CHEMICAL EQUILIBRIUM WITH APPLICATIONS X PROGRAM WITH USER INTERFACE AND VISUAL BASIC PROGRAMMING

BY

JAKE DUNCAN RUMEL

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

Professor M Quinn Brewster

ABSTRACT

This thesis details the creation of a program named CEA X Program that calculates the chemical equilibrium compositions of complex mixtures using a user interface created in Visual Basic. The program uses the iterative method of minimizing the Gibbs energy or the Helmholtz energy first utilized in the NASA CEA program[1]. The original NASA CEA Program requires the use of FORTRAN compliers and knowledge of how to create program input files. The new CEA X Program uses Excel macros and user forms which improves the user's experience while retaining the accuracy of chemical equilibrium calculations and adds new functionality to provide feedback to the user earlier in the input process. This document includes an overview of how the program was developed, details on how to use the program, and provides examples of the CEA X Program output and testing.

TABLE OF CONTENTS

Chapter 1: Introduction

The CEA X Program described in this document is a reconceptualization of the NASA CEA 2 Program[1] for calculating chemical equilibrium compositions. The CEA X Program includes a user interface which calls the Excel macros containing the subroutines necessary for performing the chemical equilibrium equations and produces an output document for printing the results. This document will begin with references to the calculations methodology and an overview of how the CEA X Program recreated this functionality. The remainder of the document details how the user interacts with the user interface and can serve as a user manual for the program. The end of the document has several example problems and details the testing done to confirm the program produces consistent results with the NASA CEA 2 Program[1].

1.1. Background on NASA CEA Program

The NASA Lewis Research Center's computer programming department created a program called CEA (Chemical Equilibrium with Applications) to ascertain the chemical equilibrium compositions of complex mixtures. The program received an update to version CEA 2 with the last modifications made May 21st, 2004. This document will refer to the CEA 2 version as the NASA CEA program[1]. The NASA CEA program[1] was written in ANSI standard FORTRAN 77 and requires the compiling of the *cea2.f* file with the *cea.inc* file and the thermodynamic data file (*thermo.inp*) and the optional thermal transport properties file (*trans.inp*). Once the program is compiled and run, it will take any (*.inp*) file and produce an (*.out*) file with the results of the chemical equilibrium calculations and an optional (*.plt*) file.

The NASA CEA program[1] uses two thermodynamic state functions to obtain chemical equilibrium compositions for species within a user specified mixture. The chemical equilibrium compositions are calculated using a descent Newton-Raphson iteration method for minimizing the Gibbs energy or Helmholtz energy based on which state functions are used to describe the thermodynamic state. The program will use this iteration process until the convergence criteria are achieved. Gaseous products only are considered first for convergence, and then the program will check if condensed phases of the products need to be included in the system.

To explain the NASA CEA program[1], the NASA Reference Publication 1311[2,3] by Sanford Gordon and Bonnie J. McBride was created and is broken into two sections. Part I of the document details the approach and the calculations used for the chemical equilibrium analysis. Part II of the document serves as a user manual for the CEA Program and describes the various subroutines in the program. The CEA X

Program in this document uses the same calculations described in Part I of the NASA Reference Publication 1311[2]. This document can be seen as a replacement of the Part II of the NASA Reference Publication 1311[3] with the details specific to the CEA X Program.

1.2. User Interface Purpose and Enhancements

The major difference between the CEA X Program and the NASA CEA program[1] is the addition of a user interface. The user interface was added to guide the user to specify the minimum necessary information required to perform the program calculations and removes the necessity for the user to understand how to create and format a problem (*.inp*) file. The new user interface also allows the user to quickly see all options for the input or output that are available from the program.

Additionally, the program adds several checks of the input data to prevent information that will automatically result in an output error in the NASA CEA program[1] or CEA X Program. These checks include:

- Check that the minimum data input requirements are met
- Check that the reactant temperatures specified can retrieve the reactant enthalpy from the thermodynamic library
- Check that the fuel-to-oxidant ratios match the number of specified reactant fuels and oxidants
- Use dropdown menus for units to ensure the program recognizes unit input
- Add data type validations (numeric inputs for temperature, etc.)

The user interface also includes the ability for the user to go in and view the thermodynamic properties that have been loaded into the reactant and product libraries of the program. The information was previously not available in the NASA CEA program[1] without the ability to read (*.lib)* files and could only be partially seen from the (*thermo.inp*) file. While the CEA X Program does not allow the user to edit these thermodynamic properties directly, it allows the user to update the reactants and products from the (*thermo.inp*) file or equivalent.

Using a user interface for selecting reactants from a list during the problem phase input allows for additional performance enhancements for the program. The CEA X Program uses an integer to keep track of which row in the stored thermodynamic data to pull the properties for selected reactants. This allows the program to avoid searching the entire list of reactants using a name search (which is performed in the NASA CEA program[1]). A similar integer approach is not used for searching for

available reactants since this search requires searching the library for available products based on the chemical formula instead of a set list.

Finally, the user interface adds a simpler management of the program parameters. The NASA CEA program[1] required updating the parameters in the *cea.inc* file and recompiling the program. The CEA X Program allows the user to quickly change these parameters based on the user preferences.

1.3. Out of Scope Components

The CEA X Program recreates much of the NASA CEA program[1] functionality for calculating the chemical equilibrium compositions and output capabilities, however the following features of the NASA CEA program[1] were considered out of scope for the CEA X Program:

- The Rocket Problem type
- The Shock Problem type
- The Chapman-Jouguet Detonation Problem type
- Calculating the thermal transport properties

Chapter 2: Methodology for Migrating Code

The CEA X Program uses the same equations for calculating the chemical equilibrium compositions as the NASA CEA program[1]. For this reason, the subroutines in the program macros use a similar naming convention and the content is easily relatable. Additionally, names of specific code lines and section comment headers were kept consistent with the NASA CEA program[1] where appropriate to aide in the comparison of the code. However, the code language was updated from ANSI standard FORTRAN 77 to Visual Basic.

In addition to the language update, the CEA X Program changed from using global variables to using cells in hidden Excel sheets to store problem values. The sheet names are consistent with the grouping of global variables in the *cea.inc* file. This allows for better debugging and visibility of the information stored within the program. The program macros contain many comments relating the cell names to the variables used in the NASA CEA program[1].

2.1. Subroutine Relationships

The CEA X Program subroutine differences from the NASA CEA program[1] are the most significant for those that relate to reading the problem input information or printing the results output. Many of the subroutines that read the input have been replaced by the user interface or parts have been integrated into the smaller subroutines that activate during specific screen interactions. The output subroutines have been modified to print the results in an Excel document instead of (*.out*) and (*.plt*) files.

The CEA X Program uses the same *thermo.inp* file format for loading the thermodynamic data for the reactants and products. This allows for consistency of the data between both programs. While the CEA X Program performs the same calculations on the file as the NASA CEA program[1], the reactant and product data is stored in a hidden sheet within the Excel document instead of creating a separate (*.lib*) file. This allows the CEA X Program to be used on its own after the initial loading of reactants and products. The *thermo.inp* file is only required when this information needs to be updated.

Table 1 below describes the relationship of each NASA CEA program[1] subroutine to the subroutines or screen in the CEA X Program.

Table 1 Subroutine Relationships

NASA CEA[1] Subroutine	CEA X Program Relationship
Main Program	Separated into several subroutines with user interface
BLOCKDATA	Stored in Hidden sheets
CPHS and ALLCON	Updated for Visual Basic
DETON	Unused
EFMT	Unused
EQLBRM	Updated for Visual Basic
FROZEN	Unused
GUASS	Updated for Visual Basic
HCALC	Unused
INFREE	Replaced by user interface
INPUT	Split between "Problem Input and Output" screen and the Problem Type Specific screens
MATRIX	Updated for Visual Basic
NEWOF	Updated for Visual Basic
OUT1	Updated for Visual Basic and writing to Excel
OUT2	Updated for Visual Basic and writing to Excel
OUT3	Updated for Visual Basic and writing to Excel
OUT4	Unused
REACT	Updated for Visual Basic
RKTOUT	Unused
ROCKET	Unused
SEARCH and READTR	Updated to search Excel instead of (*.lib*) file
SETEN	Updated for Visual Basic
SHCK	Unused
THERMP	Updated for Visual Basic
TRANIN	Unused
TRANP	Unused
UTHERM	Updated for Visual Basic and writing to Excel
UTRAN	Unused
VARFMT	Unused

2.2. Program Flow Diagram

The CEA X Program uses several screens to replicate and add to the functionality offered by the NASA CEA program[1]. Figure 1 is a diagram which shows the flow of the CEA X Program screens which are detailed in the subsequent chapters.

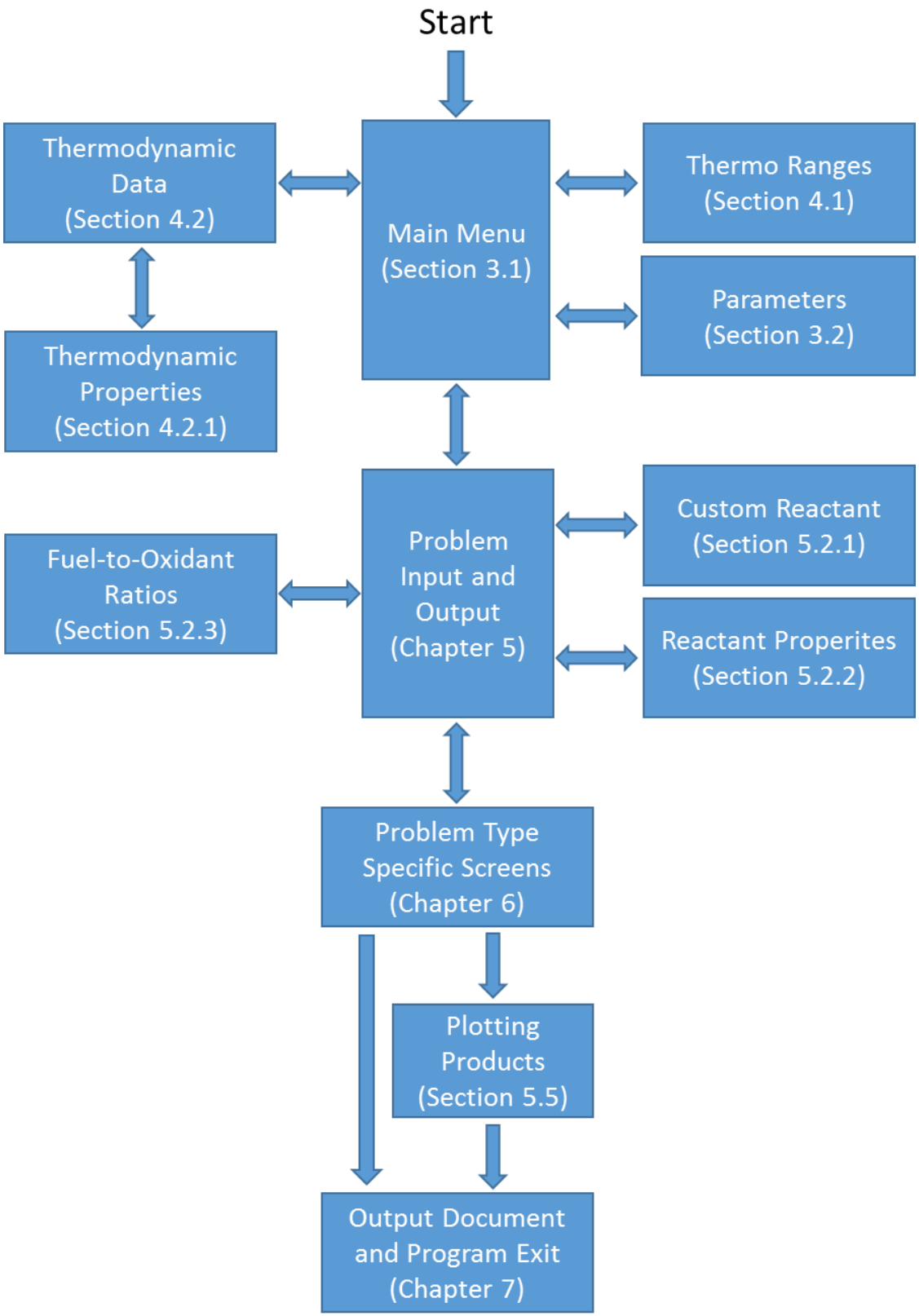

Figure 1 Program Flow Diagram

Chapter 3: Main Menu and Parameters

3.1. Main Menu

The program starts on the CEA tab which contains details about when the code was last updated and a button to start the main macro. Clicking on the Main Menu button brings up the screen "CEA Program – Main Menu" (See Figure 2). This screen and all other screens generated by the program use a subroutine to center the starting position in the middle of the open Excel Document which allows for ease of use with multiple monitor displays.

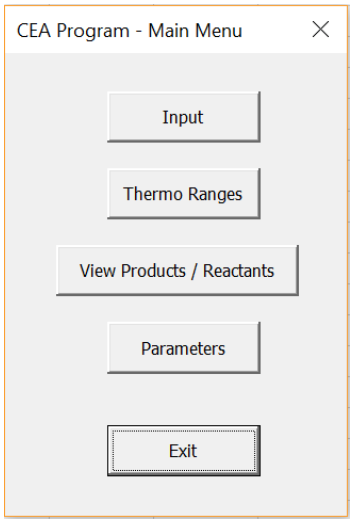

Figure 2 Main Menu

The Main Menu has five available options:

Table 2 Main Menu Options

Button	Description
Input	Begins user input for equilibrium problem (See Chapter 6)
Thermo Ranges	Displays the temperature ranges considered for the thermodynamic properties of the reactants and products (See Chapter 4)
View Products / Reactants	Displays and loads properties of the reactants and products (See Chapter 4)
Parameters	Allows the user to manage the CEA X Program settings (See Section 3.2)
Exit	Stops the program macro

The Exit button does not clear any information that has been saved from the previous screens.

3.2. Parameters

Selecting the Parameters button displays the “Program Parameters” screen seen in Figure 3.

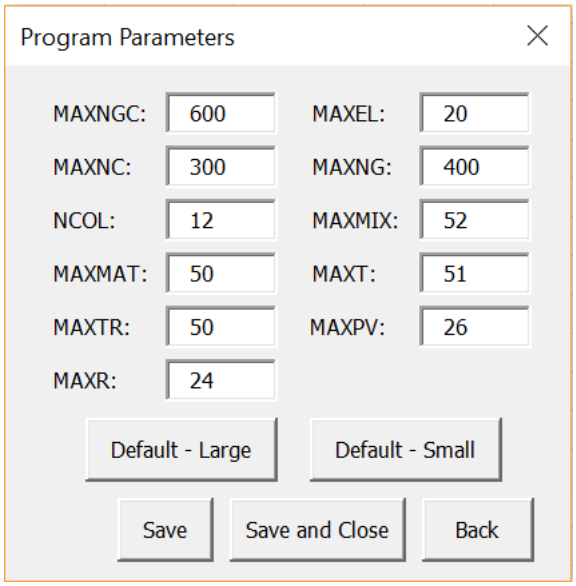

Figure 3 Program Parameters

These parameters manage the number of data elements allowed for each species or input. Increasing these limits can allow for more complicated systems to be considered, but will decrease performance. The naming convention used in this screen matches PARAMETER definitions in the NASA CEA code[1].

Table 3 Parameter Descriptions

Parameter	Description
MAXNGC	Maximum number of gaseous or condensed species that can be considered
MAXNC	Maximum number of condensed species temperature intervals allowed
NCOL	Columns of data printed in the output document for formatting
MAXMAT	Maximum number of rows for the composition iteration matrix
MAXTR	Number of gaseous products considered for thermal transport calculations (unused)
MAXR	Maximum number of reactants allowed to be specified
MAXEL	Maximum number of elements allowed for consideration
MAXNG	Maximum number of gaseous products allowed for consideration
MAXMIX	Maximum number of mixture values allowed
MAXT	Maximum number of temperature inputs allowed
MAXPV	Maximum number of pressure or volume/density inputs allowed

Each of these values can be changed and saved using the Save or Save and Close buttons. The program has two default settings that can be loading using the Default – Large or Default – Small buttons. The default values follow closely to those suggested in the NASA Reference Publication[3] and are as follows:

Table 4 Parameter Defaults

Parameter	Default - Large	Default – Small
MAXNGC	600	300
MAXNC	300	200
NCOL	12	12
MAXMAT	50	40
MAXTR	50	40
MAXR	24	24
MAXEL	20	15
MAXNG	400	200
MAXMIX	52	52
MAXT	51	51
MAXPV	26	26

All sample outputs in this document were produced using the Default – Large parameter settings.

Chapter 4: Product and Reactant Properties

4.1. Thermodynamic Temperature Ranges

The “Thermodynamic Temperature Ranges” screen is accessed through the “Main Menu” screen’s Thermo Ranges button. This screen functions as information only for the temperature ranges used for gaseous species properties. The range values are loaded during the load process for the products and reactant properties (see Section 4.2.2). Since these temperature ranges are used to set up the thermodynamic properties, and extrapolate some thermodynamic properties during the load process, they cannot be changed unless the input file is modified and all products and reactants are re-loaded.

The date listed in the “Thermodynamic Temperature Ranges” screen is also set from the product and reactant load process and describes when the thermodynamic properties were last modified in the file used for loading. The values seen in Figure 4 below are the values that were used for all sample problems in this document.

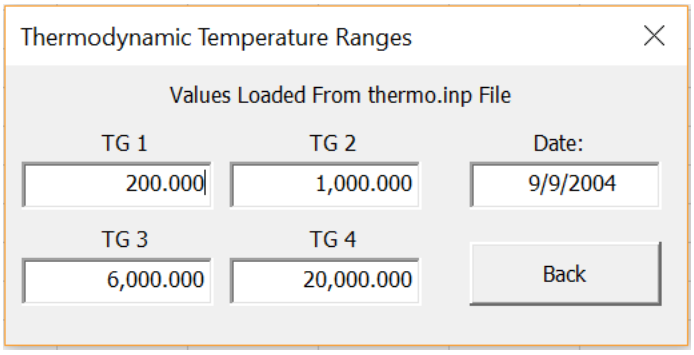

Figure 4 Thermodynamic Temperature Ranges

4.2. Thermodynamic Product and Reactant Data

The Chemical Equilibrium Program uses a set list of products and reactants which can be considered for inputs and outputs of problem sets. The products and reactants must be loaded prior to being able to run a program, but only need to be loaded once (unless a species needs to be added or modified) as the properties will be saved to a hidden sheet within the Excel document. The products and reactants can be viewed and managed through the “Thermodynamic Data” screen accessed through the View Products / Reactants button on the “Main Menu” screen.

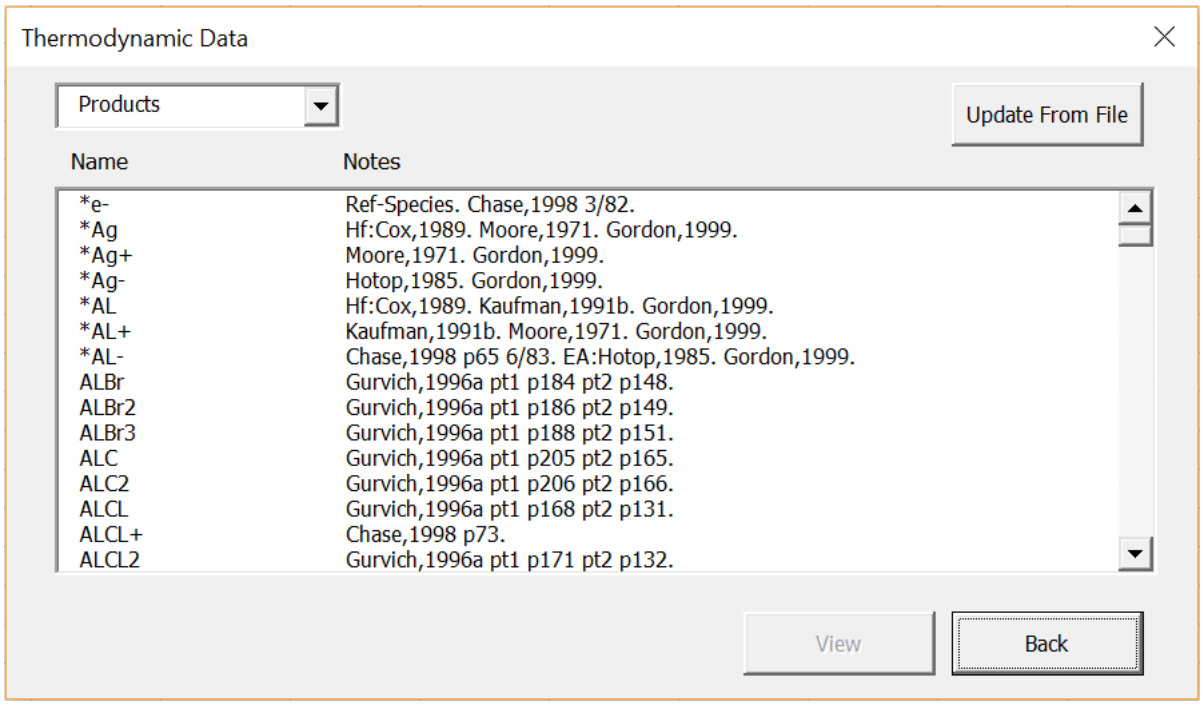

Figure 5 Thermodynamic Data

The "Thermodynamic Data" screen shown in Figure 5 above displays the Name and Notes for each species that have been loaded and are available for use by the program. The Name displayed is the name that will be presented to the User for selection in any selection screen and will appear in the output results. Names which begin with an * are gases which used explicitly defined thermodynamic properties for the third temperature interval (see Section 4.2.2). The Notes for each species are loaded from the input file and describe the source of the thermodynamic properties and the date of the source.

The dropdown menu in the upper left corner of the screen allows the user to switch between the products and reactants. The default for the screen is to display the products, but changing the dropdown allows the User to view the reactants with the same Name and Notes conventions.

The products and reactants will have only one instance of each gaseous species that is considered. However, condensed species can have multiple rows in this screen where each instance corresponds to a different temperature interval for the thermodynamic properties. This multi-instance convention is a result of the loading process (See Section 4.2.2) and is consistent with the NASA CEA program[1].

4.2.1. View Individual Species Properties

Selecting a species in the Figure 5 screen enables the View Button. Activating the View Button brings up a screen which displays the thermodynamic properties that have been loaded for the selected species.

Figure 6 below displays the properties for the $ALCL_2$ species. The screen title will display "Thermodynamic Properties -" and the name of the species selected from the previous screen.

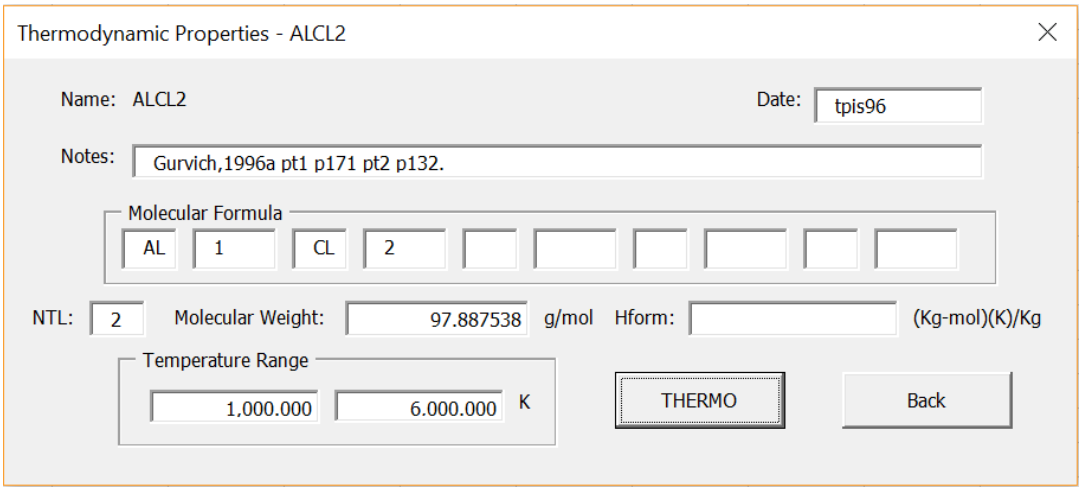

Figure 6 Species Thermodynamic Properties

The information in the "Thermodynamic Properties" screen for each species is informational only and cannot be modified. These values can only be updated by modifying the thermodynamic properties input file and re-loading through the Section 4.2.2 process. Table 5 below describes the information displayed in the "Thermodynamic Properties" screen:

Table 5 Thermodynamic Property Descriptions

Field	Description
Name	Name displayed in selection screens and "Thermodynamic Data" screen
Date	Six-character reference date code from input file. See Appendix B
Notes	Specific details about source of thermodynamic properties
Molecular Formula	Exploded form of the molecular formula. E used to identify ions
NTL	Number of temperature intervals that were supplied in the input file
Molecular Weight	Molecular weight of the species in g/mol
Hform	Enthalpy of the species in (kg-mol)*K/kg – Reactants Only
Temperature Range	Temperature range that species thermodynamic properties are defined for. If only one temperature is given, then temperature must be within +/- 10K from that temperature for thermodynamic properties to be considered.

Clicking on the THERMO button displays the thermodynamic values used by the program for the number of temperature intervals (NTL). Figure 7 below shows the "THERMO Details" screen for the $ALCL_2$

product seen in Figure 6. The THERMO button will be disabled for all species that do not have thermodynamic properties defined over a temperature interval and only have a temperature and enthalpy specified. The thermodynamic values are calculated during the Section 4.2.2 load process only and displayed as reference information in this screen.

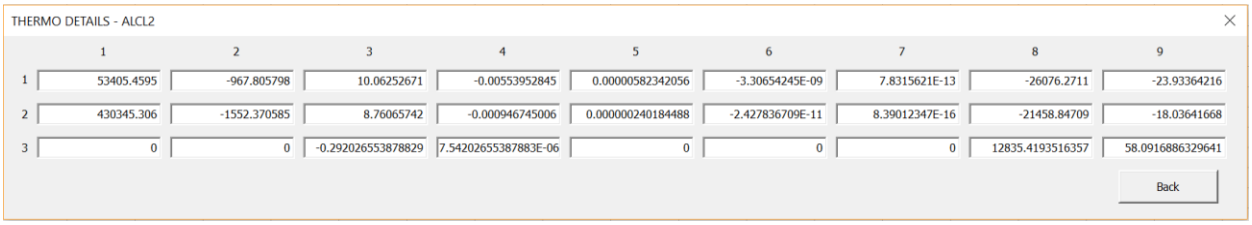

Figure 7 Species THERMO details

4.2.2. Loading Product and Reactant Properties

Updating or loading the products and reactants requires loading from an input file (.inp). The process is started by selecting the Update From File button on the "Thermodynamic Data" screen seen in Figure 5. Loading from the file removes all product and reactant information prior to loading in new data to avoid issues with duplicating species. Since this process will delete information, the User is prompted with a warning message in Figure 8 below prior to proceeding. Selecting OK will open up a browser for the User to select which Input File to load the products and reactants from.

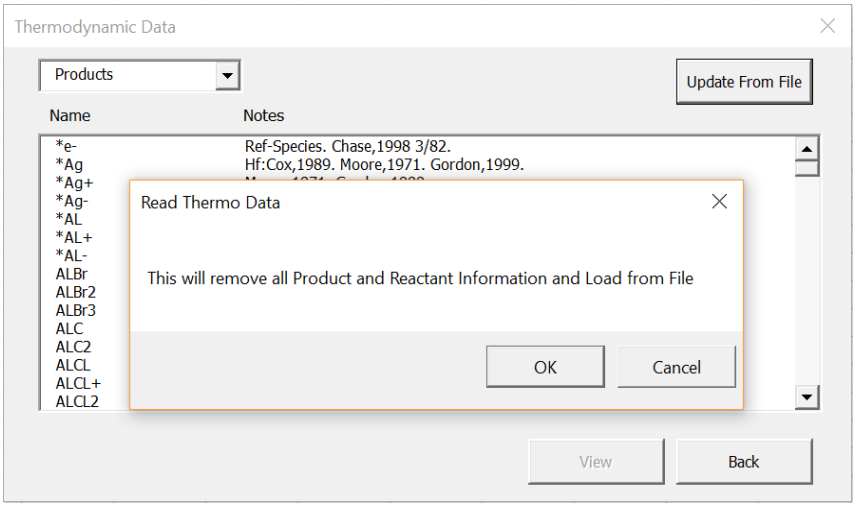

Figure 8 thermo.inp Load Warning

The products and reactants with their respective thermodynamic properties are loaded using the same thermo.inp file that is used to create the thermo.lib file in the NASA CEA program[1]. Figure 9 below shows the format of the thermo.inp files. The information under the thermo header line has the

temperature ranges and date which get loaded into the screen seen in Figure 4. Following that line is the information for the electron product with thermodynamic properties for the three thermodynamic ranges. The full description of the format required for the thermo.inp file can be found in NASA Reference Publication 1311 Appendix A[3].

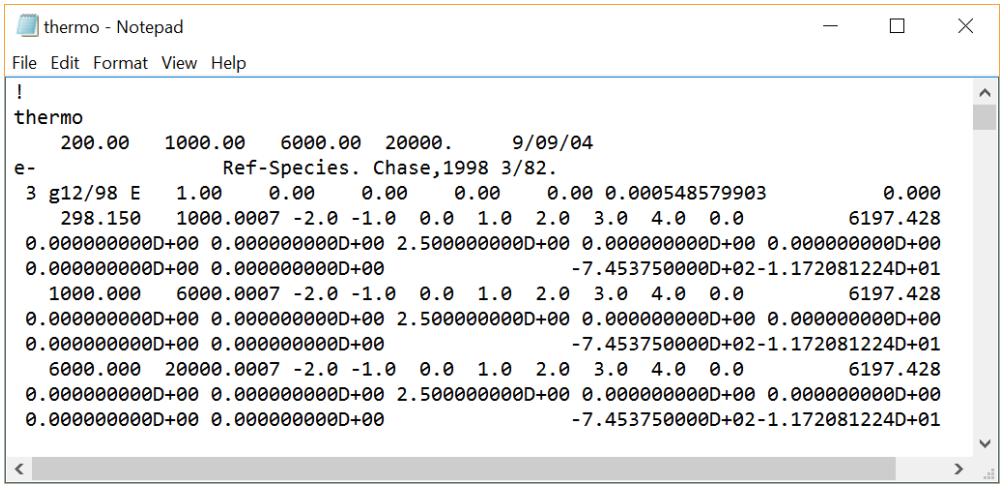

Figure 9 thermo.inp Format

The program reads in the data from the input file into a temporary hidden sheet then processes the thermodynamic properties with a process that mimics the UTHERM subroutines from the NASA CEA program[1]. For gases, if coefficients are not given for the third temperature interval, then the program uses a straight line extrapolation for Cp/R. Gases that have been defined for all three intervals without extrapolation are designated with an *. Condensed species are loaded with only one temperature range and no extrapolations are performed. If the condensed species in the input file has multiple temperature ranges specified, these are broken into multiple species lines. The calculations performed during loading of the thermodynamic data is detailed in NASA Reference Publication 1311[2] Section 4.

Chapter 5: Problem Input Screen

The majority of interaction with the program for the User is done in the “Problem Input and Output” screen. This screen is accessed by clicking the Input Button on the “Main Menu” screen and is used for specifying the various input parameters and options for the output. The problem type specific state properties are the only parameters not specified prior to advancing beyond this screen. These state properties are described in Chapter 6.

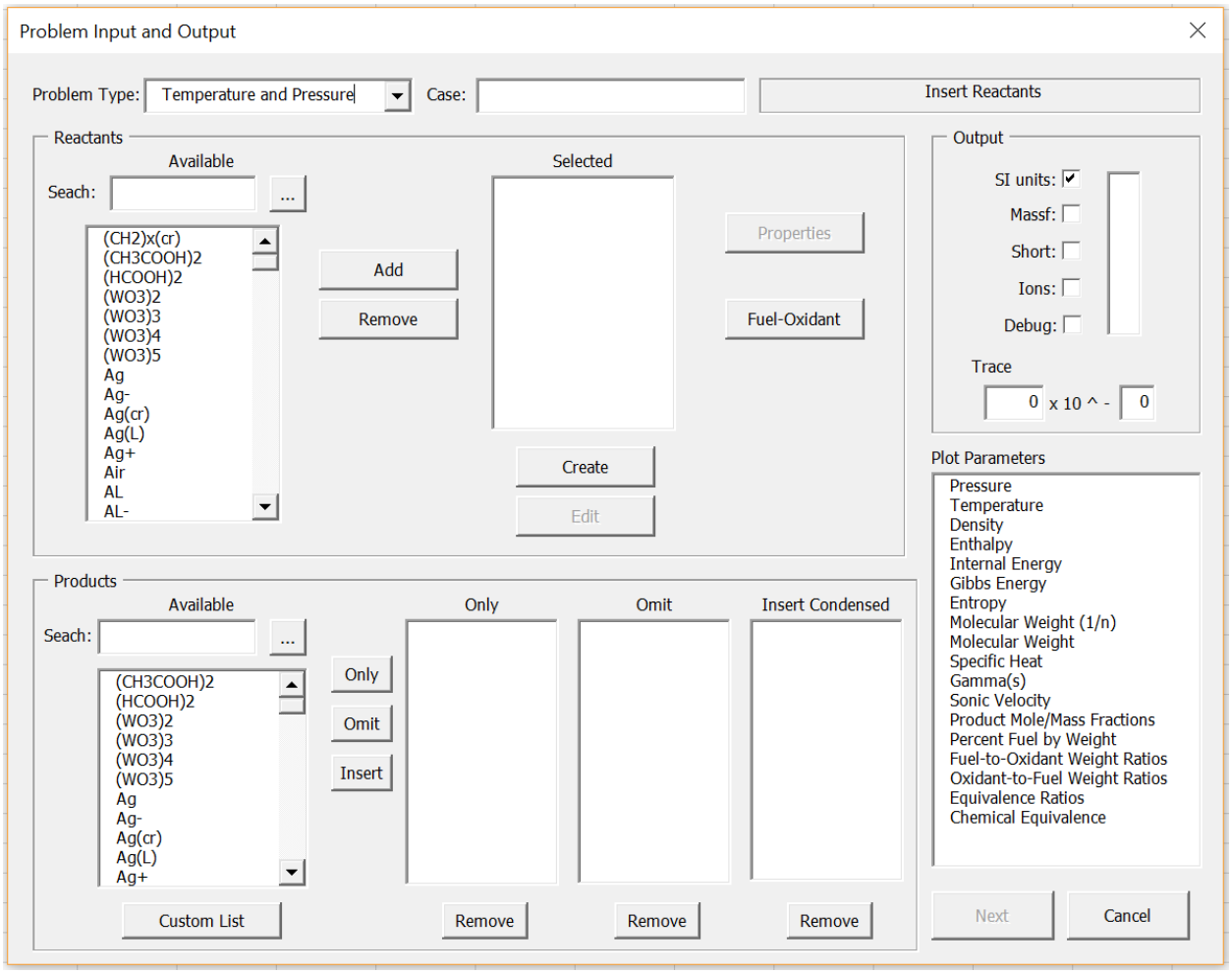

Figure 10 Problem Input and Output

Clicking on the Input Button will reset all variables to their default values and clear any input information. So, any information entered in the “Problem Input and Output” Screen is not deleted after hitting Cancel – which can be used for debugging – but is deleted with subsequent openings of the “Problem Input and Output” screen.

The “Problem Input and Output” screen seen in Figure 10 above is broken into various sections based on the what problem information is impacted by the User selections. Some sections require a minimum amount of information to be supplied before the User is allowed to advance the screen and contains some validation to prevent the user from entering values that are outside of the program requirements. However, the program cannot guarantee problem convergence for all user scenarios inputted.

5.1. Problem Information

The general problem information is located at the top of the screen just below the screen title. The component of this section is the Problem Type dropdown box. Table 6 below shows the different problem types as the following:

Table 6 Problem Types

Problem Types
Temperature and Pressure
Enthalpy and Pressure
Entropy and Pressure
Temperature and Volume
Internal-Energy and Volume
Entropy and Volume

Changing the Problem Type influences which Problem Type Specific Screens will display after advancing from the “Problem Input and Output” screen and what minimum input information is required. The Problem Type Specific Screens are described in Chapter 6. The default option for the Problem Type is Temperature and Pressure.

The problem information section allows the User to enter a name for the Case. This allows the User to provide a name that will be displayed in the problem output for reference.

In the top right corner of the “Problem Input and Output” screen is an input status indicator which is the text surrounded by a frame. This contains various messages that help guide the user to what additional information is required before the screen can advance with the Next Button. The following Table 7 details the possible messages in the input status indicator and the required actions (See Section 5.2 for completing actions required).

Table 7 Input Status Messages

Message	Action Required
Insert Reactants	Enter at least one reactant.
Need Reactant Amounts	Enter either reactant amounts or fuel-oxidant ratios.
Need Temperatures for Library Reactants	For Enthalpy or Internal-Energy problems, temperatures for each non-custom reactant have to be entered in the “Properties” screen.
Oxidant Not Permitted When 100% Fuel	If 100% Fuel is specified in the “Fuel-Oxidant” screen, then no reactant can be designated as an oxidant in the “Properties” screen.
Fuel Not Permitted When 100% Oxidant	If 100% Oxidant is specified in the “Fuel-Oxidant” screen, then no reactant can be designated as a fuel in the “Properties” screen.
Must Specify a Fuel and Oxidant	If 100% Fuel or Oxidant is not specified in the “Fuel-Oxidant” screen, then there must be at least one fuel reactant and oxidant reactant designated.
Ready	Minimum data requirements met and Next button is enabled.

5.2. Reactants

The reactants section of the “Problem Input and Output” screen contains the majority of the minimum required information and heavily influences the input status indicator (See Section 5.1). This section allows the User to specify which reactants are supplied for the chemical equilibrium problem and any necessary initial parameters needed for the reactants.

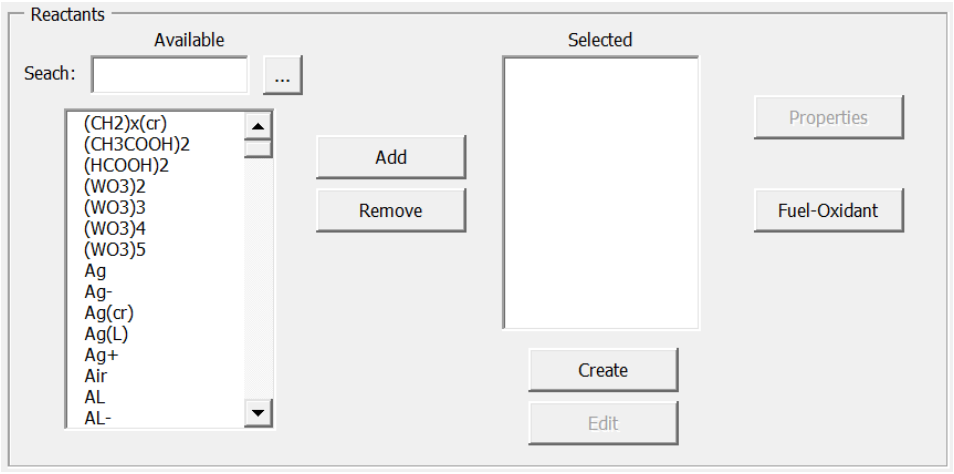

Figure 11 Selected Reactants

The list of available reactants is a multi-selectable list that is generated from the reactants that have been loaded into the program (See Section 4.2). Scrolling through the long list of reactants can be time consuming when the user already knows which reactants are desired for selection. The program has a search filter to help reduce the number of reactants that display in the multi-select list. The list of reactants will be filtered to all reactants that begin with the text displayed in the search text box after the user clicks on the "…" button. Figure 12 below shows the result of typing "O2" as the search filter and activating the filter. Clearing all text from the search filter and clicking on the "…" button will return all reactants to the multi-select list.

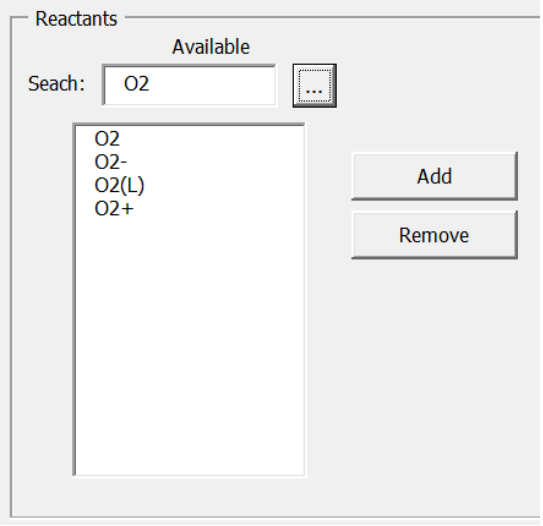

Figure 12 Search Filtered Reactants

Selecting one or more reactants in the available reactants list and clicking the Add button will place all highlighted reactants in the Selected list. Selecting a reactant will not remove that reactant from the available list, but the Add button checks the selected reactants list prior to adding an available reactant to avoid duplication. Clicking on a reactant in the selected list and activating the Remove button in the reactants section will delete that reactant from the selected list. Only one selected reactant can be removed per use of the remove button. The User will be prompted with an error message if the User tries to add more reactants than are allowed from the defined parameter (See Section 3.2).

5.2.1. Custom Reactants

If the user wants to supply a reactant that is not found in the list of available reactants, then the user may click on the Create button to add a custom reactant to the selected reactants list. The Create button will bring up the "Create Temporary Reactant" screen seen in Figure 13 below. This screen is used whenever the user adds a new custom reactant or chooses to edit a previously created reactant.

No edits are allowed for library reactants using this screen (See **Section 4.2.2** for editing library reactants).

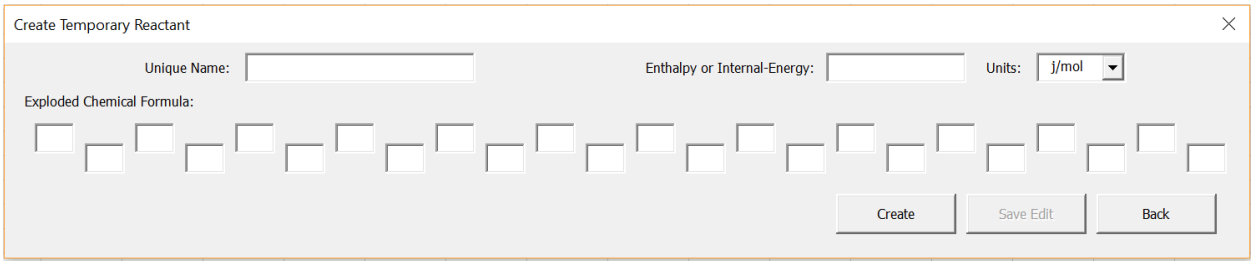

Figure 13 Custom Reactant

The reactants require a name to be supplied for the reactant (maximum length of fifteen characters). The name cannot match the name of a current library reactant and the user will be prompted with an error message if a matching name is supplied. The reactant name will be displayed in the selected reactants list after successfully creating the reactant and will be displayed in the output document (See Chapter 7) wherever the reactant name is displayed.

Additionally, each custom reactant requires a chemical formula to be supplied in exploded form. The chemical formula allows for a maximum of twelve elements and coefficients to be supplied. The element symbols are cross-referenced against the list of atomic symbols and prevent the user from creating the custom reactant unless all supplied symbols are valid. The coefficients must be numeric, but do not need to be integers. The program uses a coefficient of one if the user supplies an element symbol with no coefficient.

Finally, the custom reactant will require an enthalpy (H/R) or internal-energy (U/R) for each custom reactant. The enthalpy can be either set in the "Create Temporary Reactant" screen or the "Reactant Properties" screen (See Section 5.2.2). Leaving the value blank in the Figure 13 screen will set the enthalpy or internal-energy equal to zero. The zero value will have no impact on problems such as the "Temperature and Pressure" problem type in which the reactant enthalpy or internal-energy is not used. The enthalpy or internal-energy is supplied in j/mol, kj/mol, cal/mol, or kcal/mol units.

Custom reactants are not added to the program's internal library of reactants. If they are removed from the selected reactants list using the Remove button, then all information about the custom reactant is deleted. The user will have to recreate the custom reactant using the "Create Temporary Reactant" screen again. Additionally, custom reactants must be created every time the program is restarted. This is done for stronger data quality governance and prevents duplication of reactant species.

5.2.2. Reactant Properties

Once reactants have been added to the selected reactants list, the Properties button becomes available as seen in Figure 11.

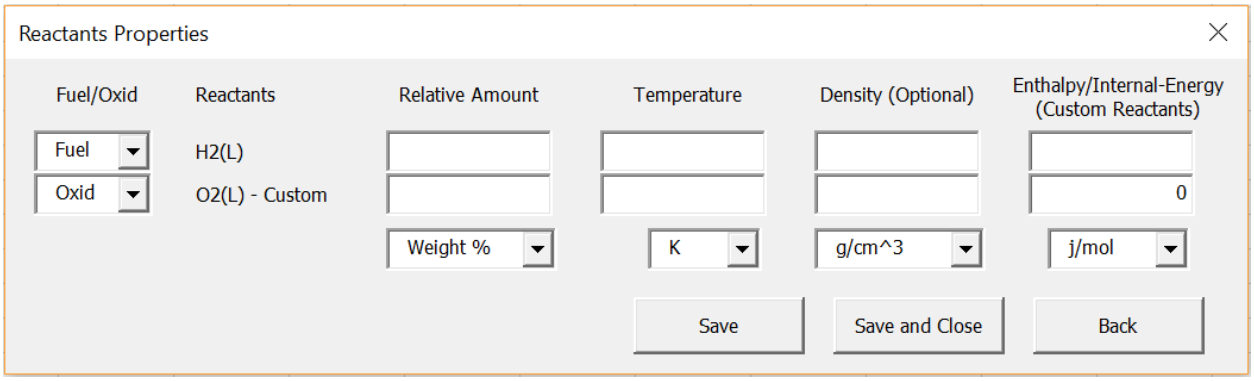

Figure 14 Reactant Properties

The "Reactants Properties" screen shown above in Figure 14 is a dynamically created screen that allows for the input of various properties for each reactant that were in the selected reactants list at the time the Properties button was activated. A row is created for each reactant and the name of the reactant becomes the label for each row.

The Fuel/Oxid column provides the choice to specify whether the reactant is to be the fuel or the oxidant for the reaction. There are no restrictions for which reactants need to be a fuel or which need to be an oxidant. However, there are rules for the number of fuels or oxidants specified if fuel/oxidant ratios have been specified (See Section 5.2.3).

The relative amount can be specified for each reactant. The dropdown menu at the bottom of the relative amount column allows the user to specify the amount in either Weight % or Moles. If the relative amounts are not supplied for all reactants, then fuel/oxidant ratios must be specified (See Section 5.2.3).

The temperature for each library reactant must be specified if the problem is an enthalpy or internal-energy problem. The temperatures can be specified in Kelvin (K), Fahrenheit (F), Rankine (R), or Celsius (C) with the use of the units' dropdown menu at the bottom of the temperature column. The temperature for each reactant must be specified such that it's thermodynamic properties have been defined for that temperature. For a gas, this is in the temperature range specified in the Section 3.2 load process or within 10K of the specified temperature for condensed species.

When the User actives the Save or Save and Close button with an invalid temperature, the program will display an error message similar to Figure 15 below to aid the User in specifying a correct value.

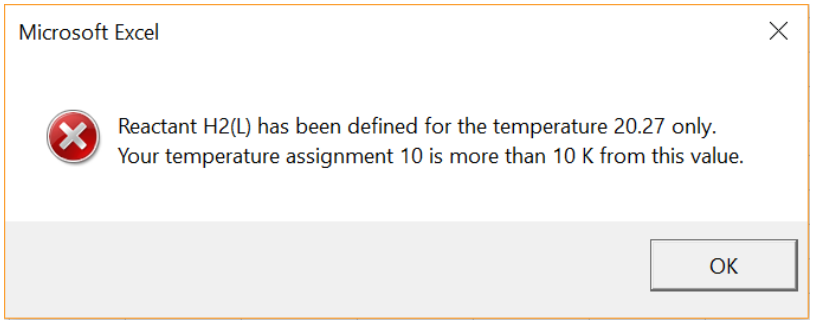

Figure 15 Temperature Property Error Message

If the User wants the output to calculate the density of the total reactant, then the User may specify the density for each reactant. The dropdown menu for the units at the bottom of the density column allow the units for g/cm^3 or kg/m^3. For problems with specific volume, densities can be supplied instead of specific volume. However, these densities are entered in the Problem Type Specific Screens (See Chapter 6) instead of this reactants properties screen.

Finally, if the reactant is a custom reactant and the problem is an enthalpy or internal-energy problem then an enthalpy (H/R) or internal energy (U/R) must be supplied. An enthalpy or internal-energy cannot be supplied for a library reactant since those will be calculated using the reactant temperature. If the user supplied an enthalpy or internal energy in Section 5.2.1, then this value will display in the "Reactant Properties" screen. The enthalpy or internal-energy must be supplied in either j/mol, kj/mol, cal/mol, or kcal/mol units.

5.2.3. Fuel-Oxidant Ratios

As an alternate to supplying relative amounts for each reactant, the User may enter in fuel/oxidant ratios using the "Fuel-Oxidant Mixture Values" screen shown in Figure 16 below after activating the Fuel-Oxidant button. The dropdown menu in the upper left corner of the screen allows the User to specify different fuel to oxidant ratios. The User may supply:

- Percent Fuel by Weight
- Fuel-to-Oxidant Weight Ratios
- Oxidant-to-Fuel Ratios
- Equivalence Ratios
- Chemical Equivalence Ratios

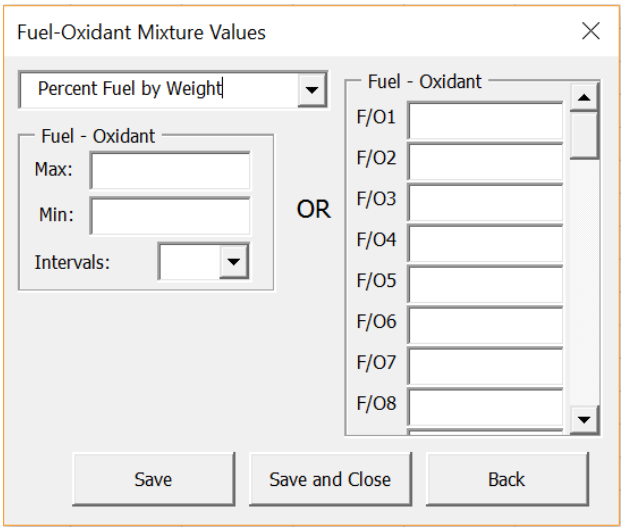

Figure 16 Fuel / Oxidant Ratios

The User has the option of supplying a maximum and minimum ratio with a number of equally spaced intervals using the Fuel – Oxidant frame on the left. An error message will display if the maximum ratio is less than the minimum ratio. Alternatively, the user can supply a number of specific ratios using the Fuel – Oxidant frame on the right. If both frames are filled out, the program will take the values from the maximum and minimum frames only.

If fuel/oxidant ratios are used, then the number of reactants labeled as a fuel or oxidant must abide by the following restrictions in Table 8 below:

Table 8 Fuel to Oxidant Constraints

Fuel – Oxidant Ratio	Reactant Constraints
100% Fuel	All reactants must be labeled as Fuel.
100% Oxidant	All reactants must be labeled as Oxidant.
Fuel and Oxidant Mixture	There must be at least one fuel and one oxidant in the reactants.

These constraints on the reactants prevent the User from running the problem with more than one of the above scenarios.

5.3. Products

The products section of the "Problem Input and Output" screen allows the User to control the products that are considered in the problem. The available products list is a multi-selectable list that is generated from the products that have been loaded into the program (See **Section 4.2**). With one or more

products selected, the User may select one of the three buttons – Only, Omit, or Insert to add all selected products to the corresponding list. The products will not be removed from the available products list after being added to another list, and the program will not add duplicates to any list. The Remove buttons under each of the lists will remove highlighted products in the Only, Omit, and Insert lists one at a time when activated. The following table describes the use case for each products list.

Table 9 Product List Types

Product List	Use Case
Only	If products are given in the Only list, the program will only consider these as the possible products for the reaction.
Omit	If products are given in the Omit list, the program will exclude these products from consideration in the reaction.
Insert	If products are given in the Insert list, the program will start the iterations with these species as condensed (instead of starting with only gaseous products in the iterations)

Due to the use cases, if the User adds a product to the omit list, then the program will remove that product from the only and insert lists (if necessary). Similarly, adding a product to either the only or insert lists will remove that product from the omit list if it has previously been added to the omit list.

To help the user select the products from the long list of products, the CEA X Program has two tools to filter the list of available reactants. The first is the same type of search filter that exists for the reactants (See beginning of Section 5.2). The second tool is the Custom Filter button. This pulls a list of the products from the "Filter" sheet which is editable by the user. This helps when running multiple problems with similar long lists of Only, Omit, or Insert products repeatedly. The program will check to determine if each product supplied in the custom list matches a product in the program library. If there is no match, then an error message is displayed to the user. Clicking on the "…" button with no text in the search bar will bring back the full list of products no matter which filter was used previously.

5.4. Output

The output section of the "Problem Input and Output" screen provides several options for the problem output. The description of each output option is summarized in the following Table 10:

Table 10 Output Options

Output Option	Description
SI units	Changes the units used in the output document.
Massf	Output values are given in mass fraction instead of mole fraction.
Short	Prints only error messages and final tables to the output file.
Ions	Sets whether ionic products are considered.
Debug	Creates a separate tab in the output document that prints the intermediate output of the program iterations for the selected parameter numbers. (See Section 7.2)
Trace	Changes the value of the trace threshold for printing product compositions.

5.5. Plot Parameters

The program is able to print several parameters in a format that is readily available for plotting. The plot parameters section of the "Problem Input and Output" screen is a multi-selection list of the available parameters for plot output. When one or more parameters have been selected the output document will create a tab with the parameters listed in table format (See **Chapter 7**).

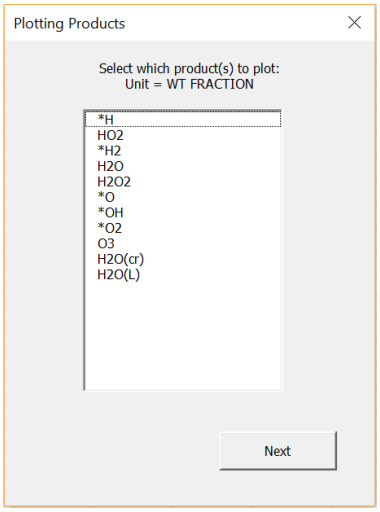

Figure 17 Plotting Products

Selecting the product mole/mass fraction plot parameter, prompts the user with an additional screen after the Problem Type Specific Screens. The "Plotting Products" screen shown in Figure 17 above provides a multi-select list of the products that are being considered for the problem. The User can select as many products as desired to add the mole/mass faction of that product to the plot output.

Chapter 6: Problem Type Specific Screens

Once the User has completed entering in all desired parameters into the "Problem Input and Output" screen, the program will advance to a Problem Type Specific Screen when the Next button is activated. The screen that displays depends on the selection in the Problem Type dropdown box from **Section 5.1**. The title of each screen will contain the selection from the dropdown for reference.

6.1. Temperature and Pressure

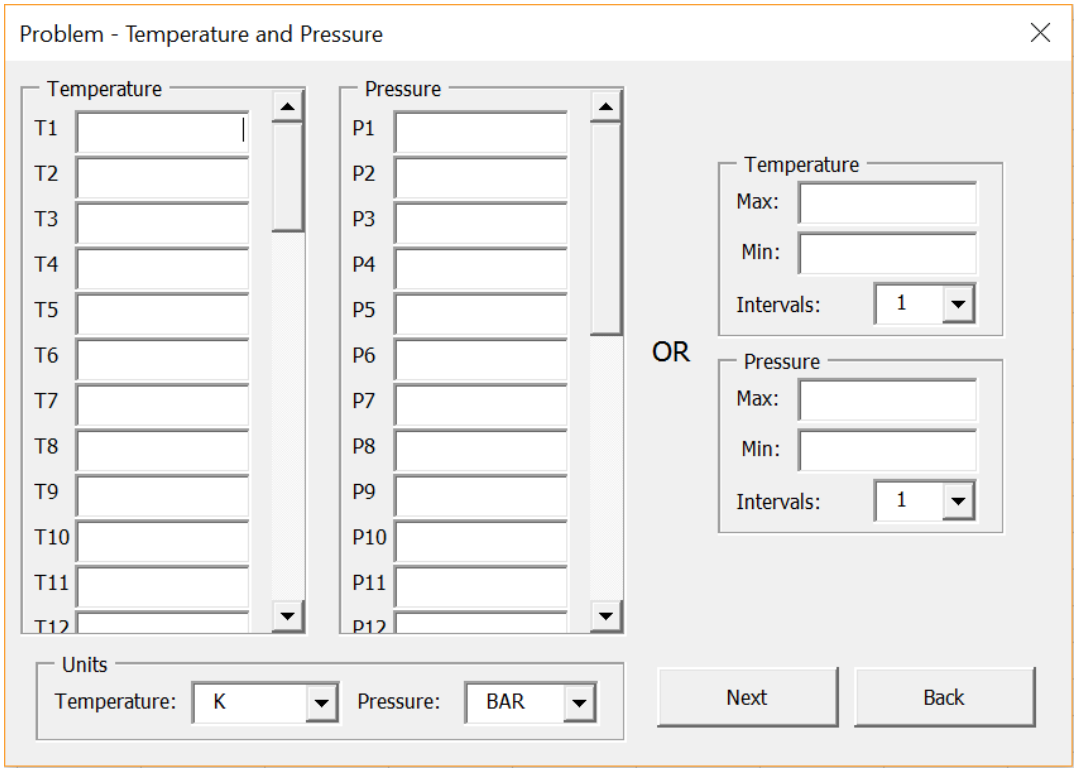

Figure 18 Temperature and Pressure Input

Temperature and Pressure problems require at least one value for each state properties, but the program also allows for a range of values to be supplied. If a range of values have been selected, the program will use a cross join method for the two ranges of parameters. This means that if two temperatures are given and one pressure, then two sets of results will be produced. Then, if two temperatures are given and two pressures are supplied, then four sets of results will be produced from the program.

The user may supply each individual temperature or pressure desired using the frames on the left of the "Problem – Temperature and Pressure" screen shown in Figure 18 above with a maximum number of elements set by the parameters (See Section 3.2). Alternatively, the user can use one or more of the

frames on the right of the screen to supply a maximum, minimum, and number of intervals that will create a range of equally spaced values for the user. The program uses the maximum and minimum range values in the event that the User enters values in both the left and right frames. Additionally, the program will check to make sure the maximum value supplied is larger than the minimum value supplied when the range frames on the right are being utilized.

The lower left section of the "Problem – Temperature and Pressure" screen controls the units for all temperature and pressure frames. The temperature may be supplied in Kelvin (K), Rankine (R), Celsius (C), or Fahrenheit (F). The pressure can be supplied in BAR, atmospheres (ATM), pounds per square inch (PSI), or millimeters Mercury (mmHg).

6.2. Enthalpy and Pressure

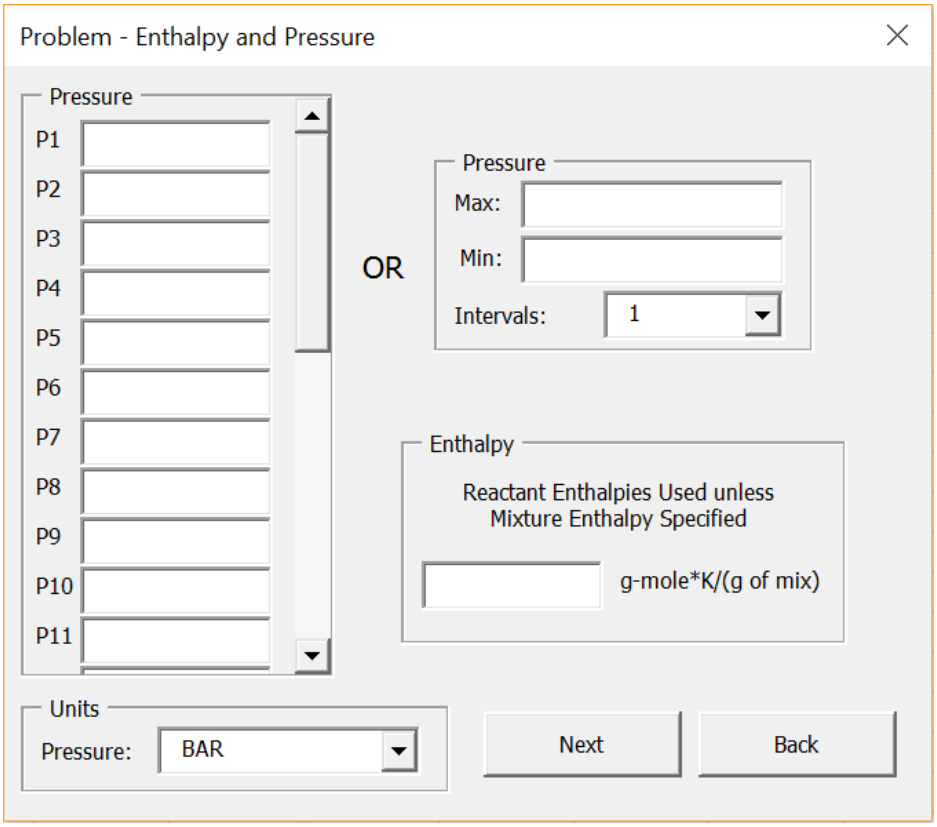

Figure 19 Enthalpy and Pressure Input

Enthalpy and Pressure problems require at least one value to be supplied for the pressure. The program will use the enthalpies of the reactants at the temperatures supplied for each reactant (See Section 5.2.2). Alternatively, the user may choose to supply one enthalpy (H/R) for the mixture that will override the enthalpies from the individual reactant temperatures. This enthalpy is supplied in the

labeled frame seen in the “Problem – Enthalpy and Pressure” screen shown in Figure 19 above and must be in units g*mole*K/(g of mixture). The pressure may be supplied in BAR, atmospheres (ATM), pounds per square inch (PSI), or millimeters Mercury (mmHg) which is controlled with the dropdown menu in the lower left section of the screen.

While only one value is allowed for the enthalpy, the program allows for a range of pressures to be supplied. The range of values can be individually specified in the frame on the left of the screen with a maximum number of elements set by the parameters (See Section 3.2). The frame on the right of the “Problem – Enthalpy and Pressure” screen allows the program to create an equally spaced range of values from a supplied maximum pressure, minimum pressure, and number of intervals. The program uses the maximum and minimum range values in the event that the user enters values in both the left and right frames. Additionally, the program will check to make sure the maximum value supplied is larger than the minimum value supplied when the range frames on the right are being utilized.

6.3. Entropy and Pressure

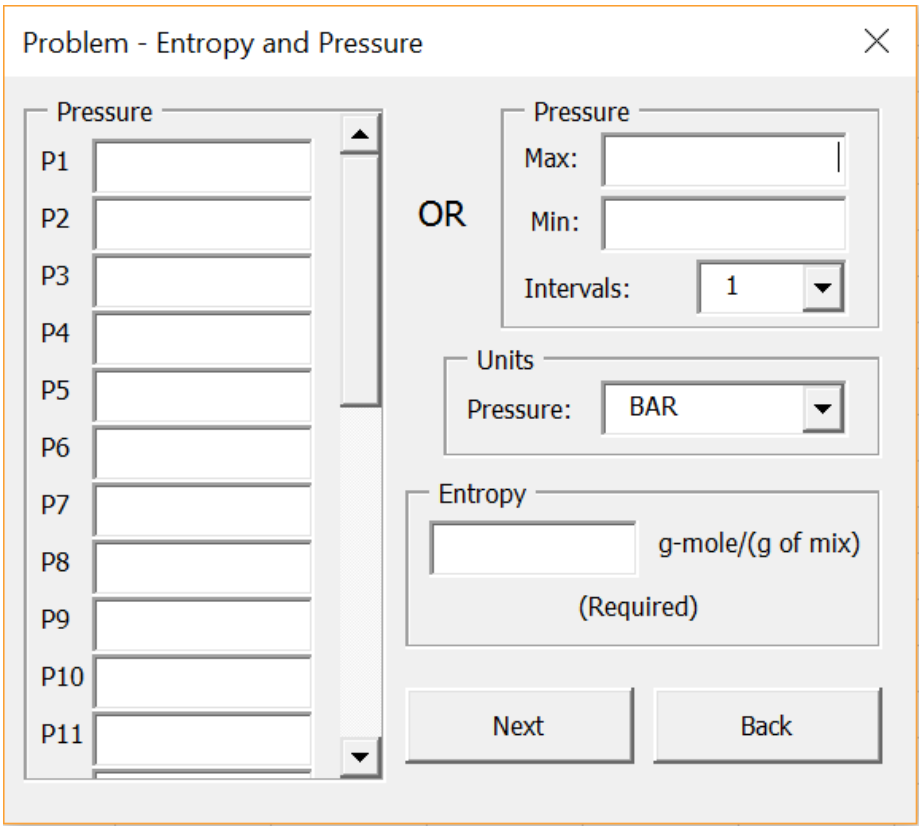

Figure 20 Entropy and Pressure input

Entropy and Pressure problems require at least one value to be supplied for the pressure and exactly one value for the entropy (S/R). The entropy is supplied in the labeled frame on the right side of the screen above the Next and Back buttons seen in the “Problem – Entropy and Pressure” screen shown in Figure 20 above and must be in units g*mole/(g of mixture). The pressure may be supplied in BAR, atmospheres (ATM), pounds per square inch (PSI), or millimeters Mercury (mmHg) which is controlled with the dropdown menu in the middle right section of the screen.

