

AD-A160 651

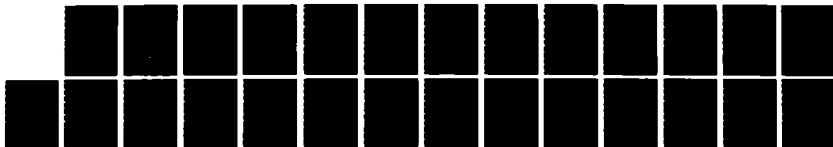
THERMOCHEMICAL MODEL CONVERSION AND USAGE IN ENERGETIC  
MATERIALS APPLICATIONS(U) NAVAL RESEARCH LAB WASHINGTON  
DC E POWELL ET AL 30 SEP 85 NRL-MR-5681

1/1

UNCLASSIFIED

F/G 9/2

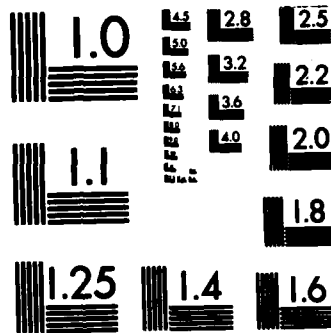
NL



END

F1860

01C



MICROCOPY RESOLUTION TEST CHART  
 NATIONAL BUREAU OF STANDARDS-1963-A

2

NRL Memorandum Report 5681

# Thermochemical Model Conversion and Usage in Energetic Materials Applications

E. POWELL AND R. B. BOGGS  
*Advanced Techniques Branch  
Tactical Electronic Warfare Division*

AD-A160 651

September 30, 1985

DTIC  
ELECTE  
OCT 25 1985  
S D  
B

DTIC FILE COPY



NAVAL RESEARCH LABORATORY  
Washington, D.C.

Approved for public release; distribution unlimited.

85 10 25 032

AD-A160651

REPORT DOCUMENTATION PAGE			
1a. REPORT SECURITY CLASSIFICATION <b>UNCLASSIFIED</b>		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION / AVAILABILITY OF REPORT	
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE		<b>Approved for public release; distribution unlimited.</b>	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) <b>NRL Memorandum Report 5681</b>		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION <b>Naval Research Laboratory</b>	6b. OFFICE SYMBOL (if applicable) <b>Code 5751</b>	7a. NAME OF MONITORING ORGANIZATION <b>Naval Electronic Systems Command</b>	
6c. ADDRESS (City, State, and ZIP Code) <b>Washington, DC 20375-5000</b>		7b. ADDRESS (City, State, and ZIP Code) <b>Washington, DC 20363-5100</b>	
8a. NAME OF FUNDING / SPONSORING ORGANIZATION <b>Naval Electronic Systems Command</b>	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code) <b>Washington, DC 20363-5100</b>		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO <b>62734N</b>	PROJECT NO. <b>XF34-375-100</b>
		TASK NO.	WORK UNIT ACCESSION NO <b>DN480-057</b>
11. TITLE (Include Security Classification) <b>Thermochemical Model Conversion and Usage in Energetic Materials Applications</b>			
12. PERSONAL AUTHOR(S) <b>Powell, E. and Boggs, R.B.</b>			
13a. TYPE OF REPORT <b>Interim</b>	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) <b>1985 September 30</b>	15. PAGE COUNT <b>32</b>
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
		Thermochemical models	
		Energetic materials	
		Combustion chemistry	
		Propellant evaluation	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>The NWC China Lake thermochemical simulation of solid fuel combustion (Propellant Evaluation Program) was converted for energetic materials applications on an NRL computer system. An overview of the model and specific steps in the code conversion are documented. A brief user's guide is provided and sample runs for simple chemistry are shown.</p>			
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION <b>UNCLASSIFIED</b>	
22a. NAME OF RESPONSIBLE INDIVIDUAL <b>Edward Powell</b>		22b. TELEPHONE (Include Area Code) <b>(202) 767-3178</b>	22c. OFFICE SYMBOL <b>Code 5751</b>

CONTENTS

I. INTRODUCTION ..... 1

II. SOFTWARE OVERVIEW ..... 3

III. INPUT FILE SETUP ..... 8

IV. PROGRAM EXECUTION ..... 13

V. SUMMARY ..... 16

REFERENCES ..... 16

APPENDIX I - Conversion Modifications in the NRL Code ..... 17

APPENDIX II - Hierarchy of Program Subroutine Calls ..... 23

APPENDIX III - Sample Runs for Thermite Combustion Case ..... 26



**S** DTIC ELECTE **D**  
OCT 25 1985

**B**

Program in Use		✓
Date		
Author		
Title		
Project		
Dist		
<b>A-1</b>		

## THERMOCHEMICAL MODEL CONVERSION AND USAGE IN ENERGETIC MATERIALS APPLICATIONS

### I. INTRODUCTION

The software described in this <sup>document</sup> ~~documentation~~ is an adaptation of the coded model of propellant combustion developed at Naval Weapon Center (NWC) China Lake, by Cruise, Villars, Browne, et al. Whereas a more complete technical description of the model can be found in reference 1 by Cruise, a brief user orientation is given here.

The program is known as the Propellant Evaluation Program (PEP). The PEP simulation is a Fortran implementation of an equilibrium model for solid fuel combustion under the processes of constant pressure, adiabatic combustion and isentropic, adiabatic expansion. This code was converted for usage on a Prime 9955 minicomputer system from an original copy of version 11978 from NWC. The NRL adaptation does not presently include all of the original I/O capabilities, but rather is structured toward the most simple and direct approach to input setup and execution. At present all I/O is terminal oriented with individual files available for lineprinter or hardcopy. The conversion has been made from Univac 1108 Fortran IV to F77. Notable steps in the conversion are described in Appendix 1.

The thermochemical model is useful in calculations of a variety of high-temperature thermodynamic properties and performance levels for solid fuel propellants. It has been employed in various flare development programs as well as in its original application of solid fuel motor characterization. The program will handle a maximum of twelve chemical elements in the initial conditions and track up to two hundred combustion products in the end state. ← Among the parameters calculated in this simulation are the flame temperature, chemical composition, enthalpy, entropy, specific heat ratio and molecular weights in both the combustion chamber and exhaust, frozen and shifting equilibrium properties, specific impulse, boost velocities, thrust coefficient, characteristic velocity, and exhaust gas velocity.

The application of this code to solid fuel characterization in ambient atmospheres has been its principle use at NRL. The theoretical grounds for validity in this somewhat restricted case have not been investigated, but have been assumed sufficient to permit utilization of the model on an ad hoc basis with certain caveats on the level of confidence to be placed in its predictions.

