aerospace Toolbox
for i = 1:rows
    for ii = 1:columns
        if isnan(Nu(i,ii))
            Nu(i,ii) = 1;
        else
        end
    end
end

A = size(Nu);

% Aerospace Toolbox: Prandtl-Meyer Expansion Fan Function
for i = 1:A(1)
    for ii = 1:A(2)
        [M(i,ii), Nu(i,ii), Mu(i,ii)] = flowprandtlmeyer(gamma, Nu(i,ii),
'nu');
    end
end

%% Solving and Plotting Points

S_p = zeros(C(end),WP(1));
S_m = zeros(C(end),WP(1));

% Slopes of characteristic lines
for i = 1:rows
    for ii = 1:columns
        if AA(i,ii) == 0
            S_m(i,ii) = 0;
            S_p(i,ii) = 0;
        elseif any(C(:) == AA(i,ii))
            S_m(i,ii) = tand( Theta(i,ii) - Mu(i,ii) );
            S_p(i,ii) = tand( 0.5*(Theta(i,ii) + Theta(i,ii+1) + Mu(i,ii) +
Mu(i,ii+1)) );
        elseif any(WP(:) == AA(i,ii))
            if WP(1) == AA(i,ii)
                S_m(i,ii) = 0;
                S_p(i,ii) = tand( Theta(i,ii) );
            else
                S_m(i,ii) = 0;
                S_p(i,ii) = tand( 0.5*(Theta(i,ii) + Theta(i-1,ii)) );
            end
        else
            S_m(i,ii) = tand( 0.5*(Theta(i,ii) + Theta(i-1,ii) - Mu(i,ii) -
Mu(i-1,ii)) );
```

```
            S_p(i,ii) = tand( 0.5*( Theta(i,ii) + Theta(i,ii+1) + Mu(i,ii) +
Mu(i,ii+1)) );
        end
    end
end

x   = zeros(C(end),WP(1));
y   = zeros(C(end),WP(1));
x_0 = 0;

Tx_Tc = zeros(C(end),WP(1));
Px_Pc = zeros(C(end),WP(1));
rx_rc = zeros(C(end),WP(1));
Vx    = zeros(C(end),WP(1));
Tx    = zeros(C(end),WP(1));
Px    = zeros(C(end),WP(1));
rx    = zeros(C(end),WP(1));

% (x,y) coordinates for each characteristic node
for i = 1:rows
    for ii = 1:columns
        if AA(i,ii) == 0
            x(i,ii) = NaN;
            y(i,ii) = NaN;
        elseif any(C(:) == AA(i,ii))
            if C(1) == AA(i,ii)
                x(i,ii) = x_0 - (Throat_Radius/S_m(i,ii));
                y(i,ii) = 0;
            else
                x(i,ii) = ( Throat_Radius - y(i,ii-1) + S_p(i,ii-1)*x(i,ii-1)
- S_m(i,ii)*x_0 )/( S_p(i,ii-1) - S_m(i,ii) );
                y(i,ii) = S_m(i,ii)*(x(i,ii) - x_0) + Throat_Radius;
            end
        elseif any(WP(:) == AA(i,ii))
            if WP(1) == AA(i,ii)
                x(i,ii) = ( y(i,ii-1) - Throat_Radius + S_p(i,ii)*x_0 -
S_p(i,ii-1)*x(i,ii-1) )/ (S_p(i,ii) - S_p(i,ii-1));
                y(i,ii) = S_p(i,ii-1)*( x(i,ii) - x(i,ii-1) ) + y(i,ii-1);
            else
                x(i,ii) = ( y(i,ii-1) - y(i-1,ii) + S_p(i,ii)*x(i-1,ii) -
S_p(i,ii-1)*x(i,ii-1) )/ (S_p(i,ii) - S_p(i,ii-1));
                y(i,ii) = S_p(i,ii-1)*( x(i,ii) - x(i,ii-1) ) + y(i,ii-1);
            end
        elseif any(SP(:) == AA(i,ii))
            x(i,ii) = x(i-1,ii) - (y(i-1,ii)/S_m(i,ii));
            y(i,ii) = 0;
        else
            x(i,ii) = ( y(i-1,ii) - y(i,ii-1) - S_m(i,ii)*x(i-1,ii) +
S_p(i,ii-1)*x(i,ii-1) )/ ( S_p(i,ii-1) - S_m(i,ii) );
            y(i,ii) = S_m(i,ii)*( x(i,ii) - x(i-1,ii) ) + y(i-1,ii);
        end
            Tx_Tc(i,ii) = ( 1 + 0.5*(gamma-1)*(M(i,ii)^2) )^-1;
            Px_Pc(i,ii) = ( 1 + 0.5*(gamma-1)*(M(i,ii)^2) )^(-gamma/(gamma-
1));
            rx_rc(i,ii) = ( 1 + 0.5*(gamma-1)*(M(i,ii)^2) )^(-1/(gamma-1));
            Vx(i,ii)    = M(i,ii)*a_f;
            Tx(i,ii)    = Tx_Tc(i,ii)*Tc;
            Px(i,ii)    = Px_Pc(i,ii)*Pc;
            rx(i,ii)    = rx_rc(i,ii)*rc;
    end
end

% Characteristic Mesh Plot
for i = 1:rows
    for ii = 1:columns
        if isnan(x(i,ii)) || isnan(y(i,ii))
        elseif any(C(:) == AA(i,ii))
            subplot(3,2,1)
            plot([x_0,x(i,ii)], [Throat_Radius,y(i,ii)],'-
o','color','#0072BD')
            hold on
            plot([x(i,ii),x(i,ii+1)],[y(i,ii),y(i,ii+1)],'-
o','color','#0072BD')
            hold on
        elseif any(WP(:) == AA(i,ii))
            if WP(1) == AA(i,ii)
                plot([x_0,x(i,ii)],[Throat_Radius,y(i,ii)],'-
','color','#77AC30','LineWidth',2)
                hold on
            else
                plot([x(i-1,ii),x(i,ii)],[y(i-1,ii),y(i,ii)],'-
o','color','#77AC30','LineWidth',2)
                hold on
                plot([x(i,ii-1),x(i,ii)],[y(i,ii-1),y(i,ii)],'-
o','color','#0072BD')
                hold on
            end
        else
            plot([x(i,ii),x(i,ii+1)],[y(i,ii),y(i,ii+1)],'-
o','color','#0072BD')
            hold on
            plot([x(i-1,ii),x(i,ii)],[y(i-1,ii),y(i,ii)],'-
o','color','#0072BD')
        end
    end
end

xlim([0,x(end,end)])
ylim([0,y(end,end)])
xlabel('x position')
ylabel('y position')
title('Nozzle Contour in CPG')
grid on
grid minor

x_wall = [x_0; x(:,end)];
y_wall = [Throat_Radius; y(:,end)];

Wall_Points = [x_wall, y_wall];

%% Various Plots for Analysis
Px_s    = zeros(1,length(SP));   % Pressure along symmetry line
```

```
Px_Pc_s = zeros(1,length(SP));   % Pressure ratio along symmetry line
Tx_Tc_s = zeros(1,length(SP));   % Temperature ratio along symmetry line
rx_rc_s = zeros(1,length(SP));   % Density ratio along the symmetry line
M_s     = zeros(1,length(SP));   % Mach number along symmetry line
x_sym   = zeros(1,length(SP));   % x location along symmetry line

% Critical Conditions
M_t = 1;
Tt_Tc = ( 1 + 0.5*(gamma-1)*(M_t^2) )^-1;
Pt_Pc = ( 1 + 0.5*(gamma-1)*(M_t^2) )^(-gamma/(gamma-1));
rt_rc = ( 1 + 0.5*(gamma-1)*(M_t^2) )^(-1/(gamma-1));

% Extracting data from symmetry line
for i = 1:length(SP)
    A = find(SP(i) == AA(:,:));
    Tx_Tc_s(i) = Tx_Tc(A);
    Px_Pc_s(i) = Px_Pc(A);
    rx_rc_s(i) = rx_rc(A);
    Px_s(i)    = Px(A);
    M_s(i)     = M(A);
    x_sym(i)   = x(A);
end

% Manipulating data for contour plots and line graphs
Px_s      = [P_t Px_s]';
Px_Pc_s   = [Pt_Pc Px_Pc_s]';
Tx_Tc_s   = [Tt_Tc Tx_Tc_s]';
rx_rc_s   = [rt_rc rx_rc_s]';
M_s       = [M_t M_s]';
x_sym     = [0 x_sym]';

Px_curve = fit(x_sym, Px_s, 'smoothingspline', 'Normalize', 'on');

x_M       = zeros(1, (C(end)+1));
x_M(:)    = x_0;
x         = [x_M; x];

y_M       = zeros(1, (C(end)+1));
y_M(:)    = Throat_Radius;
y         = [y_M; y];

Tt_M      = zeros(1, (C(end)+1));
Tt_M(:)   = T_t;
Tx        = [Tt_M; Tx];

X = zeros(C(end),WP(1));
Y = zeros(C(end),WP(1));

for i = 1:(rows+1)
    for ii = 1:columns
        if isnan(x(i,ii)) || isnan(y(i,ii))
        else
           [X(i,ii),Y(i,ii)] = meshgrid(x(i,ii),y(i,ii));
           hold on
        end
    end
end

Tx_curve = fit(x(:,end), Tx(:,end), 'smoothingspline', 'Normalize', 'on');
T_wall   = Tx(:,end);

M_M     = zeros(1, (C(end)+1));
M_M(:)  = 1;
M       = [M_M; M];

subplot(3,2,2)
plot(x_sym,Px_Pc_s)
hold on
plot(x_sym,Tx_Tc_s)
hold on
plot(x_sym,rx_rc_s)
hold on
plot(x_sym,M_s)
xlim([0 x_sym(end)])
ylim([0 M_s(end)])
ylim([0 Tx_Tc_s(1)])
title('Isentropic Ratios and Mach Number')
xlabel('x position along symmetry')
ylabel('Ratios')
legend('Pressure Ratio','Temperature Ratio','Density Ratio')
grid on
grid minor

subplot(3,2,3)
plot(Tx_curve,x(:,end), Tx(:,end))
hold on
plot(x(:,end), Tx(:,end),'--b')
xlim([0 x(end,end)])
ylim([Tx(end,end) T_t])
title('Hot Gas Side Wall Temperature [K]')
xlabel('x position of wall')
ylabel('Temperature [K]')
legend('Temperature Data','Temperature Curve Fit', 'Temperature Data Plot')
grid on
grid minor

subplot(3,2,4)
plot(Px_curve,x_sym, Px_s)
hold on
plot(x_sym, Px_s,'--b')
xlim([0 x_sym(end)])
ylim([Px_s(end) P_t])
title('Pressure along Symmetry [atm]')
xlabel('x position along symmetry')
ylabel('Pressure [atm]')
legend('Pressure Data','Pressure Curve Fit', 'Pressure Data Plot')
grid on
grid minor

subplot(3,2,5)
contourf(X,Y,Tx,500,'Fill','on','LineColor','none');
```

```
hold on
plot(x(:,end),y(:,end),'-k','LineWidth',2)
title('Temperature in Nozzle')
xlabel('x position')
ylabel('y position')
xlim([0,x(end,end)])
ylim([0,y(end,end)+0.5])
cc = colorbar;
cc.Label.String = ('Temperature [K]');

subplot(3,2,6)
contourf(X,Y,M,500,'Fill','on','LineColor','none');
hold on
plot(x(:,end),y(:,end),'-k','LineWidth',2)
title('Mach Number in Nozzle')
xlabel('x position')
ylabel('y position')
xlim([0,x(end,end)])
ylim([0,y(end,end)])
cc = colorbar;
cc.Label.String = ('Mach Number');
```

Appendix C: CEA Calling Scripts for MATLAB [20]

C.1) Script for CEA Rocket Problem