While only one value is allowed for the entropy, the program allows for a range of pressures to be supplied. The range of values can be individually specified in the frame on the left of the screen with a maximum number of elements set by the parameters (See Section 3.2). The frame on the right of the “Problem – Entropy and Pressure” screen allows the program to create an equally spaced range of values from a supplied maximum pressure, minimum pressure, and number of intervals. The program uses the maximum and minimum range values in the event that the user enters values in both the left and right frames. Additionally, the program will check to make sure the maximum value supplied is larger than the minimum value supplied when the range frames on the right are being utilized.

6.4. Temperature and Volume

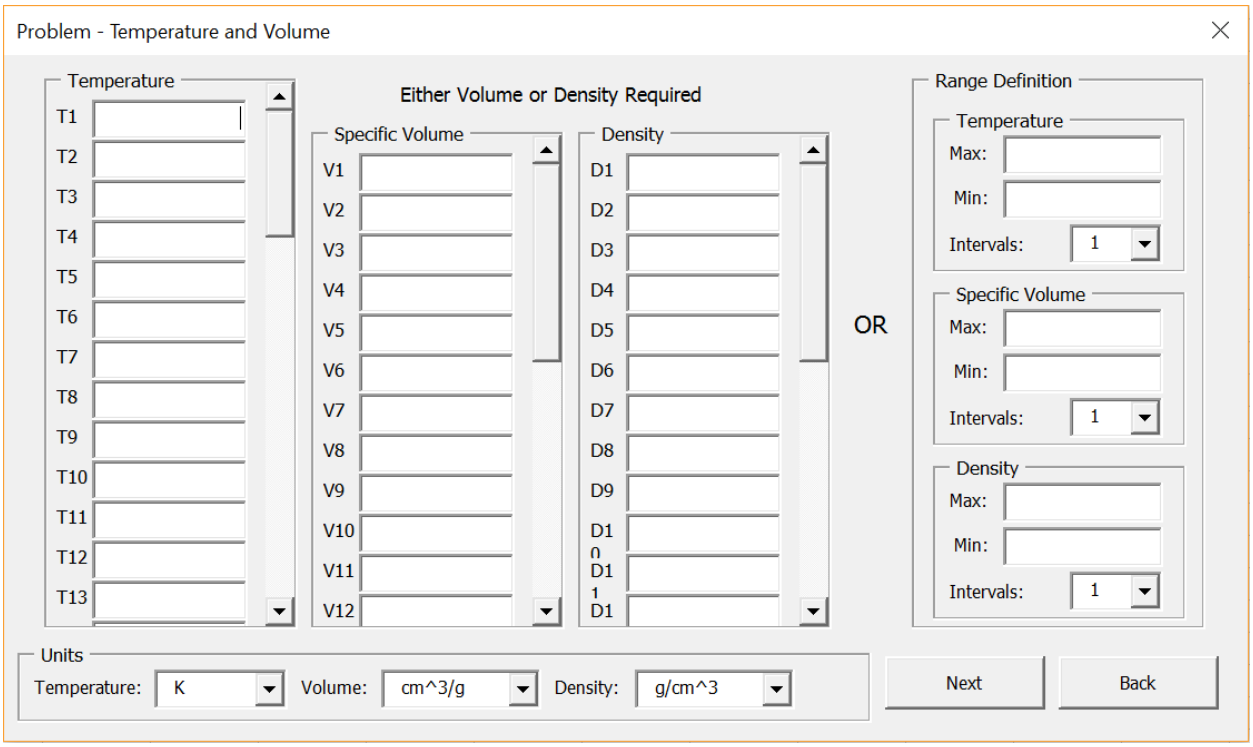

Figure 21 Temperature and Volume Input

Temperature and Volume problems require at least one value for temperature and at least one value for the volume or density. Similar to other problem types, the program also allows for a range of values to be supplied for each property. If a range of values have been selected, the program will use a cross join method for the two ranges of parameters. This means that if two temperatures are given and one pressure, then two sets of results will be produced. Then, if two temperatures are given and two pressures are supplied, then four sets of results will be produced from the program.

The user may supply each individual temperature, volume, or density desired using the frames on the left of the "Problem – Temperature and Volume" screen shown in Figure 21 above with a maximum number of elements set by the parameters (See Section 3.2). Alternatively, the user can use one or more of the frames on the right of the screen to supply a maximum, minimum, and number of intervals that will create a range of equally spaced values for the user. The program will check to make sure the maximum value supplied is larger than the minimum value supplied when the range frames on the right are being utilized.

The program uses the maximum and minimum range values in the event that the User enters values in both the left and right frames. The program will also take preference to values entered for the volume over density specified values. The user has no restrictions on whether to use either the volume or the density even if individual reactant densities were supplied (See Section 5.2.2).

The lower section of the "Problem – Temperature and Volume" screen controls the units for all temperature, volume, and density frames. The temperature may be supplied in Kelvin (K), Rankine (R), Celsius (C), or Fahrenheit (F). The specific volume uses cm^3/g or m^3/kg while the density uses grams per cubic centimeter (g/cm^3) or kilograms per cubic meter (kg/m^3) units.

6.5. Internal-Energy and Volume

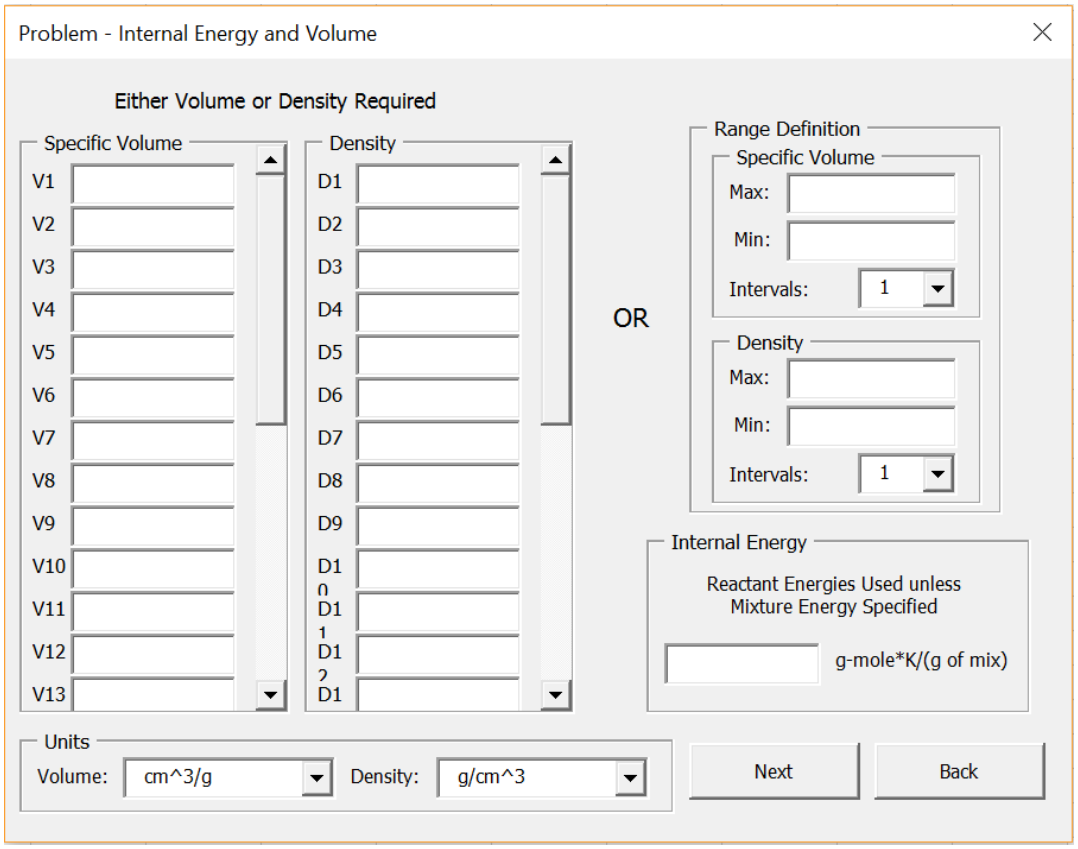

Figure 22 Internal Energy and Volume Input

Internal-Energy and Volume problems require that at least one value must be supplied for either the volume or density. The program will use the internal-energies of the reactants at the temperatures supplied for each reactant (See Section 5.2.2). Alternatively, the user may choose to supply one internal-energy (U/R) for the mixture that will override the internal-energies from the individual reactant temperatures. This internal-energy is supplied in the labeled frame seen in the "Problem – Internal-Energy and Volume" screen shown in Figure 22 above and must be in units g*mole*K/(g of mixture). The specific volume uses cm^3/g or m^3/kg while the density uses grams per cubic centimeter (g/cm^3) or kilograms per cubic meter (kg/m^3) units which is controlled with the dropdown menu in the lower left section of the screen.

The user may supply each individual volume or density desired using the frames on the left of the "Problem – Internal-Energy and Volume" screen with a maximum number of elements set by the parameters (See Section 3.2). Alternatively, the user may use one or more of the frames on the right of the screen to supply a maximum, minimum, and number of intervals that will create a range of equally

spaced values for the user. The program will check to make sure the maximum value supplied is larger than the minimum value supplied when the range frames on the right are being utilized.

The program uses the maximum and minimum range values in the event that the User enters values in both the left and right frames. The program will also take preference to values entered for the volume over density specified values. The user has no restrictions to use either the volume or the density even if individual reactant densities were supplied (See Section 5.2.2).

6.6. Entropy and Volume

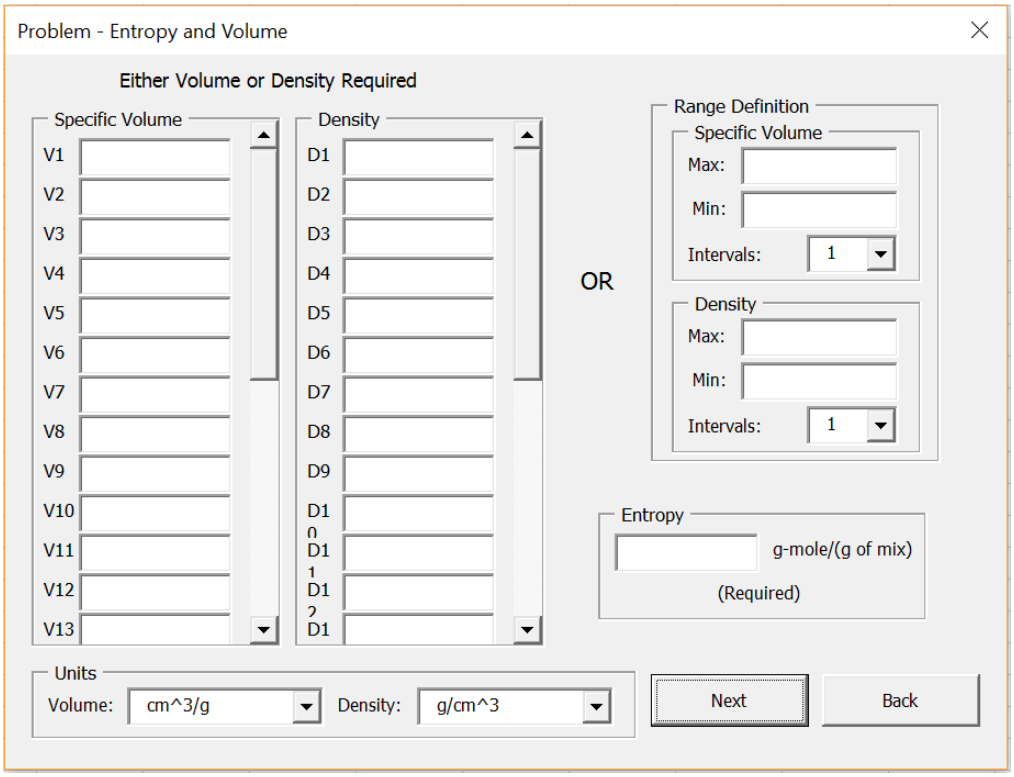

Figure 23 Entropy and Volume

Entropy and Volume problems require that at least one value must be supplied for either the volume or density and exactly one value for the entropy (S/R). The entropy is supplied in the labeled frame seen in the "Problem – Entropy and Volume" screen shown in Figure 23 above and must be in units g*mole/(g of mixture). The specific volume uses cm^3/g or m^3/kg while the density uses grams per cubic centimeter (g/cm^3) or kilograms per cubic meter (kg/m^3) units which is controlled with the dropdown menu in the lower left section of the screen.

The user may supply each individual volume or density desired using the frames on the left of the "Problem – Entropy and Volume" screen with a maximum number of elements set by the parameters

(See Section 3.2). Alternatively, the user can use one or more of the frames on the right of the screen to supply a maximum, minimum, and number of intervals that will create a range of equally spaced values for the user. The program will check to make sure the maximum value supplied is larger than the minimum value supplied when the range frames on the right are being utilized.

The program uses the maximum and minimum range values in the event that the User enters values in both the left and right frames. The program will also take preference to values entered for the volume over density specified values. The user has no restrictions to use either the volume or the density even if individual reactant densities were supplied (See Section 5.2.2).

Chapter 7: Output Document

Upon completion of entering in all problem input information from Chapters 5 and 6, the program will run the chemical equilibrium iterations until convergence has been achieved or the program encounters a fatal error due to problems with a lack of supplied information. In both cases, the program will produce a new Excel document with the results of the calculations and exit all program screens. The information contained in the output document depends on the selection during the problem input (See Section 5.4 and 5.5).

7.1. OUTPUT Tab

The OUTPUT Tab will always be created for the output document and is defaulted as the active tab. The first five rows of the page will contain the title which includes:

- Program Name and Date Modified
- The Author
- Reference to the NASA CEA program[1] version the program is based upon

The seventh row will contain the case name if one was given during the problem input. The eight row will have the problem type that was selected in dropdown menu on the "Problem Input and Output" screen (See Section 5.1). The remainder of the output is broken into sections which are printed or withheld depending on the user's selection of a short output.

During the calculations, the program can experience errors with the specified input properties which prevent the problem from converging. The program also may require some warnings be printed to the user to highlight potential inaccuracies. These warnings and error messages will be printed to the OUTUT Tab. See Appendix C for a full list of possible errors and warnings.

7.1.1. Problem Inputs

If the user has not selected the short output option, the OUTPUT Tab will contain information about the parameters that were selected by the user. If the user selected the short option, the output will skip to Section 7.1.3. This information may be used as either a check of the inputs or a record of the selections for future review. The first part of this section displays the logical selections that the user specified for the values in the following Table 11.

Table 11 Program Option Descriptions

Option	Description
TP	User selected the Temperature and Pressure Problem Type
HP	User selected the Enthalpy and Pressure Problem Type
SP	User selected the Entropy and Pressure Problem Type
TV	User selected the Temperature and Volume Problem Type
UV	User selected the Internal-Energy and Volume Problem Type
SV	User selected the Entropy and Volume Problem Type
IONS	The program was told to consider ionic products
SIUNIT	The units for output were requested in SI units
DEBUG	The output will report intermediary iteration matrices
TRNSPT	The program considers transport properties (unused)

After the options section, the output will list the range of mixture temperatures specified by the user (if applicable). The temperatures will always be listed in Kelvin (K) in this section. Next, the output will list the trace input from the user. If the trace input is zero, then the program used the default value of 5.0×10^{-6}. Then, the output will report the user specified entropy (S/R), enthalpy (H/R), or internal-energy (U/R). These values will be zero for problem types that do not require these as inputs. Finally, the output will list the range of either specific volumes or pressures specified by the user depending on the problem type. The specific volume will always be in m^3/kg and the pressure in bar units for this section.

Following the mixture state properties section, the output will report the reactants that were chosen and the properties of the reactants. The following Table 12 summarizes the information provided about each reactant and which units the property may be printed in.

Table 12 Mixture Output Columns

Column	Description	Units
1	Contains the reactant name and designation for fuel (F:) or oxidant (O:)	N/A
2	This is the user specified value for the relative amount	WT. FRAC or MOLES
3	This is the reactant enthalpy for the column 4 reactant temperature	ENERGY/R (K)
4	This is either the user specified reactant temperature or temperature pulled from the thermodynamic library	Kelvin (K)
5	This is the user specified density (if applicable)	g/cm^3
6 - 15	This is the exploded chemical formula of the reactant	N/A

After the reactants have been specified, the program will list all the species which are being considered for the products of the system. This list will be influenced by the user selections of the Only, Omit, and Insert lists (See Section 4.3). Condensed species can be displayed more than once since they are broken apart for consideration across multiple temperature ranges.

Next, the output will display the first oxidant to fuel ratio that was specified by the user in Section 5.2. If the user did not specify a ratio, the value will be calculated from the relative amounts displayed and displayed for this field. If the user specified a range of oxidant to fuel ratios, the program will continue printing the output according to the following sections then repeat each step starting from this display of the oxidant to fuel ratio (displaying the next ratio in the set).

Finally, the output will display the next display properties of the effective fuel, effective oxidant, and the mixture. Depending on the Problem Type, the output will display either the enthalpy or internal-energy for each of these three groups. These values will be specified with kgmol*K/kg units. Next, the relative atoms per kilogram for each element will be listed for each of the three groups in kg-form-wt /kg.

7.1.2. Convergence Details

If the user has not selected the short output option, the OUTPUT Tab will contain information about when each set of input parameters reached convergence. The table will contain the POINT which is a sequential integer for each set of parameters specified. Then, it will list the ITN which signals how many iterations were required to achieve convergence of gaseous species before considering condensed species. Next, the table lists the temperature in Kelvin of the mixture. Followed by a column for each

element considered in the problem with the energy which minimizes the Gibbs equation. The program will report between rows if the program considers adding or removing a condensed species to the list of considered products.

7.1.3. Thermodynamic Equilibrium Properties

The thermodynamic equilibrium properties section will display for all program outputs. The section is started with a title stating "THERMODYNAMIC EQUILIBRIUM PROPERTIES AT ASSIGNED" followed by the problem type. This section starts with a shorter version of the problem reactants and the fuel to oxidant measures. The reactants properties are given in a table with the following information shown in Table 13 below.

Table 13 Equilibrium Properties Output

Column	Description	Units
1	FUEL or OXIDANT designation	N/A
2	The reactant name	N/A
3	This is the user specified value for the relative amount	WT FRACTION or MOLES
4	This is the reactant energy	CAL/MOL or kJ/kg-mol
5	This is either the user specified reactant temperature or temperature pulled from the thermodynamic library	Kelvin (K)

Right below the reactant properties, the problem will list the oxidant to fuel ratio (O/F), the percentage fuel of the mixture by weight (%FUEL), the chemical equivalence ratio in terms of valences (R), and the equivalence ratios in terms of fuel-to-oxidant weight ratios (PHI).

Next, the program will print the output from the chemical equilibrium calculations under the THERMODYNAMIC PROPERTIES and MOLE FRACTIONS/MASS FRACTIONS headers. The output is set up in vertical columns for each set of problem parameters that were passed to the problem set. The following Table 14 details the values listed in the thermodynamic properties section and the units which are printed.

Table 14 Thermodynamic Properties Output

Label	Description	SI Units	Units
P	Pressure	BAR	ATM
T	Temperature	K	K
RHO	Density	kg/m^3	g/m^3
H	Enthalpy	kJ/kg	cal/g
U	Internal-Energy	kJ/kg	cal/g
G	Gibbs Energy	kJ/kg	cal/g
S	Entropy	kJ/kg*K	cal/g*K
M	Molecular Weight	(1/n) or mol wt	(1/n) or mol wt
(dLV/dLP)t	Thermodynamic derivative relating to Compressibility	N/A	N/A
(dLV/dLT)p	Thermodynamic derivative relating to Coefficient of Thermal Expansion	N/A	N/A
Cp	Specific Heat for Constant Pressure	kJ/kg*K	cal/g*K
GAMMAs	Heat capacity ratio	N/A	N/A
SON VEL	Sonic Velocity	m/s	m/s

The mole or mass fractions are listed for all products that have a value larger than the problem trace limit. If a range of parameters was specified creating multiple output columns, then it is possible for a reactant to be listed with a mole or mass fraction of zero. Only one column needs to have a non-zero fraction for the product to be displayed in this section. If the user has not specified the short condition for the option, then the problem will list all the products that were considered, but did not have sufficient mole or mass fractions to be displayed in the result set.

The program will only create a maximum number of columns for the thermodynamic properties. This maximum number is controlled by the NCOL parameter (See Section 3.2). If the number of parameter sets is greater than the maximum number of columns, then the program will restart the printing of the remaining thermodynamic properties starting back at printing the convergence details (See Section 7.1.2). This allows the user to control the width of the OUTPUT Tab page for printing. For comparing a large number of parameter sets, it is convenient to use the plot parameter feature described in Section 7.3.

7.2. DEBUG Tab

The DEBUG Tab will only display if the user has selected the debug option in the "Problem Input and Output" screen (See Section 5.4). The output allows for the debug output to have a maximum of nine parameter combinations using the multi-select list that displays once the debug option is selected. Example: If the program inputs for a temperature and pressure problem are (1,000 K and 50 BAR) and (3,000 K and 50 BAR), then selecting debug and the number 1 prints the debug output for the (1,000 K and 50 BAR) iterations only. Selecting debug and only the number 2 prints the debug output for the (3,000 K and 50 BAR) iterations only. Selecting debug and both the number 1 and 2 will print the debug output for both iterations.

Similar to the OUTPUT Tab, the first five rows of the DEBUG Tab will contain the title which includes:

- Program Name and Date Modified plus description "DEBUGGING"
- The Author
- Reference to the NASA CEA program[1] version the program is based upon

The seventh row will contain the case name if one was given during the problem input. The eight row will have the problem type that was selected in dropdown on the "Problem Input and Output" screen (See Section 5.1).

7.2.1. Iteration Matrices and Compositions

The debug information for each set of parameter inputs is broken up with a header reading "DEBUG - " and the number representing the set of parameters being printed. These numbers correspond to the numbers selected on the "Problem Input and Output" screen described above.

Each debug section begins by printing the iteration matrix which is used during the calculations. This matrix corresponds to the equations in Tables 2.1 and 2.2 in NASA Reference Publication 1311[2]. Next, the solution vector is printed with labels for the current components being considered. Then, the program will print the temperature (T), mole number (ENN), ln mole number (ENNL), pressure (PP), ln of pressure / mole number (LN P/N), and the control factor (AMBDA) for the current iteration. A description of which product sets the control factor follows that output.

The next section lists a row for each product that is being considered for the problem. The columns for this table are the product composition (Nj), ln of the product composition (LN Nj), the composition

corrections (DEL LN Nj), the dimensionless enthalpy (H0j/RT), dimensionless entropy (S0j/R), dimensionless standard-state Gibbs energy (G0j/RT), and the dimensionless Gibbs energy (Gj/RT).

The iteration matrices and compositions section is then repeated for each iteration until the program has reached convergence or the maximum number of iterations.

7.2.2. Condensed-Phases and Derivative Matrices

Following the iteration matrices and composition section is the condensed-phases test section. This section will list each condensed species the problem is considering and the temperature range for which that condensed species' thermodynamic properties are being considered. Some condensed species will be listed more than once with differing temperature ranges. The number of moles of that condensed species will be listed to the right of the species' temperature range.

The program calculations set up the temperature and pressure derivative matrices after the species compositions have converged. The equations for these matrices can be found in Tables 2.3 and 2.4 in NASA Reference Publication 1311[2]. Both of these matrices are printed in the DEBUG Tab after the condensed-phases section along with their respective solution vectors.

7.2.3. Thermodynamic Properties

The debug section will end with a summary of the thermodynamic properties at equilibrium for that converged set of input parameters. The properties listed in Table 15 that get displayed in this section are similar to those listed in the OUTPUT Tab (See Section 6.1.3), but do not have the density, internal-energy, or the Gibbs energy.

Table 15 Debug Thermodynamic Properties Output

Label	Description	SI Units
P	Pressure	BAR
T	Temperature	K
H/R	Enthalpy	K
S/R	Entropy	dimensionless
M	Molecular Weight	(1/n)
(dLV/dLP)t	Thermodynamic derivative relating to Compressibility	N/A
(dLV/dLT)p	Thermodynamic derivative relating to Coefficient of Thermal Expansion	N/A
Cp/R	Specific Heat for Constant Pressure	K
GAMMAs	Heat capacity ratio	N/A
SON VEL	Sonic Velocity	m/s

The DEBUG Tab then repeats all of the debug sections for each set of parameters requested for debugging by the user.

7.3. PLOT Tab

When the user selects at least one plot parameter on the "Problem Input and Output" screen (See Section 5.5), the program will create a tab in the output document labeled PLOT. The PLOT Tab output creates an output that is formatted for easily creating plots of various problem results. The NASA CEA program[1] has similar functionality where the output creates a separate (.plt) file which prints the data in the same plot friendly grid.

Similar to both the OUTPUT Tab and DEBUG Tab, the first five rows of the PLOT Tab will contain the title which includes:

- Program Name and Date Modified plus description "PLOT"
- The Author
- Reference to the NASA CEA program[1] version the program is based.

The seventh row will contain the case name if one was given during the problem input. The eight row will have the problem type that was selected in the dropdown menu on the "Problem Input and Output" screen (See Section 5.1).

The output will then create a column for every property or product composition requested by the user. Each row will correspond to one set of input parameters. The units for each property will follow the user's selection for the thermodynamic properties that are printed in Section 7.1.3. The names of the columns are summarized in the following Table 16.

Table 16 Plot Parameter Titles

Label	Description
P	Pressure
T	Temperature
RHO	Density
H	Enthalpy
Int E	Internal-Energy
G	Gibbs Energy
S	Entropy
MM	Molecular Weight (1/n)
MMW	Molecular Weight (mol wt)
Cp	Specific Heat for Constant Pressure
GAM	Heat capacity ratio
SON	Sonic velocity
%F	Percentage Fuel by Weight
F/A	Fuel-to-oxidant ratio
O/F	Oxidant-to-fuel ratio
PHI	Equivalence ratios in terms of fuel-to-oxidant weight ratios
R	Chemical equivalence ratios in terms of valences
Product Name	Product mole or mass fraction

The user may then use the in application Excel plotting tools or export the data for plotting with other applications. See Chapter 8 for examples of some graphs created from the plot output.

Chapter 8: Program Testing and Examples

Testing of the CEA X Program entailed a combination of testing the user interface functionality as well as testing the results of the program output. The user interface functionality was tested using a series of test cases that needed to pass each time changes were made to the code. Appendix A shows all of the detailed test cases that were checked and were required to pass in order for the program to be considered functionally complete.

To test the problem output, six of the example problems in the NASA Reference Publication 1311[3] were chosen to recreate using the CEA X Program. The examples chosen were 1 through 5 and example 14 since these corresponded to problem types considered in-scope for the CEA X Program. Additionally, some new examples were created to display the functionality of the remaining problem types. All examples were checked using both the CEA X Program and the NASA CEA Program to ensure the results were in agreement between the two programs. The following sections are summaries of the problem inputs and outputs.

8.1. Sample Problem #1 - TP

This sample problem is the same as the NASA Publication[3] Example #1.

Problem Type = TP

Table 17 Sample Problem #1 Parameters

Parameter	Values		
P (atm)	1	0.1	0.01
T (K)	3000	2000	
r ratio	1	1.5	

Table 18 Sample Problem #1 Reactants

Reactants	Fuel / Oxidant	Moles	
H2	Fuel	1	
Air	Oxidant	1	

Additional information: Calorie Output. Only Products: Ar, C, CO, CO2, H, H2, H20, HNO, HO2, HNO2, HNO3, N, NH, NO, N2, N2O3, O, O2, OH, O3

Results:

Table 19 Sample Problem #1 Output #1

Thermodynamic Properties for Chemical Equivalence Ratio = 1						
P, ATM	1	1	0.1	0.1	0.01	0.01
T, K	3000	2000	3000	2000	3000	2000
RHO, G/CC	9.18E-05	0.00015	8.09E-06	1.5E-05	6.61E-07	1.49E-06
H, CAL/G	663.7142	-203.569	1369.546	-191.833	2647.099	-164.346
U, CAL/G	399.8558	-365.131	1070.016	-353.764	2280.923	-327.156
G, CAL/G	-7974.53	-5290.41	-8616.7	-5662.79	-9380.96	-6036.51
S, CAL/(G)(K)	2.879414	2.543422	3.328748	2.735478	4.009353	2.936081
M, (1/n)	22.59411	24.60005	19.90339	24.54403	16.28084	24.4116
(dLV/dLP)t	-1.03442	-1.00064	-1.07867	-1.00146	-1.07479	-1.00358
(dLV/dLT)p	1.69574	1.020481	2.533313	1.046251	2.413327	1.11105
Cp, CAL/(G)(K)	1.681642	0.455165	3.439811	0.521493	3.716823	0.685773
GAMMAs	1.131193	1.225764	1.120582	1.202627	1.131821	1.166799
SON VEL,M/SEC	1117.505	910.2669	1185.051	902.6635	1316.829	891.5245
Mole Fractions						
***Ar**	0.007098	0.007728	0.006253	0.007711	0.005115	0.007669
***CO**	0.000171	1.04E-05	0.000184	2.1E-05	0.000168	4.08E-05
***CO2**	7.11E-05	0.000253	2.88E-05	0.000242	6.47E-06	0.00022
***H**	0.040752	8.95E-05	0.142895	0.000409	0.318941	0.001859
HO2	1.03E-05	1.01E-07	5.08E-06	1.03E-07	6.89E-07	1.02E-07
***H2**	0.067277	0.003061	0.082718	0.006384	0.041209	0.013191
H2O	0.207296	0.342065	0.095791	0.337141	0.011761	0.326373
***N**	1.06E-05	0	3.13E-05	0	8.97E-05	0
***NO**	0.012303	0.000483	0.013705	0.000721	0.009668	0.001066
***N2**	0.585676	0.644137	0.514484	0.64255	0.421584	0.638909
***O**	0.015397	2.13E-05	0.057868	0.0001	0.142614	0.000471
***OH**	0.045174	0.001128	0.059534	0.002435	0.032747	0.005186
***O2**	0.018761	0.001023	0.026501	0.002285	0.016095	0.005016

Products considered with mole fractions less than 0.000005: C, HNO, HNO2, HNO3, NH, N2O3, O3

Table 20 Sample Problem #1 Output #2

Thermodynamic Properties for Chemical Equivalence Ratio = 1.5						
P, ATM	1	1	0.1	0.1	0.01	0.01
T, K	3000	2000	3000	2000	3000	2000
RHO, G/CC	8.12E-05	0.00013	7.12E-06	1.3E-05	5.67E-07	1.29E-06
H, CAL/G	718.6538	-120.698	1550.122	-116.204	3208.235	-101.779
U, CAL/G	420.4753	-307.35	1209.797	-303.012	2781.243	-289.088
G, CAL/G	-8818.84	-5830.65	-9545.38	-6260.58	-10424.5	-6691.18
S, CAL/(G)(K)	3.179165	2.854975	3.6985	3.072186	4.544241	3.294702
M, (1/n)	19.99355	21.29328	17.5175	21.27547	13.96195	21.21856
(dLV/dLP)t	-1.0331	-1.00019	-1.08571	-1.00061	-1.0871	-1.00198
(dLV/dLT)p	1.665283	1.005548	2.667723	1.017654	2.642119	1.056961
Cp, CAL/(G)(K)	1.825311	0.467074	4.19285	0.500034	4.923027	0.607651
GAMMAs	1.133672	1.252882	1.119626	1.238905	1.129591	1.205121
SON VEL,M/SEC	1189.262	989.1611	1262.64	984.0398	1420.583	971.8308
Mole Fractions						
***Ar**	0.006193	0.006596	0.005426	0.00659	0.004325	0.006573
***CO**	0.000175	0.000156	0.000166	0.000156	0.000143	0.000155
***CO2**	3.58E-05	6.83E-05	1.86E-05	6.84E-05	4.52E-06	6.85E-05
***H**	0.06025	0.000621	0.182358	0.00196	0.393121	0.00616
***H2**	0.147056	0.147374	0.134716	0.14676	0.062606	0.144895
H2O	0.222242	0.295081	0.11131	0.294504	0.014651	0.292605
***N**	9.9E-06	0	2.92E-05	0	8.26E-05	0
***NO**	0.005651	8E-06	0.009123	2.54E-05	0.007294	8.06E-05
***N2**	0.513557	0.549954	0.44786	0.549485	0.356917	0.547987
***O**	0.007552	3.81E-07	0.041288	3.82E-06	0.116937	3.84E-05
***OH**	0.032758	0.00014	0.054207	0.000444	0.033096	0.001403
***O2**	0.004513	3.28E-07	0.013491	3.3E-06	0.010821	3.34E-05

Products considered with mole fractions less than 0.000005: C, HNO, HNO2, HNO3, HO2, NH, N2O3, O3

8.2. Sample Problem #2 - TV

This sample problem is the same as the NASA Publication[3] Example #2 without the transport option. This example will recreate the three pressure states at temperatures 3000 K for the equivalence ratio of one that was part of the output from the previous sample problem. The output thermodynamic properties and mole fractions will be the same to demonstrate the consistency of results for equivalent inputs of different problem types.

Problem Type = TV

Table 21 Sample Problem #2 Parameters

Parameter	Values		
rho (g/cc)	9.1864×10^{-5}	8.0877×10^{-6}	6.6054×10^{-7}
T (K)	3000		
phi ratio	1		

Table 22 Sample Problem #2 Reactants

Reactants	Fuel / Oxidant	Weight %	
H2	Fuel	100	
Air	Oxidant	100	

Additional information: Calorie Output. Only Products: Ar, C, CO, CO2, H, H2, H20, HNO, HO2, HNO2, HNO3, N, NH, NO, N2, N2O3, O, O2, OH, O3

Results:

Table 23 Sample Problem #2 Output

Thermodynamic Properties for Equivalence Ratio = 1			
P, ATM	1.000871	0.10003	0.010052
T, K	3000	3000	3000
RHO, G/CC	9.19E-05	8.09E-06	6.65E-07
H, CAL/G	663.5543	1369.409	2644.412
U, CAL/G	399.7038	1069.887	2278.379
G, CAL/G	-7974.3	-8616.61	-9379.06
S, CAL/(G)(K)	2.879284	3.328673	4.007825
M, (1/n)	22.59479	19.90386	16.28717
(dLV/dLP)t	-1.03441	-1.07867	-1.07494
(dLV/dLT)p	1.695502	2.533202	2.416265
Cp, CAL/(G)(K)	1.681194	3.439539	3.722534
GAMMAs	1.1312	1.120582	1.131752
SON VEL,M/SEC	1117.492	1185.037	1316.533
Mole Fractions			
***Ar**	0.007098	0.006253	0.005117
***CO**	0.000171	0.000184	0.000168
***CO2**	7.11E-05	2.88E-05	6.5E-06
***H**	0.040731	0.142874	0.318607
HO2	1.03E-05	5.08E-06	6.93E-07
***H2**	0.067266	0.082719	0.041336
H2O	0.207331	0.095807	0.011843
***N**	1.06E-05	3.13E-05	8.95E-05
***NO**	0.012302	0.013705	0.009683
***N2**	0.585694	0.514496	0.421742
***O**	0.015389	0.05786	0.142429
***OH**	0.045166	0.059534	0.032841
***O2**	0.018757	0.026501	0.016137

Products considered with mole fractions less than 0.000005: C, HNO, HNO2, HNO3, NH, N2O3, O3

8.3. Sample Problem #3 - HP

This sample problem is the same as the NASA Publication[3] Example #3. This example was chosen to show a combustion problem and the ability to showcase the impact of changing the trace value.

Problem Type = HP

Table 24 Sample Problem #3 Parameters

Parameter	Values		
P (bar)	100	10	1
O/F ratio	17		

Table 25 Sample Problem #3 Reactants

Reactants	Fuel / Oxidant	Weight Fraction	Temperature (K)
C7H8(L)	Fuel	0.4	298.15
C8H18(L), n-octane	Fuel	0.6	298.15
Air	Oxidant	1	700

Additional information: SI unit Output. Trace = 1 x 10^{-15}. Omit Products: CCN, CNC, C3H5,allyl, C3O2, C4H4,1,3-cyclo-, C4H8,tr2-butene, (CH3COOH)2, C4H9,s-butyl, C4H10,n-butane, C5H6,1,3cyclo-, C5H10,cyclo-, C5H12,n-pentane, C6H6, C6H12,1-hexene, C7H7,benzyl, C7H14,1-heptene, C8H8,styrene, C8H17,n-octyl, C7H8, C2N2, C3H6,propylene, C3H7,n-propyl, C4, C4H6,butadiene, C4H8,isobutene, C4H9,n-butyl, C4H9,t-butyl, C4N2, C5H8,cyclo-, C5H11,pentyl, C5H12,i-pentane, C6H5OH,phenol, C6H12,cyclo-, C7H8, C7H15,n-heptyl, C8H10,ethylbenz, C8H18,isooctane, C8H18,n-octane, C2O, C3H6,cyclo-, C3H7,i-propyl, C4H2,butadiyne, C4H6,2butyne, C4H8,cyclo-, C4H9,i-butyl, C4H10,isobutane, C5, C5H10,1-pentene, C5H11,t-pentyl, CH3C(CH3)2CH3, C6H10,cyclo-, C6H13,n-hexyl, C7H8O,cresol-mx, C7H16,n-heptane, C8H16,1-octene, C8H18,n-octane, C3H4,allene, C3H3,1-propynl, C3H3,2-propynl, C3H8O,2propanol, C3H8O,1propanol, C4H6,cyclo-, C4H8,1-butene, C4H8,cis2-buten, C3H8, C10H21,n-decyl, C12H10,biphenyl, C12H9,o-bipheny, C6H2, C6H5,phenyl, C6H5O,phenoxy, C10H8,naphthale, C9H19,n-nonyl, C6H6, C3H4,propyne, C3H4,cyclo-, C3, H2O(cr), H2O(L)

Results:

Table 26 Sample Problem #3 Output #1

Thermodynamic Properties for O/F Ratio = 17			
P, BAR	100	10	1
T, K	2418.66	2390.593	2338.84
RHO, KG/CU M	14.43106	1.456961	0.148338
H, KJ/KG	317.8376	317.8377	317.8376
U, KJ/KG	-375.112	-368.523	-356.299
G, KJ/KG	-19438	-20787.3	-21879.3
S, KJ/(KG)(K)	8.168091	8.8284	9.490679
M, (1/n)	29.02082	28.95944	28.84624
(dLV/dLP)t	-1.00068	-1.00159	-1.00324
(dLV/dLT)p	1.01893	1.044742	1.092048
Cp, KJ/(KG)(K)	1.60941	1.816676	2.206624
GAMMAs	1.225699	1.206138	1.179991
SON VEL,M/SEC	921.6009	909.8599	891.8942

Table 27 Sample Problem #3 Output #2

Mole Fractions							
***Ar**	0.008862	0.008843	0.008808	**NCO**	8.31E-11	5.82E-11	2.95E-11
***CN**	5.85E-14	1.07E-13	1.18E-13	***NH**	2.93E-09	3.86E-09	3.88E-09
***CO**	0.001677	0.004312	0.009186	**NH2**	1.71E-09	1.28E-09	7.37E-10
***CO2**	0.115369	0.112487	0.107156	**NH3**	3.89E-09	1.72E-09	6.13E-10
COOH	5.16E-08	2.33E-08	8.61E-09	**NH2OH**	1.02E-11	1.45E-12	1.68E-13
***H**	2.76E-05	0.000124	0.000455	***NO**	0.006776	0.006553	0.006145
HCN	1.06E-11	1.18E-11	8.63E-12	**NO2**	2.35E-05	7.56E-06	2.48E-06
HCO	7.38E-10	8.92E-10	7.6E-10	**NO3**	1.92E-10	1.95E-11	2E-12
HNC	1.02E-12	1.09E-12	7.48E-13	***N2**	0.735474	0.734034	0.731359
HNCO	1.23E-09	5.14E-10	1.64E-10	**N2H2**	5.17E-13	1.2E-13	2.1E-14
HNO	4.22E-07	2.08E-07	9.1E-08	**NH2NO2**	2.44E-15	6.73E-17	1.65E-18
HNO2	1.85E-06	3.27E-07	5.73E-08	**N2O**	3.64E-06	1.11E-06	3.3E-07
HNO3	1.13E-09	6.62E-11	4.04E-12	**N2O3**	2.44E-11	7.71E-13	2.43E-14
HO2	7.99E-06	4.26E-06	2.16E-06	**N2O4**	3.1E-15	3.28E-17	3.66E-19
***H2**	0.000251	0.000661	0.001484	**N3**	1.23E-12	3E-13	5.75E-14
HCHO,formaldehy	1.71E-11	1.16E-11	5.55E-12	**N3H**	3.85E-13	5.23E-14	5.58E-15
HCOOH	6.46E-09	1.64E-09	3.43E-10	***O**	0.000155	0.000431	0.001066
H2O	0.10277	0.101361	0.099015	***OH**	0.002348	0.003816	0.005721
H2O2	9.88E-07	2.95E-07	8.37E-08	***O2**	0.026252	0.027365	0.029599
***N**	1.15E-08	2.74E-08	5.07E-08	**O3**	1.21E-08	3.72E-09	1.12E-09

Products considered with mole fractions less than 0.000000000000001: C, CH, CH2, CH3, CH2OH, CH3O, CH4, CH3OH, CH3OOH, C2, C2H, C2H2,acetylene, C2H2,vinylidene, CH2CO,ketene, O(CH)2O, HO(CO)2OH, C2H3,vinyl, CH3CN, CH3CO,acetyl, C2H4, C2H4O,ethylen-o, CH3CHO,ethanol, CH3COOH, OHCH2COOH, C2H5, C2H6, CH3N2CH3, C2H5OH, CH3OCH3, CH3O2CH3, OCCN, C3H6O,propylox, C3H6O,acetone, C3H6O,propanal, CNCOCN, C4H6,1butyne, C6H14,n-hexane, C7H16,2-methylh, HCCN, HCCO, (HCOOH)2, NCN, N2H4, N2O5, C(gr)

8.4. Sample Problem #4 - UV

This sample problem is the same as the NASA Publication[3] Example #4. This problem is a recreation of the first state of the Sample Problem #3 and shows the consistency of the internal-energy and volume calculations with the enthalpy and pressure problem type.

Problem Type = UV

Table 28 Sample Problem #4 Parameters

Parameter	Values		
U/R	-45.1343		
Rho (kg/m^3)	14.428		
O/F ratio	17		

Table 29 Sample Problem #4 Reactants

Reactants	Fuel / Oxidant	Weight Fraction	Temperature (K)
C7H8(L)	Fuel	0.4	298.15
C8H18(L), n-octane	Fuel	0.6	298.15
Air	Oxidant	1	700

Additional information: Trace = 1 x 10^{-15}. Omit Products: CCN, CNC, C3H5,allyl, C3O2, C4H4,1,3-cyclo-, C4H8,tr2-butene, (CH3COOH)2, C4H9,s-butyl, C4H10,n-butane, C5H6,1,3cyclo-, C5H10,cyclo-, C5H12,n-pentane, C6H6, C6H12,1-hexene, C7H7,benzyl, C7H14,1-heptene, C8H8,styrene, C8H17,n-octyl, C7H8, C2N2, C3H6,propylene, C3H7,n-propyl, C4, C4H6,butadiene, C4H8,isobutene, C4H9,n-butyl, C4H9,t-butyl, C4N2, C5H8,cyclo-, C5H11,pentyl, C5H12,i-pentane, C6H5OH,phenol, C6H12,cyclo-, C7H8, C7H15,n-heptyl, C8H10,ethylbenz, C8H18,isooctane, C8H18,n-octane, C2O, C3H6,cyclo-, C3H7,i-propyl, C4H2,butadiyne, C4H6,2butyne, C4H8,cyclo-, C4H9,i-butyl, C4H10,isobutane, C5, C5H10,1-pentene, C5H11,t-pentyl, CH3C(CH3)2CH3, C6H10,cyclo-, C6H13,n-hexyl, C7H8O,cresol-mx, C7H16,n-heptane, C8H16,1-octene, C8H18,n-octane, C3H4,allene, C3H3,1-propynl, C3H3,2-propynl, C3H8O,2propanol, C3H8O,1propanol, C4H6,cyclo-, C4H8,1-butene, C4H8,cis2-buten, C3H8, C10H21,n-decyl, C12H10,biphenyl, C12H9,o-bipheny, C6H2, C6H5,phenyl, C6H5O,phenoxy, C10H8,naphthale, C9H19,n-nonyl, C6H6, C3H4,propyne, C3H4,cyclo-, C3, H2O(cr), H2O(L)

Results:

Table 30 Sample Problem #4 Output

Thermodynamic Properties for O/F Ratio = 17			
P, BAR	99.97369	**M, (1/n)**	29.02084
T, K	2418.538	**(dLV/dLP)t**	-1.00068
RHO, KG/CU M	14.428	**(dLV/dLT)p**	1.018922
H, KJ/KG	317.6449	**Cp, KJ/(KG)(K)**	1.60934
U, KJ/KG	-375.27	**GAMMAs**	1.225707
G, KJ/KG	-19437.2	**SON VEL,M/SEC**	921.5802
S, KJ/(KG)(K)	8.168087		
Mole Fractions			
***Ar**	0.008862	**NCO**	8.3E-11
***CN**	5.84E-14	***NH**	2.93E-09
***CO**	0.001676	**NH2**	1.71E-09
***CO2**	0.11537	**NH3**	3.89E-09
COOH	5.15E-08	**NH2OH**	1.02E-11
***H**	2.75E-05	***NO**	0.006775
HCN	1.05E-11	**NO2**	2.35E-05
HCO	7.37E-10	**NO3**	1.92E-10
HNC	1.01E-12	***N2**	0.735475
HNCO	1.23E-09	**N2H2**	5.17E-13
HNO	4.22E-07	**NH2NO2**	2.43E-15
HNO2	1.85E-06	**N2O**	3.64E-06
HNO3	1.13E-09	**N2O3**	2.44E-11
HO2	7.99E-06	**N2O4**	3.09E-15
***H2**	0.000251	**N3**	1.23E-12
HCHO,formaldehy	1.71E-11	**N3H**	3.85E-13
HCOOH	6.45E-09	***O**	0.000155
H2O	0.102771	***OH**	0.002348
H2O2	9.88E-07	***O2**	0.026252
***N**	1.15E-08	**O3**	1.21E-08

Products considered with mole fractions less than 0.000000000000001: C, CH, CH2, CH3, CH2OH, CH3O, CH4, CH3OH, CH3OOH, C2, C2H, C2H2,acetylene, C2H2,vinylidene, CH2CO,ketene, O(CH)2O, HO(CO)2OH, C2H3,vinyl, CH3CN, CH3CO,acetyl, C2H4, C2H4O,ethylen-o, CH3CHO,ethanol, CH3COOH, OHCH2COOH, C2H5, C2H6, CH3N2CH3, C2H5OH, CH3OCH3, CH3O2CH3, OCCN, C3H6O,propylox, C3H6O,acetone, C3H6O,propanal, CNCOCN, C4H6,1butyne, C6H14,n-hexane, C7H16,2-methylh, HCCN, HCCO, (HCOOH)2, NCN, N2H4, N2O5, C(gr)

8.5. Sample Problem #5 - HP

This sample problem is the same as the NASA Publication[3] Example #5. This example uses a typical solid propellant to demonstrate a problem that has the input of a custom reactant (CHOS) and requires the problem calculations to consider a phase change with the AL_2O_3(L) product.

Problem Type = HP

Table 31 Sample Problem #5 Parameters

Parameter	Values				
P (psia)	500	250	125	50	5

Table 32 Sample Problem #5 Reactants

Reactants	Chemical Formula	Weight %	Temperature (K)	Enthalpy (cal)
NH4CLO4(I)		72.06	298.15	
CHOS	$CH_{1.86995}O_{0.031256}S_{0.008415}$	18.58	298.15	-2999.082
AL(cr)		9.0	298.15	
MgO(cr)		0.2	298.15	
H2O(L)		0.16	298.15	

Additional information: Calorie Output. Omit Products: COOH, C2, CH3CO,acetyl, C2H5, C2H5OH, C2O, C3H4,propyne, C3H6,cyclo-, C3H8, C4, C4H6,2butyne, C4H8,tr2-butene, C4H9,n-butyl, C4H10,isobutane, C5H6,1,3cyclo-, C5H11,pentyl, CH3C(CH3)2CH3, C6H6, C6H12,cyclo-, C7H8O,cresol-mx, C8H8,styrene, C8H18,isooctane, C10H21,n-decyl, HNCO, NH, (HCOOH)2, C2H, C2H4O,ethylen-o, C2H6, CCN, C3, C3H4,cyclo-, C3H6O,acetone, C3H6O,propanal, C3H6O,propylox, C3H8O,1propanol, C4H2,butadiyne, C4H6,cyclo-, C4H8,isobutene, C4H9,i-butyl, C4H10,n-butane, C5H8,cyclo-, C5H11,t-pentyl, C6H2, C6H5OH,phenol, C6H13,n-hexyl, C7H14,1-heptene, C8H10,ethylbenz, C8H18,n-octane, C12H9,o-

bipheny, HNO, NH2, CH3CHO,ethanal, CH3N2CH3, CNC, C3H3,1-propynl, C3H3,2-propynl, C3H5,allyl, C3H7,n-propyl, C3H8O,2propanol, C4H4,1,3-cyclo-, C4H8,1-butene, C4H8,cyclo-, C4H9,s-butyl, C4N2, C5H10,1-pentene, C5H12,n-pentane, C6H5,phenyl, C6H10,cyclo-, C7H7,benzyl, C7H15,n-heptyl, C8H16,1-octene, C9H19,n-nonyl, C12H10,biphenyl, HNO2, NH2OH, C7H8, C2H2,vinylidene, CH3COOH, CH3OCH3, C2N2, C3H4,allene, C3H6,propylene, C3H7,i-propyl, C3O2, C4H6,butadiene, C4H8,cis2-buten, (CH3COOH)2, C4H9,t-butyl, C5, C5H10,cyclo-, C5H12,i-pentane, C6H5O,phenoxy, C6H12,1-hexene, C7H8, C7H16,n-heptane, C8H17,n-octyl, C10H8,naphthale, HNO3, NCN, C8H18,n-octane, CH2CO,ketene, (HCOOH)2, C2H3,vinyl, HCCN, N2H2, HCHO,formaldehy, NH2NO2, H2O(cr), HCOOH, N2H4, H2O(L), H2O2

Results:

Table 33 Sample Problem #5 Output #1

Thermodynamic Properties					
P, ATM	34.02285	17.01142	8.505712	3.402285	0.340228
T, K	2722.945	2706.49	2686.113	2652.957	2540.793
RHO, G/CC	0.003523	0.001769	0.000889	0.000359	3.71E-05
H, CAL/G	-484.764	-484.764	-484.764	-484.764	-484.764
U, CAL/G	-718.619	-717.627	-716.376	-714.306	-707.108
G, CAL/G	-7367.36	-7487.02	-7594.66	-7716.86	-7920.02
S, CAL/(G)(K)	2.527628	2.58721	2.64691	2.72605	2.926352
M, (1/n)	23.13863	23.09677	23.04674	22.96747	22.7085
MW, MOL WT	22.28892	22.24945	22.20298	22.12991	21.89108
(dLV/dLP)t	-1.00277	-1.00354	-1.00449	-1.006	-1.011
(dLV/dLT)p	1.052002	1.069402	1.090514	1.124605	1.242728
Cp, CAL/(G)(K)	0.578093	0.609492	0.648825	0.714566	0.958524
GAMMAs	1.192808	1.187502	1.18141	1.172528	1.149418
SON VEL,M/SEC	1080.325	1075.63	1069.981	1061.179	1034.065

Table 34 Sample Problem #5 Output #2

Mole Fractions					
ALCL	0.000184	0.000234	0.000289	0.000364	0.000506
ALCL2	2.62E-05	2.41E-05	2.16E-05	1.8E-05	9.29E-06
ALCL3	5.43E-05	3.75E-05	2.57E-05	1.53E-05	3.9E-06
***ALO**	7.21E-07	1.2E-06	1.91E-06	3.26E-06	8.3E-06
ALOCL	1.38E-05	1.74E-05	2.13E-05	2.66E-05	3.58E-05
ALOH	0.000159	0.000201	0.000246	0.000307	0.000414
ALOHCL	3.74E-05	3.41E-05	3.03E-05	2.49E-05	1.22E-05
ALOHCL2	0.000285	0.000196	0.000133	7.85E-05	1.93E-05
AL(OH)2	1.07E-05	9.64E-06	8.49E-06	6.87E-06	3.21E-06
AL(OH)2CL	9.35E-05	6.36E-05	4.28E-05	2.49E-05	5.83E-06
AL(OH)3	3.17E-05	2.15E-05	1.44E-05	8.37E-06	1.95E-06
***CO**	0.264538	0.264018	0.263392	0.262384	0.258942
COS	5.12E-05	4.24E-05	3.34E-05	2.27E-05	6.7E-06
***CO2**	0.017781	0.017816	0.017866	0.01796	0.018395
***CL**	0.001664	0.00221	0.002886	0.003996	0.007864
***H**	0.005879	0.007809	0.010207	0.01416	0.028049
HCN	5.84E-06	2.91E-06	1.45E-06	5.78E-07	5.67E-08
HCO	5.81E-06	3.93E-06	2.63E-06	1.52E-06	3.46E-07
HCL	0.131863	0.13136	0.130629	0.129342	0.12473
***H2**	0.32153	0.320711	0.319698	0.317989	0.311618
H2O	0.146502	0.145827	0.145019	0.14374	0.139621
H2S	0.001397	0.001156	0.000912	0.000621	0.000183
***Mg**	2.88E-05	5.05E-05	8.51E-05	0.000158	0.00047
MgCL	6.45E-05	8.21E-05	0.000101	0.000125	0.000143
MgCL2	0.000991	0.000948	0.000891	0.000789	0.000447
***MgO**	9.36E-07	1.57E-06	2.5E-06	4.22E-06	9.11E-06
MgOH	6.81E-06	8.47E-06	1.01E-05	1.2E-05	1.16E-05
Mg(OH)2	1.32E-05	1.21E-05	1.07E-05	8.58E-06	3.47E-06
NH3	1.45E-05	7.31E-06	3.7E-06	1.51E-06	1.62E-07

Table 34 (cont.)

Mole Fractions					
***NO**	2.58E-05	3.32E-05	4.17E-05	5.41E-05	8.49E-05
***N2**	0.068329	0.068209	0.068065	0.067836	0.06709
***O**	8.1E-06	1.41E-05	2.37E-05	4.45E-05	0.00016
***OH**	0.000766	0.001	0.001279	0.001713	0.002999
***O2**	1.49E-06	2.59E-06	4.35E-06	8.14E-06	2.92E-05
***S**	8.54E-05	0.000131	0.000186	0.000268	0.000436
SH	0.000579	0.000649	0.000686	0.000675	0.000456
SO	0.000152	0.000233	0.000335	0.000488	0.000829
SO2	5.87E-05	9.12E-05	0.000133	0.000198	0.000369
S2	2.47E-05	3.24E-05	3.83E-05	4.05E-05	2.56E-05
AL2O3(L)	0.036722	0.036686	0.036611	0.036467	0.035996

Products considered with mole fractions less than 0.000005: AL, ALC, ALC2, ALH, ALHCL, ALHCL2, ALH2, ALH2CL, ALH3, ALN, ALOCL2, ALO2, ALS, ALS2, AL2, AL2C2, AL2CL6, AL2O, AL2O2, AL2O3, AL2S, AL2S2, C, CCL, CCL2, CCL3, CCL4, CH, CHCL, CHCL2, CHCL3, CH2, CH2CL, CH2CL2, CH3, CH3CL, CH2OH, CH3O, CH4, CH3OH, CH3OOH, CN, CNN, COCL, COCL2, COHCL, CS, CS2, C2CL, C2CL2, C2CL3, C2CL4, C2CL6, C2HCL, C2HCL3, C2H2,acetylene, C2H2CL2, O(CH)2O, HO(CO)2OH, C2H3CL, CH2CL-COOH, CH3CN, C2H4, OHCH2COOH, CH3O2CH3, OCCN, C2S2, CNCOCN, C3OS, C3S2, C4H6,1butyne, C6H14,n-hexane, C7H16,2-methylh, CLCN, CLO, CLO2, CL2, CL2O, HALO, HALO2, HCCO, HNC, HOCL, HO2, H2SO4, MgH, MgN, MgS, Mg2, N, NCO, NOCL, NO2, NO2CL, NO3, N2O, N2O3, N2O4, N2O5, N3, N3H, O3, SCL, SCL2, SN, SO2CL2, SO3, S2CL2, S2O, S3, S4, S5, S6, S7, S8, AL(cr), AL(L), ALCL3(cr), ALCL3(L), ALH3(a), ALN(cr), ALN(L), AL(OH)3(a), AL2O3(a), AL2S3(a), AL2S3(b), AL2S3(L), AL4C3(cr), C(gr), H2SO4(L), Mg(cr), Mg(L), MgAL2O4(cr), MgAL2O4(L), MgCO3(cr), MgCO3(L), MgCL2(cr), MgCL2(L), MgH2(b), MgH2(L), MgO(cr), MgO(L), Mg(OH)2(cr), Mg(OH)2(L), MgS(cr), MgS(L), MgSO4(II), MgSO4(I), MgSO4(L), Mg3N2(cr), NH4CL(II), NH4CL(III), S(a), S(b), S(L), SCL2(L), S2CL2(L)

8.6. Sample Problem #6 - TP

This sample problem is the same as the NASA Publication[3] Example #14 without the debug output. This problem was chosen to recreate to highlight the ability of the program to create standard plotting output since this problem has a clear transition from when H_2O will begin to condense for lower temperatures. Additionally, this demonstrates the difference in the calculations of the molecular weight M (1/n) which only considers gaseous species in the denominator which MW considers all species.

Problem Type = TP

Table 35 Sample Problem #6 Parameters

Parameter	Values							
T (K)	1000	500	350	305	304.3	304.2	304	300
P (atm)	0.05							

Table 36 Sample Problem #6 Reactants

Reactants	Fuel / Oxidant	Moles	
H2(L)	Fuel	100	
O2(L)	Oxidant	60	

Additional information: SI Units Output. Plot Output T and Mole Fractions

Results:

Table 37 Sample Problem #6 Output

Thermodynamic Properties								
P, BAR	0.05066	0.05066	0.05066	0.05066	0.05066	0.05066	0.05066	0.05066
T, K	1000	500	350	305	304.3	304.2	304	300
RHO, KG/M^3	0.01175	0.02350	0.03358	0.03853	0.04499	0.04715	0.05146	0.13035
H, KJ/KG	-10066	-11044	-11309	-11387	-11709	-11798	-11953	-12989
U, KJ/KG	-10497	-11259	-11460	-11518	-11822	-11906	-12052	-13028
G, KJ/KG	-23602	-17140	-15356	-14841	-14833	-14832	-14830	-14801
S, KJ/(KG)(K)	13.5356	12.1924	11.5620	11.3239	10.2656	9.97256	9.46301	6.04165
M, (1/n)	19.2865	19.2865	19.2865	19.2865	22.4660	23.5411	25.6756	64.1795
MW, MOL WT	19.2865	19.2865	19.2865	19.2865	19.2865	19.2865	19.2865	19.2865
(dLV/dLP)t	-1	-1	-1	-1	-9.4432	-9.0120	-8.2628	-3.306
(dLV/dLT)p	1.00000	1	1	1	146.990	139.591	126.738	41.6028
Cp, KJ/(KG)(K)	2.11084	1.80692	1.73699	1.72331	936.243	848.837	707.196	95.8701
GAMMAs	1.25665	1.31335	1.33012	1.33362	1.10805	1.10594	1.10179	1.03439
SON VEL, M/SEC	736.036	532.067	447.993	418.753	353.253	344.707	329.339	200.504
Mole Fractions								
H2O	0.90909	0.90909	0.90909	0.90909	0.76757	0.72836	0.66025	0.2096
***O2**	0.09091	0.09091	0.09091	0.09091	0.09091	0.09091	0.09091	0.09091
H2O(L)	0	0	0	0	0.14152	0.18073	0.24884	0.69949

Products considered with mole fractions less than 0.000005: H, HO2, H2, H2O2, O, OH, O3, H2O(cr)

Figure 24 was created using the plot formatted output of the temperature and mole fraction of the outputs. This shows the impact of the mixture temperature on condensed species in the reaction.

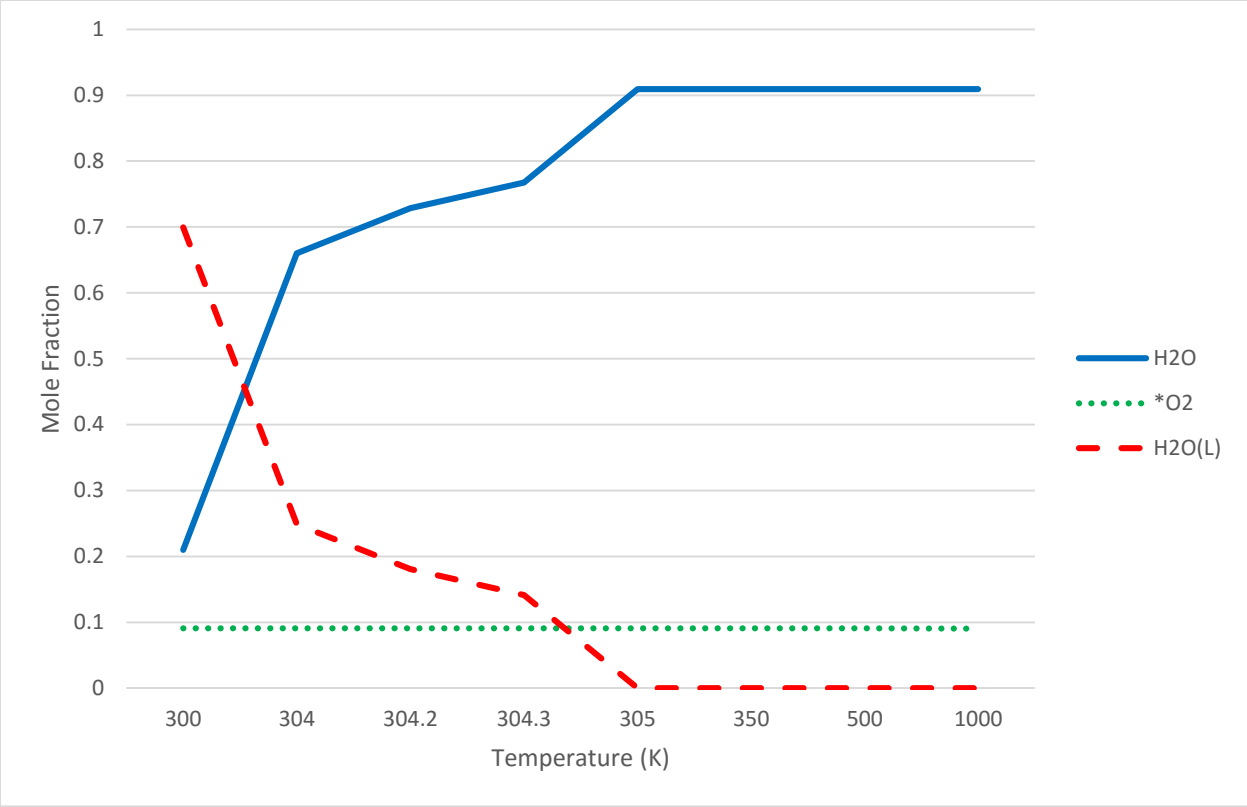

Figure 24 Mole Fractions with Increasing Temperature

8.7. Sample Problem #7 – TP with Ions

This problem starts at the first state in Sample Problem #6 and includes additional higher temperatures to display the program output when ionic species are considered.

Problem Type = TP

Table 38 Sample Problem #7 Parameters

Parameter	Values		
T (K)	1000	3000	5000
P (atm)	0.05		

Table 39 Sample Problem #7 Reactants

Reactants	Fuel / Oxidant	Moles	
H2(L)	Fuel	100	
O2(L)	Oxidant	60	

Additional information: SI Units Output. Ions

Results:

Table 40 Sample Problem #7 Output

Thermodynamic Properties			
P, BAR	0.050663	0.050663	0.050663
T, K	5000	3000	1000
RHO, KG/CU M	0.000809	0.002422	0.011752
H, KJ/KG	49390.99	11174.2	-10066
U, KJ/KG	43127.69	9082.431	-10497.1
G, KJ/KG	-117471	-55488.7	-23601.6
S, KJ/(KG)(K)	33.37231	22.22097	13.53561
M, (1/n)	6.637479	11.9246	19.28651
(dLV/dLP)t	-1.00119	-1.18117	-1
(dLV/dLT)p	1.013547	4.537313	1.000004
Cp, KJ/(KG)(K)	3.391106	51.30076	2.110843
GAMMAs	1.608456	1.109441	1.256652
SON VEL,M/SEC	3173.995	1523.383	736.0361
Mole Fractions			
***e-**	1.48E-05	1.03E-08	0
***H**	0.624207	0.267076	0
***H+**	9.62E-06	1.2E-12	0
HO2	0	1.54E-05	0
***H2**	0.000476	0.14448	0
H2O	5.04E-07	0.213171	0.90909
***O**	0.374582	0.147457	0
***OH**	0.00056	0.141766	7.6E-07
***O2**	0.000145	0.086035	0.090909

Products considered with mole fractions less than 0.000005: H-, HO2, H2+, H2-, H2O+, H2O2, H3O+, O+, O-, OH+, OH-, O2+, O2-

8.8. Sample Problem #8 – SP

Sample Problem #8 uses the first case in Sample Problem #6 where the H_2O has condensed and increases the pressure at small intervals to get a similar phase transition for the constant entropy problem. Using the same values for the initial conditions confirms the problem calculates will produce the same output for the TP and SP problem types.