The assumptions of the model can be divided into those behind the combustion process and those behind the expansion process. The constant pressure adiabatic combustion process is assumed to be such that (a) reaction kinetics are sufficiently fast that the chemical equilibrium is attained long before the product species

leave the combustion chamber and become "exhaust", (b) there is no heat exchange between propellant system and surroundings, and (c) gaseous species individually follow the perfect gas law and collectively obey Dalton's law of partial pressures. For the isentropic, adiabatic expansion the assumptions are that (d) the reaction kinetics are fast enough that chemical equilibrium is maintained throughout expansion (shifting hypothesis), (e) that the chemical composition is "frozen" during the expansion due to sufficiently slow reaction kinetics, (f) a reversible expansion process exists, (g) thermal isolation of the chemical system in expansion occurs, and (h) gaseous species each follow the perfect gas and collectively follow the Dalton law (with non-gases allowed no volume).

Duhem's theorem (Chapter XIII of reference 2) states that whatever the number of phases, components, or chemical reactions, the equilibrium state of a closed system for which we know the initial masses is completely determined by only two independent variables. The specific way in which all of these assumptions are utilized is best described in reference 1. Suffice it to say here that the approach uses enthalpy balance to determine the thermochemical state after combustion (under assumptions a-c) and entropy balance is utilized to attain solutions to the expansion problem (assume d-h). The two independent variables are temperature and pressure, with pressure as a given value and an iterative procedure of successive temperature guesses invoked for each distinct process (enthalpy balance for combustion, entropy balance for the expansion).

In this model the numerical iteration combines Newton's method with an interval-halving method as override. Whereas Newton's method is very fast, there is no guarantee of its convergence. For the interval-halving method the convergence is guaranteed if the answer is contained within the original limits, thus offering great advantages despite the relative slowness of the computational technique.

Future documentation will address the specific applications and determinations provided by the PEP simulation. These will involve energy optimization, species population tuning, and investigation of new solid fuel formulations. This memorandum will be limited to a description of the model from the user standpoint as well as the documentation of the specific changes involved in the conversion.

## II. SOFTWARE OVERVIEW

The PEP model consists of three distinct groups of software and can be operated in several different ways. The software groups consist of main, auxiliary, and library files which are respectively referred to as the PEPMAIN, PEPAUX, and PEPLIB areas. Figure 1 illustrates the organization of this software within the GLOBLE sub-directory PEP. The purposes of these groups are as follows.

The PEPLIB program allows the user to enter ingredient data by serial number if the species are in the existing library file. This file may be augmented with serial assignment of new numbering. The software for the library is a brief program which formats data and writes a file referred to as TAPE11 which can be accessed by the main program PEPMAIN if so desired by the user. As presently implemented the PEP code described here has available to it the generated file TAPE11, but this is not the most common mode of usage since the application at NRL usually involves new formulations. Figure 2 shows the generation of TAPE11 by PEPLIB. (TAPE11 is not actually a tape; however, this filename is retained to conform to previous documentation.)

The PEPAUX software is a preprocessing and updating program which accomplishes several organizational functions for the species data. These include generation of Hollerith names for each species and ordering of the species, dependent upon their physical phase. The gases are ordered first with condensed species following so as to save computation time when the PEPMAIN equilibrium program uses the PEPAUX file, here referred to as TAPE12. Figure 3 illustrates PEPAUX and its sub-programs which generate TAPE12 to be used by PEPMAIN.

In the remainder of this document the term "manual input" refers to the input stream which is described in Appendix A of reference 1. Similarly the words "card" and "deck" may be interpreted as "line" and "file" throughout this document.

The relationship between PEPMAIN and its I/O paths is summarized in Figure 4. The manual input option actually is the primary approach utilized at NRL to date for reasons of convenience. This input and execution path will be described here. Other I/O techniques such as that of the ingredient file serial specification are described in reference 1.