```
function [CD, TD, ED, moles_x, caseNamesRkt, dataRkt] = ceaRkt(ExitType, A,
OF, P_0c, Ox, T_Ox, Fuel, T_F)
%% This function ceaRkt.m was created by Edgar Aguliar, and modified by
Michael Sandoval
% Modifications include:
%  - Including all data at chamber, throat, and exit
%  - Includes usaer input of fuel/oxidizer and exit type
%  - More comments, modified input script
% INPUTS:
%  - AeAt = Supersonic area ratio
%  - P_0c = Chamber total pressure
%  - T_Ox = Initial total temperature of the oxidizer
%  - T_F  = Initial total temperature of the fuel
%  - Ox   = Script of oxidizer type
%  - Fuel = Script of oxidizer type
%  - OF   = Oxidizer/fuel mixture ratio by weight
%  - ExitType = 'AeAt' or 'Pi/Pe'

% OUTPUTS:
%  - CD = Chamber Data
%  - TD = Throat Data
%  - ED = Exit Data
%  - caseNames = Names of each row of data

%% Creating text file to read into CEA
% fopen opens the file name '1511.inp' and '1512.inp' for writing acess 'w'
% 'w' opens or creates a new file for writing. Discards existing contents

executeTP1     = fopen('1511.inp','w');
executeTP1file = fopen('1512.inp','w');

fprintf(executeTP1, sprintf('problem    o/f=%f,\n',OF));
fprintf(executeTP1, sprintf('\t rocket equilibrium tcest,k=3800\n'));
fprintf(executeTP1, sprintf('\t p,atm=%f,\n', P_0c));

if strcmpi(ExitType, 'AeAt')
    fprintf(executeTP1, sprintf('\t sup,ae/at=%f,\n', A));
else
    fprintf(executeTP1, sprintf('\t pi/p=%f,\n', A));
end

fprintf(executeTP1, sprintf('react\n'));
fprintf(executeTP1, sprintf('\t fuel=%s wt=100 t,k=%f\n',Fuel, T_F));
fprintf(executeTP1, sprintf('\t oxid=%s wt=100 t,k=%f\n', Ox, T_Ox));
fprintf(executeTP1, sprintf('output short\n'));
fprintf(executeTP1, sprintf('end\n'));

% Writing the executeable file for computer command
fprintf(executeTP1file, sprintf('1511\n'));

fclose(executeTP1);
fclose(executeTP1file);

% MATLAB calls Operating System to use Command: redirecting "<" input file
system('FCEA2.exe < 1512.inp');

% MATLAB opens the output file created by the execution of 1512.inp in
FEA2.exe
% 'rt' opens the file for reading in text mode
fid = fopen('1511.out', 'rt');

% Textscan looks into 1511.out to find all the data in particular pattern
% shown below:
data_cell = textscan(fid,'%s%s%f%f%f','delimiter',',','delimiter','\t',...
    'delimiter','\t', 'delimiter','\t');

fclose(fid);

%% Locating CEA data in output file
dataRkt = data_cell{1}
B    = length(dataRkt);

% Finding first value of output
locator = strfind(dataRkt,'P, BAR');
index   = find(~cellfun(@isempty, locator));
n       = B - index; % Number of CEA output lines to check

caseData_chamber = zeros(n,1);
caseData_throat  = zeros(n,1);
caseData_exit    = zeros(n,1);

%% Extracting data from output file

% Extracting exit data
for ii = 1:n

    dataSplit       = split(dataRkt{ii + index - 1});
    caseValues_exit = dataSplit{end};
    blank           = length(caseValues_exit);

    if blank == 1
         caseValues_exit = [];
    end


    % Names of each output
    if ii <= 13
        caseNamesRkt(ii, 1) = {strcat(dataSplit{1},{' '},dataSplit{2})};
    else
        caseNamesRkt(ii, 1) = {dataSplit(1)};
    end

    checkPos = strfind(caseValues_exit,'+');
    checkNeg = strfind(caseValues_exit,'-');
    % Converting all expontintial data from CEA to read with MATLAB
    if isempty(checkPos) == 1 && isempty(checkNeg) == 1
        caseData_exit(ii) = str2double(caseValues_exit);
    elseif isempty(checkPos) == 0 && isempty(checkNeg) == 0 &&
length(checkNeg) == 2
        Z = [caseValues_exit(1:6), 'e', caseValues_exit(7:end)];
        caseData_exit(ii) = str2double(Z);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 &&
length(checkNeg) == 2
        Z = [caseValues_exit(1:6), 'e', caseValues_exit(7:end)];
        caseData_exit(ii) = str2double(Z);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 && checkNeg == 7
        Z = [caseValues_exit(1:6), 'e', caseValues_exit(7:end)];
        caseData_exit(ii) = str2double(Z);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0
        caseData_exit(ii) = str2double(caseValues_exit);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 && checkNeg == 1
        Z = [caseValues_exit(1:6), 'e', caseValues_exit(7:end)];
        caseData_exit(ii) = str2double(Z);
    elseif isempty(checkPos) == 0 && isempty(checkNeg) == 1 && checkNeg == 7
        Z = [caseValues_exit(1:6), 'e', caseValues_exit(7:end)];
        caseData_exit(ii) = str2double(Z);
    end

    ED = caseData_exit(~isnan(caseData_exit));

end

% Extracting throat data
for ii = 1:n

    dataSplit         = split(dataRkt{ii + index - 1});
    caseValues_throat = dataSplit{end-1};
    blank             = length(caseValues_throat);

    if blank == 1
        caseValues_throat = [];
    end

    checkPos = strfind(caseValues_throat,'+');
    checkNeg = strfind(caseValues_throat,'-');

    if isempty(checkPos) == 1 && isempty(checkNeg) == 1
        caseData_throat(ii) = str2double(caseValues_throat);
    elseif isempty(checkPos) == 0 && isempty(checkNeg) == 0 &&
length(checkNeg) == 2
        Z = [caseValues_throat(1:6), 'e', caseValues_throat(7:end)];
        caseData_throat(ii) = str2double(Z);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 &&
length(checkNeg) == 2
        Z = [caseValues_throat(1:6), 'e', caseValues_throat(7:end)];
        caseData_throat(ii) = str2double(Z);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 && checkNeg == 7
        Z = [caseValues_throat(1:6), 'e', caseValues_throat(7:end)];
        caseData_throat(ii) = str2double(Z);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0
        caseData_throat(ii) = str2double(caseValues_throat);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 && checkNeg == 1
        Z = [caseValues_throat(1:6), 'e', caseValues_throat(7:end)];
        caseData_throat(ii) = str2double(Z);
    elseif isempty(checkPos) == 0 && isempty(checkNeg) == 1 && checkNeg == 7
        Z = [caseValues_throat(1:6), 'e', caseValues_throat(7:end)];
        caseData_throat(ii) = str2double(Z);
    end

    TD = caseData_throat(~isnan(caseData_throat));

end

% Extracting chamber data
for ii = 1:n

    dataSplit = split(dataRkt{ii + index - 1});

    if length(dataSplit) == 2 || length(dataSplit) == 1
        continue
    elseif ii == 3
        caseValues_chamber = dataSplit{end-3};
    else
        caseValues_chamber = dataSplit{end-2};
    end

    blank = length(caseValues_chamber);

    if blank == 1
        caseValues_chamber = [];
    end

    checkPos = strfind(caseValues_chamber,'+');
    checkNeg = strfind(caseValues_chamber,'-');

    if isempty(checkPos) == 1 && isempty(checkNeg) == 1
        caseData_chamber(ii) = str2double(caseValues_chamber);
    elseif isempty(checkPos) == 0 && isempty(checkNeg) == 0 &&
length(checkNeg) == 2
        Z = [caseValues_chamber(1:6), 'e', caseValues_chamber(7:end)];
        caseData_chamber(ii) = str2double(Z);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 &&
length(checkNeg) == 2
        Z = [caseValues_chamber(1:6), 'e', caseValues_chamber(7:end)];
        caseData_chamber(ii) = str2double(Z);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 && checkNeg == 7
        Z = [caseValues_chamber(1:6), 'e', caseValues_chamber(7:end)];
        caseData_chamber(ii) = str2double(Z);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0
        caseData_chamber(ii) = str2double(caseValues_chamber);
    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 && checkNeg == 1
        Z = [caseValues_chamber(1:6), 'e', caseValues_chamber(7:end)];
        caseData_chamber(ii) = str2double(Z);
    elseif isempty(checkPos) == 0 && isempty(checkNeg) == 1 && checkNeg == 7
```

```
        Z = [caseValues_chamber(1:6), 'e', caseValues_chamber(7:end)];
        caseData_chamber(ii) = str2double(Z);
    end

CD = caseData_chamber(~isnan(caseData_chamber));

end
moles_x = 0;
% Converting CEA Rkt case names cell into a string array for indexing
caseNamesRkt = string(caseNamesRkt);

for k = 1:length(TD)
    if strcmp(caseNamesRkt(k),'MOLE')
        caseNamesRkt(k) = [];
    elseif strcmp(caseNamesRkt(k), 'PERFORMANCE')
        caseNamesRkt(k) = [];
    elseif strcmp(caseNamesRkt(k), '*')
        caseNamesRkt(k) = [];
    else
    end
end

% Finding the index for each species from caseNamesRkt
H_i   = find(strcmp(caseNamesRkt, '*H'));
H2_i  = find(strcmp(caseNamesRkt, '*H2'));
H2O_i = find(strcmp(caseNamesRkt, 'H2O'));
NO_i  = find(strcmp(caseNamesRkt, '*NO'));
N2_i  = find(strcmp(caseNamesRkt, '*N2'));
N_i   = find(strcmp(caseNamesRkt, '*N'));
O_i   = find(strcmp(caseNamesRkt, '*O'));
OH_i  = find(strcmp(caseNamesRkt, '*OH'));
O2_i  = find(strcmp(caseNamesRkt, '*O2'));