Problem Type = SP

Table 41 Sample Problem #8 Parameters

Parameter	Values							
P (atm)	0.05	0.33	0.61	0.89	1.16	1.44	1.72	2.00
S/R	1.2347							

Table 42 Sample Problem #8 Reactants

Reactants	Fuel / Oxidant	Moles	
H2(L)	Fuel	100	
O2(L)	Oxidant	60	

Additional information: SI Units Output.

Results:

Table 43 Sample Problem #8 Output

Thermodynamic Properties								
P, BAR	0.05066	0.33293	0.61519	0.89745	1.17971	1.46198	1.74424	2.0265
T, K	304.300	342.431	357.437	367.467	375.159	382.223	399.182	414.120
RHO, KG/M^3	0.04498	0.24248	0.41697	0.58069	0.73725	0.88724	1.01357	1.13511
H, KJ/KG	-11709	-11475	-11388	-11331	-11288	-11253	-11223	-11197
U, KJ/KG	-11822	-11612	-11535	-11485	-11448	-11418	-11395	-11375
G, KJ/KG	-14833	-14991	-15057	-15103	-15139	-15177	-15321	-15448
S, KJ/(KG)(K)	10.2659	10.2659	10.2659	10.2659	10.2659	10.2659	10.2659	10.2659
M, (1/n)	22.4647	20.7371	20.1432	19.7691	19.4935	19.2865	19.2865	19.2865
MW, MOL WT	19.2865	19.2865	19.2865	19.2865	19.2865	19.2865	19.2865	19.2865
(dLV/dLP)t	-9.4438	-10.231	-10.532	-10.732	-10.883	-1	-1	-1
(dLV/dLT)p	146.999	137.684	134.220	131.959	130.253	1	1	1
Cp, KJ/(KG)(K)	936.353	813.403	770.349	743.012	722.766	1.74945	1.75672	1.76347
GAMMAs	1.10805	1.12848	1.13719	1.14325	1.14803	1.32701	1.32521	1.32356
SON VEL, M/SEC	353.264	393.620	409.608	420.343	428.606	467.613	477.551	486.102
Mole Fractions								
H2O	0.76762	0.83914	0.86656	0.88468	0.89847	0.90909	0.90909	0.90909
***O2**	0.09091	0.09091	0.09091	0.09091	0.09091	0.09091	0.09091	0.09091
H2O(L)	0.14148	0.06995	0.04253	0.02441	0.01062	0	0	0

Products considered with mole fractions less than 0.000005: H, HO2, H2, H2O2, O, OH, O3, H2O(cr)

8.9. Sample Problem #9 – SV

The final sample problem demonstrates the consistency of the output between the entropy and volume problem with the enthalpy and pressure problem using the same starting conditions as Sample Problem #3 and #4. This problem shows a mixture that is expanding isentropically towards the exhaust state and the impact on the species thermodynamic properties and mole fractions.

Problem Type = SV

Table 44 Sample Problem #9 Parameters

Parameter	Values				
RHO (KG/CU M)	14.428	5	0.5		
S/R	0.9824				
O/F	17				

Table 45 Sample Problem #9 Reactants

Reactants	Fuel / Oxidant	Weight Fraction	Temperature (K)
C7H8(L)	Fuel	0.4	298.15
C8H18(L), n-octane	Fuel	0.6	298.15
Air	Oxidant	1	700

Additional information: SI unit Output. Trace = 1 x 10^{-15}. Omit Products: CCN, CNC, C3H5,allyl, C3O2, C4H4,1,3-cyclo-, C4H8,tr2-butene, (CH3COOH)2, C4H9,s-butyl, C4H10,n-butane, C5H6,1,3cyclo-, C5H10,cyclo-, C5H12,n-pentane, C6H6, C6H12,1-hexene, C7H7,benzyl, C7H14,1-heptene, C8H8,styrene, C8H17,n-octyl, C7H8, C2N2, C3H6,propylene, C3H7,n-propyl, C4, C4H6,butadiene, C4H8,isobutene, C4H9,n-butyl, C4H9,t-butyl, C4N2, C5H8,cyclo-, C5H11,pentyl, C5H12,i-pentane, C6H5OH,phenol, C6H12,cyclo-, C7H8, C7H15,n-heptyl, C8H10,ethylbenz, C8H18,isooctane, C8H18,n-octane, C2O, C3H6,cyclo-, C3H7,i-propyl, C4H2,butadiyne, C4H6,2butyne, C4H8,cyclo-, C4H9,i-butyl, C4H10,isobutane, C5, C5H10,1-pentene, C5H11,t-pentyl, CH3C(CH3)2CH3, C6H10,cyclo-, C6H13,n-hexyl, C7H8O,cresol-mx, C7H16,n-heptane, C8H16,1-octene, C8H18,n-octane, C3H4,allene, C3H3,1-propynl, C3H3,2-propynl, C3H8O,2propanol, C3H8O,1propanol, C4H6,cyclo-, C4H8,1-butene, C4H8,cis2-buten, C3H8, C10H21,n-decyl, C12H10,biphenyl, C12H9,o-bipheny, C6H2, C6H5,phenyl, C6H5O,phenoxy, C10H8,naphthale, C9H19,n-nonyl, C6H6, C3H4,propyne, C3H4,cyclo-, C3, H2O(cr), H2O(L)

Results:

Table 46 Sample Problem #9 Output #1

Thermodynamic Properties					
P, BAR	99.98055	26.82751	1.419882		
T, K	2418.701	1875.605	992.8066		
RHO, KG/CU M	14.428	5	0.5		
H, KJ/KG	317.906	-488.106	-1662.44		
U, KJ/KG	-375.056	-1024.66	-1946.42		
G, KJ/KG	-19438.5	-15808.4	-9771.86		
S, KJ/(KG)(K)	8.168175	8.168175	8.168175		
M, (1/n)	29.02081	29.06482	29.06825		
(dLV/dLP)t	-1.00068	-1.00004	-1		
(dLV/dLT)p	1.018935	1.001536	1		
Cp, KJ/(KG)(K)	1.609452	1.41403	1.240008		
GAMMAs	1.225695	1.254525	1.299834		
SON VEL,M/SEC	921.6072	820.4362	607.5543		

Table 47 Sample Problem #9 Output #2

Mole Fractions							
***Ar**	0.008862	0.008875	0.008876	**NCO**	8.32E-11	4.01E-14	3.32E-27
***CN**	5.86E-14	6.96E-19	0	***NH**	2.94E-09	3.86E-12	9.44E-24
***CO**	0.001678	5.84E-05	2.73E-11	**NH2**	1.71E-09	3.15E-12	5.71E-23
***CO2**	0.115368	0.117165	0.117238	**NH3**	3.89E-09	2.62E-11	1.61E-19
COOH	5.16E-08	4.69E-10	7.15E-18	**NH2OH**	1.02E-11	3.33E-14	4.62E-23
***H**	2.76E-05	4.53E-07	9.41E-15	***NO**	0.006777	0.001894	1.05E-05
HCN	1.06E-11	1.15E-15	1.79E-31	**NO2**	2.35E-05	7.84E-06	2.66E-07
HCO	7.38E-10	3.38E-13	1.15E-26	**NO3**	1.92E-10	1.8E-11	1.33E-14
HNC	1.02E-12	4.51E-17	1.98E-34	***N2**	0.735473	0.739047	0.740081
HNCO	1.23E-09	2.72E-12	1.76E-22	**N2H2**	5.17E-13	3.17E-16	3.72E-28
HNO	4.22E-07	1.17E-08	1.66E-14	**NH2NO2**	2.44E-15	1.2E-17	4.42E-25
HNO2	1.85E-06	3.54E-07	2.86E-09	**N2O**	3.64E-06	5.42E-07	1E-09
HNO3	1.13E-09	2.37E-10	9.07E-12	**N2O3**	2.44E-11	8.86E-13	5.17E-17
HO2	7.99E-06	8.95E-07	2.27E-10	**N2O4**	3.1E-15	1.57E-16	1.24E-19
***H2**	0.000251	1.26E-05	3.59E-11	**N3**	1.23E-12	1.09E-15	3.41E-27
HCHO,formaldehy	1.71E-11	8.28E-15	7.79E-28	**N3H**	3.85E-13	3.16E-16	1.47E-27
HCOOH	6.46E-09	6.72E-11	4.77E-18	***O**	0.000155	7.86E-06	1.82E-11
H2O	0.10277	0.104194	0.104385	***OH**	0.002349	0.000332	7.34E-08
H2O2	9.88E-07	1.1E-07	5.52E-11	***O2**	0.026252	0.028404	0.029409
***N**	1.15E-08	2.21E-11	1.41E-22	**O3**	1.21E-08	8.15E-10	5.27E-14

Products considered with mole fractions less than 0.000000000000001: C, CH, CH2, CH3, CH2OH, CH3O, CH4, CH3OH, CH3OOH, C2, C2H, C2H2,acetylene, C2H2,vinylidene, CH2CO,ketene, O(CH)2O, HO(CO)2OH, C2H3,vinyl, CH3CN, CH3CO,acetyl, C2H4, C2H4O,ethylen-o, CH3CHO,ethanol, CH3COOH, OHCH2COOH, C2H5, C2H6, CH3N2CH3, C2H5OH, CH3OCH3, CH3O2CH3, OCCN, C3H6O,propylox, C3H6O,acetone, C3H6O,propanal, CNCOCN, C4H6,1butyne, C6H14,n-hexane, C7H16,2-methylh, HCCN, HCCO, (HCOOH)2, NCN, N2H4, N2O5, C(gr)

References

[1] PROGRAM CEA2 – Chemical Equilibrium with Applications. May 21, 2004. Available from www.grc.nasa.gov/WWW/CEAWeb/

[2] Gordon, Sanford and McBride, Bonnie J. NASA Reference Publication 1311. Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications. Part I. Analysis. National Aeronautics and Space Administration. October 1994

[3] Gordon, Sanford and McBride, Bonnie J. NASA Reference Publication 1311. Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications. Part II. User Manual and Program Description. National Aeronautics and Space Administration. October 1994

Appendix A: Detailed Test Cases

The following is a check list style table that describes the functional test cases that were tested to determine necessary bug fixes. All test cases required a passing result to consider program development complete. Status represents the final program check results.

Screen	Test Case	Status
CEA Tab	Clicking Main Menu button displays Main Menu	PASS
Main Menu	Clicking Thermo Ranges button displays Temperature ranges and date	PASS
Themo Ranges	Clicking the Back Button returns to Main Menu	PASS
Main Menu	Clicking Parameters displays the Program Parameters	PASS
Parameters	Changing each parameter and selecting save or save and close updates the program parameters	PASS
Parameters	Default – Large loads default values into each parameter	PASS
Parameters	Default – Small loads default values into each parameter	PASS
Parameters	Saving a non-numeric value displays error message	PASS
Main Menu	Clicking View Products / Reactants button displays Thermodynamic Data screen	PASS
Thermo Data	Changing the Products/Reactants drop down updates list box	PASS
Thermo Data	Update From File button displays warning	PASS
Thermo Data	Update From File remove all product and reactants and loads the thermodynamic data from an input file	PASS
Thermo Data	Selecting a Product or Reactant from the list enables the View Button	PASS
Thermo Data	Clicking the View button brings up the thermodynamic details about that reactant	PASS
Thermo Data	Thermodynamic Properties Data match those in the thermodynamic input file	PASS
Thermo Properties	THERMO button enabled / disabled depending on product selection	PASS
Thermo Properties	Back button returns to the Thermo Data screen	PASS
Main Menu	Clicking on the Input button display the Input and Output screen	PASS

Screen	Test Case	Status
Input / Output	When no reactants in Selected list, status reads Insert Reactants	PASS
Input / Output	All reactants display in Available List	PASS
Input / Output	Search button filters Available Reactants List	PASS
Input / Output	Search “” displays all reactants in Available Reactants List	PASS
Input / Output	Add Button moves highlighted reactants to selected list	PASS
Input / Output	Create button displays Create Temporary Reactant screen	PASS
Custom Reactant	Error message displays when no name is supplied	PASS
Custom Reactant	Error message displays when supplying already used reactant name	PASS
Custom Reactant	Error message displays when not supplying chemical formula	PASS
Custom Reactant	Error message displays when not using an elemental symbol	PASS
Custom Reactant	Error message displays when not supplying a numeric coefficient	PASS
Custom Reactant	The Create Button returns to Input / Output with new reactant in the selected reactants list	PASS
Custom Reactant	The back button returns to Input / Output with no updates	PASS
Input / Output	Edit button is enabled when selecting a custom reactant in the selected reactants list	PASS
Input / Output	Edit button is disabled when selecting a library reactant in the selected reactants list	PASS
Input / Output	Selecting the Edit button brings up the custom reactant for edit	PASS
Custom Reactant	Editing custom reactant performs the same functions as create without creating a new row in the selected reactants list	PASS
Input / Output	Remove button removes reactant from the selected list	PASS
Input / Output	Status message reads Need Reactant Amounts after adding reactants	PASS
Input / Output	Properties button is enabled when there are selected reactants	PASS
Input / Output	Properties button loads with a row for each selected reactant	PASS
Properties	Test reactant temperature error messages for library reactants	PASS
Properties	Error message displays when supplying non-numeric values	PASS
Properties	Test output results do not change for using different units for input	PASS
Properties	Check enthalpy/internal-energy loads for custom reactants	PASS
Properties	Test Save and Save/Close buttons save results	PASS

Screen	Test Case	Status
Properties	Test Save and Close and Back buttons return to Input / Output screen	PASS
Input / Output	Status message changes after adding reactant amounts	PASS
Input / Output	Status message reads Need Temperature for Library Reactants when no reactant temperature is supplied for enthalpy and internal-energy problems	PASS
Input / Output	Status messages changes after adding reactant temperatures	PASS
Input / Output	Fuel-Oxidant button brings up Fuel / Oxidant screen	PASS
Fuel / Oxidant	Max / Min error message displays if entered incorrectly	PASS
Fuel / Oxidant	Error message displays if adding a non-numeric value	PASS
Fuel / Oxidant	Test output results do not change for using different ratio types	PASS
Fuel / Oxidant	Test Save and Save/Close buttons save results	PASS
Fuel / Oxidant	Test Save/Close and Back buttons return to Input / Output screen	PASS
Input / Output	Test status message updates when using fuel / oxidant ratios	PASS
Input / Output	All products display in Available List	PASS
Input / Output	Search button filters Available Products List	PASS
Input / Output	Search “” displays all products in Available Products List	PASS
Input / Output	Test Only, Omit, and Insert add products to appropriate list	PASS
Input / Output	Test adding same reactant to Only/Insert and Omit do not cause a contradiction	PASS
Input / Output	Test remove buttons remove the products from the appropriate list	PASS
Input / Output	Test Custom Filter displays error if not products are in Filter list	PASS
Input / Output	Custom Filter gives error if products in Filter are not library products	PASS
Input / Output	Custom Filter will update the Available Product List	PASS
Input / Output	Test changing the SI Units checkbox changes output units displayed	PASS
Input / Output	Test Massf checkbox changes output from mole to mass faction	PASS
Input / Output	Test Short checkbox changes length of output information	PASS
Input / Output	Test Ions checkbox makes the program consider ionic products	PASS
Input / Output	Test debug checkbox enables the multi-select list	PASS
Input / Output	Test debug and multi-select choices create Debug tab output	PASS
Input / Output	Test entering a values for trace changes product species listed	PASS

Screen	Test Case	Status
Input / Output	Test entering non-numeric trace value displays error message	PASS
Input / Output	Test selecting at least one plot parameter creates Plot output tab	PASS
Input / Output	Test each plot parameter displays in output when selected	PASS
Input / Output	Selecting Product Mole/Mass Fractions displays additional screen after the Problem Type Specific screens	PASS
Input / Output	Test Problem Type filter updates status message accordingly	PASS
Input / Output	Test Problem Type filter bring up the correct Problem Type Specific screen after hitting the Next Button	PASS
Input / Output	Adding a Case name displays in the output	PASS
Input / Output	Test the Cancel button returns to the Main Menu	PASS
Problem TP	Back Button returns to Input / Output	PASS
Problem TP	Max / Min error message displays when necessary	PASS
Problem TP	Non-numeric error message displays when necessary	PASS
Problem TP	Test output results do not change for using different units for input	PASS
Problem TP	Test error message displays when not enough information is supplied	PASS
Problem TP	Test next button begins program equilibrium calculations	PASS
Problem HP	Back Button returns to Input / Output	PASS
Problem HP	Max / Min error message displays when necessary	PASS
Problem HP	Non-numeric error message displays when necessary	PASS
Problem HP	Test output results do not change for using different units for input	PASS
Problem HP	Supplying a reactant enthalpy overrides reactant enthalpies	PASS
Problem HP	Test error message displays when not enough information is supplied	PASS
Problem HP	Test next button begins program equilibrium calculations	PASS
Problem SP	Back Button returns to Input / Output	PASS
Problem SP	Max / Min error message displays when necessary	PASS
Problem SP	Non-numeric error message displays when necessary	PASS
Problem SP	Test output results do not change for using different units for input	PASS
Problem SP	Test error message displays when not enough information is supplied	PASS
Problem SP	Test next button begins program equilibrium calculations	PASS
Problem TV	Back Button returns to Input / Output	PASS

Screen	Test Case	Status
Problem TV	Max / Min error message displays when necessary	PASS
Problem TV	Non-numeric error message displays when necessary	PASS
Problem TV	Test output results do not change for using different units for input	PASS
Problem TV	Test error message displays when not enough information is supplied	PASS
Problem TV	Test next button begins program equilibrium calculations	PASS
Problem UV	Back Button returns to Input / Output	PASS
Problem UV	Max / Min error message displays when necessary	PASS
Problem UV	Non-numeric error message displays when necessary	PASS
Problem UV	Test output results do not change for using different units for input	PASS
Problem UV	Supplying a reactant internal-energy overrides reactant energies	PASS
Problem UV	Test error message displays when not enough information is supplied	PASS
Problem UV	Test next button begins program equilibrium calculations	PASS
Problem SV	Back Button returns to Input / Output	PASS
Problem SV	Max / Min error message displays when necessary	PASS
Problem SV	Non-numeric error message displays when necessary	PASS
Problem SV	Test output results do not change for using different units for input	PASS
Problem SV	Test error message displays when not enough information is supplied	PASS
Problem SV	Test next button begins program equilibrium calculations	PASS
Plot Parameters	Plot Parameters list all species being considered	PASS
Plot Parameters	Selecting products prints their mass/mole fractions to Plot output	PASS
Output	TP Problem output matches NASA CEA Program output	PASS
Output	HP Problem output matches NASA CEA Program output	PASS
Output	SP Problem output matches NASA CEA Program output	PASS
Output	TV Problem output matches NASA CEA Program output	PASS
Output	UV Problem output matches NASA CEA Program output	PASS
Output	SV Problem output matches NASA CEA Program output	PASS

Appendix B: Date Codes

These six-character date codes are used in the thermodynamic properties input file and match the convention used in the NASA CEA Code[1].

Letters	Reference	Numbers
g	Glenn Research Center	Month/year calculated
j	NIST-JANAF Thermochemical Tables. Chase, 1998	Month/year of table
tpis	Thermodynamic Properties of Individual Substances. Gurvich, 1978, 1979, 1982, 1989, 1991, 1996	Year of volume
n	TRC Thermodynamic Tables, NIST	Month/year of table
bar	Barin: Thermochemical Data of Pure Substances. Barin, 1989	Year of volume
coda	CODATA Key Values for Thermodynamics. Cox, 1989	Year of volume
srd	Standard Reference Data	Year of J.Phys.Chem.Ref Data journal

Appendix C: Error, Warning, and Informational Messages

The following is a list of error messages, warning messages, and informational messages that will display in either a message box during problem input or printed in the output document. (_____) is used to indicate where the program will enter in the appropriate name, (#) is used with the program will enter in an appropriate numerical value.

Type	Message	Location
Error	Unable to Process Equivalence Ratios	Input
Error	(_____) Inputs Must Follow Max/Min Format	Input
Error	Assigned Values for (_____) Are Missing	Input
Error	Maximum Number of Allowable Reactants Reached.	Input
Error	Maximum Number of Allowable (_____) Products Reached	Input
Error	Custom Reactant cannot have the same name as a Library Reactant	Input
Error	Exploded Formula must use elements from the periodic table	Input
Error	Must supply a Reactant Name	Input
Error	Reactant (_____) has been defined for the temperature (#) only. Your temperature assignment (#) is more than 10 K from this value	Input
Error	Reactant (_____) was specified as (#) which is out of Range (#-# K) for this reactant.	Input
Error	Error loading Fuel to Oxidant	Input
Error	Missing Reactant Amounts	Input
Error	Product (_____) is not a library product. Remove from filter and restart.	Input
Error	No Products in Custom Filter List	Input
Error	(_____) must be numeric	Input, Parameters
Error	Error in processing thermo.inp	Thermo Data
Error	(#) CONVERGENCES FAILED TO ESTABLISH SET OF CONDENSED SPECIES	Output
Error	(#) ITERATIONS DID NOT SATISFY CONVERGENCE REQUIREMENTS FOR THE POINT (#)	Output

Type	Message	Location
Error	LOW TEMPERATURE IMPLIES A CONDENSED SPECIES SHOULD HAVE BEEN INSERTED. RESTART USING THE INSERT BUTTON ON PROBLEM INPUT	Output
Error	DID NOT CONVERGE ON ELECTRON BALANCE	Output
Error	REINSERTION OF (_____) LIKELY TO CAUSE SINGULARITY	Output
Error	THE TEMPERATURE (#) IS OUT OF RANGE FOR POINT (#)	Output
Error	CALCULATIONS STOPPED AFTER POINT (#)	Output
Error	Product Species Containing the Element (_____) Missing	Output
Error	Insufficient Storage for Products	Output
Warning	WARNING!! POINT (#) USES A REDUCED SET OF COMPONENTS. SPECIES CONTAINING THE ELIMINATED COMPONENT ARE OMITTED. IT MAY BE NECESSARY TO RERUN WITH INSERTED CONDENSED SPECIES CONTAINING COMPONENT (_____)	Output
Warning	WARNING!! RESULTS MAY BE WRONG FOR POINT (#) DUE TO LOW MOLE FRACTION OF GASES (_____)	Output
Warning	WARNING! AMOUNT MISSING FOR REACTANT (_____) PROGRAM SETS WEIGHT PERCENT = 100	Output
Warning	This will remove all Product and Reactant information and Load from File	Thermo Data
Info	PHASE CHANGE, REPLACE (_____) WITH (_____)	Output
Info	Add (Condensed Species)	Output
Info	Remove (Condensed Species)	Output
Info	(_____) INSERTED	Output
Info	TRY REMOVING CONDENSED SPECIES	Output