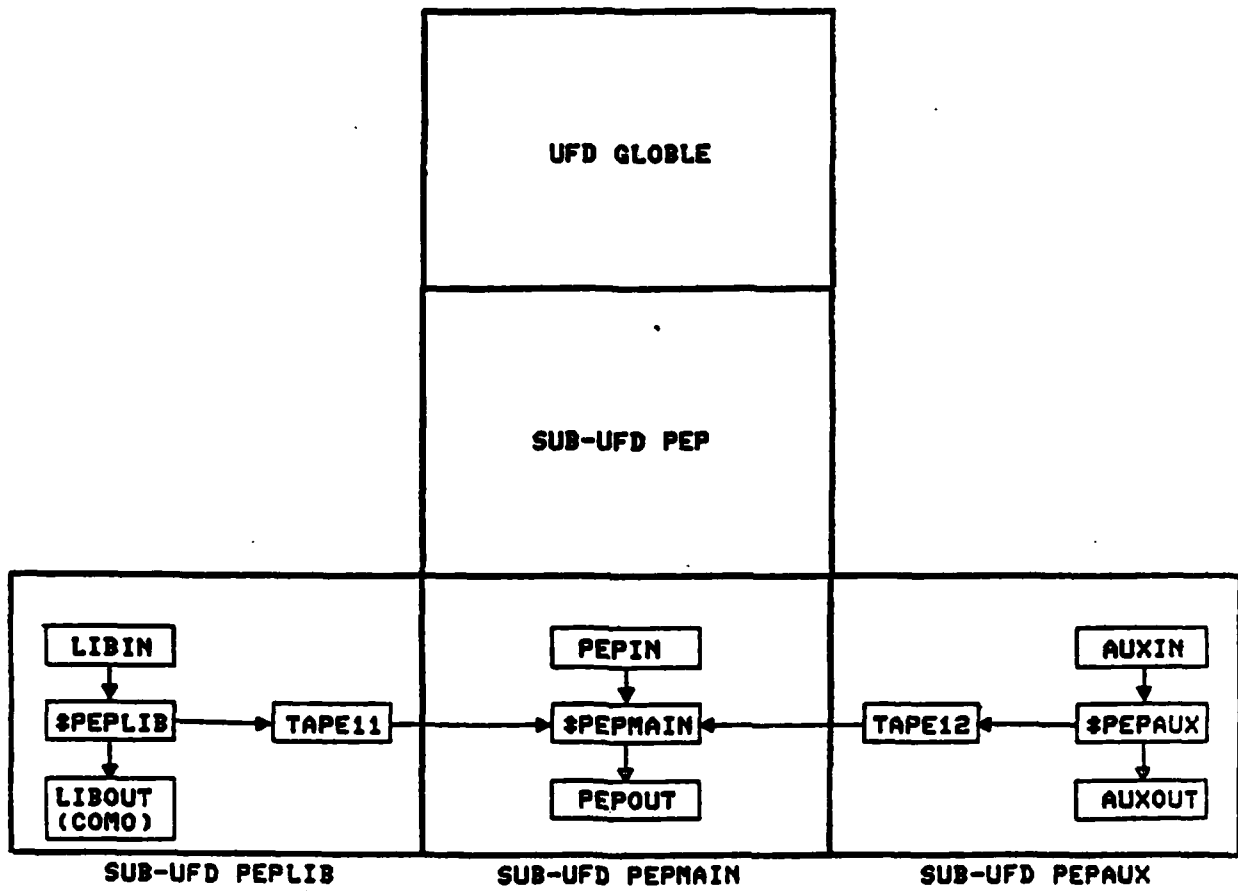


Figure 1. PEP Software Organization

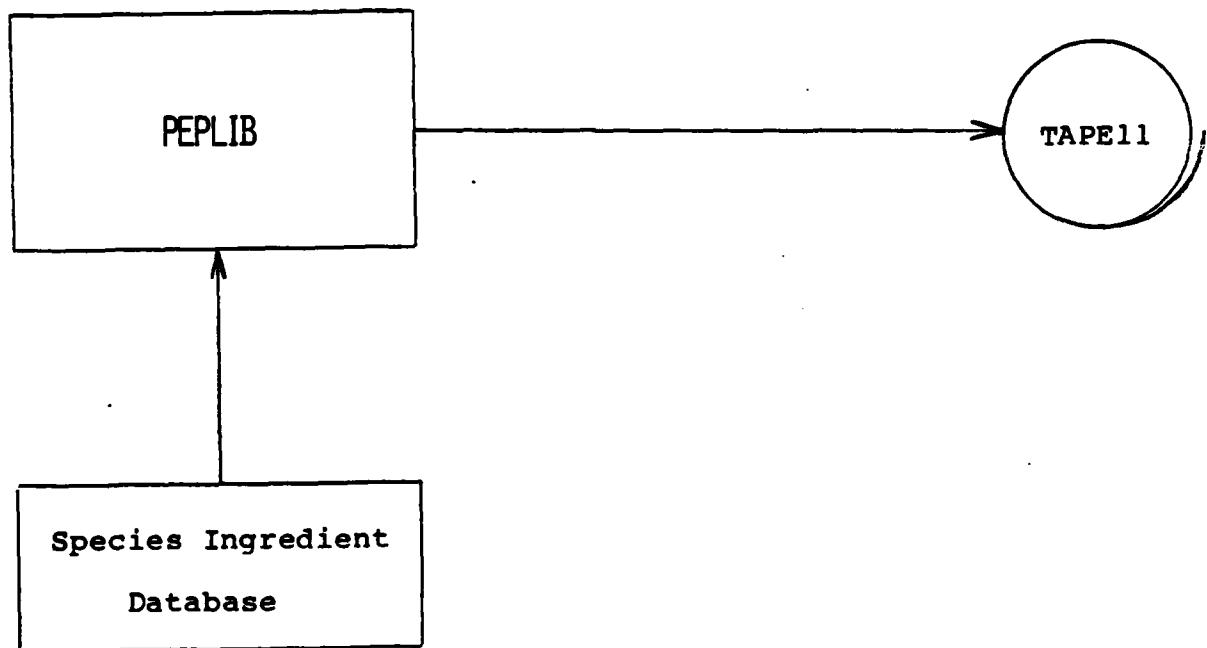


Figure 2. PEP Ingredient Tape Generation

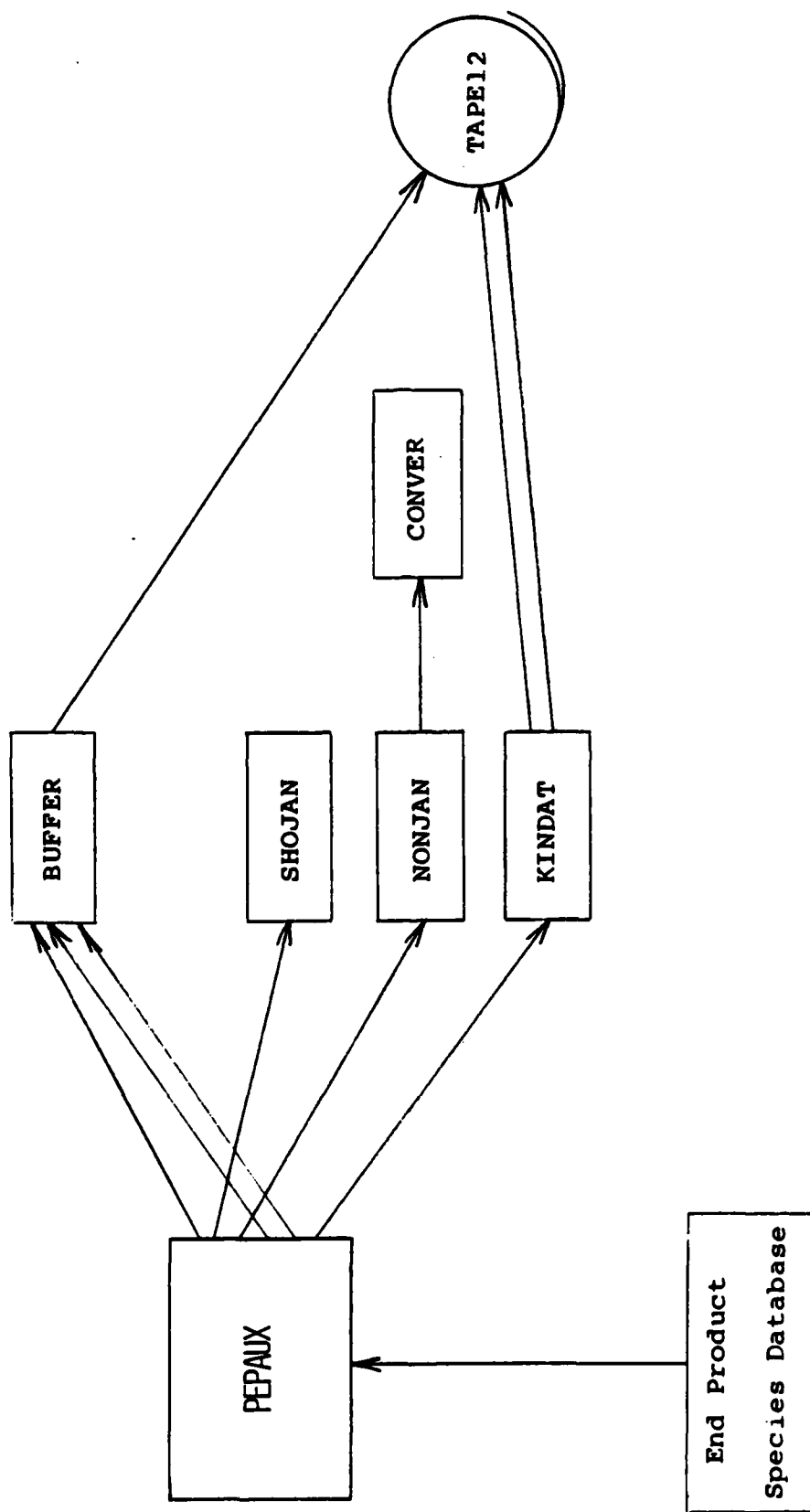


Figure 3. End Product Tape Generation

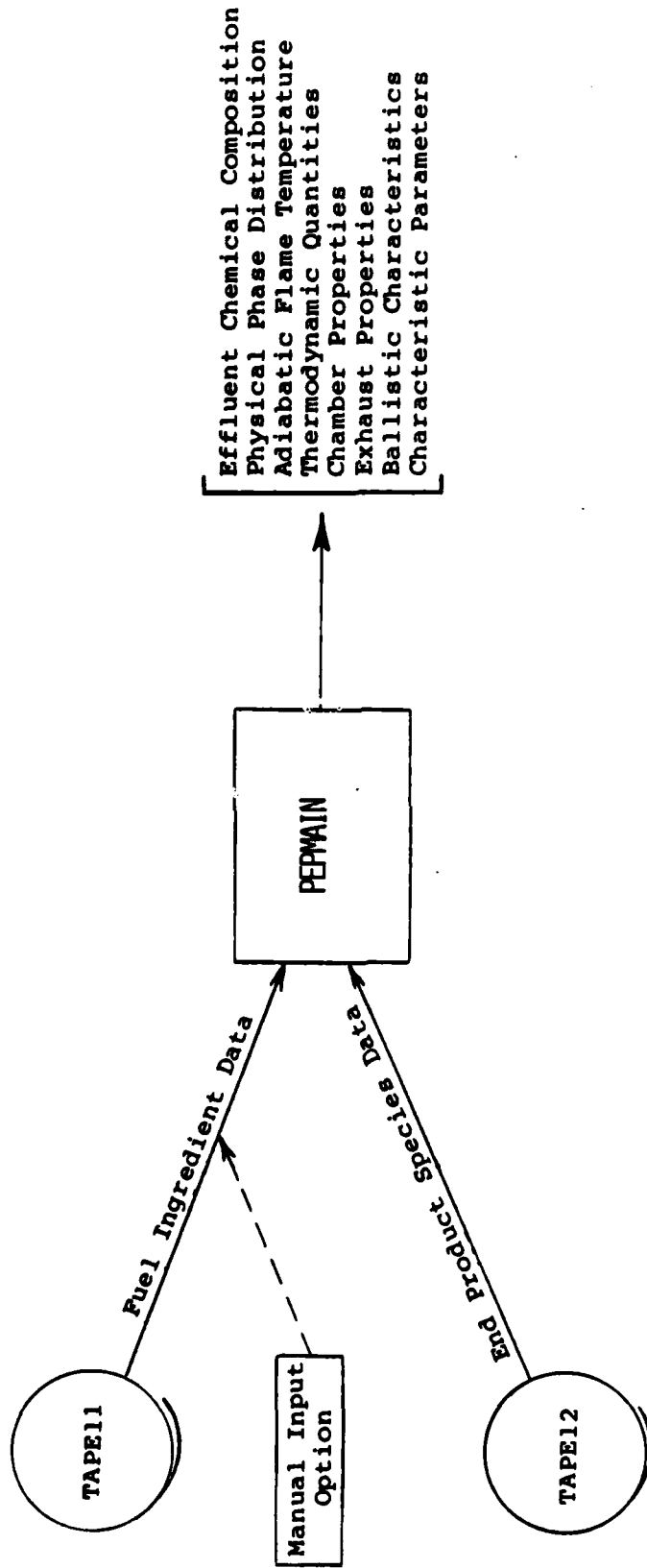


Figure 4. Main PEP Program Operation

### III. INPUT FILE SETUP

To operate the PEP simulation in the manual input mode the user must prepare an input file in the format described on pages 23-24 of reference 1. For convenience that description is repeated here.

There are three groups of input cards in the batch input file: (1) the control card, (2) the ingredient composition card(s), and (3) the pressure and weight ratio card(s).

The run control card has a forty column field consisting of option switches, user name, number of ingredients involved, and the number of runs to be made on that system of ingredients. The option switches (columns 1-19) available are shown in Table 1. The first six letters of the user appear next in the field (columns 21-26). Columns 29-30 contain the number of propellant ingredients (not to exceed 10): this number will agree with the number of ingredient composition cards to follow (sans decimal point). Ending in column forty is the number of runs to be made for the system (no decimal). (However, the capability of multiple runs is not implemented in the NRL adaptation and this field is always set to one.)

For the second grouping (ingredient composition cards) the format of each card is described in Table 2. Note that the last item in the table (columns 69-73) may be omitted if boost velocities and density impulse are not required.

An example of an ingredient composition card is as follows:

```
AMMONIUM DICHROMATE      8H   2N   70   2CR   -1688   .0776
```

Note that it is permissible to introduce arbitrary multipliers into the composition so that, for example, the following is equivalent to the card above:

```
AMMONIUM DICHROMATE      16H   4N   140   4CR   -1688   .0776
```

In the case of mixtures a single card can be used to enter the constituents as single ingredients (as if it were a compound). The common example of atmospheric air is entered as follows:

```
AIR (DRY AT SEA LEVEL)  835N 2240   5AR           0000
```

In the third grouping (pressure and weight ratio) each card will consist of 12 six-column fields. The first field contains the chamber pressure, whereas the second contains the exhaust pressure. Following these are the consecutive weight ratios for the propellant ingredients in the same order as they appear in the ingredient composition cards. There will be as many cards as there are fuel ingredients. Conventionally the weights are chosen to add up to 100 grams, although this is not mandatory. It should be noted that decimal points must be punched in all fields used on the

pressure and weight ratio cards.

A completed manual input deck is shown in Figure 5. This is the same case as described in reference 1. It should be noted that, as mentioned above, in this adaptation of the code, the multiple run option in column 40 of the run control card is not used.