% "Packaging" results into cells to find any empty cells, []
moles_x   = {TD(H_i), TD(H2_i), TD(H2O_i), TD(NO_i), TD(N2_i), TD(N_i),
TD(O_i), TD(OH_i), TD(O2_i)};
% Placing zero for empty arrays
for i = 1:length(moles_x)
    if isempty(moles_x{i})
        moles_x{i} = 0;
    end
end
% Converting cells to array
moles_x = cell2mat(moles_x);
% Removing scientific notation for scripting into CEA
moles_x = compose("%.5f", moles_x);

delete *inp
delete *out

 end
```

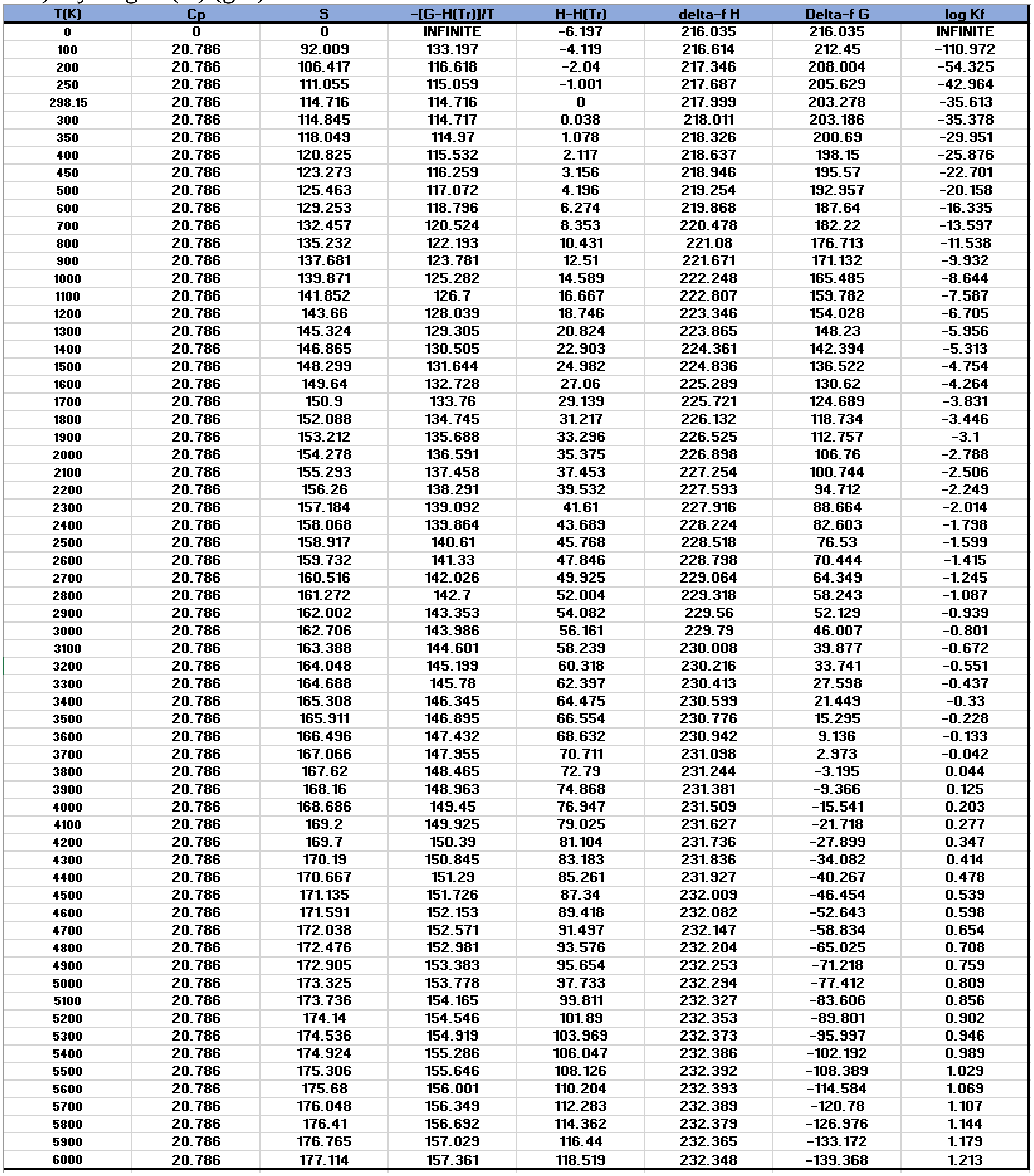
C.2) Script for CEA Temperature and Pressure Problem

```
function [n, x_data, moles_x, caseNamesTP] = ceaTP(T, P, species, moles_x)

% fopen opens the file name '1511.inp' and '1512.inp' for writing acess 'w'
% 'w' opens or creates a new file for writing. Discards existing contents

executeTP1     = fopen('1511.inp','w');
executeTP1file = fopen('1512.inp','w');

% sprintf formats data into string, fprintf writes data into a text file
fprintf(executeTP1, sprintf('problem \n'));
fprintf(executeTP1, sprintf('\t tp   t,k=%d,  p,atm=%d,\n',T,P));
fprintf(executeTP1, sprintf('react\n'));
fprintf(executeTP1, sprintf('\t name=%s moles=%s t,k = %d\n', species(1),
moles_x(1), T));
fprintf(executeTP1, sprintf('\t name=%s moles=%s t,k = %d\n', species(2),
moles_x(2), T));
fprintf(executeTP1, sprintf('\t name=%s moles=%s t,k = %d\n', species(3),
moles_x(3), T));
fprintf(executeTP1, sprintf('\t name=%s moles=%s t,k = %d\n', species(4),
moles_x(4), T));
fprintf(executeTP1, sprintf('\t name=%s moles=%s t,k = %d\n', species(5),
moles_x(5), T));
fprintf(executeTP1, sprintf('\t name=%s moles=%s t,k = %d\n', species(6),
moles_x(6), T));
fprintf(executeTP1, sprintf('\t name=%s moles=%s t,k = %d\n', species(7),
moles_x(7), T));
fprintf(executeTP1, sprintf('\t name=%s moles=%s t,k = %d\n', species(8),
moles_x(8), T));
fprintf(executeTP1, sprintf('\t name=%s moles=%s t,k = %d\n', species(9),
moles_x(9), T));
fprintf(executeTP1, sprintf('output short\n'));
fprintf(executeTP1, sprintf('end\n'));
fprintf(executeTP1file, sprintf('1511\n'));

fclose(executeTP1);
fclose(executeTP1file);

% MATLAB calls Operating System to use Command: redirecting "<" input file
system('FCEA2.exe < 1512.inp');


% MATLAB opens the output file created by the execution of 1512.inp in
FEA2.exe
% 'rt' opens the file for reading in text mode
fid = fopen('1511.out', 'rt');

% Textscan looks into 1511.out to find all the data in particular pattern
% shown below:
data_cell = textscan(fid,'%s%s%f%f%f','delimiter',',','delimiter','\t',...
    'delimiter','\t', 'delimiter','\t');

fclose(fid);
dataTP = data_cell{1}
B = length(dataTP);
```

```
% Finding first value of output
locator = strfind(dataTP,'P, BAR');
index = find(~cellfun(@isempty, locator));
n       = B - index; % Number of CEA output lines to check

x_data = zeros(n,1);

% Extracting data from output file
for ii = 1:n
    dataSplit = split(dataTP{ii + index - 1});
    caseValues = dataSplit{end};
    blank = length(caseValues);

    if blank == 1
        caseValues = dataSplit{(end - 1)};
    end

    % Text of output. Used to see what data is being checked
    if ii <= 13
        caseNamesTP(ii, 1) = {strcat(dataSplit{1},{' '},dataSplit{2})};
    else
        caseNamesTP(ii, 1) = {dataSplit(1)};
    end

    checkPos = strfind(caseValues,'+');
    checkNeg = strfind(caseValues,'-');

    if isempty(checkPos) == 1 && isempty(checkNeg) == 1
        x_data(ii) = str2double(caseValues);

    elseif isempty(checkPos) == 0 && isempty(checkNeg) == 0 &&
length(checkNeg) == 2
        Z = [caseValues(1:6), 'e', caseValues(7:end)];
        x_data(ii) = str2double(Z);

    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 &&
length(checkNeg) == 2
        Z = [caseValues(1:6), 'e', caseValues(7:end)];
        x_data(ii) = str2double(Z);

    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 && checkNeg == 7
        Z = [caseValues(1:6), 'e', caseValues(7:end)];
        x_data(ii) = str2double(Z);

    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0
        x_data(ii) = str2double(caseValues);

    elseif isempty(checkPos) == 1 && isempty(checkNeg) == 0 && checkNeg == 1
        Z = [caseValues(1:6), 'e', caseValues(7:end)];
        x_data(ii) = str2double(Z);

    elseif isempty(checkPos) == 0 && isempty(checkNeg) == 1 && checkNeg == 7
        Z = [caseValues(1:6), 'e', caseValues(7:end)];
        x_data(ii) = str2double(Z);

    end

end
caseNamesTP = string(caseNamesTP);
for j = 1:length(caseNamesTP)
    if strcmp(caseNamesTP,'MOLE')
        caseNamesTP(j) = [];
    else
    end
end

H_i_TP   = find(strcmp(caseNamesTP, '*H'));
H2_i_TP  = find(strcmp(caseNamesTP, '*H2'));
H2O_i_TP = find(strcmp(caseNamesTP, 'H2O'));
NO_i_TP  = find(strcmp(caseNamesTP, '*NO'));
N2_i_TP  = find(strcmp(caseNamesTP, '*N2'));
N_i_TP   = find(strcmp(caseNamesTP, '*N'));
O_i_TP   = find(strcmp(caseNamesTP, '*O'));
OH_i_TP  = find(strcmp(caseNamesTP, '*OH'));
O2_i_TP  = find(strcmp(caseNamesTP, '*O2'));

moles_x   = {x_data(H_i_TP), x_data(H2_i_TP), x_data(H2O_i_TP),
x_data(NO_i_TP), x_data(N2_i_TP), x_data(N_i_TP), x_data(O_i_TP),
x_data(OH_i_TP), x_data(O2_i_TP)};

for i = 1:length(moles_x)
    if isempty(moles_x{i}) || isnan(moles_x{i})
        moles_x{i} = 0;
    end
end

moles_x = cell2mat(moles_x);
moles_x = compose("%.5f", moles_x);

x_data = x_data(~isnan(x_data));