Appendix D: Code Listing

The following is a code listing of the subroutines used by the CEA X Program. All user forms controls will call one of the subroutines listed below for all on-click actions.

```
1
'*************************************************************************************************
**************************************************
2     'This is a the CEA X Program
3     'Created By Jake Rumel
4     'Based on the PROGRAM CEA 2 - CHEMICAL EQUILBIRUM WITH APPLICATIONS - 5/21/04
5
'*************************************************************************************************
**************************************************
6
7     'Establish Global Variables for the Worksheets
8         Public OUTPUT As Workbook
9         Public Parameter As Worksheet
10        Public Comp, Indx, Inpt, Misci, Miscl, Miscr As Worksheet
11        Public A, G, En As Worksheet
12        Public Cdata, Prtout, Reactn, Therm As Worksheet
13        Public Eta, Stc As Worksheet
14        Public Cphs, Coef, Gauss, Eqlbrm As Worksheet
15        Public Thermoinp, ther, reac, prod, out, deb, plt As Worksheet
16        Public reac_int, prod_int, reac_list, prod_list, prod_filter As Worksheet
17        Public reac std, prod std As Boolean
18        Public thermoFilePath As String
19        Public reacRow, prodrow As Integer
20        Public ProdReac, interval As Integer
21        Public CaseOk, ReacPropOk, CustOK, chkphi, chkeqrats, pfrac, fuel As Boolean
22        Public ParamOk, FuelOK As Boolean
23        Public outint, debint, ione As Integer
24        Public phi, pfuel As Double
25
26    Sub Main()
27
28    'Set Global Variables to the Correct Sheets
29    Set CEA = ThisWorkbook
30    Set Parameter = ThisWorkbook.Worksheets("PARAMETER")
31    Set Comp = ThisWorkbook.Worksheets("COMP")
32    Set Indx = ThisWorkbook.Worksheets("INDX")
33    Set Inpt = ThisWorkbook.Worksheets("INPT")
34    Set Misci = ThisWorkbook.Worksheets("MISCI")
35    Set Miscl = ThisWorkbook.Worksheets("MISCL")
36    Set Miscr = ThisWorkbook.Worksheets("MISCR")
37    Set A = ThisWorkbook.Worksheets("A")
38    Set G = ThisWorkbook.Worksheets("G")
39    Set En = ThisWorkbook.Worksheets("EN")
40    Set Cdata = ThisWorkbook.Worksheets("CDATA")
41    Set Prtout = ThisWorkbook.Worksheets("PRTOUT")
42    Set Reactn = ThisWorkbook.Worksheets("REACTN")
43    Set Therm = ThisWorkbook.Worksheets("THERM")
44    Set Eta = ThisWorkbook.Worksheets("ETA")
45    Set Stc = ThisWorkbook.Worksheets("STC")
46    Set Cphs = ThisWorkbook.Worksheets("CPHS")
47    Set Coef = ThisWorkbook.Worksheets("COEF")
48    Set Gauss = ThisWorkbook.Worksheets("GAUSS")
49    Set Eqlbrm = ThisWorkbook.Worksheets("EQLBRM")
50    Set ther = ThisWorkbook.Worksheets("THERMO")
51    Set Thermoinp = ThisWorkbook.Worksheets("THERMOINP")
52    Set reac = ThisWorkbook.Worksheets("REACTANTS")
53    Set reac int = ThisWorkbook.Worksheets("REAC INT")
54    Set reac_list = ThisWorkbook.Worksheets("REAC_LIST")
55    Set prod = ThisWorkbook.Worksheets("PRODUCTS")
56    Set prod_int = ThisWorkbook.Worksheets("PROD_INT")
57    Set prod_list = ThisWorkbook.Worksheets("PROD_LIST")
58    Set prod_filter = ThisWorkbook.Worksheets("Filter")
```

```
59
60        Call Form Position(MainMenu)
61        MainMenu.Show
62
63    End Sub
64
65
'*****************************************************************************************************
**************************************************
66    'Subroutines for the NASA CEA X Program
67    'Created By Jake Rumel
68
'*****************************************************************************************************
**************************************************
69
70    Sub THERMP Sub()
71    'Assigned Thermodynamic States. HP, SP, TP, UV, SV, and TV Problems.
72
73    'Set Variables
74        Dim uv, tv, sv As Boolean
75
76    'Set Index Variables
77        Dim iof, Ip, It As Integer
78
79        Miscl.Cells(2, 5) = "TRUE" 'Miscl.Cells(2,5) is Eql
80        For iof = 1 To Indx.Cells(2, 9) 'Indx.Cells(2,9) is Nof
81            Miscr.Cells(2, 5) = Inpt.Cells(iof + 1, 10) 'Miscr.Cells(2,5) is Oxfl,
Inpt.Cells(iof+1,10) is Oxf(iof)
82            Call NEWOF_Sub
83    'Set Assigned P or Volume
84            For Ip = 1 To Indx.Cells(2, 12) 'Indx.Cells(2,1) is Ip, Indx.Cells(2,12) is Np
85                Indx.Cells(2, 1) = Ip
86                Miscr.Cells(2, 7) = Inpt.Cells(Indx.Cells(2, 1) + 1, 11) 'Miscr.Cells(2,7) is
Pp, Inpt.Cells(Indx.Cells(2,1)+1,11) is P(Ip)
87    'Set Assigned T
88                For It = 1 To Indx.Cells(2, 19) 'Indx.Cells(2,3) is It, Indx.Cells(2,19) is Nt
89                    Indx.Cells(2, 3) = It
90                    Miscr.Cells(2, 17) = Inpt.Cells(Indx.Cells(2, 1) + 1, 14)
'Miscr.Cells(2,17) is Vv, Inpt.Cells(Indx.Cells(2,1)+1,14) is V(Ip)
91                    Miscr.Cells(2, 15) = Inpt.Cells(Indx.Cells(2, 3) + 1, 13)
'Miscr.Cells(2,15) is Tt, Inpt.Cells(Indx.Cells(Indx.Cells(2,3)+1,13) is T(It)
92                    Call EQLBRM_Sub
93                    If (Indx.Cells(2, 14) = 0) Then GoTo 200 'Indx.Cells(2,14) is Npt
94                    Misci.Cells(2, 3) = 0 'Misci.Cells(2,3) is Isv
95                    If (CInt(Indx.Cells(2, 1)) <> CInt(Indx.Cells(2, 12)) Or CInt(Indx.Cells(2,
3)) <> CInt(Indx.Cells(2, 19)) And CDbl(Miscr.Cells(2, 15)) <> 0) Then
96                        Misci.Cells(2, 3) = Indx.Cells(2, 14)
97                        If (CInt(Indx.Cells(2, 14)) <> CInt(Parameter.Cells(3, 2))) Then GoTo
10
98                    End If
99    'OUTPUT
100                   outint = outint + 2
101                   Call Statement_Format
102                   If Not Miscl.Cells(2, 7) Then
103                       out.Cells(outint, 3) = "THERMODYNAMIC EQUILIBRIUM PROPERTIES AT
ASSIGNED"
104                   Else
105                       out.Cells(outint, 3) = "THERMODYNAMIC COMBUSTION PROPERTIES AT
ASSIGNED"
106                   End If
107                   out.Cells(outint, 3).HorizontalAlignment = xlCenter
108                   outint = outint + 1
109                   Call Statement_Format
110                   If Not Miscl.Cells(2, 19) Then
111                       If Miscl.Cells(2, 7) Then out.Cells(outint, 3) = "PRESSURES"
112                       If Miscl.Cells(2, 17) Then out.Cells(outint, 3) = "TEMPERATURE AND
PRESSURE"
113                       If Miscl.Cells(2, 16) Then out.Cells(outint, 3) = "ENTROPY AND
PRESSURE"
114                   Else
115                       If Miscl.Cells(2, 7) Then out.Cells(outint, 3) = "VOLUME"
```

```
116                         If Miscl.Cells(2, 17) Then out.Cells(outint, 3) = "TEMPERATURE AND
VOLUME"
117                         If Miscl.Cells(2, 16) Then out.Cells(outint, 3) = "ENTROPY AND VOLUME"
118                     End If
119                     out.Cells(outint, 3).HorizontalAlignment = xlCenter
120                     outint = outint + 2
121                     Call Problem_Output
122                     If (Indx.Cells(2, 2) + Indx.Cells(2, 14) > 500) Then
123                         Indx.Cells(2, 2) = 500
124                     Else
125                         Indx.Cells(2, 2) = Indx.Cells(2, 2) + Indx.Cells(2, 14)
126                     End If
127                     If (CInt(Misci.Cells(2, 3)) = 0 And iof = CInt(Indx.Cells(2, 9))) Then GoTo
200
128                     Indx.Cells(2, 14) = 0
129   10:               Indx.Cells(2, 14) = Indx.Cells(2, 14) + 1
130                     If (Miscl.Cells(2, 17) = False And Miscr.Cells(2, 15) <> 0) Then
Inpt.Cells(2, 13) = Miscr.Cells(2, 15)
131                     If (Indx.Cells(2, 19) = 1 And Indx.Cells(2, 12) = 1) Then GoTo 100
132                     If (Indx.Cells(2, 1) = 1 And Indx.Cells(2, 3) = 1) Then Misci.Cells(2, 3) =
-Misci.Cells(2, 3)
133                     If (Indx.Cells(2, 19) <> 1) Then
134                         If (Indx.Cells(2, 3) = Indx.Cells(2, 19) Or Miscr.Cells(2, 15) = 0)
Then Misci.Cells(2, 3) = 0
135                     End If
136                     Call SETEN_Sub
137                 Next It
138             Next Ip
139   100: Next iof
140   200:
141   End Sub
142
143   Sub EQLBRM Sub()
144   'Calculate Equilibrium Composition and Properties
145
146   'Establish Local Variables
147       Dim ae, amb As String
148       Dim cpcalc, i2many, newcom, reduce As Boolean
149       Dim i, il, ilamb, ilamb1, inc, ipr, iq2, iter, ix, ixsing, iz As Integer
150       Dim j, ja, jb, jbx, jc, jcondi, jcons, jdelg, jex, jj, jkg, jneg, jsw, jnew As Integer
151       Dim k, kc, kg, kk, kmat, kneg, l, le, lelim, lk, ll, lncvg, ls, lsing, lz As Integer
152       Dim maxitn, ncvg, njc, nn, numb As Integer
153       Dim aa, ambda, ambda1, bigen, bigneg, delg, dlnt, dpie As Double
154       Dim ensol, esize, gap, gasfrc, pie, siz9, sizeg, smalno, smnol, sum, sum1, szgj As
Double
155       Dim tem, tmelt, tsize, ween, xi, xln, xsize As Double
156
157   'Initialize Variables
158       smalno = 10 ^ (-6)
159       smnol = -13.815511
160       lsing = 0
161       jsw = 0
162       jdelg = 0
163       maxitn = 50
164       ncvg = 0
165       lncvg = 3 * Indx.Cells(2, 7) 'Indx.Cells(2,7) is Nlm
166       reduce = False
167       siz9 = Miscr.Cells(2, 10) - 9.2103404 'Miscr.Cells(2,10) is Size
168       tsize = Miscr.Cells(2, 10)
169       xsize = Miscr.Cells(2, 10) + 6.90775528
170       If (CDbl(Miscr.Cells(2, 14)) <> 0) Then 'Miscr.Cells(2,14) is Trace
171           maxitn = maxitn + Indx.Cells(2, 15) / 2 'Indx.Cells(2,15) is Ngc
172           xsize = -Log(Miscr.Cells(2, 14))
173           If (xsize < CDbl(Miscr.Cells(2, 14))) Then xsize = Miscr.Cells(2, 14) + 0.1
174       End If
175       If (xsize > 80) Then xsize = 80
176       If ((xsize + 6.90775528) < 80) Then
177           esize = xsize + 6.90775528
178       Else
179           esize = 80
180       End If
```

```
181       jcons = 0
182       pie = 0
183       i2many = False
184       Miscl.Cells(2, 12) = False 'Miscl.Cells(2,12) is Pderiv
185       Miscl.Cells(2, 1) = False 'Miscl.Cells(2,1) is Convg
186       numb = 0
187    'DEBUG
188       If Miscl.Cells(Indx.Cells(2, 14) + 1, 2) Then
189           deb.Range("C" & debint, "M" & debint).Merge
190           deb.Cells(debint, 3).HorizontalAlignment = xlCenter
191           deb.Cells(debint, 3) = "DEBUG - " & Indx.Cells(2, 14)
192           debint = debint + 2
193       End If
194       cpcalc = True
195       If (Miscl.Cells(2, 17)) Then cpcalc = False 'Miscl.Cells(2,17) is Tp
196       If (CDbl(Miscr.Cells(2, 15)) <> 0) Then 'Miscr.Cells(2,15) is Tt
197           'Indx.Cells(2,13) is Npr, Inpt.Cells(2,13) is T(1)
198           If (CInt(Indx.Cells(2, 13)) = 0 Or (CDbl(Miscr.Cells(2, 15)) <> CDbl(Inpt.Cells(2,
13)) And Miscl.Cells(2, 17) = False)) Then GoTo 400
199           k = 1
200       Else
201           Miscr.Cells(2, 15) = 3800
202           If (CInt(Indx.Cells(2, 13)) = 0) Then GoTo 400
203           k = 1
204       End If
205    100: j = Indx.Cells(k + 1, 20) ' Indx.Cells(k+1, 20) is Jcond(k)
206       jc = j - Indx.Cells(2, 5) 'Indx.Cells(2,5) is Ng
207       kg = -Indx.Cells(jc + 1, 23) 'Indx.Cells(jc+1, 23) is Ifz(jc)
208       For i = 1 To 9
209           kg = kg + 1
210           kc = jc + kg
211           If (Miscr.Cells(2, 15) <= Therm.Cells(3, kc + 16)) Then 'Therm.Cells(3,kc+16) is
Temp(2,kc)
212               If (kg <> 0) Then
213                   Indx.Cells(k + 1, 20) = j + kg
214                   En.Cells(j + kg, Indx.Cells(2, 14)) = En.Cells(j, Indx.Cells(2, 14)) '
Indx.Cells(2,14) is Npt, En.Cells(...) is En(...)
215                   En.Cells(j, Indx.Cells(2, 14)) = 0
216    'OUTPUT
217                   If (CStr(Cdata.Cells(j + 1, 10)) <> CStr(Cdata.Cells(j + kg + 1, 10)) And
Not Miscl.Cells(2, 14)) Then
218                       Call Statement_Format
219                       out.Cells(outint, 3) = "PHASE CHANGE, REPLACE " & CStr(Cdata.Cells(j +
1, 10)) & " WITH " & CStr(Cdata.Cells(j + kg + 1, 10))
220                       outint = outint + 1
221                   End If
222               End If
223               GoTo 300
224           ElseIf (kc >= CInt(Indx.Cells(2, 4)) Or CInt(Indx.Cells(kc + 2, 23)) <=
CInt(Indx.Cells(kc + 1, 23))) Then 'Indx.Cells(2,4) is Nc
225               GoTo 200
226           End If
227       Next i
228    200: If (Miscl.Cells(2, 17) = False) Then
229           Miscr.Cells(2, 15) = Therm.Cells(3, kc + 16) - 10
230           k = 1
231           GoTo 100
232       End If
233    'OUTPUT
234       Call Statement_Format
235       out.Cells(outint, 3) = "Remove " & CStr(Cdata.Cells(j + 1, 10))
236       outint = outint + 1
237       En.Cells(j, Indx.Cells(2, 14)) = 0
238       Comp.Cells(j + 1, 6) = 0 'Comp.Cells(j+1,6) is Enln(j)
239       Comp.Cells(j + 1, 5) = 0 'Comp.Cells(j+1,5) is Deln(j)
240       For i = k To Indx.Cells(2, 13)
241           Indx.Cells(i + 1, 20) = Indx.Cells(i + 2, 20)
242       Next i
243       Indx.Cells(2, 13) = Indx.Cells(2, 13) - 1
244    300: k = k + 1
245       If (k <= CInt(Indx.Cells(2, 13))) Then GoTo 100
```

```
246   400: Miscr.Cells(2, 12) = Log(Miscr.Cells(2, 15)) 'Miscr.Cells(2,12) is Tln
247       'Miscl.Cell(2,19) is Vol, Miscr.Cells(2,7) is Pp, Miscr.Cell(2,9) is Rr,
Comp.Cells(2,1) is Enn, Miscr.Cells(2,17) is Vv
248       If (Miscl.Cells(2, 19)) Then Miscr.Cells(2, 7) = Miscr.Cells(2, 9) * Comp.Cells(2, 1) *
Miscr.Cells(2, 15) / Miscr.Cells(2, 17)
249       Call CPHS_Sub
250       Miscr.Cells(2, 13) = Log(CDbl(Miscr.Cells(2, 7)) / CDbl(Comp.Cells(2, 1)))
'Miscr.Cells(2,13) is Tm
251       le = Indx.Cells(2, 7)
252       If (CInt(Misci.Cells(2, 6)) <> 0 And CInt(Indx.Cells(2, 7)) <> CInt(Misci.Cells(2, 6)))
Then 'Misci.Cells(2,6) is Lsave
253           tem = Exp(-tsize)
254           For i = Misci.Cells(2, 6) + 1 To Indx.Cells(2, 7)
255               For j = 1 To Indx.Cells(2, 5)
256                   If (CDbl(A.Cells(i, j)) <> 0) Then
257                       En.Cells(j, Indx.Cells(2, 14)) = tem
258                       Comp.Cells(j + 1, 6) = -tsize
259                   End If
260               Next j
261           Next i
262       End If
263       ls = Indx.Cells(2, 7)
264       lelim = 0
265       lz = ls
266       If (Miscl.Cells(2, 8)) Then lz = ls - 1 'Miscl.Cells(2,8) is Ions
267   'OUTPUT
268       If ((CInt(Indx.Cells(2, 14)) = 1) And (Miscl.Cells(2, 13) = False) And (Miscl.Cells(2,
14) = False)) Then
269           out.Cells(outint, 1) = "POINT"
270           out.Cells(outint, 1).HorizontalAlignment = xlCenter
271           out.Cells(outint, 2) = "ITN"
272           out.Cells(outint, 2).HorizontalAlignment = xlCenter
273           out.Cells(outint, 3) = "T"
274           out.Cells(outint, 3).HorizontalAlignment = xlCenter
275           For i = 1 To Indx.Cells(2, 7)
276               out.Cells(outint, 3 + i) = CStr(Cdata.Cells(i + 1, 1))
277               out.Cells(outint, 3 + i).HorizontalAlignment = xlCenter
278           Next i
279           outint = outint + 1
280       End If
281   'DEBUG
282       If Miscl.Cells(Indx.Cells(2, 14) + 1, 2) Then
283           For i = 1 To Indx.Cells(2, 7)
284               Eqlbrm.Cells(i + 1, 1) = Cdata.Cells(i + 1, 1)
285           Next i
286       End If
287
288   'Begin Iteration
289   500: If (cpcalc) Then
290           Therm.Cells(2, 1) = 0 'Therm.Cells(2,1) is Cpsum
291           For j = 1 To Indx.Cells(2, 5)
292               Therm.Cells(2, 1) = Therm.Cells(2, 1) + En.Cells(j, Indx.Cells(2, 14)) *
Therm.Cells(j + 1, 3) 'Therm.Cells(j+1,3) is Cp(j)
293           Next j
294           If (CInt(Indx.Cells(2, 13)) <> 0) Then
295               For k = 1 To Indx.Cells(2, 13)
296                   j = CInt(Indx.Cells(k + 1, 20).Value)
297                   Therm.Cells(2, 1) = Therm.Cells(2, 1) + En.Cells(j, Indx.Cells(2, 14)) *
Therm.Cells(j + 1, 3)
298               Next k
299               cpcalc = False
300           End If
301       End If
302       numb = numb + 1
303       Call MATRIX_Sub
304       iq2 = Misci.Cells(2, 2) + 1 'Misci.Cells(2,2) is Iq1
305       If (Miscl.Cells(2, 1)) Then Misci.Cells(2, 1) = Misci.Cells(2, 1) - 1 '
Misci.Cells(2,1) is Imat
306   'DEBUG
307       If (Miscl.Cells(Indx.Cells(2, 14) + 1, 2)) Then 'Miscl.Cells(Indx.Cells(2, 14) + 1, 2)
is Debug(Npt)
```

```
308           kmat = Misci.Cells(2, 1) + 1
309           If (Miscl.Cells(2, 1) = False) Then
310               deb.Range("A" & debint, deb.Cells(debint, kmat)).Merge
311               deb.Cells(debint, 1).HorizontalAlignment = xlCenter
312               deb.Cells(debint, 1) = "ITERATION " & CStr(numb) & " MATRIX"
313               debint = debint + 2
314           Else
315               If (Miscl.Cells(2, 12) = False) Then
316                   deb.Range("A" & debint, deb.Cells(debint, kmat)).Merge
317                   deb.Cells(debint, 1).HorizontalAlignment = xlCenter
318                   deb.Cells(debint, 1) = "T Deriv Matrix"
319                   debint = debint + 2
320               End If
321               If (Miscl.Cells(2, 12)) Then
322                   deb.Range("A" & debint, deb.Cells(debint, kmat)).Merge
323                   deb.Cells(debint, 1).HorizontalAlignment = xlCenter
324                   deb.Cells(debint, 1) = "P Derix Matrix"
325                   debint = debint + 2
326               End If
327           End If
328           For i = 1 To Misci.Cells(2, 1)
329               For k = 1 To kmat
330                   deb.Cells(debint, k) = G.Cells(i, k)
331               Next k
332               debint = debint + 1
333           Next i
334           debint = debint + 1
335       End If
336       Misci.Cells(2, 7) = 0 'Misci.Cells(2,7) is Msing
337       Call GAUSS_Sub
338       If (CInt(Misci.Cells(2, 7)) = 0) Then
339    'DEBUG INFO
340       If (Miscl.Cells(Indx.Cells(2, 14) + 1, 2)) Then
341           deb.Range("A" & debint, "B" & debint).Merge
342           deb.Cells(debint, 1).HorizontalAlignment = xlCenter
343           deb.Cells(debint, 1) = "SOLUTION VECTOR"
344           debint = debint + 1
345           For k = 1 To le
346               deb.Cells(debint, k + 1) = Cdata.Cells(k + 1, 1)
347           Next k
348           debint = debint + 1
349           For i = 1 To Misci.Cells(2, 1)
350               deb.Cells(debint, i + 1) = Miscr.Cells(i + 1, 20)
351           Next i
352           debint = debint + 2
353       End If
354           If (Miscl.Cells(2, 1) = False) Then
355    'Obtain Corrections to the Estimates
356               If (Miscl.Cells(2, 19)) Then Miscr.Cells(iq2 + 1, 20) =
Miscr.Cells(Misci.Cells(2, 2) + 1, 20) ' Miscr.Cells(iq2+1,20) is X(iq2)
357               If (Miscl.Cells(2, 17)) Then Miscr.Cells(iq2 + 1, 20) = 0
358               dlnt = Miscr.Cells(iq2 + 1, 20)
359               sum = Miscr.Cells(Misci.Cells(2, 2) + 1, 20)
360               If (Miscl.Cells(2, 19)) Then
361                   Miscr.Cells(Misci.Cells(2, 2) + 1, 20) = 0
362                   sum = -dlnt
363               End If
364               For j = 1 To Indx.Cells(2, 5)
365                   If (lelim <> 0) Then
366                       Comp.Cells(j + 1, 5) = 0
367                       For i = lelim To ls
368                           If (CDbl(A.Cells(i, j)) <> 0) Then GoTo 520
369                       Next i
370                   End If
371                   Comp.Cells(j + 1, 5) = -Therm.Cells(j + 1, 6) + Therm.Cells(j + 1, 4) *
dlnt + sum 'Therm.Cells(j+1,6) is Mu(j), Therm.Cells(j+1,4) is H0
372                   For k = 1 To Indx.Cells(2, 7)
373                       Comp.Cells(j + 1, 5) = Comp.Cells(j + 1, 5) + A.Cells(k, j) *
Miscr.Cells(k + 1, 20)
374                   Next k
```

```
375                     If (pie <> 0) Then Comp.Cells(j + 1, 5) = Comp.Cells(j + 1, 5) +
A.Cells(ls, j) * pie
376   520:          Next j
377                 If (CInt(Indx.Cells(2, 13)) <> 0) Then
378                     For k = 1 To Indx.Cells(2, 13)
379                         j = Indx.Cells(k + 1, 20)
380                         kk = Indx.Cells(2, 7) + k
381                         Comp.Cells(j + 1, 5) = Miscr.Cells(kk + 1, 20)
382                     Next k
383                 End If
384   'Calculate Control factor, Ambda
385                 ambda = 1
386                 ambda1 = 1
387                 ilamb = 0
388                 ilamb1 = 0
389                 If (Abs(CDbl(Miscr.Cells(Misci.Cells(2, 2) + 1, 20))) > Abs(dlnt)) Then
390                     sum = Abs(Miscr.Cells(Misci.Cells(2, 2) + 1, 20))
391                 Else
392                     sum = Abs(dlnt)
393                 End If
394                 sum = sum * 5
395                 For j = 1 To Indx.Cells(2, 5)
396                     If (CDbl(Comp.Cells(j + 1, 5)) > 0) Then
397                         If (CDbl(Comp.Cells(j + 1, 6) - Comp.Cells(2, 2) + Miscr.Cells(2, 10))
<= 0) Then 'Comp.Cells(2,2) is Ennl
398                             sum1 = Abs(Comp.Cells(j + 1, 5) - Miscr.Cells(Misci.Cells(2, 2) +
1, 20))
399                             If (sum1 >= siz9) Then
400                                 sum1 = Abs(-9.2103404 - Comp.Cells(j + 1, 6) + Comp.Cells(2,
2)) / sum1
401                                 If (sum1 < ambda1) Then
402                                     ambda1 = sum1
403                                     ilamb1 = j
404                                 End If
405                             End If
406                         ElseIf (CDbl(Comp.Cells(j + 1, 5)) > sum) Then
407                             sum = Comp.Cells(j + 1, 5)
408                             ilamb = j
409                         End If
410                     End If
411                 Next j
412                 If (sum > 2) Then ambda = 2 / sum
413                 If (ambda1 <= ambda) Then
414                     ambda = ambda1
415                     ilamb = ilamb1
416                 End If
417                 If (Miscl.Cells(Indx.Cells(2, 14) + 1, 2)) Then
418   'DEBUG
419                     deb.Range("A" & debint, "B" & debint).Merge
420                     deb.Cells(debint, 1).HorizontalAlignment = xlCenter
421                     deb.Cells(debint, 1) = "T"
422                     deb.Range("C" & debint, "D" & debint).Merge
423                     deb.Cells(debint, 3).HorizontalAlignment = xlCenter
424                     deb.Cells(debint, 3) = "ENN"
425                     deb.Range("E" & debint, "F" & debint).Merge
426                     deb.Cells(debint, 5).HorizontalAlignment = xlCenter
427                     deb.Cells(debint, 5) = "ENNL"
428                     deb.Range("G" & debint, "H" & debint).Merge
429                     deb.Cells(debint, 7).HorizontalAlignment = xlCenter
430                     deb.Cells(debint, 7) = "PP"
431                     deb.Range("I" & debint, "J" & debint).Merge
432                     deb.Cells(debint, 9).HorizontalAlignment = xlCenter
433                     deb.Cells(debint, 9) = "LN P/N"
434                     deb.Range("K" & debint, "L" & debint).Merge
435                     deb.Cells(debint, 11).HorizontalAlignment = xlCenter
436                     deb.Cells(debint, 11) = "AMBDA"
437                     debint = debint + 1
438                     deb.Range("A" & debint, "B" & debint).Merge
439                     deb.Cells(debint, 1).HorizontalAlignment = xlCenter
440                     deb.Cells(debint, 1) = CDbl(Miscr.Cells(2, 15))
441                     deb.Range("C" & debint, "D" & debint).Merge
```

```
442                     deb.Cells(debint, 3).HorizontalAlignment = xlCenter
443                     deb.Cells(debint, 3) = CDbl(Comp.Cells(2, 1))
444                     deb.Range("E" & debint, "F" & debint).Merge
445                     deb.Cells(debint, 5).HorizontalAlignment = xlCenter
446                     deb.Cells(debint, 5) = CDbl(Comp.Cells(2, 2))
447                     deb.Range("G" & debint, "H" & debint).Merge
448                     deb.Cells(debint, 7).HorizontalAlignment = xlCenter
449                     deb.Cells(debint, 7) = CDbl(Miscr.Cells(2, 7))
450                     deb.Range("I" & debint, "J" & debint).Merge
451                     deb.Cells(debint, 9).HorizontalAlignment = xlCenter
452                     deb.Cells(debint, 9) = CDbl(Miscr.Cells(2, 13))
453                     deb.Range("K" & debint, "L" & debint).Merge
454                     deb.Cells(debint, 11).HorizontalAlignment = xlCenter
455                     deb.Cells(debint, 11) = CDbl(ambda)
456                     debint = debint + 2
457                     If ambda <> 1 Then
458                         amb = "ENN"
459                         If (Abs(Miscr.Cells(iq2 + 1, 20)) > Abs(Miscr.Cells(Misci.Cells(2, 2) +
1, 20))) Then amb = "TEMP"
460                         If ilamb <> 0 Then amb = Cdata.Cells(ilamb + 1, 10)
461                         Call Debug_Statement_Format
462                         deb.Cells(debint, 3) = "AMBDA SET BY " & amb
463                         debint = debint + 2
464                     End If
465                     If Miscl.Cells(2, 19) Then
466                         Call Debug_Statement_Format
467                         deb.Cells(debint, 3) = "VOLUME = " & (Miscr.Cells(2, 17) * 0.001) & "
CC/G"
468                         debint = debint + 2
469                     End If
470                     Call Debug_Table_Format
471                     deb.Cells(debint, 3) = "Nj"
472                     deb.Cells(debint, 5) = "LN Nj"
473                     deb.Cells(debint, 7) = "DEL LN Nj"
474                     deb.Cells(debint, 9) = "H0j/RT"
475                     deb.Cells(debint, 11) = "S0j/R"
476                     deb.Cells(debint, 13) = "G0j/RT"
477                     deb.Cells(debint, 15) = "Gj/RT"
478                     debint = debint + 1
479                     For j = 1 To Indx.Cells(2, 15)
480                         Call Debug_Table_Format
481                         deb.Cells(debint, 1) = Cdata.Cells(j + 1, 10)
482                         deb.Cells(debint, 3) = CDbl(En.Cells(j, Indx.Cells(2, 14)))
483                         deb.Cells(debint, 5) = CDbl(Comp.Cells(j + 1, 6))
484                         deb.Cells(debint, 7) = CDbl(Comp.Cells(j + 1, 5))
485                         deb.Cells(debint, 9) = CDbl(Therm.Cells(j + 1, 4))
486                         deb.Cells(debint, 11) = CDbl(Therm.Cells(j + 1, 5))
487                         deb.Cells(debint, 13) = CDbl(Therm.Cells(j + 1, 4) - Therm.Cells(j + 1,
5))
488                         deb.Cells(debint, 15) = CDbl(Therm.Cells(j + 1, 6))
489                         debint = debint + 1
490                     Next j
491                     debint = debint + 1
492                 End If
493   'Apply Corrections to Estimates
494                 Prtout.Cells(Indx.Cells(2, 14) + 1, 8) = 0 ' Prtout.Cells(Indx.Cells(2,14)+1,8)
is Totn(Npt)
495                 For j = 1 To Indx.Cells(2, 5)
496                     Comp.Cells(j + 1, 6) = Comp.Cells(j + 1, 6) + ambda * Comp.Cells(j + 1, 5)
497                 Next j
498                 For j = 1 To Indx.Cells(2, 5)
499                     En.Cells(j, Indx.Cells(2, 14)) = 0
500                     If (lelim <> 0) Then
501                         For i = lelim To ls
502                             If (CDbl(A.Cells(i, j)) <> 0) Then GoTo 540
503                         Next i
504                     End If
505                     If (CDbl(Comp.Cells(j + 1, 6) - Comp.Cells(2, 2) + tsize) > 0) Then
506                         En.Cells(j, Indx.Cells(2, 14)) = Exp(Comp.Cells(j + 1, 6))
507                         Prtout.Cells(Indx.Cells(2, 14) + 1, 8) = Prtout.Cells(Indx.Cells(2, 14)
+ 1, 8) + En.Cells(j, Indx.Cells(2, 14))
```

```
508                     End If
509   540:          Next j
510                 If (Miscl.Cells(2, 8) And CStr(Cdata.Cells(Indx.Cells(2, 7) + 1, 1)) = "E")
Then 'Cdata.Cells(Indx.Cells(2,7)+1,1) is Elmt(Nlm)
511                     For j = 1 To Indx.Cells(2, 5)
512                         If (CDbl(A.Cells(ls, j)) <> 0 And CDbl(En.Cells(j, Indx.Cells(2, 14)))
= 0) Then
513                             If (CDbl(Comp.Cells(j + 1, 6) - Comp.Cells(2, 2) + esize) > 0) Then
514                                 En.Cells(j, Indx.Cells(2, 14)) = Exp(Comp.Cells(j + 1, 6))
515                                 Prtout.Cells(Indx.Cells(2, 14) + 1, 8) =
Prtout.Cells(Indx.Cells(2, 14) + 1, 8) + En.Cells(j, Indx.Cells(2, 14))
516                             End If
517                         End If
518                     Next j
519                 End If
520                 Comp.Cells(2, 4) = Prtout.Cells(Indx.Cells(2, 14) + 1, 8) 'Comp.Cells(2,4) is
Sumn
521                 If (CInt(Indx.Cells(2, 13)) <> 0) Then
522                     For k = 1 To Indx.Cells(2, 13)
523                         j = Indx.Cells(k + 1, 20)
524                         En.Cells(j, Indx.Cells(2, 14)) = En.Cells(j, Indx.Cells(2, 14)) + ambda
* Comp.Cells(j + 1, 5)
525                         Prtout.Cells(Indx.Cells(2, 14) + 1, 8) = Prtout.Cells(Indx.Cells(2, 14)
+ 1, 8) + En.Cells(j, Indx.Cells(2, 14))
526                     Next k
527                 End If
528                 If (Miscl.Cells(2, 17) = False) Then
529                     Miscr.Cells(2, 12) = Miscr.Cells(2, 12) + ambda * dlnt
530                     Miscr.Cells(2, 15) = Exp(Miscr.Cells(2, 12))
531                     cpcalc = True
532                     Call CPHS_Sub
533                 End If
534                 If (Miscl.Cells(2, 19)) Then
535                     Comp.Cells(2, 1) = Comp.Cells(2, 4)
536                     Comp.Cells(2, 2) = Log(Comp.Cells(2, 1))
537                     Miscr.Cells(2, 7) = Miscr.Cells(2, 9) * Miscr.Cells(2, 15) * Comp.Cells(2,
1) / Miscr.Cells(2, 17)
538                 Else
539                     Comp.Cells(2, 2) = Comp.Cells(2, 2) + ambda * Miscr.Cells(Misci.Cells(2, 2)
+ 1, 20)
540                     Comp.Cells(2, 1) = Exp(Comp.Cells(2, 2))
541                 End If
542                 Miscr.Cells(2, 13) = Log(Miscr.Cells(2, 7) / Comp.Cells(2, 1))
543                 If (CStr(Cdata.Cells(Indx.Cells(2, 7) + 1, 1)) = "E") Then
544    'Check on Removing Ions
545                     For j = 1 To Indx.Cells(2, 15)
546                         If (CDbl(A.Cells(Indx.Cells(2, 7), j)) <> 0) Then
547                             If (CDbl(En.Cells(j, Indx.Cells(2, 14))) > 0) Then GoTo 560
548                         End If
549                     Next j
550                     pie = Miscr.Cells(Indx.Cells(2, 7) + 1, 20)
551                     lelim = CInt(Indx.Cells(2, 7))
552                     Indx.Cells(2, 7) = Indx.Cells(2, 7) - 1
553                     GoTo 500
554                 End If
555    'Test for Convergence
556    560:         If (numb > maxitn) Then
557    'OUTPUT
558                     Call Statement_Format
559                     out.Cells(outint, 3) = maxitn & " ITERATIONS DID NOT SATISFY CONVERGENCE
REQUIREMENTS FOR THE POINT " & Indx.Cells(2, 14)
560                     outint = outint + 2
561                     If (CInt(Indx.Cells(2, 4)) = 0 Or i2many) Then GoTo 1500
562                     i2many = True
563                     If (Miscl.Cells(2, 7) = False Or CInt(Indx.Cells(2, 14)) <> 1 Or
CDbl(Miscr.Cells(2, 15)) > 100) Then 'Miscl.Cells(2,7) is Hp
564                         If (CInt(Indx.Cells(2, 13)) <> 1 Or CDbl(Comp.Cells(2, 1)) > 0.0001)
Then GoTo 1500
565    'High Temperature, Included Condensed Condition
566    'OUTPUT
567                         Call Statement_Format
```

```
568                         out.Cells(outint, 3) = "TRY REMOVING CONDENSED SPECIES"
569                         outint = outint + 1
570                         Comp.Cells(2, 1) = 0.1
571                         Comp.Cells(2, 2) = -2.3025851
572                         Comp.Cells(2, 4) = Comp.Cells(2, 1)
573                         xi = Indx.Cells(2, 5)
574                         xi = Comp.Cells(2, 1) / xi
575                         xln = Log(xi)
576                         For j = 1 To Indx.Cells(2, 5)
577                             En.Cells(j, Indx.Cells(2, 14)) = xi
578                             Comp.Cells(j + 1, 6) = xln
579                         Next j
580                         j = Indx.Cells(2, 20)
581                         k = 1
582                         GoTo 1000
583                     Else
584   'OUTPUT
585                         Call Statement_Format
586                         out.Cells(outint, 3) = "LOW TEMPERATURE IMPLIES A CONDENSED SPECIES
SHOULD HAVE BEEN INSERTED"
587                         outint = outint + 1
588                         Call Statement Format
589                         out.Cells(outint, 3) = "RESTART USING THE INSERT BUTTON ON PROBLEM
INPUT"
590                         outint = outint + 2
591                         GoTo 1500
592                     End If
593                 Else
594                     sum = (Miscr.Cells(Misci.Cells(2, 2) + 1, 20) * Comp.Cells(2, 1) /
Prtout.Cells(Indx.Cells(2, 14) + 1, 8))
595                     If (Abs(sum) > 0.000005) Then GoTo 500
596                     For j = 1 To Indx.Cells(2, 5)
597                         If (Abs(Comp.Cells(j + 1, 5) * En.Cells(j, Indx.Cells(2, 14)) /
Prtout.Cells(Indx.Cells(2, 14) + 1, 8)) > 0.000005) Then GoTo 500
598                     Next j
599                     If (Abs(dlnt) > 0.0001) Then GoTo 500
600                     If (CInt(Indx.Cells(2, 13)) <> 0) Then
601                         For k = 1 To Indx.Cells(2, 13)
602                             j = Indx.Cells(k + 1, 20)
603                             If (Abs(Comp.Cells(j + 1, 5) / Prtout.Cells(Indx.Cells(2, 14) + 1,
8)) > 0.000005) Then GoTo 500
604                             If (CDbl(En.Cells(j, Indx.Cells(2, 14))) < 0) Then GoTo 700
605                         Next k
606                     End If
607                     le = Indx.Cells(2, 7)
608                     For i = 1 To Indx.Cells(2, 7)
609                         If (Abs(Miscr.Cells(i + 1, 19)) >= 0.000001) Then 'Miscr.Cells(i+1,19)
is B0(i)
610                             sum = 0
611                             For j = 1 To Indx.Cells(2, 15)
612                                 sum = sum + En.Cells(j, Indx.Cells(2, 14)) * A.Cells(i, j)
613                             Next j
614                                 If (Abs(Miscr.Cells(i + 1, 19) - sum) > CDbl(Inpt.Cells(2, 3)))
Then GoTo 500  'Inpt.Cells(2,3) is Bcheck
615                         End If
616                     Next i
617                     If (CDbl(Miscr.Cells(2, 14)) <> 0) Then
618                         tsize = xsize
619                         tem = 1
620                         If (numb <> 1) Then
621                             lk = lz
622                             If (CInt(Indx.Cells(2, 7)) < lz) Then lk = Indx.Cells(2, 7)
623                             For i = 1 To lk
624                                 If (i <> lsing) Then
625                                     tem = 0
626                                     If (CDbl(Miscr.Cells(i + 1, 20)) <> 0) Then
627                                         tem = Abs((Eqlbrm.Cells(i + 1, 3) - Miscr.Cells(i + 1,
20)) / Miscr.Cells(i + 1, 20))
628                                         If (tem > 0.001) Then GoTo 565
629                                     End If
630                                 End If
```

```
631                             Next i
632                         End If
633   565:                  For i = 1 To Indx.Cells(2, 7)
634                             Eqlbrm.Cells(i + 1, 3) = Miscr.Cells(i + 1, 20)
635                         Next i
636                         If (tem > 0.001) Then GoTo 500
637                         If (Miscl.Cells(2, 8)) Then
638   'Check on Electron Balance
639                             iter = 1
640                             If (pie <> 0) Then
641                                 le = Indx.Cells(2, 7) + 1
642                                 Miscr.Cells(le + 1, 20) = pie
643                             End If
644   566:                      sum1 = 0
645                             sum = 0
646                             pie = Miscr.Cells(le + 1, 20)
647                             For j = 1 To Indx.Cells(2, 5)
648                                 If (CDbl(A.Cells(ls, j)) <> 0) Then
649                                     En.Cells(j, Indx.Cells(2, 14)) = 0
650                                     tem = 0
651                                     If (CDbl(Comp.Cells(j + 1, 6)) > -87) Then tem =
Exp(Comp.Cells(j + 1, 6))
652                                     If (CDbl(Comp.Cells(j + 1, 6) - Comp.Cells(2, 2) + tsize) >
0 And CStr(Cdata.Cells(Indx.Cells(2, 7) + 1, 1)) = "E") Then
653                                         pie = 0
654                                         En.Cells(j, Indx.Cells(2, 14)) = tem
655                                     End If
656                                     aa = A.Cells(ls, j) * tem
657                                     sum = sum + aa
658                                     sum1 = sum1 + aa * A.Cells(ls, j)
659                                 End If
660                             Next j
661                             If (sum1 <> 0) Then
662                                 dpie = -sum / sum1
663                                 For j = 1 To Indx.Cells(2, 5)
664                                     If (CDbl(A.Cells(ls, j)) <> 0) Then Comp.Cells(j + 1, 6) =
Comp.Cells(j + 1, 6) + A.Cells(ls, j) * dpie
665                                 Next j
666   'DEBUG
667                                 If Miscl.Cells(Indx.Cells(2, 14) + 1, 2) Then
668                                     Call Debug_Statement_Format
669                                     deb.Cells(debint, 3) = "ELECTRON BALANCE ITER NO. = " &
iter & " DELTA PI = " & dpie
670                                     debint = debint + 2
671                                 End If
672                                 If (Abs(dpie) > 0.0001) Then
673                                     Miscr.Cells(le + 1, 20) = Miscr.Cells(le + 1, 20) + dpie
674                                     iter = iter + 1
675                                     If (iter <= 80) Then GoTo 566
676   'OUTPUT
677                                     Call Statement_Format
678                                     out.Cells(outint, 3) = "DID NOT CONVERGE ON ELECTRON
BALANCE"
679                                     outint = outint + 2
680                                     GoTo 1500
681                                 ElseIf (CStr(Cdata.Cells(Indx.Cells(2, 7) + 1, 1)) = "E" And
pie <> 0) Then
682                                     Indx.Cells(2, 7) = Indx.Cells(2, 7) - 1
683                                     newcom = True
684                                 End If
685                             End If
686                         End If
687                     End If
688                 End If
689             ElseIf (Miscl.Cells(2, 12) = False) Then
690   'Temperature Derivatives--Convg = T, Pderiv = F
691                 Prtout.Cells(Indx.Cells(2, 14) + 1, 3) = 1 - Miscr.Cells(Misci.Cells(2, 2) + 1,
20) 'Prtout.Cells(Indx.Cells(2,14)+1,3) is Dlvtp(Npt)
692                 Prtout.Cells(Indx.Cells(2, 14) + 1, 1) = G.Cells(iq2, iq2)
'Prtout.Cells(Indx.Cells(2,14)+1,1) is Cpr(Npt)
693                 For j = 1 To Misci.Cells(2, 2)
```

```
694                     Prtout.Cells(Indx.Cells(2, 14) + 1, 1) = Prtout.Cells(Indx.Cells(2, 14) +
1, 1) - G.Cells(iq2, j) * Miscr.Cells(j + 1, 20)
695                 Next j
696    'Pressure Derivative--Convg = T, Pderiv = T
697                 Miscl.Cells(2, 12) = True
698                 GoTo 500
699             Else
700                 Prtout.Cells(Indx.Cells(2, 14) + 1, 2) = -1 + Miscr.Cells(Misci.Cells(2, 2) +
1, 20) 'Prtout.Cells(Indx.Cells(2,14)+1,2) is Dlvpt(Npt)
701                 If (CInt(Misci.Cells(2, 4)) = 0) Then 'Misci.Cells(2,4) is Jliq
702                     Prtout.Cells(Indx.Cells(2, 14) + 1, 4) = -1 / (Prtout.Cells(Indx.Cells(2,
14) + 1, 2) + ((Prtout.Cells(Indx.Cells(2, 14) + 1, 3)) ^ 2) * Comp.Cells(2, 1) /
Prtout.Cells(Indx.Cells(2, 14) + 1, 1)) 'Prtout.Cells(Indx.Cells(2,14)+1,4) is Gammas(Npt)
703                 Else
704                     En.Cells(Misci.Cells(2, 5), Indx.Cells(2, 14)) = ensol 'Misci.Cells(2,5) is
Jsol
705                     Prtout.Cells(Indx.Cells(2, 14) + 1, 5) = Prtout.Cells(Indx.Cells(2, 14) +
1, 5) + En.Cells(Misci.Cells(2, 4), Indx.Cells(2, 14)) * (Therm.Cells(Misci.Cells(2, 4) + 1, 4) -
Therm.Cells(Misci.Cells(2, 5) + 1, 4)) 'Prtout.Cells(Indx.Cells(2,14)+1,5) is Hsum(Npt)
706                     Prtout.Cells(Indx.Cells(2, 14) + 1, 4) = -1 / Prtout.Cells(Indx.Cells(2,
14) + 1, 2)
707                     Indx.Cells(2, 13) = Indx.Cells(2, 13) + 1
708                     Indx.Cells(Indx.Cells(2, 13) + 1, 20) = Misci.Cells(2, 4)
709                 End If
710                 GoTo 1400
711             End If
712    'Singular Matrix
713         Else
714             If (Miscl.Cells(2, 1)) Then
715    'OUTPUT
716                 Call Statement_Format
717                 out.Cells(outint, 3) = "DERIVATIVE MATRIX SINGULAR"
718                 outint = outint + 1
719                 Prtout.Cells(Indx.Cells(2, 14) + 1, 2) = -1
720                 Prtout.Cells(Indx.Cells(2, 14) + 1, 3) = 1
721                 Prtout.Cells(Indx.Cells(2, 14) + 1, 1) = Therm.Cells(2, 1)
722                 Prtout.Cells(Indx.Cells(2, 14) + 1, 4) = -1 / (Prtout.Cells(Indx.Cells(2, 14) +
1, 2) + ((Prtout.Cells(Indx.Cells(2, 14) + 1, 3)) ^ 2) * Comp.Cells(2, 1) /
Prtout.Cells(Indx.Cells(2, 14) + 1, 1))
723                 GoTo 1400
724             Else
725    'OUTPUT
726                 Call Statement_Format
727                 out.Cells(outint, 3) = "SINGULAR MATRIX, ITERATION " & numb & " VARIABLE " &
Misci.Cells(2, 7)
728                 outint = outint + 1
729                 lsing = Misci.Cells(2, 7)
730                 ixsing = ixsing + 1
731                 If (ixsing <= 8) Then
732                     xsize = 80
733                     tsize = xsize
734                     If (CInt(Misci.Cells(2, 7)) > CInt(Indx.Cells(2, 7)) And numb < 1 And
CInt(Indx.Cells(2, 13)) > 1 And jdelg > 0) Then
735                         ween = 1000
736                         j = 0
737                         For i = 1 To Indx.Cells(2, 13)
738                             jcondi = Indx.Cells(i + 1, 20)
739                             If (jcondi <> jdelg) Then
740                                 For ll = 1 To Indx.Cells(2, 7)
741                                     If (CDbl(A.Cells(ll, jdelg)) <> 0 And CDbl(A.Cells(ll,
jcondi)) <> 0) Then
742                                         If (CDbl(En.Cells(jcondi, Indx.Cells(2, 14))) <= ween)
Then
743
744                                             ween = En.Cells(jcondi, Indx.Cells(2, 14))
745                                             j = jcondi
746                                             k = i
747                                         End If
748                                         GoTo 570
749                                     End If
750                                 Next ll
                            End If
```

```
570:                Next i
                    If (j > 0) Then
'OUTPUT
                        Call Statement Format
                        out.Cells(outint, 3) = "TRY REMOVING CONDENSED SPECIES"
                        outint = outint + 1
                        GoTo 1000
                    End If
                ElseIf (Miscl.Cells(2, 7) = False Or CInt(Indx.Cells(2, 14)) <> 1 Or
CInt(Indx.Cells(2, 4)) = 0 Or CDbl(Miscr.Cells(2, 15)) > 100) Then
                    If (ixsing >= 3) Then
                        If (CInt(Misci.Cells(2, 7)) < CInt(Misci.Cells(2, 2))) Then
                            If (reduce And CInt(Misci.Cells(2, 7)) <= CInt(Indx.Cells(2,
7))) Then
                                If (CInt(Indx.Cells(2, 7)) < lelim) Then GoTo 1500
'OUTPUT
                                Call Statement_Format
                                out.Cells(outint, 3) = "WARNING!! POINT " & Indx.Cells(2,
14) & "USES A REDUCED SET OF COMPONENTS"
                                outint = outint + 1
                                Call Statement_Format
                                out.Cells(outint, 3) = "SPECIES CONTAINING THE ELIMINATED
COMPONENT ARE OMITTED."
                                outint = outint + 1
                                Call Statement Format
                                out.Cells(outint, 3) = "IT MAY BE NECESSARY TO RERUN WITH
INSERTED CONDENSED SPECIES"
                                outint = outint + 1
                                Call Statement_Format
                                out.Cells(outint, 3) = "CONTAINING COMPONENT " &
Cdata.Cells(Indx.Cells(2, 7) + 1, 1)
                                outint = outint + 1
                                Indx.Cells(2, 7) = Indx.Cells(2, 7) - 1
                                GoTo 500
                            ElseIf (CInt(Misci.Cells(2, 7)) <= CInt(Indx.Cells(2, 7))) Then
'Find New Components
                                If (Miscl.Cells(2, 8) = False) Then GoTo 1100
                                If (CStr(Cdata.Cells(Indx.Cells(2, 7) + 1, 1)) <> "E") Then
GoTo 1100
                                For j = 1 To Indx.Cells(2, 5)
                                    If (CDbl(A.Cells(Indx.Cells(2, 7), j)) <> 0) Then
En.Cells(j, Indx.Cells(2, 14)) = 0
                                Next j
                                pie = Miscr.Cells(Indx.Cells(2, 7) + 1, 20)
                                Indx.Cells(2, 7) = Indx.Cells(2, 7) - 1
                                If (CInt(Misci.Cells(2, 7)) > CInt(Indx.Cells(2, 7))) Then
GoTo 500
                                GoTo 1100
                            Else
'Remove Condensed Species To Correct Singularity
                                k = Misci.Cells(2, 7) - Indx.Cells(2, 7)
                                j = Indx.Cells(k + 1, 20)
                                If (j <> jcons) Then
                                    jcons = j
                                    GoTo 1000
                                End If
                            End If
                        End If
                    End If
                    For jj = 1 To Indx.Cells(2, 5)
                        If (Miscl.Cells(2, 8)) Then
                            If (CStr(Cdata.Cells(Indx.Cells(2, 7) + 1, 1)) <> "E") Then
                                If (CDbl(A.Cells(ls, jj)) <> 0) Then GoTo 575
                            End If
                        End If
                        If (CDbl(En.Cells(jj, Indx.Cells(2, 14))) = 0) Then
                            En.Cells(jj, Indx.Cells(2, 14)) = smalno
                            Comp.Cells(jj + 1, 6) = smnol
                        End If
575:                Next jj
                    GoTo 500
```

```
813                   Else
814   'OUTPUT
815                       Call Statement_Format
816                       out.Cells(outint, 3) = "LOW TEMPERATURE IMPLIES A CONDENSED SPECIES
SHOULD HAVE BEEN INSERTED"
817                       outint = outint + 1
818                       Call Statement_Format
819                       out.Cells(outint, 3) = "RESTART USING THE INSERT BUTTON ON PROBLEM
INPUT"
820                       outint = outint + 2
821                   End If
822               End If
823           End If
824           GoTo 1500
825       End If
826   'Calculate Entropy, Check on Delta S for SP Problems
827   600: Prtout.Cells(Indx.Cells(2, 14) + 1, 7) = 0 'Prtout.Cells(Indx.Cells(2,14)+1,7) is
Ssum(Npt)
828       For j = 1 To Indx.Cells(2, 5)
829           Prtout.Cells(Indx.Cells(2, 14) + 1, 7) = Prtout.Cells(Indx.Cells(2, 14) + 1, 7) +
En.Cells(j, Indx.Cells(2, 14)) * (Therm.Cells(j + 1, 5) - Comp.Cells(j + 1, 6) - Miscr.Cells(2,
13)) 'Therm.Cells(j+1,5) is S(j)
830       Next j
831       If (CInt(Indx.Cells(2, 13)) > 0) Then
832           For k = 1 To Indx.Cells(2, 13)
833               j = Indx.Cells(k + 1, 20)
834               Prtout.Cells(Indx.Cells(2, 14) + 1, 7) = Prtout.Cells(Indx.Cells(2, 14) + 1, 7)
+ En.Cells(j, Indx.Cells(2, 14)) * Therm.Cells(j + 1, 5)
835           Next k
836       End If
837       If (Miscl.Cells(2, 16) = False) Then 'Miscl.Cells(2,16) is Sp
838           Miscl.Cells(2, 1) = True
839       Else
840           tem = Prtout.Cells(Indx.Cells(2, 14) + 1, 7) - Miscr.Cells(2, 11)
'Miscr.Cells(2,11) is S0
841           If (Abs(tem) > 0.0005) Then GoTo 500
842   'DEBUG
843           If Miscl.Cells(Indx.Cells(2, 14) + 1, 2) Then
844               Call Debug Statement Format
845               deb.Cells(debint, 3) = "DELTA S/R = " & tem
846               debint = debint + 2
847           End If
848           Miscl.Cells(2, 1) = True
849       End If
850   'Convergence Tests Are Satisfied, Test Condensed Species
851   700: ncvg = ncvg + 1
852       If (ncvg > lncvg) Then
853   'Error, Set TT = 0
854   'OUTPUT
855           Call Statement Format
856           out.Cells(outint, 3) = lncvg & " CONVERGENCES FAILED TO ESTABLISH SET OF CONDENSED
SPECIES."
857           outint = outint + 2
858           GoTo 1500
859       Else
860           If (Miscl.Cells(2, 13) = False) Then
861               For il = 1 To le
862                   Eqlbrm.Cells(il + 1, 4) = Miscr.Cells(il + 1, 20)
863               Next il
864   'OUTPUT
865               If Not Miscl.Cells(2, 14) Then
866                   If newcom Then
867                       out.Cells(outint, 1) = "POINT"
868                       out.Cells(outint, 1).HorizontalAlignment = xlCenter
869                       out.Cells(outint, 2) = "ITN"
870                       out.Cells(outint, 2).HorizontalAlignment = xlCenter
871                       out.Cells(outint, 3) = "T"
872                       out.Cells(outint, 3).HorizontalAlignment = xlCenter
873                       For k = 1 To le
874                           out.Cells(outint, 3 + k) = CStr(Eqlbrm.Cells(k + 1, 1))
875                           out.Cells(outint, 3 + k).HorizontalAlignment = xlCenter
```

```
876                     Next k
877                     outint = outint + 1
878                 End If
879                 out.Cells(outint, 1) = Indx.Cells(2, 14)
880                 out.Cells(outint, 2) = numb
881                 out.Cells(outint, 3) = Miscr.Cells(2, 15)
882                 For j = 1 To le
883                     out.Cells(outint, 3 + j) = Eqlbrm.Cells(j + 1, 4)
884                 Next j
885                 outint = outint + 1
886             End If
887             If (Miscl.Cells(2, 17) = False And CInt(Indx.Cells(2, 13)) = 0 And
CDbl(Miscr.Cells(2, 15)) <= (CDbl(Therm.Cells(2, 2)) * 0.2)) Then 'Therm.Cells(2,2) is Tg(1)
888   'OUTPUT
889                 Call Statement_Format
890                 out.Cells(outint, 3) = "LOW TEMPERATURE IMPLIES A CONDENSED SPECIES SHOULD
HAVE BEEN INSERTED"
891                 outint = outint + 1
892                 Call Statement_Format
893                 out.Cells(outint, 3) = "RESTART USING THE INSERT BUTTON ON PROBLEM INPUT"
894                 outint = outint + 2
895                 GoTo 1500
896             End If
897             newcom = False
898         End If
899         If (CInt(Indx.Cells(2, 13)) <> 0) Then
900             bigneg = 0
901             jneg = 0
902             For k = 1 To Indx.Cells(2, 13)
903                 j = Indx.Cells(k + 1, 20)
904                 If ((CDbl(En.Cells(j, Indx.Cells(2, 14))) * CDbl(Therm.Cells(j + 1, 3))) <=
bigneg) Then
905                     bigneg = En.Cells(j, Indx.Cells(2, 14)) * Therm.Cells(j + 1, 3)
906                     jneg = j
907                     kneg = k
908                 End If
909             Next k
910             If (jneg <> 0) Then
911                 j = jneg
912                 k = kneg
913                 If (j = CInt(Misci.Cells(2, 5)) Or j = CInt(Misci.Cells(2, 4))) Then
914                     Misci.Cells(2, 5) = 0
915                     Misci.Cells(2, 4) = 0
916                 End If
917                 GoTo 1000
918             End If
919         End If
920         If (CInt(Indx.Cells(2, 15)) <> CInt(Indx.Cells(2, 5)) Or Miscl.Cells(2, 17)) Then
921             Indx.Cells(2, 5) = Indx.Cells(2, 15)
922             Call CPHS Sub
923             Indx.Cells(2, 5) = Indx.Cells(2, 6) - 1 'Indx.Cells(2,6) is Ngp1
924             cpcalc = True
925             If (CInt(Indx.Cells(2, 15)) = CInt(Indx.Cells(2, 5))) Then GoTo 750
926             Call ALLCON_Sub
927             If (CInt(Indx.Cells(2, 13)) <> 0 And Miscl.Cells(2, 17) = False) Then
928                 gap = 50
929                 For ipr = 1 To Indx.Cells(2, 13)
930                     j = Indx.Cells(ipr + 1, 20)
931                     If (j <> CInt(Misci.Cells(2, 5)) And j <> CInt(Misci.Cells(2, 4))) Then
932                         inc = j - Indx.Cells(2, 5)
933                         kg = -Indx.Cells(inc + 1, 23)
934                         For iz = 1 To 20
935                             kg = kg + 1
936                             kc = inc + kg
937                             If (CDbl(Miscr.Cells(2, 15)) <= CDbl(Therm.Cells(3, kc + 16)))
Then
938                                 If (kg <> 0) Then
939                                     jkg = j + kg
940                                     If (Abs(kg) > 1 Or CStr(Cdata.Cells(j + 1, 10)) =
CStr(Cdata.Cells(jkg + 1, 10))) Then GoTo 740
941                                     If (jkg = jsw) Then GoTo 720
```

```
942                                     If (CDbl(Miscr.Cells(2, 15)) < (CDbl(Therm.Cells(2, inc
+ 16)) - gap) Or CDbl(Miscr.Cells(2, 15)) > (CDbl(Therm.Cells(3, inc + 16)) + gap)) Then GoTo 740
943                                         GoTo 720
944                                     End If
945                                     GoTo 710
946                                 ElseIf (CInt(Indx.Cells(kc + 2, 23)) <= CInt(Indx.Cells(kc + 1,
23))) Then
947                                     GoTo 710
948                                 End If
949                             Next iz
950                             If (CDbl(Miscr.Cells(2, 15)) > (CDbl(Therm.Cells(3, kc + 16)) *
1.2)) Then GoTo 1000
951                         End If
952   710:              Next ipr
953               End If
954               sizeg = 0
955               szgj = 0
956               For inc = 1 To Indx.Cells(2, 4)
957                   j = inc + Indx.Cells(2, 5)
958   'DEBUG
959                   If Miscl.Cells(Indx.Cells(2, 14) + 1, 2) Then
960                       deb.Cells(debint, 1) = Cdata.Cells(j + 1, 10)
961                       deb.Cells(debint, 2) = Therm.Cells(2, inc + 16)
962                       deb.Cells(debint, 3) = Therm.Cells(3, inc + 16)
963                       deb.Cells(debint, 4) = En.Cells(j, Indx.Cells(2, 14))
964                       debint = debint + 2
965                   End If
966                   If CDbl(En.Cells(j, Indx.Cells(2, 14))) <= 0 Then
967                       If (CDbl(Miscr.Cells(2, 15)) > CDbl(Therm.Cells(2, inc + 16)) Or
CDbl(Therm.Cells(2, inc + 16)) = CDbl(Therm.Cells(2, 2))) Then
968                           If (CDbl(Miscr.Cells(2, 15)) <= CDbl(Therm.Cells(3, inc + 16)))
Then
969                               sum = 0
970                               For i = 1 To Indx.Cells(2, 7)
971                                   sum = sum + A.Cells(i, j) * Miscr.Cells(i + 1, 20)
972                               Next i
973                               delg = (Therm.Cells(j + 1, 4) - Therm.Cells(j + 1, 5) - sum) /
Therm.Cells(j + 1, 7) 'Therm.Cells(j+1,7) is Mw
974                               If (delg < sizeg And delg < 0) Then
975                                   If (j <> jcons) Then
976                                       sizeg = delg
977                                       jdelg = j
978                                   Else
979                                       szgj = delg
980                                   End If
981                                   ipr = ipr - 1
982                               End If
983   'DEBUG
984                               If Miscl.Cells(Indx.Cells(2, 14) + 1, 2) Then
985                                   deb.Range("A" & debint, "C" & debint).Merge
986                                   deb.Cells(debint, 1).HorizontalAlignment = xlRight
987                                   deb.Cells(debint, 1) = "[G0j-SUM(Aij*PIi)]/Mj ="
988                                   deb.Cells(debint, 4) = delg
989                                   deb.Range("E" & debint, "G" & debint).Merge
990                                   deb.Cells(debint, 5).HorizontalAlignment = xlRight
991                                   deb.Cells(debint, 5) = "MAX NEG DELTA G ="
992                                   deb.Cells(debint, 8) = sizeg
993                                   debint = debint + 2
994                               End If
995                           End If
996                       End If
997                   End If
998               Next inc
999               If (sizeg = 0 And szgj = 0) Then GoTo 750
1000              If (sizeg <> 0) Then
1001                  j = jdelg
1002                  GoTo 800
1003              Else
1004  'OUTPUT
1005                  Call Statement_Format
```

```
1006                 out.Cells(outint, 3) = "REINSERTION OF " & Cdata.Cells(jcons + 1, 10) & "
LIKELY TO CAUSE SINGULARITY"
1007                 outint = outint + 2
1008                 GoTo 1500
1009             End If
1010  720:       If (kg > 0) Then
1011                 kk = kg
1012             Else
1013                 kk = 0
1014             End If
1015             tmelt = Therm.Cells(kk + 2, inc + 16)
1016             Miscr.Cells(2, 15) = tmelt
1017             Miscr.Cells(2, 12) = Log(Miscr.Cells(2, 15))
1018             If (j > jkg) Then
1019                 Misci.Cells(2, 5) = jkg
1020             Else
1021                 Miscr.Cells(2, 5) = j
1022             End If
1023             Misci.Cells(2, 4) = Misci.Cells(2, 5) + 1
1024             En.Cells(jkg, Indx.Cells(2, 14)) = 0.5 * En.Cells(j, Indx.Cells(2, 14))
1025             En.Cells(j, Indx.Cells(2, 14)) = En.Cells(jkg, Indx.Cells(2, 14))
1026             j = jkg
1027             GoTo 800
1028  'Wrong Phase Included For T Interval, Switch En
1029  740:       En.Cells(jkg, Indx.Cells(2, 14)) = En.Cells(j, Indx.Cells(2, 14))
1030             Indx.Cells(ipr + 1, 20) = jkg
1031             En.Cells(j, Indx.Cells(2, 14)) = 0
1032             jsw = j
1033  'OUTPUT
1034             If (CStr(Cdata.Cells(j + 1, 10)) <> CStr(Cdata.Cells(jkg + 1, 10)) And Not
Miscl.Cells(2, 14)) Then
1035             Call Statement_Format
1036             out.Cells(outint, 3) = "PHASE CHANGE, REPLACE " & CStr(Cdata.Cells(j + 1, 10))
& " WITH " & CStr(Cdata.Cells(jkg + 1, 10))
1037             outint = outint + 1
1038             End If
1039             j = jkg
1040             GoTo 900
1041         End If
1042  'Converged with No Condensed Changes.  If Both Solid and Liquid Present,
1043  'Temporarily Remove Liquid to Prevent Singular Derivative Matrix
1044  750:    Comp.Cells(2, 4) = Comp.Cells(2, 1)
1045         If (CInt(Misci.Cells(2, 5)) <> 0) Then
1046             ensol = En.Cells(Misci.Cells(2, 5), Indx.Cells(2, 14))
1047             En.Cells(Misci.Cells(2, 5), Indx.Cells(2, 14)) = En.Cells(Misci.Cells(2, 5),
Indx.Cells(2, 14)) + En.Cells(Misci.Cells(2, 4), Indx.Cells(2, 14))
1048             Prtout.Cells(Indx.Cells(2, 14) + 1, 3) = 0
1049             Prtout.Cells(Indx.Cells(2, 14) + 1, 1) = 0
1050             Prtout.Cells(Indx.Cells(2, 14) + 1, 4) = 0
1051             Miscl.Cells(2, 12) = True
1052             For k = 1 To Indx.Cells(2, 13)
1053                 If (CInt(Indx.Cells(k + 1, 20)) = CInt(Misci.Cells(2, 4))) Then GoTo 760
1054             Next k
1055  760:       For i = k To Indx.Cells(2, 13)
1056                 Indx.Cells(i + 1, 20) = Indx.Cells(i + 2, 20)
1057             Next i
1058             Indx.Cells(2, 13) = Indx.Cells(2, 13) - 1
1059         End If
1060         GoTo 500
1061     End If
1062  'Add Condensed Species
1063  800: Indx.Cells(2, 13) = Indx.Cells(2, 13) + 1
1064     i = Indx.Cells(2, 13)
1065     For ix = 2 To Indx.Cells(2, 13)
1066         Indx.Cells(i + 1, 20) = Indx.Cells(i, 20)
1067         i = i - 1
1068     Next ix
1069     Indx.Cells(2, 20) = j
1070  'OUTPUT
1071     If Not Miscl.Cells(2, 14) Then
1072         Call Statement_Format
```

```
1073          out.Cells(outint, 3) = "ADD " & CStr(Cdata.Cells(j + 1, 10))
1074          outint = outint + 1
1075      End If
1076  900: inc = j - Indx.Cells(2, 5)
1077      Miscl.Cells(2, 1) = False
1078      If (Miscl.Cells(2, 17)) Then cpcalc = False
1079      numb = -1
1080      GoTo 500
1081  'Remove Condensed Species
1082  1000: En.Cells(j, Indx.Cells(2, 14)) = 0
1083      Comp.Cells(j + 1, 5) = 0
1084      Comp.Cells(j + 1, 6) = 0
1085      For i = k To Indx.Cells(2, 13)
1086          Indx.Cells(i + 1, 20) = Indx.Cells(i + 2, 20)
1087      Next i
1088  'OUTPUT
1089      If Not Miscl.Cells(2, 14) Then
1090          Call Statement_Format
1091          out.Cells(outint, 3) = "REMOVE " & CStr(Cdata.Cells(j + 1, 10))
1092          outint = outint + 1
1093      End If
1094      Indx.Cells(2, 13) = Indx.Cells(2, 13) - 1
1095      For i = 1 To Indx.Cells(2, 7)
1096          If (CStr(Eqlbrm.Cells(i + 1, 1)) = CStr(Cdata.Cells(j + 1, 10))) Then
1097              numb = -1
1098              Miscl.Cells(2, 1) = False
1099              If (Miscl.Cells(2, 17)) Then cpcalc = False
1100              GoTo 1100
1101          End If
1102      Next i
1103      GoTo 900
1104  1100: newcom = False
1105      nn = Indx.Cells(2, 7)
1106      If (CStr(Cdata.Cells(Indx.Cells(2, 7) + 1, 1)) = "E") Then nn = Indx.Cells(2, 7) - 1
1107  'Find Order of Species for Components - Biggest to Smallest
1108      njc = 0
1109      For lc = 1 To nn
1110          Eqlbrm.Cells(lc + 1, 2) = 0
1111      Next lc
1112  1200: bigen = -1 * 10 ^ (-35)
1113      For j = 1 To Indx.Cells(2, 5)
1114          If (CDbl(En.Cells(j, Indx.Cells(2, 14))) > bigen) Then
1115              If (Miscl.Cells(2, 8) = False Or A.Cells(ls, j) = 0) Then
1116                  bigen = En.Cells(j, Indx.Cells(2, 14))
1117                  jbx = j
1118              End If
1119          End If
1120      Next j
1121      If (bigen > 0) Then
1122          For lc = 1 To nn
1123              If (jbx = 0) Then jbx = Indx.Cells(lc + 1, 21) 'Indx.Cells(lc+1,21) is Jx(lc)
1124              If (CDbl(A.Cells(lc, jbx)) > smalno) Then
1125                  If (njc <> 0) Then
1126                      For i = 1 To njc
1127                          l = Eqlbrm.Cells(i + 1, 2)
1128                          If (l = lc) Then GoTo 1250
1129                          If (l = 0) Then GoTo 1210
1130                          j = Misci.Cells(l + 1, 8) 'Misci.Cells(2,8) is Jcm(l)
1131                          For l = 1 To nn
1132                              If (CDbl(A.Cells(l, jbx)) <> CDbl(A.Cells(l, j))) Then GoTo
1205
1133                          Next l
1134                          GoTo 1250
1135  1205:                   Next i
1136                  End If
1137  1210:           For i = 1 To nn
1138                      If (i <> lc) Then
1139                          jex = Indx.Cells(i + 1, 21)
1140                          If (Abs(A.Cells(lc, jbx) * A.Cells(i, jex) - A.Cells(lc, jex) *
A.Cells(i, jbx)) <= smalno) Then GoTo 1250
1141                      End If
```

```
1142                 Next i
1143                 njc = njc + 1
1144                 If (jbx <> CInt(Misci.Cells(lc + 1, 8))) Then newcom = True
1145                 Misci.Cells(lc + 1, 8) = jbx
1146                 Eqlbrm.Cells(njc + 1, 2) = lc
1147                 GoTo 1300
1148             End If
1149  1250:   Next lc
1150  1300:   En.Cells(jbx, Indx.Cells(2, 14)) = -En.Cells(jbx, Indx.Cells(2, 14))
1151          If (njc < nn) Then GoTo 1200
1152      End If
1153      For j = 1 To Indx.Cells(2, 5)
1154          En.Cells(j, Indx.Cells(2, 14)) = Abs(En.Cells(j, Indx.Cells(2, 14)))
1155      Next j
1156      If (newcom) Then
1157  'Switch Components
1158          For lc = 1 To nn