Table 1. PEP Program Control Option Switch Settings

<u>Option</u>	<u>Type</u>	<u>Function Performed</u>
1	1	Deletes exit calculations
2	1	Includes ionic species in the calculations
3	1	Deletes boost velocities and nozzle design data
4	1	Inputs pressures in psi instead of atmospheres
5	1	Increases precision of species concentrations 1 magn.
5	2 or higher	Increases precision even further
6	1	Inputs an extra identification card
7	1	Inputs a pressure-temperature point instead of chamber and exhaust pressure - allows a P-T-H-S chart
8	1	Outputs a list of all combustion species considered
9	1	Allows serial number input for ingredients (PEPLIB)
10	1	Allows modification of H and p data
Options 11-15 are used only for debugging		
11	1	Prints out thermo data computed at every temperature guess
12	1	Prints out first guess of the composition
13	1	Prints out compositions every fourth iteration
14	1	Prints out the log of the equilibrium constants at every temperature guess
15	1	Outputs a code that indicates the classification the program has applied to various species at each iteration
16-19	Leave Blank	For internal use

Table 2. Column Formats of Ingredient Composition Cards

<u>Column</u>	<u>Item Description</u>
1-30	Name of ingredient (alphanumeric)
31-33	Number of atoms in first element in compound (punch no decimal)
34-35	Symbol of first element (left adjust)
36-38	Number of atoms of second element in compound
39-40	Symbol of second element and so on as needed up to six elements and column sixty
63-67	Heat of formation of compound in calories/gram (right adjust with no decimal)
69-73	Density of compound in pounds/cubic inch (with decimal) (This item optional if boost and impulse unwanted)



0011000000 RBB 2 1  
SULPHUR 1S  
MOLASSES 22H 12C 110  
1000. 14.7 10. 90. +0000 .0474  
-1550 .0574

Figure 5. Manual Input File

#### IV. PROGRAM EXECUTION

To execute the PEP simulation the user should type in the following command:

```
SEG GLOBLE>PEP>PEPMAIN>#PEPMAIN
```

The program will then ask for the name of the input file as follows:

```
INPUT TREENAME OF MAIN INPUT FILE:
```

After the filename has been entered, the program runs to completion and the output is available in file PEPOUT. (This procedure is used for both the manual and automated input of ingredient data. However, the formats of the input file differ.)

Figure 6 shows the combustion product output file PEPOUT which results from program execution. The printout is in the form of five general categories of information including an ingredient weights summary, gram atoms summary, chamber results, exhaust results, and design performance results. It should be noted that in the chamber and exhaust results the predictions include temperature, pressure, and gas parameter information as well as predictions of the number of moles of each product species per 100 grams of fuel. The physical phase of the species is denoted by a special character following it, unless it is a gas (which has a blank). Solid products are denoted by "\$", and liquids are denoted by "\*".

As a matter of convenience the precise formatting of the output file (and its units) is described here.

The first four lines contain CPU information, a code version record, column indicators, flags for number of ingredients (M) and number of runs (N), user ID, and labeling for heat of formation (DH) and composition of the ingredients. The fifth line starts a list of ingredients with their heat of formation (cal/gm) and molecular composition following. After the ingredient lines come a labeling (header) line which briefly states the order of the rest of the output to come ("INGRED.WTS./ GRAM ATOMS/ CHAMBER/..", etc.).

The following line contains the ingredient weights and the total system weight, which is conventionally chosen to be 100 grams. If another total is chosen it will affect the value of other outputs to come. The gram atom amounts of each ingredient chemical element come next (based on a given system weight).

Following this come the most important output results of the simulation. These are in terms of predicted conditions and species populations, firstly in the combustion chamber, and then in the exhaust region. For each set of outputs there is a labeling header ("T(K) T(F) ...") for the conditions, followed by several lines

0(CPU 0.00SECS.)

11978 VERSION OF PEP.

01234567890

1 RBB

(NAME)

M

N

DH

COMPOSITION

SULPHUR 0 1S  
MOLASSES -1550 22H 12C 11O  
INGRED.WTS.&TOTAL/ GRAM ATOMS/ CHAMBER/ EXHAUST RESULTS/ PERFORMANCE

10.00000 90.00000 100.00000

5.784264 H 3.155053 C 2.892132 O 0.311857 S

T(K)	T(F)	P(ATM)	P(PSI)	ENTHALPY	ENTROPY	CP/CV	GAS	RT/V
850.	1071.	68.02	1000.00	-139.50	169.12	1.1664	3.169	21.465

1.75954	C\$	1.26287	H2O	0.79301	CO2	0.55911	CH4
0.30977	H2S	0.20104	H2	0.04116	CO	0.00208	CSO
6.68E-05	C2H6	1.25E-06	CS2				

T(K)	T(F)	P(ATM)	P(PSI)	ENTHALPY	ENTROPY	CP/CV	GAS	RT/V
501.	442.	1.00	14.70	-156.92	169.12	1.2045	3.059	0.327

2.15011	C\$	1.72023	H2O	0.58590	CO2	0.41893	CH4
0.31181	H2S	0.02221	H2	0.00005	CO	0.00004	CSO

OIMPULSE	IS	EX	T*	P*	CF	ISP*	OPT	EX	D-ISP	A*M.	EX T
----------	----	----	----	----	----	------	-----	----	-------	------	------

120.2	1.1938	775.	38.48	1.625			8.98	187.1	0.07401	429.
123.1	1.1453	797.	39.14	1.627	93.4	9.67	191.6	0.07568	501.	

INGRED. DENSITIES ARE

0.0474 0.0574

0(CPU 0.00SECS.)

11978 VERSION OF PEP.

01234567890

(NAME)

M

N

Figure 6. Sample PEP Output File

listing the species populations (gases, liquids, and solids) in units of moles per 100 grams of ingredients (or per whatever was the total system weight). If one prefers partial pressures, multiply each composition by "RT/V" described below (for gases).

Units for the outputs here are as follows. The enthalpy is in units of calories (per system weight) and the entropy has units of calories/K per system weight. The quantity "CP/CV" is the ratio of specific heats (net ratio of partial heat capacities of the gas volume in the traditional sense) and "GAS" is the total number of moles of all gases produced per system weight (the proper number for all gas dynamics calculations). The quantity "RT/V" is the variable designated by "A" in reference 1 and is given by

$$\text{"RT/V"} = \frac{R (0.08205 \text{ l-atm/mole/K}) T(K)}{V (\text{system volume in liters})}$$

Exhaust plane results follow those of the chamber and are in the same units and format.

Performance results follow the exhaust output in three lines of information. The first is headings. The second has results for a frozen flow (no chemical reactions) through the nozzle. The third contains the results for a shifting flow (reactions in equilibrium) through the nozzle. Impulse has units of seconds. (This quantity is what was previously called the "theoretical exhaust velocity". To obtain the conventional SI impulse multiply this value by 9.806 m/sec).