delete *inp
delete *out

end
```

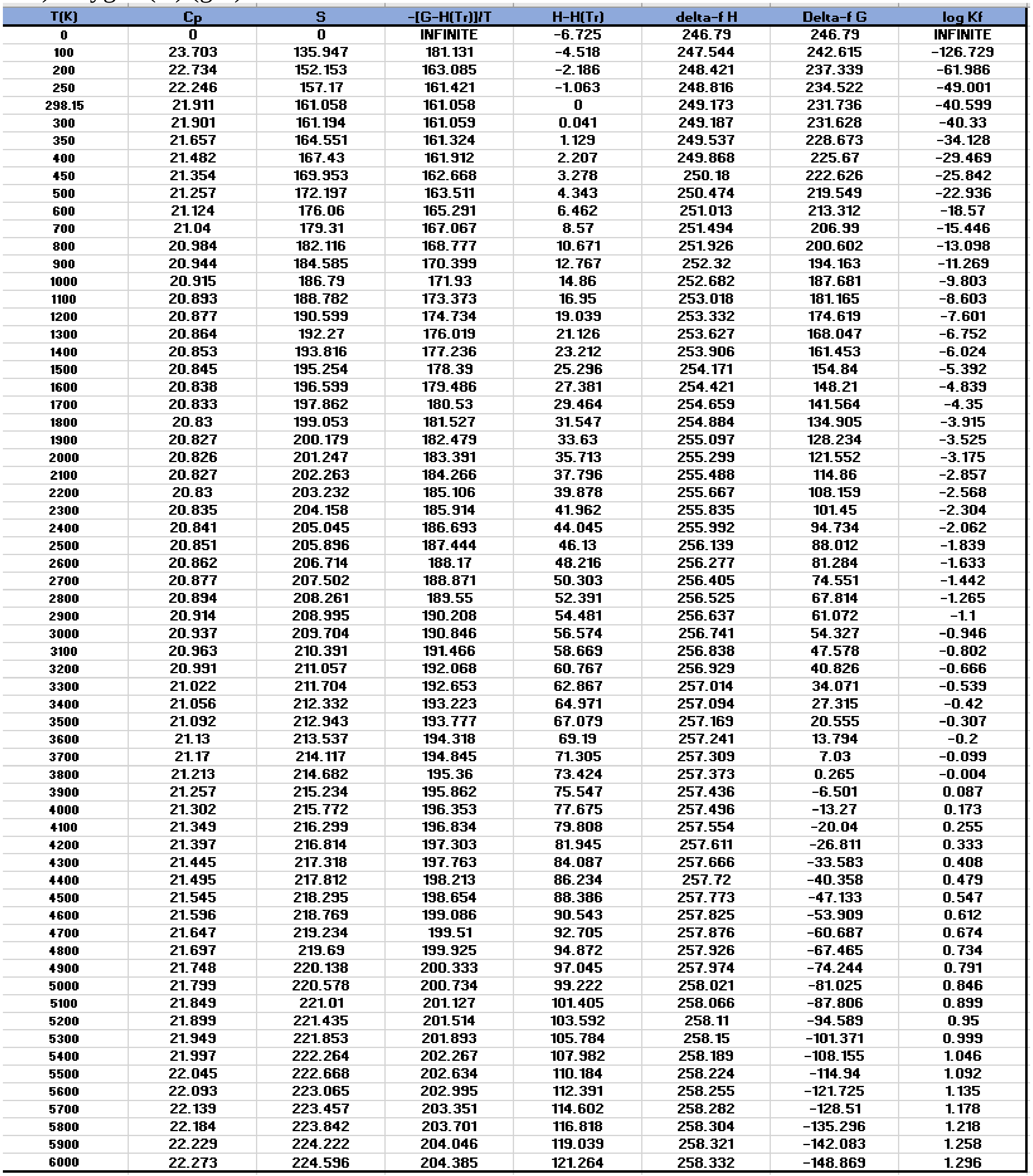
Appendix D: JANAF Tables for Species H_2, O_2, O, H, OH and H_2O

D.1) Hydrogen (H) (gas)



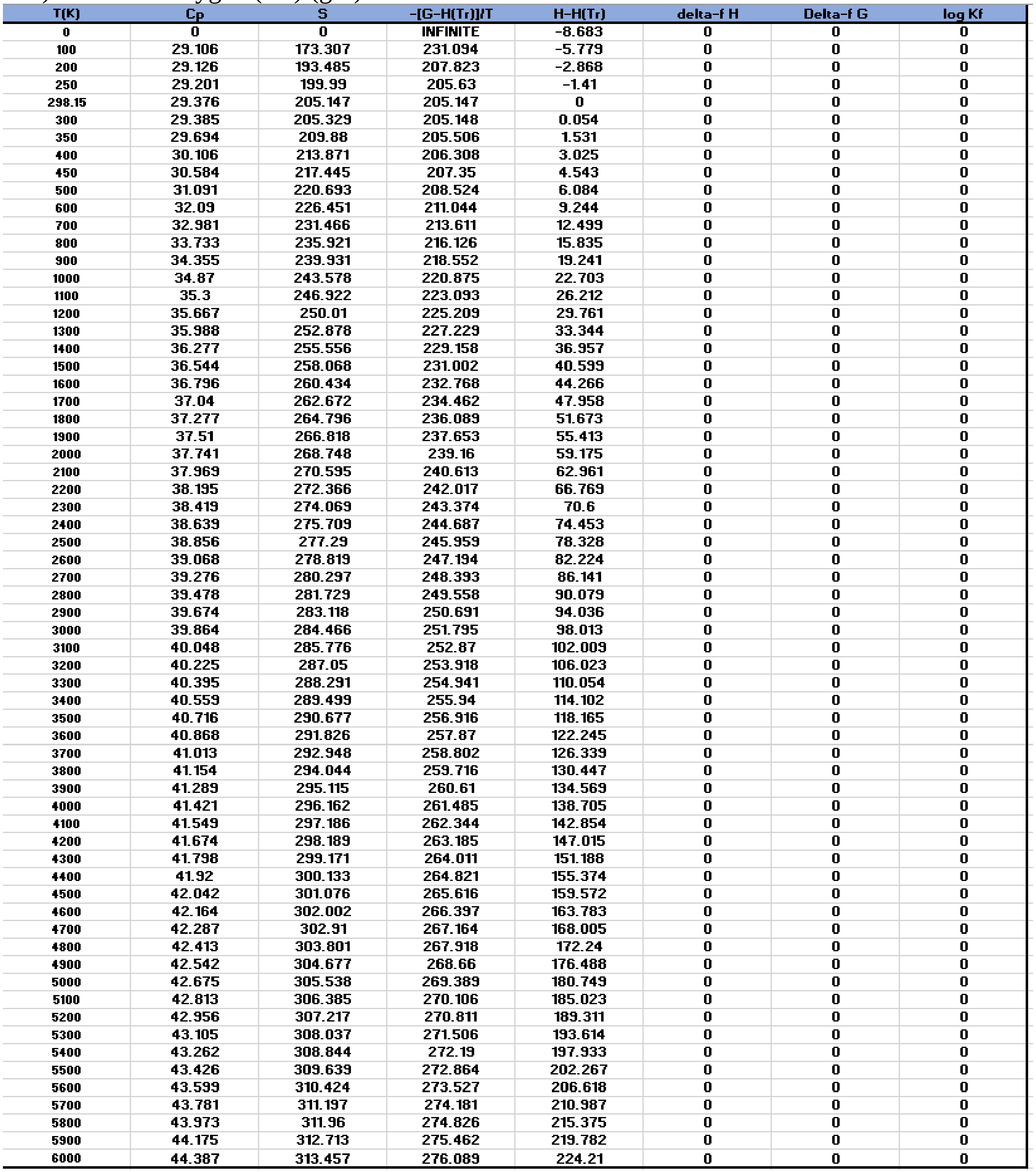
T(K)	Cp	S	-[G-H(Tr)]/T	H-H(Tr)	delta-f H	Delta-f G	log Kf
0	0	0	INFINITE	-6.197	216.035	216.035	INFINITE
100	20.786	92.009	133.197	-4.119	216.614	212.45	-110.972
200	20.786	106.417	116.618	-2.04	217.346	208.004	-54.325
250	20.786	111.055	115.059	-1.001	217.687	205.629	-42.964
298.15	20.786	114.716	114.716	0	217.999	203.278	-35.613
300	20.786	114.845	114.717	0.038	218.011	203.186	-35.378
350	20.786	118.049	114.97	1.078	218.326	200.69	-29.951
400	20.786	120.825	115.532	2.117	218.637	198.15	-25.876
450	20.786	123.273	116.259	3.156	218.946	195.57	-22.701
500	20.786	125.463	117.072	4.196	219.254	192.957	-20.158
600	20.786	129.253	118.796	6.274	219.868	187.64	-16.335
700	20.786	132.457	120.524	8.353	220.478	182.22	-13.597
800	20.786	135.232	122.193	10.431	221.08	176.713	-11.538
900	20.786	137.681	123.781	12.51	221.671	171.132	-9.932
1000	20.786	139.871	125.282	14.589	222.248	165.485	-8.644
1100	20.786	141.852	126.7	16.667	222.807	159.782	-7.587
1200	20.786	143.66	128.039	18.746	223.346	154.028	-6.705
1300	20.786	145.324	129.305	20.824	223.865	148.23	-5.956
1400	20.786	146.865	130.505	22.903	224.361	142.394	-5.313
1500	20.786	148.299	131.644	24.982	224.836	136.522	-4.754
1600	20.786	149.64	132.728	27.06	225.289	130.62	-4.264
1700	20.786	150.9	133.76	29.139	225.721	124.689	-3.831
1800	20.786	152.088	134.745	31.217	226.132	118.734	-3.446
1900	20.786	153.212	135.688	33.296	226.525	112.757	-3.1
2000	20.786	154.278	136.591	35.375	226.898	106.76	-2.788
2100	20.786	155.293	137.458	37.453	227.254	100.744	-2.506
2200	20.786	156.26	138.291	39.532	227.593	94.712	-2.249
2300	20.786	157.184	139.092	41.61	227.916	88.664	-2.014
2400	20.786	158.068	139.864	43.689	228.224	82.603	-1.798
2500	20.786	158.917	140.61	45.768	228.518	76.53	-1.599
2600	20.786	159.732	141.33	47.846	228.798	70.444	-1.415
2700	20.786	160.516	142.026	49.925	229.064	64.349	-1.245
2800	20.786	161.272	142.7	52.004	229.318	58.243	-1.087
2900	20.786	162.002	143.353	54.082	229.56	52.129	-0.939
3000	20.786	162.706	143.986	56.161	229.79	46.007	-0.801
3100	20.786	163.388	144.601	58.239	230.008	39.877	-0.672
3200	20.786	164.048	145.199	60.318	230.216	33.741	-0.551
3300	20.786	164.688	145.78	62.397	230.413	27.598	-0.437
3400	20.786	165.308	146.345	64.475	230.599	21.449	-0.33
3500	20.786	165.911	146.895	66.554	230.776	15.295	-0.228
3600	20.786	166.496	147.432	68.632	230.942	9.136	-0.133
3700	20.786	167.066	147.955	70.711	231.098	2.973	-0.042
3800	20.786	167.62	148.465	72.79	231.244	-3.195	0.044
3900	20.786	168.16	148.963	74.868	231.381	-9.366	0.125
4000	20.786	168.686	149.45	76.947	231.509	-15.541	0.203
4100	20.786	169.2	149.925	79.025	231.627	-21.718	0.277
4200	20.786	169.7	150.39	81.104	231.736	-27.899	0.347
4300	20.786	170.19	150.845	83.183	231.836	-34.082	0.414
4400	20.786	170.667	151.29	85.261	231.927	-40.267	0.478
4500	20.786	171.135	151.726	87.34	232.009	-46.454	0.539
4600	20.786	171.591	152.153	89.418	232.082	-52.643	0.598
4700	20.786	172.038	152.571	91.497	232.147	-58.834	0.654
4800	20.786	172.476	152.981	93.576	232.204	-65.025	0.708
4900	20.786	172.905	153.383	95.654	232.253	-71.218	0.759
5000	20.786	173.325	153.778	97.733	232.294	-77.412	0.809
5100	20.786	173.736	154.165	99.811	232.327	-83.606	0.856
5200	20.786	174.14	154.546	101.89	232.353	-89.801	0.902
5300	20.786	174.536	154.919	103.969	232.373	-95.997	0.946
5400	20.786	174.924	155.286	106.047	232.386	-102.192	0.989
5500	20.786	175.306	155.646	108.126	232.392	-108.389	1.029
5600	20.786	175.68	156.001	110.204	232.393	-114.584	1.069
5700	20.786	176.048	156.349	112.283	232.389	-120.78	1.107
5800	20.786	176.41	156.692	114.362	232.379	-126.976	1.144
5900	20.786	176.765	157.029	116.44	232.365	-133.172	1.179
6000	20.786	177.114	157.361	118.519	232.348	-139.368	1.213

D.2) Oxygen (O) (gas)



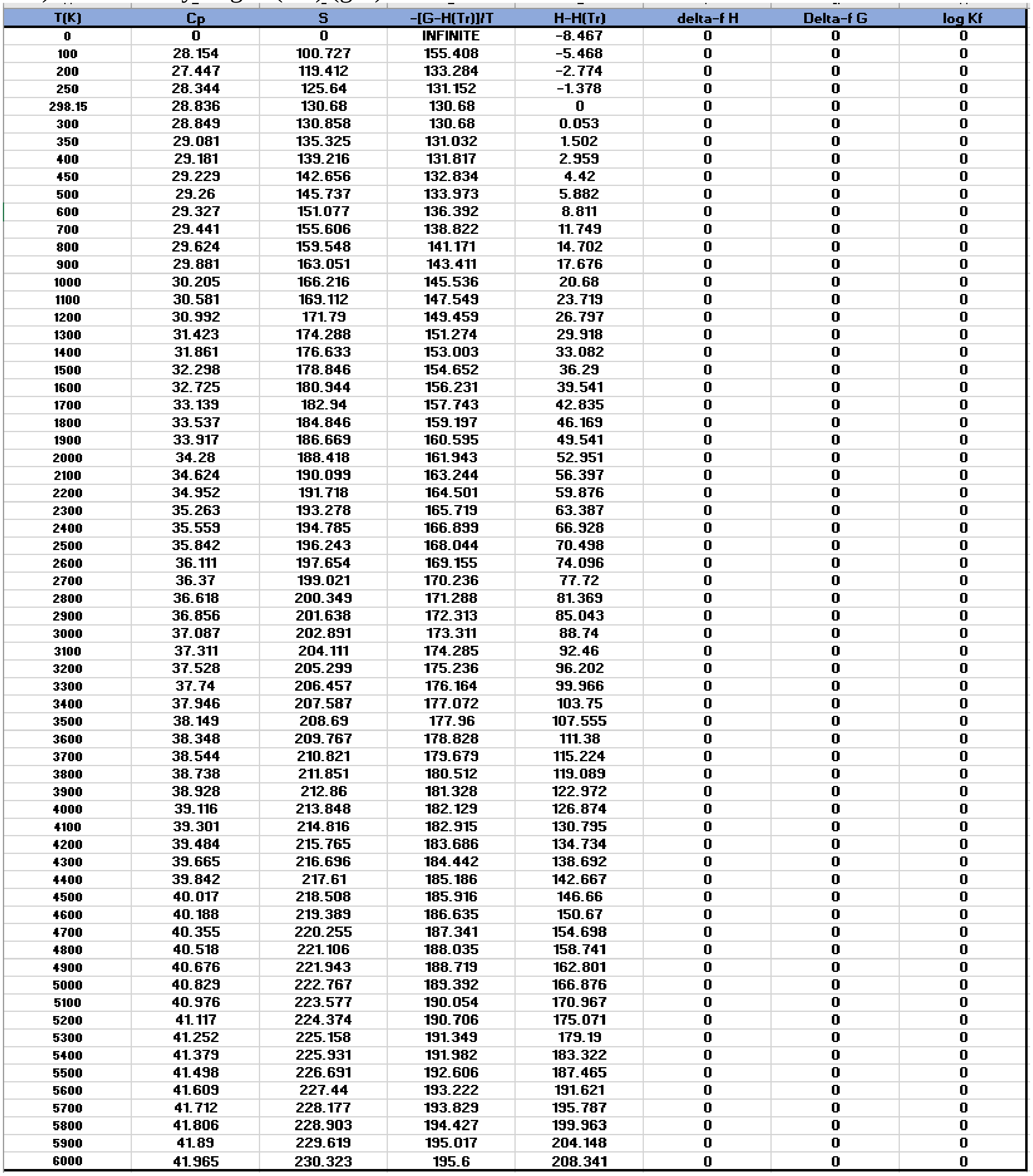
T(K)	Cp	S	-[G-H(Tr)]/T	H-H(Tr)	delta-f H	Delta-f G	log Kf
0	0	0	INFINITE	-6.725	246.79	246.79	INFINITE
100	23.703	135.947	181.131	-4.518	247.544	242.615	-126.729
200	22.734	152.153	163.085	-2.186	248.421	237.339	-61.986
250	22.246	157.17	161.421	-1.063	248.816	234.522	-49.001
298.15	21.911	161.058	161.058	0	249.173	231.736	-40.599
300	21.901	161.194	161.059	0.041	249.187	231.628	-40.33
350	21.657	164.551	161.324	1.129	249.537	228.673	-34.128
400	21.482	167.43	161.912	2.207	249.868	225.67	-29.469
450	21.354	169.953	162.668	3.278	250.18	222.626	-25.842
500	21.257	172.197	163.511	4.343	250.474	219.549	-22.936
600	21.124	176.06	165.291	6.462	251.013	213.312	-18.57
700	21.04	179.31	167.067	8.57	251.494	206.99	-15.446
800	20.984	182.116	168.777	10.671	251.926	200.602	-13.098
900	20.944	184.585	170.399	12.767	252.32	194.163	-11.269
1000	20.915	186.79	171.93	14.86	252.682	187.681	-9.803
1100	20.893	188.782	173.373	16.95	253.018	181.165	-8.603
1200	20.877	190.599	174.734	19.039	253.332	174.619	-7.601
1300	20.864	192.27	176.019	21.126	253.627	168.047	-6.752
1400	20.853	193.816	177.236	23.212	253.906	161.453	-6.024
1500	20.845	195.254	178.39	25.296	254.171	154.84	-5.392