1159              jb = Misci.Cells(lc + 1, 8)
1160              If (CDbl(A.Cells(lc, jb)) = 0) Then
1161                  jb = Indx.Cells(lc + 1, 21)
1162                  Misci.Cells(lc + 1, 8) = jb
1163              End If
1164              tem = A.Cells(lc, jb)
1165              If (tem <> 0) Then
1166                  Eqlbrm.Cells(lc + 1, 3) = Therm.Cells(jb + 1, 4) - Therm.Cells(jb + 1, 5)
1167                  If (jb <= CInt(Indx.Cells(2, 5))) Then Eqlbrm.Cells(lc + 1, 3) =
Eqlbrm.Cells(lc + 1, 3) + Comp.Cells(jb + 1, 6) + Miscr.Cells(2, 13)
1168                  Eqlbrm.Cells(lc + 1, 1) = Cdata.Cells(jb + 1, 10)
1169  'Calculate New Coefficients
1170                  If (tem <> 1) Then
1171                      Miscr.Cells(lc + 1, 19) = Miscr.Cells(lc + 1, 19) / tem
1172                      Inpt.Cells(lc + 1, 16) = Inpt.Cells(lc + 1, 16) / tem
'Inpt.Cells(lc+1,16) is B0p(lc,1)
1173                      Inpt.Cells(lc + 1, 17) = Inpt.Cells(lc + 1, 17) / tem
1174                      For j = 1 To Indx.Cells(2, 18) 'Indx.Cells(2,18) is Nspx
1175                          A.Cells(lc, j) = A.Cells(lc, j) / tem
1176                      Next j
1177                  End If
1178                  For i = 1 To nn
1179                      If (CDbl(A.Cells(i, jb)) <> 0 And i <> lc) Then
1180                          tem = A.Cells(i, jb)
1181                          For j = 1 To Indx.Cells(2, 18)
1182                              A.Cells(i, j) = A.Cells(i, j) - A.Cells(lc, j) * tem
1183                              If (Abs(A.Cells(i, j)) < 0.00001) Then A.Cells(i, j) = 0
1184                          Next j
1185                          Miscr.Cells(i + 1, 19) = Miscr.Cells(i + 1, 19) - Miscr.Cells(lc +
1, 19) * tem
1186                          Inpt.Cells(i + 1, 16) = Inpt.Cells(i + 1, 16) - Inpt.Cells(lc + 1,
16) * tem
1187                          Inpt.Cells(i + 1, 17) = Inpt.Cells(i + 1, 17) - Inpt.Cells(lc + 1,
17) * tem
1188                      End If
1189                  Next i
1190              End If
1191          Next lc
1192  'DEBUG
1193          If Miscl.Cells(Indx.Cells(2, 14) + 1, 2) Then
1194              deb.Range("A" & debint, "B" & debint).Merge
1195              deb.Cells(debint, 1) = "NEW COMPONENTS"
1196              debint = debint + 1
1197              For k = 1 To nn
1198                  deb.Cells(debint, k) = Eqlbrm.Cells(k + 1, 1)
1199              Next k
1200              debint = debint + 1
1201          End If
1202      End If
1203      If (CInt(Misci.Cells(2, 7)) <> 0) Then
1204  'Switch Order of Msing and Nlm Components
1205          reduce = True
1206          lelim = Indx.Cells(2, 7)
1207          lsing = Indx.Cells(2, 7)
```

```
1208          If (CInt(Misci.Cells(2, 7)) <> CInt(Indx.Cells(2, 7))) Then
1209              For j = 1 To Indx.Cells(2, 18)
1210                  aa = A.Cells(Misci.Cells(2, 7), j)
1211                  A.Cells(Misci.Cells(2, 7), j) = A.Cells(Indx.Cells(2, 7), j)
1212                  A.Cells(Indx.Cells(2, 7), j) = aa
1213              Next j
1214              ja = Misci.Cells(Misci.Cells(2, 7) + 1, 8)
1215              Misci.Cells(Misci.Cells(2, 7) + 1, 8) = Misci.Cells(Indx.Cells(2, 7) + 1, 8)
1216              Misci.Cells(Indx.Cells(2, 7) + 1, 8) = ja
1217              ae = CStr(Eqlbrm.Cells(Misci.Cells(2, 7) + 1, 1))
1218              Eqlbrm.Cells(Misci.Cells(2, 7) + 1, 1) = Eqlbrm.Cells(Indx.Cells(2, 7) + 1, 1)
1219              Eqlbrm.Cells(Indx.Cells(2, 7) + 1, 1) = ae
1220              ae = CStr(Cdata.Cells(Misci.Cells(2, 7) + 1, 1))
1221              Cdata.Cells(Misci.Cells(2, 7) + 1, 1) = Cdata.Cells(Indx.Cells(2, 7) + 1, 1)
1222              Cdata.Cells(Indx.Cells(2, 7) + 1, 1) = ae
1223              ja = Indx.Cells(Misci.Cells(2, 7) + 1, 21)
1224              Indx.Cells(Misci.Cells(2, 7) + 1, 21) = Indx.Cells(Indx.Cells(2, 7) + 1, 21)
1225              Indx.Cells(Indx.Cells(2, 7) + 1, 21) = ja
1226              aa = Miscr.Cells(Misci.Cells(2, 7) + 1, 18)
1227              Miscr.Cells(Misci.Cells(2, 7) + 1, 18) = Miscr.Cells(Indx.Cells(2, 7) + 1, 18)
1228              Miscr.Cells(Indx.Cells(2, 7) + 1, 18) = aa
1229              aa = Miscr.Cells(Misci.Cells(2, 7) + 1, 19)
1230              Miscr.Cells(Misci.Cells(2, 7) + 1, 19) = Miscr.Cells(Indx.Cells(2, 7) + 1, 19)
1231              Miscr.Cells(Indx.Cells(2, 7) + 1, 19) = aa
1232              aa = Eqlbrm.Cells(Misci.Cells(2, 7) + 1, 3)
1233              Eqlbrm.Cells(Misci.Cells(2, 7) + 1, 3) = Eqlbrm.Cells(Indx.Cells(2, 7) + 1, 3)
1234              Eqlbrm.Cells(Indx.Cells(2, 7) + 1, 3) = aa
1235              For i = 1 To 2
1236                  aa = Inpt.Cells(Misci.Cells(2, 7) + 1, i + 15)
1237                  Inpt.Cells(Misci.Cells(2, 7) + 1, i + 15) = Inpt.Cells(Indx.Cells(2, 7) +
1, i + 15)
1238                  Inpt.Cells(Indx.Cells(2, 7) + 1, i + 15) = aa
1239              Next i
1240          End If
1241      ElseIf (newcom = False And CDbl(Miscr.Cells(2, 14)) = 0) Then
1242          GoTo 600
1243      End If
1244      Misci.Cells(2, 7) = 0
1245      tsize = xsize
1246      GoTo 500
1247  1400: Prtout.Cells(Indx.Cells(2, 14) + 1, 9) = Miscr.Cells(2, 15)
'Prtout.Cells(Indx.Cells(2,14)+1,9) is Ttt(Npt)
1248      Prtout.Cells(Indx.Cells(2, 14) + 1, 6) = Miscr.Cells(2, 7)
'Prtout.Cells(Indx.Cells(2,14)+1,6) is Ppp(Npt)
1249      Prtout.Cells(Indx.Cells(2, 14) + 1, 10) = Miscr.Cells(2, 9) * Comp.Cells(2, 1) *
Miscr.Cells(2, 15) / Miscr.Cells(2, 7) 'Prtout.Cells(Indx.Cells(2,14)+1,10) is Vlm(Npt)
1250      Prtout.Cells(Indx.Cells(2, 14) + 1, 5) = Prtout.Cells(Indx.Cells(2, 14) + 1, 5) *
Miscr.Cells(2, 15)
1251      Prtout.Cells(Indx.Cells(2, 14) + 1, 11) = 1 / Comp.Cells(2, 1)
1252      gasfrc = Comp.Cells(2, 1) / Prtout.Cells(Indx.Cells(2, 14) + 1, 8)
1253   'OUTPUT
1254      If gasfrc < 0.0001 Then
1255          Call Statement_Format
1256          out.Cells(outint, 3) = "WARNING! RESULTS MAY BE WRONG FOR POINT " & Indx.Cells(2,
14) & " DUE TO"
1257          outint = outint + 1
1258          Call Statement_Format
1259          out.Cells(outint, 3) = "LOW MOLE FRACTION OF GASES " & gasfrc
1260          outint = outint + 1
1261      End If
1262      If (CDbl(Miscr.Cells(2, 14)) <> 0) Then
1263          For j = 1 To Indx.Cells(2, 5)
1264              If (lelim <> 0) Then
1265                  For i = lelim To ls
1266                      If (CDbl(A.Cells(i, j)) <> 0) Then GoTo 1450
1267                  Next i
1268              End If
1269              If (CDbl(Comp.Cells(j + 1, 6)) > -87) Then En.Cells(j, Indx.Cells(2, 14)) =
Exp(Comp.Cells(j + 1, 6))
1270  1450:   Next j
1271      End If
```

```
1272  'DEBUG
1273      If Miscl.Cells(Indx.Cells(2, 14) + 1, 2) Then
1274          deb.Cells(debint, 1) = "Point"
1275          deb.Cells(debint, 2) = Indx.Cells(2, 14)
1276          debint = debint + 1
1277          deb.Cells(debint, 1) = "P"
1278          deb.Cells(debint, 2) = Miscr.Cells(2, 7)
1279          debint = debint + 1
1280          deb.Cells(debint, 1) = "T"
1281          deb.Cells(debint, 2) = Miscr.Cells(2, 15)
1282          debint = debint + 1
1283          deb.Cells(debint, 1) = "H/R"
1284          deb.Cells(debint, 2) = Prtout.Cells(Indx.Cells(2, 14) + 1, 5)
1285          debint = debint + 1
1286          deb.Cells(debint, 1) = "S/R"
1287          deb.Cells(debint, 2) = Prtout.Cells(Indx.Cells(2, 14) + 1, 7)
1288          debint = debint + 1
1289          deb.Cells(debint, 1) = "M"
1290          deb.Cells(debint, 2) = Prtout.Cells(Indx.Cells(2, 14) + 1, 11)
1291          debint = debint + 1
1292          deb.Cells(debint, 1) = "Cp/R"
1293          deb.Cells(debint, 2) = Prtout.Cells(Indx.Cells(2, 14) + 1, 1)
1294          debint = debint + 1
1295          deb.Cells(debint, 1) = "DLVPT"
1296          deb.Cells(debint, 2) = Prtout.Cells(Indx.Cells(2, 14) + 1, 2)
1297          debint = debint + 1
1298          deb.Cells(debint, 1) = "DLVTP"
1299          deb.Cells(debint, 2) = Prtout.Cells(Indx.Cells(2, 14) + 1, 3)
1300          debint = debint + 1
1301          deb.Cells(debint, 1) = "Gamma(S)"
1302          deb.Cells(debint, 2) = Prtout.Cells(Indx.Cells(2, 14) + 1, 4)
1303          debint = debint + 1
1304          deb.Cells(debint, 1) = "V"
1305          deb.Cells(debint, 2) = Prtout.Cells(Indx.Cells(2, 14) + 1, 10)
1306          debint = debint + 2
1307      End If
1308      If (CDbl(Miscr.Cells(2, 15)) >= CDbl(Therm.Cells(2, 2)) And CDbl(Miscr.Cells(2, 15)) <=
CDbl(Therm.Cells(5, 2))) Then GoTo 1600
1309      If (Miscl.Cells(2, 13)) Then GoTo 1600
1310  'OUTPUT
1311      Call Statement_Format
1312      out.Cells(outint, 3) = "THE TEMPERATURE = " & Miscr.Cells(2, 15) & " IS OUT OF RANGE
FOR POINT " & Indx.Cells(2, 14)
1313      outint = outint + 2
1314      If (CDbl(Miscr.Cells(2, 15)) >= (CDbl(Therm.Cells(2, 2)) * 0.8) And CDbl(Miscr.Cells(2,
15)) <= (CDbl(Therm.Cells(5, 2)) * 1.1)) Then GoTo 1600
1315      Indx.Cells(2, 14) = Indx.Cells(2, 14) + 1
1316  1500: Miscr.Cells(2, 15) = 0
1317      Indx.Cells(2, 14) = Indx.Cells(2, 14) - 1
1318  'OUTPUT
1319      Call Statement_Format
1320      out.Cells(outint, 3) = "CALCULATIONS STOPPED AFTER POINT " & Indx.Cells(2, 14)
1321      outint = outint + 2
1322  1600: Misci.Cells(2, 6) = Indx.Cells(2, 7)
1323      Indx.Cells(2, 7) = ls
1324      If (CInt(Indx.Cells(2, 13)) > 0) Then Miscl.Cells(2, 6) = False 'Miscr.Cells(2,6) is
Gonly
1325  End Sub
1326
1327  Sub CPHS_Sub()
1328  'CALCULATES THERMODYNAMIC PROPERTIES FOR INDIVIDUAL SPECIES
1329
1330  'Set Variables
1331      Dim tt As Double
1332      Dim Tg(4) As Double
1333      Dim Tln As Double
1334      Dim Ng As Integer
1335      Dim Npr As Integer
1336      Dim Ngc As Integer
1337
1338      tt = Miscr.Cells(2, 15)
```

```
1339      Tg(0) = Therm.Cells(2, 2)
1340      Tg(1) = Therm.Cells(3, 2)
1341      Tg(2) = Therm.Cells(4, 2)
1342      Tg(3) = Therm.Cells(5, 2)
1343      Tln = Miscr.Cells(2, 12)
1344      Ng = Indx.Cells(2, 5)
1345      Npr = Indx.Cells(2, 13)
1346      Ngc = Indx.Cells(2, 15)
1347
1348  'Set Index Variables
1349      Dim i, ij, j, jj, k As Integer
1350
1351  'Initialize cx, hcx, and scx
1352      With Cphs
1353          .Cells(2, 1) = 0
1354          .Cells(3, 1) = 0
1355          .Cells(4, 1) = 1
1356          .Cells(5, 1) = 0.5
1357          .Cells(6, 1) = CDbl(2 / 3)
1358          .Cells(7, 1) = 0.75
1359          .Cells(8, 1) = 0.8
1360          For i = 1 To 7
1361              .Cells(i + 1, 2) = 0
1362              .Cells(i + 1, 3) = 0
1363          Next i
1364          .Cells(4, 2) = 1
1365      End With
1366
1367
1368  'Define k
1369      k = 0
1370      If (tt > Tg(1)) Then k = 1
1371      If (tt > Tg(2)) Then k = 2
1372
1373  'Set Coefficients
1374      Cphs.Cells(3, 1) = 1 / tt
1375      Cphs.Cells(2, 1) = Cphs.Cells(3, 1) ^ 2
1376      Cphs.Cells(4, 3) = Tln
1377      Cphs.Cells(3, 3) = -Cphs.Cells(3, 1)
1378      Cphs.Cells(3, 2) = Tln * Cphs.Cells(3, 1)
1379      Cphs.Cells(2, 2) = -Cphs.Cells(2, 1)
1380      Cphs.Cells(2, 3) = Cphs.Cells(2, 2) * 0.5
1381      For i = 4 To 7
1382          Cphs.Cells(i + 1, 2) = Cphs.Cells(i + 1, 1) * tt
1383          Cphs.Cells(i + 1, 3) = Cphs.Cells(i, 1) * tt
1384      Next i
1385  'Initialize H0 and S
1386      For j = 1 To Ng
1387          Therm.Cells(j + 1, 4) = 0 'Therm.Cells(j+1,4) is H0(j)
1388          Therm.Cells(j + 1, 5) = 0 'Therm.Cells(j+1,5) is S(j)
1389      Next j
1390  'Calculate H0 and S using the Coefficients
1391      For i = 7 To 4 Step -1
1392          For j = 1 To Ng
1393              Therm.Cells(j + 1, 5) = (Therm.Cells(j + 1, 5) + Coef.Cells(j + 1, (i + 9 *
k))) * Cphs.Cells(i + 1, 3)
1394              Therm.Cells(j + 1, 4) = (Therm.Cells(j + 1, 4) + Coef.Cells(j + 1, (i + 9 *
k))) * Cphs.Cells(i + 1, 2)
1395          Next j
1396      Next i
1397      For i = 1 To 3
1398          For j = 1 To Ng
1399              Therm.Cells(j + 1, 5) = Therm.Cells(j + 1, 5) + Coef.Cells(j + 1, (i + 9 * k))
* Cphs.Cells(i + 1, 3)
1400              Therm.Cells(j + 1, 4) = Therm.Cells(j + 1, 4) + Coef.Cells(j + 1, (i + 9 * k))
* Cphs.Cells(i + 1, 2)
1401          Next j
1402      Next i
1403      For j = 1 To Ng
1404          Therm.Cells(j + 1, 5) = Therm.Cells(j + 1, 5) + Coef.Cells(j + 1, 9 + 9 * k)
```

```
1405          Therm.Cells(j + 1, 4) = Therm.Cells(j + 1, 4) + Coef.Cells(j + 1, (8 + 9 * k)) *
Cphs.Cells(3, 1)
1406      Next j
1407  'Calculate Cp using the Coefficients
1408      If (Miscl.Cells(2, 17) = False Or Miscl.Cells(2, 1) = True) Then 'Miscl.Cells(2,17) is
Tp, Miscl.Cells(2,1) is Convg
1409          For j = 1 To Ng
1410              Therm.Cells(j + 1, 3) = 0 'Therm.Cells(j+1,3) is Cp(j)
1411          Next j
1412          For i = 7 To 4 Step -1
1413              For j = 1 To Ng
1414                  Therm.Cells(j + 1, 3) = (Therm.Cells(j + 1, 3) + Coef.Cells(j + 1, (i + 9 *
k))) * tt
1415              Next j
1416          Next i
1417          For i = 1 To 3
1418              For j = 1 To Ng
1419                  Therm.Cells(j + 1, 3) = Therm.Cells(j + 1, 3) + Coef.Cells(j + 1, (i + 9 *
k)) * Cphs.Cells(i + 1, 1)
1420              Next j
1421          Next i
1422      End If
1423  'Update Equations for Condensed Species
1424      If (Npr <> 0 And k <> 3 And Ng <> Ngc) Then
1425          For ij = 1 To Npr
1426              j = Indx.Cells(ij + 1, 20) 'Indx.Cells(ij+1, 20) is Jcond(ij)
1427              jj = Indx.Cells(ij + 1, 20) - Ng
1428              Therm.Cells(j + 1, 3) = 0
1429              Therm.Cells(j + 1, 4) = 0
1430              Therm.Cells(j + 1, 5) = 0
1431              For i = 7 To 4 Step -1
1432                  Therm.Cells(j + 1, 5) = (Therm.Cells(j + 1, 5) + Therm.Cells(jj + 1, i +
7)) * Cphs.Cells(i + 1, 3)
1433                  Therm.Cells(j + 1, 4) = (Therm.Cells(j + 1, 4) + Therm.Cells(jj + 1, i +
7)) * Cphs.Cells(i + 1, 2)
1434                  Therm.Cells(j + 1, 3) = (Therm.Cells(j + 1, 3) + Therm.Cells(jj + 1, i +
7)) * tt
1435              Next i
1436              For i = 1 To 3
1437                  Therm.Cells(j + 1, 5) = Therm.Cells(j + 1, 5) + Therm.Cells(jj + 1, i + 7)
* Cphs.Cells(i + 1, 3)
1438                  Therm.Cells(j + 1, 4) = Therm.Cells(j + 1, 4) + Therm.Cells(jj + 1, i + 7)
* Cphs.Cells(i + 1, 2)
1439                  Therm.Cells(j + 1, 3) = Therm.Cells(j + 1, 3) + Therm.Cells(jj + 1, i + 7)
* Cphs.Cells(i + 1, 1)
1440              Next i
1441              Therm.Cells(j + 1, 5) = Therm.Cells(j + 1, 5) + Therm.Cells(jj + 1, 16)
1442              Therm.Cells(j + 1, 4) = Therm.Cells(j + 1, 4) + Therm.Cells(jj + 1, 15) *
Cphs.Cells(3, 1)
1443          Next ij
1444      End If
1445  End Sub
1446
1447  Sub ALLCON_Sub()
1448  'CALCULATES THERMODYNAMIC PROPERTIES FOR INDIVIDUAL SPECIES - ALL CONDENSED
1449
1450  'Set Variables
1451      Dim Ng As Integer
1452      Dim Nc As Integer
1453      Dim Npr As Integer
1454      Dim Ngc As Integer
1455
1456      Ng = Indx.Cells(2, 5)
1457      Nc = Indx.Cells(2, 4)
1458      Npr = Indx.Cells(2, 13)
1459      Ngc = Indx.Cells(2, 15)
1460      tt = Miscr.Cells(2, 15)
1461
1462  'Set Index Variables
1463      Dim i, ij, j, jj As Integer
1464
```

```
1465  'Calculate Cp, H0, and S for the All Condensed Species
1466      For jj = 1 To Nc
1467          j = jj + Ng
1468          Therm.Cells(j + 1, 3) = 0 'Therm.Cells(j+1,3) is Cp(j)
1469          Therm.Cells(j + 1, 4) = 0 'Therm.Cells(j+1,4) is H0(j)
1470          Therm.Cells(j + 1, 5) = 0 'Therm.Cells(j+1,5) is S(j)
1471          For i = 7 To 4 Step -1
1472              Therm.Cells(j + 1, 5) = (Therm.Cells(j + 1, 5) + Therm.Cells(jj + 1, i + 7)) *
Cphs.Cells(i + 1, 3)
1473              Therm.Cells(j + 1, 4) = (Therm.Cells(j + 1, 4) + Therm.Cells(jj + 1, i + 7)) *
Cphs.Cells(i + 1, 2)
1474              Therm.Cells(j + 1, 3) = (Therm.Cells(j + 1, 3) + Therm.Cells(jj + 1, i + 7)) *
tt
1475          Next i
1476          For i = 1 To 3
1477              Therm.Cells(j + 1, 5) = Therm.Cells(j + 1, 5) + Therm.Cells(jj + 1, i + 7) *
Cphs.Cells(i + 1, 3)
1478              Therm.Cells(j + 1, 4) = Therm.Cells(j + 1, 4) + Therm.Cells(jj + 1, i + 7) *
Cphs.Cells(i + 1, 2)
1479              Therm.Cells(j + 1, 3) = Therm.Cells(j + 1, 3) + Therm.Cells(jj + 1, i + 7) *
Cphs.Cells(i + 1, 1)
1480          Next i
1481          Therm.Cells(j + 1, 5) = Therm.Cells(j + 1, 5) + Therm.Cells(jj + 1, 16)
1482          Therm.Cells(j + 1, 4) = Therm.Cells(j + 1, 4) + Therm.Cells(jj + 1, 15) *
Cphs.Cells(3, 1)
1483      Next jj
1484  End Sub
1485
1486  Sub MATRIX_Sub()
1487  'SET UP ITERATION OR DERIVATIVE MATRIX
1488
1489  'Set Variables
1490      Dim Nlm, Npr, Npt As Integer
1491      Dim engergyl, f, h, ss, sss, term, term1 As Double
1492      Nlm = Indx.Cells(2, 7)
1493      Npr = Indx.Cells(2, 13)
1494      Npt = Indx.Cells(2, 14)
1495  'Set Index Variables
1496      Dim i, iq, iq2, iq3, isym As Integer
1497      Dim j, k, kk, kmat As Integer
1498  'Set Variables that define the Matrix Size
1499      iq = Nlm + Npr
1500      Misci.Cells(2, 2) = iq + 1 'Misci.Cells(2,2) is Iq1
1501      iq2 = Misci.Cells(2, 2) + 1
1502      iq3 = iq2 + 1
1503      kmat = iq3
1504      If (Miscl.Cells(2, 1) = False And Miscl.Cells(2, 17)) Then kmat = iq2 'Miscl.Cells(2,1)
is Convg, Miscl.Cells(2,17) is Tp
1505      Misci.Cells(2, 1) = kmat - 1 'Misci.Cells(2,1) is Imat
1506  'Clear Matrix Storages to Zero
1507      For i = 1 To Misci.Cells(2, 1)
1508          For k = 1 To kmat
1509              G.Cells(i, k) = 0
1510          Next k
1511      Next i
1512      G.Cells(iq2, Misci.Cells(2, 2)) = 0
1513      sss = 0
1514      Prtout.Cells(Npt + 1, 5) = 0 'Prtout.Cells(Npt+1,5) is Hsum(Npt)
1515  'Begin Set-up of Iteration or Derivative Matrix
1516      For j = 1 To Indx.Cells(2, 5)
1517          'Therm.Cells(j+1,6) is Mu(j), Therm.Cells(j+1,4) is H0(j), Therm.Cells(j+1,5) is
S(j)
1518          'Comp.Cells(j+1,6) is Enln(j), Miscr.Cells(2,13) is Tm
1519          Therm.Cells(j + 1, 6) = Therm.Cells(j + 1, 4) - Therm.Cells(j + 1, 5) +
Comp.Cells(j + 1, 6) + Miscr.Cells(2, 13)
1520          If (CDbl(En.Cells(j, Npt)) <> 0) Then 'En.Cells(j, Npt) is En(j,Npt)
1521              h = Therm.Cells(j + 1, 4) * En.Cells(j, Npt)
1522              f = Therm.Cells(j + 1, 6) * En.Cells(j, Npt)
1523              ss = h - f
1524              term1 = h
1525              If (kmat = iq2) Then term1 = f
```

```
1526              For i = 1 To Nlm
1527                  If (CDbl(A.Cells(i, j)) <> 0) Then
1528                      term = A.Cells(i, j) * En.Cells(j, Npt)
1529                      For k = i To Nlm
1530                          G.Cells(i, k) = G.Cells(i, k) + A.Cells(k, j) * term
1531                      Next k
1532                      G.Cells(i, Misci.Cells(2, 2)) = G.Cells(i, Misci.Cells(2, 2)) + term
1533                      G.Cells(i, iq2) = G.Cells(i, iq2) + A.Cells(i, j) * term1
1534                      If (Not (Miscl.Cells(2, 1) Or Miscl.Cells(2, 17))) Then
1535                          G.Cells(i, iq3) = G.Cells(i, iq3) + A.Cells(i, j) * f
1536                          If Miscl.Cells(2, 16) Then G.Cells(iq2, i) = G.Cells(iq2, i) +
A.Cells(i, j) * ss
1537                      End If
1538                  End If
1539              Next i
1540              If (kmat <> iq2) Then
1541                  If (Miscl.Cells(2, 1) Or Miscl.Cells(2, 7)) Then 'Miscl.Cells(2,7) is Hp
1542                      G.Cells(iq2, iq2) = G.Cells(iq2, iq2) + Therm.Cells(j + 1, 4) * h
1543                      If (Miscl.Cells(2, 1) = False) Then
1544                          G.Cells(iq2, iq3) = G.Cells(iq2, iq3) + Therm.Cells(j + 1, 4) * f
1545                          G.Cells(Misci.Cells(2, 2), iq3) = G.Cells(Misci.Cells(2, 2), iq3) +
f
1546                      End If
1547                  Else
1548                      G.Cells(iq2, Misci.Cells(2, 2)) = G.Cells(iq2, Misci.Cells(2, 2)) + ss
1549                      G.Cells(iq2, iq2) = G.Cells(iq2, iq2) + Therm.Cells(j + 1, 4) * ss
1550                      G.Cells(iq2, iq3) = G.Cells(iq2, iq3) + Therm.Cells(j + 1, 6) * ss
1551                      G.Cells(Misci.Cells(2, 2), iq3) = G.Cells(Misci.Cells(2, 2), iq3) + f
1552                  End If
1553              End If
1554              G.Cells(Misci.Cells(2, 2), iq2) = G.Cells(Misci.Cells(2, 2), iq2) + term1
1555          End If
1556      Next j
1557  'Condensed Species
1558      If (Npr <> 0) Then
1559          For k = 1 To Npr
1560              j = Indx.Cells(k + 1, 20) 'Indx.Cells(k+1,20) is Jcond(k)
1561              kk = Nlm + k
1562              Therm.Cells(j + 1, 6) = Therm.Cells(j + 1, 4) - Therm.Cells(j + 1, 5)
1563              For i = 1 To Nlm
1564                  G.Cells(i, kk) = A.Cells(i, j)
1565                  G.Cells(i, kmat) = G.Cells(i, kmat) - A.Cells(i, j) * En.Cells(j, Npt)
1566              Next i
1567              G.Cells(kk, iq2) = Therm.Cells(j + 1, 4)
1568              G.Cells(kk, kmat) = Therm.Cells(j + 1, 6)
1569              Prtout.Cells(Npt + 1, 5) = Prtout.Cells(Npt + 1, 5) + Therm.Cells(j + 1, 4) *
En.Cells(j, Npt)
1570              If (Miscl.Cells(2, 16)) Then
1571                  sss = sss + Therm.Cells(j + 1, 5) * En.Cells(j, Npt)
1572                  G.Cells(iq2, kk) = Therm.Cells(j + 1, 5)
1573              End If
1574          Next k
1575      End If
1576      sss = sss + G.Cells(iq2, Misci.Cells(2, 2))
1577      Prtout.Cells(Npt + 1, 5) = Prtout.Cells(Npt + 1, 5) + G.Cells(Misci.Cells(2, 2), iq2)
1578      G.Cells(Misci.Cells(2, 2), Misci.Cells(2, 2)) = Comp.Cells(2, 4) - Comp.Cells(2, 1)
'Comp.Cells(2,4) is Sumn, Comp.Cells(2,1) is Enn
1579  'Reflect Symmetric Portions of the Matrix
1580      isym = Misci.Cells(2, 2)
1581      If (Miscl.Cells(2, 7) Or Miscl.Cells(2, 1)) Then isym = iq2
1582      For i = 1 To isym
1583          For j = i To isym
1584              G.Cells(j, i) = G.Cells(i, j)
1585          Next j
1586      Next i
1587  'Complete the Right Hand Side
1588      If (Miscl.Cells(2, 1) = False) Then
1589          For i = 1 To Nlm
1590              G.Cells(i, kmat) = G.Cells(i, kmat) + Miscr.Cells(i + 1, 19) - G.Cells(i,
Misci.Cells(2, 2)) 'Miscr.Cells(i+1,19) is B0(i)
1591          Next i
```

```
1592          G.Cells(Misci.Cells(2, 2), kmat) = G.Cells(Misci.Cells(2, 2), kmat) + Comp.Cells(2,
1) - Comp.Cells(2, 4)
1593  'Complete Energy Row and Temperature Column
1594          If (kmat <> iq2) Then
1595              If (Miscl.Cells(2, 16)) Then energyl = Miscr.Cells(2, 11) + Comp.Cells(2, 1) -
Comp.Cells(2, 4) - sss 'Miscr.Cells(2,11) is S0
1596              If (Miscl.Cells(2, 7)) Then energyl = (Miscr.Cells(2, 4) / Miscr.Cells(2, 15))
- Prtout.Cells(Npt + 1, 5) 'Miscr.Cells(2,4) is Hsub0, Miscr.Cells(2,15) is Tt
1597              G.Cells(iq2, iq3) = G.Cells(iq2, iq3) + energyl
1598              G.Cells(iq2, iq2) = G.Cells(iq2, iq2) + Therm.Cells(2, 1) 'Therm.Cells(2,1) is
Cpsum
1599          End If
1600      Else
1601          If (Miscl.Cells(2, 12)) Then 'Miscl.Cells(2,12) is PDERIV = TRUE -- Set Up Matrix
to Solve for DLVPT
1602              G.Cells(Misci.Cells(2, 2), iq2) = Comp.Cells(2, 1)
1603              For i = 1 To iq
1604                  G.Cells(i, iq2) = G.Cells(i, Misci.Cells(2, 2))
1605              Next i
1606          End If
1607          G.Cells(iq2, iq2) = G.Cells(iq2, iq2) + Therm.Cells(2, 1)
1608      End If
1609  'Constant Volume Matrix
1610      If (Miscl.Cells(2, 19) And Miscl.Cells(2, 1) = False) Then 'Miscl.Cells(2,19) is Vol
1611          If (kmat = iq2) Then
1612              For i = 1 To iq
1613                  G.Cells(i, Misci.Cells(2, 2)) = G.Cells(i, iq2)
1614              Next i
1615          Else
1616              For i = 1 To iq
1617                  G.Cells(Misci.Cells(2, 2), i) = G.Cells(iq2, i) - G.Cells(Misci.Cells(2,
2), i)
1618                  G.Cells(i, Misci.Cells(2, 2)) = G.Cells(i, iq2) - G.Cells(i, Misci.Cells(2,
2))
1619                  G.Cells(i, iq2) = G.Cells(i, iq3)
1620              Next i
1621              G.Cells(Misci.Cells(2, 2), Misci.Cells(2, 2)) = G.Cells(iq2, iq2) -
G.Cells(Misci.Cells(2, 2), iq2) - G.Cells(iq2, Misci.Cells(2, 2))
1622              G.Cells(Misci.Cells(2, 2), iq2) = G.Cells(iq2, iq3) - G.Cells(Misci.Cells(2,
2), iq3)
1623              If (Miscl.Cells(2, 7)) Then G.Cells(Misci.Cells(2, 2), iq2) =
G.Cells(Misci.Cells(2, 2), iq2) + Comp.Cells(2, 1)
1624          End If
1625          kmat = Misci.Cells(2, 1)
1626          Misci.Cells(2, 1) = Misci.Cells(2, 1) - 1
1627      End If
1628  End Sub
1629
1630  Sub GAUSS_Sub()
1631  'SOLVE ANY LINEAR SET OF UP TO MAXMAT EQUATIONS
1632  'NUMBER OF EQUATIONS = IMAT
1633
1634  'Set Variables
1635      Dim bigno, tmp As Double
1636      bigno = 10 ^ 25
1637  'Set Index Variables
1638      Dim i, imatp1, j, k, nn, nnp1 As Integer
1639  'Begin Elimination of NNth Variable
1640      imatp1 = Misci.Cells(2, 1) + 1 'Misci.Cells(2, 1) is Imat
1641      For nn = 1 To Misci.Cells(2, 1)
1642          If (nn <> CInt(Misci.Cells(2, 1))) Then
1643  'Search for Maximum Cefficient in Each Row
1644              nnp1 = nn + 1
1645              For i = nn To Misci.Cells(2, 1)
1646                  Gauss.Cells(i + 1, 1) = bigno 'Gauss.Cells(i+1,1) is coefx(i)
1647                  If (CDbl(G.Cells(i, nn)) <> 0) Then
1648                      Gauss.Cells(i + 1, 1) = 0
1649                      For j = nnp1 To imatp1
1650                          If (Gauss.Cells(i + 1, 1) < Abs(G.Cells(i, j))) Then Gauss.Cells(i
+ 1, 1) = Abs(G.Cells(i, j))
1651                      Next j
```

```
1652                        tmp = Abs(G.Cells(i, nn))
1653                        If ((bigno * tmp) > Gauss.Cells(i + 1, 1)) Then
1654                            Gauss.Cells(i + 1, 1) = Gauss.Cells(i + 1, 1) / tmp
1655                        Else
1656                            Gauss.Cells(i + 1, 1) = bigno
1657                        End If
1658                    End If
1659                Next i
1660  'Locate row with smallest maximum coefficient
1661                tmp = bigno
1662                i = 0
1663                For j = nn To Misci.Cells(2, 1)
1664                    If (CDbl(Gauss.Cells(j + 1, 1)) < tmp) Then
1665                        tmp = Gauss.Cells(j + 1, 1)
1666                        i = j
1667                    End If
1668                Next j
1669                If (i = 0) Then
1670                    Misci.Cells(2, 7) = nn 'Misci.Cells(2,7) is Msing
1671                    GoTo 99999
1672  'Index i locates equation to be used for eliminating the nth variable from the remaining
equations
1673  'Interchange equations i and nn
1674                ElseIf (nn <> i) Then
1675                    For j = nn To imatp1
1676                        tmp = G.Cells(i, j)
1677                        G.Cells(i, j) = G.Cells(nn, j)
1678                        G.Cells(nn, j) = tmp
1679                    Next j
1680                End If
1681            ElseIf (CDbl(G.Cells(nn, nn)) = 0) Then
1682                Misci.Cells(2, 7) = nn
1683                GoTo 99999
1684            End If
1685  'Divide nth row by nth diagonal element and eliminate the nth variable from the remaining
equations
1686            k = nn + 1
1687            tmp = G.Cells(nn, nn)
1688            If (tmp = 0) Then
1689                Misci.Cells(2, 7) = nn
1690                GoTo 99999
1691            Else
1692                For j = k To imatp1
1693                    G.Cells(nn, j) = G.Cells(nn, j) / tmp
1694                Next j
1695                If (k <> imatp1) Then
1696                    For i = k To Misci.Cells(2, 1)
1697                        For j = k To imatp1
1698                            G.Cells(i, j) = G.Cells(i, j) - (G.Cells(i, nn) * G.Cells(nn, j))
1699                        Next j
1700                    Next i
1701                End If
1702            End If
1703        Next nn
1704  'Backsolve for the variables
1705        k = Misci.Cells(2, 1)
1706  100: j = k + 1
1707        Miscr.Cells(k + 1, 20) = 0 'Miscr.Cells(k+1,20) is X(k)
1708        tmp = 0
1709        If (CInt(Misci.Cells(2, 1)) >= j) Then
1710            For i = j To Misci.Cells(2, 1)
1711                tmp = tmp + G.Cells(k, i) * Miscr.Cells(i + 1, 20)
1712            Next i
1713        End If
1714        Miscr.Cells(k + 1, 20) = G.Cells(k, imatp1) - tmp
1715        k = k - 1
1716        If (k <> 0) Then GoTo 100
1717  99999:
1718  End Sub
1719
1720  Sub NEWOF_Sub()
```

```
1721  'Calculate New Values of B0 and Hsub0 For New Oil/Fuel Ratio
1722
1723  'Set Variables
1724      Dim assval, bigb, bratio, dbi, smalb, tem, v1, v2 As Double
1725  'Set Index Variables
1726      Dim i, j As Integer
1727  'OUTPUT
1728      If Not Miscl.Cells(2, 14) Then
1729          out.Cells(outint, 1) = "O/F ="
1730          out.Cells(outint, 2) = CDbl(Miscr.Cells(2, 5))
1731          outint = outint + 2
1732      End If
1733  'Sub Start
1734      Miscr.Cells(2, 3) = 0 'Miscr.Cells(2,3) is Eqrat
1735      tem = Miscr.Cells(2, 5) + 1 'Miscr.Cells(2,5) is Oxfl
1736      v2 = (Miscr.Cells(2, 5) * Inpt.Cells(2, 6) + Inpt.Cells(3, 6)) / tem 'Inpt.Cells(2,6)
is Vmin(1)
1737      v1 = (Miscr.Cells(2, 5) * Inpt.Cells(2, 7) + Inpt.Cells(3, 7)) / tem 'Inpt.Cells(2,7)
is Vpls(1)
1738      If (v2 <> 0) Then Miscr.Cells(2, 3) = Abs(v1 / v2)
1739      For i = 1 To Indx.Cells(2, 7) 'Indx.Cells(2,7) is Nlm
1740          Miscr.Cells(i + 1, 19) = (Miscr.Cells(2, 5) * Inpt.Cells(i + 1, 16) + Inpt.Cells(i
+ 1, 17)) / tem 'Miscr.Cells(i+1,19) is B0(i), Inpt.Cells(i+1,16) is B0p(i,1)
1741          dbi = Abs(Miscr.Cells(i + 1, 19))
1742          If (i = 1) Then
1743              bigb = dbi
1744              smalb = dbi
1745          ElseIf (dbi <> 0) Then
1746              If (dbi < smalb) Then smalb = dbi
1747              If (dbi > bigb) Then bigb = dbi
1748          End If
1749      Next i
1750      Inpt.Cells(2, 3) = bigb * 0.000001 'Inpt.Cells(2,3) is Bcheck
1751  'Calculate Molecular Weight of Total Reactant, Wmix
1752      If (Inpt.Cells(2, 4) <> 0 And Inpt.Cells(3, 4) <> 0) Then 'Inpt.Cells(2,4) is Am(1)
1753          Inpt.Cells(2, 2) = (Miscr.Cells(2, 5) + 1) * Inpt.Cells(2, 4) * Inpt.Cells(3, 4) /
(Inpt.Cells(2, 4) + Miscr.Cells(2, 5) * Inpt.Cells(3, 4)) 'Inpt.Cells(2,2) is Wmix
1754      Else
1755          Inpt.Cells(2, 2) = Inpt.Cells(3, 4)
1756          If (Inpt.Cells(3, 4) = 0) Then Inpt.Cells(2, 2) = Inpt.Cells(2, 4)
1757      End If
1758      Indx.Cells(2, 14) = 1 'Indx.Cells(2,14) is Npt
1759  'If Assigned U or H Not Given in Prob Data, Initial Hsub0 = 1 x 10^30
1760      If (Miscr.Cells(2, 10) = 0) Then assval = Miscr.Cells(2, 4) 'Miscr.Cells(2,10) is Size,
Miscr.Cells(2,4) is Hsub0
1761      If (assval >= (10 ^ 30)) Then Miscr.Cells(2, 4) = (Miscr.Cells(2, 5) * Inpt.Cells(2, 5)
+ Inpt.Cells(3, 5)) / tem 'Inpt.Cells(2,5) is Hpp(1)
1762  'Note that "Bratio" is "Bratio" in Sec 3.2 in RP-1311
1763      bratio = smalb / bigb
1764      Miscr.Cells(2, 10) = 18.420681
1765      If (bratio < 0.00001) Then Miscr.Cells(2, 10) = Log(1000 / bratio)
1766      Misci.Cells(2, 5) = 0 'Misci.Cells(2,5) is Jsol
1767      Misci.Cells(2, 4) = 0 'Misci.Cells(2,4) is Jliq
1768  'OUTPUT
1769      If Not Miscl.Cells(2, 14) Then
1770          out.Range("C" & outint, "D" & outint).Merge
1771          out.Cells(outint, 3) = "EFFECTIVE FUEL"
1772          out.Cells(outint, 3).HorizontalAlignment = xlCenter
1773          out.Range("E" & outint, "F" & outint).Merge
1774          out.Cells(outint, 5) = "EFFECTIVE OXIDANT"
1775          out.Cells(outint, 5).HorizontalAlignment = xlCenter
1776          out.Range("G" & outint, "H" & outint).Merge
1777          out.Cells(outint, 7) = "MIXTURE"
1778          out.Cells(outint, 7).HorizontalAlignment = xlCenter
1779          outint = outint + 1
1780          Call Mix_Table_Format
1781          If Miscl.Cells(2, 19) Then
1782              out.Cells(outint, 1) = "INTERNAL ENERGY"
1783              out.Cells(outint, 3) = "u(2)/R"
1784              out.Cells(outint, 5) = "u(1)/R"
1785              out.Cells(outint, 7) = "u0/R"
```

```
1786          Else
1787              out.Cells(outint, 1) = "ENTHALPY"
1788              out.Cells(outint, 3) = "h(2)/R"
1789              out.Cells(outint, 5) = "h(1)/R"
1790              out.Cells(outint, 7) = "h0/R"
1791          End If
1792          outint = outint + 1
1793          Call Mix Table Format
1794          out.Cells(outint, 1) = "(KG-MOL)(K)/KG"
1795          out.Cells(outint, 3) = CDbl(Inpt.Cells(3, 5))
1796          out.Cells(outint, 5) = CDbl(Inpt.Cells(2, 5))
1797          out.Cells(outint, 7) = CDbl(Miscr.Cells(2, 4))
1798          outint = outint + 2
1799          Call Mix Table Format
1800          out.Cells(outint, 1) = "KG-FORM.WT./KG"
1801          out.Cells(outint, 3) = "bi(2)"
1802          out.Cells(outint, 5) = "bi(1)"
1803          out.Cells(outint, 7) = "b0i"
1804          outint = outint + 1
1805      End If
1806      For i = 1 To Indx.Cells(2, 7) 'Indx.Cells(2,7)
1807          j = Misci.Cells(i + 1, 8) 'Misci.Cells(i+1,8) is Jcm(i)
1808  'OUTPUT
1809          If Not Miscl.Cells(2, 14) Then
1810              Call Mix Table Format
1811              out.Cells(outint, 1) = CStr(Cdata.Cells(j + 1, 10))
1812              out.Cells(outint, 3) = CDbl(Inpt.Cells(i + 1, 17))
1813              out.Cells(outint, 5) = CDbl(Inpt.Cells(i + 1, 16))
1814              out.Cells(outint, 7) = CDbl(Miscr.Cells(i + 1, 19))
1815              outint = outint + 1
1816          End If
1817      Next i
1818      If Not Miscl.Cells(2, 14) Then outint = outint + 1
1819
1820  End Sub
1821
1822  Sub SETEN_Sub()
1823  'Use Compositions From Previous Point As Initial Estimates For Current Point Npt. If -
1824  'ISV > 0 Use Compositions From Point ISV.
1825  'ISV < 0 Save Compositions From Point -ISV For Possible Later Use. Also Use Compositions
From Point -ISV For Npt.
1826  'ISV = 0 Use Compositions Saved When ISV < 0
1827
1828  'Set Variables
1829      Dim tsave As Double
1830  'Set Index Variables
1831      Dim j, lsav As Integer
1832      If (CInt(Misci.Cells(2, 3)) < 0) Then 'Misci.Cells(2,3) is Isv
1833  'First T-- Save Compositions for Future Points with This T
1834          Misci.Cells(2, 3) = -Misci.Cells(2, 3)
1835          tsave = Prtout.Cells(Misci.Cells(2, 3) + 1, 9) 'Prtout.Cells(Misci.Cells(2,3)+1,9)
is Ttt(Isv)
1836          Comp.Cells(2, 8) = Comp.Cells(2, 1) 'Comp.Cells(2,8) is Ensave, Comp.Cells(2,1) is
Enn
1837          Comp.Cells(2, 3) = Comp.Cells(2, 2) 'Comp.Cells(2,3) is Enlsav, Comp.Cells(2,2) is
Ennl
1838          lsav = Misci.Cells(2, 6) 'Misci.Cells(2,6) is Lsave
1839          For j = 1 To Indx.Cells(2, 5) 'Indx.Cells(2,5) is Ng
1840              Comp.Cells(j + 1, 7) = Comp.Cells(j + 1, 6) 'Comp.Cells(j+1,7) is Sln(j),
Comp.Cells(j+1,6) is Enln(j)
1841          Next j
1842          For j = 1 To Indx.Cells(2, 5)
1843              En.Cells(j, Indx.Cells(2, 14)) = En.Cells(j, Misci.Cells(2, 3))
'En.Cells(j,Indx.Cells(2,14)) is En(j,Npt)
1844          Next j
1845          Indx.Cells(2, 13) = 0 'Indx.Cells(2,13) is Npr
1846          For j = Indx.Cells(2, 6) To Indx.Cells(2, 15) 'Indx.Cells(2,6) is Ngp1,
Indx.Cells(2,15) is Ngc
1847              Comp.Cells(j + 1, 7) = En.Cells(j, Misci.Cells(2, 3))
1848              En.Cells(j, Indx.Cells(2, 14)) = Comp.Cells(j + 1, 7)
1849              If (CInt(Misci.Cells(2, 4)) = j) Then 'Misci.Cells(2,4) is Jliq
```

```
1850                 En.Cells(Misci.Cells(2, 5), Indx.Cells(2, 14)) = En.Cells(Misci.Cells(2,
5), Misci.Cells(2, 3)) + En.Cells(Misci.Cells(2, 4), Misci.Cells(2, 3)) 'Misci.Cells(2,5) is Jsol
1851                 En.Cells(Misci.Cells(2, 4), Indx.Cells(2, 14)) = 0
1852                 Misci.Cells(2, 5) = 0
1853                 Misci.Cells(2, 4) = 0
1854                 tsave = tsave - 5
1855                 Miscr.Cells(2, 15) = tsave 'Miscr.Cells(2,15) is Tt
1856                 Comp.Cells(j + 1, 7) = 0
1857             ElseIf CDbl(En.Cells(j, Indx.Cells(2, 14))) > 0 Then
1858                 Indx.Cells(2, 13) = Indx.Cells(2, 13) + 1
1859                 Indx.Cells(Indx.Cells(2, 13) + 1, 20) = j
'Indx.Cells(Indx.Cells(2,13)+1,20) is Jcond(Npr)
1860             End If
1861         Next j
1862     ElseIf (CInt(Misci.Cells(2, 3)) = 0) Then
1863  'Next Point First T in Schedule, Use Previous Compositions for This T
1864         Misci.Cells(2, 5) = 0
1865         Misci.Cells(2, 4) = 0
1866         Comp.Cells(2, 1) = Comp.Cells(2, 8)
1867         Comp.Cells(2, 2) = Comp.Cells(2, 3)
1868         Misci.Cells(2, 6) = lsav
1869         Indx.Cells(2, 13) = 0
1870         For j = Indx.Cells(2, 6) To Indx.Cells(2, 15)
1871             En.Cells(j, Indx.Cells(2, 14)) = Comp.Cells(j + 1, 7)
1872             If CDbl(En.Cells(j, Indx.Cells(2, 14))) > 0 Then
1873                 Indx.Cells(2, 13) = Indx.Cells(2, 13) + 1
1874                 Indx.Cells(Indx.Cells(2, 13) + 1, 20) = j
1875             End If
1876         Next j
1877         For j = 1 To Indx.Cells(2, 5)
1878             En.Cells(j, Indx.Cells(2, 14)) = 0
1879             Comp.Cells(j + 1, 6) = Comp.Cells(j + 1, 7)
1880             If CDbl(Comp.Cells(j + 1, 7)) <> 0 Then
1881                 If ((Comp.Cells(j + 1, 6) - Comp.Cells(2, 2) + 18.5) > 0) Then En.Cells(j,
Indx.Cells(2, 14)) = Exp(Comp.Cells(j + 1, 6))
1882             End If
1883         Next j
1884         If (Miscl.Cells(2, 17) = False) Then Miscr.Cells(2, 15) = tsave 'Miscl.Cells(2,17)
is Tp
1885         Comp.Cells(2, 4) = Comp.Cells(2, 1) 'Comp.Cells(2,4) is Sumn
1886     ElseIf CInt(Misci.Cells(2, 3)) > 0 Then
1887  'Use Compositions From Previous Point
1888         For j = 1 To Indx.Cells(2, 15)
1889             En.Cells(j, Indx.Cells(2, 14)) = En.Cells(j, Misci.Cells(2, 3))
1890         Next j
1891     End If
1892  End Sub
1893
1894
'*************************************************************************************************
*****************************************************
1895  'Subroutines for the Program Screens
1896  'Created By Jake Rumel
1897
'*************************************************************************************************
*****************************************************
1898
1899  '*********************************************************************
1900  'Parameters
1901  '*********************************************************************
1902  'This Sub Returns the Parameters to a suggested Default - Large
1903  Sub Parameter DefaultLarge()
1904      With SetParameters
1905          .TextBox1 = 600
1906          .TextBox2 = 300
1907          .TextBox3 = 12
1908          .TextBox4 = 50
1909          .TextBox5 = 50
1910          .TextBox6 = 24
1911          .TextBox7 = 20
1912          .TextBox8 = 400
```

```
1913          .TextBox9 = 52
1914          .TextBox10 = 51
1915          .TextBox11 = 26
1916      End With
1917  End Sub
1918  'This Sub Returns the Parameters to a suggested Default - Large
1919  Sub Parameter_DefaultSmall()
1920      With SetParameters
1921          .TextBox1 = 300
1922          .TextBox2 = 200
1923          .TextBox3 = 12
1924          .TextBox4 = 40
1925          .TextBox5 = 40
1926          .TextBox6 = 24
1927          .TextBox7 = 15
1928          .TextBox8 = 200
1929          .TextBox9 = 52
1930          .TextBox10 = 51
1931          .TextBox11 = 26
1932      End With
1933  End Sub
1934  'This pulls the current parameters for the SetParameters Screen
1935  Sub Pull_Parameters()
1936      With SetParameters
1937          .TextBox1 = Parameter.Cells(1, 2)
1938          .TextBox2 = Parameter.Cells(2, 2)
1939          .TextBox3 = Parameter.Cells(3, 2)
1940          .TextBox4 = Parameter.Cells(4, 2)
1941          .TextBox5 = Parameter.Cells(5, 2)
1942          .TextBox6 = Parameter.Cells(6, 2)
1943          .TextBox7 = Parameter.Cells(7, 2)
1944          .TextBox8 = Parameter.Cells(8, 2)
1945          .TextBox9 = Parameter.Cells(9, 2)
1946          .TextBox10 = Parameter.Cells(10, 2)
1947          .TextBox11 = Parameter.Cells(11, 2)
1948      End With
1949  End Sub
1950  'This updates the parameters from the SetParameters Screen
1951  Sub Save Parameters()
1952      ParamOk = True
1953      For i = 1 To 11
1954          If IsNumeric(SetParameters.Controls.Item("TextBox" & i)) = False Then ParamOk =
False
1955      Next i
1956          If ParamOk Then
1957          For i = 1 To 11
1958              Parameter.Cells(i, 2) = SetParameters.Controls.Item("TextBox" & i)
1959          Next i
1960      Else
1961          MsgBox ("Parameters must be numeric values")
1962      End If
1963  End Sub
1964
1965  '*********************************************************************
1966  'ThermoRanges
1967  '*********************************************************************
1968  'This loads the Temperatures Ranges for the ThermoRanges Screen
1969  Sub ThermoRanges_Load()
1970      With ThermoRanges
1971          .TextBox1 = Format(ther.Cells(3, 2), "#,##0.000")
1972          .TextBox1.Locked = True
1973          .TextBox1.ForeColor = vbBlack
1974          .TextBox2 = Format(ther.Cells(3, 3), "#,##0.000")
1975          .TextBox2.Locked = True
1976          .TextBox3 = Format(ther.Cells(3, 4), "#,##0.000")
1977          .TextBox3.Locked = True
1978          .TextBox4 = Format(ther.Cells(3, 5), "#,##0.000")
1979          .TextBox4.Locked = True
1980          .TextBox5 = ther.Cells(3, 6)
1981          .TextBox5.Locked = True
1982      End With
```

```
End Sub


'*********************************************************************
'ThermoData
'*********************************************************************
'This sets up the ThermoData Screen
Sub ThermoData Setup()
    If ThermoData.DisplayProdReac.ListCount = 0 Then
        ThermoData.DisplayProdReac.AddItem "Products", 0
        ThermoData.DisplayProdReac.AddItem "Reactants", 1
        ThermoData.DisplayProdReac.ListIndex = 0
    End If
End Sub

'This gets the current selection of Reactants or Products for the Details Screen
Sub Get_ReacOrProd()
    If ThermoData.DisplayProdReac.ListIndex = 0 Then
        prodrow = ThermoData.ListProdReac.ListIndex + 2
        ThermoDetails.Caption = "Thermodynamic Properties - " & prod.Cells(prodrow, 1)
    ElseIf ThermoData.DisplayProdReac.ListIndex = 1 Then
        reacRow = ThermoData.ListProdReac.ListIndex + 2
        ThermoDetails.Caption = "Thermodynamic Properties - " & reac.Cells(reacRow, 1)
    End If
End Sub
'This pulls the List of Reactants or Products
Sub ThermoData Refresh()
    Dim rows As Long
    If ThermoData.DisplayProdReac.ListIndex = 0 Then
        ProdReac = 0
        rows = prod.UsedRange.rows.Count
        If rows > 2 Then ThermoData.ListProdReac.List = prod.Range("A2:B" & rows).Value
    ElseIf ThermoData.DisplayProdReac.ListIndex = 1 Then
        ProdReac = 1
        rows = reac.UsedRange.rows.Count
        If rows > 2 Then ThermoData.ListProdReac.List = reac.Range("A2:B" & rows).Value
    End If
    ThermoData.ButtonViewThermo.Enabled = False
End Sub

'*********************************************************************
'ThermoDetails
'*********************************************************************
'This pulls the details for the Select Reactant or Product
Sub ThermoDetails Load()
    Dim row As Integer
    Dim sht As Worksheet
    If ProdReac = 0 Then
        row = prodrow
        Set sht = ThisWorkbook.Worksheets("PRODUCTS")
        If sht.Cells(row, 19) = "" Then
            ThermoDetails.ButtonThermo.Enabled = False
        Else
            ThermoDetails.ButtonThermo.Enabled = True
        End If
    ElseIf ProdReac = 1 Then
        row = reacRow
        Set sht = ThisWorkbook.Worksheets("REACTANTS")
        If sht.Cells(row, 20) = "" Then
            ThermoDetails.ButtonThermo.Enabled = False
        Else
            ThermoDetails.ButtonThermo.Enabled = True
        End If
    End If
    With ThermoDetails
        .Label2 = sht.Cells(row, 1)
        .TextBox1 = sht.Cells(row, 2)
        .TextBox1.Locked = True
        .TextBox17 = sht.Cells(row, 3)
        .TextBox17.Locked = True
        .TextBox2 = sht.Cells(row, 4)
```

```
2054          .TextBox2.Locked = True
2055          .TextBox3 = sht.Cells(row, 5)
2056          .TextBox3.Locked = True
2057          If sht.Cells(row, 10) <> 0 Then
2058              .TextBox4 = sht.Cells(row, 10)
2059          Else
2060              .TextBox4 = ""
2061          End If
2062          .TextBox4.Locked = True
2063          .TextBox5 = sht.Cells(row, 6)
2064          .TextBox5.Locked = True
2065          If sht.Cells(row, 11) <> 0 Then
2066              .TextBox6 = sht.Cells(row, 11)
2067          Else
2068              .TextBox6 = ""
2069          End If
2070          .TextBox6.Locked = True
2071          .TextBox7 = sht.Cells(row, 7)
2072          .TextBox7.Locked = True
2073          If sht.Cells(row, 12) <> 0 Then
2074              .TextBox8 = sht.Cells(row, 12)
2075          Else
2076              .TextBox8 = ""
2077          End If
2078          .TextBox8.Locked = True
2079          .TextBox9 = sht.Cells(row, 8)
2080          .TextBox9.Locked = True
2081          If sht.Cells(row, 13) <> 0 Then
2082              .TextBox10 = sht.Cells(row, 13)
2083          Else
2084              .TextBox10 = ""
2085          End If
2086          .TextBox10.Locked = True
2087          .TextBox11 = sht.Cells(row, 9)
2088          .TextBox11.Locked = True
2089          If sht.Cells(row, 14) <> 0 Then
2090              .TextBox12 = sht.Cells(row, 14)
2091          Else
2092              .TextBox12 = ""
2093          End If
2094          .TextBox12.Locked = True
2095          .TextBox13 = sht.Cells(row, 18)
2096          .TextBox13.Locked = True
2097          If ProdReac = 1 Then
2098              .TextBox14 = Format(sht.Cells(row, 19), "#,##0.000")
2099          Else
2100              .TextBox14 = ""
2101          End If
2102          .TextBox14.Locked = True
2103          .TextBox15 = Format(sht.Cells(row, 16), "#,##0.000")
2104          .TextBox15.Locked = True
2105          If CDbl(sht.Cells(row, 17)) > 0 Then
2106              .TextBox16 = Format(sht.Cells(row, 17), "#,##0.000")
2107          Else
2108              .TextBox16 = ""
2109          End If
2110          .TextBox16.Locked = True
2111      End With
2112
2113  End Sub
2114
2115
2116  '*********************************************************************
2117  'ThermoInput
2118  '*********************************************************************
2119  'This loads the Thermo information for the Product or React for the specified interval
2120  Sub Thermo_Display()
2121      Dim row, tint As Integer
2122      Dim sht As Worksheet
2123      If ProdReac = 0 Then
2124          row = prodrow
```

```
2125          Set sht = ThisWorkbook.Worksheets("PRODUCTS")
2126      ElseIf ProdReac = 1 Then
2127          row = reacRow
2128          Set sht = ThisWorkbook.Worksheets("REACTANTS")
2129      End If
2130      With ThermoInput
2131          .Caption = "THERMO DETAILS - " & sht.Cells(row, 1)
2132          For i = 1 To 9
2133              For k = 1 To 3
2134              tint = i + 9 * (k - 1)
2135          .Controls("TextBox" & tint) = sht.Cells(row, tint + ProdReac + 18)
2136              Next k
2137          Next i
2138      End With
2139  End Sub
2140
2141  '*********************************************************************
2142  'ReadThermoProgress
2143  '*********************************************************************
2144  Sub OpenThermo()
2145
2146
2147      Dim Temp As String
2148
2149      ChDir (ThisWorkbook.Path)
2150      thermoFilePath = Application.GetOpenFilename("Input File(*.inp), *.inp")
2151      If thermoFilePath <> "False" And thermoFilePath <> "" Then
2152          Call Form_Position(ReadThermoProgress)
2153          ReadThermoProgress.Show
2154      End If
2155
2156  End Sub
2157  'This reads the thermo data from the input file
2158  Sub ReadThermo()
2159
2160      Dim name, rdate, symb(4), note As String
2161      Dim thermoText As String
2162      Dim fill(3), products As Boolean
2163      Dim readrow, ifzm1, ifaz, fno(4), ncoef, intr, inew, kk As Integer
2164      Dim prodrow, reacRow As Integer
2165      Dim ntl, num, ngl, ns, nall As Integer
2166      Dim aa, atms, tinf, tl(1), mwt, hform, thermo(8, 2), tgl(3), expn(7), hh, templ(8),
ttl, cpfix, dtl, tex, tx As Double
2167
2168      Dim rtgl() As String
2169      Dim sym() As String
2170
2171      If Thermoinp.UsedRange.rows.Count > 1 Then Thermoinp.Range("A2", "AS" &
Thermoinp.UsedRange.rows.Count).ClearContents
2172      If prod.UsedRange.rows.Count > 1 Then prod.Range("A2", "AS" &
prod.UsedRange.rows.Count).ClearContents
2173      If reac.UsedRange.rows.Count > 1 Then reac.Range("A2", "AT" &
reac.UsedRange.rows.Count).ClearContents
2174
2175      thermCount = 0
2176      readrow = 2
2177      prodrow = 2
2178      reacRow = 2
2179
2180      ngl = 0
2181      ns = 0
2182      nall = 0
2183      ifzm1 = 0
2184      inew = 0
2185      tinf = 10 ^ 6
2186
2187      Open thermoFilePath For Input As #1
2188
2189  50:        Line Input #1, thermoText
2190
2191  'Skip the Comments in the Text File
```

```
2192      If Left(thermoText, 1) = "!" Then GoTo 50
2193
2194  'Determines if the line reads thermo
2195      If Left(thermoText, 6) = "thermo" Then
2196          Line Input #1, thermoText
2197            rtgl() = Split(thermoText, " ")
2198            For Each Temp In rtgl
2199              If Temp <> "" Then
2200                  thermCount = thermCount + 1
2201                  If thermCount >= 1 And thermCount < 5 Then
2202                      ther.Cells(3, thermCount + 1) = Format(Temp, "General Number")
2203                      tgl(thermCount - 1) = CDbl(Temp)
2204                  ElseIf thermCount = 5 Then
2205                      ther.Cells(3, thermCount + 1) = Format(Temp, "mm/dd/yy")
2206                  End If
2207              End If
2208            Next Temp
2209      End If
2210
2211  100: For i = 1 To 3
2212          fill(i - 1) = True
2213          For j = 1 To 9
2214              thermo(j - 1, i - 1) = 0
2215          Next j
2216      Next i
2217      hform = 0
2218      tl(0) = 0
2219      tl(1) = 0
2220      Line Input #1, thermoText
2221
2222  'Reads the Product/Reactant Name and Notes
2223      sym = Split(thermoText, " ")
2224      name = Trim(sym(0))
2225      note = Trim(Right(thermoText, (Len(thermoText) - Len(sym(0)))))
2226
2227  'Checks if it is the End of the Products or End of the File
2228      If Left(thermoText, 3) = "END" Or Left(thermoText, 3) = "end" Then
2229          If InStr(thermoText, "ROD") = 0 And InStr(thermoText, "rod") = 0 Then
2230          GoTo 300
2231          End If
2232          ns = nall
2233          GoTo 100
2234      End If
2235  'Reads in the NTL, Date, Sym's, Fno's, Ifaz, Mwt, and Hfrom
2236      Line Input #1, thermoText
2237      ntl = CInt(Trim(Mid(thermoText, 2, 1)))
2238      rdate = Trim(Mid(thermoText, 4, 7))
2239      For j = 1 To 5
2240          symb(j - 1) = Trim(Mid(thermoText, (3 + 8 * j), 2))
2241          fno(j - 1) = CDbl(Trim(Mid(thermoText, (5 + 8 * j), 5)))
2242      Next j
2243      ifaz = CInt(Trim(Mid(thermoText, 51, 2)))
2244      mwt = CDbl(Trim(Mid(thermoText, 53, 13)))
2245      hform = CDbl(Trim(Mid(thermoText, 66, 15)))
2246
2247  'IF NTL=0, REACTANT WITHOUT COEFFICIENTS
2248      If ntl = 0 Then
2249          If ns = 0 Then GoTo 300
2250          nall = nall + 1
2251          Line Input #1, thermoText
2252          tl(0) = CDbl(Trim(Mid(thermoText, 2, 10)))
2253          tl(1) = CDbl(Trim(Mid(thermoText, 13, 10)))
2254          ncoef = CInt(Mid(thermoText, 23, 1))
2255          For j = 0 To 7
2256              expn(j) = CDbl(Trim(Mid(thermoText, 24 + 5 * j, 5)))
2257          Next j
2258          hh = CDbl(Trim(Mid(thermoText, 66, 15)))
2259          thermo(0, 0) = hform
2260          With Thermoinp
2261              .Cells(readrow, 1) = name
2262              .Cells(readrow, 2) = note
```

```
2263              .Cells(readrow, 3) = ntl
2264              .Cells(readrow, 4) = rdate
2265              For j = 0 To 4
2266                  .Cells(readrow, j + 5) = symb(j)
2267                  .Cells(readrow, j + 10) = fno(j)
2268              Next j
2269              .Cells(readrow, 15) = ifaz
2270              .Cells(readrow, 16) = tl(0)
2271              .Cells(readrow, 17) = tl(1)
2272              .Cells(readrow, 18) = mwt
2273              For i = 0 To 2
2274                  For j = 0 To 8
2275                      .Cells(readrow, j + i * 9 + 19) = thermo(j, i)
2276                  Next j
2277              Next i
2278          End With
2279          readrow = readrow + 1
2280          GoTo 100
2281      ElseIf name = "Air" Then
2282          sym(0) = "N"
2283          fno(0) = 1.56168
2284          sym(1) = "O"
2285          fno(1) = 0.41959
2286          sym(2) = "AR"
2287          fno(2) = 0.009365
2288          sym(3) = "C"
2289          fno(3) = 0.000319
2290      ElseIf name = "e-" Then
2291          mwt = 5.48579903 * 10 ^ (-4)
2292      End If
2293   'IF NTL NOT EQUAL TO 0
2294      For i = 1 To ntl
2295          Line Input #1, thermoText
2296          tl(0) = CDbl(Trim(Mid(thermoText, 2, 10)))
2297          tl(1) = CDbl(Trim(Mid(thermoText, 13, 10)))
2298          ncoef = CInt(Mid(thermoText, 23, 1))
2299          For j = 0 To 7
2300              expn(j) = CDbl(Trim(Mid(thermoText, 24 + 5 * j, 5)))
2301          Next j
2302          hh = CDbl(Trim(Mid(thermoText, 66, 15)))
2303          Line Input #1, thermoText
2304          For j = 0 To 4
2305              templ(j) = CDbl(Mid(thermoText, j * 16 + 1, 12)) * 10 ^ (CDbl(Mid(thermoText, j
* 16 + 14, 3)))
2306          Next j
2307          Line Input #1, thermoText
2308          For j = 0 To 1
2309              templ(j + 5) = CDbl(Mid(thermoText, j * 16 + 1, 12)) * 10 ^
(CDbl(Mid(thermoText, j * 16 + 14, 3)))
2310              If Mid(thermoText, j * 16 + 62, 1) = "+" Then
2311                  templ(j + 7) = CDbl(Mid(thermoText, j * 16 + 49, 12)) * 10 ^
(Abs(Mid(thermoText, j * 16 + 63, 2)))
2312              Else
2313                  templ(j + 7) = CDbl(Mid(thermoText, j * 16 + 49, 12)) * 10 ^ (-
Abs(Mid(thermoText, j * 16 + 63, 2)))
2314              End If
2315          Next j
2316          If ifaz = 0 And i > 3 Then
2317   'ERROR MESSAGE
2318   MsgBox ("Error in processing thermo.inp")
2319              Exit Sub
2320          End If
2321          If ifaz <= 0 Then
2322              If tl(1) > (tgl(3) - 0.01) Then
2323                  ifaz = -1
2324                  name = "*" & name
2325              End If
2326              If tl(0) >= tgl(i) Then GoTo 200
2327              intr = i
2328              fill(i - 1) = False
2329          Else
```

```
2330             intr = 1
2331             If i > 1 Then
2332                 For k = 0 To 6
2333                     thermo(k, 1) = 0
2334                 Next k
2335             End If
2336         End If
2337         For l = 0 To ncoef - 1
2338             For k = 0 To 6
2339                 If expn(l) = CDbl(k - 2) Then
2340                     thermo(k, intr - 1) = templ(l)
2341                     GoTo 150
2342                 End If
2343             Next k
2344 150:    Next l
2345         thermo(7, intr - 1) = templ(7)
2346         thermo(8, intr - 1) = templ(8)
2347         If ifaz > 0 Then
2348             nall = nall + 1
2349             If ifaz > ifzm1 Then
2350                 inew = inew + 1
2351             Else
2352                 inew = i
2353             End If
2354             With Thermoinp
2355                 .Cells(readrow, 1) = name
2356                 .Cells(readrow, 2) = note
2357                 .Cells(readrow, 3) = ntl
2358                 .Cells(readrow, 4) = rdate
2359                 For j = 0 To 4
2360                     .Cells(readrow, j + 5) = symb(j)
2361                     .Cells(readrow, j + 10) = fno(j)
2362                 Next j
2363                 .Cells(readrow, 15) = inew
2364                 .Cells(readrow, 16) = tl(0)
2365                 .Cells(readrow, 17) = tl(1)
2366                 .Cells(readrow, 18) = mwt
2367                 For k = 0 To 2
2368                     For j = 0 To 8
2369                         .Cells(readrow, j + k * 9 + 19) = thermo(j, k)
2370                     Next j
2371                 Next k
2372             End With
2373             readrow = readrow + 1
2374         End If
2375 200: Next i
2376     ifzm1 = ifaz
2377     If ifaz <= 0 Then
2378         inew = 0
2379         nall = nall + 1
2380         If ifaz <= 0 And ns = 0 Then
2381             ngl = ngl + 1
2382             If fill(2) Then
2383                 atms = 0
2384                 For i = 0 To 4
2385                     If symb(i) = "" Or symb(i) = "E" Or symb(i) = " " Then GoTo 210
2386                     atms = atms + fno(i)
2387                 Next i
2388 'For Gases with no coefficients for TGL(3) - TGL(4) interval,
2389 'Calculate estimated coefficients. (Straight Line for CP/R)
2390 210:            aa = 2.5
2391                 If atms > 1.9 Then aa = 4.5
2392                 If atms > 2.1 Then aa = 3 * atms - 1.75
2393                 ttl = tl(1)
2394                 tx = ttl - tinf
2395                 cpfix = 0
2396                 templ(7) = 0
2397                 templ(8) = 0
2398                 dlt = Log(ttl)
2399                 For k = 6 To 0 Step -1
2400                     kk = k - 2
```

```
2401                    If kk = 0 Then
2402                        cpfix = cpfix = thermo(k, 1)
2403                        templ(7) = templ(7) + thermo(k, 1)
2404                        templ(8) = templ(8) + thermo(k, 1) * dlt
2405                    Else
2406                        tex = ttl ^ kk
2407                        cpfix = cpfix + thermo(k, 1) * tex
2408                        templ(8) = templ(8) + thermo(k, 1) * tex / kk
2409                        If kk = -1 Then
2410                            templ(7) = templ(7) + thermo(k, 1) * dlt / ttl
2411                        Else
2412                            templ(7) = templ(7) + thermo(k, 1) * tex / (kk + 1)
2413                        End If
2414                    End If
2415                Next k
2416                templ(1) = (cpfix - aa) / tx
2417                thermo(3, 2) = templ(1)
2418                templ(0) = cpfix - ttl * templ(1)
2419                thermo(2, 2) = templ(0)
2420                thermo(7, 2) = thermo(7, 1) + ttl * (templ(7) - templ(0) - 0.5 * templ(1) *
ttl)
2421                thermo(8, 2) = -templ(0) * dlt + thermo(8, 1) + templ(8) - templ(1) * ttl
2422            End If
2423        End If
2424  'Write Coefficients on Thermoinp
2425        With Thermoinp
2426            .Cells(readrow, 1) = name
2427            .Cells(readrow, 2) = note
2428            .Cells(readrow, 3) = ntl
2429            .Cells(readrow, 4) = rdate
2430            For j = 0 To 4
2431                .Cells(readrow, j + 5) = symb(j)
2432                .Cells(readrow, j + 10) = fno(j)
2433            Next j
2434            .Cells(readrow, 15) = ifaz
2435            If ifaz < 1 Then
2436                .Cells(readrow, 16) = tgl(0)
2437                If ns = 0 Then
2438                    .Cells(readrow, 17) = tgl(3)
2439                Else
2440                    .Cells(readrow, 17) = tl(1)
2441                End If
2442            Else
2443                .Cells(readrow, 16) = tl(0)
2444                .Cells(readrow, 17) = tl(1)
2445            End If
2446            .Cells(readrow, 18) = mwt
2447            For i = 0 To 2
2448                For j = 0 To 8
2449                    .Cells(readrow, j + i * 9 + 19) = thermo(j, i)
2450                Next j
2451            Next i
2452        End With
2453        readrow = readrow + 1
2454    End If
2455    GoTo 100
2456  'END OF DATA. Copy Condensed & reactant Data from Thermoinp and add to PRODUCTS and
REACTANTS
2457  300: If ns = 0 Then ns = nall
2458  'Write Gaseous Products on PRODUCTS
2459    If ngl <> 0 Then
2460        For i = 1 To ns
2461            If Thermoinp.Cells(i + 1, 15) <= 0 Then
2462                For j = 1 To 45
2463                    prod.Cells(prodrow, j) = Thermoinp.Cells(i + 1, j)
2464                Next j
2465                For j = 1 To 19
2466                    reac.Cells(reacRow, j) = Thermoinp.Cells(i + 1, j)
2467                Next j
2468                If Thermoinp.Cells(i + 1, 3) > 0 Then
2469                    For j = 19 To 45
```

```
2470                        reac.Cells(reacRow, j + 1) = Thermoinp.Cells(i + 1, j)
2471                    Next j
2472                End If
2473                reacRow = reacRow + 1
2474                prodrow = prodrow + 1
2475            End If
2476        Next i
2477    End If
2478    If ngl <> nall Then
2479 'Write Condensed Products and Reactants on PRODUCTS and REACTANTS
2480        For i = 1 To nall
2481            If i > ns Then
2482                For j = 1 To 19
2483                    reac.Cells(reacRow, j) = Thermoinp.Cells(i + 1, j)
2484                Next j
2485                If Thermoinp.Cells(i + 1, 3) > 0 Then
2486                    For j = 19 To 45
2487                        reac.Cells(reacRow, j + 1) = Thermoinp.Cells(i + 1, j)
2488                    Next j
2489                End If
2490                reacRow = reacRow + 1
2491            ElseIf (Thermoinp.Cells(i + 1, 15) > 0) Then
2492                For j = 1 To 27
2493                    prod.Cells(prodrow, j) = Thermoinp.Cells(i + 1, j)
2494                Next j
2495                For j = 1 To 19
2496                    reac.Cells(reacRow, j) = Thermoinp.Cells(i + 1, j)
2497                Next j
2498                If Thermoinp.Cells(i + 1, 3) > 0 Then
2499                    For j = 19 To 45
2500                        reac.Cells(reacRow, j + 1) = Thermoinp.Cells(i + 1, j)
2501                    Next j
2502                End If
2503                reacRow = reacRow + 1
2504                prodrow = prodrow + 1
2505            End If
2506        Next i
2507    End If
2508    Indx.Cells(2, 26) = nall
2509    Indx.Cells(2, 25) = ns
2510    Indx.Cells(2, 24) = ngl
2511    Close #1
2512
2513 'Creates the REAC_LIST
2514    Dim rowsReac, rowsProd, rowsInt As Long
2515    rowsReac = reac.UsedRange.rows.Count
2516    rowsInt = reac_list.UsedRange.rows.Count
2517    If rowsInt > 1 Then reac_list.Range("A2", "B" & rowsInt).ClearContents
2518    For i = 2 To rowsReac
2519        If Left(reac.Cells(i, 1), 1) = "*" Then
2520            reac_list.Cells(i, 1) = Right(reac.Cells(i, 1), Len(reac.Cells(i, 1)) - 1)
2521        Else
2522            reac_list.Cells(i, 1) = reac.Cells(i, 1)
2523        End If
2524        reac_list.Cells(i, 2) = i
2525    Next i
2526    rowsInt = reac_list.UsedRange.rows.Count
2527    reac_list.Columns("A:B").Sort key1:=reac_list.Cells(1, 1), Header:=xlYes
2528
2529 'Creates the PROD_LIST
2530    rowsProd = prod.UsedRange.rows.Count
2531    rowsInt = prod list.UsedRange.rows.Count
2532    If rowsInt > 1 Then prod_list.Range("A2", "A" & rowsInt).ClearContents
2533    For i = 2 To rowsProd
2534        If Left(prod.Cells(i, 1), 1) = "*" Then
2535            If Right(prod.Cells(i, 1), Len(prod.Cells(i, 1)) - 1) <> prod_list.Cells(i - 1,
1) Then prod_list.Cells(i, 1) = Right(prod.Cells(i, 1), Len(prod.Cells(i, 1)) - 1)
2536        Else
2537            If prod.Cells(i, 1) <> prod_list.Cells(i - 1, 1) Then prod_list.Cells(i, 1) =
prod.Cells(i, 1)
2538        End If
```

```
2539      Next i
2540      rowsInt = prod_list.UsedRange.rows.Count
2541      prod_list.Columns("A:A").Sort key1:=prod_list.Cells(1, 1), Header:=xlYes
2542
2543
2544  'Signal Processing has completed
2545      ReadThermoProgress.Label1 = "File Loading Complete"
2546      ReadThermoProgress.ButtonClose.Enabled = True
2547      ReadThermoProgress.Repaint
2548
2549  End Sub
2550
2551
2552  '*********************************************************************
2553  'ProbInput
2554  '*********************************************************************
2555  'This will add a Reactant to the Problem List
2556  Sub Prob_Input_Add_Reac()
2557      For i = 0 To ProbInput.ListBox1.ListCount - 1
2558          If ProbInput.ListBox1.Selected(i) = True Then