The quantity "IS EX" is the isentropic constant  $c$  such that

$$PV^c = \text{constant}$$

for isentropic flow near the nozzle throat. Note that the values of IS EX and CP/CV do not agree because the gas is not perfect.

The variables  $T^*$  and  $P^*$  are throat temperature (in K) and pressure (atmospheres). The variable "CF" is the nozzle thrust coefficient. (If one desires the characteristic velocity  $C^*$  one can obtain it from  $C^*=32.17 \text{ ISP/CF}$ ). "ISP" is the vacuum impulse from a sonic nozzle. The optimum expansion ratio "OPT EX" is the ratio of the nozzle exit area to throat area at which exit pressure equals ambient pressure. Following this are the density impulse "D-ISP", and the exit plane temperature "EX" (in K). Then comes "A\*M" which actually is  $A^*/M$ , the ratio of nozzle throat area to the mass flow rate expressed as sq.in.-sec/lb.

Optional outputs are available and are found in reference 1.

## V. SUMMARY

This program is currently utilized within NRL Code 5751 and individuals interested in the adaptation described here may contact the authors of this report. It should be noted that no new model capabilities have been added to the simulation and the work described here is documented mainly for the convenience of the simulation user. The Univac code version used for this work was kindly supplied by D.R. Cruise of Naval Weapon Center. For more recent versions of the code one should contact the original NWC authors of the model. It is hoped that this report may offer some assistance to others converting or utilizing this model.

## REFERENCES

1. D.R. Cruise, "Theoretical Computations of Equilibrium Compositions, Thermodynamic Properties, and Performance Characteristics of Propellant Systems," Report NWC TP 6037, April 1979, China Lake, CA.
2. I. Prigogine and R. Defay, Chemical Thermodynamics. Longmans, Green, and Co., London, 1954.
3. B. Boggs, "Organization of PEP Software Package," NRL Code 5751 Internal Memorandum, July 6, 1985.

## APPENDIX I: Conversion Modifications in the NRL Code

This appendix documents the specific code line changes that were made to the NWC version 11978 in the conversion to F77 for the NRL Prime 9955 system (Code 5751). Code changes were required for the PEPMAIN, PEPAUX, and the PEPLIB software components and they are respectively listed here.

### Changes made to PEPMAIN software:

1. Created inserts C.A and C.IBRIUM as follows:

```
OCOMMON A(12,12), KR(20), AMAT(10,12), JAT(12), ASPEC(12), IN, IS,  
1FIE(10,6), IE(10,6), ALP(12), W27, N, BLOK(10,7), DH(10),RHO(10),  
2ISERI(10), WATE(10), W1(6), W43, IG, NP, VNT(201), W47, NAME, SER  
3,FLOOR,ITAG(100),WING(10)
```

```
OCOMMON /IBRIUM/ TL(200,2), TU(200,2), W3(200), VNU(200,12), QA,  
1TAU, H(200), SD(200), Y(200), JC, IR(200,2), DMU(200), VLNK(200),  
2IOJ(12), RA(200,2), RB(200,2), RC(200,2), RD(200,2), RE(200,2),  
3RF(200,2),CH(200,2),JM,W48,CP,FN,C(12,200),SPECIE(200),ADHOC(200)  
4,LLL(200)
```

2. In COMMON /IBRIUM/ above, changed LL(200) to LLL(200) so that it could be used throughout. (LL is sometimes used locally.) Added ADHOC(200) array.
3. Changed all REAL\*4 variables to double precision (except for SPECIE and ADHOC) with the following insert, C.IMPLICIT:

```
IMPLICIT REAL*8 (A-H,O-Z)  
REAL*4 SPECIE,ADHOC
```

SPECIE and ADHOC arrays are left single precision to interface with the form of the binary input read from TAPE11.

NOTE: Changing to double precision required extensive changing of intrinsic function names throughout the program. (ALOG to DLOG, etc.)

4. In subroutine PUTIN, commented out the following calls:

```
CALL DATE  
CALL LKCLKS  
CALL SETCLK (Also removed from main program.)  
CALL TOPDAY
```

PRIME equivalent calls for date, time of day, etc. could be added later.

5. Added code to open files as follows in the main program:

Fortran Unit:	File Unit:	Purpose:
6	2	Main output file PEPOUT.
11	7	TAPE11 input file from PEPLIB.
12	8	TAPE12 input file from PEPAUX.
13	9	Main input file.

These files are closed either in the main driver or in subroutine DEFIOJ.

Changed all READ statements with Fortran unit number 5 to 9.

6. In subroutine PUT IN, changed formats of statements numbered 2, 222, and 83. Changed 5A6 to 7A4,2x. Changed BLOK(10,5) to BLOK(10,7) in common. (This was done prior to changing to double precision and may not have been necessary.)
7. In subroutine STOICH, changed the data statement for the 100 chemical symbols to correspond to a 4-byte word rather than 6-byte. (This is the SYMB data statement.)
8. Restored subroutine SLITE with entry point SLITET using the form in the book. Switched to F77 which allows entry points. Added statement:

```
DATA LIT/0,0,0,0/
```

9. In subroutine SEARCH, changed:

```
CALL TAPEB(1,0,0,0)

to

IZERO=0
CALL TAPEB(1,0,IZERO,IZERO)
```

This prevents corruption of the constant "0" through the calling sequence by a returned variable. This may not have really been necessary since execution is brief when the first parameter is "1".

10. In subroutine TAPEB, added a new R\*4 array, BIN2(20) to contain the last two characters of REDUND which were lost when the word length dropped from 6 to 4. This new array is read from the TAPE12 interface and PEPAUX subroutine BUFFER was correspondingly modified. Also added the array ADHOC(200) to common as part of this change to hold these two characters. (This was done prior to changing all of the real defaults

from REAL\*4 to REAL\*8 via an inserted IMPLICIT statement and possibly could be improved. However, this is a tricky part of the PEP package and anything done here must be done very carefully and slowly.)

11. Modified subroutine OUT to print these two characters from array ADHOC. Needed to add local array SPOT2 in OUT to do this.
12. Modified MAIN and subroutine PUT IN to terminate the program if end-of-file is encountered at the first read statement using Fortran unit 13. This is the normal exit. This version will not make multiple runs from a single execution as a result of this change. (Our version is not to be run interactively: a file of input data is prepared in a format such as the one at the bottom of page 24 of the blue PEP document. The name of this file is then entered upon query.)

For automated input of ingredients this file must, of course, be correspondingly formatted.

It should be pointed out that the lines on page 24 beginning with "-RUN ...", "-ADD ...", and "-FTN ..." are not to be included in the input file.