1600	20.838	196.599	179.486	27.381	254.421	148.21	-4.839
1700	20.833	197.862	180.53	29.464	254.659	141.564	-4.35
1800	20.83	199.053	181.527	31.547	254.884	134.905	-3.915
1900	20.827	200.179	182.479	33.63	255.097	128.234	-3.525
2000	20.826	201.247	183.391	35.713	255.299	121.552	-3.175
2100	20.827	202.263	184.266	37.796	255.488	114.86	-2.857
2200	20.83	203.232	185.106	39.878	255.667	108.159	-2.568
2300	20.835	204.158	185.914	41.962	255.835	101.45	-2.304
2400	20.841	205.045	186.693	44.045	255.992	94.734	-2.062
2500	20.851	205.896	187.444	46.13	256.139	88.012	-1.839
2600	20.862	206.714	188.17	48.216	256.277	81.284	-1.633
2700	20.877	207.502	188.871	50.303	256.405	74.551	-1.442
2800	20.894	208.261	189.55	52.391	256.525	67.814	-1.265
2900	20.914	208.995	190.208	54.481	256.637	61.072	-1.1
3000	20.937	209.704	190.846	56.574	256.741	54.327	-0.946
3100	20.963	210.391	191.466	58.669	256.838	47.578	-0.802
3200	20.991	211.057	192.068	60.767	256.929	40.826	-0.666
3300	21.022	211.704	192.653	62.867	257.014	34.071	-0.539
3400	21.056	212.332	193.223	64.971	257.094	27.315	-0.42
3500	21.092	212.943	193.777	67.079	257.169	20.555	-0.307
3600	21.13	213.537	194.318	69.19	257.241	13.794	-0.2
3700	21.17	214.117	194.845	71.305	257.309	7.03	-0.099
3800	21.213	214.682	195.36	73.424	257.373	0.265	-0.004
3900	21.257	215.234	195.862	75.547	257.436	-6.501	0.087
4000	21.302	215.772	196.353	77.675	257.496	-13.27	0.173
4100	21.349	216.299	196.834	79.808	257.554	-20.04	0.255
4200	21.397	216.814	197.303	81.945	257.611	-26.811	0.333
4300	21.445	217.318	197.763	84.087	257.666	-33.583	0.408
4400	21.495	217.812	198.213	86.234	257.72	-40.358	0.479
4500	21.545	218.295	198.654	88.386	257.773	-47.133	0.547
4600	21.596	218.769	199.086	90.543	257.825	-53.909	0.612
4700	21.647	219.234	199.51	92.705	257.876	-60.687	0.674
4800	21.697	219.69	199.925	94.872	257.926	-67.465	0.734
4900	21.748	220.138	200.333	97.045	257.974	-74.244	0.791
5000	21.799	220.578	200.734	99.222	258.021	-81.025	0.846
5100	21.849	221.01	201.127	101.405	258.066	-87.806	0.899
5200	21.899	221.435	201.514	103.592	258.11	-94.589	0.95
5300	21.949	221.853	201.893	105.784	258.15	-101.371	0.999
5400	21.997	222.264	202.267	107.982	258.189	-108.155	1.046
5500	22.045	222.668	202.634	110.184	258.224	-114.94	1.092
5600	22.093	223.065	202.995	112.391	258.255	-121.725	1.135
5700	22.139	223.457	203.351	114.602	258.282	-128.51	1.178
5800	22.184	223.842	203.701	116.818	258.304	-135.296	1.218
5900	22.229	224.222	204.046	119.039	258.321	-142.083	1.258
6000	22.273	224.596	204.385	121.264	258.332	-148.869	1.296

D.3) Diatomic Oxygen (O2) (gas)



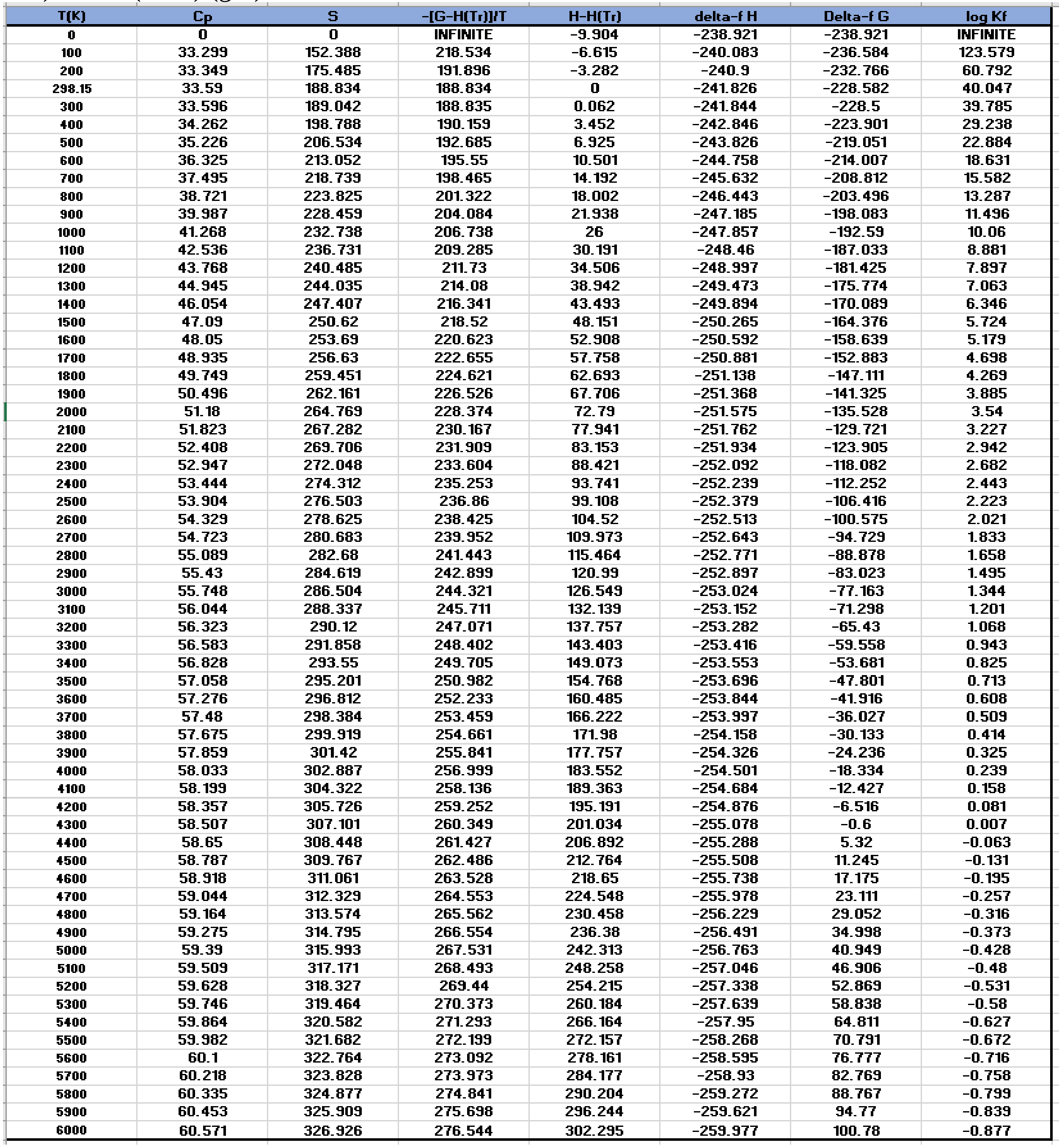
T(K)	Cp	S	-[G-H(Tr)]/T	H-H(Tr)	delta-f H	Delta-f G	log Kf
0	0	0	INFINITE	-8.683	0	0	0
100	29.106	173.307	231.094	-5.779	0	0	0
200	29.126	193.485	207.823	-2.868	0	0	0
250	29.201	199.99	205.63	-1.41	0	0	0
298.15	29.376	205.147	205.147	0	0	0	0
300	29.385	205.329	205.148	0.054	0	0	0
350	29.694	209.88	205.506	1.531	0	0	0
400	30.106	213.871	206.308	3.025	0	0	0
450	30.584	217.445	207.35	4.543	0	0	0
500	31.091	220.693	208.524	6.084	0	0	0
600	32.09	226.451	211.044	9.244	0	0	0
700	32.981	231.466	213.611	12.499	0	0	0
800	33.733	235.921	216.126	15.835	0	0	0
900	34.355	239.931	218.552	19.241	0	0	0
1000	34.87	243.578	220.875	22.703	0	0	0
1100	35.3	246.922	223.093	26.212	0	0	0
1200	35.667	250.01	225.209	29.761	0	0	0
1300	35.988	252.878	227.229	33.344	0	0	0
1400	36.277	255.556	229.158	36.957	0	0	0
1500	36.544	258.068	231.002	40.599	0	0	0
1600	36.796	260.434	232.768	44.266	0	0	0
1700	37.04	262.672	234.462	47.958	0	0	0
1800	37.277	264.796	236.089	51.673	0	0	0
1900	37.51	266.818	237.653	55.413	0	0	0
2000	37.741	268.748	239.16	59.175	0	0	0
2100	37.969	270.595	240.613	62.961	0	0	0
2200	38.195	272.366	242.017	66.769	0	0	0
2300	38.419	274.069	243.374	70.6	0	0	0
2400	38.639	275.709	244.687	74.453	0	0	0
2500	38.856	277.29	245.959	78.328	0	0	0
2600	39.068	278.819	247.194	82.224	0	0	0
2700	39.276	280.297	248.393	86.141	0	0	0
2800	39.478	281.729	249.558	90.079	0	0	0
2900	39.674	283.118	250.691	94.036	0	0	0
3000	39.864	284.466	251.795	98.013	0	0	0
3100	40.048	285.776	252.87	102.009	0	0	0
3200	40.225	287.05	253.918	106.023	0	0	0
3300	40.395	288.291	254.941	110.054	0	0	0
3400	40.559	289.499	255.94	114.102	0	0	0
3500	40.716	290.677	256.916	118.165	0	0	0
3600	40.868	291.826	257.87	122.245	0	0	0
3700	41.013	292.948	258.802	126.339	0	0	0
3800	41.154	294.044	259.716	130.447	0	0	0
3900	41.289	295.115	260.61	134.569	0	0	0
4000	41.421	296.162	261.485	138.705	0	0	0
4100	41.549	297.186	262.344	142.854	0	0	0
4200	41.674	298.189	263.185	147.015	0	0	0
4300	41.798	299.171	264.011	151.188	0	0	0
4400	41.92	300.133	264.821	155.374	0	0	0
4500	42.042	301.076	265.616	159.572	0	0	0
4600	42.164	302.002	266.397	163.783	0	0	0
4700	42.287	302.91	267.164	168.005	0	0	0
4800	42.413	303.801	267.918	172.24	0	0	0
4900	42.542	304.677	268.66	176.488	0	0	0
5000	42.675	305.538	269.389	180.749	0	0	0
5100	42.813	306.385	270.106	185.023	0	0	0
5200	42.956	307.217	270.811	189.311	0	0	0
5300	43.105	308.037	271.506	193.614	0	0	0
5400	43.262	308.844	272.19	197.933	0	0	0
5500	43.426	309.639	272.864	202.267	0	0	0
5600	43.599	310.424	273.527	206.618	0	0	0