2559              If Reactn.Cells(2, 1) >= Parameter.Cells(6, 2) Then
2560                  MsgBox ("Maximum Number of Allowable Reactants Reached.")
2561                  GoTo 200
2562              Else
2563                  If Reactn.Cells(2, 1) > 0 Then
2564                      For j = 2 To Reactn.Cells(2, 1) + 1
2565                          If ProbInput.ListBox1.List(i) = Cdata.Cells(j, 11) Then GoTo 100
2566                      Next j
2567                  End If
2568                  Reactn.Cells(2, 1) = Reactn.Cells(2, 1) + 1
2569                  Cdata.Cells(Reactn.Cells(2, 1) + 1, 11) = ProbInput.ListBox1.List(i)
2570                  If reac_std Then
2571                      Cdata.Cells(Reactn.Cells(2, 1) + 1, 26) = reac list.Cells(i + 2, 2) - 1
2572                  Else
2573                      Cdata.Cells(Reactn.Cells(2, 1) + 1, 26) = reac_int.Cells(i + 2, 2) - 1
2574                  End If
2575                  Call InitiatizeReactant
2576              End If
2577          End If
2578  100:    ProbInput.ListBox1.Selected(i) = False
2579      Next i
2580  200: Call GetAssignedReac
2581      Call Check_Input_Status
2582  End Sub
2583  'This will Remove a Reactant from the Problem List
2584  Sub Prob_Input_Remove_Reactant()
2585      Dim row As Integer
2586      row = Reactn.Cells(2, 1) + 1
2587      For i = 0 To ProbInput.ListBox2.ListCount - 1
2588          If ProbInput.ListBox2.Selected(i) = True Then
2589              For j = i + 2 To row - 1
2590                  Cdata.Cells(j, 11) = Cdata.Cells(j + 1, 11)
2591                  For m = 13 To 24
2592                      Cdata.Cells(j, m) = Cdata.Cells(j + 1, m)
2593                  Next m
2594                  Cdata.Cells(j, 7) = Cdata.Cells(j + 1, 7)
2595                  Cdata.Cells(j, 4) = Cdata.Cells(j + 1, 4)
2596                  Indx.Cells(j, 22) = Indx.Cells(j + 1, 22)
2597                  For m = 2 To 19
2598                      Reactn.Cells(j, m) = Reactn.Cells(j + 1, m)
2599                  Next m
2600              Next j
2601              Cdata.Cells(row, 11).ClearContents
2602              For m = 13 To 24
2603                  Cdata.Cells(row, m).ClearContents
2604              Next m
2605              Cdata.Cells(row, 7).ClearContents
2606              Cdata.Cells(row, 4).ClearContents
2607              Cdata.Cells(row, 26).ClearContents
2608              Indx.Cells(row, 22).ClearContents
2609              Reactn.Cells(row, 5) = -1
```

```
2610              Reactn.Cells(row, 3).ClearContents
2611              Reactn.Cells(row, 4).ClearContents
2612              Reactn.Cells(row, 2).ClearContents
2613              For m = 6 To 19
2614                  Reactn.Cells(row, m).ClearContents
2615              Next m
2616              Reactn.Cells(2, 1) = Reactn.Cells(2, 1) - 1
2617          End If
2618      Next i
2619      Call GetAssignedReac
2620      Call Check_Input_Status
2621  End Sub
2622  'This will add a Product to the Problem Insert List
2623  Sub Prob Input Insert Prod()
2624      For i = 0 To ProbInput.ListBox3.ListCount - 1
2625          If ProbInput.ListBox3.Selected(i) = True Then
2626              If Indx.Cells(2, 16) >= 20 Then
2627                  MsgBox ("Maximum Number of Allowable Inserted Products Reached.")
2628                  GoTo 200
2629              Else
2630                  If Indx.Cells(2, 16) > 0 Then
2631                      For j = 2 To Indx.Cells(2, 16) + 1
2632                          If ProbInput.ListBox3.List(i) = Cdata.Cells(j, 25) Then GoTo 100
2633                      Next j
2634                  End If
2635                  If Indx.Cells(2, 10) > 0 Then
2636                      For k = 2 To Indx.Cells(2, 10) + 1
2637                          If ProbInput.ListBox3.List(i) = Cdata.Cells(k, 8) Then Call
RemoveProduct(k, 10, 8)
2638                      Next k
2639                  End If
2640                  Indx.Cells(2, 16) = Indx.Cells(2, 16) + 1
2641                  Cdata.Cells(Indx.Cells(2, 16) + 1, 25) = ProbInput.ListBox3.List(i)
2642              End If
2643          End If
2644  100:    ProbInput.ListBox3.Selected(i) = False
2645      Next i
2646  200: Call GetProd
2647  End Sub
2648  'This will add a Product to the Problem Omit List
2649  Sub Prob_Input_Omit_Prod()
2650      For i = 0 To ProbInput.ListBox3.ListCount - 1
2651          If ProbInput.ListBox3.Selected(i) = True Then
2652              If Indx.Cells(2, 10) >= Parameter.Cells(1, 2) Then
2653                  MsgBox ("Maximum Number of Allowable Omitted Products Reached.")
2654                  GoTo 200
2655              Else
2656                  If Indx.Cells(2, 10) > 0 Then
2657                      For j = 2 To Indx.Cells(2, 10) + 1
2658                          If ProbInput.ListBox3.List(i) = Cdata.Cells(j, 8) Then GoTo 100
2659                      Next j
2660                  End If
2661                  If Indx.Cells(2, 11) > 0 Then
2662                      For k = 2 To Indx.Cells(2, 11) + 1
2663                          If ProbInput.ListBox3.List(i) = Cdata.Cells(k, 10) Then Call
RemoveProduct(k, 11, 10)
2664                      Next k
2665                  End If
2666                  If Indx.Cells(2, 16) > 0 Then
2667                      For l = 2 To Indx.Cells(2, 16) + 1
2668                          If ProbInput.ListBox3.List(i) = Cdata.Cells(l, 25) Then Call
RemoveProduct(l, 16, 25)
2669                      Next l
2670                  End If
2671                  Indx.Cells(2, 10) = Indx.Cells(2, 10) + 1
2672                  Cdata.Cells(Indx.Cells(2, 10) + 1, 8) = ProbInput.ListBox3.List(i)
2673              End If
2674          End If
2675  100:    ProbInput.ListBox3.Selected(i) = False
2676      Next i
2677  200: Call GetProd
```

```
2678  End Sub
2679  'This will add a Product to the Problem Only List
2680  Sub Prob_Input_Only_Prod()
2681      For i = 0 To ProbInput.ListBox3.ListCount - 1
2682          If ProbInput.ListBox3.Selected(i) = True Then
2683              If Indx.Cells(2, 11) >= Parameter.Cells(1, 2) Then
2684                  MsgBox ("Maximum Number of Allowable Products Reached.")
2685                  GoTo 200
2686              Else
2687                  If Indx.Cells(2, 11) > 0 Then
2688                      For j = 2 To Indx.Cells(2, 11) + 1
2689                          If ProbInput.ListBox3.List(i) = Cdata.Cells(j, 10) Then GoTo 100
2690                      Next j
2691                  End If
2692                  If Indx.Cells(2, 10) > 0 Then
2693                      For k = 2 To Indx.Cells(2, 10) + 1
2694                          If ProbInput.ListBox3.List(i) = Cdata.Cells(k, 8) Then Call
RemoveProduct(k, 10, 8)
2695                      Next k
2696                  End If
2697                  Indx.Cells(2, 11) = Indx.Cells(2, 11) + 1
2698                  Cdata.Cells(Indx.Cells(2, 11) + 1, 10) = ProbInput.ListBox3.List(i)
2699              End If
2700          End If
2701  100:    ProbInput.ListBox3.Selected(i) = False
2702      Next i
2703  200: Call GetProd
2704  End Sub
2705  'This will remove a Product from the Problem Insert List
2706  Sub Prob_Input_Remove_Insert()
2707      Dim row As Integer
2708      row = Indx.Cells(2, 16) + 1
2709      For i = 0 To ProbInput.ListBox6.ListCount - 1
2710          If ProbInput.ListBox6.Selected(i) = True Then
2711              If row > 1 Then
2712                  For j = 2 To row
2713                      If ProbInput.ListBox6.List(i) = Cdata.Cells(j, 25) Then
2714                          Call RemoveProduct(j, 16, 25)
2715                      End If
2716                  Next j
2717              End If
2718          End If
2719          ProbInput.ListBox6.Selected(i) = False
2720      Next i
2721      Call GetProd
2722  End Sub
2723  'This will remove a Product from the Problem Omit List
2724  Sub Prob_Input_Remove_Omit()
2725      Dim row As Integer
2726      row = Indx.Cells(2, 10) + 1
2727      For i = 0 To ProbInput.ListBox5.ListCount - 1
2728          If ProbInput.ListBox5.Selected(i) = True Then
2729              If row > 1 Then
2730                  For j = 2 To row
2731                      If ProbInput.ListBox5.List(i) = Cdata.Cells(j, 8) Then
2732                          Call RemoveProduct(j, 10, 8)
2733                      End If
2734                  Next j
2735              End If
2736          End If
2737          ProbInput.ListBox5.Selected(i) = False
2738      Next i
2739      Call GetProd
2740  End Sub
2741  'This will remove a Product from the Problem Only List
2742  Sub Prob_Input_Remove_Only()
2743      Dim row As Integer
2744      row = Indx.Cells(2, 11) + 1
2745      For i = 0 To ProbInput.ListBox4.ListCount - 1
2746          If ProbInput.ListBox4.Selected(i) = True Then
2747              If row > 1 Then
```

```
                For j = 2 To row
                    If ProbInput.ListBox4.List(i) = Cdata.Cells(j, 10) Then
                        Call RemoveProduct(j, 11, 10)
                    End If
                Next j
            End If
        End If
        ProbInput.ListBox4.Selected(i) = False
    Next i
    Call GetProd
End Sub
'This Updates the Debug Setting of the Problem
Sub Update_Debug_Status()
    If ProbInput.CheckBox1.Value = True Then
        Miscl.Cells(2, 3) = "TRUE"
        For i = 1 To Parameter.Cells(3, 2)
            ProbInput.ListBox7.AddItem i, (i - 1)
        Next i
    ElseIf ProbInput.CheckBox1.Value = False Then
        Miscl.Cells(2, 3) = "FALSE"
        ProbInput.ListBox7.Clear
        For i = 2 To Parameter.Cells(3, 2) + 1
            Miscl.Cells(i, 2) = "FALSE"
        Next i
    End If
End Sub
'This Updates the Ion Setting of the Problem
Sub Update_Ion_Status()
    If ProbInput.CheckBox2.Value = True Then
        Miscl.Cells(2, 8) = "TRUE"
    ElseIf ProbInput.CheckBox2.Value = False Then
        Miscl.Cells(2, 8) = "FALSE"
    End If
End Sub
'This Updates the SiUnits Setting of the Problem
Sub Update_SiUnit_Status()
    If ProbInput.CheckBox3.Value = True Then
        Miscl.Cells(2, 15) = "TRUE"
    ElseIf ProbInput.CheckBox3.Value = False Then
        Miscl.Cells(2, 15) = "FALSE"
    End If
End Sub
'This Updates the Massf Setting of the Problem
Sub Update_Massf_Status()
    If ProbInput.CheckBox4.Value = True Then
        Miscl.Cells(2, 9) = "TRUE"
    ElseIf ProbInput.CheckBox4.Value = False Then
        Miscl.Cells(2, 9) = "FALSE"
    End If
End Sub
'This Updates the Short Setting of the Problem
Sub Update Short Status()
    If ProbInput.CheckBox5.Value = True Then
        Miscl.Cells(2, 14) = "TRUE"
    ElseIf ProbInput.CheckBox5.Value = False Then
        Miscl.Cells(2, 14) = "FALSE"
    End If
End Sub
'This Changes the Problem for which Type is Selected
Sub Change_Problem_Type()
    Call SetFalse
    If ProbInput.ComboBox1.ListIndex = 0 Then
        Miscl.Cells(2, 17) = "TRUE"
    ElseIf ProbInput.ComboBox1.ListIndex = 1 Then
        Miscl.Cells(2, 7) = "TRUE"
    ElseIf ProbInput.ComboBox1.ListIndex = 2 Then
        Miscl.Cells(2, 16) = "TRUE"
    ElseIf ProbInput.ComboBox1.ListIndex = 3 Then
        Miscl.Cells(2, 17) = "TRUE"
        Miscl.Cells(2, 19) = "TRUE"
    ElseIf ProbInput.ComboBox1.ListIndex = 4 Then
```

```
2819          Miscl.Cells(2, 7) = "TRUE"
2820          Miscl.Cells(2, 19) = "TRUE"
2821      ElseIf ProbInput.ComboBox1.ListIndex = 5 Then
2822          Miscl.Cells(2, 16) = "TRUE"
2823          Miscl.Cells(2, 19) = "TRUE"
2824      End If
2825  End Sub
2826  'This pulls the Assigned Reactants for the Problem Input Page
2827  Sub GetAssignedReac()
2828      If CInt(Reactn.Cells(2, 1)) <> 0 Then
2829          If CInt(Reactn.Cells(2, 1)) > 1 Then
2830              ProbInput.ListBox2.List = Cdata.Range("K2:K" & (Reactn.Cells(2, 1) + 1)).Value
2831          Else
2832              ProbInput.ListBox2.Clear
2833              ProbInput.ListBox2.AddItem Cdata.Cells(2, 11).Value
2834          End If
2835          ProbInput.ButtonReacProp.Enabled = True
2836      Else
2837          ProbInput.ListBox2.Clear
2838          ProbInput.ButtonReacProp.Enabled = False
2839      End If
2840  End Sub
2841  'This pulls the Products for the Problem Input Page
2842  Sub GetProd()
2843      Cdata.Columns("H:H").Sort key1:=Cdata.Cells(1, 8), Header:=xlYes
2844      Cdata.Columns("J:J").Sort key1:=Cdata.Cells(1, 10), Header:=xlYes
2845      Cdata.Columns("Y:Y").Sort key1:=Cdata.Cells(1, 25), Header:=xlYes
2846      If Indx.Cells(2, 10) <> 0 Then
2847          If Indx.Cells(2, 10) > 1 Then
2848              ProbInput.ListBox5.List = Cdata.Range("H2:H" & (Indx.Cells(2, 10) + 1)).Value
2849          Else
2850              ProbInput.ListBox5.Clear
2851              ProbInput.ListBox5.AddItem Cdata.Cells(2, 8).Value
2852          End If
2853      Else
2854          ProbInput.ListBox5.Clear
2855      End If
2856      If Indx.Cells(2, 16) <> 0 Then
2857          If Indx.Cells(2, 16) > 1 Then
2858              ProbInput.ListBox6.List = Cdata.Range("Y2:Y" & (Indx.Cells(2, 16) + 1)).Value
2859          Else
2860              ProbInput.ListBox6.Clear
2861              ProbInput.ListBox6.AddItem Cdata.Cells(2, 25).Value
2862          End If
2863      Else
2864          ProbInput.ListBox6.Clear
2865      End If
2866      If Indx.Cells(2, 11) <> 0 Then
2867          If Indx.Cells(2, 11) > 1 Then
2868              ProbInput.ListBox4.List = Cdata.Range("J2:J" & (Indx.Cells(2, 11) + 1)).Value
2869          Else
2870              ProbInput.ListBox4.Clear
2871              ProbInput.ListBox4.AddItem Cdata.Cells(2, 10).Value
2872          End If
2873      Else
2874          ProbInput.ListBox4.Clear
2875      End If
2876  End Sub
2877  'This sets the Problem Types to False
2878  Sub SetFalse()
2879      With Miscl
2880          .Cells(2, 17) = "FALSE"
2881          .Cells(2, 7) = "FALSE"
2882          .Cells(2, 16) = "FALSE"
2883          .Cells(2, 19) = "FALSE"
2884          .Cells(2, 13) = "FALSE"
2885          .Cells(2, 4) = "FALSE"
2886      End With
2887  End Sub
2888  'This Removes a specified product
2889  Sub RemoveProduct(j, icol, ccol)
```

```
2890      Cdata.Cells(j, ccol).ClearContents
2891      Indx.Cells(2, icol) = Indx.Cells(2, icol) - 1
2892  End Sub
2893  'This initializes the Reactants
2894  Sub InitiatizeReactant()
2895      Indx.Cells(Reactn.Cells(2, 1) + 1, 22) = 0
2896      Cdata.Cells(Reactn.Cells(2, 1) + 1, 7) = "lib"
2897      Cdata.Cells(Reactn.Cells(2, 1) + 1, 4) = "FUEL"
2898      Reactn.Cells(Reactn.Cells(2, 1) + 1, 4) = 0
2899      Reactn.Cells(Reactn.Cells(2, 1) + 1, 2) = 0
2900      Reactn.Cells(Reactn.Cells(2, 1) + 1, 5) = -1
2901      Reactn.Cells(Reactn.Cells(2, 1) + 1, 8) = 0
2902      Reactn.Cells(Reactn.Cells(2, 1) + 1, 6) = 0
2903      Reactn.Cells(Reactn.Cells(2, 1) + 1, 7) = 0
2904  End Sub
2905  'This Runs to Reset all properties for the Problem Start
2906  Sub ProblemSetup()
2907      CaseOk = True
2908      Indx.Cells(2, 2) = 0
2909      Indx.Cells(2, 8) = 0
2910      Miscl.Cells(2, 11) = "FALSE"
2911      ProbInput.ButtonNext.Enabled = False
2912  'Reset Case Name
2913      Cdata.Cells(2, 6) = ""
2914  'Set fuel to True
2915      fuel = True
2916  'Reset Calculation Sheets
2917      En.Range("A1", En.Cells(En.UsedRange.rows.Count,
En.UsedRange.Columns.Count)).ClearContents
2918      If Comp.UsedRange.rows.Count > 1 Then Comp.Range("A2", "H" &
Comp.UsedRange.rows.Count).ClearContents
2919      If Inpt.UsedRange.rows.Count > 1 Then Inpt.Range("A2", "H" &
Inpt.UsedRange.rows.Count).ClearContents
2920      If Inpt.UsedRange.rows.Count > 1 Then Inpt.Range("J2", "N" &
Inpt.UsedRange.rows.Count).ClearContents
2921      If Inpt.UsedRange.rows.Count > 1 Then Inpt.Range("P2", "Q" &
Inpt.UsedRange.rows.Count).ClearContents
2922      Miscr.Cells(2, 4).ClearContents
2923      Miscr.Cells(2, 5).ClearContents
2924      Miscr.Cells(2, 7).ClearContents
2925      If Miscr.UsedRange.rows.Count > 1 Then Miscr.Range("J2", "U" &
Miscr.UsedRange.rows.Count).ClearContents
2926      A.Range("A1", A.Cells(A.UsedRange.rows.Count, A.UsedRange.Columns.Count)).ClearContents
2927      G.Range("A1", G.Cells(G.UsedRange.rows.Count, G.UsedRange.Columns.Count)).ClearContents
2928      Cdata.Range("A2", "A101").ClearContents
2929      If Cdata.UsedRange.rows.Count > 1 Then Cdata.Range("C2", "Y" &
Cdata.UsedRange.rows.Count).ClearContents
2930      Cphs.Range("A2", "C8").ClearContents
2931      If Prtout.UsedRange.rows.Count > 1 Then Prtout.Range("A2", "AF" &
Prtout.UsedRange.rows.Count).ClearContents
2932      If Reactn.UsedRange.rows.Count > 1 Then Reactn.Range("A2", "S" &
Reactn.UsedRange.rows.Count).ClearContents
2933      Therm.Cells(2, 1).ClearContents
2934      Therm.Range("C2", Therm.Cells(Therm.UsedRange.rows.Count,
Therm.UsedRange.Columns.Count)).ClearContents
2935      If Coef.UsedRange.rows.Count > 1 Then Coef.Range("A2", "AA" &
Coef.UsedRange.rows.Count).ClearContents
2936      If Gauss.UsedRange.rows.Count > 1 Then Gauss.Range("A2", "A" &
Gauss.UsedRange.rows.Count).ClearContents
2937      If Eqlbrm.UsedRange.rows.Count > 1 Then Eqlbrm.Range("A2", "D" &
Eqlbrm.UsedRange.rows.Count).ClearContents
2938  'Set the hr and ur = 1 * 10^30
2939      Miscr.Cells(5, 4) = 10 ^ 30
2940      Miscr.Cells(7, 4) = 10 ^ 30
2941  'Remove Only, Omit, and Insert Reactants/Products
2942      If Indx.Cells(2, 11) <> 0 Then
2943          For i = 2 To Indx.Cells(2, 11) + 1
2944              Cdata.Cells(i, 10).ClearContents
2945          Next i
2946          Indx.Cells(2, 11) = 0
2947      End If
```

```
2948      If Indx.Cells(2, 10) <> 0 Then
2949          For i = 2 To Indx.Cells(2, 10) + 1
2950              Cdata.Cells(i, 8).ClearContents
2951          Next i
2952          Indx.Cells(2, 10) = 0
2953      End If
2954      If Indx.Cells(2, 16) <> 0 Then
2955          For i = 2 To Indx.Cells(2, 16) + 1
2956              Cdata.Cells(i, 25).ClearContents
2957          Next i
2958          Indx.Cells(2, 16) = 0
2959      End If
2960  'Remove the Reactants
2961      If Reactn.Cells(2, 1) <> 0 Then
2962          For i = 2 To Reactn.Cells(2, 1) + 1
2963              Cdata.Cells(i, 11).ClearContents
2964              Reactn.Cells(i, 3).ClearContents
2965              Reactn.Cells(i, 4).ClearContents
2966              Reactn.Cells(i, 5).ClearContents
2967              Reactn.Cells(i, 7).ClearContents
2968              Cdata.Cells(i, 22) = ""
2969              Cdata.Cells(i, 7) = ""
2970              Reactn.Cells(i, 2).ClearContents
2971              Reactn.Cells(i, 8).ClearContents
2972              Reactn.Cells(i, 6).ClearContents
2973              Cdata.Cells(i, 4).ClearContents
2974              Cdata.Cells(i, 26).ClearContents
2975          Next i
2976          Reactn.Cells(2, 1) = 0
2977      End If
2978  'Set Trace to 0
2979      Miscr.Cells(2, 14) = 0
2980      ProbInput.TextBox1 = Miscr.Cells(2, 14)
2981      ProbInput.TextBox2 = 0
2982  'Set Moles to False
2983      Miscl.Cells(2, 10) = "FALSE"
2984  'Set SI units to True
2985      Miscl.Cells(2, 15) = "TRUE"
2986      ProbInput.CheckBox3.Value = True
2987  'Set Massf to False
2988      Miscl.Cells(2, 9) = "FALSE"
2989      ProbInput.CheckBox4.Value = False
2990  'Set Short to False
2991      Miscl.Cells(2, 14) = "FALSE"
2992      ProbInput.CheckBox5.Value = False
2993  'Set Debug to False
2994      ProbInput.CheckBox1.Value = False
2995      For i = 2 To Parameter.Cells(3, 2) + 1
2996          Miscl.Cells(i, 2) = "FALSE"
2997      Next i
2998      Miscl.Cells(2, 3) = "FALSE"
2999  'Set Nplt to 0
3000      Indx.Cells(2, 8) = 0
3001  'Set 1 to MAXR Pecwt to -1
3002      For i = 2 To Parameter.Cells(6, 2) + 1
3003          Reactn.Cells(i, 5) = -1
3004      Next i
3005  'Initialize the Parameters
3006      For i = 2 To Parameter.Cells(11, 2) + 1
3007          Inpt.Cells(i, 11) = 0
3008          Inpt.Cells(i, 14) = 0
3009      Next i
3010      Inpt.Cells(2, 11) = 1
3011      For i = 2 To Parameter.Cells(10, 2) + 1
3012          Inpt.Cells(i, 13) = 0
3013      Next i
3014      Misci.Cells(2, 6) = 0
3015      Miscr.Cells(2, 21) = Miscr.Cells(2, 9) / 4184
3016      Miscr.Cells(2, 11) = 0
3017  'Set Problems Types to False
3018      Miscl.Cells(2, 17) = "FALSE"
```

```
3019      Miscl.Cells(2, 7) = "FALSE"
3020      Miscl.Cells(2, 16) = "FALSE"
3021      Miscl.Cells(2, 13) = "FALSE"
3022      Miscl.Cells(2, 4) = "FALSE"
3023      Miscl.Cells(2, 19) = "FALSE"
3024      Miscl.Cells(2, 8) = "FALSE"
3025  'Initialize Indexes
3026      Indx.Cells(2, 12) = 0
3027      Indx.Cells(2, 19) = 1
3028  'Initialize Fuel to Oxidant Ratios
3029      For i = 2 To Indx.Cells(2, 9) + 1
3030          Inpt.Cells(i, 10).ClearContents
3031      Next i
3032      Indx.Cells(2, 9) = 0
3033  End Sub
3034  'This resets the Problem Input Page when it is active
3035  Sub Prob_Input_Refresh()
3036      Dim rowsProdInt As Long
3037      Dim rowsReacInt As Long
3038      rowsReacInt = reac_list.UsedRange.rows.Count
3039      rowsProdInt = prod_list.UsedRange.rows.Count
3040      reac std = True
3041      prod_std = True
3042      ProbInput.ListBox1.List = reac_list.Range("A2:A" & rowsReacInt).Value
3043      ProbInput.ListBox3.List = prod list.Range("A2:A" & rowsProdInt).Value
3044      ProbInput.TextBox4 = ""
3045      ProbInput.TextBox5 = ""
3046      ProbInput.ButtonEditReac.Enabled = False
3047      Call Check_Input_Status
3048      If ProbInput.ComboBox1.ListCount = 0 Then
3049          With ProbInput
3050              .ComboBox1.AddItem "Temperature and Pressure", 0
3051              .ComboBox1.AddItem "Enthalpy and Pressure", 1
3052              .ComboBox1.AddItem "Entropy and Pressure", 2
3053              .ComboBox1.AddItem "Temperature and Volume", 3
3054              .ComboBox1.AddItem "Internal-Energy and Volume", 4
3055              .ComboBox1.AddItem "Entropy and Volume", 5
3056              .ComboBox1.ListIndex = 0
3057          End With
3058      End If
3059      If ProbInput.ListBox8.ListCount = 0 Then
3060          With ProbInput.ListBox8
3061              .AddItem "Pressure", 0
3062              .AddItem "Temperature", 1
3063              .AddItem "Density", 2
3064              .AddItem "Enthalpy", 3
3065              .AddItem "Internal Energy", 4
3066              .AddItem "Gibbs Energy", 5
3067              .AddItem "Entropy", 6
3068              .AddItem "Molecular Weight (1/n)", 7
3069              .AddItem "Molecular Weight", 8
3070              .AddItem "Specific Heat", 9
3071              .AddItem "Gamma(s)", 10
3072              .AddItem "Sonic Velocity", 11
3073              .AddItem "Product Mole/Mass Fractions", 12
3074              .AddItem "Percent Fuel by Weight", 13
3075              .AddItem "Fuel-to-Oxidant Weight Ratios", 14
3076              .AddItem "Oxidant-to-Fuel Weight Ratios", 15
3077              .AddItem "Equivalence Ratios", 16
3078              .AddItem "Chemical Equivalence", 17
3079          End With
3080      End If
3081      Call GetAssignedReac
3082      Call GetProd
3083  End Sub
3084  'This will create the list of Available Reactants using a filter criteria
3085  Sub Create_Avail_Reac(fcheck As String)
3086      Dim rowsReac, rowsInt As Long
3087      rowsReac = reac.UsedRange.rows.Count
3088      rowsInt = reac int.UsedRange.rows.Count
3089      If rowsInt > 1 Then reac_int.Range("A2", "B" & rowsInt).ClearContents
```

```
3090      For i = 2 To rowsReac
3091          If Left(reac.Cells(i, 1), 1) = "*" Then
3092              If Left(reac.Cells(i, 1), Len(fcheck) + 1) = "*" & fcheck Then
3093                  reac int.Cells(i, 1) = Right(reac.Cells(i, 1), Len(reac.Cells(i, 1)) - 1)
3094                  reac_int.Cells(i, 2) = i
3095              End If
3096          Else
3097              If Left(reac.Cells(i, 1), Len(fcheck)) = fcheck Then
3098                  reac_int.Cells(i, 1) = reac.Cells(i, 1)
3099                  reac int.Cells(i, 2) = i
3100              End If
3101          End If
3102      Next i
3103      rowsInt = reac int.UsedRange.rows.Count
3104      reac_int.Columns("A:B").Sort key1:=reac_int.Cells(1, 1), Header:=xlYes
3105  End Sub
3106  'This will call the Available Reactants with a filter of the Reactants Search box
3107  Sub Filter_Avail_Reac()
3108      Dim filter_text As String
3109      Dim rowsReacInt As Long
3110      filter_text = ProbInput.TextBox4
3111      If filter text = "" Then
3112          reac_std = True
3113          rowsReacInt = reac_list.UsedRange.rows.Count
3114          ProbInput.ListBox1.List = reac list.Range("A2:A" & rowsReacInt).Value
3115      Else
3116          reac std = False
3117          Call Create_Avail_Reac(filter_text)
3118          rowsReacInt = reac_int.UsedRange.rows.Count
3119          If rowsReacInt > 1 Then
3120              If rowsReacInt = 2 Then
3121                  ProbInput.ListBox1.Clear
3122                  ProbInput.ListBox1.AddItem reac int.Cells(2, 1).Value
3123              Else
3124                  ProbInput.ListBox1.Clear
3125                  ProbInput.ListBox1.List = reac_int.Range("A2:A" & rowsReacInt).Value
3126              End If
3127          Else
3128              ProbInput.ListBox1.Clear
3129          End If
3130      End If
3131  End Sub
3132  'This will create the list of Available Products using a filter criteria
3133  Sub Create_Avail_Prod(fcheck As String)
3134      Dim rowsProd, rowsInt As Long
3135      rowsProd = prod.UsedRange.rows.Count
3136      rowsInt = prod_int.UsedRange.rows.Count
3137      If rowsInt > 1 Then prod_int.Range("A2", "A" & rowsInt).ClearContents
3138      For i = 2 To rowsProd
3139          If Left(prod.Cells(i, 1), 1) = "*" Then
3140              If Left(prod.Cells(i, 1), Len(fcheck) + 1) = "*" & fcheck Then
3141              If Right(prod.Cells(i, 1), Len(prod.Cells(i, 1)) - 1) <> prod int.Cells(i - 1,
1) Then prod_int.Cells(i, 1) = Right(prod.Cells(i, 1), Len(prod.Cells(i, 1)) - 1)
3142              End If
3143          Else
3144              If Left(prod.Cells(i, 1), Len(fcheck)) = fcheck Then
3145                  If prod.Cells(i, 1) <> prod_int.Cells(i - 1, 1) Then prod_int.Cells(i, 1) =
prod.Cells(i, 1)
3146              End If
3147          End If
3148      Next i
3149      rowsInt = prod int.UsedRange.rows.Count
3150      prod_int.Columns("A:A").Sort key1:=prod_int.Cells(1, 1), Header:=xlYes
3151  End Sub
3152  'This will call the Available Products with a filter of the Products Search box
3153  Sub Filter_Avail_Prod()
3154      Dim filter_text As String
3155      Dim rowsProdInt As Long
3156      filter_text = ProbInput.TextBox5
3157      If filter text = "" Then
3158          prod_std = True
```

```
3159          rowsProdInt = prod_list.UsedRange.rows.Count
3160          ProbInput.ListBox3.List = prod list.Range("A2:A" & rowsProdInt).Value
3161      Else
3162          prod std = False
3163          Call Create_Avail_Prod(filter_text)
3164          rowsProdInt = prod_int.UsedRange.rows.Count
3165          If rowsProdInt > 1 Then
3166              If rowsProdInt = 2 Then
3167                  ProbInput.ListBox3.Clear
3168                  ProbInput.ListBox3.AddItem prod int.Cells(2, 1).Value
3169              Else
3170                  ProbInput.ListBox3.Clear
3171                  ProbInput.ListBox3.List = prod_int.Range("A2:A" & rowsProdInt).Value
3172              End If
3173          Else
3174              ProbInput.ListBox3.Clear
3175          End If
3176      End If
3177  End Sub
3178  'This will filter the Available Products with a custom list
3179  Sub Custom_Filter_Avail_Prod()
3180      Dim rowsProdInt, rowsProd As Long
3181      rowsProdInt = prod_filter.UsedRange.rows.Count
3182      rowsProd = prod_list.UsedRange.rows.Count
3183      If rowsProdInt > 1 And prod filter.Cells(2, 1) <> "" Then
3184          For i = 2 To rowsProdInt
3185              For j = 2 To rowsProd
3186                  If prod_filter.Cells(i, 1) = prod_list.Cells(j, 1) Or prod_filter.Cells(i,
1) = "*" & prod_list.Cells(j, 1) Then GoTo 50
3187              Next j
3188              MsgBox ("Product " & prod_filter.Cells(i, 1) & " is not a library product.
Remove from filter and restart.")
3189              GoTo 100
3190  50:     Next i
3191          ProbInput.ListBox3.List = prod_filter.Range("A2:A" & rowsProdInt).Value
3192      Else
3193          MsgBox ("No Products in Custom Filter List")
3194  100: End If
3195  End Sub
3196
3197  'This Setups up the Custom Reactant User Form
3198  Sub Custom_Reac_Setup()
3199      With CustomReac.ComboBox1
3200          .AddItem "j/mol", 0
3201          .AddItem "kj/mol", 1
3202          .AddItem "c/mol", 2
3203          .AddItem "kc/mol", 3
3204          .ListIndex = 0
3205      End With
3206  End Sub
3207  'This makes sure the previous screen is cleared for new custom reactant
3208  Sub Custom Reac New()
3209      For i = 1 To 26
3210          CustomReac.Controls.Item("TextBox" & i) = ""
3211      Next i
3212      CustomReac.ComboBox1.ListIndex = 0
3213  End Sub
3214
3215
3216  'This Setups up the the screen to edit a previously created Custom Reactant
3217  Sub Edit_Reac_Setup()
3218      Dim row As Integer
3219      row = ProbInput.ListBox2.ListIndex + 2
3220      CustomReac.TextBox1 = Cdata.Cells(row, 11)
3221      If CustomReac.TextBox2 <> "" Then
3222          If CustomReac.ComboBox1.ListIndex = 0 Then
3223              CustomReac.TextBox2 = Reactn.Cells(row, 4) / 1000 * Miscr.Cells(2, 9)
3224          ElseIf CustomReac.ComboBox1.ListIndex = 1 Then
3225              CustomReac.TextBox2 = Reactn.Cells(row, 4) / 1000 * Miscr.Cells(2, 9) / 1000
3226          ElseIf CustomReac.ComboBox1.ListIndex = 2 Then
3227              CustomReac.TextBox2 = Reactn.Cells(row, 4) / 1000 * Miscr.Cells(2, 9) / 4.184
```

```
3228          ElseIf CustomReac.ComboBox1.ListIndex = 3 Then
3229              CustomReac.TextBox2 = Reactn.Cells(row, 4) / 1000 * Miscr.Cells(2, 9) / 4.184 /
1000
3230          End If
3231      End If
3232      For i = 3 To 14
3233          CustomReac.Controls.Item("TextBox" & i) = Cdata.Cells(row, i + 10)
3234      Next i
3235      For i = 15 To 26
3236          If CDbl(Reactn.Cells(row, i - 7)) <> 1 Then CustomReac.Controls.Item("TextBox" & i)
= Reactn.Cells(row, i - 7)
3237      Next i
3238
3239  End Sub
3240  'This checks if the Selected Reactant is a Custom Reactant
3241  Sub Edit Reac Check()
3242      Dim row As Integer
3243      row = ProbInput.ListBox2.ListIndex + 2
3244      If CStr(Cdata.Cells(row, 7)) = "lib" Then ProbInput.ButtonEditReac.Enabled = False
3245      If CStr(Cdata.Cells(row, 7)) = "Custom" Then ProbInput.ButtonEditReac.Enabled = True
3246  End Sub
3247
3248  'This will save the edits of the Custom Reactant
3249  Sub Edit_Reac_Save()
3250      Dim row As Integer
3251      Dim sym As Boolean
3252      Dim ifrmla, reacRow As Long
3253      ifrmla = 0
3254      CustOK = True
3255      row = ProbInput.ListBox2.ListIndex + 2
3256      reacRow = reac_list.UsedRange.rows.Count
3257          If CustomReac.TextBox1 <> "" Then
3258              For i = 2 To reacRow
3259                  If CustomReac.TextBox1 = reac_list.Cells(i, 1) Then
3260                      MsgBox ("Custom Reactant cannot have the same name as a Library
Reactant.")
3261                      CustOK = False
3262                      GoTo 50
3263                  End If
3264              Next i
3265              If CustomReac.TextBox3 <> "" Then
3266                  If IsNumeric(CustomReac.TextBox2) = False And CustomReac.TextBox2 <> ""
Then
3267                      MsgBox ("Reactant Enthalpy must be numeric")
3268                      CustOK = False
3269                      GoTo 50
3270                  End If
3271                  For i = 15 To 26
3272                      If IsNumeric(CustomReac.Controls.Item("TextBox" & i)) = False And
CustomReac.Controls.Item("TextBox" & i) <> "" Then
3273                          MsgBox ("Exploded Formula subscripts must be numeric")
3274                          CustOK = False
3275                          GoTo 50
3276                      End If
3277                  Next i
3278                  For i = 3 To 14
3279                      If CustomReac.Controls.Item("TextBox" & i) <> "" Then
3280                          sym = False
3281                          For j = 2 To 101
3282                              If CStr(CustomReac.Controls.Item("TextBox" & i)) =
CStr(Cdata.Cells(j, 2)) Then sym = True
3283                          Next j
3284                          If sym = False Then
3285                              MsgBox ("Exploded Formula must use elements from the periodic
table")
3286                              CustOK = False
3287                              GoTo 50
3288                          End If
3289                      End If
3290                  Next i
3291                  Cdata.Cells(row, 11) = CustomReac.TextBox1
```

```
3292                 Cdata.Cells(row, 26).ClearContents
3293                 Cdata.Cells(row, 7) = "Custom"
3294                 If CustomReac.TextBox2 <> "" Then
3295                     If CustomReac.ComboBox1.ListIndex = 0 Then
3296                         Reactn.Cells(row, 4) = CDbl(CustomReac.TextBox2) * 1000 /
Miscr.Cells(2, 9)
3297                     ElseIf CustomReac.ComboBox1.ListIndex = 1 Then
3298                         Reactn.Cells(row, 4) = CDbl(CustomReac.TextBox2) * 1000 /
Miscr.Cells(2, 9) * 1000
3299                     ElseIf CustomReac.ComboBox1.ListIndex = 2 Then
3300                         Reactn.Cells(row, 4) = CDbl(CustomReac.TextBox2) * 1000 /
Miscr.Cells(2, 9) * 4.184
3301                     ElseIf CustomReac.ComboBox1.ListIndex = 3 Then
3302                         Reactn.Cells(row, 4) = CDbl(CustomReac.TextBox2) * 1000 /
Miscr.Cells(2, 9) * 4.184 * 1000
3303                     End If
3304                 End If
3305                 For i = 3 To 14
3306                     If CustomReac.Controls.Item("TextBox" & i) <> "" Then
3307                         Cdata.Cells(row, i + 10) = UCase(CustomReac.Controls.Item("TextBox"
& i))
3308                         Reactn.Cells(row, i + 5) = 1
3309                         ifrmla = ifrmla + 1
3310                     End If
3311                 Next i
3312                     Indx.Cells(row, 22) = ifrmla
3313                 For i = 15 To 26
3314                     If CustomReac.Controls.Item("TextBox" & i) <> "" Then Reactn.Cells(row,
i - 7) = CDbl(CustomReac.Controls.Item("TextBox" & i))
3315                 Next i
3316             Else
3317                 MsgBox ("Must enter an Exploded Chemical Formula")
3318                 CustOK = False
3319             End If
3320         Else
3321             MsgBox ("Must supply a Reactant Name")
3322             CustOK = False
3323  50:    End If
3324  End Sub
3325
3326  'This will save the details of the Custom Reactant
3327  Sub Add_Custom_Reac()
3328      Dim ifrmla, reacRow As Long
3329      Dim sym As Boolean
3330      ifrmla = 0
3331      CustOK = True
3332      reacRow = reac_list.UsedRange.rows.Count
3333      If Reactn.Cells(2, 1) < Parameter.Cells(6, 2) Then
3334          If CustomReac.TextBox1 <> "" Then
3335              For i = 2 To reacRow
3336                  If CustomReac.TextBox1 = reac_list.Cells(i, 1) Then
3337                      MsgBox ("Custom Reactant cannot have the same name as a Library
Reactant.")
3338                      CustOK = False
3339                      GoTo 50
3340                  End If
3341              Next i
3342              If CustomReac.TextBox3 <> "" Then
3343                  If IsNumeric(CustomReac.TextBox2) = False And CustomReac.TextBox2 <> ""
Then
3344                      MsgBox ("Reactant Enthalpy must be numeric")
3345                      CustOK = False
3346                      GoTo 50
3347                  End If
3348                  For i = 15 To 26
3349                      If IsNumeric(CustomReac.Controls.Item("TextBox" & i)) = False And
CustomReac.Controls.Item("TextBox" & i) <> "" Then
3350                          MsgBox ("Exploded Formula subscripts must be numeric")
3351                          CustOK = False
3352                          GoTo 50
3353                      End If
```

```
3354                 Next i
3355                 For i = 3 To 14
3356                     If CustomReac.Controls.Item("TextBox" & i) <> "" Then
3357                         sym = False
3358                         For j = 2 To 101
3359                             If CStr(CustomReac.Controls.Item("TextBox" & i)) =
CStr(Cdata.Cells(j, 2)) Then sym = True
3360                         Next j
3361                         If sym = False Then
3362                             MsgBox ("Exploded Formula must use elements from the periodic
table")
3363                             CustOK = False
3364                             GoTo 50
3365                         End If
3366                     End If
3367                 Next i
3368                 Reactn.Cells(2, 1) = Reactn.Cells(2, 1) + 1
3369                 Cdata.Cells(Reactn.Cells(2, 1) + 1, 11) = CustomReac.TextBox1
3370                 Cdata.Cells(Reactn.Cells(2, 1) + 1, 26).ClearContents
3371                 Cdata.Cells(Reactn.Cells(2, 1) + 1, 7) = "Custom"
3372                 If CustomReac.TextBox2 <> "" Then
3373                     If CustomReac.ComboBox1.ListIndex = 0 Then
3374                         Reactn.Cells(Reactn.Cells(2, 1) + 1, 4) = CDbl(CustomReac.TextBox2)
* 1000 / Miscr.Cells(2, 9)
3375                     ElseIf CustomReac.ComboBox1.ListIndex = 1 Then
3376                         Reactn.Cells(Reactn.Cells(2, 1) + 1, 4) = CDbl(CustomReac.TextBox2)
* 1000 / Miscr.Cells(2, 9) * 1000
3377                     ElseIf CustomReac.ComboBox1.ListIndex = 2 Then
3378                         Reactn.Cells(Reactn.Cells(2, 1) + 1, 4) = CDbl(CustomReac.TextBox2)
* 1000 / Miscr.Cells(2, 9) * 4.184
3379                     ElseIf CustomReac.ComboBox1.ListIndex = 3 Then
3380                         Reactn.Cells(Reactn.Cells(2, 1) + 1, 4) = CDbl(CustomReac.TextBox2)
* 1000 / Miscr.Cells(2, 9) * 4.184 * 1000
3381                     End If
3382                 End If
3383                 For i = 3 To 14
3384                     If CustomReac.Controls.Item("TextBox" & i) <> "" Then
3385                         Cdata.Cells(Reactn.Cells(2, 1) + 1, i + 10) =
UCase(CustomReac.Controls.Item("TextBox" & i))
3386                         Reactn.Cells(Reactn.Cells(2, 1) + 1, i + 5) = 1
3387                         ifrmla = ifrmla + 1
3388                     End If
3389                 Next i
3390                     Indx.Cells(Reactn.Cells(2, 1) + 1, 22) = ifrmla
3391                 For i = 15 To 26
3392                     If CustomReac.Controls.Item("TextBox" & i) <> "" Then
Reactn.Cells(Reactn.Cells(2, 1) + 1, i - 7) = CDbl(CustomReac.Controls.Item("TextBox" & i))
3393                 Next i
3394             Else
3395                 MsgBox ("Must enter an Exploded Chemical Formula")
3396                 CustOK = False
3397             End If
3398         Else
3399             MsgBox ("Must supply a Reactant Name")
3400             CustOK = False
3401         End If
3402     Else
3403         MsgBox ("Maximum Number of Allowable Reactants Reached.")
3404         CustOK = False
3405  50: End If
3406  End Sub
3407
3408  'This checks the progress of the Input for a successful problem input
3409  Sub Check Input Status()
3410      Dim inputOK As Boolean
3411      inputOK = False
3412  'Check If Reactants have been specified
3413      If Reactn.Cells(2, 1) <= 0 Then
3414          ProbInput.StatusLabel = "Insert Reactants"
3415      Else
3416          inputOK = True
```

```
3417      End If
3418  'Check If Reactant Amounts have been specified if no Fuel/Oxidant ratio specified
3419      If inputOK And CInt(Indx.Cells(2, 9)) = 0 Then
3420          For i = 1 To Reactn.Cells(2, 1)
3421              If Reactn.Cells(i + 1, 5) <= 0 Then inputOK = False
3422          Next i
3423          If Not inputOK Then ProbInput.StatusLabel = "Need Reactant Amounts"
3424  'Check If Reactant Amounts have been specified when MOLES = TRUE
3425      ElseIf inputOK And Miscl.Cells(2, 10) Then
3426          For i = 1 To Reactn.Cells(2, 1)
3427              If Reactn.Cells(i + 1, 5) <= 0 Then inputOK = False
3428          Next i
3429          If Not inputOK Then ProbInput.StatusLabel = "Need Reactant Amounts"
3430      End If
3431  'Checks if Reactant Temperatures have been Specified for Enthalpy or Internal Energy
Problems
3432      If inputOK And (ProbInput.ComboBox1.ListIndex = 1 Or ProbInput.ComboBox1.ListIndex = 4)
Then
3433          For i = 1 To Reactn.Cells(2, 1)
3434              If CStr(Cdata.Cells(i + 1, 7)) = "lib" And CDbl(Reactn.Cells(i + 1, 7)) <= 0
Then inputOK = False
3435          Next i
3436          If Not inputOK Then ProbInput.StatusLabel = "Need Temperatures for Library
Reactants"
3437      End If
3438  'Check Fuel/Oxidant Ratios vs. Specified Reactants
3439      If inputOK And CInt(Indx.Cells(2, 9)) <> 0 Then
3440          For j = 1 To Indx.Cells(2, 9)
3441              If CDbl(Inpt.Cells(j + 1, 10)) = 0 Then
3442                  If fuel Then
3443                      For i = 1 To Indx.Cells(2, 7)
3444                          If Cdata.Cells(i + 1, 4) = "OXIDANT" Then
3445                              ProbInput.StatusLabel = "OXIDANT NOT PERMITTED WHEN 100% FUEL"
3446                              inputOK = False
3447                          End If
3448                      Next i
3449                  Else
3450                      For i = 1 To Indx.Cells(2, 7)
3451                          If Cdata.Cells(i + 1, 4) = "FUEL" Then
3452                              ProbInput.StatusLabel = "FUEL NOT PERMITTED WHEN 100% OXIDANT"
3453                              inputOK = False
3454                          End If
3455                      Next i
3456                  End If
3457              Else
3458              inputOK = True
3459                  For i = 1 To Indx.Cells(2, 7)
3460                      If Cdata.Cells(i + 1, 4) = "FUEL" Then GoTo 10
3461                  Next i
3462                  inputOK = False
3463  10:             For i = 1 To Indx.Cells(2, 7)
3464                      If Cdata.Cells(i + 1, 4) = "OXIDANT" Then GoTo 20
3465                  Next i
3466                  inputOK = False
3467  20:             If Not inputOK Then ProbInput.StatusLabel = "MUST SPECIFY A FUEL AND
OXIDANT"
3468              End If
3469          Next j
3470      End If
3471      If inputOK Then
3472          ProbInput.StatusLabel = "READY"
3473          ProbInput.ButtonNext.Enabled = True
3474      Else
3475          ProbInput.ButtonNext.Enabled = False
3476      End If
3477  End Sub
3478
3479  'This starts the setup of the Problem
3480  Sub ProblemStartUpdate()
3481  'Enter in the Case Name
3482      Cdata.Cells(2, 6) = ProbInput.TextBox3
```

```
3483  'Enter the Trace Number
3484      Miscr.Cells(2, 14) = ProbInput.TextBox1 * 10 ^ (-ProbInput.TextBox2)
3485  'Set which Interations to Debug
3486      If ProbInput.CheckBox1.Value = True Then
3487          For i = 0 To ProbInput.ListBox7.ListCount - 1
3488              If ProbInput.ListBox7.Selected(i) = True Then Miscl.Cells(i + 2, 2) = "TRUE"
3489          Next i
3490      End If
3491      If ProbInput.ListBox8.Selected(0) = True Then
3492          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3493          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "p"
3494      End If
3495      If ProbInput.ListBox8.Selected(1) = True Then
3496          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3497          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "t"
3498      End If
3499      If ProbInput.ListBox8.Selected(2) = True Then
3500          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3501          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "rho"
3502      End If
3503      If ProbInput.ListBox8.Selected(3) = True Then
3504          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3505          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "h"
3506      End If
3507      If ProbInput.ListBox8.Selected(4) = True Then
3508          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3509          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "u"
3510      End If
3511      If ProbInput.ListBox8.Selected(5) = True Then
3512          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3513          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "g"
3514      End If
3515      If ProbInput.ListBox8.Selected(6) = True Then
3516          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3517          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "s"
3518      End If
3519      If ProbInput.ListBox8.Selected(7) = True Then
3520          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3521          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "m"
3522      End If
3523      If ProbInput.ListBox8.Selected(8) = True Then
3524          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3525          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "mw"
3526      End If
3527      If ProbInput.ListBox8.Selected(9) = True Then
3528          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3529          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "cp"
3530      End If
3531      If ProbInput.ListBox8.Selected(10) = True Then
3532          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3533          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "gam"
3534      End If
3535      If ProbInput.ListBox8.Selected(11) = True Then
3536          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3537          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "son"
3538      End If
3539      If ProbInput.ListBox8.Selected(12) = True Then pfrac = True
3540      If ProbInput.ListBox8.Selected(13) = True Then
3541          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3542          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "%f"
3543      End If
3544      If ProbInput.ListBox8.Selected(14) = True Then
3545          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3546          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "f/a"
3547      End If
3548      If ProbInput.ListBox8.Selected(15) = True Then
3549          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3550          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "o/f"
3551      End If
3552      If ProbInput.ListBox8.Selected(16) = True Then
3553          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
```

```
3554          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "phi"
3555      End If
3556      If ProbInput.ListBox8.Selected(17) = True Then
3557          Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
3558          Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = "r"
3559      End If
3560  End Sub
3561  'This moves the program to the correct screen for problem type
3562  Sub Prob_Input_Type_Screen()
3563      Dim response As Integer
3564      Call ProblemStartUpdate
3565      ProbInput.Hide
3566      If ProbInput.ComboBox1.ListIndex = 0 Then 'Temperature and Pressure
3567          Call Form Position(ProbTP)
3568          ProbTP.Show
3569      ElseIf ProbInput.ComboBox1.ListIndex = 1 Then 'Enthaply and Pressure
3570          Call Form_Position(ProbHP)
3571          ProbHP.Show
3572      ElseIf ProbInput.ComboBox1.ListIndex = 2 Then 'Entropy and Pressure
3573          Call Form_Position(ProbSP)
3574          ProbSP.Show
3575      ElseIf ProbInput.ComboBox1.ListIndex = 3 Then 'Temperature and Volume
3576          Call Form_Position(ProbTV)
3577          ProbTV.Show
3578      ElseIf ProbInput.ComboBox1.ListIndex = 4 Then 'Internal-Energy and Volume
3579          Call Form_Position(ProbUV)
3580          ProbUV.Show
3581      ElseIf ProbInput.ComboBox1.ListIndex = 5 Then 'Entropy and Volume
3582          Call Form_Position(ProbSV)
3583          ProbSV.Show
3584      End If
3585  End Sub
3586  'This removes set properties if the Back Button was used while in the ProbTP, etc. windows
3587  Sub Prob_Input_Reset()
3588  'Reset the Plot Parameters
3589      If CInt(Indx.Cells(2, 8)) > 0 Then
3590          For i = 1 To Indx.Cells(2, 8)
3591              Cdata.Cells(i + 1, 9).ClearContents
3592          Next i
3593          Indx.Cells(2, 8) = 0
3594      End If
3595  'Reset the Temperatures
3596      If CInt(Indx.Cells(2, 19)) > 0 Then
3597          For i = 1 To Indx.Cells(2, 19)
3598              Inpt.Cells(i + 1, 13) = 0
3599          Next i
3600          Indx.Cells(2, 19) = 1
3601      End If
3602  'Reset the Pressures, Specific Volume, Or Density
3603      If CInt(Indx.Cells(2, 12)) > 0 Then
3604          For i = 1 To Indx.Cells(2, 12)
3605              Inpt.Cells(i + 1, 11) = 0
3606              Inpt.Cells(i + 1, 14) = 0
3607          Next i
3608          Indx.Cells(2, 12) = 0
3609      End If
3610  'Reset the Enthaply and Internal Energy
3611      Miscr.Cells(5, 4).ClearContents
3612      Miscr.Cells(7, 4).ClearContents
3613  'Reset the Entropy
3614      Miscr.Cells(2, 11).ClearContents
3615  End Sub
3616
3617  '*********************************************************************
3618  'ReacProperties
3619  '*********************************************************************
3620  'This updates if the Reactant Amounts are specified in Moles
3621  Sub Reac Input Moles Status()
3622      If ReacProperties.ComboBox1.ListIndex = 0 Then
3623          Miscl.Cells(2, 10) = "FALSE"
3624      ElseIf ReacProperties.ComboBox1.ListIndex = 1 Then
```

```
3625          Miscl.Cells(2, 10) = "TRUE"
3626      End If
3627  End Sub
3628  'This Removes the TextBoxes for the Reactant Properties Input
3629  Sub Remove_Previous()
3630      For i = 1 To Reactn.Cells(2, 1)
3631          ReacProperties.Controls.Remove ("React" & i)
3632          ReacProperties.Controls.Remove ("Perct" & i)
3633          ReacProperties.Controls.Remove ("Temp" & i)
3634          ReacProperties.Controls.Remove ("Dens" & i)
3635          ReacProperties.Controls.Remove ("Enth" & i)
3636          ReacProperties.Controls.Remove ("Fox" & i)
3637      Next i
3638  End Sub
3639  'This saves the Reactant Properties entered by the user
3640  Sub Update Properties()
3641      Dim response As Integer
3642      Dim reacPerct As Object
3643      Dim reacTemp As Object
3644      Dim reacDens As Object
3645      Dim reacEnth As Object
3646      Dim reacFox As Object
3647      Dim t1, tt, dift As Double
3648      ReacPropOk = True
3649      fuel = False
3650      For i = 1 To Reactn.Cells(2, 1)
3651          Set reacPerct = ReacProperties.Controls.Item("Perct" & i)
3652          Set reacTemp = ReacProperties.Controls.Item("Temp" & i)
3653          Set reacDens = ReacProperties.Controls.Item("Dens" & i)
3654          Set reacEnth = ReacProperties.Controls.Item("Enth" & i)
3655          If IsNumeric(reacPerct.Value) = False And reacPerct.Value <> "" Then ReacPropOk =
False
3656          If IsNumeric(reacTemp.Value) = False And reacTemp.Value <> "" Then ReacPropOk =
False
3657          If IsNumeric(reacDens.Value) = False And reacDens.Value <> "" Then ReacPropOk =
False
3658          If IsNumeric(reacEnth.Value) = False And reacEnth.Value <> "" Then ReacPropOk =
False
3659      Next i
3660      If ReacPropOk Then
3661      For i = 1 To Reactn.Cells(2, 1)
3662              t1 = reac.Cells(Cdata.Cells(i + 1, 26) + 1, 16)
3663              t2 = reac.Cells(Cdata.Cells(i + 1, 26) + 1, 17)
3664              Set reacPerct = ReacProperties.Controls.Item("Perct" & i)
3665              Set reacTemp = ReacProperties.Controls.Item("Temp" & i)
3666              Set reacDens = ReacProperties.Controls.Item("Dens" & i)
3667              Set reacEnth = ReacProperties.Controls.Item("Enth" & i)
3668              Set reacFox = ReacProperties.Controls.Item("Fox" & i)
3669              If reacTemp.Value <> "" Then
3670                  If ReacProperties.ComboBox2.ListIndex = 0 Then
3671                      tt = reacTemp.Value
3672                  ElseIf ReacProperties.ComboBox2.ListIndex = 1 Then
3673                      tt = reacTemp.Value / 1.8
3674                  ElseIf ReacProperties.ComboBox2.ListIndex = 2 Then
3675                      tt = reacTemp.Value + 273.15
3676                  ElseIf ReacProperties.ComboBox2.ListIndex = 3 Then
3677                      tt = (reacTemp.Value - 32) / 1.8 + 273.15
3678                  End If
3679                  If Cdata.Cells(i + 1, 26) <> "" Then
3680                      If reac.Cells(Cdata.Cells(i + 1, 26) + 1, 3) = 0 Then
3681                          dift = Abs(tt - t1)
3682                          If dift > 1 Then
3683                              If dift > 10 Then
3684                                  response = MsgBox("Reactant " & reac.Cells(Cdata.Cells(i +
1, 26) + 1, 1) & " has been defined for the temperature " & t1 & " only. " & vbNewLine & "Your
temperature assignment " & tt & " is more than 10 K from this value.", vbCritical)
3685                                  ReacPropOk = False
3686                                  GoTo 200
3687                              Else
3688                                  response = MsgBox("Reactant " & reac.Cells(Cdata.Cells(i +
1, 26) + 1, 1) & " has been defined for the temperature " & t1 & " only. " & vbNewLine & "Your
```

```
temperature assignment " & tt & " is not equal but less than 10 K from this value.",
vbExclamation)
3689                                End If
3690                            End If
3691                        Else
3692                            If CDbl(reac.Cells(Cdata.Cells(i + 1, 26) + 1, 15)) <= 0 And
(CDbl(tt) < CDbl(t1) Or CDbl(tt) > CDbl(t2)) Then
3693                                response = MsgBox("Reactant " & reac.Cells(Cdata.Cells(i + 1,
26) + 1, 1) & " was specificed as " & tt & " which is out of the Range (" & t1 & " - " & t2 & " K
) for this reactant.", vbCritical)
3694                                ReacPropOk = False
3695                                GoTo 200
3696                            End If
3697                        End If
3698                    End If
3699                End If
3700                With Reactn
3701                    If reacPerct.Value <> "" Then
3702                        .Cells(i + 1, 5) = reacPerct.Value
3703                    Else
3704                        .Cells(i + 1, 5) = -1
3705                    End If
3706                    If reacTemp.Value <> "" Then
3707                        .Cells(i + 1, 7) = tt
3708                    Else
3709                        .Cells(i + 1, 7) = 0
3710                    End If
3711                    If reacDens.Value <> "" Then
3712                        If ReacProperties.ComboBox4.ListIndex = 0 Then
3713                            .Cells(i + 1, 3) = reacDens.Value
3714                        ElseIf ReacProperties.ComboBox4.ListIndex = 1 Then
3715                            .Cells(i + 1, 3) = reacDens.Value / 1000
3716                        End If
3717                    Else
3718                        .Cells(i + 1, 3) = ""
3719                    End If
3720                    If reacEnth.Value <> "" Then
3721                        Cdata.Cells(i + 1, 7) = "Custom"
3722                        If ReacProperties.ComboBox5.ListIndex = 0 Then
3723                            .Cells(i + 1, 4) = reacEnth.Value * 1000 / Miscr.Cells(2, 9)
3724                        ElseIf ReacProperties.ComboBox5.ListIndex = 1 Then
3725                            .Cells(i + 1, 4) = reacEnth.Value * 1000 / Miscr.Cells(2, 9) * 1000
3726                        ElseIf ReacProperties.ComboBox5.ListIndex = 2 Then
3727                            .Cells(i + 1, 4) = reacEnth.Value * 1000 / Miscr.Cells(2, 9) *
4.184
3728                        ElseIf ReacProperties.ComboBox5.ListIndex = 3 Then
3729                            .Cells(i + 1, 4) = reacEnth.Value * 1000 / Miscr.Cells(2, 9) *
4.184 * 1000
3730                        End If
3731                    End If
3732                End With
3733                If reacFox.ListIndex = 1 Then
3734                    Cdata.Cells(i + 1, 4) = "OXIDANT"
3735                Else
3736                    Cdata.Cells(i + 1, 4) = "FUEL"
3737                    fuel = True
3738                End If
3739            Next i
3740        Else
3741            MsgBox ("Reactant Properties must be numeric")
3742        End If
3743  200: End Sub
3744  'This sets up the Reactant Properties Screen
3745  Sub Reac Properties Setup()
3746        Dim reacLabel As Object
3747        Dim reacPerct As Object
3748        Dim reacTemp As Object
3749        Dim reacDens As Object
3750        Dim reacFox As Object
3751        Dim reacEnth As Object
3752  'Set up the Relative Amount ComboBox
```

```
3753      If ReacProperties.ComboBox1.ListCount = 0 Then
3754          ReacProperties.ComboBox1.AddItem "Weight %", 0
3755          ReacProperties.ComboBox1.AddItem "Moles", 1
3756      End If
3757      If Miscl.Cells(2, 10) Then
3758          ReacProperties.ComboBox1.ListIndex = 1
3759      Else
3760          ReacProperties.ComboBox1.ListIndex = 0
3761      End If
3762  'Set up the Temperature Unit ComboBox
3763      If ReacProperties.ComboBox2.ListCount = 0 Then
3764          ReacProperties.ComboBox2.AddItem "K", 0
3765          ReacProperties.ComboBox2.AddItem "R", 1
3766          ReacProperties.ComboBox2.AddItem "C", 2
3767          ReacProperties.ComboBox2.AddItem "F", 3
3768          ReacProperties.ComboBox2.ListIndex = 0
3769      End If
3770  'Set up the Density Unit ComboBox
3771      If ReacProperties.ComboBox4.ListCount = 0 Then
3772          ReacProperties.ComboBox4.AddItem "g/cm^3", 0
3773          ReacProperties.ComboBox4.AddItem "kg/m^3", 1
3774          ReacProperties.ComboBox4.ListIndex = 0
3775      End If
3776  'Set up the Enthalpy Unit ComboBox
3777      If ReacProperties.ComboBox5.ListCount = 0 Then
3778          ReacProperties.ComboBox5.AddItem "j/mol", 0
3779          ReacProperties.ComboBox5.AddItem "kj/mol", 1
3780          ReacProperties.ComboBox5.AddItem "c/mol", 2
3781          ReacProperties.ComboBox5.AddItem "kc/mol", 3
3782          ReacProperties.ComboBox5.ListIndex = 0
3783      End If
3784  'Add the TextBoxes for Input
3785      If Reactn.Cells(2, 1) <> 0 Then
3786          For i = 1 To Reactn.Cells(2, 1)
3787              Set reacLabel = ReacProperties.Controls.Add("Forms.Label.1", "React" & i, True)
3788              With reacLabel
3789                  .Caption = Cdata.Cells(i + 1, 11)
3790                  .Height = 18
3791                  .Width = 70
3792                  .Left = 80
3793                  .Top = 16 + 18 * i
3794              End With
3795              Set reacPerct = ReacProperties.Controls.Add("Forms.TextBox.1", "Perct" & i,
True)
3796              With reacPerct
3797                  .Height = 18
3798                  .Width = 70
3799                  .Left = 160
3800                  .Top = 12 + 18 * i
3801                  .TextAlign = 3
3802              End With
3803              If Reactn.Cells(i + 1, 5) = -1 Then
3804                  reacPerct.Value = ""
3805              Else
3806                  reacPerct.Value = Reactn.Cells(i + 1, 5)
3807              End If
3808              Set reacTemp = ReacProperties.Controls.Add("Forms.TextBox.1", "Temp" & i, True)
3809              With reacTemp
3810                  If Reactn.Cells(i + 1, 7) <> 0 Then
3811                      If ReacProperties.ComboBox2.ListIndex = 0 Then
3812                          .Value = Reactn.Cells(i + 1, 7)
3813                      ElseIf ReacProperties.ComboBox2.ListIndex = 1 Then
3814                          .Value = Reactn.Cells(i + 1, 7) * 1.8
3815                      ElseIf ReacProperties.ComboBox2.ListIndex = 2 Then
3816                          .Value = Reactn.Cells(i + 1, 7) - 273.15
3817                      ElseIf ReacProperties.ComboBox2.ListIndex = 3 Then
3818                          .Value = (Reactn.Cells(i + 1, 7) + 32) * 1.8 - 273.15
3819                      End If
3820                  End If
3821                  .Height = 18
3822                  .Width = 70
```

```
3823                .Left = 250
3824                .Top = 12 + 18 * i
3825                .TextAlign = 3
3826            End With
3827            Set reacDens = ReacProperties.Controls.Add("Forms.TextBox.1", "Dens" & i, True)
3828            With reacDens
3829                If Reactn.Cells(i + 1, 3) <> "" Then
3830                    If ReacProperties.ComboBox4.ListIndex = 0 Then
3831                        .Value = Reactn.Cells(i + 1, 3)
3832                    ElseIf ReacProperties.ComboBox4.ListIndex = 1 Then
3833                        .Value = Reactn.Cells(i + 1, 3) * 1000
3834                    End If
3835                End If
3836                .Height = 18
3837                .Width = 70
3838                .Left = 340
3839                .Top = 12 + 18 * i
3840                .TextAlign = 3
3841            End With
3842            Set reacEnth = ReacProperties.Controls.Add("Forms.TextBox.1", "Enth" & i, True)
3843            With reacEnth
3844                If CDbl(Reactn.Cells(i + 1, 4)) <> 0 Or Cdata.Cells(i + 1, 7) = "Custom"
Then
3845                    .Enabled = True
3846                    If ReacProperties.ComboBox5.ListIndex = 0 Then
3847                        .Value = Reactn.Cells(i + 1, 4) / 1000 * Miscr.Cells(2, 9)
3848                    ElseIf ReacProperties.ComboBox5.ListIndex = 1 Then
3849                        .Value = Reactn.Cells(i + 1, 4) / 1000 * Miscr.Cells(2, 9) / 1000
3850                    ElseIf ReacProperties.ComboBox5.ListIndex = 2 Then
3851                        .Value = Reactn.Cells(i + 1, 4) / 1000 * Miscr.Cells(2, 9) / 4.184
3852                    ElseIf ReacProperties.ComboBox5.ListIndex = 3 Then
3853                        .Value = Reactn.Cells(i + 1, 4) / 1000 * Miscr.Cells(2, 9) / 4.184
/ 1000
3854                    End If
3855                Else
3856                    .Value = ""
3857                    .Enabled = False
3858                End If
3859                .Height = 18
3860                .Width = 70
3861                .Left = 430
3862                .Top = 12 + 18 * i
3863                .TextAlign = 3
3864            End With
3865            Set reacFox = ReacProperties.Controls.Add("Forms.ComboBox.1", "Fox" & i, True)
3866            With reacFox
3867                .Height = 18
3868                .Width = 50
3869                .Left = 12
3870                .Top = 12 + 18 * i
3871                .AddItem "Fuel", 0
3872                .AddItem "Oxid", 1
3873                If Cdata.Cells(i + 1, 4) = "FUEL" Then
3874                    .ListIndex = 0
3875                ElseIf Cdata.Cells(i + 1, 4) = "OXIDANT" Then
3876                    .ListIndex = 1
3877                Else
3878                    .ListIndex = 0
3879                End If
3880            End With
3881        Next i
3882    End If
3883    With ReacProperties
3884        .ComboBox1.Top = 35 + 18 * Reactn.Cells(2, 1)
3885        .ComboBox2.Top = 35 + 18 * Reactn.Cells(2, 1)
3886        .ComboBox4.Top = 35 + 18 * Reactn.Cells(2, 1)
3887        .ComboBox5.Top = 35 + 18 * Reactn.Cells(2, 1)
3888        .ButtonSave.Top = 64 + 18 * Reactn.Cells(2, 1)
3889        .ButtonSaveClose.Top = 64 + 18 * Reactn.Cells(2, 1)
3890        .ButtonBack.Top = 64 + 18 * Reactn.Cells(2, 1)
3891        .Height = 124 + 18 * Reactn.Cells(2, 1)
```

```
3892      End With
3893  End Sub
3894
3895  '*********************************************************************
3896  'FuelOxid
3897  '*********************************************************************
3898  'This setups up the ComboBoxes for the FuelOxid Screen
3899  Sub FuelOxid Active()
3900  'Set up the Fuel Type ComboBox
3901      If FuelOxid.ComboBox1.ListCount = 0 Then
3902          FuelOxid.ComboBox1.AddItem "Percent Fuel by Weight", 0
3903          FuelOxid.ComboBox1.AddItem "Fuel-to-Oxidant Weight Ratios", 1
3904          FuelOxid.ComboBox1.AddItem "Oxidant-to-Fuel Weight Ratios", 2
3905          FuelOxid.ComboBox1.AddItem "Equivalence Ratios", 3