13. In subroutine DESIGN, changed the index within two WRITE statements (those using formats 23 and 24) from "I" to "IJK". Since "I" was in the calling sequence this corrupted the constant 2 in MAIN through the calling program.
14. Added statement in MAIN to delete PEPOUT if it exists prior to opening it.
15. In subroutine EQUIL, removed unused COMMON/MOON/.
16. In subroutine TABLO, added IRUN to COMMON/MOON/ ... to make length of common area uniform.
17. In subroutine TSALT, added XXXXXX, IRUN to COMMON/MOON/ to make length uniform.
18. In MAIN, changed 1 to "KON1" and 0 to "KON0" in several places and loaded them before using as follows:



```
CALL EQUIL (TE, PR, HE, SE, 1)
      CHANGED TO
      KON1=1
CALL EQUIL (TE, PR, HE, SE, KON1)
```

```
14 CALL H BAL (TE, PR, SYSENT, 1)
      CHANGED TO
14 KON1=1
      CALL H BAL (TE, PR, SYSENT, KON1)
```

```
114 CALL OUT (PR, TE, HE, SYSENT, 1)
      CHANGED TO
114 KON1=1
      CALL OUT (PR, TE, HE, SYSENT, KON1)
```

```
      CALL S BAL (TE, PR, HE, SYSENT, TCH, 0)
      CALL DESIGN (TE, PR, HE, SYSENT, 0, 2)
22 TE = .5*(TCH+TE)
70 CALL S BAL (TE, PR, HE, SYSENT, TCH, 1)
      CHANGED TO
      KONO=0
      CALL S BAL (TE, PR, HE, SYSENT, TCH, KONO)
      CALL DESIGN (TE, PR, HE, SYSENT, 0, 2)
22 TE = .5*(TCH+TE)
70 KON1=1
      CALL S BAL (TE, PR, HE, SYSENT, TCH, KON1)
```

19. In subroutine DEFIOJ, changed the variable named "IF" to "IJF" throughout.

Changes made to PEPAUX programs:

1. Commented out statement END FILE 12 in main program of PEPAUX.
2. Added code to open and close 5 files using names and file numbers as follows:

Fortran Unit:	File Unit:	Name:	
13	9	AUXIN	Input file.
18	14	TAPE28	Temporary file.
19	15	TAPE29	Temporary file.
6	2	AUXOUT	ASCII output file, not used elsewhere.
12	8	TAPE12	Principal interface into PEPMAIN.

Added code to delete files TAPE28 and TAPE29 at end of run.

3. Created an common area insert GLOBLE>PEP>PEPCOM>C.PAUX as follows and replaced existing common statements where used.

```
COMMON /PAUX/ IE(101), HI(101,2), IN(101), HK(50,2), KN(50), JN(7),  
1,JE(7), OUT(22), SPEC(5), IS(5), PARA(20),REDUND(3,7777), JD, NJD
```

In this insertable common statement, changed REDUND(2,7777) to REDUND(3,7777) as shown above. This was done to preserve 2 characters which would have been lost when changing machine wordlength from 6 to 4 characters.

4. Changed various FORMAT statements from "A6" to "A4". (i.e. format statements associated with REDUND array.)
5. In subroutine KINDAT, changed PARA(20) from single to double precision. Changed following IF statement:

```
IF(PARA(1).EQ.0.) GO TO 210  
CHANGED TO  
IF(PARA(1).EQ.0.D0) GO TO 210
```

6. In subroutine NONJAN, changed:  
18 IAB = ABS(JN(1))  
CHANGED TO  
18 IAB = IABS(JN(1))
7. In subroutine BUFFER, added a new array BIN2(20) to hold the 2 characters which would have been lost from REDUND(1,...) etc. Also added a new parameter, REDUND(2,...), to the calling sequence of BUFFER for the same purpose.

Following changes were made to the PEPLIB program:

1. Added code to use file numbers and names as follows:

Fortran	File	Unit:	Name:	
13	9		INLIB	Input file.
11	7		TAPE11	Output. Principal interface into PEPMAIN.

2. Changed:

```
DIMENSION A(20),B(20)
      TO
REAL*8 A(20),B(20)
```

3. Changed:

```
ENCODE(19,B) A(11)
      TO
ENCODE(6,19,B) A(11)
```

## APPENDIX II: Hierarchy of Program Subroutine Calls

For the convenience of the user a hierarchy diagram of the main PEP program is shown in Figure AII.1. The figure is not a flowchart but does prove beneficial when the data flow is studied from the standpoint of calling structure.

The PEP subroutines are briefly described as follows.

PEP - The main driver program which manages the computation.

PUTIN - The main input routine.

STOIC - Preliminary analysis of elementary composition.

SEARCH - Searches combustion data (TAPE12) for pertinent species.

GUESS - Computes initial guess of elementary composition.

SLITE, SLITET - Simulated lights (flags) are setup here.

GIBBS - Computes enthalpy, entropy, and Gibbs energies for species.

DEFIOJ - Computes optimal basis.

LINDEP - Establishes linear independence of basis.

ADJUST - Corrects errors in gram-atom balance due to truncation.

REACT - Computes stoichiometric coefficients and equilibrium constants.

RANK - Sorts an array into decreasing order of size.

SETUP - Preliminary analysis of equilibrium situation, computing maximum and minimum shifts in concentration so that negative concentrations do not occur.

EQUIL - Computes composition for a pressure-temperature point.

FIXBAS - Fixes basis to compensate for phase changes that occur due to temperature change.

TWITCH - The main equilibrium routine (see flow in reference 1).

THERMO - Computes system enthalpy and entropy.

TWID - Computes equilibrium relation for TWITCH to modify.

TABLO - Updates optimal basis with tableau method of linear programs.

HBAL - Computes constant pressure combustion (P-H point).

OUT - Outputs temperature and composition.

SBAL - Computes isentropic exhaust state (P-S point).

DESIGN - Computes and outputs performance parameters.

TSBAL - Fast equilibrium computation for specified temperature and entropy (T,S). Occasionally fails to converge.

TSALT - Computes a T,S point by slow but reliable method when TSBAL fails.

ONED - One-dimensional flow calculations.

BOOST - Computes and outputs boost velocities.

DESNOZ - Outputs nozzle performance.