5700	43.781	311.197	274.181	210.987	0	0	0
5800	43.973	311.96	274.826	215.375	0	0	0
5900	44.175	312.713	275.462	219.782	0	0	0
6000	44.387	313.457	276.089	224.21	0	0	0

D.4) Diatomic Hydrogen (H2) (gas)



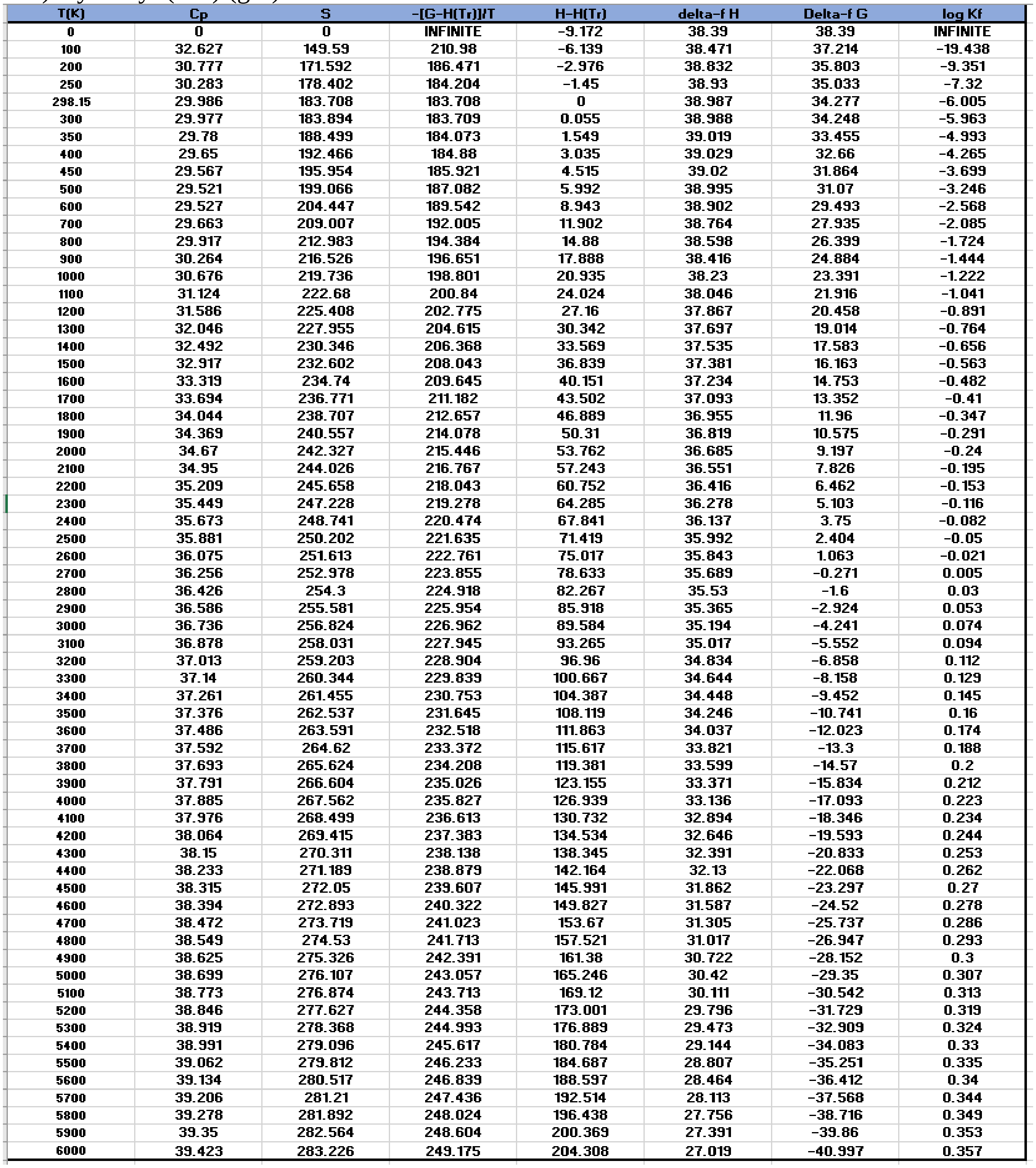
T(K)	Cp	S	-[G-H(Tr)]/T	H-H(Tr)	delta-f H	Delta-f G	log Kf
0	0	0	INFINITE	-8.467	0	0	0
100	28.154	100.727	155.408	-5.468	0	0	0
200	27.447	119.412	133.284	-2.774	0	0	0
250	28.344	125.64	131.152	-1.378	0	0	0
298.15	28.836	130.68	130.68	0	0	0	0
300	28.849	130.858	130.68	0.053	0	0	0
350	29.081	135.325	131.032	1.502	0	0	0
400	29.181	139.216	131.817	2.959	0	0	0
450	29.229	142.656	132.834	4.42	0	0	0
500	29.26	145.737	133.973	5.882	0	0	0
600	29.327	151.077	136.392	8.811	0	0	0
700	29.441	155.606	138.822	11.749	0	0	0
800	29.624	159.548	141.171	14.702	0	0	0
900	29.881	163.051	143.411	17.676	0	0	0
1000	30.205	166.216	145.536	20.68	0	0	0
1100	30.581	169.112	147.549	23.719	0	0	0
1200	30.992	171.79	149.459	26.797	0	0	0
1300	31.423	174.288	151.274	29.918	0	0	0
1400	31.861	176.633	153.003	33.082	0	0	0
1500	32.298	178.846	154.652	36.29	0	0	0
1600	32.725	180.944	156.231	39.541	0	0	0
1700	33.139	182.94	157.743	42.835	0	0	0
1800	33.537	184.846	159.197	46.169	0	0	0
1900	33.917	186.669	160.595	49.541	0	0	0
2000	34.28	188.418	161.943	52.951	0	0	0
2100	34.624	190.099	163.244	56.397	0	0	0
2200	34.952	191.718	164.501	59.876	0	0	0
2300	35.263	193.278	165.719	63.387	0	0	0
2400	35.559	194.785	166.899	66.928	0	0	0
2500	35.842	196.243	168.044	70.498	0	0	0
2600	36.111	197.654	169.155	74.096	0	0	0
2700	36.37	199.021	170.236	77.72	0	0	0
2800	36.618	200.349	171.288	81.369	0	0	0
2900	36.856	201.638	172.313	85.043	0	0	0
3000	37.087	202.891	173.311	88.74	0	0	0
3100	37.311	204.111	174.285	92.46	0	0	0
3200	37.528	205.299	175.236	96.202	0	0	0
3300	37.74	206.457	176.164	99.966	0	0	0
3400	37.946	207.587	177.072	103.75	0	0	0
3500	38.149	208.69	177.96	107.555	0	0	0
3600	38.348	209.767	178.828	111.38	0	0	0
3700	38.544	210.821	179.679	115.224	0	0	0
3800	38.738	211.851	180.512	119.089	0	0	0
3900	38.928	212.86	181.328	122.972	0	0	0
4000	39.116	213.848	182.129	126.874	0	0	0
4100	39.301	214.816	182.915	130.795	0	0	0
4200	39.484	215.765	183.686	134.734	0	0	0
4300	39.665	216.696	184.442	138.692	0	0	0
4400	39.842	217.61	185.186	142.667	0	0	0
4500	40.017	218.508	185.916	146.66	0	0	0
4600	40.188	219.389	186.635	150.67	0	0	0
4700	40.355	220.255	187.341	154.698	0	0	0
4800	40.518	221.106	188.035	158.741	0	0	0
4900	40.676	221.943	188.719	162.801	0	0	0
5000	40.829	222.767	189.392	166.876	0	0	0
5100	40.976	223.577	190.054	170.967	0	0	0
5200	41.117	224.374	190.706	175.071	0	0	0
5300	41.252	225.158	191.349	179.19	0	0	0
5400	41.379	225.931	191.982	183.322	0	0	0
5500	41.498	226.691	192.606	187.465	0	0	0
5600	41.609	227.44	193.222	191.621	0	0	0
5700	41.712	228.177	193.829	195.787	0	0	0
5800	41.806	228.903	194.427	199.963	0	0	0
5900	41.89	229.619	195.017	204.148	0	0	0
6000	41.965	230.323	195.6	208.341	0	0	0

D.5) Water (H2O) (gas)



T(K)	Cp	S	-[G-H(Tr)]/T	H-H(Tr)	delta-f H	Delta-f G	log Kf
0	0	0	INFINITE	-9.904	-238.921	-238.921	INFINITE
100	33.299	152.388	218.534	-6.615	-240.083	-236.584	123.579
200	33.349	175.485	191.896	-3.282	-240.9	-232.766	60.792
298.15	33.59	188.834	188.834	0	-241.826	-228.582	40.047
300	33.596	189.042	188.835	0.062	-241.844	-228.5	39.785
400	34.262	198.788	190.159	3.452	-242.846	-223.901	29.238
500	35.226	206.534	192.685	6.925	-243.826	-219.051	22.884
600	36.325	213.052	195.55	10.501	-244.758	-214.007	18.631
700	37.495	218.739	198.465	14.192	-245.632	-208.812	15.582
800	38.721	223.825	201.322	18.002	-246.443	-203.496	13.287
900	39.987	228.459	204.084	21.938	-247.185	-198.083	11.496
1000	41.268	232.738	206.738	26	-247.857	-192.59	10.06
1100	42.536	236.731	209.285	30.191	-248.46	-187.033	8.881
1200	43.768	240.485	211.73	34.506	-248.997	-181.425	7.897
1300	44.945	244.035	214.08	38.942	-249.473	-175.774	7.063
1400	46.054	247.407	216.341	43.493	-249.894	-170.089	6.346
1500	47.09	250.62	218.52	48.151	-250.265	-164.376	5.724
1600	48.05	253.69	220.623	52.908	-250.592	-158.639	5.179
1700	48.935	256.63	222.655	57.758	-250.881	-152.883	4.698
1800	49.749	259.451	224.621	62.693	-251.138	-147.111	4.269
1900	50.496	262.161	226.526	67.706	-251.368	-141.325	3.885
2000	51.18	264.769	228.374	72.79	-251.575	-135.528	3.54
2100	51.823	267.282	230.167	77.941	-251.762	-129.721	3.227
2200	52.408	269.706	231.909	83.153	-251.934	-123.905	2.942
2300	52.947	272.048	233.604	88.421	-252.092	-118.082	2.682
2400	53.444	274.312	235.253	93.741	-252.239	-112.252	2.443
2500	53.904	276.503	236.86	99.108	-252.379	-106.416	2.223
2600	54.329	278.625	238.425	104.52	-252.513	-100.575	2.021
2700	54.723	280.683	239.952	109.973	-252.643	-94.729	1.833
2800	55.089	282.68	241.443	115.464	-252.771	-88.878	1.658
2900	55.43	284.619	242.899	120.99	-252.897	-83.023	1.495
3000	55.748	286.504	244.321	126.549	-253.024	-77.163	1.344
3100	56.044	288.337	245.711	132.139	-253.152	-71.298	1.201
3200	56.323	290.12	247.071	137.757	-253.282	-65.43	1.068
3300	56.583	291.858	248.402	143.403	-253.416	-59.558	0.943
3400	56.828	293.55	249.705	149.073	-253.553	-53.681	0.825
3500	57.058	295.201	250.982	154.768	-253.696	-47.801	0.713