3906          FuelOxid.ComboBox1.AddItem "Chemical Equivalence", 4
3907      End If
3908      FuelOxid.ComboBox1.ListIndex = 0
3909  'Set up the Intervals ComboBoxs
3910      If FuelOxid.ComboBox3.ListCount = 0 Then
3911          For i = 1 To Parameter.Cells(9, 2) - 1
3912              FuelOxid.ComboBox3.AddItem i, (i - 1)
3913          Next i
3914      End If
3915  End Sub
3916  'This sets up the FuelOxid Screen
3917  Sub FuelOxid_Setup()
3918      Dim tempLabel As Object
3919      Dim tempValue As Object
3920      If FuelOxid.Frame1.Height < (12 + 18 * Parameter.Cells(9, 2)) Then
3921          FuelOxid.Frame1.ScrollBars = fmScrollBarsVertical
3922          FuelOxid.Frame1.ScrollHeight = 12 + 18 * Parameter.Cells(9, 2)
3923      End If
3924      For i = 1 To Parameter.Cells(9, 2)
3925          Set tempLabel = FuelOxid.Frame1.Controls.Add("Forms.Label.1", "F" & i, True)
3926          With tempLabel
3927              .Caption = "F/O" & i
3928              .Height = 18
3929              .Width = 24
3930              .Left = 4
3931              .Top = 10 + 18 * (i - 1)
3932          End With
3933          Set tempValue = FuelOxid.Frame1.Controls.Add("Forms.TextBox.1", "Fuel" & i, True)
3934          With tempValue
3935              .Height = 18
3936              .Width = 70
3937              .Left = 26
3938              .Top = 8 + 18 * (i - 1)
3939              .TextAlign = 3
3940          End With
3941      Next i
3942  End Sub
3943  'This saves the user entered Fuel-Oxidant Properties
3944  Sub FuelOxid_Update()
3945      Dim stpTemp As Long
3946      Dim valueTemp As Object
3947      chkphi = False
3948      chkeqrats = False
3949      FuelOK = True
3950  'Data validation
3951      If FuelOxid.TextBox1 <> "" And FuelOxid.TextBox2 <> "" Then
3952          If IsNumeric(FuelOxid.TextBox1) = False Or IsNumeric(FuelOxid.TextBox2) = False
Then FuelOK = False
3953      End If
3954      For i = 1 To Parameter.Cells(9, 2)
3955          Set valueTemp = FuelOxid.Controls.Item("Fuel" & i)
3956          If valueTemp.Value <> "" And IsNumeric(valueTemp.Value) = False Then FuelOK = False
3957      Next i
3958      If FuelOK Then
3959  'Input the Temperatures from FuelOxid
3960          If FuelOxid.TextBox1 <> "" And FuelOxid.TextBox2 <> "" And
FuelOxid.ComboBox3.ListIndex >= 0 Then
```

```
3961              If (FuelOxid.TextBox1.Value - FuelOxid.TextBox2.Value) > 0 And
FuelOxid.TextBox2.Value > 0 Then
3962                  stpTemp = (FuelOxid.TextBox1.Value - FuelOxid.TextBox2.Value) /
FuelOxid.ComboBox3.Value
3963                  If FuelOxid.ComboBox1.ListIndex = 0 Then
3964                      Inpt.Cells(2, 10) = (100 - FuelOxid.TextBox2.Value) /
FuelOxid.TextBox2.Value
3965                      Inpt.Cells(FuelOxid.ComboBox3.ListIndex + 3, 10) = (100 -
FuelOxid.TextBox1.Value) / FuelOxid.TextBox1.Value
3966                      If FuelOxid.ComboBox3.ListIndex > 0 Then
3967                          For i = 1 To FuelOxid.ComboBox3.ListIndex
3968                              Inpt.Cells(2 + i, 10) = (100 - (FuelOxid.TextBox2.Value +
stpTemp * i)) / (FuelOxid.TextBox2.Value + stpTemp * i)
3969                          Next i
3970                      End If
3971                  ElseIf FuelOxid.ComboBox1.ListIndex = 1 Then
3972                      Inpt.Cells(2, 10) = 1 / FuelOxid.TextBox2.Value
3973                      Inpt.Cells(FuelOxid.ComboBox3.ListIndex + 3, 10) = 1 /
FuelOxid.TextBox1.Value
3974                      If FuelOxid.ComboBox3.ListIndex > 0 Then
3975                          For i = 1 To FuelOxid.ComboBox3.ListIndex
3976                              Inpt.Cells(2 + i, 10) = 1 / (FuelOxid.TextBox2.Value + stpTemp
* i)
3977                          Next i
3978                      End If
3979                  ElseIf FuelOxid.ComboBox1.ListIndex = 2 Then
3980                      Inpt.Cells(2, 10) = FuelOxid.TextBox2.Value
3981                      Inpt.Cells(FuelOxid.ComboBox3.ListIndex + 3, 10) =
FuelOxid.TextBox1.Value
3982                      If FuelOxid.ComboBox3.ListIndex > 0 Then
3983                          For i = 1 To FuelOxid.ComboBox3.ListIndex
3984                              Inpt.Cells(2 + i, 10) = FuelOxid.TextBox2.Value + stpTemp * i
3985                          Next i
3986                      End If
3987                  ElseIf FuelOxid.ComboBox1.ListIndex = 3 Then
3988                      chkphi = True
3989                      Inpt.Cells(2, 10) = FuelOxid.TextBox2.Value
3990                      Inpt.Cells(FuelOxid.ComboBox3.ListIndex + 3, 10) =
FuelOxid.TextBox1.Value
3991                      If FuelOxid.ComboBox3.ListIndex > 0 Then
3992                          For i = 1 To FuelOxid.ComboBox3.ListIndex
3993                              Inpt.Cells(2 + i, 10) = FuelOxid.TextBox2.Value + stpTemp * i
3994                          Next i
3995                      End If
3996                  ElseIf FuelOxid.ComboBox1.ListIndex = 4 Then
3997                      chkeqrats = True
3998                      Inpt.Cells(2, 10) = FuelOxid.TextBox2.Value
3999                      Inpt.Cells(FuelOxid.ComboBox3.ListIndex + 3, 10) =
FuelOxid.TextBox1.Value
4000                      If FuelOxid.ComboBox3.ListIndex > 0 Then
4001                          For i = 1 To FuelOxid.ComboBox3.ListIndex
4002                              Inpt.Cells(2 + i, 10) = FuelOxid.TextBox2.Value + stpTemp * i
4003                          Next i
4004                      End If
4005                  End If
4006                  Indx.Cells(2, 9) = FuelOxid.ComboBox3.ListIndex + 2
4007              Else
4008                  MsgBox ("Error Loading Fuel to Oxidant")
4009              End If
4010          Else
4011              Indx.Cells(2, 9) = 0
4012              For i = 1 To Parameter.Cells(9, 2)
4013                  Set valueTemp = FuelOxid.Controls.Item("Fuel" & i)
4014                  If valueTemp.Value <> "" Then
4015                      If FuelOxid.ComboBox1.ListIndex = 0 Then
4016                          If valueTemp.Value <> 0 Then
4017                              Inpt.Cells(i + 1, 10) = (100 - valueTemp.Value) /
valueTemp.Value
4018                          Else
4019                              Inpt.Cells(i + 1, 10) = 0
4020                          End If
```

```
                    ElseIf FuelOxid.ComboBox1.ListIndex = 1 Then
                        If valueTemp.Value <> 0 Then
                            Inpt.Cells(i + 1, 10) = 1 / valueTemp.Value
                        Else
                            Inpt.Cells(i + 1, 10) = 0
                        End If
                    ElseIf FuelOxid.ComboBox1.ListIndex = 2 Then
                        Inpt.Cells(i + 1, 10) = valueTemp.Value
                    ElseIf FuelOxid.ComboBox1.ListIndex = 3 Then
                        chkphi = True
                        Inpt.Cells(i + 1, 10) = valueTemp.Value
                    ElseIf FuelOxid.ComboBox1.ListIndex = 4 Then
                        chkeqrats = True
                        Inpt.Cells(i + 1, 10) = valueTemp.Value
                    End If
                    Indx.Cells(2, 9) = i
                Else
                    Inpt.Cells(i + 1, 10).ClearContents
                End If
            Next i
        End If
    Else
        MsgBox ("Values for Fuel-Oxidant Ratios must be numeric")
    End If
End Sub

'*********************************************************************
'ProbTP - Temperature and Pressure
'*********************************************************************
'This sets up the ProbTP Screen
Sub ProbTP_Setup()
    Dim tempLabel As Object
    Dim tempValue As Object
'Temperatures
    If ProbTP.Frame1.Height < (12 + 18 * Parameter.Cells(10, 2)) Then
        ProbTP.Frame1.ScrollBars = fmScrollBarsVertical
        ProbTP.Frame1.ScrollHeight = 12 + 18 * Parameter.Cells(10, 2)
    End If
    For i = 1 To Parameter.Cells(10, 2)
        Set tempLabel = ProbTP.Frame1.Controls.Add("Forms.Label.1", "T" & i, True)
        With tempLabel
            .Caption = "T" & i
            .Height = 18
            .Width = 15
            .Left = 4
            .Top = 10 + 18 * (i - 1)
        End With
        Set tempValue = ProbTP.Frame1.Controls.Add("Forms.TextBox.1", "Temp" & i, True)
        With tempValue
            .Height = 18
            .Width = 70
            .Left = 20
            .Top = 8 + 18 * (i - 1)
            .TextAlign = 3
        End With
    Next i
'Pressures
    If ProbTP.Frame2.Height < (12 + 18 * Parameter.Cells(11, 2)) Then
        ProbTP.Frame2.ScrollBars = fmScrollBarsVertical
        ProbTP.Frame2.ScrollHeight = 12 + 18 * Parameter.Cells(11, 2)
    End If
    For i = 1 To Parameter.Cells(11, 2)
        Set tempLabel = ProbTP.Frame2.Controls.Add("Forms.Label.1", "P" & i, True)
        With tempLabel
            .Caption = "P" & i
            .Height = 18
            .Width = 15
            .Left = 4
            .Top = 10 + 18 * (i - 1)
        End With
        Set tempValue = ProbTP.Frame2.Controls.Add("Forms.TextBox.1", "Press" & i, True)
```

```
4092          With tempValue
4093              .Height = 18
4094              .Width = 70
4095              .Left = 20
4096              .Top = 8 + 18 * (i - 1)
4097              .TextAlign = 3
4098          End With
4099      Next i
4100  'Set up the Temperature Unit ComboBox
4101      If ProbTP.ComboBox1.ListCount = 0 Then
4102          ProbTP.ComboBox1.AddItem "K", 0
4103          ProbTP.ComboBox1.AddItem "R", 1
4104          ProbTP.ComboBox1.AddItem "C", 2
4105          ProbTP.ComboBox1.AddItem "F", 3
4106      End If
4107      ProbTP.ComboBox1.ListIndex = 0
4108  'Set up the Pressure Unit ComboBox
4109      If ProbTP.ComboBox2.ListCount = 0 Then
4110          ProbTP.ComboBox2.AddItem "BAR", 0
4111          ProbTP.ComboBox2.AddItem "ATM", 1
4112          ProbTP.ComboBox2.AddItem "PSI", 2
4113          ProbTP.ComboBox2.AddItem "mmH", 3
4114      End If
4115      ProbTP.ComboBox2.ListIndex = 0
4116  'Set up the Intervals ComboBoxs
4117      If ProbTP.ComboBox3.ListCount = 0 Then
4118          For i = 1 To Parameter.Cells(10, 2) - 1
4119              ProbTP.ComboBox3.AddItem i, (i - 1)
4120          Next i
4121          ProbTP.ComboBox3.ListIndex = 0
4122      End If
4123      If ProbTP.ComboBox4.ListCount = 0 Then
4124          For i = 1 To Parameter.Cells(11, 2) - 1
4125              ProbTP.ComboBox4.AddItem i, (i - 1)
4126          Next i
4127          ProbTP.ComboBox4.ListIndex = 0
4128      End If
4129  End Sub
4130  'This saves the Problem Temperature and Pressure
4131  Sub InputTP()
4132      Dim stpTemp As Double
4133      Dim stpPress As Double
4134      Dim valueTemp As Object
4135      Dim valuePress As Object
4136      CaseOk = True
4137  'Data validation
4138      If ProbTP.TextBox1 <> "" And ProbTP.TextBox2 <> "" Then
4139          If IsNumeric(ProbTP.TextBox1) = False Or IsNumeric(ProbTP.TextBox2) = False Then
CaseOk = False
4140      End If
4141      If ProbTP.TextBox3 <> "" And ProbTP.TextBox4 <> "" Then
4142          If IsNumeric(ProbTP.TextBox3) = False Or IsNumeric(ProbTP.TextBox4) = False Then
CaseOk = False
4143      End If
4144      For i = 1 To Parameter.Cells(10, 2)
4145          Set valueTemp = ProbTP.Controls.Item("Temp" & i)
4146          If valueTemp.Value <> "" And IsNumeric(valueTemp.Value) = False Then CaseOk = False
4147      Next i
4148      For i = 1 To Parameter.Cells(11, 2)
4149          Set valuePress = ProbTP.Controls.Item("Press" & i)
4150          If valuePress.Value <> "" And IsNumeric(valuePress.Value) = False Then CaseOk =
False
4151      Next i
4152      If CaseOk Then
4153  'Input the Temperatures from ProbTP
4154          If ProbTP.TextBox1 <> "" And ProbTP.TextBox2 <> "" And ProbTP.ComboBox3.ListIndex
>= 0 Then
4155              If (ProbTP.TextBox1.Value - ProbTP.TextBox2.Value) > 0 Then
4156                  stpTemp = (ProbTP.TextBox1.Value - ProbTP.TextBox2.Value) /
ProbTP.ComboBox3.Value
4157                  If ProbTP.ComboBox1.ListIndex = 0 Then
```

```
4158                        Inpt.Cells(2, 13) = ProbTP.TextBox2.Value
4159                        Inpt.Cells(ProbTP.ComboBox3.ListIndex + 3, 13) = ProbTP.TextBox1.Value
4160                        If ProbTP.ComboBox3.ListIndex > 0 Then
4161                            For i = 1 To ProbTP.ComboBox3.ListIndex
4162                                Inpt.Cells(2 + i, 13) = ProbTP.TextBox2.Value + stpTemp * i
4163                            Next i
4164                        End If
4165                    ElseIf ProbTP.ComboBox1.ListIndex = 1 Then
4166                        Inpt.Cells(2, 13) = ProbTP.TextBox2.Value / 1.8
4167                        Inpt.Cells(ProbTP.ComboBox3.ListIndex + 3, 13) = ProbTP.TextBox1.Value
/ 1.8
4168                        If ProbTP.ComboBox3.ListIndex > 0 Then
4169                            For i = 1 To ProbTP.ComboBox3.ListIndex
4170                                Inpt.Cells(2 + i, 13) = (ProbTP.TextBox2.Value + stpTemp * i) /
1.8
4171                            Next i
4172                        End If
4173                    ElseIf ProbTP.ComboBox1.ListIndex = 2 Then
4174                        Inpt.Cells(2, 13) = ProbTP.TextBox2.Value + 273.15
4175                        Inpt.Cells(ProbTP.ComboBox3.ListIndex + 3, 13) = ProbTP.TextBox1.Value
+ 273.15
4176                        If ProbTP.ComboBox3.ListIndex > 0 Then
4177                            For i = 1 To ProbTP.ComboBox3.ListIndex
4178                                Inpt.Cells(2 + i, 13) = ProbTP.TextBox2.Value + stpTemp * i +
273.15
4179                            Next i
4180                        End If
4181                    ElseIf ProbTP.ComboBox1.ListIndex = 3 Then
4182                        Inpt.Cells(2, 13) = (ProbTP.TextBox2.Value - 32) / 1.8 + 273.15
4183                        Inpt.Cells(ProbTP.ComboBox3.ListIndex + 3, 13) = (ProbTP.TextBox1.Value
- 32) / 1.8 + 273.15
4184                        If ProbTP.ComboBox3.ListIndex > 0 Then
4185                            For i = 1 To ProbTP.ComboBox3.ListIndex
4186                                Inpt.Cells(2 + i, 13) = (ProbTP.TextBox2.Value + stpTemp * i -
32) / 1.8 + 273.15
4187                            Next i
4188                        End If
4189                    End If
4190                    Indx.Cells(2, 19) = ProbTP.ComboBox3.ListIndex + 2
4191                Else
4192                    MsgBox ("Temperature Inputs Must follow Max/Min Format.")
4193                    CaseOk = False
4194                    GoTo 200
4195                End If
4196            Else
4197                For i = 1 To Parameter.Cells(10, 2)
4198                    Set valueTemp = ProbTP.Controls.Item("Temp" & i)
4199                    If valueTemp.Value <> "" Then
4200                        If ProbTP.ComboBox1.ListIndex = 0 Then
4201                            Inpt.Cells(i + 1, 13) = valueTemp.Value
4202                        ElseIf ProbTP.ComboBox1.ListIndex = 1 Then
4203                            Inpt.Cells(i + 1, 13) = valueTemp.Value / 1.8
4204                        ElseIf ProbTP.ComboBox1.ListIndex = 2 Then
4205                            Inpt.Cells(i + 1, 13) = valueTemp.Value + 273.15
4206                        ElseIf ProbTP.ComboBox1.ListIndex = 3 Then
4207                            Inpt.Cells(i + 1, 13) = (valueTemp.Value - 32) / 1.8 + 273.15
4208                        End If
4209                        Indx.Cells(2, 19) = i
4210                    End If
4211                Next i
4212            End If
4213  'Input the Pressures from ProbTP
4214            If ProbTP.TextBox3 <> "" And ProbTP.TextBox4 <> "" And ProbTP.ComboBox4.ListIndex
>= 0 Then
4215                If (ProbTP.TextBox3.Value - ProbTP.TextBox4.Value) > 0 Then
4216                    stpPress = (ProbTP.TextBox3.Value - ProbTP.TextBox4.Value) /
ProbTP.ComboBox4.Value
4217                    If ProbTP.ComboBox2.ListIndex = 0 Then
4218                        Inpt.Cells(2, 11) = ProbTP.TextBox4.Value
4219                        Inpt.Cells(ProbTP.ComboBox4.ListIndex + 3, 11) = ProbTP.TextBox3.Value
4220                        If ProbTP.ComboBox4.ListIndex > 0 Then
```

```
4221                            For i = 1 To ProbTP.ComboBox4.ListIndex
4222                                Inpt.Cells(2 + i, 11) = (ProbTP.TextBox4.Value + stpPress * i)
4223                            Next i
4224                        End If
4225                    ElseIf ProbTP.ComboBox2.ListIndex = 1 Then
4226                        Inpt.Cells(2, 11) = ProbTP.TextBox4.Value * 1.01325
4227                        Inpt.Cells(ProbTP.ComboBox4.ListIndex + 3, 11) = ProbTP.TextBox3.Value
* 1.01325
4228                        If ProbTP.ComboBox4.ListIndex > 0 Then
4229                            For i = 1 To ProbTP.ComboBox4.ListIndex
4230                                Inpt.Cells(2 + i, 11) = (ProbTP.TextBox4.Value + stpPress * i)
* 1.01325
4231                            Next i
4232                        End If
4233                    ElseIf ProbTP.ComboBox2.ListIndex = 2 Then
4234                        Inpt.Cells(2, 11) = (ProbTP.TextBox4.Value) / 14.696006 * 1.01325
4235                        Inpt.Cells(ProbTP.ComboBox4.ListIndex + 3, 11) =
(ProbTP.TextBox3.Value) / 14.696006 * 1.01325
4236                        If ProbTP.ComboBox4.ListIndex > 0 Then
4237                            For i = 1 To ProbTP.ComboBox4.ListIndex
4238                                Inpt.Cells(2 + i, 11) = (ProbTP.TextBox4.Value + stpPress * i)
/ 14.696006 * 1.01325
4239                            Next i
4240                        End If
4241                    ElseIf ProbTP.ComboBox2.ListIndex = 3 Then
4242                        Inpt.Cells(2, 11) = (ProbTP.TextBox4.Value) / 760 * 1.01325
4243                        Inpt.Cells(ProbTP.ComboBox4.ListIndex + 3, 11) =
(ProbTP.TextBox3.Value) / 760 * 1.01325
4244                        If ProbTP.ComboBox4.ListIndex > 0 Then
4245                            For i = 1 To ProbTP.ComboBox4.ListIndex
4246                                Inpt.Cells(2 + i, 11) = (ProbTP.TextBox4.Value + stpPress * i)
/ 760 * 1.01325
4247                            Next i
4248                        End If
4249                    End If
4250                    Indx.Cells(2, 12) = ProbTP.ComboBox4.ListIndex + 2
4251                Else
4252                    MsgBox ("Pressure Inputs Must follow Max/Min Format.")
4253                    CaseOk = False
4254                    GoTo 200
4255                End If
4256            Else
4257                For i = 1 To Parameter.Cells(11, 2)
4258                    Set valuePress = ProbTP.Controls.Item("Press" & i)
4259                    If valuePress.Value <> "" Then
4260                        If ProbTP.ComboBox2.ListIndex = 0 Then
4261                            Inpt.Cells(i + 1, 11) = valuePress.Value
4262                        ElseIf ProbTP.ComboBox2.ListIndex = 1 Then
4263                            Inpt.Cells(i + 1, 11) = valuePress.Value * 1.01325
4264                        ElseIf ProbTP.ComboBox2.ListIndex = 2 Then
4265                            Inpt.Cells(i + 1, 11) = valuePress.Value / 14.696006 * 1.01325
4266                        ElseIf ProbTP.ComboBox2.ListIndex = 3 Then
4267                            Inpt.Cells(i + 1, 11) = valuePress.Value / 760 * 1.01325
4268                        End If
4269                        Indx.Cells(2, 12) = i
4270                    End If
4271                Next i
4272            End If
4273            If CDbl(Inpt.Cells(2, 13)) <= 0 Then
4274                CaseOk = False
4275                MsgBox ("Assigned Values for Temperature Are Missing")
4276            ElseIf ProbTP.Controls.Item("Press1") = "" And ProbTP.TextBox3.Value = "" Then
4277                CaseOk = False
4278                MsgBox ("Assigned Values for Pressure Are Missing")
4279            Else
4280                CaseOk = True
4281            End If
4282        Else
4283            MsgBox ("Values for Temperature and Pressure must be numeric")
4284        End If
4285  200: End Sub
```

```
4286
4287  '*********************************************************************
4288  'ProbTV - Temperature and Specific Volume
4289  '*********************************************************************
4290  'This sets up the ProbTV Screen
4291  Sub ProbTV_Setup()
4292      Dim tempLabel As Object
4293      Dim tempValue As Object
4294  'Temperatures
4295      If ProbTV.Frame1.Height < (12 + 18 * Parameter.Cells(10, 2)) Then
4296          ProbTV.Frame1.ScrollBars = fmScrollBarsVertical
4297          ProbTV.Frame1.ScrollHeight = 12 + 18 * Parameter.Cells(10, 2)
4298      End If
4299      For i = 1 To Parameter.Cells(10, 2)
4300          Set tempLabel = ProbTV.Frame1.Controls.Add("Forms.Label.1", "T" & i, True)
4301          With tempLabel
4302              .Caption = "T" & i
4303              .Height = 18
4304              .Width = 15
4305              .Left = 4
4306              .Top = 10 + 18 * (i - 1)
4307          End With
4308          Set tempValue = ProbTV.Frame1.Controls.Add("Forms.TextBox.1", "Temp" & i, True)
4309          With tempValue
4310              .Height = 18
4311              .Width = 70
4312              .Left = 20
4313              .Top = 8 + 18 * (i - 1)
4314              .TextAlign = 3
4315          End With
4316      Next i
4317  'Specific Volume
4318      If ProbTV.Frame2.Height < (12 + 18 * Parameter.Cells(11, 2)) Then
4319          ProbTV.Frame2.ScrollBars = fmScrollBarsVertical
4320          ProbTV.Frame2.ScrollHeight = 12 + 18 * Parameter.Cells(11, 2)
4321      End If
4322      For i = 1 To Parameter.Cells(11, 2)
4323          Set tempLabel = ProbTV.Frame2.Controls.Add("Forms.Label.1", "V" & i, True)
4324          With tempLabel
4325              .Caption = "V" & i
4326              .Height = 18
4327              .Width = 15
4328              .Left = 4
4329              .Top = 10 + 18 * (i - 1)
4330          End With
4331          Set tempValue = ProbTV.Frame2.Controls.Add("Forms.TextBox.1", "Vol" & i, True)
4332          With tempValue
4333              .Height = 18
4334              .Width = 70
4335              .Left = 20
4336              .Top = 8 + 18 * (i - 1)
4337              .TextAlign = 3
4338          End With
4339      Next i
4340  'Density
4341      If ProbTV.Frame8.Height < (12 + 18 * Parameter.Cells(11, 2)) Then
4342          ProbTV.Frame8.ScrollBars = fmScrollBarsVertical
4343          ProbTV.Frame8.ScrollHeight = 12 + 18 * Parameter.Cells(11, 2)
4344      End If
4345      For i = 1 To Parameter.Cells(11, 2)
4346          Set tempLabel = ProbTV.Frame8.Controls.Add("Forms.Label.1", "D" & i, True)
4347          With tempLabel
4348              .Caption = "D" & i
4349              .Height = 18
4350              .Width = 15
4351              .Left = 4
4352              .Top = 10 + 18 * (i - 1)
4353          End With
4354          Set tempValue = ProbTV.Frame8.Controls.Add("Forms.TextBox.1", "Den" & i, True)
4355          With tempValue
4356              .Height = 18
```

```
4357              .Width = 70
4358              .Left = 20
4359              .Top = 8 + 18 * (i - 1)
4360              .TextAlign = 3
4361          End With
4362      Next i
4363  'Set up the Temperature Unit ComboBox
4364      If ProbTV.ComboBox1.ListCount = 0 Then
4365          ProbTV.ComboBox1.AddItem "K", 0
4366          ProbTV.ComboBox1.AddItem "R", 1
4367          ProbTV.ComboBox1.AddItem "C", 2
4368          ProbTV.ComboBox1.AddItem "F", 3
4369      End If
4370      ProbTV.ComboBox1.ListIndex = 0
4371  'Set up the Volume Unit ComboBox
4372      If ProbTV.ComboBox2.ListCount = 0 Then
4373          ProbTV.ComboBox2.AddItem "cm^3/g", 0
4374          ProbTV.ComboBox2.AddItem "m^3/kg", 1
4375      End If
4376      ProbTV.ComboBox2.ListIndex = 0
4377  'Set up the Density Unit ComboBox
4378      If ProbTV.ComboBox6.ListCount = 0 Then
4379          ProbTV.ComboBox6.AddItem "g/cm^3", 0
4380          ProbTV.ComboBox6.AddItem "kg/m^3", 1
4381      End If
4382      ProbTV.ComboBox6.ListIndex = 0
4383  'Set up the Intervals ComboBoxs
4384      If ProbTV.ComboBox3.ListCount = 0 Then
4385          For i = 1 To Parameter.Cells(10, 2) - 1
4386              ProbTV.ComboBox3.AddItem i, (i - 1)
4387          Next i
4388          ProbTV.ComboBox3.ListIndex = 0
4389      End If
4390      If ProbTV.ComboBox4.ListCount = 0 Then
4391          For i = 1 To Parameter.Cells(11, 2) - 1
4392              ProbTV.ComboBox4.AddItem i, (i - 1)
4393          Next i
4394          ProbTV.ComboBox4.ListIndex = 0
4395      End If
4396      If ProbTV.ComboBox5.ListCount = 0 Then
4397          For i = 1 To Parameter.Cells(11, 2) - 1
4398              ProbTV.ComboBox5.AddItem i, (i - 1)
4399          Next i
4400          ProbTV.ComboBox5.ListIndex = 0
4401      End If
4402  End Sub
4403  'This saves the Problem Temperature and Volume
4404  Sub InputTV()
4405      Dim stpTemp As Double
4406      Dim stpVol As Double
4407      Dim valueTemp As Object
4408      Dim valueVol As Object
4409      CaseOk = True
4410  'Data validation
4411      If ProbTV.TextBox1 <> "" And ProbTV.TextBox2 <> "" Then
4412          If IsNumeric(ProbTV.TextBox1) = False Or IsNumeric(ProbTV.TextBox2) = False Then
CaseOk = False
4413      End If
4414      If ProbTV.TextBox3 <> "" And ProbTV.TextBox4 <> "" Then
4415          If IsNumeric(ProbTV.TextBox3) = False Or IsNumeric(ProbTV.TextBox4) = False Then
CaseOk = False
4416      End If
4417      If ProbTV.TextBox5 <> "" And ProbTV.TextBox6 <> "" Then
4418          If IsNumeric(ProbTV.TextBox5) = False Or IsNumeric(ProbTV.TextBox6) = False Then
CaseOk = False
4419      End If
4420      For i = 1 To Parameter.Cells(10, 2)
4421          Set valueTemp = ProbTV.Controls.Item("Temp" & i)
4422          If valueTemp.Value <> "" And IsNumeric(valueTemp.Value) = False Then CaseOk = False
4423      Next i
4424      For i = 1 To Parameter.Cells(11, 2)
```

```
4425          Set valueVol = ProbTV.Controls.Item("Vol" & i)
4426          If valueVol.Value <> "" And IsNumeric(valueVol.Value) = False Then CaseOk = False
4427      Next i
4428      For i = 1 To Parameter.Cells(11, 2)
4429          Set valueVol = ProbTV.Controls.Item("Den" & i)
4430          If valueVol.Value <> "" And IsNumeric(valueVol.Value) = False Then CaseOk = False
4431      Next i
4432      If CaseOk Then
4433  'Input the Temperatures from ProbTV
4434          If ProbTV.TextBox1 <> "" And ProbTV.TextBox2 <> "" And ProbTV.ComboBox3.ListIndex
>= 0 Then
4435              If (ProbTV.TextBox1.Value - ProbTV.TextBox2.Value) > 0 Then
4436                  stpTemp = (ProbTV.TextBox1.Value - ProbTV.TextBox2.Value) /
ProbTV.ComboBox3.Value
4437                  If ProbTV.ComboBox1.ListIndex = 0 Then
4438                      Inpt.Cells(2, 13) = ProbTV.TextBox2.Value
4439                      Inpt.Cells(ProbTV.ComboBox3.ListIndex + 3, 13) = ProbTV.TextBox1.Value
4440                      If ProbTV.ComboBox3.ListIndex > 0 Then
4441                          For i = 1 To ProbTV.ComboBox3.ListIndex
4442                              Inpt.Cells(2 + i, 13) = ProbTV.TextBox2.Value + stpTemp * i
4443                          Next i
4444                      End If
4445                  ElseIf ProbTV.ComboBox1.ListIndex = 1 Then
4446                      Inpt.Cells(2, 13) = ProbTV.TextBox2.Value / 1.8
4447                      Inpt.Cells(ProbTV.ComboBox3.ListIndex + 3, 13) = ProbTV.TextBox1.Value
/ 1.8
4448                      If ProbTV.ComboBox3.ListIndex > 0 Then
4449                          For i = 1 To ProbTV.ComboBox3.ListIndex
4450                              Inpt.Cells(2 + i, 13) = (ProbTV.TextBox2.Value + stpTemp * i) /
1.8
4451                          Next i
4452                      End If
4453                  ElseIf ProbTV.ComboBox1.ListIndex = 2 Then
4454                      Inpt.Cells(2, 13) = ProbTV.TextBox2.Value + 273.15
4455                      Inpt.Cells(ProbTV.ComboBox3.ListIndex + 3, 13) = ProbTV.TextBox1.Value
+ 273.15
4456                      If ProbTV.ComboBox3.ListIndex > 0 Then
4457                          For i = 1 To ProbTV.ComboBox3.ListIndex
4458                              Inpt.Cells(2 + i, 13) = ProbTV.TextBox2.Value + stpTemp * i +
273.15
4459                          Next i
4460                      End If
4461                  ElseIf ProbTV.ComboBox1.ListIndex = 3 Then
4462                      Inpt.Cells(2, 13) = (ProbTV.TextBox2.Value - 32) / 1.8 + 273.15
4463                      Inpt.Cells(ProbTV.ComboBox3.ListIndex + 3, 13) = (ProbTV.TextBox1.Value
- 32) / 1.8 + 273.15
4464                      If ProbTV.ComboBox3.ListIndex > 0 Then
4465                          For i = 1 To ProbTV.ComboBox3.ListIndex
4466                              Inpt.Cells(2 + i, 13) = (ProbTV.TextBox2.Value + stpTemp * i -
32) / 1.8 + 273.15
4467                          Next i
4468                      End If
4469                  End If
4470                  Indx.Cells(2, 19) = ProbTV.ComboBox3.ListIndex + 2
4471              Else
4472                  MsgBox ("Temperature Inputs Must follow Max/Min Format.")
4473                  CaseOk = False
4474                  GoTo 200
4475              End If
4476          Else
4477              For i = 1 To Parameter.Cells(10, 2)
4478                  Set valueTemp = ProbTV.Controls.Item("Temp" & i)
4479                  If valueTemp.Value <> "" Then
4480                      If ProbTV.ComboBox1.ListIndex = 0 Then
4481                          Inpt.Cells(i + 1, 13) = valueTemp.Value
4482                      ElseIf ProbTV.ComboBox1.ListIndex = 1 Then
4483                          Inpt.Cells(i + 1, 13) = valueTemp.Value / 1.8
4484                      ElseIf ProbTV.ComboBox1.ListIndex = 2 Then
4485                          Inpt.Cells(i + 1, 13) = valueTemp.Value + 273.15
4486                      ElseIf ProbTV.ComboBox1.ListIndex = 3 Then
4487                          Inpt.Cells(i + 1, 13) = (valueTemp.Value - 32) / 1.8 + 273.15
```

```
4488                        End If
4489                        Indx.Cells(2, 19) = i
4490                    End If
4491                Next i
4492            End If
4493  'Input the Volume or Density from ProbTV
4494            If ProbTV.TextBox3 <> "" And ProbTV.TextBox4 <> "" And ProbTV.ComboBox4.ListIndex
>= 0 Then
4495                If (ProbTV.TextBox3.Value - ProbTV.TextBox4.Value) > 0 Then
4496                    stpVol = (ProbTV.TextBox3.Value - ProbTV.TextBox4.Value) /
ProbTV.ComboBox4.Value
4497                    If ProbTV.ComboBox2.ListIndex = 0 Then
4498                        Inpt.Cells(2, 14) = ProbTV.TextBox4.Value * 100
4499                        Inpt.Cells(ProbTV.ComboBox4.ListIndex + 3, 14) = ProbTV.TextBox3.Value
* 100
4500                        If ProbTV.ComboBox4.ListIndex > 0 Then
4501                            For i = 1 To ProbTV.ComboBox4.ListIndex
4502                                Inpt.Cells(2 + i, 14) = (ProbTV.TextBox4.Value + stpVol * i) *
100
4503                            Next i
4504                        End If
4505                    ElseIf ProbTV.ComboBox2.ListIndex = 1 Then
4506                        Inpt.Cells(2, 14) = ProbTV.TextBox4.Value * 100000
4507                        Inpt.Cells(ProbTV.ComboBox4.ListIndex + 3, 14) = ProbTV.TextBox3.Value
* 100000
4508                        If ProbTV.ComboBox4.ListIndex > 0 Then
4509                            For i = 1 To ProbTV.ComboBox4.ListIndex
4510                                Inpt.Cells(2 + i, 14) = (ProbTV.TextBox4.Value + stpVol * i) *
100000
4511                            Next i
4512                        End If
4513                    End If
4514                    Indx.Cells(2, 12) = ProbTV.ComboBox4.ListIndex + 2
4515                Else
4516                    MsgBox ("Volume Inputs Must follow Max/Min Format.")
4517                    CaseOk = False
4518                    GoTo 200
4519                End If
4520            ElseIf ProbTV.TextBox5 <> "" And ProbTV.TextBox6 <> "" And
ProbTV.ComboBox5.ListIndex >= 0 Then
4521                If (ProbTV.TextBox5.Value - ProbTV.TextBox6.Value) > 0 Then
4522                    stpVol = (ProbTV.TextBox5.Value - ProbTV.TextBox6.Value) /
ProbTV.ComboBox5.Value
4523                    If ProbTV.ComboBox6.ListIndex = 0 Then
4524                        Inpt.Cells(2, 14) = 100 / ProbTV.TextBox6.Value
4525                        Inpt.Cells(ProbTV.ComboBox5.ListIndex + 3, 14) = 100 /
ProbTV.TextBox5.Value
4526                        If ProbTV.ComboBox5.ListIndex > 0 Then
4527                            For i = 1 To ProbTV.ComboBox5.ListIndex
4528                                Inpt.Cells(2 + i, 14) = 100 / (ProbTV.TextBox6.Value + stpVol *
i)
4529                            Next i
4530                        End If
4531                    ElseIf ProbTV.ComboBox6.ListIndex = 1 Then
4532                        Inpt.Cells(2, 14) = 100000 / ProbTV.TextBox6.Value
4533                        Inpt.Cells(ProbTV.ComboBox5.ListIndex + 3, 14) = 100000 /
ProbTV.TextBox5.Value
4534                        If ProbTV.ComboBox5.ListIndex > 0 Then
4535                            For i = 1 To ProbTV.ComboBox5.ListIndex
4536                                Inpt.Cells(2 + i, 14) = 100000 / (ProbTV.TextBox6.Value +
stpVol * i)
4537                            Next i
4538                        End If
4539                    End If
4540                    Indx.Cells(2, 12) = ProbTV.ComboBox5.ListIndex + 2
4541                Else
4542                    MsgBox ("Density Inputs Must follow Max/Min Format.")
4543                    CaseOk = False
4544                    GoTo 200
4545                End If
4546            Else
```

```
4547             Set valueVol = ProbTV.Controls.Item("Vol1")
4548             If valueVol.Value <> "" Then
4549                 For i = 1 To Parameter.Cells(11, 2)
4550                     Set valueVol = ProbTV.Controls.Item("Vol" & i)
4551                     If valueVol.Value <> "" Then
4552                         If ProbTV.ComboBox2.ListIndex = 0 Then
4553                             Inpt.Cells(i + 1, 14) = valueVol.Value * 100
4554                         ElseIf ProbTV.ComboBox2.ListIndex = 1 Then
4555                             Inpt.Cells(i + 1, 14) = valueVol.Value * 100000
4556                         End If
4557                         Indx.Cells(2, 12) = i
4558                     End If
4559                 Next i
4560             Else
4561                 For i = 1 To Parameter.Cells(11, 2)
4562                     Set valueVol = ProbTV.Controls.Item("Den" & i)
4563                     If valueVol.Value <> "" Then
4564                         If ProbTV.ComboBox6.ListIndex = 0 Then
4565                             Inpt.Cells(i + 1, 14) = 100 / valueVol.Value
4566                         ElseIf ProbTV.ComboBox6.ListIndex = 1 Then
4567                             Inpt.Cells(i + 1, 14) = 100000 / valueVol.Value
4568                         End If
4569                         Indx.Cells(2, 12) = i
4570                     End If
4571                 Next i
4572             End If
4573         End If
4574         If CDbl(Inpt.Cells(2, 13)) <= 0 Then
4575             CaseOk = False
4576             MsgBox ("Assigned Values for Temperature Are Missing")
4577         ElseIf ProbTV.Controls.Item("Vol1") = "" And ProbTV.TextBox3.Value = "" And
ProbTV.Controls.Item("Den1") = "" And ProbTV.TextBox5.Value = "" Then
4578             CaseOk = False
4579             MsgBox ("Assigned Values for Volume Or Density Are Missing")
4580         Else
4581             CaseOk = True
4582         End If
4583     Else
4584         MsgBox ("Temperature and Volume/Density must be numeric")
4585     End If
4586 200: End Sub
4587
4588 '*********************************************************************
4589 'ProbUV - Internal Energy and Specific Volume
4590 '*********************************************************************
4591 'This sets up the ProbUV Screen
4592 Sub ProbUV_Setup()
4593     Dim tempLabel As Object
4594     Dim tempValue As Object
4595 'Specific Volume
4596     If ProbUV.Frame2.Height < (12 + 18 * Parameter.Cells(11, 2)) Then
4597         ProbUV.Frame2.ScrollBars = fmScrollBarsVertical
4598         ProbUV.Frame2.ScrollHeight = 12 + 18 * Parameter.Cells(11, 2)
4599     End If
4600     For i = 1 To Parameter.Cells(11, 2)
4601         Set tempLabel = ProbUV.Frame2.Controls.Add("Forms.Label.1", "V" & i, True)
4602         With tempLabel
4603             .Caption = "V" & i
4604             .Height = 18
4605             .Width = 15
4606             .Left = 4
4607             .Top = 10 + 18 * (i - 1)
4608         End With
4609         Set tempValue = ProbUV.Frame2.Controls.Add("Forms.TextBox.1", "Vol" & i, True)
4610         With tempValue
4611             .Height = 18
4612             .Width = 70
4613             .Left = 20
4614             .Top = 8 + 18 * (i - 1)
4615             .TextAlign = 3
4616         End With
```

```
4617      Next i
4618  'Density
4619      If ProbUV.Frame8.Height < (12 + 18 * Parameter.Cells(11, 2)) Then
4620          ProbUV.Frame8.ScrollBars = fmScrollBarsVertical
4621          ProbUV.Frame8.ScrollHeight = 12 + 18 * Parameter.Cells(11, 2)
4622      End If
4623      For i = 1 To Parameter.Cells(11, 2)
4624          Set tempLabel = ProbUV.Frame8.Controls.Add("Forms.Label.1", "D" & i, True)
4625          With tempLabel
4626              .Caption = "D" & i
4627              .Height = 18
4628              .Width = 15
4629              .Left = 4
4630              .Top = 10 + 18 * (i - 1)
4631          End With
4632          Set tempValue = ProbUV.Frame8.Controls.Add("Forms.TextBox.1", "Den" & i, True)
4633          With tempValue
4634              .Height = 18
4635              .Width = 70
4636              .Left = 20
4637              .Top = 8 + 18 * (i - 1)
4638              .TextAlign = 3
4639          End With
4640      Next i
4641  'Set up the Volume Unit ComboBox
4642      If ProbUV.ComboBox2.ListCount = 0 Then
4643          ProbUV.ComboBox2.AddItem "cm^3/g", 0
4644          ProbUV.ComboBox2.AddItem "m^3/kg", 1
4645      End If
4646      ProbUV.ComboBox2.ListIndex = 0
4647  'Set up the Density Unit ComboBox
4648      If ProbUV.ComboBox6.ListCount = 0 Then
4649          ProbUV.ComboBox6.AddItem "g/cm^3", 0
4650          ProbUV.ComboBox6.AddItem "kg/m^3", 1
4651      End If
4652      ProbUV.ComboBox6.ListIndex = 0
4653  'Set up the Intervals ComboBoxs
4654      If ProbUV.ComboBox4.ListCount = 0 Then
4655          For i = 1 To Parameter.Cells(11, 2) - 1
4656              ProbUV.ComboBox4.AddItem i, (i - 1)
4657          Next i
4658          ProbUV.ComboBox4.ListIndex = 0
4659      End If
4660      If ProbUV.ComboBox5.ListCount = 0 Then
4661          For i = 1 To Parameter.Cells(11, 2) - 1
4662              ProbUV.ComboBox5.AddItem i, (i - 1)
4663          Next i
4664          ProbUV.ComboBox5.ListIndex = 0
4665      End If
4666  End Sub
4667  'This saves the Problem Interal Energy and Specific Volume
4668  Sub InputUV()
4669      Dim stpVol As Double
4670      Dim valueVol As Object
4671      CaseOk = True
4672  'Data validation
4673      If ProbUV.TextBox7 <> "" And IsNumeric(ProbUV.TextBox7) = False Then CaseOk = False
4674      If ProbUV.TextBox3 <> "" And ProbUV.TextBox4 <> "" Then
4675          If IsNumeric(ProbUV.TextBox3) = False Or IsNumeric(ProbUV.TextBox4) = False Then
CaseOk = False
4676      End If
4677      If ProbUV.TextBox5 <> "" And ProbUV.TextBox6 <> "" Then
4678          If IsNumeric(ProbUV.TextBox5) = False Or IsNumeric(ProbUV.TextBox6) = False Then
CaseOk = False
4679      End If
4680      For i = 1 To Parameter.Cells(11, 2)
4681          Set valueVol = ProbUV.Controls.Item("Vol" & i)
4682          If valueVol.Value <> "" And IsNumeric(valueVol.Value) = False Then CaseOk = False
4683      Next i
4684      For i = 1 To Parameter.Cells(11, 2)
4685          Set valueVol = ProbUV.Controls.Item("Den" & i)
```

```
4686          If valueVol.Value <> "" And IsNumeric(valueVol.Value) = False Then CaseOk = False
4687      Next i
4688      If CaseOk Then
4689  'Input the Volume or Density from ProbUV
4690          If ProbUV.TextBox3 <> "" And ProbUV.TextBox4 <> "" And ProbUV.ComboBox4.ListIndex
>= 0 Then
4691              If (ProbUV.TextBox3.Value - ProbUV.TextBox4.Value) > 0 Then
4692                  stpVol = (ProbUV.TextBox3.Value - ProbUV.TextBox4.Value) /
ProbUV.ComboBox4.Value
4693                  If ProbUV.ComboBox2.ListIndex = 0 Then
4694                      Inpt.Cells(2, 14) = ProbUV.TextBox4.Value * 100
4695                      Inpt.Cells(ProbUV.ComboBox4.ListIndex + 3, 14) = ProbUV.TextBox3.Value
* 100
4696                      If ProbUV.ComboBox4.ListIndex > 0 Then
4697                          For i = 1 To ProbUV.ComboBox4.ListIndex
4698                              Inpt.Cells(2 + i, 14) = (ProbUV.TextBox4.Value + stpVol * i) *
100
4699                          Next i
4700                      End If
4701                  ElseIf ProbUV.ComboBox2.ListIndex = 1 Then
4702                      Inpt.Cells(2, 14) = ProbUV.TextBox4.Value * 100000
4703                      Inpt.Cells(ProbUV.ComboBox4.ListIndex + 3, 14) = ProbUV.TextBox3.Value
* 100000
4704                      If ProbUV.ComboBox4.ListIndex > 0 Then
4705                          For i = 1 To ProbUV.ComboBox4.ListIndex
4706                              Inpt.Cells(2 + i, 14) = (ProbUV.TextBox4.Value + stpVol * i) *
100000
4707                          Next i
4708                      End If
4709                  End If
4710                  Indx.Cells(2, 12) = ProbUV.ComboBox4.ListIndex + 2
4711              Else
4712                  MsgBox ("Volume Inputs Must follow Max/Min Format.")
4713                  CaseOk = False
4714                  GoTo 200
4715              End If
4716          ElseIf ProbUV.TextBox5 <> "" And ProbUV.TextBox6 <> "" And
ProbUV.ComboBox5.ListIndex >= 0 Then
4717              If (ProbUV.TextBox5.Value - ProbUV.TextBox6.Value) > 0 Then
4718                  stpVol = (ProbUV.TextBox5.Value - ProbUV.TextBox6.Value) /
ProbUV.ComboBox5.Value
4719                  If ProbUV.ComboBox6.ListIndex = 0 Then
4720                      Inpt.Cells(2, 14) = 100 / ProbUV.TextBox6.Value
4721                      Inpt.Cells(ProbUV.ComboBox5.ListIndex + 3, 14) = 100 /
ProbUV.TextBox5.Value
4722                      If ProbUV.ComboBox5.ListIndex > 0 Then
4723                          For i = 1 To ProbUV.ComboBox5.ListIndex
4724                              Inpt.Cells(2 + i, 14) = 100 / (ProbUV.TextBox6.Value + stpVol *
i)
4725                          Next i
4726                      End If
4727                  ElseIf ProbUV.ComboBox6.ListIndex = 1 Then
4728                      Inpt.Cells(2, 14) = 100000 / ProbUV.TextBox6.Value
4729                      Inpt.Cells(ProbUV.ComboBox5.ListIndex + 3, 14) = 100000 /
ProbUV.TextBox5.Value
4730                      If ProbUV.ComboBox5.ListIndex > 0 Then
4731                          For i = 1 To ProbUV.ComboBox5.ListIndex
4732                              Inpt.Cells(2 + i, 14) = 100000 / (ProbUV.TextBox6.Value +
stpVol * i)
4733                          Next i
4734                      End If
4735                  End If
4736                  Indx.Cells(2, 12) = ProbUV.ComboBox5.ListIndex + 2
4737              Else
4738                  MsgBox ("Density Inputs Must follow Max/Min Format.")
4739                  CaseOk = False
4740                  GoTo 200
4741              End If
4742          Else
4743              Set valueVol = ProbUV.Controls.Item("Vol1")
4744              If valueVol.Value <> "" Then
```

```
4745                 For i = 1 To Parameter.Cells(11, 2)
4746                     Set valueVol = ProbUV.Controls.Item("Vol" & i)
4747                     If valueVol.Value <> "" Then
4748                         If ProbUV.ComboBox2.ListIndex = 0 Then
4749                             Inpt.Cells(i + 1, 14) = valueVol.Value * 100
4750                         ElseIf ProbUV.ComboBox2.ListIndex = 1 Then
4751                             Inpt.Cells(i + 1, 14) = valueVol.Value * 100000
4752                         End If
4753                         Indx.Cells(2, 12) = i
4754                     End If
4755                 Next i
4756             Else
4757                 For i = 1 To Parameter.Cells(11, 2)
4758                     Set valueVol = ProbUV.Controls.Item("Den" & i)
4759                     If valueVol.Value <> "" Then
4760                         If ProbUV.ComboBox6.ListIndex = 0 Then
4761                             Inpt.Cells(i + 1, 14) = 100 / valueVol.Value
4762                         ElseIf ProbUV.ComboBox6.ListIndex = 1 Then
4763                             Inpt.Cells(i + 1, 14) = 100000 / valueVol.Value
4764                         End If
4765                         Indx.Cells(2, 12) = i
4766                     End If
4767                 Next i
4768             End If
4769         End If
4770  'Input the Internal Energy
4771         If ProbUV.TextBox7.Value <> "" Then Miscr.Cells(7, 4) = ProbUV.TextBox7.Value
4772         If ProbUV.Controls.Item("Vol1") = "" And ProbUV.TextBox3.Value = "" And
ProbUV.Controls.Item("Den1") = "" And ProbUV.TextBox5.Value = "" Then
4773             CaseOk = False
4774             MsgBox ("Assigned Values for Volume Or Density Are Missing")
4775         Else
4776             CaseOk = True
4777         End If
4778     Else
4779         MsgBox ("Volume/Density and Internal-Energy must be numeric")
4780     End If
4781  200: End Sub
4782
4783  '*********************************************************************
4784  'ProbSV - Entropy and Specific Volume
4785  '*********************************************************************
4786  'This sets up the ProbSV Screen
4787  Sub ProbSV_Setup()
4788     Dim tempLabel As Object
4789     Dim tempValue As Object
4790  'Specific Volume
4791     If ProbSV.Frame2.Height < (12 + 18 * Parameter.Cells(11, 2)) Then
4792         ProbSV.Frame2.ScrollBars = fmScrollBarsVertical
4793         ProbSV.Frame2.ScrollHeight = 12 + 18 * Parameter.Cells(11, 2)
4794     End If
4795     For i = 1 To Parameter.Cells(11, 2)
4796         Set tempLabel = ProbSV.Frame2.Controls.Add("Forms.Label.1", "V" & i, True)
4797         With tempLabel
4798             .Caption = "V" & i
4799             .Height = 18
4800             .Width = 15
4801             .Left = 4
4802             .Top = 10 + 18 * (i - 1)
4803         End With
4804         Set tempValue = ProbSV.Frame2.Controls.Add("Forms.TextBox.1", "Vol" & i, True)
4805         With tempValue
4806             .Height = 18
4807             .Width = 70
4808             .Left = 20
4809             .Top = 8 + 18 * (i - 1)
4810             .TextAlign = 3
4811         End With
4812     Next i
4813  'Density
4814     If ProbSV.Frame8.Height < (12 + 18 * Parameter.Cells(11, 2)) Then
```

```
4815          ProbSV.Frame8.ScrollBars = fmScrollBarsVertical
4816          ProbSV.Frame8.ScrollHeight = 12 + 18 * Parameter.Cells(11, 2)
4817      End If
4818      For i = 1 To Parameter.Cells(11, 2)
4819          Set tempLabel = ProbSV.Frame8.Controls.Add("Forms.Label.1", "D" & i, True)
4820          With tempLabel
4821              .Caption = "D" & i
4822              .Height = 18
4823              .Width = 15
4824              .Left = 4
4825              .Top = 10 + 18 * (i - 1)
4826          End With
4827          Set tempValue = ProbSV.Frame8.Controls.Add("Forms.TextBox.1", "Den" & i, True)
4828          With tempValue
4829              .Height = 18
4830              .Width = 70
4831              .Left = 20
4832              .Top = 8 + 18 * (i - 1)
4833              .TextAlign = 3
4834          End With
4835      Next i
4836  'Set up the Volume Unit ComboBox
4837      If ProbSV.ComboBox2.ListCount = 0 Then
4838          ProbSV.ComboBox2.AddItem "cm^3/g", 0
4839          ProbSV.ComboBox2.AddItem "m^3/kg", 1
4840      End If
4841      ProbSV.ComboBox2.ListIndex = 0
4842  'Set up the Density Unit ComboBox
4843      If ProbSV.ComboBox6.ListCount = 0 Then
4844          ProbSV.ComboBox6.AddItem "g/cm^3", 0
4845          ProbSV.ComboBox6.AddItem "kg/m^3", 1
4846      End If
4847      ProbSV.ComboBox6.ListIndex = 0
4848  'Set up the Intervals ComboBoxs
4849      If ProbSV.ComboBox4.ListCount = 0 Then
4850          For i = 1 To Parameter.Cells(11, 2) - 1
4851              ProbSV.ComboBox4.AddItem i, (i - 1)
4852          Next i
4853          ProbSV.ComboBox4.ListIndex = 0
4854      End If
4855      If ProbSV.ComboBox5.ListCount = 0 Then
4856          For i = 1 To Parameter.Cells(11, 2) - 1
4857              ProbSV.ComboBox5.AddItem i, (i - 1)
4858          Next i
4859          ProbSV.ComboBox5.ListIndex = 0
4860      End If
4861  End Sub
4862  'This saves the Problem Entropy and Specific Volume
4863  Sub InputSV()
4864      Dim stpVol As Double
4865      Dim valueVol As Object
4866      CaseOk = True
4867  'Data validation
4868      If ProbSV.TextBox7 <> "" And IsNumeric(ProbSV.TextBox7) = False Then CaseOk = False
4869      If ProbSV.TextBox3 <> "" And ProbSV.TextBox4 <> "" Then
4870          If IsNumeric(ProbSV.TextBox3) = False Or IsNumeric(ProbSV.TextBox4) = False Then
CaseOk = False
4871      End If
4872      If ProbSV.TextBox5 <> "" And ProbSV.TextBox6 <> "" Then
4873          If IsNumeric(ProbSV.TextBox5) = False Or IsNumeric(ProbSV.TextBox6) = False Then
CaseOk = False
4874      End If
4875      For i = 1 To Parameter.Cells(11, 2)
4876          Set valueVol = ProbSV.Controls.Item("Vol" & i)
4877          If valueVol.Value <> "" And IsNumeric(valueVol.Value) = False Then CaseOk = False
4878      Next i
4879      For i = 1 To Parameter.Cells(11, 2)
4880          Set valueVol = ProbSV.Controls.Item("Den" & i)
4881          If valueVol.Value <> "" And IsNumeric(valueVol.Value) = False Then CaseOk = False
4882      Next i
4883      If CaseOk Then
```

```
4884  'Input the Volume or Density from ProbSV
4885          If ProbSV.TextBox3 <> "" And ProbSV.TextBox4 <> "" And ProbSV.ComboBox4.ListIndex
>= 0 Then
4886              If (ProbSV.TextBox3.Value - ProbSV.TextBox4.Value) > 0 Then
4887                  stpVol = (ProbSV.TextBox3.Value - ProbSV.TextBox4.Value) /
ProbSV.ComboBox4.Value
4888                  If ProbSV.ComboBox2.ListIndex = 0 Then
4889                      Inpt.Cells(2, 14) = ProbSV.TextBox4.Value * 100
4890                      Inpt.Cells(ProbSV.ComboBox4.ListIndex + 3, 14) = ProbSV.TextBox3.Value
* 100
4891                      If ProbSV.ComboBox4.ListIndex > 0 Then
4892                          For i = 1 To ProbSV.ComboBox4.ListIndex
4893                              Inpt.Cells(2 + i, 14) = (ProbSV.TextBox4.Value + stpVol * i) *
100
4894                          Next i
4895                      End If
4896                  ElseIf ProbSV.ComboBox2.ListIndex = 1 Then
4897                      Inpt.Cells(2, 14) = ProbSV.TextBox4.Value * 100000
4898                      Inpt.Cells(ProbSV.ComboBox4.ListIndex + 3, 14) = ProbSV.TextBox3.Value
* 100000
4899                      If ProbSV.ComboBox4.ListIndex > 0 Then
4900                          For i = 1 To ProbSV.ComboBox4.ListIndex
4901                              Inpt.Cells(2 + i, 14) = (ProbSV.TextBox4.Value + stpVol * i) *
100000
4902                          Next i
4903                      End If
4904                  End If
4905                  Indx.Cells(2, 12) = ProbSV.ComboBox4.ListIndex + 2
4906              Else
4907                  MsgBox ("Volume Inputs Must follow Max/Min Format.")
4908                  CaseOk = False
4909                  GoTo 200
4910              End If
4911          ElseIf ProbSV.TextBox5 <> "" And ProbSV.TextBox6 <> "" And
ProbSV.ComboBox5.ListIndex >= 0 Then
4912              If (ProbSV.TextBox5.Value - ProbSV.TextBox6.Value) > 0 Then
4913                  stpVol = (ProbSV.TextBox5.Value - ProbSV.TextBox6.Value) /
ProbSV.ComboBox5.Value
4914                  If ProbSV.ComboBox6.ListIndex = 0 Then
4915                      Inpt.Cells(2, 14) = 100 / ProbSV.TextBox6.Value
4916                      Inpt.Cells(ProbSV.ComboBox5.ListIndex + 3, 14) = 100 /
ProbSV.TextBox5.Value
4917                      If ProbSV.ComboBox5.ListIndex > 0 Then
4918                          For i = 1 To ProbSV.ComboBox5.ListIndex
4919                              Inpt.Cells(2 + i, 14) = 100 / (ProbSV.TextBox6.Value + stpVol *
i)
4920                          Next i
4921                      End If
4922                  ElseIf ProbSV.ComboBox6.ListIndex = 1 Then
4923                      Inpt.Cells(2, 14) = 100000 / ProbSV.TextBox6.Value
4924                      Inpt.Cells(ProbSV.ComboBox5.ListIndex + 3, 14) = 100000 /
ProbSV.TextBox5.Value
4925                      If ProbSV.ComboBox5.ListIndex > 0 Then
4926                          For i = 1 To ProbSV.ComboBox5.ListIndex
4927                              Inpt.Cells(2 + i, 14) = 100000 / (ProbSV.TextBox6.Value +
stpVol * i)
4928                          Next i
4929                      End If
4930                  End If
4931                  Indx.Cells(2, 12) = ProbSV.ComboBox5.ListIndex + 2
4932              Else
4933                  MsgBox ("Density Inputs Must follow Max/Min Format.")
4934                  CaseOk = False
4935                  GoTo 200
4936              End If
4937          Else
4938              Set valueVol = ProbSV.Controls.Item("Vol1")
4939              If valueVol.Value <> "" Then
4940                  For i = 1 To Parameter.Cells(11, 2)
4941                      Set valueVol = ProbSV.Controls.Item("Vol" & i)
4942                      If valueVol.Value <> "" Then
```

```
                        If ProbSV.ComboBox2.ListIndex = 0 Then
                            Inpt.Cells(i + 1, 14) = valueVol.Value * 100
                        ElseIf ProbSV.ComboBox2.ListIndex = 1 Then
                            Inpt.Cells(i + 1, 14) = valueVol.Value * 100000
                        End If
                        Indx.Cells(2, 12) = i
                    End If
                Next i
            Else
                For i = 1 To Parameter.Cells(11, 2)
                Set valueVol = ProbSV.Controls.Item("Den" & i)
                If valueVol.Value <> "" Then
                    If ProbSV.ComboBox6.ListIndex = 0 Then
                        Inpt.Cells(i + 1, 14) = 100 / valueVol.Value
                    ElseIf ProbSV.ComboBox6.ListIndex = 1 Then
                        Inpt.Cells(i + 1, 14) = 100000 / valueVol.Value
                    End If
                    Indx.Cells(2, 12) = i
                End If
                Next i
            End If
        End If
'Input the Entropy
        If ProbSV.TextBox7.Value <> "" Then
            Miscr.Cells(2, 11) = ProbSV.TextBox7.Value
            If ProbSV.Controls.Item("Vol1") = "" And ProbSV.TextBox3.Value = "" And
ProbSV.Controls.Item("Den1") = "" And ProbSV.TextBox5.Value = "" Then
                CaseOk = False
                MsgBox ("Assigned Values for Volume Or Density Are Missing")
            Else
                CaseOk = True
            End If
        Else
            CaseOk = False
            MsgBox ("Assigned Value for Mixture Entropy Is Missing")
        End If
    Else
        MsgBox ("Volume/Density and Entropy must be numeric")
    End If
200: End Sub

'*********************************************************************
'ProbSP - Entropy and Pressure
'*********************************************************************
'This sets up the ProbSP Screen
Sub ProbSP_Setup()
    Dim tempLabel As Object
    Dim tempValue As Object
'Pressures
    If ProbSP.Frame2.Height < (12 + 18 * Parameter.Cells(11, 2)) Then
        ProbSP.Frame2.ScrollBars = fmScrollBarsVertical
        ProbSP.Frame2.ScrollHeight = 12 + 18 * Parameter.Cells(11, 2)
    End If
    For i = 1 To Parameter.Cells(11, 2)
        Set tempLabel = ProbSP.Frame2.Controls.Add("Forms.Label.1", "P" & i, True)
        With tempLabel
            .Caption = "P" & i
            .Height = 18
            .Width = 15
            .Left = 4
            .Top = 10 + 18 * (i - 1)
        End With
        Set tempValue = ProbSP.Frame2.Controls.Add("Forms.TextBox.1", "Press" & i, True)
        With tempValue
            .Height = 18
            .Width = 70
            .Left = 20
            .Top = 8 + 18 * (i - 1)
            .TextAlign = 3
        End With
    Next i
```

```
5013  'Set up the Pressure Unit ComboBox
5014      If ProbSP.ComboBox3.ListCount = 0 Then
5015          ProbSP.ComboBox3.AddItem "BAR", 0
5016          ProbSP.ComboBox3.AddItem "ATM", 1
5017          ProbSP.ComboBox3.AddItem "PSI", 2
5018          ProbSP.ComboBox3.AddItem "mmH", 3
5019      End If
5020      ProbSP.ComboBox3.ListIndex = 0
5021  'Set up the Intervals ComboBoxs
5022      If ProbSP.ComboBox1.ListCount = 0 Then
5023          For i = 1 To Parameter.Cells(11, 2) - 1
5024              ProbSP.ComboBox1.AddItem i, (i - 1)
5025          Next i
5026          ProbSP.ComboBox1.ListIndex = 0
5027      End If
5028  End Sub
5029  'This saves the Problem Entropy and Pressure
5030  Sub InputSP()
5031      Dim stpPress As Double
5032      Dim valuePress As Object
5033      CaseOk = True
5034  'Data validation
5035      If ProbSP.TextBox3 <> "" And IsNumeric(ProbSP.TextBox3) = False Then CaseOk = False
5036      If ProbSP.TextBox1 <> "" And ProbSP.TextBox2 <> "" Then
5037          If IsNumeric(ProbSP.TextBox1) = False Or IsNumeric(ProbSP.TextBox2) = False Then
CaseOk = False
5038      End If
5039      For i = 1 To Parameter.Cells(11, 2)
5040          Set valuePress = ProbSP.Controls.Item("Press" & i)
5041          If valuePress.Value <> "" And IsNumeric(valuePress.Value) = False Then CaseOk =
False
5042      Next i
5043      If CaseOk Then
5044  'Input the Pressures from ProbSP
5045          If ProbSP.TextBox1 <> "" And ProbSP.TextBox2 <> "" And ProbSP.ComboBox1.ListIndex
>= 0 Then
5046              If (ProbSP.TextBox1.Value - ProbSP.TextBox2.Value) > 0 Then
5047                  stpPress = (ProbSP.TextBox1.Value - ProbSP.TextBox2.Value) /
ProbSP.ComboBox1.Value
5048                  If ProbSP.ComboBox3.ListIndex = 0 Then
5049                      Inpt.Cells(2, 11) = ProbSP.TextBox2.Value
5050                      Inpt.Cells(ProbSP.ComboBox1.ListIndex + 3, 11) = ProbSP.TextBox1.Value
5051                      If ProbSP.ComboBox1.ListIndex > 0 Then
5052                          For i = 1 To ProbSP.ComboBox1.ListIndex
5053                              Inpt.Cells(2 + i, 11) = (ProbSP.TextBox2.Value + stpPress * i)
5054                          Next i
5055                      End If
5056                  ElseIf ProbSP.ComboBox3.ListIndex = 1 Then
5057                      Inpt.Cells(2, 11) = ProbSP.TextBox2.Value * 1.01325
5058                      Inpt.Cells(ProbSP.ComboBox1.ListIndex + 3, 11) = ProbSP.TextBox1.Value
* 1.01325
5059                      If ProbSP.ComboBox1.ListIndex > 0 Then
5060                          For i = 1 To ProbSP.ComboBox1.ListIndex
5061                              Inpt.Cells(2 + i, 11) = (ProbSP.TextBox2.Value + stpPress * i)
* 1.01325
5062                          Next i
5063                      End If
5064                  ElseIf ProbSP.ComboBox3.ListIndex = 2 Then
5065                      Inpt.Cells(2, 11) = (ProbSP.TextBox2.Value) / 14.696006 * 1.01325
5066                      Inpt.Cells(ProbSP.ComboBox1.ListIndex + 3, 11) =
(ProbSP.TextBox1.Value) / 14.696006 * 1.01325
5067                      If ProbSP.ComboBox1.ListIndex > 0 Then
5068                          For i = 1 To ProbSP.ComboBox1.ListIndex
5069                              Inpt.Cells(2 + i, 11) = (ProbSP.TextBox2.Value + stpPress * i)
/ 14.696006 * 1.01325
5070                          Next i
5071                      End If
5072                  ElseIf ProbSP.ComboBox3.ListIndex = 3 Then
5073                      Inpt.Cells(2, 11) = (ProbSP.TextBox2.Value) / 760 * 1.01325
5074                      Inpt.Cells(ProbSP.ComboBox1.ListIndex + 3, 11) =
(ProbSP.TextBox1.Value) / 760 * 1.01325
```

```
5075                    If ProbSP.ComboBox1.ListIndex > 0 Then
5076                        For i = 1 To ProbSP.ComboBox1.ListIndex
5077                            Inpt.Cells(2 + i, 11) = (ProbSP.TextBox2.Value + stpPress * i)
/ 760 * 1.01325
5078                        Next i
5079                    End If
5080                End If
5081                Indx.Cells(2, 12) = ProbSP.ComboBox1.ListIndex + 2
5082            Else
5083                MsgBox ("Pressure Inputs Must follow Max/Min Format.")
5084                CaseOk = False
5085                GoTo 200
5086            End If
5087        Else
5088            For i = 1 To Parameter.Cells(11, 2)
5089                Set valuePress = ProbSP.Controls.Item("Press" & i)
5090                If valuePress.Value <> "" Then
5091                    If ProbSP.ComboBox3.ListIndex = 0 Then
5092                        Inpt.Cells(i + 1, 11) = valuePress.Value
5093                    ElseIf ProbSP.ComboBox3.ListIndex = 1 Then
5094                        Inpt.Cells(i + 1, 11) = valuePress.Value * 1.01325
5095                    ElseIf ProbSP.ComboBox3.ListIndex = 2 Then
5096                        Inpt.Cells(i + 1, 11) = valuePress.Value / 14.696006 * 1.01325
5097                    ElseIf ProbSP.ComboBox3.ListIndex = 3 Then
5098                        Inpt.Cells(i + 1, 11) = valuePress.Value / 760 * 1.01325
5099                    End If
5100                    Indx.Cells(2, 12) = i
5101                End If
5102            Next i
5103        End If
5104 'Input the Entropy
5105        If ProbSP.TextBox3.Value <> "" Then
5106            Miscr.Cells(2, 11) = ProbSP.TextBox3.Value
5107            If ProbSP.Controls.Item("Press1") = "" And ProbSP.TextBox1.Value = "" Then
5108                CaseOk = False
5109                MsgBox ("Assigned Values for Pressure Are Missing")
5110            Else
5111                CaseOk = True
5112            End If
5113        Else
5114            CaseOk = False
5115            MsgBox ("Assigned Value for Mixture Entropy Is Missing")
5116        End If
5117    Else
5118        MsgBox ("Entropy and Pressure must be numeric")
5119    End If
5120 200: End Sub
5121
5122 '*********************************************************************
5123 'ProbHP - Enthalpy and Pressure
5124 '*********************************************************************
5125 'This sets up the ProbHP Screen
5126 Sub ProbHP_Setup()
5127     Dim tempLabel As Object
5128     Dim tempValue As Object
5129 'Pressures
5130     If ProbHP.Frame2.Height < (12 + 18 * Parameter.Cells(11, 2)) Then
5131         ProbHP.Frame2.ScrollBars = fmScrollBarsVertical
5132         ProbHP.Frame2.ScrollHeight = 12 + 18 * Parameter.Cells(11, 2)
5133     End If
5134     For i = 1 To Parameter.Cells(11, 2)
5135         Set tempLabel = ProbHP.Frame2.Controls.Add("Forms.Label.1", "P" & i, True)
5136         With tempLabel
5137             .Caption = "P" & i
5138             .Height = 18
5139             .Width = 15
5140             .Left = 4
5141             .Top = 10 + 18 * (i - 1)
5142         End With
5143         Set tempValue = ProbHP.Frame2.Controls.Add("Forms.TextBox.1", "Press" & i, True)
5144         With tempValue
```

```
5145              .Height = 18
5146              .Width = 70
5147              .Left = 20
5148              .Top = 8 + 18 * (i - 1)
5149              .TextAlign = 3
5150          End With
5151      Next i
5152  'Set up the Pressure Unit ComboBox
5153      If ProbHP.ComboBox3.ListCount = 0 Then
5154          ProbHP.ComboBox3.AddItem "BAR", 0
5155          ProbHP.ComboBox3.AddItem "ATM", 1
5156          ProbHP.ComboBox3.AddItem "PSI", 2
5157          ProbHP.ComboBox3.AddItem "mmH", 3
5158      End If
5159      ProbHP.ComboBox3.ListIndex = 0
5160  'Set up the Intervals ComboBoxs
5161      If ProbHP.ComboBox1.ListCount = 0 Then
5162          For i = 1 To Parameter.Cells(11, 2) - 1
5163              ProbHP.ComboBox1.AddItem i, (i - 1)
5164          Next i
5165          ProbHP.ComboBox1.ListIndex = 0
5166      End If
5167  End Sub
5168  'This saves the Problem Enthaply and Pressure
5169  Sub InputHP()