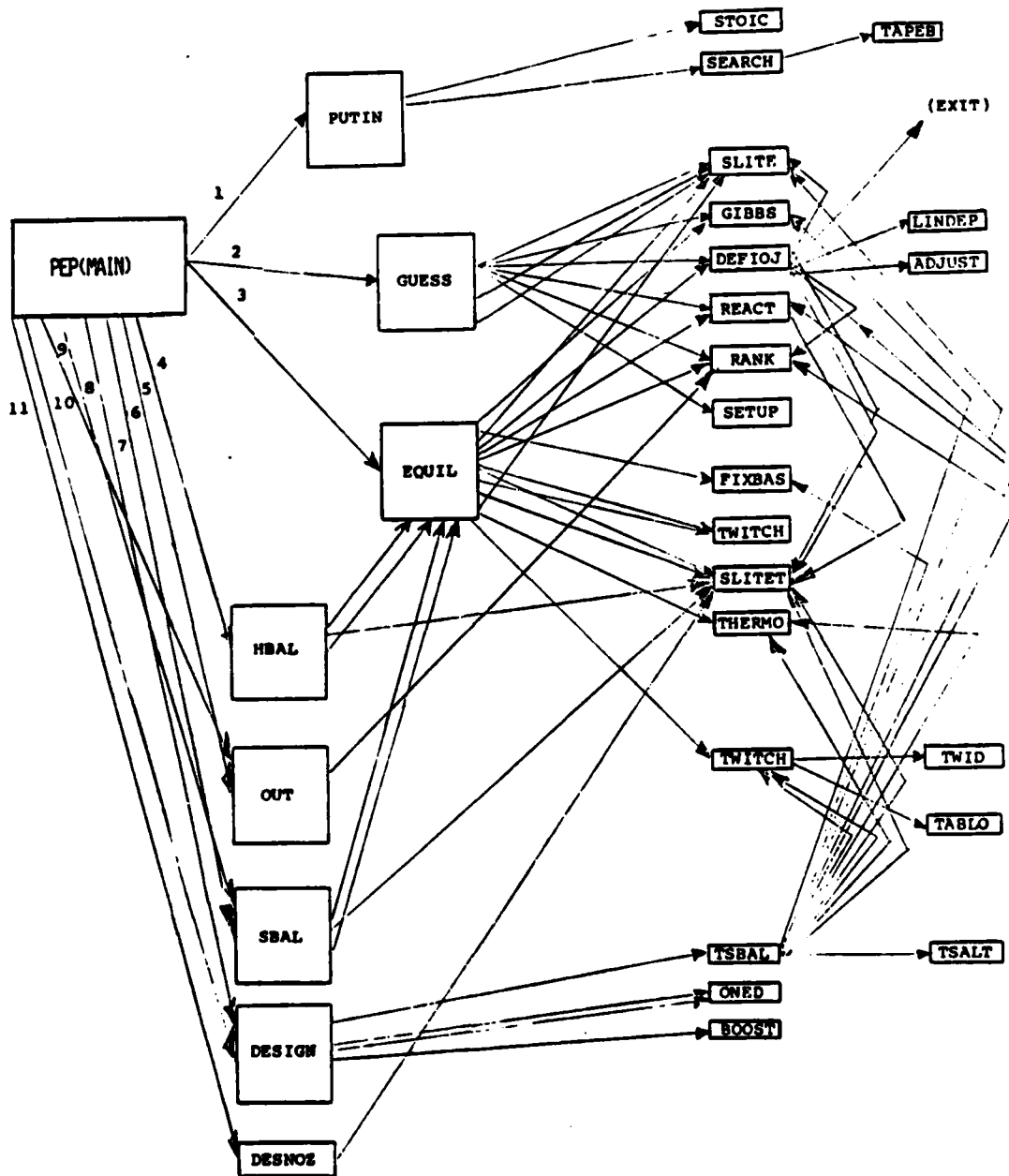


Figure AII.1. PEPMAIN Program Hierarchy

### APPENDIX III: Sample Runs for Thermite Combustion Case

As another example of the model outputs the thermite reaction for the reduction of iron is chosen as an example. The process is described by the Goldschmidt reaction:



For these ingredients the heat of formation (cal/gm) and density (lb/cu.in.) for ferric (Iron III) oxide were taken respectively to be -1230 and .1840. For aluminum the corresponding quantities are +0000 and .0976.

Figure AIII.1 illustrates the input file for this execution and Figure AIII.2 is the resultant output file. Of note is the high flame temperature of 3531 degrees Kelvin, as well as the dominance of liquid Fe and Al<sub>2</sub>O<sub>3</sub> in the species predictions, as expected from basic considerations. (Small amounts of trace species such as FeO are characteristic of this simulation and are to be expected).

00110001000000	NAMES	2	1		
ALUMINUM			1AL		+0000 .0976
FERRIC OXIDE			30	2FE	-1230 .1840
50.	14.7	25.2	74.8		

Figure AIII.1. Thermite Combustion Input File



0(CPU 0.00SECS.)  
 11978 VERSION OF PEP.  
 01234567890 (NAME) M N  
 OCOMPLETE SPECIES LIST FOLLOWS  
 AL ALO AL2O AL2O2 O O2 O3 FE FEO AL\* AL2O  
 3\$  
 AL2O3\* FE\$ FE\$ FE\$ FE\$ FE\* FEO\$ FEO\$ FEO\* FE2O3\$ FE2O  
 3\$  
 FE3O4\$ FE3O4\$

1 NAME DH COMPOSITION  
 ALUMINUM 0 1AL  
 FERRIC OXIDE -1230 30 2FE  
 OINGRED.WTS.&TOTAL/ GRAM ATOMS/ CHAMBER/ EXHAUST RESULTS/ PERFORMANCE

25.20000 74.80000 100.00000

1.405135 O 0.934025 AL 0.936756 FE

T(K)	T(F)	P(ATM)	P(PSI)	ENTHALPY	ENTROPY	CP/CV	GAS	RT/V
3531.	5896.	3.40	50.00	-92.00	74.42	1.0000	0.000\$	61465.9

0.93259	FE*	0.46701	AL2O3*	0.00408	FEO*	0.00007	FE
5.23E-06	FEO						

T(K)	T(F)	P(ATM)	P(PSI)	ENTHALPY	ENTROPY	CP/CV	GAS	RT/V
3530.	5894.	1.00	14.70	-92.00	74.42	1.0000	0.000\$	12308.2

0.93232	FE*	0.46701	AL2O3*	0.00405	FEO*	0.00034	FE
2.61E-05	FEO	4.29E-06	O	1.16E-06	AL		

OIMPULSE	IS	EX	T*	P*	CF	ISP*	OPT	EX	D-ISP	A*M.	EX T
0.9	1.0000	3531.	2.06	1.247			1.32	3.7	0.01433	3531.	
0.2	0.4420	3530.	2.73	0.056		3.4	4.11	1.0	0.08326	3530.	

OINGRED. DENSITIES ARE

0.0976 0.1840

0(CPU 0.00SECS.)  
 11978 VERSION OF PEP.

01234567890 (NAME) M N

Figure AIII.2. Thermite Combustion Output File

**END**

**FILMED**

**12-85**

**DTIC**