3600	57.276	296.812	252.233	160.485	-253.844	-41.916	0.608
3700	57.48	298.384	253.459	166.222	-253.997	-36.027	0.509
3800	57.675	299.919	254.661	171.98	-254.158	-30.133	0.414
3900	57.859	301.42	255.841	177.757	-254.326	-24.236	0.325
4000	58.033	302.887	256.999	183.552	-254.501	-18.334	0.239
4100	58.199	304.322	258.136	189.363	-254.684	-12.427	0.158
4200	58.357	305.726	259.252	195.191	-254.876	-6.516	0.081
4300	58.507	307.101	260.349	201.034	-255.078	-0.6	0.007
4400	58.65	308.448	261.427	206.892	-255.288	5.32	-0.063
4500	58.787	309.767	262.486	212.764	-255.508	11.245	-0.131
4600	58.918	311.061	263.528	218.65	-255.738	17.175	-0.195
4700	59.044	312.329	264.553	224.548	-255.978	23.111	-0.257
4800	59.164	313.574	265.562	230.458	-256.229	29.052	-0.316
4900	59.275	314.795	266.554	236.38	-256.491	34.998	-0.373
5000	59.39	315.993	267.531	242.313	-256.763	40.949	-0.428
5100	59.509	317.171	268.493	248.258	-257.046	46.906	-0.48
5200	59.628	318.327	269.44	254.215	-257.338	52.869	-0.531
5300	59.746	319.464	270.373	260.184	-257.639	58.838	-0.58
5400	59.864	320.582	271.293	266.164	-257.95	64.811	-0.627
5500	59.982	321.682	272.199	272.157	-258.268	70.791	-0.672
5600	60.1	322.764	273.092	278.161	-258.595	76.777	-0.716
5700	60.218	323.828	273.973	284.177	-258.93	82.769	-0.758
5800	60.335	324.877	274.841	290.204	-259.272	88.767	-0.799
5900	60.453	325.909	275.698	296.244	-259.621	94.77	-0.839
6000	60.571	326.926	276.544	302.295	-259.977	100.78	-0.877

D.6) Hydroxyl (OH) (gas)



T(K)	Cp	S	-[G-H(Tr)]/T	H-H(Tr)	delta-f H	Delta-f G	log Kf
0	0	0	INFINITE	-9.172	38.39	38.39	INFINITE
100	32.627	149.59	210.98	-6.139	38.471	37.214	-19.438
200	30.777	171.592	186.471	-2.976	38.832	35.803	-9.351
250	30.283	178.402	184.204	-1.45	38.93	35.033	-7.32
298.15	29.986	183.708	183.708	0	38.987	34.277	-6.005
300	29.977	183.894	183.709	0.055	38.988	34.248	-5.963
350	29.78	188.499	184.073	1.549	39.019	33.455	-4.993
400	29.65	192.466	184.88	3.035	39.029	32.66	-4.265
450	29.567	195.954	185.921	4.515	39.02	31.864	-3.699
500	29.521	199.066	187.082	5.992	38.995	31.07	-3.246
600	29.527	204.447	189.542	8.943	38.902	29.493	-2.568
700	29.663	209.007	192.005	11.902	38.764	27.935	-2.085
800	29.917	212.983	194.384	14.88	38.598	26.399	-1.724
900	30.264	216.526	196.651	17.888	38.416	24.884	-1.444
1000	30.676	219.736	198.801	20.935	38.23	23.391	-1.222
1100	31.124	222.68	200.84	24.024	38.046	21.916	-1.041
1200	31.586	225.408	202.775	27.16	37.867	20.458	-0.891
1300	32.046	227.955	204.615	30.342	37.697	19.014	-0.764
1400	32.492	230.346	206.368	33.569	37.535	17.583	-0.656
1500	32.917	232.602	208.043	36.839	37.381	16.163	-0.563
1600	33.319	234.74	209.645	40.151	37.234	14.753	-0.482
1700	33.694	236.771	211.182	43.502	37.093	13.352	-0.41
1800	34.044	238.707	212.657	46.889	36.955	11.96	-0.347
1900	34.369	240.557	214.078	50.31	36.819	10.575	-0.291
2000	34.67	242.327	215.446	53.762	36.685	9.197	-0.24
2100	34.95	244.026	216.767	57.243	36.551	7.826	-0.195
2200	35.209	245.658	218.043	60.752	36.416	6.462	-0.153
2300	35.449	247.228	219.278	64.285	36.278	5.103	-0.116
2400	35.673	248.741	220.474	67.841	36.137	3.75	-0.082
2500	35.881	250.202	221.635	71.419	35.992	2.404	-0.05
2600	36.075	251.613	222.761	75.017	35.843	1.063	-0.021
2700	36.256	252.978	223.855	78.633	35.689	-0.271	0.005
2800	36.426	254.3	224.918	82.267	35.53	-1.6	0.03
2900	36.586	255.581	225.954	85.918	35.365	-2.924	0.053
3000	36.736	256.824	226.962	89.584	35.194	-4.241	0.074
3100	36.878	258.031	227.945	93.265	35.017	-5.552	0.094
3200	37.013	259.203	228.904	96.96	34.834	-6.858	0.112
3300	37.14	260.344	229.839	100.667	34.644	-8.158	0.129
3400	37.261	261.455	230.753	104.387	34.448	-9.452	0.145
3500	37.376	262.537	231.645	108.119	34.246	-10.741	0.16
3600	37.486	263.591	232.518	111.863	34.037	-12.023	0.174
3700	37.592	264.62	233.372	115.617	33.821	-13.3	0.188
3800	37.693	265.624	234.208	119.381	33.599	-14.57	0.2
3900	37.791	266.604	235.026	123.155	33.371	-15.834	0.212
4000	37.885	267.562	235.827	126.939	33.136	-17.093	0.223
4100	37.976	268.499	236.613	130.732	32.894	-18.346	0.234
4200	38.064	269.415	237.383	134.534	32.646	-19.593	0.244
4300	38.15	270.311	238.138	138.345	32.391	-20.833	0.253
4400	38.233	271.189	238.879	142.164	32.13	-22.068	0.262
4500	38.315	272.05	239.607	145.991	31.862	-23.297	0.27
4600	38.394	272.893	240.322	149.827	31.587	-24.52	0.278
4700	38.472	273.719	241.023	153.67	31.305	-25.737	0.286
4800	38.549	274.53	241.713	157.521	31.017	-26.947	0.293
4900	38.625	275.326	242.391	161.38	30.722	-28.152	0.3
5000	38.699	276.107	243.057	165.246	30.42	-29.35	0.307
5100	38.773	276.874	243.713	169.12	30.111	-30.542	0.313
5200	38.846	277.627	244.358	173.001	29.796	-31.729	0.319
5300	38.919	278.368	244.993	176.889	29.473	-32.909	0.324
5400	38.991	279.096	245.617	180.784	29.144	-34.083	0.33
5500	39.062	279.812	246.233	184.687	28.807	-35.251	0.335
5600	39.134	280.517	246.839	188.597	28.464	-36.412	0.34
5700	39.206	281.21	247.436	192.514	28.113	-37.568	0.344
5800	39.278	281.892	248.024	196.438	27.756	-38.716	0.349
5900	39.35	282.564	248.604	200.369	27.391	-39.86	0.353
6000	39.423	283.226	249.175	204.308	27.019	-40.997	0.357

Appendix E: Thermochemical Equilibrium Nozzle Design Code