5170      Dim stpPress As Double
5171      Dim valuePress As Object
5172      CaseOk = True
5173  'Data validation
5174      If ProbHP.TextBox3 <> "" And IsNumeric(ProbHP.TextBox3) = False Then CaseOk = False
5175      If ProbHP.TextBox1 <> "" And ProbHP.TextBox2 <> "" Then
5176          If IsNumeric(ProbHP.TextBox1) = False Or IsNumeric(ProbHP.TextBox2) = False Then
CaseOk = False
5177      End If
5178      For i = 1 To Parameter.Cells(11, 2)
5179          Set valuePress = ProbHP.Controls.Item("Press" & i)
5180          If valuePress.Value <> "" And IsNumeric(valuePress.Value) = False Then CaseOk =
False
5181      Next i
5182      If CaseOk Then
5183  'Input the Pressures from ProbHP
5184          If ProbHP.TextBox1 <> "" And ProbHP.TextBox2 <> "" And ProbHP.ComboBox1.ListIndex
>= 0 Then
5185              If (ProbHP.TextBox1.Value - ProbHP.TextBox2.Value) > 0 Then
5186                  stpPress = (ProbHP.TextBox1.Value - ProbHP.TextBox2.Value) /
ProbHP.ComboBox1.Value
5187                  If ProbHP.ComboBox3.ListIndex = 0 Then
5188                      Inpt.Cells(2, 11) = ProbHP.TextBox2.Value
5189                      Inpt.Cells(ProbHP.ComboBox1.ListIndex + 3, 11) = ProbHP.TextBox1.Value
5190                      If ProbHP.ComboBox1.ListIndex > 0 Then
5191                          For i = 1 To ProbHP.ComboBox1.ListIndex
5192                              Inpt.Cells(2 + i, 11) = (ProbHP.TextBox2.Value + stpPress * i)
5193                          Next i
5194                      End If
5195                  ElseIf ProbHP.ComboBox3.ListIndex = 1 Then
5196                      Inpt.Cells(2, 11) = ProbHP.TextBox2.Value * 1.01325
5197                      Inpt.Cells(ProbHP.ComboBox1.ListIndex + 3, 11) = ProbHP.TextBox1.Value
* 1.01325
5198                      If ProbHP.ComboBox1.ListIndex > 0 Then
5199                          For i = 1 To ProbHP.ComboBox1.ListIndex
5200                              Inpt.Cells(2 + i, 11) = (ProbHP.TextBox2.Value + stpPress * i)
* 1.01325
5201                          Next i
5202                      End If
5203                  ElseIf ProbHP.ComboBox3.ListIndex = 2 Then
5204                      Inpt.Cells(2, 11) = (ProbHP.TextBox2.Value) / 14.696006 * 1.01325
5205                      Inpt.Cells(ProbHP.ComboBox1.ListIndex + 3, 11) =
(ProbHP.TextBox1.Value) / 14.696006 * 1.01325
5206                      If ProbHP.ComboBox1.ListIndex > 0 Then
5207                          For i = 1 To ProbHP.ComboBox1.ListIndex
```

```
                            Inpt.Cells(2 + i, 11) = (ProbHP.TextBox2.Value + stpPress * i)
/ 14.696006 * 1.01325
                        Next i
                    End If
                ElseIf ProbHP.ComboBox3.ListIndex = 3 Then
                    Inpt.Cells(2, 11) = (ProbHP.TextBox2.Value) / 760 * 1.01325
                    Inpt.Cells(ProbHP.ComboBox1.ListIndex + 3, 11) =
(ProbHP.TextBox1.Value) / 760 * 1.01325
                    If ProbHP.ComboBox1.ListIndex > 0 Then
                        For i = 1 To ProbHP.ComboBox1.ListIndex
                            Inpt.Cells(2 + i, 11) = (ProbHP.TextBox2.Value + stpPress * i)
/ 760 * 1.01325
                        Next i
                    End If
                End If
                Indx.Cells(2, 12) = ProbHP.ComboBox1.ListIndex + 2
            Else
                MsgBox ("Pressure Inputs Must follow Max/Min Format.")
                CaseOk = False
                GoTo 200
            End If
        Else
            For i = 1 To Parameter.Cells(11, 2)
                Set valuePress = ProbHP.Controls.Item("Press" & i)
                If valuePress.Value <> "" Then
                    If ProbHP.ComboBox3.ListIndex = 0 Then
                        Inpt.Cells(i + 1, 11) = valuePress.Value
                    ElseIf ProbHP.ComboBox3.ListIndex = 1 Then
                        Inpt.Cells(i + 1, 11) = valuePress.Value * 1.01325
                    ElseIf ProbHP.ComboBox3.ListIndex = 2 Then
                        Inpt.Cells(i + 1, 11) = valuePress.Value / 14.696006 * 1.01325
                    ElseIf ProbHP.ComboBox3.ListIndex = 3 Then
                        Inpt.Cells(i + 1, 11) = valuePress.Value / 760 * 1.01325
                    End If
                    Indx.Cells(2, 12) = i
                End If
            Next i
        End If
'Input the Enthalpy
        If ProbHP.TextBox3.Value <> "" Then Miscr.Cells(5, 4) = ProbHP.TextBox3.Value
        If ProbHP.Controls.Item("Press1") = "" And ProbHP.TextBox1.Value = "" Then
            CaseOk = False
            MsgBox ("Assigned Values for Pressure Are Missing")
        Else
            CaseOk = True
        End If
    Else
        MsgBox ("Pressure and Enthalpy must be numeric")
    End If
200: End Sub

'*********************************************************************
'Prob Input End Tasks
'*********************************************************************
'This Completes clean up tasks using the given Input Data and Does Final Input Checks
Sub Prob Input End()
    Dim eratio, xyz, denmtr As Double
    If Miscl.Cells(2, 18) Then Miscr.Cells(2, 16) = 0.3125 * (100000 * Miscr.Cells(2, 2) /
(Miscr.Cells(2, 6) * Miscr.Cells(2, 1))) ^ (1 / 2)
    If Miscl.Cells(2, 15) Then Miscr.Cells(2, 21) = Miscr.Cells(2, 9) / 1000
    If (Miscl.Cells(2, 4) Or Miscl.Cells(2, 13)) Then Miscl.Cells(2, 11) = "TRUE"
    If Miscr.Cells(5, 4) > Miscr.Cells(7, 4) Then
        Miscr.Cells(2, 4) = Miscr.Cells(7, 4)
    Else
        Miscr.Cells(2, 4) = Miscr.Cells(5, 4)
    End If
    If Miscr.Cells(5, 4) > 0.9 * 10 ^ 30 Then Miscr.Cells(5, 4) = 0
    If Miscr.Cells(7, 4) > 0.9 * 10 ^ 30 Then Miscr.Cells(7, 4) = 0


    Call Output_Setup
```

```
5275      For i = 1 To Parameter.Cells(3, 2)
5276          If Miscl.Cells(i + 1, 2) Then
5277              Call Debug_Setup
5278              GoTo 100
5279          End If
5280      Next i
5281      If Indx.Cells(2, 8) > 0 Or pfrac Then Call Plot_Setup
5282  100: Call REACT Sub
5283
5284      If Indx.Cells(2, 9) = 0 Then
5285          Indx.Cells(2, 9) = 1
5286          Inpt.Cells(2, 10) = 0
5287          If Inpt.Cells(3, 8) > 0 Then
5288              Inpt.Cells(2, 10) = Inpt.Cells(2, 8) / Inpt.Cells(3, 8)
5289          Else
5290              MsgBox ("Missing Reactant Amounts")
5291              CaseOk = False
5292          End If
5293      End If
5294
5295      If chkphi Or chkeqrats Then
5296          For i = 1 To Indx.Cells(2, 9)
5297              eratio = Inpt.Cells(i + 1, 10)
5298              If chkeqrats Then
5299                  xyz = -eratio * Inpt.Cells(3, 6) - Inpt.Cells(3, 7)
5300                  denmtr = eratio * Inpt.Cells(2, 6) + Inpt.Cells(2, 7)
5301              Else
5302                  xyz = -Inpt.Cells(3, 6) - Inpt.Cells(3, 7)
5303                  denmtr = eratio * (Inpt.Cells(2, 6) + Inpt.Cells(2, 7))
5304              End If
5305              If (Abs(denmtr) < (10 ^ -30)) Then
5306                  MsgBox ("Unable to Process Equivalence Ratios")
5307                  CaseOk = False
5308              End If
5309              Inpt.Cells(i + 1, 10) = xyz / denmtr
5310          Next i
5311      End If
5312  End Sub
5313
5314
5315  '*********************************************************************
5316  'REACT_Sub - Read and Process Reactant Records.
5317  '*********************************************************************
5318
5319  Sub REACT Sub()
5320      Dim wdone(3) As Boolean
5321      Dim bb(4), dat(), t1save, t2save, dift, rm, pcwt, t1, t2, rcf(8, 2), eform As Double
5322      ReDim dat(Parameter.Cells(7, 2) + 1)
5323      Dim rcoefs As Boolean
5324      Dim m, ifrmla, nj, icf, nint, ifaz, ntgas, ntot, nall As Integer
5325      Dim name, el(4) As String
5326
5327      fuel = False
5328      For k = 1 To 2
5329          wdone(k) = False
5330          Inpt.Cells(k + 1, 8) = 0
5331          Inpt.Cells(k + 1, 5) = 0
5332          Inpt.Cells(k + 1, 7) = 0
5333          Inpt.Cells(k + 1, 6) = 0
5334          Inpt.Cells(k + 1, 4) = 0
5335          Inpt.Cells(k + 1, 12) = 0
5336          For j = 1 To Parameter.Cells(7, 2)
5337              Cdata.Cells(j + 1, 1) = ""
5338              Inpt.Cells(j + 1, k + 15) = 0
5339          Next j
5340      Next k
5341      For i = 1 To Parameter.Cells(7, 2)
5342          dat(i) = 0
5343      Next i
5344  'If Oxidant then KR = 1, If Fuel then KR = 2
5345      For n = 1 To Reactn.Cells(2, 1)
```

```
5346          t1save = 20000
5347          t2save = 0
5348          rcoefs = True
5349          If (CStr(Cdata.Cells(n + 1, 7)) = "lib" Or Reactn.Cells(n + 1, 8) = 0) Then
5350              Miscr.Cells(2, 15) = Reactn.Cells(n + 1, 7)
5351              ntgas = Indx.Cells(2, 24)
5352              ntot = Indx.Cells(2, 25)
5353              nall = Indx.Cells(2, 26)
5354              name = CStr(reac.Cells(Cdata.Cells(n + 1, 26) + 1, 1))
5355              nint = reac.Cells(Cdata.Cells(n + 1, 26) + 1, 3)
5356              For j = 0 To 4
5357                  el(j) = reac.Cells(Cdata.Cells(n + 1, 26) + 1, j + 5)
5358                  bb(j) = CDbl(reac.Cells(Cdata.Cells(n + 1, 26) + 1, j + 10))
5359              Next j
5360              ifaz = reac.Cells(Cdata.Cells(n + 1, 26) + 1, 15)
5361              t1 = reac.Cells(Cdata.Cells(n + 1, 26) + 1, 16)
5362              t2 = reac.Cells(Cdata.Cells(n + 1, 26) + 1, 17)
5363              rm = reac.Cells(Cdata.Cells(n + 1, 26) + 1, 18)
5364              If CInt(Cdata.Cells(n + 1, 26) + 1) <= ntot Then
5365                  icf = 3
5366                  If CInt(Cdata.Cells(n + 1, 26) + 1) > ntgas Then icf = 1
5367                  For i = 0 To 8
5368                      For j = 0 To icf - 1
5369                          rcf(i, j) = reac.Cells(Cdata.Cells(n + 1, 26) + 1, i + j * 9 + 20)
5370                      Next j
5371                  Next i
5372              Else
5373                  eform = reac.Cells(Cdata.Cells(n + 1, 26) + 1, 19)
5374                  If nint > 0 Then
5375                  For i = 0 To 8
5376                      For j = 0 To nint - 1
5377                          rcf(i, j) = reac.Cells(Cdata.Cells(n + 1, 26) + 1, i + j * 9 + 20)
5378                      Next j
5379                  Next i
5380                  End If
5381              End If
5382              If name = CStr(Cdata.Cells(n + 1, 11)) Or name = "*" & CStr(Cdata.Cells(n + 1,
11)) Then
5383                  If nint = 0 Then
5384                      rcoefs = False
5385                      Reactn.Cells(n + 1, 4) = eform * 1000 / Miscr.Cells(2, 9)
5386                      If Miscr.Cells(2, 15) = 0 Then
5387                          Miscr.Cells(2, 15) = t1
5388                          Reactn.Cells(n + 1, 7) = t1
5389                      End If
5390                  End If
5391                  For j = 0 To 4
5392                      If (bb(j) = 0) Then GoTo 5
5393                      Indx.Cells(n + 1, 22) = j + 1
5394                      Cdata.Cells(n + 1, j + 13) = el(j)
5395                      Reactn.Cells(n + 1, j + 8) = bb(j)
5396                  Next j
5397   5:             If Not Miscl.Cells(2, 7) And CDbl(Miscr.Cells(2, 15)) = 0 Then GoTo 50
5398                  If (rcoefs) Then
5399                      Miscr.Cells(2, 12) = Log(Miscr.Cells(2, 15))
5400                      m = 1
5401                      If ifaz <= 0 Then
5402                          If Miscr.Cells(2, 15) > Therm.Cells(3, 2) Then m = 2
5403                          If Miscr.Cells(2, 15) > Therm.Cells(4, 2) Then m = 3
5404                      End If
5405                      Reactn.Cells(n + 1, 4) = (((((rcf(6, m - 1) / 5) * Miscr.Cells(2, 15) +
rcf(5, m - 1) / 4) * Miscr.Cells(2, 15) + rcf(4, m - 1) / 3) * Miscr.Cells(2, 15) + rcf(3, m - 1)
/ 2) * Miscr.Cells(2, 15) + rcf(2, m - 1)) * Miscr.Cells(2, 15) - (rcf(0, m - 1) / Miscr.Cells(2,
15)) + rcf(1, m - 1) * Miscr.Cells(2, 12) + rcf(7, m - 1)
5406                      If (Miscl.Cells(2, 19) And ifaz <= 0) Then Reactn.Cells(n + 1, 4) =
Reactn.Cells(n + 1, 4) - Miscr.Cells(2, 15)
5407                  End If
5408              End If
5409          End If
5410   50:    ifrmla = Indx.Cells(n + 1, 22)
5411          If Cdata.Cells(n + 1, 4) = "FUEL" Then
```

```
5412             fuel = True
5413             kr = 2
5414         Else
5415             kr = 1
5416         End If
5417         For i = 1 To Parameter.Cells(7, 2)
5418             dat(i) = 0
5419         Next i
5420  'Store Atomic Symbols In Elmt Array, Calculate Molecular Weight, and
5421  'Temporarily Store Atomic Valence in X
5422         rm = 0
5423         For jj = 1 To ifrmla
5424             For j = 1 To Parameter.Cells(7, 2)
5425                 nj = j
5426                 If Cdata.Cells(j + 1, 1) = "" Then GoTo 60
5427                 If Cdata.Cells(n + 1, jj + 12) = Cdata.Cells(j + 1, 1) Then GoTo 80
5428             Next j
5429  60:        Indx.Cells(2, 7) = nj
5430             Cdata.Cells(j + 1, 1) = Cdata.Cells(n + 1, jj + 12)
5431  80:        For kk = 1 To 100
5432                 If Cdata.Cells(kk + 1, 2) = Cdata.Cells(n + 1, jj + 12) Then
5433                     rm = rm + Reactn.Cells(n + 1, jj + 7) * Inpt.Cells(kk + 1, 9)
5434                     Miscr.Cells(j + 1, 18) = Inpt.Cells(kk + 1, 9)
5435                     Miscr.Cells(j + 1, 20) = Inpt.Cells(kk + 1, 15)
5436                     dat(j) = dat(j) + Reactn.Cells(n + 1, jj + 7)
5437                     GoTo 100
5438                 End If
5439             Next kk
5440             Indx.Cells(2, 7) = 0
5441             GoTo 200
5442  100:    Next jj
5443         If CDbl(Reactn.Cells(n + 1, 5)) < 0 Then
5444             Reactn.Cells(n + 1, 5) = 0
5445             If (Not Miscl.Cells(2, 10) And Not wdone(kr)) Then
5446                 wdone(kr) = True
5447                 Reactn.Cells(n + 1, 5) = 100
5448  'OUTPUT
5449                 Call Statement_Format
5450                 out.Cells(outint, 3) = "WARNGING! AMOUNT MISSING FOR REACTANT " &
Cdata.Cells(n + 1, 11) & "PROGRAM SETS WEIGHT PERCENT = 100"
5451                 outint = outint + 2
5452             Else
5453                 Indx.Cells(2, 7) = 0
5454                 GoTo 200
5455             End If
5456         End If
5457  'Add Contributions to Wp(k), Hpp(k), Am(k), and B0p(i,k)
5458         If Reactn.Cells(n + 1, 5) > 0 Then wdone(kr) = True
5459         pcwt = Reactn.Cells(n + 1, 5)
5460         If Miscl.Cells(2, 10) Then pcwt = pcwt * rm
5461         Inpt.Cells(kr + 1, 8) = Inpt.Cells(kr + 1, 8) + pcwt
5462         If rm <= 0 Then
5463             Indx.Cells(2, 7) = 0
5464             GoTo 200
5465         Else
5466             Inpt.Cells(kr + 1, 5) = Inpt.Cells(kr + 1, 5) + Reactn.Cells(n + 1, 4) * pcwt /
rm
5467             Inpt.Cells(kr + 1, 4) = Inpt.Cells(kr + 1, 4) + pcwt / rm
5468             If Reactn.Cells(n + 1, 3) <> 0 Then
5469                 Inpt.Cells(kr + 1, 12) = Inpt.Cells(kr + 1, 12) + pcwt / Reactn.Cells(n +
1, 3)
5470             Else
5471                 Inpt.Cells(2, 12) = 0
5472                 Inpt.Cells(3, 12) = 0
5473             End If
5474             For j = 1 To Indx.Cells(2, 7)
5475                 Inpt.Cells(j + 1, kr + 15) = dat(j) * pcwt / rm + Inpt.Cells(j + 1, kr +
15)
5476             Next j
5477             Reactn.Cells(n + 1, 6) = rm
5478         End If
```

```
5479      Next n
5480  ' 100 Percent Oxidant, Switch indices
5481      If fuel = False Then
5482          For n = 1 To Reactn.Cells(2, 1)
5483              Cdata.Cells(n + 1, 4) = ""
5484          Next n
5485          Inpt.Cells(3, 8) = Inpt.Cells(2, 8)
5486          Inpt.Cells(2, 8) = 0
5487          Inpt.Cells(3, 5) = Inpt.Cells(2, 5)
5488          Inpt.Cells(3, 4) = Inpt.Cells(2, 4)
5489          Inpt.Cells(2, 4) = 0
5490          For j = 1 To Indx.Cells(2, 7)
5491              Inpt.Cells(j + 1, 17) = Inpt.Cells(j + 1, 16)
5492          Next j
5493      End If
5494   'Normalize Hpp(kr), Am(kr), B0p(i,kr), and Pecwt(n), Calculate Vpls(kr) and Vmin(kr)
5495      If Indx.Cells(2, 7) <> 0 Then
5496          For kr = 1 To 2
5497              If Inpt.Cells(kr + 1, 8) <> 0 Then
5498                  Inpt.Cells(kr + 1, 5) = Inpt.Cells(kr + 1, 5) / Inpt.Cells(kr + 1, 8)
5499                  Inpt.Cells(kr + 1, 4) = Inpt.Cells(kr + 1, 8) / Inpt.Cells(kr + 1, 4)
5500                  If Inpt.Cells(kr + 1, 12) <> 0 Then Inpt.Cells(kr + 1, 12) = Inpt.Cells(kr
+ 1, 8) / Inpt.Cells(kr + 1, 12)
5501                  For j = 1 To Indx.Cells(2, 7)
5502                      Inpt.Cells(j + 1, kr + 15) = Inpt.Cells(j + 1, kr + 15) / Inpt.Cells(kr
+ 1, 8)
5503                      If Miscr.Cells(j + 1, 20) < 0 Then Inpt.Cells(kr + 1, 6) =
Inpt.Cells(kr + 1, 6) + Inpt.Cells(j + 1, kr + 15) * Miscr.Cells(j + 1, 20)
5504                      If Miscr.Cells(j + 1, 20) > 0 Then Inpt.Cells(kr + 1, 7) =
Inpt.Cells(kr + 1, 7) + Inpt.Cells(j + 1, kr + 15) * Miscr.Cells(j + 1, 20)
5505                  Next j
5506                  If Miscl.Cells(2, 10) = False Then
5507                      For n = 1 To Reactn.Cells(2, 1)
5508                          If Cdata.Cells(n + 1, 4) <> "OXIDANT" Or kr <> 2 Then
5509                              If Cdata.Cells(n + 1, 4) = "OXIDANT" Or kr <> 1 Then
Reactn.Cells(n + 1, 5) = Reactn.Cells(n + 1, 5) / Inpt.Cells(kr + 1, 8)
5510                          End If
5511                      Next n
5512                  End If
5513              End If
5514          Next kr
5515          If Not Miscl.Cells(2, 14) Then
5516              out.Range("A" & outint, "B" & outint).Merge
5517              out.Cells(outint, 1) = "Reactant"
5518              If Miscl.Cells(2, 10) Then
5519                  out.Cells(outint, 3) = "MOLES"
5520              Else
5521                  out.Cells(outint, 3) = "WT. FRAC"
5522              End If
5523              out.Range("D" & outint, "E" & outint).Merge
5524              out.Cells(outint, 4) = "(ENERGY/R),K"
5525              out.Cells(outint, 4).HorizontalAlignment = xlCenter
5526              out.Cells(outint, 6) = "TEMP,K"
5527              out.Cells(outint, 7) = "DENSITY"
5528              out.Range("H" & outint, "J" & outint).Merge
5529              out.Cells(outint, 8) = "EXPLODED FORMULA"
5530              outint = outint + 1
5531              For n = 1 To Reactn.Cells(2, 1)
5532                  out.Range("A" & outint, "B" & outint).Merge
5533                  out.Range("D" & outint, "E" & outint).Merge
5534                  If CStr(Cdata.Cells(n + 1, 4)) = "FUEL" Then
5535                      out.Cells(outint, 1) = "F: " & CStr(Cdata.Cells(n + 1, 11))
5536                  Else
5537                      out.Cells(outint, 1) = "O: " & CStr(Cdata.Cells(n + 1, 11))
5538                  End If
5539                  out.Cells(outint, 3) = CDbl(Reactn.Cells(n + 1, 5))
5540                  out.Cells(outint, 4) = CDbl(Reactn.Cells(n + 1, 4))
5541                  out.Cells(outint, 6) = CDbl(Reactn.Cells(n + 1, 7))
5542                  out.Cells(outint, 7) = CDbl(Reactn.Cells(n + 1, 3))
5543                  For i = 1 To Indx.Cells(n + 1, 22)
5544                      out.Cells(outint, 8 + 2 * (i - 1)) = Cdata.Cells(n + 1, i + 12)
```

```
                    out.Cells(outint, 8 + 2 * (i - 1)).HorizontalAlignment = xlRight
                    out.Cells(outint, 9 + 2 * (i - 1)) = Reactn.Cells(n + 1, i + 7)
                Next i
                outint = outint + 1
            Next n
            outint = outint + 1
        End If
    End If
200: End Sub


'*************************************************************************************************
**************************************************
 'Start Problem Calculations - After Problem Input and Information Checks

'*************************************************************************************************
**************************************************
 'Search the Product List for Thermo Data for Species being Considered
 Sub SEARCH_Sub()
    Dim ne, ii, i5, i6, nall, nint, ntgas, ntot, ifaz, mrow As Integer
    Dim name, el(4) As String
    Dim b(4), t1, t2, thermo(8, 2) As Double
    Indx.Cells(2, 4) = 0
    ne = 0
    For i = 1 To Indx.Cells(2, 7)
        Indx.Cells(i + 1, 21) = 0
    Next i
    For j = 1 To Parameter.Cells(1, 2)
        Therm.Cells(j + 1, 5) = 0
        Therm.Cells(j + 1, 4) = 0
        Comp.Cells(j + 1, 5) = 0
        For i = 1 To Indx.Cells(2, 7)
            A.Cells(i, j) = 0
        Next i
    Next j
    ntgas = Indx.Cells(2, 24)
    ntot = Indx.Cells(2, 25)
    nall = Indx.Cells(2, 26)
    Indx.Cells(2, 15) = 1
    Indx.Cells(2, 4) = 1
 'Begin Loop for Reading Species Data from PRODUCTS
    For itot = 1 To ntot
        If itot > ntgas Then
            name = prod.Cells(itot + 1, 1)
            nint = prod.Cells(itot + 1, 3)
            For j = 0 To 4
                el(j) = prod.Cells(itot + 1, j + 5)
                b(j) = prod.Cells(itot + 1, j + 10)
            Next j
            Indx.Cells(Indx.Cells(2, 4) + 1, 23) = prod.Cells(itot + 1, 15)
            Therm.Cells(2, Indx.Cells(2, 4) + 16) = prod.Cells(itot + 1, 16)
            Therm.Cells(3, Indx.Cells(2, 4) + 16) = prod.Cells(itot + 1, 17)
            Therm.Cells(Indx.Cells(2, 15) + 1, 7) = prod.Cells(itot + 1, 18)
            For k = 0 To 8
                Therm.Cells(Indx.Cells(2, 4) + 1, k + 8) = prod.Cells(itot + 1, k + 19)
            Next k
        Else
            name = prod.Cells(itot + 1, 1)
            nint = prod.Cells(itot + 1, 3)
            For j = 0 To 4
                el(j) = prod.Cells(itot + 1, j + 5)
                b(j) = prod.Cells(itot + 1, j + 10)
            Next j
            ifaz = prod.Cells(itot + 1, 15)
            t1 = prod.Cells(itot + 1, 16)
            t2 = prod.Cells(itot + 1, 17)
            Therm.Cells(Indx.Cells(2, 15) + 1, 7) = prod.Cells(itot + 1, 18)
            For k = 0 To 8
                For j = 0 To 2
                    thermo(k, j) = prod.Cells(itot + 1, k + j * 9 + 19)
                Next j
```

```
5612             Next k
5613         End If
5614         If CInt(Indx.Cells(2, 11)) <> 0 Then
5615             i = 1
5616  20:        If CStr(Cdata.Cells(i + 1, 10)) <> name And "*" & CStr(Cdata.Cells(i + 1, 10))
<> name Then
5617                 i = i + 1
5618                 If i <= CInt(Indx.Cells(2, 11)) Then GoTo 20
5619                 GoTo 200
5620             Else
5621                 If name = CStr(Cdata.Cells(Indx.Cells(2, 15), 10)) Then
5622                     Indx.Cells(2, 11) = Indx.Cells(2, 11) + 1
5623                     For k = Indx.Cells(2, 11) To i + 1 Step -1
5624                         Cdata.Cells(k + 1, 10) = Cdata.Cells(k, 10)
5625                     Next k
5626                 Else
5627                     Cdata.Cells(i + 1, 10) = Cdata.Cells(Indx.Cells(2, 15) + 1, 10)
5628                 End If
5629                 Cdata.Cells(Indx.Cells(2, 15) + 1, 10) = name
5630             End If
5631         ElseIf CInt(Indx.Cells(2, 10)) <> 0 Then
5632             For i = 1 To Indx.Cells(2, 10)
5633                 If CStr(Cdata.Cells(i + 1, 8)) = name Or "*" & CStr(Cdata.Cells(i + 1, 8))
= name Then GoTo 200
5634             Next i
5635         End If
5636         For k = 0 To 4
5637             If b(k) = 0 Then GoTo 100
5638             For i = 1 To Indx.Cells(2, 7)
5639                 If CStr(Cdata.Cells(i + 1, 1)) = el(k) Then
5640                     A.Cells(i, Indx.Cells(2, 15)) = b(k)
5641                     GoTo 50
5642                 End If
5643             Next i
5644             For j = 1 To Indx.Cells(2, 7)
5645                 A.Cells(j, Indx.Cells(2, 15)) = 0
5646             Next j
5647             GoTo 200
5648  50:    Next k
5649  100:   Cdata.Cells(Indx.Cells(2, 15) + 1, 10) = name
5650         If itot > ntgas Then
5651             Indx.Cells(2, 4) = Indx.Cells(2, 4) + 1
5652             If CInt(Indx.Cells(2, 4)) > CInt(Parameter.Cells(2, 2)) Then GoTo 400
5653         Else
5654             Indx.Cells(2, 5) = Indx.Cells(2, 15)
5655             If CInt(Indx.Cells(2, 5)) > CInt(Parameter.Cells(8, 2)) Then GoTo 400
5656             For i = 1 To 3
5657                 For j = 1 To 9
5658                     Coef.Cells(Indx.Cells(2, 5) + 1, j + 9 * (i - 1)) = thermo(j - 1, i -
1)
5659                 Next j
5660             Next i
5661  'If Species is an atomic gas, store index in Jx
5662             If b(1) = 0 And b(0) = 1 Then
5663                 For i = 1 To Indx.Cells(2, 7)
5664                     If CStr(Cdata.Cells(i + 1, 1)) = el(0) Then
5665                         ne = ne + 1
5666                         Indx.Cells(i + 1, 21) = Indx.Cells(2, 15)
5667                         Misci.Cells(i + 1, 8) = Indx.Cells(2, 15)
5668                         GoTo 150
5669                     End If
5670                 Next i
5671             End If
5672         End If
5673  150:   Indx.Cells(2, 15) = Indx.Cells(2, 15) + 1
5674         If CInt(Indx.Cells(2, 15)) > CInt(Parameter.Cells(1, 2)) Then GoTo 400
5675  200: Next itot
5676  'Finished Reading Thermo Data from PRODUCTS
5677      Indx.Cells(Indx.Cells(2, 4) + 1, 23) = 0
5678      Indx.Cells(2, 4) = Indx.Cells(2, 4) - 1
5679      Indx.Cells(2, 15) = Indx.Cells(2, 15) - 1
```

```
5680      Indx.Cells(2, 6) = Indx.Cells(2, 5) + 1
5681  'Find Missing Elements (If any) for components
5682      Indx.Cells(2, 18) = Indx.Cells(2, 15)
5683      If ne < CInt(Indx.Cells(2, 7)) Then
5684          For i = 1 To Indx.Cells(2, 7)
5685              If CInt(Indx.Cells(2, 18)) > CInt(Parameter.Cells(1, 2)) Then GoTo 400
5686              If Indx.Cells(i + 1, 21) = 0 Then
5687                  Indx.Cells(2, 18) = Indx.Cells(2, 18) + 1
5688                  For k = 1 To Indx.Cells(2, 7)
5689                      A.Cells(k, Indx.Cells(2, 18)) = 0
5690                  Next k
5691                  A.Cells(i, Indx.Cells(2, 18)) = 1
5692                  Cdata.Cells(Indx.Cells(2, 18) + 1, 10) = Cdata.Cells(i + 1, 1)
5693                  For k = 1 To 100
5694                      If Cdata.Cells(i + 1, 1) = Cdata.Cells(k + 1, 2) Then
5695                          Therm.Cells(Indx.Cells(2, 18) + 1, 7) = Inpt.Cells(k + 1, 9)
5696                          Miscr.Cells(i + 1, 18) = Inpt.Cells(k + 1, 9)
5697                          Therm.Cells(Indx.Cells(2, 18) + 1, 3) = 2.5
5698                          GoTo 210
5699                      End If
5700                  Next k
5701  210:            Indx.Cells(i + 1, 21) = Indx.Cells(2, 18)
5702                  Misci.Cells(i + 1, 8) = Indx.Cells(2, 18)
5703              End If
5704          Next i
5705      End If
5706  'Are All Elements in Product Species?
5707      For i = 1 To Indx.Cells(2, 7)
5708          For j = 1 To Indx.Cells(2, 15)
5709              If CDbl(A.Cells(i, j)) <> 0 Then GoTo 300
5710              ii = i
5711          Next j
5712  'OUTPUT
5713          Call Statement_Format
5714          out.Cells(outint, 3) = "Product Species Containing the Element " & Cdata.Cells(ii +
1, 1) & " Missing"
5715          outint = outint + 2
5716          Indx.Cells(2, 15) = 0
5717          GoTo 600
5718  300: Next i
5719  'OUTPUT
5720      If Not Miscl.Cells(2, 14) Then
5721          out.Range("A" & outint, "N" & (outint + 1)).Merge
5722          out.Cells(outint, 1) = "SPECIES BEING CONSIDERED IN THIS SYSTEM (CONDENSED PHASE
MAY HAVE NAME LISTED SEVERAL TIMES)"
5723          out.Cells(outint, 1).HorizontalAlignment = xlCenter
5724          out.Cells(outint + 1, 1) = "THERMO PROPERTIES LAST UPDATED " & ther.Cells(3, 6)
5725          out.Cells(outint + 1, 1).HorizontalAlignment = xlCenter
5726          outint = outint + 3
5727          i5 = 0
5728          i6 = 0
5729          For i = 1 To Indx.Cells(2, 15)
5730              i6 = i6 + 1
5731              out.Cells(outint + i5, i6) = CStr(Cdata.Cells(i + 1, 10))
5732              If i6 = CInt(Parameter.Cells(3, 2)) + 1 And i <> CInt(Indx.Cells(2, 15)) Then
5733                  i6 = 0
5734                  i5 = i5 + 1
5735              End If
5736          Next i
5737          outint = outint + i5 + 2
5738      End If
5739      GoTo 600
5740  'OUTPUT
5741  400: Call Statement Format
5742      out.Cells(outint, 3) = "Insufficient Storage for Products"
5743      outint = outint + 1
5744      Indx.Cells(2, 15) = 0
5745      GoTo 600
5746  600: End Sub
5747
5748  Sub Calc_Start()
```

```
5749      Dim j As Integer
5750      Dim xi, xln As Double
5751      Dim calcrow As Integer
5752
5753      If Miscl.Cells(2, 8) Then
5754          If Cdata.Cells(Indx.Cells(2, 7) + 1, 1) <> "E" Then
5755              Indx.Cells(2, 7) = Indx.Cells(2, 7) + 1
5756              Cdata.Cells(Indx.Cells(2, 7) + 1, 1) = "E"
5757              Inpt.Cells(Indx.Cells(2, 7) + 1, 16) = 0
5758              Inpt.Cells(Indx.Cells(2, 7) + 1, 17) = 0
5759          End If
5760      ElseIf CStr(Cdata.Cells(Indx.Cells(2, 7) + 1, 1)) = "E" Then
5761          Indx.Cells(2, 7) = Indx.Cells(2, 7) - 1
5762      End If
5763      For n = 1 To Reactn.Cells(2, 1)
5764          Reactn.Cells(n + 1, 2) = 0
5765      Next n
5766      Call SEARCH_Sub
5767      If Indx.Cells(2, 15) = 0 Then GoTo 300
5768      Miscl.Cells(2, 11) = "FALSE"
5769      If pfrac Then
5770          Call Form Position(PlotProd)
5771          PlotProd.Show
5772      End If
5773  'Initial Estimates
5774      Indx.Cells(2, 13) = 0
5775      Miscl.Cells(2, 6) = "TRUE"
5776      Comp.Cells(2, 1) = 0.1
5777      Comp.Cells(2, 2) = -2.3025851
5778      Comp.Cells(2, 4) = Comp.Cells(2, 1)
5779      xi = Indx.Cells(2, 5)
5780      If xi = 0 Then xi = 1
5781      xi = Comp.Cells(2, 1) / xi
5782      xln = Log(xi)
5783      For inc = 1 To Indx.Cells(2, 4)
5784          j = CInt(Indx.Cells(2, 5)) + inc
5785          En.Cells(j, 1) = 0
5786          Comp.Cells(j + 1, 6) = 0
5787      Next inc
5788      For j = 1 To Indx.Cells(2, 5)
5789          En.Cells(j, 1) = xi
5790          Comp.Cells(j + 1, 6) = xln
5791      Next j
5792      If CInt(Indx.Cells(2, 4)) <> 0 And CInt(Indx.Cells(2, 16)) <> 0 Then
5793          For i = 1 To Indx.Cells(2, 16)
5794              For j = Indx.Cells(2, 15) To Indx.Cells(2, 6) Step -1
5795                  If CStr(Cdata.Cells(j + 1, 10)) = CStr(Cdata.Cells(i + 1, 25)) Then
5796                      Indx.Cells(2, 13) = Indx.Cells(2, 13) + 1
5797                      Indx.Cells(Indx.Cells(2, 13) + 1, 20) = j
5798  'OUTPUT
5799                      If Not Miscl.Cells(2, 14) Then
5800                          Call Statement Format
5801                          out.Cells(outint, 3) = Cdata.Cells(j + 1, 1) & " INSERTED"
5802                          outint = outint + 2
5803                      End If
5804                      GoTo 120
5805                  End If
5806              Next j
5807  120:    Next i
5808      End If
5809      If (Miscl.Cells(2, 17) Or Miscl.Cells(2, 7) Or Miscl.Cells(2, 16)) Then Call THERMP_Sub
5810      If Indx.Cells(2, 14) > 0 Then
5811      End If
5812      OUTPUT.Worksheets("OUTPUT").Activate
5813  300: End
5814  End Sub
5815
5816
'*************************************************************************************************
**************************************************
5817  'PLOTPROD - SELECT WHICH PRODUCTS TO PLOT OUTPUT - IF PLOTTING SELECTED
```

```
5818
'*************************************************************************************************
************************************************
5819
5820  'This pulls which products are being considered for the system
5821  Sub PlotProd_Setup()
5822      If Miscl.Cells(2, 10) Then
5823          PlotProd.Label1.Caption = "Select which product(s) to plot: " & vbNewLine & "Unit =
MOLES"
5824      Else
5825          PlotProd.Label1.Caption = "Select which product(s) to plot: " & vbNewLine & "Unit =
WT FRACTION"
5826      End If
5827      If PlotProd.ListBox1.ListCount = 0 Then
5828          For i = 1 To Indx.Cells(2, 15)
5829              If (Cdata.Cells(i + 1, 10) <> Cdata.Cells(i, 10)) Then
PlotProd.ListBox1.AddItem Cdata.Cells(i + 1, 10), i - 1
5830          Next i
5831      End If
5832  End Sub
5833
5834  'This Saves which Products were selected to plot output
5835  Sub Input_PlotProd()
5836      For i = 1 To PlotProd.ListBox1.ListCount
5837          If PlotProd.ListBox1.Selected(i - 1) = True Then
5838              Indx.Cells(2, 8) = Indx.Cells(2, 8) + 1
5839              Cdata.Cells(Indx.Cells(2, 8) + 1, 9) = PlotProd.ListBox1.List(i - 1)
5840          End If
5841      Next i
5842  End Sub
5843
5844
'*************************************************************************************************
************************************************
5845  'OUTPUT SUBROUTINES
5846
'*************************************************************************************************
************************************************
5847
5848  'This Sets up a Separate Excel Document to load the problem output
5849  Sub Output_Setup()
5850  Application.ScreenUpdating = False
5851      Dim mrow As Integer
5852      Set OUTPUT = Workbooks.Add
5853      Set out = OUTPUT.Worksheets("Sheet1")
5854      out.name = "OUTPUT"
5855      outint = 0
5856      For i = 1 To 5
5857          out.Range("A" & i, "N" & i).Merge
5858          out.Cells(i, 1).HorizontalAlignment = xlCenter
5859      Next i
5860      out.Cells(1, 1) =
"*************************************************************************************************
***************************"
5861      out.Cells(2, 1) = "CHEMICAL EQUILIBRIUM X PROGRAM MARCH 1ST, 2016"
5862      out.Cells(3, 1) = "BY JAKE D. RUMEL"
5863      out.Cells(4, 1) = "BASED ON THE NASA-GLENN CHEMICAL EQUILIBRIUM PROGRAM CEA2, MAY 21,
2004"
5864      out.Cells(5, 1) =
"*************************************************************************************************
***************************"
5865
5866      out.Cells(7, 1) = "Case:"
5867      out.Range("C7", "G7").Merge
5868      out.Cells(7, 3) = ProbInput.TextBox3.Value
5869      out.Range("A8", "B8").Merge
5870      out.Cells(8, 1) = "Problem Type:"
5871      out.Range("C8", "G8").Merge
5872      out.Cells(8, 3) = ProbInput.ComboBox1.Value
5873
5874      If Not Miscl.Cells(2, 14) Then
```

```
5875          out.Cells(10, 1) = "Options:"
5876          out.Range("A11", "N12").HorizontalAlignment = xlCenter
5877          out.Cells(11, 3) = "TP"
5878          out.Cells(12, 3) = Miscl.Cells(2, 17)
5879          out.Cells(11, 4) = "HP"
5880          If Miscl.Cells(2, 7) And Not Miscl.Cells(2, 19) Then
5881              out.Cells(12, 4) = "TRUE"
5882          Else
5883              out.Cells(12, 4) = "FALSE"
5884          End If
5885          out.Cells(11, 5) = "SP"
5886          out.Cells(12, 5) = Miscl.Cells(2, 16)
5887          out.Cells(11, 6) = "TV"
5888          If Miscl.Cells(2, 17) And Miscl.Cells(2, 19) Then
5889              out.Cells(12, 6) = "TRUE"
5890          Else
5891              out.Cells(12, 6) = "FALSE"
5892          End If
5893          out.Cells(11, 7) = "UV"
5894          If Miscl.Cells(2, 7) And Miscl.Cells(2, 19) Then
5895              out.Cells(12, 7) = "TRUE"
5896          Else
5897              out.Cells(12, 7) = "FALSE"
5898          End If
5899          out.Cells(11, 8) = "SV"
5900          If Miscl.Cells(2, 16) And Miscl.Cells(2, 19) Then
5901              out.Cells(12, 8) = "TRUE"
5902          Else
5903              out.Cells(12, 8) = "FALSE"
5904          End If
5905          out.Cells(11, 9) = "IONS"
5906          out.Cells(12, 9) = Miscl.Cells(2, 8)
5907          out.Cells(11, 10) = "SIUNIT"
5908          out.Cells(12, 10) = Miscl.Cells(2, 15)
5909          out.Cells(11, 11) = "DEBUG"
5910          out.Cells(12, 11) = "FALSE"
5911          For i = 2 To Parameter.Cells(3, 2)
5912              If Miscl.Cells(i, 2) Then out.Cells(12, 11) = "TRUE"
5913          Next i
5914          out.Cells(11, 12) = "TRNSPT"
5915          out.Cells(12, 12) = Miscl.Cells(2, 18)
5916          outint = 14
5917          If CDbl(Inpt.Cells(2, 13)) > 0 Then
5918              out.Cells(outint, 1) = "T,K ="
5919              mrow = 0
5920              For i = 1 To Indx.Cells(2, 19)
5921                  out.Cells(outint, i - mrow * CInt(Parameter.Cells(3, 2)) + 1) =
CDbl(Inpt.Cells(i + 1, 13))
5922                  If i = CInt(Parameter.Cells(3, 2)) Then
5923                      outint = outint + 1
5924                      mrow = mrow + 1
5925                  End If
5926              Next i
5927              outint = outint + 2
5928          End If
5929          out.Cells(outint, 1) = "Trace:"
5930          out.Cells(outint, 2) = Miscr.Cells(2, 14)
5931          outint = outint + 2
5932          out.Cells(outint, 3).HorizontalAlignment = xlCenter
5933          out.Cells(outint, 3) = "S/R"
5934          out.Cells(outint + 1, 3).HorizontalAlignment = xlCenter
5935          out.Cells(outint + 1, 3) = CDbl(Miscr.Cells(2, 11))
5936          out.Cells(outint, 6).HorizontalAlignment = xlCenter
5937          out.Cells(outint, 6) = "H/R"
5938          out.Cells(outint + 1, 6).HorizontalAlignment = xlCenter
5939          out.Cells(outint + 1, 6) = CDbl(Miscr.Cells(5, 4))
5940          out.Cells(outint, 9).HorizontalAlignment = xlCenter
5941          out.Cells(outint, 9) = "U/R"
5942          out.Cells(outint + 1, 9).HorizontalAlignment = xlCenter
5943          out.Cells(outint + 1, 9) = CDbl(Miscr.Cells(7, 4))
5944          outint = outint + 3
```

```
5945          If CInt(Indx.Cells(2, 12)) > 0 And Miscl.Cells(2, 19) Then
5946              out.Range("A" & outint, "C" & outint).Merge
5947              out.Cells(outint, 1) = "Specific Volume, M^3/KG:"
5948              mrow = 0
5949              For i = 1 To Indx.Cells(2, 12)
5950                  out.Cells(outint, i - mrow * CInt(Parameter.Cells(3, 2)) + 3) =
CDbl(Inpt.Cells(i + 1, 14) * (10 ^ (-5)))
5951                  If i = CInt(Parameter.Cells(3, 2)) Then
5952                      outint = outint + 1
5953                      mrow = mrow + 1
5954                  End If
5955              Next i
5956              outint = outint + 2
5957          End If
5958          If Not Miscl.Cells(2, 19) Then
5959              out.Cells(outint, 1) = "P,Bar:"
5960              mrow = 0
5961              For i = 1 To Indx.Cells(2, 12)
5962                  out.Cells(outint, i - mrow * CInt(Parameter.Cells(3, 2)) + 1) =
CDbl(Inpt.Cells(i + 1, 11))
5963                  If i = CInt(Parameter.Cells(3, 2)) Then
5964                      outint = outint + 1
5965                      mrow = mrow + 1
5966                  End If
5967              Next i
5968              outint = outint + 2
5969          End If
5970      Else
5971          outint = 10
5972      End If
5973  End Sub
5974  'This Sets up a Separate Excel Worksheet if Debugging Output is Requested
5975  Sub Debug Setup()
5976      Set deb = OUTPUT.Worksheets.Add(After:=OUTPUT.Worksheets("OUTPUT"))
5977      deb.name = "DEBUG"
5978      debint = 0
5979      For i = 1 To 5
5980          deb.Range("A" & i, "P" & i).Merge
5981          deb.Cells(i, 1).HorizontalAlignment = xlCenter
5982      Next i
5983      deb.Cells(1, 1) =
"*****************************************************************************************
****************************************"
5984      deb.Cells(2, 1) = "CHEMICAL EQUILIBRIUM X PROGRAM MARCH 1ST, 2016 - DEBUGGING"
5985      deb.Cells(3, 1) = "BY JAKE D. RUMEL"
5986      deb.Cells(4, 1) = "BASED ON THE NASA-GLENN CHEMICAL EQUILIBRIUM PROGRAM CEA2, MAY 21,
2004"
5987      deb.Cells(5, 1) =
"*****************************************************************************************
****************************************"
5988
5989      deb.Cells(7, 1) = "Case:"
5990      deb.Range("C7", "G7").Merge
5991      deb.Cells(7, 3) = ProbInput.TextBox3.Value
5992      deb.Range("A8", "B8").Merge
5993      deb.Cells(8, 1) = "Problem Type:"
5994      deb.Range("C8", "G8").Merge
5995      deb.Cells(8, 3) = ProbInput.ComboBox1.Value
5996
5997      debint = 10
5998
5999  End Sub
6000
6001  'This Sets up a Separate Excel Worksheet if Plotting is Requested
6002  Sub Plot_Setup()
6003      Set plt = OUTPUT.Worksheets.Add(After:=OUTPUT.Worksheets("OUTPUT"))
6004      plt.name = "PLOT"
6005      For i = 1 To 5
6006          plt.Range("A" & i, "P" & i).Merge
6007          plt.Cells(i, 1).HorizontalAlignment = xlCenter
6008      Next i
```

```
6009      plt.Cells(1, 1) =
"*****************************************************************************************
*****************************************"
6010      plt.Cells(2, 1) = "CHEMICAL EQUILIBRIUM X PROGRAM MARCH 1ST, 2016 - PLOT"
6011      plt.Cells(3, 1) = "BY JAKE D. RUMEL"
6012      plt.Cells(4, 1) = "BASED ON THE NASA-GLENN CHEMICAL EQUILIBRIUM PROGRAM CEA2, MAY 21,
2004"
6013      plt.Cells(5, 1) =
"*****************************************************************************************
*****************************************"
6014
6015      plt.Cells(7, 1) = "Case:"
6016      plt.Range("C7", "G7").Merge
6017      plt.Cells(7, 3) = ProbInput.TextBox3.Value
6018      plt.Range("A8", "B8").Merge
6019      plt.Cells(8, 1) = "Problem Type:"
6020      plt.Range("C8", "G8").Merge
6021      plt.Cells(8, 3) = ProbInput.ComboBox1.Value
6022
6023  End Sub
6024
6025  'This Manages the Output Subroutines
6026  Sub Problem_Output()
6027      Call OUT1_Sub
6028      out.Range("A" & outint, "D" & outint).Merge
6029      out.Cells(outint, 1) = "THERMODYNAMIC PROPERTIES"
6030      out.Cells(outint, 1).HorizontalAlignment = xlCenter
6031      outint = outint + 2
6032      Call OUT2_Sub
6033      Call OUT3_Sub
6034  Application.ScreenUpdating = True
6035  End Sub
6036
6037  'Writes Reactant and Fuel-Oxidant Ratio Information.
6038  Sub OUT1_Sub()
6039      Dim tem, rho As Double
6040      Call Reac_Table_Format
6041      out.Cells(outint, 2) = "REACTANT"
6042      If Miscl.Cells(2, 10) Then
6043          out.Cells(outint, 4) = "MOLES"
6044      Else
6045          out.Cells(outint, 4) = "WT FRACTION"
6046      End If
6047      out.Cells(outint, 6) = "ENERGY"
6048      out.Cells(outint, 8) = "TEMP"
6049      outint = outint + 1
6050      Call Reac_Table_Format
6051      If Not Miscl.Cells(2, 10) Then out.Cells(outint, 4) = "(SEE NOTE)"
6052      If Not Miscl.Cells(2, 15) Then
6053          out.Cells(outint, 6) = "CAL/MOL"
6054      Else
6055          out.Cells(outint, 6) = "KJ/KG-MOL"
6056      End If
6057      out.Cells(outint, 8) = "K"
6058      outint = outint + 1
6059      For n = 1 To Reactn.Cells(2, 1)
6060          Call Reac_Table_Format
6061          out.Cells(outint, 1) = Cdata.Cells(n + 1, 4)
6062          out.Cells(outint, 2) = Cdata.Cells(n + 1, 11)
6063          out.Cells(outint, 4) = CDbl(Reactn.Cells(n + 1, 5))
6064          out.Cells(outint, 6) = CDbl(Reactn.Cells(n + 1, 4) * Miscr.Cells(2, 21))
6065          out.Cells(outint, 8) = CDbl(Reactn.Cells(n + 1, 7))
6066          outint = outint + 1
6067      Next n
6068      outint = outint + 1
6069      phi = 0
6070      tem = (Inpt.Cells(2, 7) + Inpt.Cells(2, 6)) * Miscr.Cells(2, 5)
6071      If Abs(tem) >= 0.001 Then phi = -(Inpt.Cells(3, 6) + Inpt.Cells(3, 7)) / tem
6072      pfuel = 100 / (1 + Miscr.Cells(2, 5))
6073      If (CDbl(Inpt.Cells(2, 12)) <> 0 Or CDbl(Inpt.Cells(3, 12)) <> 0) Then
6074          If (CDbl(Inpt.Cells(2, 12)) = 0 Or CDbl(Inpt.Cells(3, 12)) = 0) Then
```

```
6075              If (CDbl(Inpt.Cells(2, 12)) > CDbl(Inpt.Cells(3, 12))) Then
6076                  rho = CDbl(Inpt.Cells(2, 12))
6077              Else
6078                  rho = CDbl(Inpt.Cells(3, 12))
6079              End If
6080          Else
6081              rho = (Miscr.Cells(2, 5) + 1) * Inpt.Cells(2, 12) * Inpt.Cells(3, 12) /
(Inpt.Cells(2, 12) + Miscr.Cells(2, 5) * Inpt.Cells(3, 12))
6082          End If
6083          out.Range("A" & outint, "B" & outint).Merge
6084          out.Cells(outint, 1) = "REACTANT DENSITY="
6085          If Miscl.Cells(2, 15) Then
6086              rho = rho * 1000
6087              out.Cells(outint, 3) = rho
6088              out.Cells(outint, 4) = "KG/CU M"
6089          Else
6090              out.Cells(outint, 3) = rho
6091              out.Cells(outint, 4) = "G/CC"
6092          End If
6093          outint = outint + 1
6094      End If
6095      out.Cells(outint, 1) = "O/F="
6096      out.Cells(outint, 2) = CDbl(Miscr.Cells(2, 5))
6097      out.Cells(outint, 4) = "%FUEL"
6098      out.Cells(outint, 5) = pfuel
6099      out.Range("G" & outint, "H" & outint).Merge
6100      out.Cells(outint, 7) = "R,EQ.RATIO="
6101      out.Cells(outint, 9) = CDbl(Miscr.Cells(2, 3))
6102      out.Range("K" & outint, "L" & outint).Merge
6103      out.Cells(outint, 11) = "PHI,EQ.RATIO="
6104      out.Cells(outint, 13) = phi
6105      outint = outint + 2
6106  End Sub
6107
6108  'Writes Thermodynamic Properties
6109  Sub OUT2_Sub()
6110      Dim fp, frh, fh, fu, fgi, fs, fc As String
6111      Dim pfactor, vnum, mxx() As Double
6112      Dim mp, mt, mrho, mh, mie, mg, ms, mm, mmw, mcp, mgam, mson, mof, mpf, mph, mfa, meq As
Integer
6113
6114      ione = 0
6115  'SET MXX ARRAY FOR PLOTTING PARAMETERS
6116      For i = 1 To 24
6117  '        mxx(i) = 0
6118      Next i
6119      For i = 1 To Indx.Cells(2, 8)
6120      If Cdata.Cells(i + 1, 9) = "p" Then
6121          mp = i
6122          plt.Cells(10, mp) = "P"
6123      ElseIf Cdata.Cells(i + 1, 9) = "t" Then
6124          mt = i
6125          plt.Cells(10, mt) = "T"
6126      ElseIf Cdata.Cells(i + 1, 9) = "rho" Then
6127          mrho = i
6128          plt.Cells(10, mrho) = "RHO"
6129      ElseIf Cdata.Cells(i + 1, 9) = "h" Then
6130          mh = i
6131          plt.Cells(10, mh) = "H"
6132      ElseIf Cdata.Cells(i + 1, 9) = "u" Then
6133          mie = i
6134          plt.Cells(10, mie) = "Int E"
6135      ElseIf Cdata.Cells(i + 1, 9) = "g" Then
6136          mg = i
6137          plt.Cells(10, mg) = "G"
6138      ElseIf Cdata.Cells(i + 1, 9) = "s" Then
6139          ms = i
6140          plt.Cells(10, ms) = "S"
6141      ElseIf Cdata.Cells(i + 1, 9) = "m" Then
6142          mm = i
6143          plt.Cells(10, mm) = "MM"
```

```
6144      ElseIf Cdata.Cells(i + 1, 9) = "mw" Then
6145          If Not Miscl.Cells(2, 6) Then
6146              mmw = i
6147              plt.Cells(10, mmw) = "MMW"
6148          Else
6149              mm = i
6150              plt.Cells(10, mm) = "MM"
6151          End If
6152      ElseIf Cdata.Cells(i + 1, 9) = "cp" Then
6153          mcp = i
6154          plt.Cells(10, mcp) = "Cp"
6155      ElseIf Cdata.Cells(i + 1, 9) = "gam" Then
6156          mgam = i
6157          plt.Cells(10, mgam) = "GAM"
6158      ElseIf Cdata.Cells(i + 1, 9) = "son" Then
6159          mson = i
6160          plt.Cells(10, mson) = "SON"
6161      ElseIf Cdata.Cells(i + 1, 9) = "%f" Then
6162          mpf = i
6163          plt.Cells(10, mpf) = "%F"
6164      ElseIf Cdata.Cells(i + 1, 9) = "f/a" Then
6165          mfa = i
6166          plt.Cells(10, mfa) = "F/A"
6167      ElseIf Cdata.Cells(i + 1, 9) = "o/f" Then
6168          mof = i
6169          plt.Cells(10, mof) = "O/F"
6170      ElseIf Cdata.Cells(i + 1, 9) = "phi" Then
6171          mph = i
6172          plt.Cells(10, mph) = "PHI"
6173      ElseIf Cdata.Cells(i + 1, 9) = "r" Then
6174          meq = i
6175          plt.Cells(10, meq) = "R"
6176      End If
6177
6178  100: Next i
6179      For i = CInt(Indx.Cells(2, 2)) + 1 To CInt(Indx.Cells(2, 2) + Indx.Cells(2, 14))
6180          If mof > 0 Then plt.Cells(i + 10, mof) = CDbl(Miscr.Cells(2, 5))
6181          If mpf > 0 Then plt.Cells(i + 10, mpf) = pfuel
6182          If mph > 0 Then plt.Cells(i + 10, mph) = phi
6183          If mfa > 0 And CDbl(Miscr.Cells(2, 5)) <> 0 Then plt.Cells(i + 10, mfa) = 1 /
CDbl(Miscr.Cells(2, 5))
6184          If meq > 0 Then plt.Cells(i + 10, meq) = CDbl(Miscr.Cells(2, 3))
6185      Next i
6186      If Miscl.Cells(2, 15) Then
6187          pfactor = 1
6188          fp = "P, BAR"
6189          vnum = 10 ^ 5
6190          frh = "RHO, KG/CU M"
6191          fh = "H, KJ/KG"
6192          fu = "U, KJ/KG"
6193          fgi = "G, KJ/KG"
6194          fs = "S, KJ/(KG)(K)"
6195          fc = "Cp, KJ/(KG)(K)"
6196      Else
6197          pfactor = 1 / 1.01325
6198          fp = "P, ATM"
6199          vnum = 100
6200          frh = "RHO, G/CC"
6201          fh = "H, CAL/G"
6202          fu = "U, CAL/G"
6203          fgi = "G, CAL/G"
6204          fs = "S, CAL/(G)(K)"
6205          fc = "Cp, CAL/(G)(K)"
6206      End If
6207      Cdata.Cells(5, 3) = Cdata.Cells(7, 3)
6208   'PRESSURE
6209      For i = 1 To Indx.Cells(2, 14)
6210          Miscr.Cells(i + 1, 20) = Prtout.Cells(i + 1, 6) * pfactor
6211          If CInt(Indx.Cells(2, 8)) <> 0 And i > ione Then
6212              If mp > 0 Then plt.Cells(i + Indx.Cells(2, 2) - ione + 10, mp) = Miscr.Cells(i
+ 1, 20)
```

```
6213              If mt > 0 Then plt.Cells(i + Indx.Cells(2, 2) - ione + 10, mt) = Prtout.Cells(i
+ 1, 9)
6214          End If
6215      Next i
6216      Call Thermo_Table_Format
6217      out.Cells(outint, 1) = fp
6218      For j = 1 To Indx.Cells(2, 14)
6219          out.Cells(outint, 2 + j) = Miscr.Cells(j + 1, 20)
6220      Next j
6221      outint = outint + 1
6222  'TEMPERATURE
6223      Call Thermo_Table_Format
6224      out.Cells(outint, 1) = "T, K"
6225      For j = 1 To Indx.Cells(2, 14)
6226          out.Cells(outint, 2 + j) = Prtout.Cells(j + 1, 9)
6227      Next j
6228      outint = outint + 1
6229  'DENSITY
6230      For i = 1 To Indx.Cells(2, 14)
6231          If Prtout.Cells(i + 1, 10) <> 0 Then Miscr.Cells(i + 1, 20) = vnum / Prtout.Cells(i
+ 1, 10)
6232          If Indx.Cells(2, 8) <> 0 And i > ione And mrho > 0 Then plt.Cells(i + Indx.Cells(2,
2) - ione + 10, mrho) = Miscr.Cells(i + 1, 20)
6233      Next i
6234      Call Thermo Table Format
6235      out.Cells(outint, 1) = frh
6236      For j = 1 To Indx.Cells(2, 14)
6237          out.Cells(outint, 2 + j) = Miscr.Cells(j + 1, 20)
6238      Next j
6239      outint = outint + 1
6240  'ENTHALPY
6241      For i = 1 To Indx.Cells(2, 14)
6242          Miscr.Cells(i + 1, 20) = Prtout.Cells(i + 1, 5) * Miscr.Cells(2, 21)
6243          If Indx.Cells(2, 8) <> 0 And i > ione And mh > 0 Then plt.Cells(i + Indx.Cells(2,
2) - ione + 10, mh) = Miscr.Cells(i + 1, 20)
6244      Next i
6245      Call Thermo_Table_Format
6246      out.Cells(outint, 1) = fh
6247      For j = 1 To Indx.Cells(2, 14)
6248          out.Cells(outint, 2 + j) = Miscr.Cells(j + 1, 20)
6249      Next j
6250      outint = outint + 1
6251  'INTERNAL ENERGY
6252      For i = 1 To Indx.Cells(2, 14)
6253          Miscr.Cells(i + 1, 20) = (Prtout.Cells(i + 1, 5) - Prtout.Cells(i + 1, 6) *
Prtout.Cells(i + 1, 10) / Miscr.Cells(2, 9)) * Miscr.Cells(2, 21)
6254          If Indx.Cells(2, 8) <> 0 And i > ione And mie > 0 Then plt.Cells(i + Indx.Cells(2,
2) - ione + 10, mie) = Miscr.Cells(i + 1, 20)
6255      Next i
6256      Call Thermo Table Format
6257      out.Cells(outint, 1) = fu
6258      For j = 1 To Indx.Cells(2, 14)
6259          out.Cells(outint, 2 + j) = Miscr.Cells(j + 1, 20)
6260      Next j
6261      outint = outint + 1
6262  'GIBBS ENERGY
6263      For i = 1 To Indx.Cells(2, 14)
6264          Miscr.Cells(i + 1, 20) = (Prtout.Cells(i + 1, 5) - Prtout.Cells(i + 1, 9) *
Prtout.Cells(i + 1, 7)) * Miscr.Cells(2, 21)
6265          If Indx.Cells(2, 8) <> 0 And i > ione Then
6266              If mg > 0 Then plt.Cells(i + Indx.Cells(2, 2) - ione + 10, mg) = Miscr.Cells(i
+ 1, 20)
6267              If mm > 0 Then plt.Cells(i + Indx.Cells(2, 2) - ione + 10, mm) = Prtout.Cells(i
+ 1, 11)
6268              If mmw > 0 Then plt.Cells(i + Indx.Cells(2, 2) - ione + 10, mmw) = 1 /
Prtout.Cells(i + 1, 8)
6269              If ms > 0 Then plt.Cells(i + Indx.Cells(2, 2) - ione + 10, ms) = Prtout.Cells(i
+ 1, 7) * Miscr.Cells(2, 21)
6270              If mcp > 0 Then plt.Cells(i + Indx.Cells(2, 2) - ione + 10, mcp) =
Prtout.Cells(i + 1, 1) * Miscr.Cells(2, 21)
```

```
6271              If mgam > 0 Then plt.Cells(i + Indx.Cells(2, 2) - ione + 10, mgam) =
Prtout.Cells(i + 1, 4)
6272          End If
6273      Next i
6274      Call Thermo_Table_Format
6275      out.Cells(outint, 1) = fgi
6276      For j = 1 To Indx.Cells(2, 14)
6277          out.Cells(outint, 2 + j) = Miscr.Cells(j + 1, 20)
6278      Next j
6279      outint = outint + 1
6280  'ENTROPY
6281      Call Thermo_Table_Format
6282      out.Cells(outint, 1) = fs
6283      For j = 1 To Indx.Cells(2, 14)
6284          out.Cells(outint, 2 + j) = Prtout.Cells(j + 1, 7) * Miscr.Cells(2, 21)
6285      Next j
6286      outint = outint + 2
6287  'MOLECULAR WEIGHT
6288      Call Thermo_Table_Format
6289      out.Cells(outint, 1) = "M, (1/n)"
6290      For j = 1 To Indx.Cells(2, 14)
6291          out.Cells(outint, 2 + j) = Prtout.Cells(j + 1, 11)
6292      Next j
6293      outint = outint + 1
6294      If Not Miscl.Cells(2, 6) Then
6295          Call Thermo_Table_Format
6296          out.Cells(outint, 1) = "MW, MOL WT"
6297          For j = 1 To Indx.Cells(2, 14)
6298              out.Cells(outint, 2 + j) = 1 / Prtout.Cells(j + 1, 8)
6299          Next j
6300          outint = outint + 1
6301      End If
6302  '(DLV/DLP)T
6303      If Miscl.Cells(2, 5) Then
6304          Call Thermo_Table_Format
6305          out.Cells(outint, 1) = "(dLV/dLP)t"
6306          For j = 1 To Indx.Cells(2, 14)
6307              out.Cells(outint, 2 + j) = Prtout.Cells(j + 1, 2)
6308          Next j
6309          outint = outint + 1
6310      End If
6311  '(DLV/DLT)P
6312      If Miscl.Cells(2, 5) Then
6313          Call Thermo_Table_Format
6314          out.Cells(outint, 1) = "(dLV/dLT)p"
6315          For j = 1 To Indx.Cells(2, 14)
6316              out.Cells(outint, 2 + j) = Prtout.Cells(j + 1, 3)
6317          Next j
6318          outint = outint + 1
6319      End If
6320  'HEAT CAPACITY
6321      Call Thermo Table Format
6322      out.Cells(outint, 1) = fc
6323      For j = 1 To Indx.Cells(2, 14)
6324          out.Cells(outint, 2 + j) = Prtout.Cells(j + 1, 1) * Miscr.Cells(2, 21)
6325      Next j
6326      outint = outint + 1
6327  'GAMMA(S)
6328      Call Thermo_Table_Format
6329      out.Cells(outint, 1) = "GAMMAs"
6330      For j = 1 To Indx.Cells(2, 14)
6331          out.Cells(outint, 2 + j) = Prtout.Cells(j + 1, 4)
6332      Next j
6333      outint = outint + 1
6334  'SONIC VELOCITY
6335      For i = 1 To Indx.Cells(2, 14)
6336          Prtout.Cells(i + 1, 12) = (Miscr.Cells(2, 9) * Prtout.Cells(i + 1, 4) *
Prtout.Cells(i + 1, 9) / Prtout.Cells(i + 1, 11)) ^ (0.5)
6337          If Indx.Cells(2, 8) <> 0 And i > ione And mson > 0 Then plt.Cells(i + Indx.Cells(2,
2) - ione + 10, mson) = Prtout.Cells(i + 1, 12)
6338      Next i
```

```
6339      Call Thermo_Table_Format
6340      out.Cells(outint, 1) = "SON VEL,M/SEC"
6341      For j = 1 To Indx.Cells(2, 14)
6342          out.Cells(outint, 2 + j) = Prtout.Cells(j + 1, 12)
6343      Next j
6344      outint = outint + 2
6345
6346  End Sub
6347
6348
6349  'Writes Mole Fractions
6350  Sub OUT3_Sub()
6351      Dim tra, tem As Double
6352      Dim im, kin, notuse, m, i5, i6 As Integer
6353      Dim kok As Boolean
6354      tra = 0.000005
6355      If Miscr.Cells(2, 14) <> 0 Then tra = Miscr.Cells(2, 14)
6356  'Mass or Mole Fractions
6357      If Miscl.Cells(2, 5) Then
6358          out.Range("A" & outint, "D" & outint).Merge
6359          If Miscl.Cells(2, 9) Then
6360              out.Cells(outint, 1) = "MASS FRACTIONS"
6361          Else
6362              out.Cells(outint, 1) = "MOLE FRACTIONS"
6363          End If
6364          out.Cells(outint, 1).HorizontalAlignment = xlCenter
6365          outint = outint + 2
6366          notuse = 0
6367          For k = 1 To Indx.Cells(2, 15)
6368              kok = True
6369              If (k > CInt(Indx.Cells(2, 5)) And k < CInt(Indx.Cells(2, 15)) And
CStr(Cdata.Cells(k + 1, 10)) = CStr(Cdata.Cells(k + 2, 10))) Then
6370                  kok = False
6371                  im = 0
6372                  GoTo 120
6373              End If
6374              If pfrac Then
6375                  For m = 1 To Indx.Cells(2, 8)
6376                      im = 0
6377                      If CStr(Cdata.Cells(m + 1, 9)) = CStr(Cdata.Cells(k + 1, 10)) Or "*" &
CStr(Cdata.Cells(m + 1, 9)) = CStr(Cdata.Cells(k + 1, 10)) Then
6378                          im = m
6379                          GoTo 120
6380                      End If
6381                  Next m
6382              End If
6383  120:        kin = 0
6384              For i = 1 To Indx.Cells(2, 14)
6385                  If Miscl.Cells(2, 9) Then
6386                      tem = Therm.Cells(k + 1, 7)
6387                  Else
6388                      tem = 1 / Prtout.Cells(i + 1, 8)
6389                  End If
6390                  If k <= CInt(Indx.Cells(2, 5)) Then
6391                      Miscr.Cells(i + 1, 20) = En.Cells(k, i) * tem
6392                  Else
6393                      If CStr(Cdata.Cells(k + 1, 10)) <> CStr(Cdata.Cells(k, 10)) Then
Miscr.Cells(i + 1, 20) = 0
6394                      If En.Cells(k, i) > 0 Then Miscr.Cells(i + 1, 20) = En.Cells(k, i) *
tem
6395                  End If
6396                  If Indx.Cells(2, 8) <> 0 And i > ione And im > 0 Then
6397                      plt.Cells(10, im) = Cdata.Cells(im + 1, 9)
6398                      plt.Cells(i + Indx.Cells(2, 2) - ione + 10, im) = Miscr.Cells(i + 1,
20)
6399                  End If
6400                  If (kok And Miscr.Cells(i + 1, 20) >= tra) Then kin = 1
6401              Next i
6402              If kin = 1 Then
6403                  For i = 1 To Indx.Cells(2, 14)
6404                      Call Frac_Table_Format
```

```
6405                        out.Cells(outint, 1) = Cdata.Cells(k + 1, 10)
6406                        out.Cells(outint, 2 + i) = Miscr.Cells(i + 1, 20)
6407                    Next i
6408                    outint = outint + 1
6409                    If CStr(Cdata.Cells(k + 1, 10)) = CStr(Cdata.Cells(notuse + 1, 8)) Then
notuse = notuse - 1
6410                ElseIf CStr(Cdata.Cells(k + 1, 10)) <> CStr(Cdata.Cells(k, 10)) Then
6411                    notuse = notuse + 1
6412                    Cdata.Cells(notuse + 1, 8) = Cdata.Cells(k + 1, 10)
6413                End If
6414            Next k
6415            outint = outint + 1
6416        End If
6417        out.Range("A" & outint, "G" & outint).Merge
6418        out.Cells(outint, 1) = " * THERMODYNAMIC PROPERTIES FITTED TO " & Format(Therm.Cells(5,
2), "#,##0") & " K"
6419        outint = outint + 2
6420        If Not Miscl.Cells(2, 14) Then
6421            Call Statement_Format
6422            If Miscl.Cells(2, 9) Then
6423                out.Cells(outint, 3) = "PRODUCTS WHICH WERE CONSIDERED BUT WHOSE MASS
FRACTIONS"
6424            Else
6425                out.Cells(outint, 3) = "PRODUCTS WHICH WERE CONSIDERED BUT WHOSE MOLE
FRACTIONS"
6426            End If
6427            outint = outint + 1
6428            Call Statement_Format
6429            out.Cells(outint, 3) = "WERE LESS THAN " & tra & " FOR ALL ASSIGNED CONDITIONS"
6430            outint = outint + 2
6431            i5 = 0
6432            i6 = 0
6433            For i = 1 To notuse
6434                i6 = i6 + 1
6435                out.Cells(outint + i5, i6) = CStr(Cdata.Cells(i + 1, 8))
6436                If i6 = CInt(Parameter.Cells(3, 2)) + 1 And i <> CInt(Indx.Cells(2, 15)) Then
6437                    i6 = 0
6438                    i5 = i5 + 1
6439                End If
6440            Next i
6441            outint = outint + i5 + 2
6442        End If
6443        If Not Miscl.Cells(2, 10) Then
6444            Call Statement_Format
6445            out.Cells(outint, 3) = "NOTE: WEIGHT FRACTION OF FUEL IN TOTAL FUELS AND OF OXIDANT
IN TOTAL OXIDDANTS"
6446            outint = outint + 2
6447        End If
6448    End Sub
6449
6450    'This formats cells in the Output for statements
6451    Sub Statement Format()
6452        out.Range("C" & outint, "K" & outint).Merge
6453    End Sub
6454
6455    'This formats cells in the Debug for statements
6456    Sub Debug_Statement_Format()
6457        deb.Range("C" & debint, "K" & debint).Merge
6458    End Sub
6459
6460    'This formats the Cells for the Mixture Inputs
6461    Sub Mix Table Format()
6462            out.Range("A" & outint, "B" & outint).Merge
6463            out.Range("C" & outint, "D" & outint).Merge
6464            out.Cells(outint, 3).HorizontalAlignment = xlCenter
6465            out.Range("E" & outint, "F" & outint).Merge
6466            out.Cells(outint, 5).HorizontalAlignment = xlCenter
6467            out.Range("G" & outint, "H" & outint).Merge
6468            out.Cells(outint, 7).HorizontalAlignment = xlCenter
6469    End Sub
6470
```

```
'This formats the Cells for the Reactants Inputs
Sub Reac Table Format()
        out.Range("B" & outint, "C" & outint).Merge
        out.Range("D" & outint, "E" & outint).Merge
        out.Cells(outint, 4).HorizontalAlignment = xlCenter
        out.Range("F" & outint, "G" & outint).Merge
        out.Cells(outint, 6).HorizontalAlignment = xlCenter
        out.Range("H" & outint, "I" & outint).Merge
        out.Cells(outint, 8).HorizontalAlignment = xlCenter
End Sub

'This formats the Cells for the Thermodynamics Outputs
Sub Thermo_Table_Format()
        out.Range("A" & outint, "B" & outint).Merge
End Sub

'This formats the Cells for the Mole/Mass Fractions Outputs
Sub Frac_Table_Format()
        out.Range("A" & outint, "B" & outint).Merge
End Sub

'This formats the Cells for the Debug Output Table
Sub Debug_Table_Format()
    deb.Range("A" & debint, "B" & debint).Merge
    deb.Cells(debint, 1).HorizontalAlignment = xlCenter
    deb.Range("C" & debint, "D" & debint).Merge
    deb.Cells(debint, 3).HorizontalAlignment = xlCenter
    deb.Range("E" & debint, "F" & debint).Merge
    deb.Cells(debint, 5).HorizontalAlignment = xlCenter
    deb.Range("G" & debint, "H" & debint).Merge
    deb.Cells(debint, 7).HorizontalAlignment = xlCenter
    deb.Range("I" & debint, "J" & debint).Merge
    deb.Cells(debint, 9).HorizontalAlignment = xlCenter
    deb.Range("K" & debint, "L" & debint).Merge
    deb.Cells(debint, 11).HorizontalAlignment = xlCenter
    deb.Range("M" & debint, "N" & debint).Merge
    deb.Cells(debint, 13).HorizontalAlignment = xlCenter
    deb.Range("O" & debint, "P" & debint).Merge
    deb.Cells(debint, 15).HorizontalAlignment = xlCenter
End Sub

'This Centers a Userform for use with Multiple Monitors
Sub Form_Position(Form)
    Dim GetInfo As Variant
    #If Win32 Or Win64 Then
        With Form
          .StartUpPosition = 0
          .Left = Application.Left + (0.5 * Application.Width) - (0.5 * .Width)
          .Top = Application.Top + (0.5 * Application.Height) - (0.5 * .Height)
        End With
    #Else
        GetInfo = GetExcelPositionAndSize(ActiveWorkbook.name)
        If IsArray(GetInfo) Then
            With Form
              .StartUpPosition = 0
              .Left = GetInfo(0) + (0.5 * (GetInfo(2) - GetInfo(0))) - (0.5 * .Width)
              .Top = GetInfo(1) + (0.5 * (GetInfo(3) - GetInfo(1))) - (0.5 * .Height)
            End With
        Else
            MsgBox ("Error getting screen resolution.")
        End If
    #End If
End Sub
'This creates a Variant to get the location of the Excel Document on a Mac
Function GetExcelPositionAndSize(ByVal winName As String) As Variant
    Dim ScriptToRun As String, ScriptResult As String, MySplit As Variant

    ScriptToRun = "tell application """ & Application.name & """" & Chr(13)
    ScriptToRun = ScriptToRun + "try" & Chr(13)
    ScriptToRun = ScriptToRun + "set rect to bounds of window """ & winName & """" &
Chr(13)
```

```
6541      ScriptToRun = ScriptToRun + "on error" & Chr(13)
6542      ScriptToRun = ScriptToRun + "set rect to ""error""" & Chr(13)
6543      ScriptToRun = ScriptToRun + "end try" & Chr(13)
6544      ScriptToRun = ScriptToRun + "end tell" & Chr(13)
6545      ScriptToRun = ScriptToRun + "return rect"
6546
6547      ScriptResult = MacScript(ScriptToRun)
6548
6549      If ScriptResult <> "error" Then
6550          GetExcelPositionAndSize = Split(ScriptResult, Chr(44))
6551      Else
6552          GetExcelPositionAndSize = "error"
6553      End If
6554  End Function
```