```
%% 2D TCE Method of Characteristics
%% Code created by Michael Sandoval
close all; clear; clc; fclose('all');

%% Initial data CEA rocket problem in equilibrium
P_0c     = 10;              % [atm] Chamber pressure
ExitType = "AeAt";          % Exit type for CEA
A        = 2.5;             % Exit type entry for CEA
OF       = 34.29623;        % Mixture ratio
Ox       = "Air";           % Oxidizer type
T_Ox     = 600;             % [K] Oxidizer temperature
Fuel     = "H2";            % Fuel type
T_F      = 298.15;          % {K] Fuel temperature

% Species being considered for TCE calculation
species = ["H", "H2", "H2O", "NO", "N2", "N", "O", "OH", "O2"];

% FUNCTION: CEA(Rkt) Problem
[CD, TD, ED, moles_Throat, caseNamesRkt, dataRkt] = ceaRkt(ExitType, A, OF,
P_0c, Ox, T_Ox, Fuel, T_F);

%% Data from CEA Rkt are the initial conditions to calculate MoC in CPG
% This data will be used as the first iteration for every point in the
flowfield
gamma = TD(12);             % Throat gamma
M_e   = ED(14);             % Exit mach number
Tc    = CD(2);              % [K] Chamber temperature
Pc    = P_0c;               % [atm] Chamber pressure
rc    = CD(3);              % [kg/m^3] Chamber density
a_f   = TD(13);             % [m/s^2] Frozen speed of sound at throat
H_t   = CD(4)*1000;         % [J/kg] Total enthalpy (same everywhere)
Throat_Height = 0.0508;     % [m] Throat radius
C_Lines = 10;               % Number of characteristic lines

% FUNCTION: MoC_Analysis (CPG)
[Wall_Points, x, y, Theta_x, Mu_x, Nu_x, K_p, K_m, M_x, P_x, T_x, r_x, V_x] =
MoC_Analysis(M_e, gamma, Tc, Pc, rc, a_f, C_Lines, Throat_Height);

% FUNCTION: CEA(T,P) Problem
[n, x_data, moles_x, caseNamesTP] = ceaTP(T_x(1,1), P_x(1,1), species,
moles_Throat);

% Throat conditions
y_a      = Throat_Height;  % [m] Throat height
x_a      = 0;              % [m] Throat x coordinate
P_a     = TD(1)*0.986923;  % [Bar to atm] Throat pressure
rho_a   = TD(3);           % [kg/m^3] Throat density
V_a     = TD(19);          % [m/s] Throat velocity
Mu_a    = 89;              % [deg] Wave angle (not sure if this is a good
assumption)
Theta_a = 0;               % [deg]
```

```
% Universial gas constant
R_u = 8314;               % [J/kg*mol*k]

%% Point 2 iteration
k = 1;              % Index for cells
iterations = 100;    % Number of iterations

% Formating data for iterations
T       = cell(1,1,iterations);
P       = cell(1,1,iterations);
V       = cell(1,1,iterations);
Mu      = cell(1,1,iterations);
rho     = cell(1,1,iterations);
h       = cell(1,1,iterations);
Theta   = cell(1,1,iterations);
moles   = cell(1,1,iterations);
x_point = cell(1,1,iterations);
y_point = cell(1,1,iterations);
ratio_1 = cell(1,1,iterations);
ratio_2 = cell(1,1,iterations);

% First iteration initial conditions
P(1)       = {P_x(1,2)};
T(1)       = {T_x(1,1)};
V(1)       = {V_x(1,1)};
Mu(1)      = {Mu_x(1,1)};
rho(1)     = {r_x(1,1)};
h(1)       = {x_data(4)*1000};
Theta(1)   = {Theta_x(1,1)};
moles(1)   = {moles_x};
x_point(1) = {0};
y_point(1) = {0};

while k < iterations+1

% Slopes for x and y equations
N_Slope = tand( 0.5*(Theta{1,1,k} + Theta_a - Mu{1,1,k} - Mu_a) );
P_Slope = tand( 0.5*(Theta{1,1,k} + Theta_x(1,1) + Mu{1,1,k} + Mu_x(1,1)) );

% Next iteration [x,y] coordinates for point 2
x_point{1,1,k+1} = ( y_a - y(1,1) - x_a*N_Slope + x(1,1)*P_Slope )/( P_Slope
- N_Slope );
y_point{1,1,k+1} = y(1,1) + (x_point{1,1,k} - x(1,1))*P_Slope;

% Values for pressure equation
A1 = 0.5*( (rho{1,1,k}*(V{1,1,k}^2)*tand(Mu{1,1,k}) ) + (
r_x(1,1)*(V_x(1,1)^2)*tand(Mu_x(1,1)) ) );
AA = 0.5*( (rho{1,1,k}*(V{1,1,k}^2)*tand(Mu{1,1,k}) ) + (
rho_a*(V_a^2)*tand(Mu_a) ) );

% Next iteration of pressure and theta for point 2
P{1,1,k+1}     = (Theta_x(1,1) - Theta_a + (101325*P_x(1,1)/A1) +
(101325*P_a/AA))/( (1/A1) + (1/AA) );
Theta{1,1,k+1} = Theta_x(1,1) - ( (P{1,1,k+1} - 101325*P_x(1,1))/A1 );
```

```
% Prime values, linear interpolation of conditions between point a and point
1
x_prime   = 0.5*(x(1,1)+x_a);
y_prime   = y(1,1)   + (x_prime - x(1,1))*( (y_a - y(1,1))/(x_a - x(1,1)) );
V_prime   = V_x(1,1) + (x_prime - x(1,1))*( (V_a - V_x(1,1))/(x_a - x(1,1))
);
rho_prime = r_x(1,1) + (x_prime - x(1,1))*( (rho_a - r_x(1,1))/(x_a - x(1,1))
);
P_prime   = 101325*(P_x(1,1) + (x_prime - x(1,1))*( (P_a - P_x(1,1))/(x_a -
x(1,1)) ));

% Average velocity and density between point 2' and 2
V_avg   = 0.5*(V{1,1,k} + V_prime);
rho_avg = 0.5*(rho{1,1,k} + rho_prime);

% Calculation of velocity at point 2
V{1,1,k+1} = V_prime - ( (P{1,1,k+1} - P_prime)/(V_avg*rho_avg) );
% V{1,1,k+1} = (V_prime - ( (P{k+1} - P_prime)/(rho_avg) ))/V_avg;
% V{1,1,k+1} = sqrt(2*(H_t - h{1,1,k}));

% Velocity components and interpolated velocity for point 2'
v_2 = V{1,1,k+1}*sind(Theta{1,1,k+1});
v_1 = V_x(1,1)*sind(Theta_x(1,1));
v_a = V_a*sind(Theta_a);
v_p = v_1 + (x_prime - x(1,1))*( (v_a-v_1)/(x_a-x(1,1)) );

u_2 = V{1,1,k+1}*cosd(Theta{1,1,k+1});
u_1 = V_x(1,1)*cosd(Theta_x(1,1));
u_a = V_a*cosd(Theta_a);
u_p = u_1 + (x_prime - x(1,1))*( (u_a-u_1)/(x_a-x(1,1)) );

% Average velocity between point 2' and 2
v2_avg = 0.5*(v_2 + v_p);
u2_avg = 0.5*(u_2 + u_p);

ratio_1{1,1,k} = v2_avg/u2_avg;
ratio_2{1,1,k} = (y_point{1,1,k} - y_prime)/(x_point{1,1,k}-x_prime);

% New data
Cp           = x_data(11)*1000;
T{1,1,k+1}   = T_x(1,1) + ( (V_x(1,1)^2) - (V{1,1,k+1}^2) )/(2*Cp);
rho{1,1,k+1} = (P{1,1,k+1}*x_data(8))/(R_u*T{1,1,k+1});
Mu{1,1,k+1}  = asind(a_f/V{1,1,k+1});

P{1,1,k+1} = P{1,1,k+1}*9.86923e-6;
% [x_data, moles_x, caseNamesTP] = ceaTP(T{1,1,k+1}, P{1,1,k+1}, species,
moles_x{1,1,k});
% h{1,1,k+1} = x_data(4)*1000;
% rho{1,1,k+1} = x_data(3);

k = k + 1;

end
```